2015

Lattice Boltzmann Method and Its Applications in Soft Matter

Jifu Tan
Lehigh University

Lattice Boltzmann Method and Its Applications in Soft Matter

by

Jifu Tan

Presented to the Graduate and Research Committee

of Lehigh University

in Candidacy for the Degree of

Doctor of Philosophy

in

Mechanical Engineering

Lehigh University

May, 2015

Approved and recommended for acceptance as a dissertation draft.

Date

Dissertation Adviser

Accepted Date

Committee Members:

Dr. Donald Rockwell

Dr. Alparslan Oztekin

Dr. Edmund Webb

Dr. Michael Spear

Dr. Yaling Liu

Acknowledgements

Firstly, I want to thank my adviser Dr. Yaling Liu for the continuous support of my PhD study and research. His mentoring in research, his hands-on experience in programming, and his patience and motivation helped me a lot in writing the dissertation. Without his guidance and support the dissertation would have not been possible.

Secondly, I would like to thank all the professors who lead me to study fluid mechanics, solid mechanics and how to solve practical engineering problems. In particular, I am grateful to Prof. Rockwell, Prof. Webb, Prof. Oztekin, and Prof. Spear for serving as my dissertation committee member. Their insights, comments, and discussions are very helpful for the doctoral research. I am also grateful to Dr. Keller for the collaboration on Lattice Boltzmann code developing project. I would thank Dr. Li Tao for his hands-on training for me on microfabrication, Dr. Yuan Wan for the collaboration on circulating tumor cell detection. Many thanks to my lab mates including but not limited to Antony Thomas, Shunqiang Wang, Ran He, Wentao Shi, Christopher Uhl, Salman Sohrabi, Doruk Yunus, Yue Xiao, Jia Hu, Yihua Zhou. It is joyful to work with you all.

Last but not least, I am in debt to my parents for supporting me studying over the last 20 years. I also appreciate my girlfriend's encouraging to the thesis writing. I love you all.

Contents

List of Figures

List of Tables

Abstract

Transport phenomena in biological flow and soft matter is very important in understanding human disease and health. The interaction between cells and blood plasma is important because it not only shows complex mechanical behavior but also advance our knowledge in medical research. This dissertation presents modeling work in drug carrier delivery in blood suspensions and early detection of circulating tumor cells. Methodologically, the Lattice Boltzmann method (chapter 2) was employed as Navier-Stokes fluid solver due to its competence in modeling single phase and multiphase flow, handling complex geometries, and the capacity in parallel computing. A significant part of the work was devoted to the theory, algorithm, boundary conditions, and code implementations. The cells were implemented using a coarse grained molecular dynamics model (chapter 3) because of its capacity in modeling solid nonlinear large deformations. Besides the suspending fluid and cells, nanoparticles (drug carriers) were also introduced into the system (chapter 4). The coupling fluid and solid was based on the Immersed Boundary Method (chapter 5) which removes the burden of expensive mesh updating in traditional Arbitrary Lagrangian Eulerian approach.

The developed code was validated for lid driven cavity flow, cell stretching test, and sphere dropping test in a quiescent fluid. Numerical models were created to study nanoparticle transport in blood cell suspensions(chapter 6). Nanoparticle (NP) dispersion rate is found to be strongly influenced by Red blood cell (RBC) motion, and to have an approximately linear relationship with shear rate in the

RBC tumbling ($\eta < 40s^{-1}$) and RBC tank treading ($\eta > 200s^{-1}$) regions of the flow regime. Between these two regions, there is a transition region where cell gradually transit from a tumbling motion initially into a tank treading motion eventually. From NP dispersion rate under different shear rate, a general formula to estimate NP dispersion rate was developed as $D = k\eta + D_0$ where D_0 is the thermal diffusion coefficient, k is a constant that depends on the hematocrit and cell capillary number. The formula was extended to predict NP dispersion with cell suspensions in channel flows. The formula relates the normalized NP dispersion rate with hematocrit levels, shear rate, thermal diffusion rate and cell size. The predictions given by the proposed empirical formulae agree well with data reported in the literature. Thus, these simple predictive analytical formulae provide an efficient approach for assessing NP dispersion under various flow conditions and hematocrit levels, thereby facilitating practical modeling of NP transport and distribution in large scale vascular systems. That is the novelty of this work compared to other studies in literature. The general formula was also much needed in NP transport and distribution prediction in a large scale vascular network.

Another contribution of the work is the systematic parametric study of the cell translocation through a micropore under different pressure difference and micropore size(chapter 7). The goal of the study is to optimize the microfluidic design so that it can efficiently separate cancer cells from other blood cells. Different cell deformability characterized by membrane compressibility modulus were selected to represent cancer cells and white blood cells. We found that the cell translocation time increases with the cell membrane compressibility modulus, but not very sensitive to the membrane compressibility. However, the cell translocation time grows exponentially as the pressure or micropore diameter decreases. Thus, the pressure difference and the size of the micropore become the key parameters in microfluidic design. Traditionally the Laplace-Young equation was widely used to analyze the cell shape and the pressure difference and tension balance. We found

that the tension of the cell membrane during the cell squeezing process is not uniform, with high stress at the leading membrane or at the membrane contacting the wall. This is contradictory to the uniform tension distribution assumption in the Laplace-Young equation. However, when the bending is relatively small, and a local averaged tension is used, the Laplace-Young equation can still provide a rough prediction for the minimum required pressure.

Chapter 1

Introduction

1.1 Introduction to computational fluid dynamics

Fluid flow has been the interest of human since ancient civilization began in prehistory times. For example, human beings knew how to use fluid mechanics to design boats, arrows, and worship vortices[1, 2]. Their knowledge of fluid motion is more pragmatic if not scientific. Only until the invention of calculus did people begin to study fluid mechanics using partial differential equations (PDE). The most widely used fluid model is the incompressible flow of Newtonian fluids, which are also called incompressible Navier-Stokes (NS) equations. They can be written as

$$\nabla \cdot \boldsymbol{u} = 0 \tag{1.1}$$

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} = -\nabla p + \nu \nabla^2 \boldsymbol{u} + \boldsymbol{f} \tag{1.2}$$

where $\boldsymbol{u}$ is the velocity vector, ν is the kinetic viscosity, p is the pressure, t is the time, $\boldsymbol{f}$ is the body force term. Eqn(1.1) represents the conservation of mass. Eqn(1.2) is essentially Newton's second law of motion. Typically there are four unknowns in NS equations, three components of $\boldsymbol{u}$ and p. They are coupled,

nonlinear and are second order partial differential equations. Thus, in general there is no analytical solutions for naiver stokes equations given sufficient initial and boundary conditions.

Computational fluid dynamics (CFD) is a branch of fluid mechanics that uses numerical methods and algorithms to solve and analyze problems that involve fluid flows. Generally all the numerical methods to simulate fluid flow can be divided into two categories: continuum mechanics based methods and particle based methods. The strategy of continuum mechanics based methods is to use a grid or mesh domain to replace the continuous domain described in the NS equation. In this way the partial differential equations will be replaced by a set of algebraic equations so that computers can be used to find the solution. The simplest and easiest approach is to use finite difference method to approximate the derivatives shown in NS equations. Typically it will be applied to structural grids. A very good practical introduction to develop finite difference code can be found in Ref[3]. However, the most widely used method is based on finite volume approach where the integral form of conservation equations are used. NS equations can be casted into the forms of conversation of mass and momentum. The solution domain is divided into a finite number of cells that are similar to the concept of control volume (CV) in fluid mechanics. At the centroid of each cell the variables such as u_x,u_y,u_z and p will be computed. Interpretation will be used to get the integral form of the conservation equations along the CV surface.

The particle based method is another category of simulating fluid of which the idea is originated from molecular dynamics. In atomic scale, molecular dynamics would be the choice but the length and time are usually limited to small scales. In mesoscale, there are a lot candidates, such as Smooth Particle Hydrodynamics[4], Multiparticle Collision Dynamics(also called Stochastic Rotation Dynamics)[5], Dissipative Particle Dynamics[6], Lattice Boltzmann Method[7, 8]. In this dissertation, I want to highlight the Lattice Boltzmann Method (LBM). LBM is

different from traditional PDE based NS solver. e.g., it does not solve (u_x,u_y,u_z) and p explicitly, as conventional methods do. Instead, it simulates the dynamics of particle distribution function, which is a quantity derived from kinetic theory of gases that represents the probability density of finding gas molecules in the phase space. All the macroscopic variables such as $\boldsymbol{u}$ and p will be derived from the probability density function. LBM also has many advantages over conventional methods. First, it is ideal for parallel computing. Its streaming step in the algorithm only involves the nearest neighbor data, and collision step is purely local. Second, many sets of density distribution functions and their interaction models can be included to model convective heat transfer and multiphase flows. These problems are the challenging ones in the field of computational mechanics. Third, the pressure term is directly obtained from the equation of state that eliminates the burden of solving Poisson equations in PDE based methods. Details of the LBM can be found in Chapter 2.

1.2 Modeling in blood flow and drug delivery

Blood is one of the most important components to transport nutrients, hormones, metabolic wastes, O_2 and CO_2 throughout the body to maintain a live body. Normally 7-8% of human body weight is from blood. Blood is not a Newtonian fluid where it contains cells and plasma. Cells include erythrocytes (red blood cells, 45% by volume), leukocytes(white blood cells), platelets. Plasma is made of water(92% by volume), sugar, fat, protein, and salt solution. Blood flow has been the interest of scientist and researchers for many years. First, understanding the blood flow is important for biomedical research. For example, blood rheology has been reported to be altered in various physiopathological processes, such as sickle cell hemoglobin disease[9], coronary heart disease[10], blood coagulation[11], etc. Second, from the mechanical point of view, the unique mechanical features of red

blood cells (highly deformable, biconcave discoid shape, cytoplasmic viscosity) will result in very complex phenomena in flow. It is well known that RBCs will undergo tumbling motion or tank treading motion in shear flow, depending on the external shear stress and cell membrane and cytoplasm properties[12, 13]. The biconcave geometry enables cell shape change without increasing the surface area. Thus, studying blood flow can advance our knowledge in both medical science as well as computational mechanics.

Recently, the application of nanotechnology into medicine brought a new research area named nanomedicine. Various nanoparticles (NP) based platforms have been used as drug carriers to achieve targeted drug delivery. One of the reasons is that carriers in the nanoscale offer advantages of enhanced delivery efficiency, targeting, controlled release, and ability to bypass the biological barriers. NPs can be engineered into different sizes, shapes and surface chemistry to meet these requirements. As a key characteristic of NPs, size has been studied extensively and reported in literature. For example, it is known that spherical particles bigger than 200 nm are efficiently filtered by liver, spleen and bone marrow, while particles smaller than 10 nm can be quickly cleared by kidney or through extravasation, thus making 10-200 nm the ideal size range for the circulating spherical carriers. Similar to size, shape is also a fundamental property of NPs that is critical for their intended biological functions. Most NPs have a spherical shape. With the advanced nanofabrication techniques, different shapes and forms of NP have emerged in recent years with unique geometrical, physical and chemical properties. For example, nanorods with suitable aspect ratio have been fabricated as a novel contrast agent for both molecular imaging and photothermal cancer therapy[14]; asymmetrically functionalized Au-NPs have been assembled to build nanochains[15]; superparamagnetic iron oxide based nanoworms are studied for tumor targeting[16], and nanonecklaces are assembled using gold NPs by covalent bonding[17]. It has been reported that cylindrically shaped filamentous micelles

can effectively evade non-specific uptake by the Reticulo-Endothelial system, allowing persistent circulation for up to one week after intravenous injection[18].

Although NPs offer many advantages over conventional drug carriers, their transport, binding and distribution are not fully understood. First, NPs are too small to be visualized in vivo with dimensions between ten to a few hundred nanometers; it is challenging to track NP trajectories in vivo. Second, biological environment in vivo for NP is very complex, which involves blood flow, circulation, elimination and extravasation. NPs cannot always reach targeted sites because they may be filtered by liver, eliminated by spleen, bonded to healthy cells. Generally, after administration, the NPs will undergo margination from the core of the blood stream, then the ligands coated on the NP surface will interact with receptors expressed over endothelial cell surfaces, which will lead to firm adhesion, and finally the NPs will be internalized by the cells. A schematic illustration of the process is shown in Fig.1.1.

To achieve better delivery efficacy, the diseased region type, size, location, and the patient's physical parameters (e.g. vascular diameter, blood flow rate, surface area, blood components) need to be considered. The NP targeted delivery in vascular system involves interplay of transport, hydrodynamic force, and multivalent interactions with targeted biosurfaces. It is very challenging to explore these phenomena experimentally in vivo, due to the small size of NPs, the dynamic delivery process, and the complex vascular environment. Thus a mathematical tool that can predict and evaluate the NP binding and distribution is necessary. Capillaries are the main places where NPs binding happen. The size of capillaries is of the same order of red blood cell diameter. To model drug delivery in capillaries the particulate nature of the blood, RBCs, need to be considered.

In large scale vascular network, it is almost impossible to model all the blood components explicitly even with the latest computational technologies. It crosses several order of magnitude in scales, e.g., the size of the vascular network in human

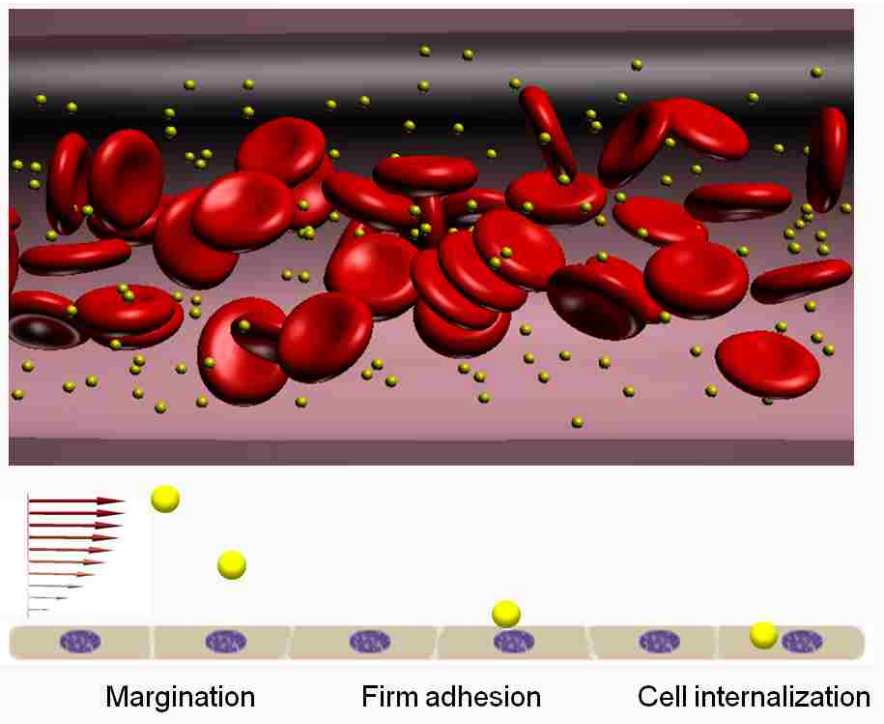
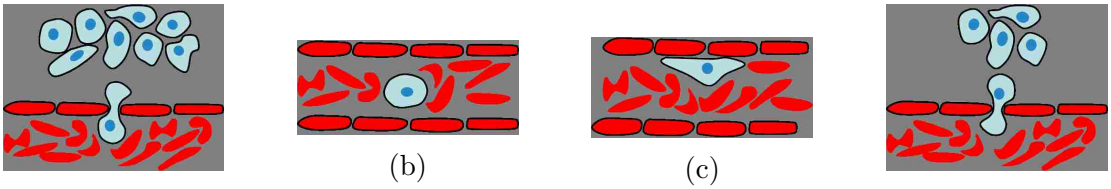


Figure 1.1: A schematic illustration of nanoparticle (NP) delivery process in blood stream. First, the NPs will migrate toward the wall (Margination), then the ligands of the NPs will interact with the receptors over the endothelial cell surfaces and form bonding (Firm adhesion), and finally the NP will be internalized by the cells (Cell internalization). The particles and cells are not drawn in scale.

lung is in order of centimeters, while the size of RBCs is in micrometers[19, p.333]. It also involves an enormous amount of cells, e.g., a billion of red blood cells[20] in 1 mL blood sample. Thus, the NP delivery prediction in large scale vascular network are typically modeled using a convection-diffusion-reaction based partial differential equation (PDE) models[21–23]. This is a reasonable assumption since NPs is relatively small compared to large vessel diameters in vascular network. One of the key parameters in the convection-diffusion-reaction models is the diffusion rate. Research have shown that RBCs influence particle dispersion rate[24–26]. However, there is no general quantitative law to predict the dispersion rate considering the influence of blood cells and flow rate. We studied the effect of different shear rates and hematocrit levels on NP dispersion rate systematically, and proposed a formula to predict the NP dispersion rate so that it can be directly used in convection-diffusion-reaction equations based models. The detailed studying methods and results will be introduced in Chapter 6.

1.3 Circulating tumor cell detection

In USA, cancer has surpassed heart disease as leading cause of death for people younger than age 85[27]. Metastasis is the spread of a cancer cells from the primary tumor to the other parts of the body to form a secondary tumor. Metastasis accounts for about 90% of the death of cancer patients[28]. It is a complex process with multiple steps. In the beginning, tumor cells detach from a primary and vascularized tumor. As cells transform into cancerous state, they become very soft because the organized cytoskeleton network is changed into an irregular one, thus, they can easily penetrate the surrounding tissue, enter nearby blood vessels, travel through small capillaries, and circulate in the vascular system(intravasation). So they are also named circulating tumor cells(CTC). Some of these cells eventually adhere to blood vessel walls and are able to extravasate and migrate into



(a) (b) (c) (d)

Figure 1.2: An illustration of metastasis process. (a) A tumor cell is detaching from the primary tumor site and penetrate into blood stream. (b) the tumor cell is in the circulation system. (c) the circulating tumor cell adhesion. (d) the extravasation and the growth of the secondary tumor.

the local tissue(extravasation), where they can form a secondary tumor. During the intravasation and extravasation process, the cancer cells must undergo large deformations. Thus, generally it is agreed that cancer cells are more deformable than normal healthy cells. A schematic illustration of the tumor cell detaching from the primary site and intravasation process are shown in Fig.1.2.

In many cases, the person's chance of full recovery is much higher if the cancer is diagnosed and treated earlier. Thus, it is very important to detect CTCs in the early stage. However, detecting CTCs is very challenging, because CTC is a rare event with about a few CTCs in millions of white blood cells and a billion of red blood cells[20] in 1 mL blood sample. Many different physical mechanism have been used to enrich CTCs, including size[29, 30], magnetic field[31, 32], electric field[33], optical force[34], acoustic fields[35]. Meanwhile, the specific binding between receptors expressed on cancer cell membrane and ligands coated on microfluidic chips have been explored to detect CTCs[36]. Among this methods, they either require sophisticated cell preparation, careful microfluidics design, or external fields to enumerate CTCs. Alternatively, a low cost microfluidic chip based on cell deformability has been used to isolate CTCs[37, 38].

Cell deformability is an important biomarker to differentiate diseased cells from healthy cells[39]. Deformability is indicative of cell physical properties associated with cell functional changes in underlying cell membrane, cytoskeletal network, or nuclear changes[40–42]. For example, healthy red blood cells(RBC) are highly

deformable that can be squeezed through a channel size as small as 4 um under flow[43]. However, the diseased RBCs, i.e., malaria parasite infected ones, are up to 10-folder times stiffer than healthy ones[44, 45]. Another example are cancer cells. The highly deformability enables cancer cells escape the primary tumor and traverse extracellular matrix and invade circulation system[41, 42]. Not only the cell deformability will change the cell metastasis process, the reduced friction on the cell surface may also contribute to the migration process[46]. In both examples, the cell stiffness changes are related to the pathophysiological state of individual cells. Thus, the cell deformability can be used as a biomarker to detect diseased cells. Microfluidics with proper size of micropores or gaps have been used to differentiate cancer cells from other cells based on the cell deformability. However, it not clear what micropore size or proper pressure should be used to differentiate the cells efficiently. In the dissertation, we studied the cell translocation process through a narrow pore numerically. The effect of cell deformability, the pressure difference, and the pore size on cell translocation time were studied using the developed combined lattice Boltzmann method and coarse grained cell membrane model. It is also demonstrated the capabilities of the developed model to optimize the microfluidics design so that the cancer cells can be separated from other blood cells efficiently.

Chapter 2

Lattice Boltzmann method

Lattice Boltzmann Method (LBM) was originated from Lattice Gas Automata[47, 48] in the 90s last century. Then, it was recognized that it can be used as an alternative to compute fluid dynamics. Many papers[7, 49, 50], reviews[51, 52] and monographs[8, 53] have been published on its CFD capabilities. An efficient algorithm for stokes flow and treatment for complex solid boundaries in a porous media were presented in paper[54, 55]. Another advantage of LBM is its capabilities in modeling multiphase flow[56] and convective heat transfer[57]. The simple algorithm involving local streaming and collision steps makes it relatively easy to take advantage of parallel computing. Thus, we choose it as our fluid solver in the dissertation.

2.1 Statistical Mechanics

Simulating fluid behavior can be done in different scales. In engineering applications, typically fluids are assumed to be a continuum where Naiver-Stokes equations are usually employed to predict the fluid behavior. Meanwhile, it is well known that fluids consist of a tremendous amount ($\sim 10^{23}$) of individual atoms in atomic scale. There is no surprise that molecular dynamics (MD) can also

be used to simulate fluid behavior from a bottom up approach. However, it is too expensive if the MD approach is used to model a large system in micro-size and running in time of a few seconds even with the current computing resources. LBM is the method bridge the gap between MD and conventional CFD. LBM is a simplified Boltzmann equation which describes the statistical behavior of a thermodynamic system not in thermodynamic equilibrium. The description of motion of a thermodynamic system in molecule level typically is given in phase space. Consider a system with N number of molecules, the Newton's equations can be written as:

$$\begin{aligned} \frac{d\boldsymbol{x}_i}{dt} &= \frac{\boldsymbol{p}_i}{m} \\ \frac{d\boldsymbol{p}_i}{dt} &= \boldsymbol{F}_i \end{aligned} \tag{2.1}$$

where $i = 1, ..., N$, $\boldsymbol{x}_i$ is the spacial coordinate of the i^{th} molecule, $\boldsymbol{p}_i := m\boldsymbol{\xi}_i$ is the linear momentum,m is the mass of a molecule, $\boldsymbol{\xi}_i$ is the velocity vector, and $\boldsymbol{F}_i$ is the external force(e.g., intermolecular interaction potential or electric force) acting on the molecule. In three dimensional space, there are total of $6N$ functions of time $(\boldsymbol{x}_i(t), \boldsymbol{p}_i(t))$. Eqn (2.1) provide every detailed information about each molecule in the system over phase space. However, practically it is impossible to track all the molecules since N is such an enormous number. However, such a big number can enable us to treat the thermodynamic system as a continuum in phase space. A density function can be introduced to describe their distributions. We can also normalize it so that it is a probability density distribution in phase space. The normalized probability density distribution function will be denoted as $f_N(\boldsymbol{x}_1, \boldsymbol{p}_1, \cdots, \boldsymbol{x}_N, \boldsymbol{p}_N, t)$. Thus, $f_N(\boldsymbol{x}_1, \boldsymbol{p}_1, \cdots, \boldsymbol{x}_N, \boldsymbol{p}_N, t)d\boldsymbol{x}_1 d\boldsymbol{p}_1 \cdots d\boldsymbol{x}_N d\boldsymbol{p}_N$ is the probability to find a particle within the interval $[\boldsymbol{x}_1, \boldsymbol{x}_1 + d\boldsymbol{x}_1] \times [\boldsymbol{p}_1, \boldsymbol{p}_1 + d\boldsymbol{p}_1] \times \cdots \times [\boldsymbol{x}_N, \boldsymbol{x}_N + d\boldsymbol{x}_N] \times [\boldsymbol{p}_N, \boldsymbol{p}_N + d\boldsymbol{p}_N]$. The evolution of f_N follows the

Liouville theorem:

$$\frac{df_N}{dt} = \frac{\partial f_N}{\partial t} + \sum_{j=1}^{3N} \left(\frac{\partial f_N}{\partial x_j}\frac{\partial H_N}{\partial p_j} - \frac{\partial f_N}{\partial p_j}\frac{\partial H_N}{\partial x_j} \right) = 0 \tag{2.2}$$

where $H_N$is the Hamiltonian of the system. Integrating f_N over part of the phase space, we define the R particle reduced distribution functions as

$$F_R(\boldsymbol{x}_1, \boldsymbol{p}_1, \cdots, \boldsymbol{x}_R, \boldsymbol{p}_R, t) = \int f_N(\boldsymbol{x}_1, \boldsymbol{p}_1, \cdots, \boldsymbol{x}_N, \boldsymbol{p}_N, t) d\boldsymbol{x}_{R+1} d\boldsymbol{p}_{R+1} \cdots d\boldsymbol{x}_N d\boldsymbol{p}_N \tag{2.3}$$

It is clear that $F_1(\boldsymbol{x}_1, \boldsymbol{p}_1, t) d\boldsymbol{x}_1 d\boldsymbol{p}_1$ is the probability of finding molecule 1 in the incremental volume element $d\boldsymbol{x}_1 d\boldsymbol{p}_1$ about the phase point $(\boldsymbol{x}_1, \boldsymbol{p}_1)$ at time t. The coupled equations for temporal evolution of $F_R (1 \leq R \leq N)$ from integrating Eqn. (2.2) over part of the phase space is called BBGKY hierarchy of equations[58, p18]. Boltzmann obtained the closed equation for F_1 from the BBGKY equations by a few assumptions of binary collisions and uncorrelated velocities before collision and free of external forces. Rewriting $F_1(\boldsymbol{x}_1, \boldsymbol{p}_1, t)$ as $f(\boldsymbol{x}, \boldsymbol{\xi}, t)$, the Boltzmann equation becomes

$$(\partial_t + \xi \cdot \nabla_{\boldsymbol{x}} + \boldsymbol{g} \cdot \nabla_\xi) f(\boldsymbol{x}, \boldsymbol{\xi}, t) = \Omega(f, f) \tag{2.4}$$

where $\boldsymbol{g}(\boldsymbol{x}, t)$ is the acceleration term, $\Omega(f, f)$ is the differential collision cross section for the binary collision transforming from the incoming velocities to outgoing velocities. The conservation of mass, momentum, and energy would require

$$\int \Omega(f, f) \phi_k(\boldsymbol{\xi}) d\boldsymbol{\xi} = 0 \tag{2.5}$$

where $\phi_k(\boldsymbol{\xi})$ is a function of $\boldsymbol{\xi}, k = 0, 1, 2, 3, 4$, as shown

$$\phi_k(\boldsymbol{\xi}) = \begin{cases} 1, & k = 0 \quad \text{mass} \\ \boldsymbol{e}_k \cdot \boldsymbol{\xi}, & k = 1, 2, 3 \quad \text{momentum} \\ \boldsymbol{\xi}^2, & k = 4 \quad \text{energy} \end{cases} \tag{2.6}$$

where $\boldsymbol{e}_k$ is the basis vector of a Catesian coordinate system. The dynamics of Boltzmann equation (2.4) brings the velocity distribution function closer to the Maxwell-Boltzmann equilibrium distribution function

$$f^{(0)}(\boldsymbol{x}, \boldsymbol{\xi}, t) = \frac{\rho(\boldsymbol{x}, t)}{2\pi\theta(\boldsymbol{x}, t)^{D/2}} exp\left(-\frac{(\boldsymbol{\xi} - \boldsymbol{u})^2}{2\theta(\boldsymbol{x}, t)}\right) \tag{2.7}$$

where $\rho(\boldsymbol{x}, t)$ and $\boldsymbol{u}(\boldsymbol{x}, t)$ are the density and velocity of the fluid at spatial position $\boldsymbol{x}$ and time t; $\theta = k_B T/m$ with k_B the Boltzmann constant, m the mass of a single molecule particle, T is the temperature in the Kelvin units, D is the demensions. Details of the derivation of the Boltzmann equation can be found in Ref[58].

2.2 From Boltzmann equation to Navier-Stokes equations

In this section, the derivation from Boltzmann equation to Navier-Stokes equtions will be presented. There are two most widely used approaches to derive them. The first one is multiscale Chapman-Enskog expansion analysis[51, 59, 60]. The basic idea is to separate time into two different scales such as diffusion and convection, and then use perturbation method to analyze the zero, first, second order of moments of probability distribution function f. Details can be found in the above references. However, this approach is very tedious and requires some knowledge in multiscale analysis. The second approach is through integration of the Boltzmann

equation over velocity space with respect to the zero, first, and second order moments of f. This approach is straight forward compared to the multiscale analysis approach. Thus, it is used here. The following derivation is largely inspired from Ref [61].

For simplicity, the Boltzmann-BGK equation will be used. The Boltzmann-BGK equation is defined as

$$(\partial_t + \boldsymbol{\xi} \cdot \nabla_{\boldsymbol{x}} + \boldsymbol{g} \cdot \nabla_{\xi}) f(\boldsymbol{x}, \boldsymbol{\xi}, t) = \Omega(f, f) \tag{2.8}$$

The sound speed is defined as $c_s = \sqrt{k_B T/(mD)}$ Noted that macroscopic properties can be expressed in terms of the integration of the moments of the density distribution function over phase space. Specifically, we have

$$\text{mass density:} \quad \rho(\boldsymbol{x}, t) = \int f(\boldsymbol{x}, \boldsymbol{\xi}, t) d\boldsymbol{\xi} \tag{2.9}$$

$$\text{momentum density:} \quad \rho \boldsymbol{u}(\boldsymbol{x}, t) = \int \boldsymbol{\xi} f(\boldsymbol{x}, \boldsymbol{\xi}, t) d\boldsymbol{\xi} \tag{2.10}$$

$$\text{kinetic density:} \quad \rho\epsilon(\boldsymbol{x}, t) = \frac{D}{2}\rho\theta = \frac{1}{2}\int \boldsymbol{c}^2 f(\boldsymbol{x}, \boldsymbol{\xi}, t) d\boldsymbol{\xi} \tag{2.11}$$

where $\boldsymbol{c} = \boldsymbol{\xi} - \boldsymbol{u}$ is the velocity deviation from the mean velocity. Next, we are going to multiply Eqn (2.8) with $1, \boldsymbol{\xi}$ and integrate it over velocity space $\boldsymbol{\xi}$ in order to get Navier-Stokes equation. For the ease of discussion, let us define an integration operator $I[\cdot]$ as

$$I[h](\boldsymbol{x}, t) = \int h(\boldsymbol{\xi}) f(\boldsymbol{x}, \boldsymbol{\xi}, t) d\boldsymbol{\xi} \tag{2.12}$$

where $h(\boldsymbol{\xi})$ is an integrable function dependent only on the velocity $\boldsymbol{\xi}$, In particualr, we are interested on $1, \boldsymbol{\xi}, \boldsymbol{\xi}^2$, corresponding to mass, momentum, and energy. Multiplying $h(\boldsymbol{\xi})$ with the Boltzmann equation (2.8) and integrating over velocity

space, we obtain

$$\int (\partial_t f + \boldsymbol{\xi} \cdot \nabla_{\boldsymbol{x}} f + \boldsymbol{g} \cdot \nabla_{\boldsymbol{\xi}} f) h d\boldsymbol{\xi} = \int \Omega(f, f) h d\boldsymbol{\xi} \tag{2.13}$$

the right hand of Eqn (2.13) vanishes due to the conservation law of the collision operator(see Eqn 2.5). For each term on the left hand side, we have,

$$\int (\partial_t f) h(\boldsymbol{\xi}) d\boldsymbol{\xi} = \partial_t (I[h(\boldsymbol{\xi})]) \tag{2.14}$$

$$\begin{aligned} \int (\xi \cdot \nabla_{\boldsymbol{x}} f) h(\boldsymbol{\xi}) d\boldsymbol{\xi} &= \nabla_{\boldsymbol{x}} \cdot \left(\int f h(\boldsymbol{\xi}) \boldsymbol{\xi} d\boldsymbol{\xi} \right) - \int f \underbrace{\nabla_{\boldsymbol{x}} \cdot (\boldsymbol{\xi} h(\boldsymbol{\xi}))}_{=0} d\boldsymbol{\xi} \\ &= \nabla_{\boldsymbol{x}} \cdot (I[h(\boldsymbol{\xi}) \boldsymbol{\xi}]) \end{aligned} \tag{2.15}$$

$$\begin{aligned} \int (g(\boldsymbol{x}, t) \cdot \nabla_{\boldsymbol{\xi}} f) h(\boldsymbol{\xi}) d\boldsymbol{\xi} &= \int (g(\boldsymbol{x}, t) h(\boldsymbol{\xi}) \cdot \nabla_{\boldsymbol{\xi}} f) d\boldsymbol{\xi} \quad \text{(using divergence theorem)} \\ &= \int (g(\boldsymbol{x}, t) h(\boldsymbol{\xi}) \underbrace{f \boldsymbol{n}) d\eta}_{=0} - \int \nabla_{\boldsymbol{\xi}} \cdot (g(\boldsymbol{x}, t) h(\boldsymbol{\xi})) f d\boldsymbol{\xi} \\ &= - \int g(\boldsymbol{x}, t) \cdot \nabla_{\boldsymbol{\xi}} h(\boldsymbol{\xi}) f d\boldsymbol{\xi} \\ &= -I[g(\boldsymbol{x}, t) \cdot \nabla_{\boldsymbol{\xi}} h(\boldsymbol{\xi})] \end{aligned} \tag{2.16}$$

Note that the vanishing term in Eqn (2.15) is because the term $\boldsymbol{\xi} h(\boldsymbol{\xi})$ does not depend on $\boldsymbol{x}$, the vanishing term in Eqn (2.16) is because the probability distribution function $f(\boldsymbol{x}, \boldsymbol{\xi}, t)$ at the boundary of velocity space is approaching zero. Substituting the above three equations into the Eqn(2.13) to obtain

$$\partial_t (I[h(\boldsymbol{\xi})]) + \nabla_{\boldsymbol{x}} \cdot (I[h(\boldsymbol{\xi}) \boldsymbol{\xi}]) - I[g(\boldsymbol{x}, t) \cdot \nabla_{\boldsymbol{\xi}} h(\boldsymbol{\xi})] = 0 \tag{2.17}$$

Evaluate Eqn (2.12) when $h(\boldsymbol{\xi}) = \{1, \boldsymbol{\xi}, \boldsymbol{\xi}^2/2\}$, we get

$$\begin{cases} I[1] = \rho, I[\boldsymbol{\xi}] = \rho\boldsymbol{u}, I[\boldsymbol{c}] = I[\boldsymbol{\xi}] - I[\boldsymbol{u}] = 0 \\ I[\boldsymbol{\xi}^2] = I[(\boldsymbol{c}+\boldsymbol{u})^2] = I[\boldsymbol{c}^2] + 2I[\boldsymbol{c}]\cdot\boldsymbol{u} + \rho\boldsymbol{u}^2 = 2\rho\epsilon + \rho\boldsymbol{u}^2 \\ I[\boldsymbol{\xi}\boldsymbol{\xi}] = I[(\boldsymbol{c}+\boldsymbol{u})(\boldsymbol{c}+\boldsymbol{u})] = I[\boldsymbol{c}\boldsymbol{c}] + \rho\boldsymbol{u}\boldsymbol{u} = \boldsymbol{P} + \rho\boldsymbol{u}\boldsymbol{u} \\ I[\boldsymbol{\xi}^2\boldsymbol{\xi}] = I[(\boldsymbol{c}+\boldsymbol{u})^2(\boldsymbol{c}+\boldsymbol{u}] = I[\boldsymbol{c}^2\boldsymbol{c}] + 2I[\boldsymbol{c}\boldsymbol{c}]\cdot\boldsymbol{u} + I[\boldsymbol{\xi}^2]\boldsymbol{u} \\ \qquad = 2\boldsymbol{q} + 2\boldsymbol{P}\cdot\boldsymbol{u} + (2\rho\epsilon + \rho\boldsymbol{u}^2)\boldsymbol{u} \end{cases} \tag{2.18}$$

where the pressure tensor is defined as $\boldsymbol{P} := I[\boldsymbol{c}\boldsymbol{c}]$, energy or heat flux is defined as $\boldsymbol{q} = \frac{1}{2}I[\boldsymbol{c}^2\boldsymbol{c}]$. Notice that $\boldsymbol{c}\boldsymbol{c}$ is a tensor product.

Taking Eqn (2.9) and (2.10) into account, and substitute identities (2.18) into Eqn (2.17) we get the celebrated Navier-Stokes equations

$$\partial_t\rho + \nabla\cdot(\rho\boldsymbol{u}) = 0 \tag{2.19}$$

$$\partial_t(\rho\boldsymbol{u}) + \nabla\cdot(\rho\boldsymbol{u}\boldsymbol{u}) + \nabla\cdot\boldsymbol{P} = \rho\boldsymbol{g} \tag{2.20}$$

where $\boldsymbol{P} := I[\boldsymbol{c}\boldsymbol{c}]$. With the continuity equation (2.19), momentum equation (2.20) can be simplified as

$$\rho\partial_t(\boldsymbol{u}) + \rho(\boldsymbol{u}\cdot\nabla)\boldsymbol{u} + \nabla\cdot\boldsymbol{P} = \rho\boldsymbol{g} \tag{2.21}$$

Let us define stress tensors

$$\boldsymbol{Q} := \boldsymbol{c}\boldsymbol{c} - c_s^2\boldsymbol{I} \tag{2.22}$$

$$\boldsymbol{\Pi} := I[\boldsymbol{Q}] = \int(\boldsymbol{c}\boldsymbol{c} - c_s^2\boldsymbol{I})f(\boldsymbol{x},\boldsymbol{\xi},t)d\boldsymbol{\xi} \tag{2.23}$$

and reformulate Eqn (2.21)

$$\rho\partial_t(\boldsymbol{u}) + \rho(\boldsymbol{u}\cdot\nabla)\boldsymbol{u} + \nabla\cdot\boldsymbol{\Pi} + \nabla\cdot(p\boldsymbol{I}) = \rho\boldsymbol{g} \tag{2.24}$$

where p is the hydrostatic pressure defined as

$$p = c_s^2 \rho \tag{2.25}$$

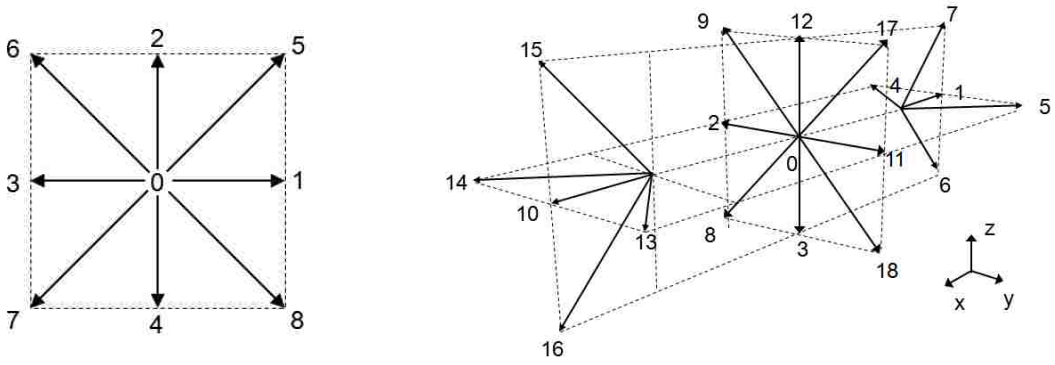
2.3 Lattice BGK Model

As illustrated in section (2.2), the simplest collision operator is the lattice BGK model[7] characterized by a single relaxation time τ

$$\Omega = -\frac{1}{\tau}(f - f^{(0)}) \tag{2.26}$$

where τ is the relaxation time and $f^{(0)}$ is the equilibrium density distribution function defined at Eqn (2.7). Once the collision operator is selected, the next step is to discretize the velocity space into a few representative velocities. A particular discretized velocity space including weights, direction, magnitude, etc. will be named as a lattice structure. A particular lattice structure will be denoted as DdQq lattice. For example, D2Q9 represents a 2D lattice structure with 9 discretized velocity vectors. Depending on the velocity discretization, there are several different lattice structures, such as D2Q6, D2Q7, D2Q9, D3Q13, D3Q15, D3Q19, D3Q27, etc. The criteria to determine the efficient lattice structure is to achieve the highest order of accuracy with minimum number of discrete velocities. Meanwhile, the efficient lattice structure should also consider the convenience of streaming step between different lattices. Gauss quadrature is usually employed to achieve the better order of accuracy in integration, which will introduce different weights for each discretized velocity. The detailed derivation process can be found in Ref[61, 62]. The most widely used lattice structures are D2Q9 for 2D and D3Q19 for 3D, which will be discussed next.

D2Q9 and D3Q19 are shown in Fig 2.1. Note that part of the velocities of D3Q19 has been shifted from the origin for the purpose of easy visualization. The



(a) D2Q9 lattice

(b) D3Q19 lattice

Figure 2.1: Illustrations of D2Q9 and D3Q19 lattices

$c_s^2 = 1/3$			
$w_0 = 4/9$	$w_s = 1/9$	$w_l = 1/36$	
$\boldsymbol{\xi}_0 = (0,0)$			
$\boldsymbol{\xi}_1 = (1,0)$	$\boldsymbol{\xi}_2 = (0,1)$	$\boldsymbol{\xi}_3 = (-1,0)$	$\boldsymbol{\xi}_4 = (0,-1)$
$\boldsymbol{\xi}_5 = (1,1)$	$\boldsymbol{\xi}_6 = (-1,1)$	$\boldsymbol{\xi}_7 = (-1,-1)$	$\boldsymbol{\xi}_8 = (1,-1)$

Table 2.1: D2Q9 lattice structure parameters

order of the velocity vectors depends programmer's preference. For example, the order of velocity vectors in the D3Q19 lattice(see Fig 2.1b) is shown to be better in double swap algorithm for the streaming step[63]. The lattice speed is assumed to be unity for the easy of discussion. The corresponding speed of sound (c_s), weight (w_i), velocity vectors (c_i) for D2Q9 and D3Q19 are shown in Table 2.1 and Table 2.2, respectively. The subscript s and l represent short velocity vectors (e.g., $(-1,0)$ in 2D, $(0,1,0)$ in 3D) and long velocity vectors (e.g., $(1,1)$ in 2D, $(-1,0,1)$ in 3D).

The next step is the discretization of BGK Eqn (2.8) over the space and time. Let δx and δt represent the physical distance between two adjacent lattice nodes and the time step. The discretized BGK scheme with force term is described by

$$\underbrace{f_\alpha(\boldsymbol{x}_i + \boldsymbol{\xi}_\alpha \delta t, t+\delta t) - f_\alpha(\boldsymbol{x}_i, t)}_{\text{streaming}} = \underbrace{-\frac{\delta t}{\tau}[f_\alpha(\boldsymbol{x}_i, t) - f_\alpha^{(0)}(\boldsymbol{x}_i, t)]}_{\text{collision}} + F_\alpha \tag{2.27}$$

where α is the index of discretized velocities, $\boldsymbol{\xi}_\alpha$ is the velocity vector, i is the

$c_s^2 = 1/3$		
$w_0 = 1/3 \quad w_s = 1/18 \quad w_l = 1/36$		
$\boldsymbol{\xi}_0 = (0,0,0)$		
$\boldsymbol{\xi}_1 = (-1,0,0)$	$\boldsymbol{\xi}_2 = (0,-1,0)$	$\boldsymbol{\xi}_3 = (0,0,-1)$
$\boldsymbol{\xi}_4 = (-1,-1,0)$	$\boldsymbol{\xi}_5 = (-1,1,0)$	$\boldsymbol{\xi}_6 = (-1,0,-1)$
$\boldsymbol{\xi}_7 = (-1,0,1)$	$\boldsymbol{\xi}_8 = (0,-1,-1)$	$\boldsymbol{\xi}_9 = (0,-1,1)$
$\boldsymbol{\xi}_{10} = (1,0,0)$	$\boldsymbol{\xi}_{11} = (0,1,0)$	$\boldsymbol{\xi}_{12} = (0,0,1)$
$\boldsymbol{\xi}_{13} = (1,1,0)$	$\boldsymbol{\xi}_{14} = (1,-1,0)$	$\boldsymbol{\xi}_{15} = (1,0,1)$
$\boldsymbol{\xi}_{16} = (1,0,-1)$	$\boldsymbol{\xi}_{17} = (0,1,1)$	$\boldsymbol{\xi}_{18} = (0,1,-1)$

Table 2.2: D3Q19 lattice structure parameters

index of spacial lattice site, F_α is the force term. $f_\alpha^{(0)}(\boldsymbol{x}_i, t)$ is the population distribution at equilibrium. Note that it is not convenient to express Lattice Boltzmann Equation (LBE) in physical units. We introduce lattice units so that $\delta x = 1$ and $\delta t = 1$. The conversion between physical units and lattice units will be discussed in next section. The equilibrium distribution $f_\alpha^{(0)}$ is related to the local macroscale fluid velocity $\boldsymbol{u}$ and the speed of sound c_s as

$$f_\alpha^{(0)}(\boldsymbol{x}_i, t) = w_\alpha \rho \left(1 + \frac{\boldsymbol{\xi}_\alpha \cdot \boldsymbol{u}}{c_s^2} + \frac{(\boldsymbol{\xi}_\alpha \cdot \boldsymbol{u})^2}{2c_s^4} - \frac{\boldsymbol{u^2}}{2c_s^2}\right) \tag{2.28}$$

The fluid viscosity υ is related to the single relaxation τ

$$\upsilon = c_s^2(\tau - \frac{1}{2}) = \frac{\tau - 1/2}{3} \tag{2.29}$$

The force term F_α is introduced to model external force field, such as gravity or force exerted by structure deformation. It can be expressed in terms of external body force density $\rho\boldsymbol{g}$ and fluid macroscale velocity $\boldsymbol{u}$

$$F_\alpha = \left(1 - \frac{1}{\tau}\right) w_\alpha \left(\frac{\boldsymbol{\xi}_\alpha - \boldsymbol{u}}{c_s^2} + \frac{\boldsymbol{\xi}_\alpha \cdot \boldsymbol{u}}{c_s^4}\boldsymbol{\xi}_\alpha\right) \cdot \rho\boldsymbol{g} \tag{2.30}$$

The discretized version of density and velocity would become

$$\rho(\boldsymbol{x}, t) = \sum_{\alpha} f_{\alpha}(\boldsymbol{x}, t) \tag{2.31}$$

$$\rho \boldsymbol{u}(\boldsymbol{x}, t) = \sum_{\alpha} \boldsymbol{\xi}_{\alpha} f_{\alpha}(\boldsymbol{x}, t) \tag{2.32}$$

When the Reynolds number Re is very small, such as the flow in microfluidics, the equilibrium distribution 2.28 can be simplified by dropping the nonlinear term

$$f_{\alpha}^{(0)}(\boldsymbol{x}_i, t) = w_{\alpha} \rho \left(1 + \frac{\boldsymbol{\xi}_{\alpha} \cdot \boldsymbol{u}}{c_s^2}\right) \tag{2.33}$$

This method to simulate stokes flow was first proposed by Ladd [54] and has been used by others as well[64]. Without the nonlinear terms, the computing speed can be improved by 25% in our simulations. The general flow chart of the algorithm for LBM is shown Fig. 2.2

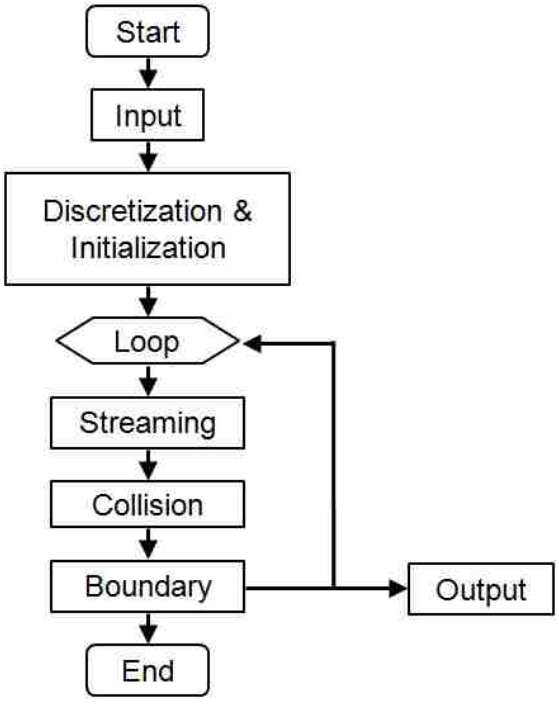


Figure 2.2: An example of the flow chart for Lattice Boltzmann algorithm.

2.4 Regularized BGK model

The BGK scheme is very simple to code. However, it suffers from instability at high Reynolds numbers. A regularize-equilibrium distribution functions was proposed by Latt and Chopard[65, 66] to reduce the instability. From Eqn2.12 we can get the momentum flux tensor Φ by substituting $h(\boldsymbol{\xi}) = \boldsymbol{\xi}\boldsymbol{\xi}$,

$$\Phi = \int f\boldsymbol{\xi}\boldsymbol{\xi}d\boldsymbol{\xi} \tag{2.34}$$

Rewrite Eqn2.34 in discretized component notation,

$$\Phi_{ij} = \sum_{\alpha} f_{\alpha}\xi_{\alpha i}\xi_{\alpha j} \tag{2.35}$$

From multiscale Chapman-Enskog expansion[7, 59], the distribution function f can be expanded as $f = \sum_k f^k$. The zero order term is the equilibrium distribution, $f^{eq} = f^{(0)}$, and the nonequilibrium terms can be defined as

$$f_{\alpha}^{neq} = f_{\alpha} - f_{\alpha}^{0}, \quad \text{and} \quad \Phi^{neq} = \Phi - \Phi^{eq} \tag{2.36}$$

In BGK model, the nonequilibrium distribution is approximated by the first order term of the distribution function

$$f_{\alpha}^{neq} \approx f_{\alpha}^{(1)} = -\frac{\delta t}{\omega c_s^2} w_{\alpha} Q_{\alpha ij} \partial_i \rho u_j \tag{2.37}$$

where $Q_{\alpha ij}$ is defined in Eqn(2.22).

$$\Phi_{ij}^{neq} \approx \sum_{\alpha} f_{\alpha}^{(1)} \xi_{\alpha i}\xi_{\alpha j} = -\frac{\delta t c_s^2}{\omega}(\partial_i \rho u_j + \partial_j \rho u_i) \tag{2.38}$$

where $\omega = 1/\tau$. Because the use of the first order term to approximate the nonequilibrium term, the numerical results deviate from its correct hydrodynamic

behavior that leads to inaccuracy and instabilities[65, 66]. To reduce the discrepancy between f^{neq} and $f^{(1)}$, Ref[65] suggest to reinforce $f^{neq} = f^{(1)}$ and calculate the regularized $f^{(1)}$ as

$$f_\alpha^{(1)} = \frac{w_\alpha}{2c_s^4} Q_{\alpha ij} \Phi_{ij}^{neq} \tag{2.39}$$

The standard BGK collision operator can be used to find the distribution function after collision

$$f_\alpha^* = f_\alpha^{eq} + (1-\omega) f_\alpha^{(1)} \tag{2.40}$$

where f_α^* denotes the distribution after collision.

The algorithm of the regularized BGK method is summarized as follows:

1. find the nonequilibrium part of the distribution function f^{neq} from Eqn. 2.36;
2. find the nonequilibrium part of the stress term Φ_{ij}^{neq} from Eqn. 2.38 by replacing $f^{(1)}$ with f^{neq};
3. evaluate the regularized $f^{(1)}$ using Eqn. 2.39. The $Q_{\alpha ij}$ can be found using Eqn. 2.22;
4. evaluate the distribution function after collision using Eqn. 2.40.

During our test, the regularized method can provide more stable solutions for vortex shedding flows up to $Re = 1000$ with the characteristic length defined as the obstacle diameter. The computational time for regularized method is about 15% slower than BGK method. The details of the comparison between BGK and Regularized BGK can be found in section 2.8.2.

2.5 Multiple-relaxation-time models

The severe instability problem can be even reduced through Multiple-relaxation-time (MRT) models[67, 68]. The principle of MRT models is to deal with moments

instead of distribution functions itself. From kinetic theory it is well known that the hydrodynamics of fluid flow is directly related to mass and momenta moments, as shown in Eqn.2.9 and 2.10. Thus, the moment representation provides an intuitive way to interpret the LBM models. Given a set of discrete velocities $\boldsymbol{\xi_\alpha}, \alpha = \{0, 1, ..., N\}$ and corresponding distribution functions f_α, we can construct the same number of moments m_α through

$$m_\alpha = \langle \phi_\alpha | f \rangle = \langle f | \phi_\alpha \rangle, \quad \langle f | = (f_0, f_1, ..., f_N) \tag{2.41}$$

where the Dirac notaions of bra $\langle \cdot |$ and ket $| \cdot \rangle$ vectors are used to denote the row and column vectors, respectively. $\{|\phi_\alpha\rangle | \alpha = 0, 1, ..., N\}$ is an orthogonal dual basis set constructed by the Gram-Schmidt orthogonalization. The moment space $|m\rangle = (m_0, m_1, ..., m_N)^T$ are related to the distribution function as $|m\rangle = M|f\rangle$, by carefully selected matrix collision S, the evolution of lattice Boltzmann Equation 2.27 without force term can be rewritten

$$f_\alpha(\boldsymbol{x}_i + \boldsymbol{\xi}_\alpha \delta t, t + \delta t) - f_\alpha(\boldsymbol{x}_i, t) = -M^{-1} S[|m(\boldsymbol{x}_i, t)\rangle - m^{(0)}(\boldsymbol{x}_i, t)\rangle] \tag{2.42}$$

where S is diagonal defined as $S = diag(s_0, s_1, ..., s_N)$, and $m^{(0)}$ is the moments at the equilibrium. For D3Q19 model, the components of 19 orthogonal basis vectors are given by

$$\begin{cases} |\phi_0\rangle_\alpha = |\boldsymbol{\xi}_\alpha|^0, \\ |\phi_1\rangle_\alpha = 19|\boldsymbol{\xi}_\alpha|^2 - 30, \\ |\phi_2\rangle_\alpha = (21|\boldsymbol{\xi}_\alpha|^4 - 53|\boldsymbol{\xi}_\alpha|^2 + 24)/2, \end{cases} \tag{2.43}$$

where $|\boldsymbol{\xi}_\alpha|^0 = 1, |\boldsymbol{\xi}_\alpha| = (\xi_{\alpha x}^2 + \xi_{\alpha y}^2 + \xi_{\alpha z}^2)^{1/2}$.

$$\begin{cases} |\phi_3\rangle_\alpha = \xi_{\alpha x}, \\ |\phi_5\rangle_\alpha = \xi_{\alpha y}, \\ |\phi_7\rangle_\alpha = \xi_{\alpha z}, \end{cases} \tag{2.44}$$

$$\begin{cases} |\phi_4\rangle_\alpha = (5|\boldsymbol{\xi}_\alpha|^2 - 9)\xi_{\alpha x}, \\ |\phi_6\rangle_\alpha = (5|\boldsymbol{\xi}_\alpha|^2 - 9)\xi_{\alpha y}, \\ |\phi_8\rangle_\alpha = (5|\boldsymbol{\xi}_\alpha|^2 - 9)\xi_{\alpha z}, \end{cases} \tag{2.45}$$

$$\begin{cases} |\phi_9\rangle_\alpha = 3\xi_{\alpha x}^2 - |\boldsymbol{\xi}_\alpha|^2, \\ |\phi_{11}\rangle_\alpha = \xi_{\alpha y}^2 - \xi_{\alpha z}^2, \end{cases} \tag{2.46}$$

$$\begin{cases} |\phi_{13}\rangle_\alpha = \xi_{\alpha x}\xi_{\alpha y}, \\ |\phi_{14}\rangle_\alpha = \xi_{\alpha y}\xi_{\alpha z}, \\ |\phi_{15}\rangle_\alpha = \xi_{\alpha x}\xi_{\alpha z}, \end{cases} \tag{2.47}$$

$$\begin{cases} |\phi_{10}\rangle_\alpha = (3|\boldsymbol{\xi}_\alpha|^2 - 5)(3\xi_{\alpha x}^2 - |\boldsymbol{\xi}_\alpha|^2), \\ |\phi_{12}\rangle_\alpha = (3|\boldsymbol{\xi}_\alpha|^2 - 5)(\xi_{\alpha y}^2 - \xi_{\alpha z}^2), \end{cases} \tag{2.48}$$

$$\begin{cases} |\phi_{16}\rangle_\alpha = (\xi_{\alpha y}^2 - \xi_{\alpha z}^2)\xi_{\alpha x}, \\ |\phi_{17}\rangle_\alpha = (\xi_{\alpha z}^2 - \xi_{\alpha x}^2)\xi_{\alpha y}, \\ |\phi_{18}\rangle_\alpha = (\xi_{\alpha x}^2 - \xi_{\alpha y}^2)\xi_{\alpha z}, \end{cases} \tag{2.49}$$

The corresponding 19 moments $\{m_\alpha | \alpha = 0, 1, ..., 18\}$ are listed as

$$|m\rangle = (\rho, e, \varepsilon, j_x, q_x, j_y, q_y, j_z, q_z, 3p_{xx}, 3\pi_{xx}, p_{ww}, \pi_{ww}, p_{xy}, p_{yz}, p_{xz}, m_x, m_y, m_z)^T \tag{2.50}$$

The transforming matrix M can be assembled by evaluating $|\phi\rangle_\alpha$. The collision matrix S in 2.42 is given by

$$S = (0, s_1, s_2, 0, s_4, 0, s_4, 0, s_4, s_9, s_{10}, s_9, s_{10}, s_{13}, s_{13}, s_{13}, s_{16}, s_{16}, s_{16})^T \tag{2.51}$$

The recommended values for the S as well as the equilibrium values for non-conserved moments are given Ref[67, 68].

2.6 Units choice analysis and conversion

As pointed out in Section 2.3, lattice units are commonly used in literature for convenience. Knowing how to convert the physical units into lattice Boltzmann units is important in modeling and result interpretation, and vise versa. However, the conversion process is quite confusing for the new beginners. That is why we put a short discussion on units conversion here. The following analysis is mainly inspired by Ref [69]. Generally, two criteria should be followed in the parameter choice in any modeling process. Firstly, the numerical model should represent the physical system we are trying to study. Take the incompressible Navier-Stokes equations as an example, that requires us to make sure the Reynolds number (Re) is the same for both physical system and lattice Boltzmann model. Secondly, the parameters should be fine tuned in order to reach the designated accuracy. Lattice Boltzmann method is generally considered as a second order accurate scheme in both space and time for the simulation of weakly compressible, athermal flows at small Mach numbers. Ref [69] recommends $\delta t \sim \delta x^2$. However, in multiblock grid refinement of LB method, external velocity in lattice units would be kept the same across different grid blocks, which would require $\delta t \sim \delta x$.

In practice, typically $\tau, \delta x,$ and ρ are three parameters provided by the user for athermal flows, among which the first two are free parameters to tune. υ can

be calculated from Eqn (2.29) with τ given. The time step

$$\delta t = \upsilon \delta x^2 / \upsilon_p \tag{2.52}$$

where υ_p is the kinetic viscosity of physical fluids. So far all three physical values for δx, δt, and ρ have been identified. During the units conversion, we should follow a general principle that any quantity in lattice units should be equal to the quantity in physical units divided by a scaling factor. The scaling factor can be determined by checking the dimensions of the quantity. To clarify these procedures, we will consider an example of red blood cell (RBC) transport modeling in a microfluidic channel with detailed calculation steps. The fluid in the microfluidic channel is assumed to be water with a density of $\rho = 1000 kg/m^3$ and kinetic viscosity of $\upsilon_p = 10^{-6} m^2/s$. In order to have enough resolution for RBC motion, let $\delta x = 0.1 \mu m$. The relaxation parameter τ is usually taken as $\tau = 1$ considering LBM stability and efficiency. Using Eqn (2.29) we get $\upsilon = 0.167$ and then time step $\delta t = 1.67 \times 10^{-9} s$. Typically the Young's modulus E of RBC membrane is about $E = 5 \times 10^{-6} N/m$. Knowing the dimension of E is $[kg/(s^2)]$, we can get the corresponding scaling factor $E_{scale} = \rho dx^3/dt^2 = 0.3586 N/m$. In the end, we get the Young's modulus in lattice units $E_{lb} = E/E_{scale} = 1.39 \times 10^{-5}$. Noted that E_{lb} is dimensionless. That is the example of converting Young's modulus for the application of cell modeling. Note that the value δt is in nanoscale, which requires us to run many iterations (in order of $\sim 10^9$) in order to model RBC behavior in seconds. That is indeed the case. Fortunately, we can tune two free parameters to increase δt, either increasing δx or decreasing τ. However, generally the recommended range of τ is between 0.8 and 1, depending on Reynolds numbers[70]. τ close to 0.5 or greater than 1 will result in inaccuracies in the simulation. One of the tricks to accelerate the low Re simulation is to scale the Re number by a factor of n. The reasoning behind it is that the time scale in highly viscous flow does not

depend on the Reynolds number as long as it is relatively small[71]. e.g., physics of the flow at the $Re = 0.001$ is very similar to the flow with a large $Re = 0.1$. In this case the Re has been scaled up by 100 times. The simulation results at $Re = 0.1$ at time $t = 0.1s$ should be interpreted as the results from $Re = 0.001$ at time $t = 10s$, which is 100 times bigger based on the relation shown in Eqn. 2.52. Following the discussion presented in [70], we can find the relationship between dimensionless number Re, Ma and lattice resolution N and speed of sound c_s

$$\frac{Ma}{Re} = \frac{1}{c_s}\frac{\tau - \frac{1}{2}}{N} \tag{2.53}$$

Ideally all the dimensionless parameters in the LB simulation should exactly match the physical system we want to model. However, this is very difficult to achieve(e.g., at the cost of long simulation time or even impossible with many dimensionless numbers). It is also not the goal of mesoscopic simulation approach[71, 72]. In practice, we can sacrifice the Mach number (Ma) to gain large time steps (recall $\delta t \propto Ma^2$) as long as the compressibility does not significantly affect the simulation results. More details on simulation parameter selection can be found on Ref[70, 71].

2.7 Boundary Conditions

For general partial differential equations (PDE), boundary and initial conditions are required in order to determine the unique solution. In Navier-Stokes equations, velocity and pressure boundary conditions are usually specified at the boundaries at the initial time $t = 0$. However, it is not that straight forward to deal with the boundary conditions in LB simulations, as the macroscopic variables have to be translated into the density distribution functions. Translating the macroscopic variables into density distribution is not one to one mapping, e.g., one set of fluid velocity and density could refer to many different density distributions, as long as

the density distribution satisfies Eqn (2.31) and (2.32). For the boundary nodes, the population distribution can be classified into two categories, the incoming and outgoing groups. The incoming distributions refer to those distribution along the velocity vector $\boldsymbol{\xi}_\alpha$ such that

$$\boldsymbol{\xi}_\alpha \cdot \boldsymbol{n} < 0 \tag{2.54}$$

where $\boldsymbol{n}$ is the outward surface norm of the boundary. Similarly, the velocity vector of outgoing distributions satisfy $\boldsymbol{\xi}_\alpha \cdot \boldsymbol{n} > 0$. During streaming step, the outgoing distributions come from the adjacent internal fluid nodes, while the incoming distribution should have come from the nodes inside the wall (or outside of the fluid domain) which does not exist. So the incoming distribution on the boundary nodes is unknown. To solve this, many constrictions have been proposed to find the closed solution to the density (or population) distribution function (see section 2.7.2 for details). In what follows, we will discuss the most widely used boundary conditions, such as nonslip boundary conditions, velocity boundary conditions, and pressure boundaries.

2.7.1 Non-slip boundary conditions

Nonslip boundary conditions refer to the boundaries where the fluid velocity is zero. It is widely used to model stationary solid walls. A bounce back algorithm is commonly used to model this type of boundaries. The name bounce back came from an intuitive idea that a hard wall reflects particles back to where they originally came from. There are two types of bounce back algorithms, namely full way bounce back and half way bounce back. The full way bounce back is easy to implement; just reverse all the populations on the lattice site during the collision process. The algorithm is illustrated in Fig. 2.3

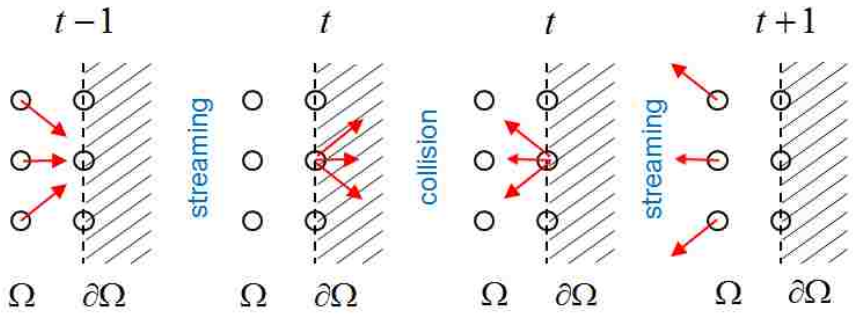


Figure 2.3: The full way bounce back algorithm. Ω is the fluid domain, $\partial\Omega$ is the boundary.

Since all the population at the boundary sites have been reversed, it can be verified that there is no tangent or normal flux to the boundary surface. The full way bounce back algorithm is local and very easy to program. However, it is only first order accurate because of the one-sided character of the streaming operator. The half way bounce back is credited with second order accuracy but with some complications. The half way bounce back is to inverse all the populations during streaming step. The unknown populations at the boundaries directly come from the adjacent nodes, as illustrated in Fig. 2.4

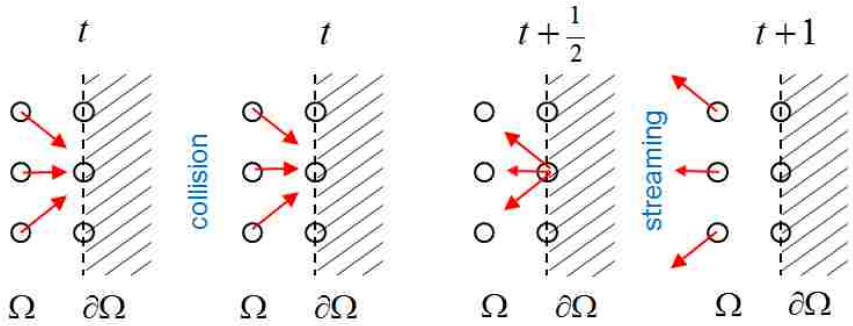


Figure 2.4: The half way bounce back algorithm

2.7.2 Velocity and pressure boundaries

Another common boundary condition is to specify the velocity directly on the boundary, which is usually called *Dirichlet boundary condition* in mathematics. Bounce back idea was also used to specify moving walls in LBM. In this section,

we will introduce two commonly used boundary conditions. The first one was proposed by Ladd [54]. The incoming density at the wall can be calculated by

$$f_i(\boldsymbol{x}, t+1) = f^*_{-i}(\boldsymbol{x}, t) - 2w_i\rho(\boldsymbol{x}, t)\frac{\boldsymbol{u}_w \cdot \boldsymbol{\xi}_i}{c_s^2} \tag{2.55}$$

where $f_i(\boldsymbol{x}, t+1)$ is the unknown density distribution at the moving wall in the direction of $\boldsymbol{\xi}_i$, $f^*_{-i}(\boldsymbol{x}, t)$ is the post collision density distribution in the opposite direction of $\boldsymbol{\xi}_i$. Here $-i$ denotes the opposition direction of i. $\boldsymbol{u}_w$ is the velocity of the moving wall. The Ladd velocity boundary conditions is also shown in Fig. 2.5. The unknown density distribution f_i and post collision density distribution f_{-i} are shown in dashed frames.

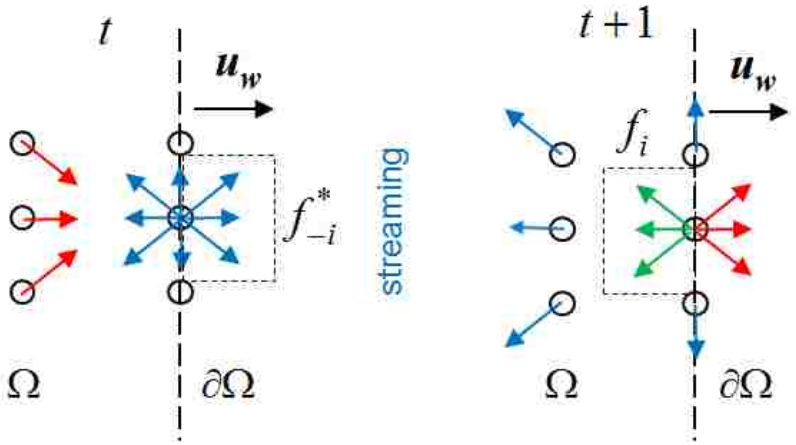
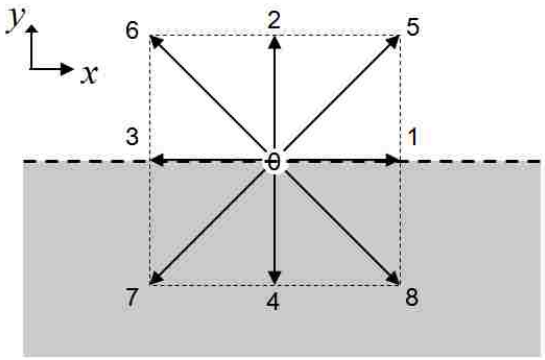


Figure 2.5: Illustration of moving wall boundary conditions based on modified bounce back rule

The second one was proposed by Q. Zou and X. He[73]. To determine the incoming velocities on the boundary, their velocity boundary conditions are based on the idea of bounce back of the non-equilibrium part of the distribution function. Let us consider a lattice node on the bottom boundary surface, as shown in Fig. 2.6. Based on Eqn (2.54), we can see $f_2, f_5, \text{and} f_6$ are the unknown incoming distributions. From Eqn (2.31) and (2.32), we have

$$f_2 + f_5 + f_6 = \rho - (f_0 + f_1 + f_3 + f_4 + f_7 + f_8) \tag{2.56}$$



H

Figure 2.6: A lattice node on the bottom boundary surface. The shaded area is the boundary wall.

$$f_5 - f_6 = \rho u_x - (f_1 - f_3 + f_8 - f_7) \tag{2.57}$$

$$f_2 + f_5 + f_6 = \rho u_y + (f_4 + f_7 + f_8) \tag{2.58}$$

Combination of Eqn (2.56) and Eqn (2.58) gives

$$\rho = \frac{1}{1 - u_y} f_0 + f_1 + f_3 + 2(f_4 + f_7 + f_8) \tag{2.59}$$

However, there is not enough informationt to find $f_2, f_5, \text{and} f_6$. In Ref[73], the bounce back rule for the non-equilibrium part of the particle distribution normal to the boundary surface is applied. In this case, $f_2 - f_2^{(0)} = f_4 - f_4^{(0)}$ is assumed. With this assumption we can find f_2, f_5, f_6 through

$$\begin{aligned} f_2 &= f_4 + \frac{2}{3}\rho u_y \\ f_5 &= f_7 - \frac{1}{2}(f_1 - f_3) + \frac{1}{2}\rho u_x + \frac{1}{6}\rho u_y \\ f_6 &= f_8 + \frac{1}{2}(f_1 - f_3) - \frac{1}{2}\rho u_x + \frac{1}{6}\rho u_y \end{aligned} \tag{2.60}$$

Similarly, the unknown distributions can be found following the procedure described above. For reference, the final result for left, right, top, and bottom boundaries are listed below.

left
$\rho = \frac{1}{1-u_x}(f_0 + f_2 + f_4 + 2(f_3 + f_6 + f_7))$
$f_1 = f_3 + \frac{2}{3}\rho u_x$
$f_5 = f_7 - \frac{1}{2}(f_2 - f_4) + \frac{1}{2}\rho u_y + \frac{1}{6}\rho u_x$
$f_8 = f_6 + \frac{1}{2}(f_2 - f_4) - \frac{1}{2}\rho u_y + \frac{1}{6}\rho u_x$
right
$\rho = \frac{1}{1+u_x}(f_0 + f_2 + f_4 + 2(f_1 + f_5 + f_8))$
$f_3 = f_1 - \frac{2}{3}\rho u_x$
$f_7 = f_5 + \frac{1}{2}(f_2 - f_4) - \frac{1}{2}\rho u_y - \frac{1}{6}\rho u_x$
$f_6 = f_8 - \frac{1}{2}(f_2 - f_4) + \frac{1}{2}\rho u_y - \frac{1}{6}\rho u_x$
top
$\rho = \frac{1}{1+u_y}(f_0 + f_1 + f_3 + 2(f_2 + f_5 + f_6))$
$f_4 = f_2 - \frac{2}{3}\rho u_y$
$f_7 = f_5 + \frac{1}{2}(f_1 - f_3) - \frac{1}{2}\rho u_x - \frac{1}{6}\rho u_y$
$f_8 = f_6 - \frac{1}{2}(f_1 - f_3) + \frac{1}{2}\rho u_x - \frac{1}{6}\rho u_y$
bottom
$\rho = \frac{1}{1-u_y}(f_0 + f_1 + f_3 + 2(f_4 + f_7 + f_8))$
$f_2 = f_4 + \frac{2}{3}\rho u_y$
$f_5 = f_7 - \frac{1}{2}(f_1 - f_3) + \frac{1}{2}\rho u_x + \frac{1}{6}\rho u_y$
$f_6 = f_8 + \frac{1}{2}(f_1 - f_3) - \frac{1}{2}\rho u_x + \frac{1}{6}\rho u_y$

Table 2.3: Unknown distribution reference table for left, right, top, and bottom boundaries

The same concept can be applied to D3Q19 as well. The detailed derivation and calculation formula for the incoming population distribution can be found in Ref [74]. The 3D code developed in our group also followed this reference.

Pressure boundary conditions are also widely used in LBM. Here the pressure boundary conditions are referred to pressure difference, because pressure itself does not drive the fluid flow. In LBM, applying pressure boundary condition is essentially to specify the density at the boundaries. Generally the fluid density in LBM is initialized as unity over the whole fluid domain. In 3D LBM, due to the continuity equation 2.19, we can only specify three of the four unknowns (u, v, w, p) on the boundary. In code implementation process, we generally convert pressure difference into LB units first, and then using equation of state 2.25 to convert pressure difference into density difference next, and add the density difference to the boundary that has higher pressure. The formula to convert pressure into

density is shown in Eqn.2.61

$$\rho = 1 + \frac{1}{c_s^2}\frac{p}{\delta\rho}\frac{\delta t^2}{\delta x^2} \tag{2.61}$$

where p is the pressure difference in physical scale in units of Pa, $\delta\rho$ is the density scaling factor, δt and δx are the time and length scaling factor, respectively. c_s^2 is the square of speed of sound.

2.8 Fluid flow benchmark

2.8.1 2D channel flow

We implemented the LBM algorithm and the 2D code was released over github (https://github.com/TJFord/iblb2d) as an open source code. To test the fluid solver, we created a 2D channel flow case with 20×20 lattices over the length and width. A parabolic fluid velocity with $V_{max} = 0.02$ was applied at the inlet. A periodic boundary condition was used for the outlet. The top and bottom boundaries were assumed to be nonslip using bounce back algorithm. The relaxation parameter $\tau = 1$. We ran the simulation to time step 1200, where the fluid velocity already reached the steady state. The fluid velocity across the channel at the middle point of the length were selected and compared with the theoretical parabolic profile. The comparison between the simulation and the theoretical values are shown in Fig. 2.7 As shown from the figure, the numerical simulation data agree well with the theoretical predictions. The relative error of the maximum velocity is 0.4%. Thus, it states that the code can correctly solve the fluid flow.

2.8.2 2D flow past cylinder

To demonstrate the bounce back boundary conditions, a 2D vortex shedding flow was created. The 2D channel has length of 400 and width 100. The diameter of

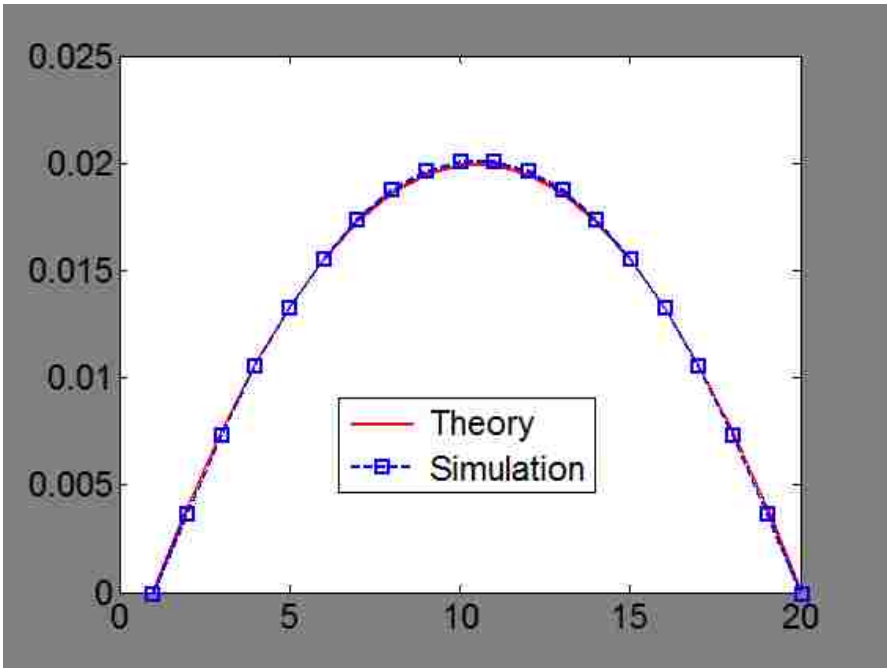


Figure 2.7: Comparison between theoretical values and the simulation data of the fluid velocity across the channel at the middle of the channel length

the cylinder is 20. The cylinder center was positioned at point (81, 53), as shown in the Fig.2.8.

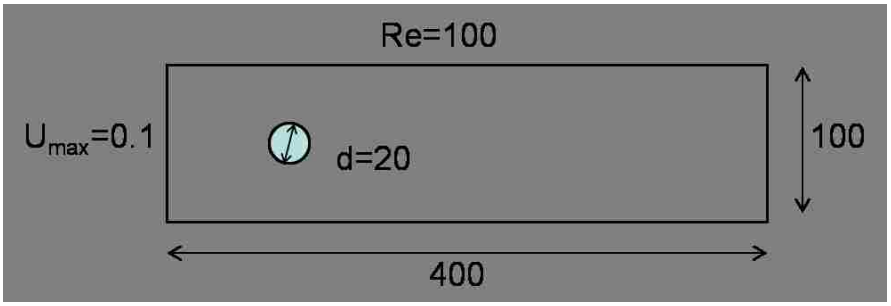
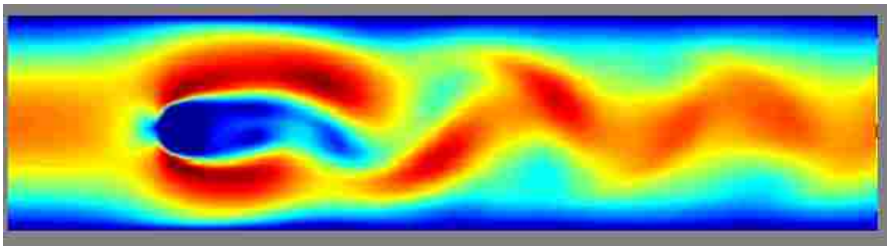


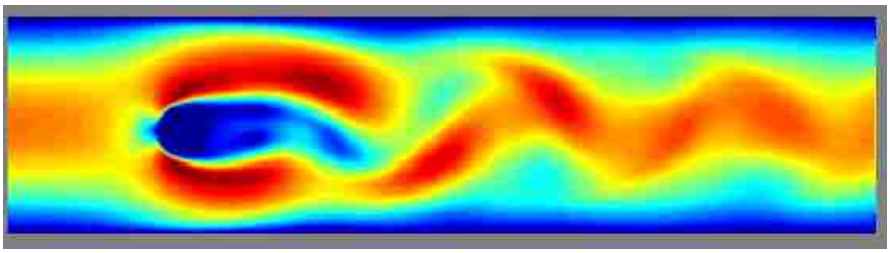
Figure 2.8: The geometry and the fluid flow settings for the vortex shedding simulation. All the parameters are given in lattice Boltzmann units.

The boundary of the cylinder and channel side wall were modeled using bounce back algorithm. Zou/He velocity boundary conditions were applied at the inlet with a parabolic velocity profile. The maximum velocity is 0.1 in lattice Boltzmann units. The Re with cylinder diameter as the characteristic length is 100. The relaxation time $\tau = 0.566$. Both the BGK and regularized BGK collision scheme were used. The fluid velocity was initialized with a parabolic flow profile

along the whole channel. The fluid flow distribution for BGK and regularized BGK simulation is shown in Fig.2.9a and 2.9b, respectively. It looks like almost



(a) BGK collision scheme



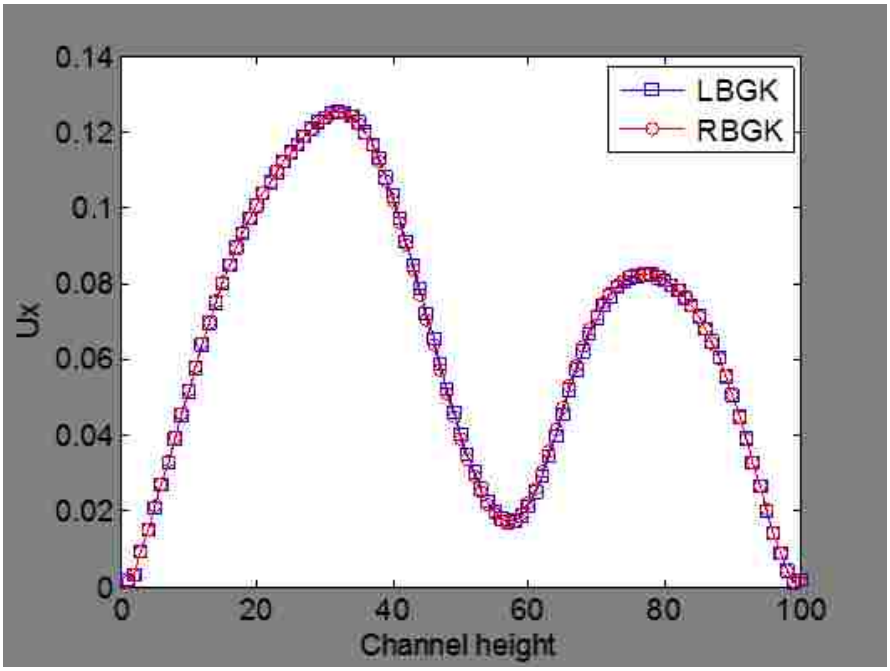
(b) Regularized BGK collision scheme

Figure 2.9: The fluid velocity field and the alternatively shed vortex after the cylinder in the flow past cylinder simulation. (a) BGK collision scheme; (b) Regularized BGK collision scheme

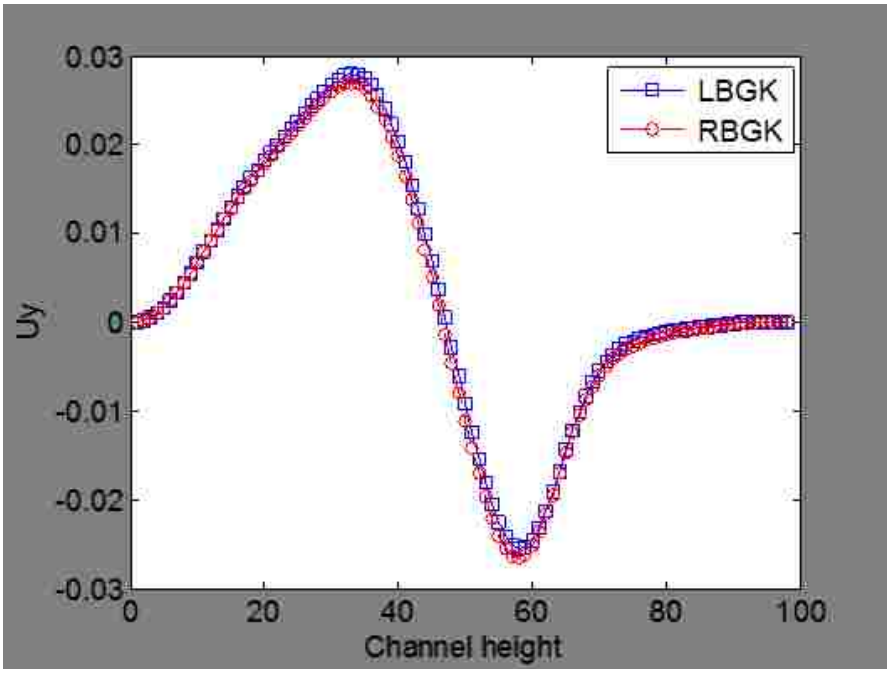
identical for the flow flied from Fig.2.9. The velocity components in x, y direction along the line $x = 150$ were selected for comparison for both BGK and regularized BGK scheme. The velocity comparison plot are shown in Fig. 2.10. The velocity distribution from both methods agree well.

2.8.3 3D lid driven cavity flow

A 3D lid driven cavity flow was also created to bench mark the 3D fluid solver. The fluid domain is a cubic box. Each side of the cubic is $1m$. A fluid velocity of $1 \times 10^{-4} m/s$ was applied to the lid. The other 5 boundary faces were set to be nonslip using the boundary conditions presented in Ref. [74]. The fluid was assumed to be water with kinetic viscosity of $1 \times 10^{-3} m^2/s$. Thus, the Re is 100. The cubic box was discretized with $33 \times 33 \times 33$lattices. The relaxation parameter was set as $\tau = 0.7$. The snap shot of the velocity profile at the middle plane of



(a) velocity component u_x along the channel at $x = 150$



(b) velocity component u_y along the channel at $x = 150$

Figure 2.10: The fluid velocity field along the line $x = 150$ in the flow past cylinder simulation. (a) velocity component u_x in x direction; (b) velocity component u_y in y direction.

the lid with velocity arrows were shown in 2.11. Notice that the legend shows the velocity in lattice Boltzmann units, which is different from the physical value of

$1 \times 10^{-4} m/s$. The velocity component along the lid driven direction from the lid

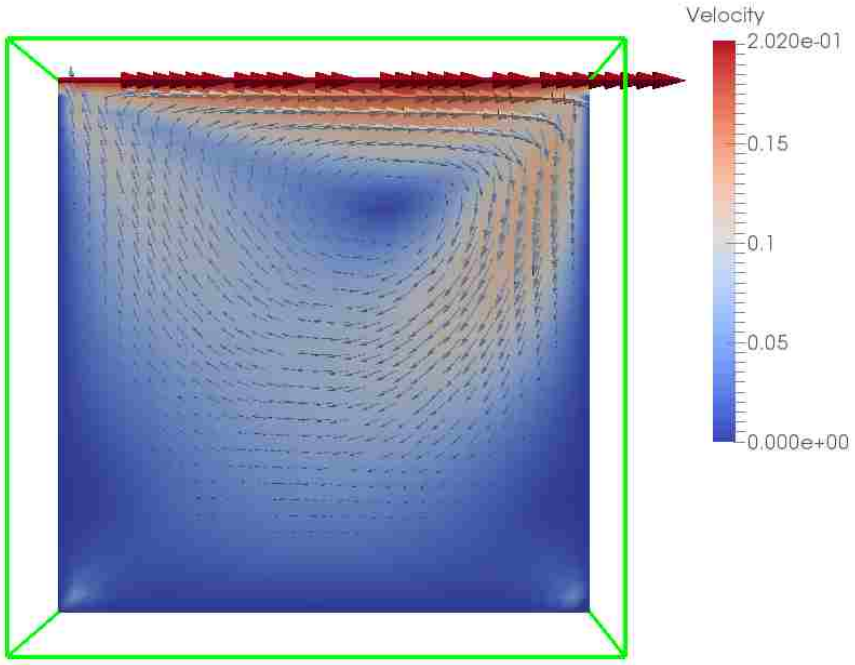


Figure 2.11: The snapshot of the flow profile at the middle plane of a lid driven cavity flow. A velocity color map is shown in the figure as well. The velocity is in lattice Boltzmann units.

to the bottom was also compared with the data presented in Jiang&Lin[75]. The velocity component and the cavity height were normalized to 1. As shown from the figure, we can see that our code can correctly reproduce the velocity field in the lid driven cavity flow.

2.9 Further discussion

So far we only scratch a little bit of Lattice Boltzmann method. One of the problems is that the time step used in microscale simulation is very small, as shown in the units conversion example in Section 2.6. The time step used is in nanoseconds. That would require a large number of time steps to achieve a few seconds in physical time. Except the method of scaling Reynolds number(see Section 2.6), another solution to speed up the code is using hierarchical grid refinement[76]. The whole fluid domain is covered with a coarse grid while a finer

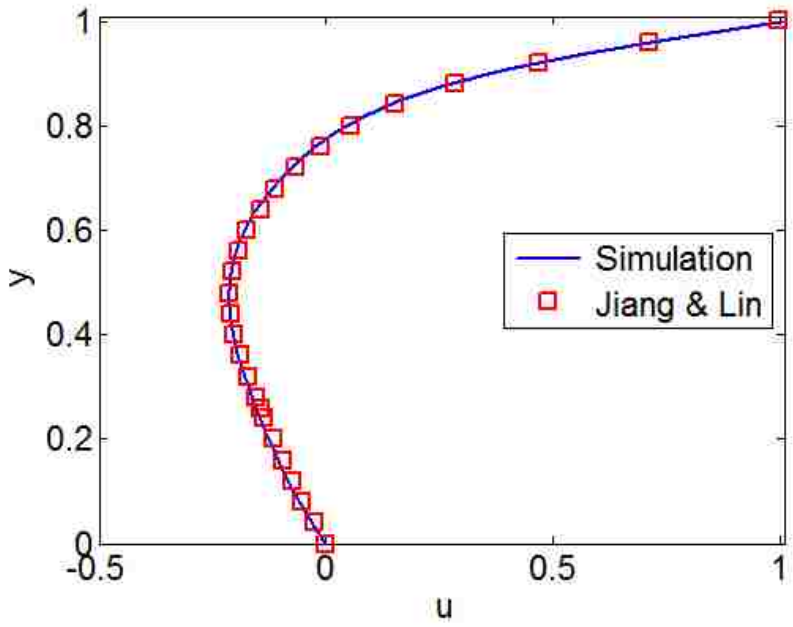


Figure 2.12: Velocity component comparison between our simulation and the data presented in Jiang& Lin[75]

grid is superposed on the coarse grid during the critical region. The calculation proceeds with large time steps over the coarse grid, while on the finer grid several time steps are performed in order to catch the same time step size in the coarse grid. Interested readers on multi-grid or multi-lattice techniques will refer to Ref[76–78].

Another nice feature of LBM is the multiphase/multicomponent flow simulation capabilities. Through introducing multiple distribution functions, we can model as many components as we can. One of the most popular multiphase flow model was Shan-Chen model proposed in Ref.[79], where a nonlocal interactions among the particles are introduced to model the interaction between different components. The Shan-Chen model was used to model rising bubbles in Ref.[80]. A review on multiphase flow modeling using LBM can be found in Ref[81]. Similarly, the heat convection can also be included by introducing one more set of distribution functions for temperature/energy[82]. Another method to model heat transfer is through introducing a lattice with more discrete velocities[83, 84]. A short

summary on Lattice Boltzmann thermal models were presented in [85, p.160].

LBM is also known for its easiness to take advantage of parallel computing, because the collision step is local to the lattice sites, while the streaming only involves the nearest neighbor lattices. Most boundary conditions are also local. The Stress evaluation is also local. Conventional two lattice algorithm consumes a lot of memory. To address this problem, an efficient swap algorithm was proposed[86]. Performance of different data layout and algorithms were summarized in Ref[87, 88]. Techniques to develop LBM code using GPU device can be found in Ref[89–91].

Chapter 3

Numerical models for red blood cells

Red blood cells (RBCs), also called erythrocytes, occupy about 40% $\sim$ 45% of the blood volume in human circulation system. They pick up oxygen(O_2) from the lung and deliver them to the body tissues through capillaries. RBCs have a bi-concave shape with a diameter of 6-8μm and a thickness of 2 μm. They also don't have nucleus. The biconcave shaped and *anucleate* structure enable RBCs to be squeezed through capillaries as small as 2 μm in diameter while keep RBCs volume and surface area relatively unchanged. Thus, there is no surprise that RBCs transport in blood flow has been the focus of both biomedical and mechanical research for many decades. If we look at the molecular level of the RBC membrane, it consists of a phospholipid bilayer, cholesterol molecules, transmembrane proteins and an underlying spectrin network[92]. The underlying spectrin network mainly determines the cell's membrane shape and mechanical behavior. To capture the realistic mechanical and rheological characteristics of RBCs, mechanical models should have the capabilities to address membrane elastic and viscous properties, bending resistance, and the viscosities of the external fluids and cytoplasm. In modeling, a cell membrane is usually discretized into a 2D mesh consisting of

many individual triangular surfaces that are connected by nodes $\boldsymbol{X}_i$. Different energy density function or potentials will be defined for the cell membrane elastic energy, bending energy, and area and volume constraints. In mathematics, the total energy of a cell membrane is

$$V(\boldsymbol{X}_i) = V_{stretch} + V_{bending} + V_{area} + V_{volume} \tag{3.1}$$

The nodal force derived from the potential energy is given by

$$\boldsymbol{F}_i = -\frac{\partial V(\boldsymbol{x}_i)}{\partial \boldsymbol{x}_i} \tag{3.2}$$

A 2D illustration of the cell membrane with stretching and bending energy is shown in Fig.3.1. Area and volume potential are not drawn in the figure.

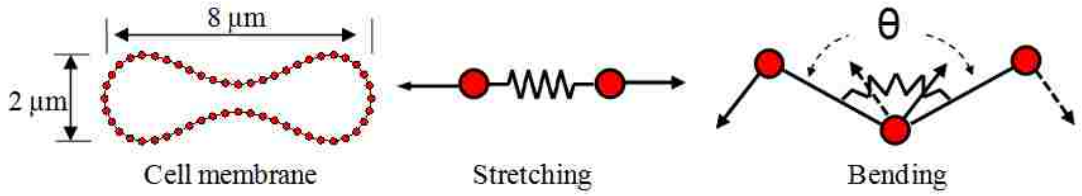


Figure 3.1: Illustration of a 2D red blood cell membrane with stretching and bending energies.

In this chapter, we will mainly focus on the modeling techniques of RBCs, particularly on different energy potentials related to membrane elastics, bending, and area and volume constraints.

3.1 Cell membrane model

3.1.1 Strain Energy Density Functions

RBC membrane was first studied using a continuum mechanics approach. The membrane is modeled as a 2D elastic sheet due to the small thickness of the

membrane compared to the total surface area. A strain energy density function (ϵ^S) is typically used to model the membrane behavior under both small stress loading and large deformations. The total strain energy of the cell membrane is $E_S = \int \epsilon^S dA$. The most popular model for RBC membrane was proposed by Skalak in 1973[93] and then used by Ref.[94, 95]. In that model, the strain energy density function is defined as

$$\epsilon^S = \frac{K_s}{12}(I_1^2 + 2I_1 - 2I_2) + \frac{K_\alpha}{12} I_2^2 \tag{3.3}$$

where K_s and K_α are the membrane shear elastic shear modulus and area dilation modulus. I_1 and I_2 are strain invariants defined as

$$I_1 = \lambda_1^2 + \lambda_2^2 - 2 \tag{3.4}$$

$$I_2 = \lambda_1^2 \lambda_2^2 - 1 \tag{3.5}$$

where λ_1 and λ_2 are the eigenvalues of the deformation gradient $\boldsymbol{D}$ which is the derivative of each component of the deformed $\boldsymbol{x}$ vector with respect to each component of the reference $\boldsymbol{X}$ vector. For $\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{X})$, then

$$D_{ij} = x_{ij} = \frac{\partial x_i}{\partial X_j} = \begin{bmatrix} \frac{\partial x_1}{\partial X_1} & \frac{\partial x_1}{\partial X_2} \\ \frac{\partial x_2}{\partial X_1} & \frac{\partial x_2}{\partial X_2} \end{bmatrix} \tag{3.6}$$

Introducing the displacement vector $\boldsymbol{u}$, we have $\boldsymbol{x} = \boldsymbol{X} + \boldsymbol{u}$. Rewriting $\boldsymbol{D}$ in terms of displacement vector $\boldsymbol{u}$,

$$D_{ij} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} \frac{\partial u_1}{\partial X_1} & \frac{\partial u_1}{\partial X_2} \\ \frac{\partial u_2}{\partial X_1} & \frac{\partial u_2}{\partial X_2} \end{bmatrix} \tag{3.7}$$

It can be shown that $I_1 = Trace(\boldsymbol{D}^T\boldsymbol{D}) - 2$ and $I_2 = det(\boldsymbol{D}^T\boldsymbol{D}) - 1$. With I_1 and I_2, the strain energy density ϵ^S for each distretized surface can be found. The total strain energy for the membrane is $E_S = \sum_i A_i^{(0)} \epsilon_i^S$, where i sums over all the triangulated surfaces of the membrane. The strain force applied to node i at position $\boldsymbol{x}_i$ can be computed

$$\boldsymbol{F}_i^S = -\frac{\partial E_s(\boldsymbol{x}_i)}{\partial \boldsymbol{x}_i} \tag{3.8}$$

The derivation is tedious but straight forward. The explicit expression of nodal force for each node can be found in Ref[64].

3.1.2 Coarse grained molecular dynamics approach

Another approach is to investigate the membrane elasticity from molecular level, as first proposed by[92], and then widely used by [96, 97]. The idea is originated from Molecular dynamics that models the individual interaction between atoms. Theoretically it is possible to model the blood flow from molecular dynamics approach. However, the computational cost of molecular dynamics is very expensive for continuum scale simulations. Thus, a mesoscale simulation method is needed. Coarse grained molecular dynamics (CGMD) is one of the approaches. Many atoms are conceptualized as one molecular cluster (or virtual particle), then, an effective interaction potential between each particle will be defined to model the mechanical response of the system. Each particle will move follow Newton's law where the force field is derived from the potential energy between particles. During the coarse graining process, the details of individual atomic dynamics are lost, but the longer range temporal and spatial phenomena are preserved. For a coarse grained cell membrane model, a number of potentials have been used, from the simple linear harmonic potential to complex worm-like-chain (WLC) potential and the finitely extensible nonlinear elastic (FENE) potential. One of the advantages

of CGMD for cell membrane modeling is the simplicity in mathematical description and programming. The CGMD model has been shown to provide consistent predictions with energy density functions used in the continuum model[92, 98].

The harmonic potential for a single triangle surface is given by

$$V_{stretch} = \frac{1}{2}k_s \sum_{j=1,\ldots,N_s} \left(\frac{l_j - l_{j0}}{l_{j0}}\right)^2 \tag{3.9}$$

where k_s is the stretching constant, l_j is the length of the jth spring, and l_{j0} is the equilibrium spring length. The total stretching energy should be summed over all the triangulated surfaces. This model has been used to model malaria-infected RBCs[99], mesoscopic blood flows[100]. However, this simple linear model cannot capture the nonlinear behavior of the cell membrane. An exponential relationship for mechanical stiffness k_s related to bond stretch ratio (λ) were developed to capture the nonlinear behavior[101]. The exponential form of the spring constant can be expressed as $k_s = k_{s0}exp[2(\lambda-1)]$, where λ is the bond stretch ratio defined as $\lambda = l/l_0$.

Another approach is to use nonlinear potentials to model the membrane stretching energy. Two most popular nonlinear potentials are WLC potential and FENE potential. Explicitly, they are defined as

$$U_{WLC} = \frac{k_B T l_m}{4p}\frac{3x^2 - 2x^3}{1-x} \qquad U_{FENE} = -\frac{k_s}{2}l_m^2 log[1-x^2] \tag{3.10}$$

where $x = l/l_m \in (0,1)$, l_m is the maximum spring extension, p is the persistence length, k_B is the Boltzmann constant, T is the temperature, k_s is the FENE spring constant. These two models are very similar in terms of force strain relations. The force normalized with respect to $\frac{k_B T}{p}$ and $k_s l_m$ for WLC and FENE springs are shown in Fig.3.2.

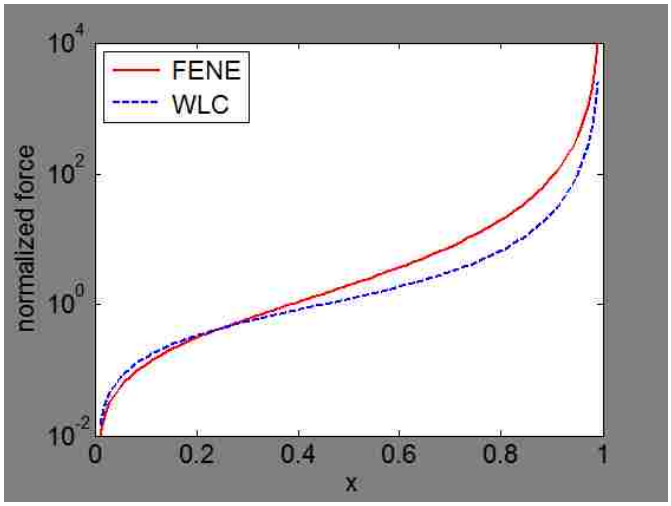


Figure 3.2: The normalized force for both WLC and FENE spring models under different stretching ratio x

Note that the force goes to infinity as the bond length approaches maximum length l_m for both the nonlinear springs. It is noted that these springs only provide attractive forces, thus they tend to reduce the cell membrane area. A repulsive force field should be combined with these nonlinear potentials. Ref[102] suggested two repulsive potentials. One is based on the surface area with a functional form of $\frac{C_q}{A_k^q}$, where A_k is the k th triangle area, q is the power, C_q is a constant determined by the equilibrium spring length. Another one is directly based on spring length with a functional form of $\frac{k_p}{l^n}$. This spring length based potential will be introduced in Eqn. 3.11.

During our test, we found that the conbination of nonlinar term with area based repulsive potential is not easy to reach a steady configuration for RBCs. Instead, we used a power repulsive potential suggested in [96].

$$U_{POW}(l) = \frac{k_p}{(n-1)l^{n-1}} \qquad n > 0, n \neq 1 \tag{3.11}$$

where l is the bond length, k_p is the repulsive stiffness. The value of k_p can be calculated by equating the attractive fore from Eqn(3.10) to the repulsive force derived from Eqn(3.11) at the equilibrium bond length. Again, we normalized the

repulsive force derived from Eqn. 3.11 with $\frac{k_p}{l_m^n}$ and plot out the force stretching ratio x relation in Fig. 3.3

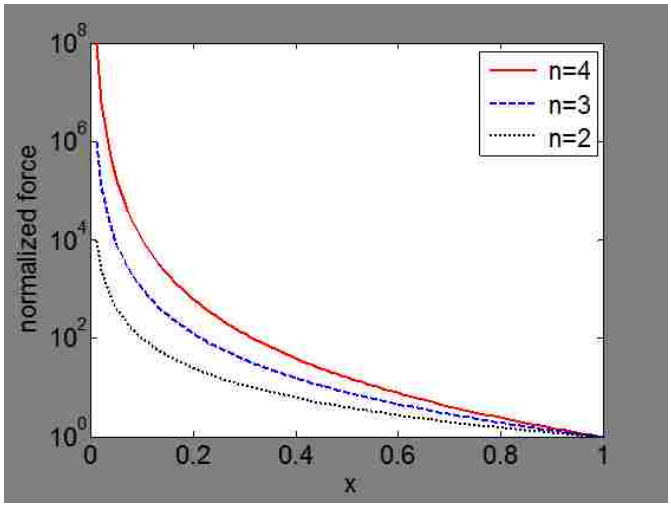


Figure 3.3: The normalized force for power law potential under different power n

One thing we should point out is that the stretching energy defined based on Eqn3.10 depends on the total number of cell membrane nodes. e.g., the maximum bond length in a coase membrane mesh, denoted by l_m^c would be different from the maximum bond length in a fine mesh, denoted by l_m^f. So are the persistent length p^c and p^f in a coarse and fine mesh. Following Ref [102, p.50], we summarize the scaling relations between the mechanical and geometrical parameters for the nonlinear springs. The equilibrium and maximum bond length are given by

$$l_0^c = l_0^f \sqrt{\frac{N^f - 2}{N^c - 2}}, \qquad l_m^c = l_m^f \sqrt{\frac{N^f - 2}{N^c - 2}} \tag{3.12}$$

where N is the number of membrane mesh nodes. Superscript f and c represent a fine and coarse mesh. To keep the shear and area compression moduli unchanged in different mesh resolutions with the same ratio x_0 of the equilibrium bond length (l_0) over the maximum bond length (l_m), the persistent length and spring constant

should follow

$$p^c = p^f \frac{l_0^f}{l_0^c}\,(WLC), \quad k_s^c = k_s^f\,(FENE), \quad k_p^c = k_p^f \left(\frac{l_0^c}{l_0^f}\right)^{n+1}\,(POW) \tag{3.13}$$

3.1.3 Bending energy and other constraints

Bending energy is defined

$$V_{bending} = \sum_{j=1...N_s} k_b(1 - cos(\theta_j - \theta_0)) \tag{3.14}$$

where k_b is the bending constant, θ_j is the instantaneous angle formed by the two outward surface norm of two adjacent triangular meshes that share the same edge j. θ_0 is the corresponding equilibrium or spontaneous angle.

During deformation, RBC's membrane and volume are relatively conserved. Area and volume conservation energy are introduced

$$V_{area} = \frac{k_g(A - A_0)^2}{2A_0} + \sum_{j=1...N_t} \frac{k_l(A_j - A_{j0})^2}{2A_{j0}} \tag{3.15}$$

$$V_{volume} = \frac{k_v(V - V_0)^2}{2V_0} \tag{3.16}$$

where k_g, k_l are the global and local area conservation potential constants; A, A_0 are the instantaneous and spontaneous total surface area of the cell membrane; A_j, A_{j0} are the instantaneous and spontaneous surface area for the jth triangle surface. k_v is the volume conservation constant, V, V_0 are the instantaneous and the equilibrium cell volume.

In this work, all the parameters are derived from a fine mesh data suggested in Ref [102, p.50], as shown in Table 3.1.

Notice that since the mesh we used in this work is not uniform. We used individual bond dependent POW stiffness constant k_p. In a coarse mesh with 642

N	$l_0(nm)$	x_0	$p(nm)$	$k_b(k_BT)$	$k_g(\mu N/m)$	$k_l(\mu N/m)$	$k_v(kpa)$
27344	75.5	2.05	18.7	50	200	200	10

Table 3.1: A standard fine cell membrane mesh parameters. All other coarse grained membrane models are derived from this one.

nodes, the $\frac{k_BTl_m}{p} = 4.0 \times 10^{-5} N/m$. The initial biconcave shape of the RBC was used as an equilibrium shape, as used in many other researches[94]

3.2 Numerical implementation of cell membrane model

With all those potential defined in Eqn(3.1), it is straight forward to derive the nodal force based on Eqn(3.2). However, it is nontrivial to implement the force calculation. This section is to give a brief summary of the nodal force calculation process. A membrane mesh is needed in both continuum approach and a coarse grained approach. The mesh is used to define the nodal position and membrane triangulation. The connectivity of each face, bond list and dihedral bond list can be done in preprocessing, which will not be discussed here. All the nodes in our discussions are listed in a counterclockwise order. A mesh for RBC membrane can be done in several ways. The simplest one is to subdivide an icosahedron several times into a target spherical mesh, and then map the spherical nodes into a biconcave shape following the formula suggested by [103].

$$z(\rho) = \pm\sqrt{1 - \frac{\rho}{r}}\left[a_0 + a_1\left(\frac{\rho}{r}\right)^2 + a_2\left(\frac{\rho}{r}\right)^4\right] \tag{3.17}$$

where $a_0 = 0.81\mu m$, $a_1 = 7.83\mu m$, $a_2 = -4.39\mu m$ and $r = 3.91\mu m$. $\rho = x^+y^2$ where x, y are the components in the spherical nodes (x, y, z). The new obtained mesh nodes for RBC membrane will be $(x, y, z(\rho))$. There are some other methods, such as energy relaxation on a constrained membrane, or using commercial

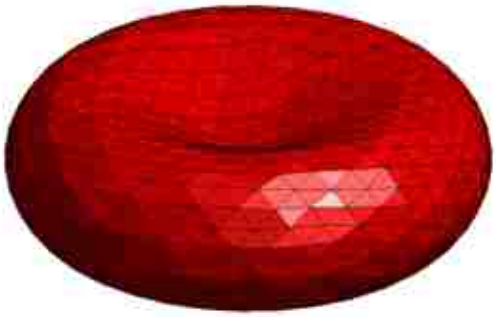


Figure 3.4: An example of Red blood cell membrane mesh

software. Interested readers can refer to [96, p. 49] and [64, p. 53]. An example of the RBC mesh with 642 nodes and 1280 faces are shown in Fig.3.4.

The harmonic spring force is omitted here due to simplicity. The derived force for WLC, FENE springs and repulsive force from a power potential are given as

$$\begin{aligned} f_{WLC}(l) &= -\frac{k_B T}{p}\left(\frac{1}{4(1-x)^2} + x - 0.25\right)\hat{\boldsymbol{l}}_{ij} \\ f_{FENE}(l) &= -\frac{k_s l}{1-x^2}\hat{\boldsymbol{l}}_{ij} \\ f_{repulsive}(l) &= \frac{k_p}{l^n}\hat{\boldsymbol{l}}_{ij} \end{aligned} \tag{3.18}$$

where $\hat{\boldsymbol{l}}_{ij}$ is the unit vector for bond connecting node i, j. The bending force is much more complicated than the spring force. For the ease of discussion, Fig.3.5 shows two adjacent surfaces shared an edge where the bending energy is defined. The plane angle is defined as θ between surface A_1 and A_2. Two surfaces sharing an edge have 4 nodes with position vector $\boldsymbol{p}_i, i = 1, 2, 3, 4$, as shown in the figure. The edge vector $\vec{a}_{ij}$ is defined as $\vec{a}_{ij} = \boldsymbol{p}_i - \boldsymbol{p}_j$. The surface norm $\vec{\xi}$ of A_1 is $\vec{\xi} = \vec{a}_{21} \times \vec{a}_{32}$, and surface norm $\vec{\varsigma}$ of A_2 is $\vec{\varsigma} = \vec{a}_{34} \times \vec{a}_{24}$, and the corresponding surface area $A_1 = |\vec{\xi}|/2, A_2 = |\vec{\varsigma}|/2$. The bending force is

$$f_{x_i} = -\frac{\partial k_b(1 - cos(\theta - \theta_0))}{\partial x_i} = -k_b sin(\theta - \theta_0)\frac{\partial \theta}{\partial x_i} \tag{3.19}$$

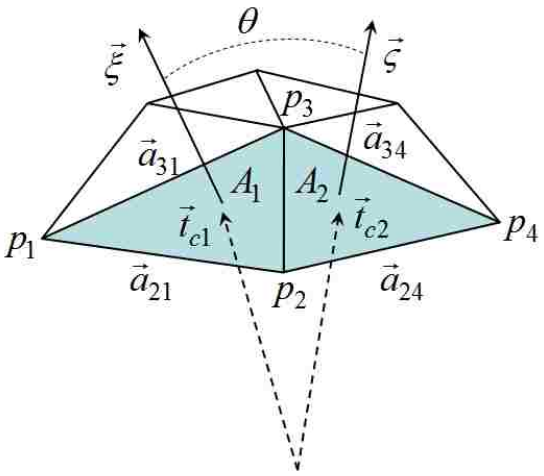


Figure 3.5: Illustration of the geometry of adjacent surfaces where bending energy is defined. The normal is pointing outside which is formed by cross product of two edge vectors.

where $\dfrac{\partial\theta}{\partial x_i}$ can be evaluated as

$$\frac{\partial\theta}{\partial x_i} = \frac{\partial\left[arccos\left(\frac{\vec{\xi}}{|\vec{\xi}|}\cdot\frac{\vec{\varsigma}}{|\vec{\varsigma}|}\right)\right]}{\partial x_i} = -\frac{1}{\sqrt{1-cos^2\theta}}\frac{\partial\left[\left(\frac{\vec{\xi}}{|\vec{\xi}|}\cdot\frac{\vec{\varsigma}}{|\vec{\varsigma}|}\right)\right]}{\partial x_i} \tag{3.20}$$

reorganizing them in nodal component format,

$$\begin{aligned}
(f_{x1}, f_{y1}, f_{z1}) &= b_{11}(\vec{\xi}\times\vec{a}_{32}) + b_{12}(\vec{\varsigma}\times\vec{a}_{32}) \\
(f_{x2}, f_{y2}, f_{z2}) &= b_{11}(\vec{\xi}\times\vec{a}_{13}) + b_{12}(\vec{\xi}\times\vec{a}_{34} + \vec{\varsigma}\times\vec{a}_{13}) + b_{22}(\vec{\varsigma}\times\vec{a}_{34}) \\
(f_{x3}, f_{y3}, f_{z3}) &= b_{11}(\vec{\xi}\times\vec{a}_{21}) + b_{12}(\vec{\xi}\times\vec{a}_{42} + \vec{\varsigma}\times\vec{a}_{21}) + b_{22}(\vec{\varsigma}\times\vec{a}_{42}) \\
(f_{x4}, f_{y4}, f_{z4}) &= b_{12}(\vec{\xi}\times\vec{a}_{23}) + b_{22}(\vec{\varsigma}\times\vec{a}_{23})
\end{aligned} \tag{3.21}$$

where $b_{11} = -\dfrac{\beta_b cos\theta}{|\vec{\xi}|^2}$, $b_{12} = -\dfrac{\beta_b}{|\vec{\xi}||\vec{\varsigma}|}$, $b_{22} = -\dfrac{\beta_b cos\theta}{|\vec{\varsigma}|^2}$
with $\beta_b = -\dfrac{k_b(sin\theta cos\theta_0 - cos\theta sin\theta_0)}{\sqrt{1-cos^2\theta}}$.

Before we derive the nodal force for area conservation, it is helpful to evaluate $\dfrac{\partial A_j}{\partial x_i}$. Notice that the surface area $A_j = |\vec{\xi}|/2 = \frac{1}{2}\sqrt{\xi_x^2+\xi_y^2+\xi_z^2}$ where the surface

norm $\vec{\xi} = \vec{a}_{21} \times \vec{a}_{31}$. Then, we have

$$\frac{\partial A_j}{\partial x_i} = \frac{1}{2\sqrt{\xi_x^2 + \xi_y^2 + \xi_z^2}} \left(\xi_x \frac{\partial \xi_x}{\partial x_i} + \xi_y \frac{\partial \xi_y}{\partial x_i} + \xi_z \frac{\partial \xi_z}{\partial x_i} \right) \tag{3.22}$$

Considering Eqn(3.22), we have the nodal force for area conservation

$$\begin{aligned} f_{xi} &= -\frac{\partial [k_g (A - A_0)^2 / (2A_0)]}{\partial x_i} \\ &= -\frac{k_g (A - A_0)}{A_0} \frac{\partial A}{\partial x_i} \\ &= \beta_g \sum_{j=1,\ldots N_t} \frac{\partial A_j}{\partial x_i} \\ &= \beta_g \sum_{j=1,\ldots N_t} \frac{1}{4A_j} \left(\xi_x \frac{\partial \xi_x}{\partial x_i} + \xi_y \frac{\partial \xi_y}{\partial x_i} + \xi_z \frac{\partial \xi_z}{\partial x_i} \right) \end{aligned} \tag{3.23}$$

where $\beta_g = -k_g(A - A_0)/A_0$. It is convenient to express the force contribution form each surface 3.23 in the form of

$$\begin{aligned} (f_{x1}, f_{y1}, f_{z1}) &= \alpha_g (\vec{\xi} \times \vec{a}_{32}) \\ (f_{x2}, f_{y2}, f_{z2}) &= \alpha_g (\vec{\xi} \times \vec{a}_{13}) \\ (f_{x3}, f_{y3}, f_{z3}) &= \alpha_g (\vec{\xi} \times \vec{a}_{21}) \end{aligned} \tag{3.24}$$

where $\alpha_g = \beta_g/(4A_j), j = 1, \ldots N_t$. The total global area conservation force should sum over all the triangular surfaces. Similarly, the local area conservation can also be calculated using the above approach. The local area conservation force has exactly the same functional form as Eqn (3.24) but with a different coefficient $\alpha_l = -k_l(A_j - A_{j0})/(4A_j A_{j0})$.

The global volume conservation force is

$$\begin{aligned} f_{xi} &= -\frac{\partial[k_v(V-V_0)^2/(2V_0)]}{\partial x_i} \\ &= -\frac{k_v(V-V_0)}{V_0}\frac{\partial V}{\partial x_i} \\ &= \beta_v \sum_{j=1,\ldots N_t} \frac{\partial V_j}{\partial x_i} \end{aligned} \tag{3.25}$$

where $V_j = \frac{1}{6}\vec{\xi_j} \cdot \vec{t_c}$, and $\vec{t_c}$ is the position vector at the center of mass of the j triangle defined as $\vec{t_c} = (\vec{p}_{j1} + \vec{p}_{j2} + \vec{p}_{j3})/3$. The volume conservation contributions to the nodal force for a single triangle are

$$\begin{aligned} (f_{x1}, f_{y1}, f_{z1}) &= \beta_v(\vec{t_c} \times \vec{a}_{32} + \vec{\xi}/3) \\ (f_{x2}, f_{y2}, f_{z2}) &= \beta_v(\vec{t_c} \times \vec{a}_{13} + \vec{\xi}/3) \\ (f_{x3}, f_{y3}, f_{z3}) &= \beta_v(\vec{t_c} \times \vec{a}_{21} + \vec{\xi}/3) \end{aligned} \tag{3.26}$$

3.3 Cell model benchmark test

Any cell model should be able to capture the cell mechanical properties. One of the standard validation approaches for cell model is the stretching test. Experimentally, optical tweezers were used to apply force to a two patches of cell membrane at the opposite ends. Then, the cell was pulled away under the control of the optical tweezer. The force and extension curve and other elastic properties were measured during the stretching test[104]. The experimental data for force extension curve has been used by many researchers to benchmark their cell models[92, 96, 105].

The cell membrane used here consists of 642 nodes with a diameter of 7.82 μm using the Eqn.(3.17) . A pair of force was applied to 5% of the total nodes on each side of the cell membrane. So in total there was 10% of the nodes subjecting to external forces. The applied force was shared uniformly among all the nodes, as shown in Fig.3.6. The cell membrane positions was updated through the immersed

boundary method(IBM), with details shown in Chapter 5. The fluid here served as a damping for the whole system. Without the fluid, we found that cell shape is not easy to evolve to a steady shape due to the elastic nature of the membrane model.

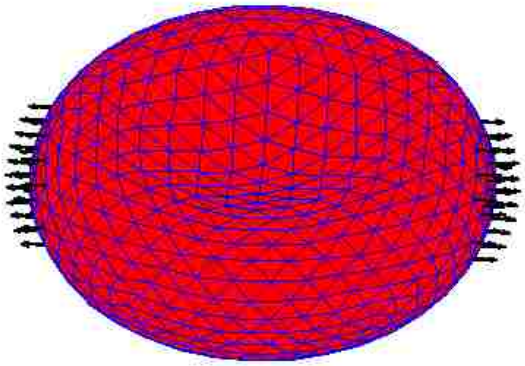
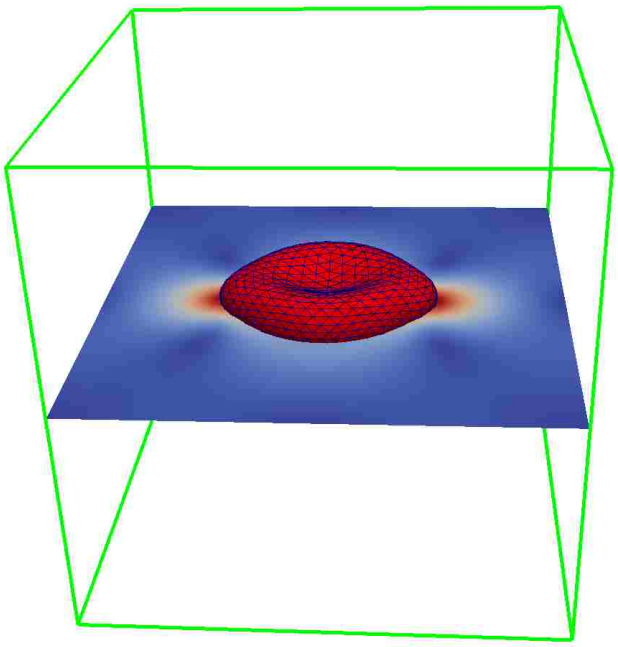


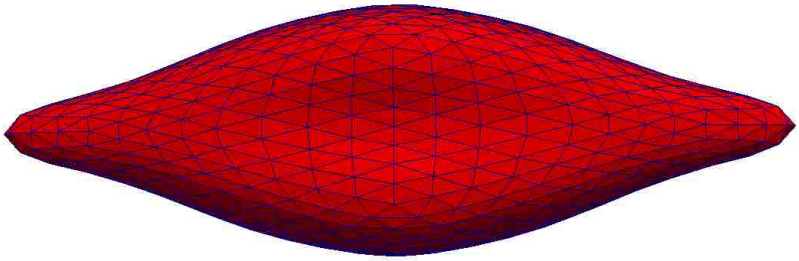
Figure 3.6: A pair of 200 pN force applied to 66 nodes on the cell membrane, with 33 nodes on each side. The cell membrane is meshed with 642 nodes in total.

The fluid domain is a cubic box with dimensions of $20\mu m \times 20\mu m \times 20\mu m$. Each side was discretized with 50 lattices. Open boundary conditions $\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{n}} = 0$ were applied to all the six surfaces of the fluid box. The relaxation parameter for LBM solver is $\tau = 1$ and the time step used here is $2.67 \times 10^{-8} s$. When the force were applied to the cell membrane, it was spread out into the local fluid. As shown in Fig.3.7a, after stretching, the fluid near the cell membrane where the force pair is applied has relatively large velocity compared to distant fluid, as indicated by the color. The local fluid motion would be used to update the cell membrane positions. As the cell deforms, the cell membrane would generate resistant force to balance the external applied force. When the net force approaches toward zero, the local fluid would be quiescent. The cell membrane would reach a steady shape as well. During the stretching process, the applied force would induce local fluid vortex. The steady cell shape under applied force pair of 200 pN would achieved in the end, as shown in Fig. 3.7b.

We applied several pairs of forces $(25pN, 50pN, 100pN, 150pN, 200pN)$ during



(a) Cell stretching model setup



(b) Steady shape of cell membrane under applied force of 200 pN

Figure 3.7: The cell stretching test model setup and the steady shape of the cell under applied force pair of 200 pN

the stretching test. The longest diameter and transverse diameter were measured when the cell reached a steady state. The force extension curve as well as the experimental data obtained from [92] is shown in Fig. 3.8. The curve on the top is the longest diameter along the stretching direction, the curve at the bottom is the cell transverse diameter in the direction perpendicular to stretching. As shown from the figure, the elongation diameter agrees very well with the experiments. The transverse diameter is slightly larger than the experimental data.

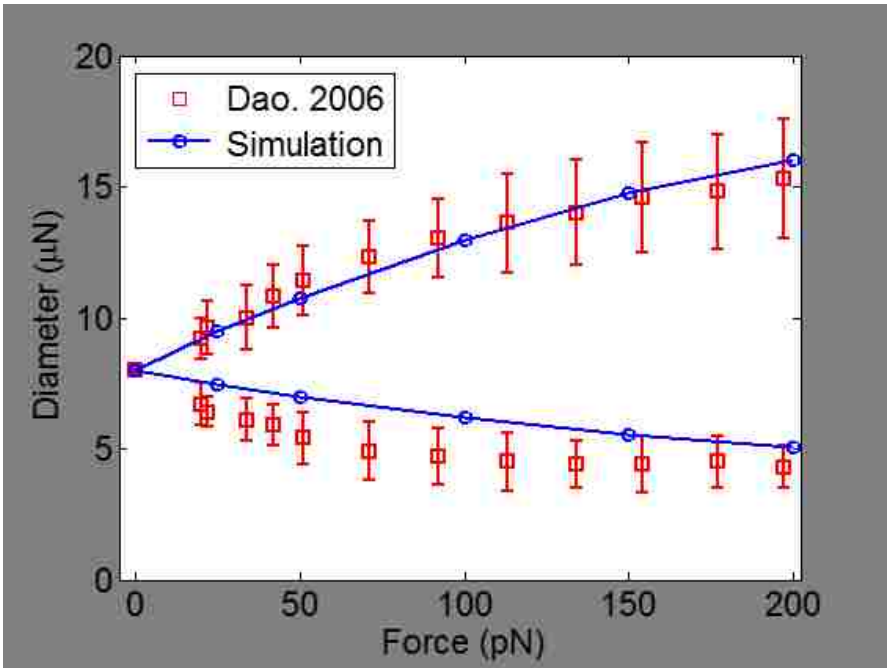


Figure 3.8: The force extension curve during cell stretching test. The data with error bar is obtained from Ref.[92]. It shows that our modeling results agree well with experimental tests.

Chapter 4

Numerical models for nanoparticles

Nanoparticles (NPs) have been extensively studied in recent years as the next generation of drug carriers and imaging probes[106–110]. Liposomes and polymer-drug cojugates based NPs provide the foundations for targeted drug delivery using nanomedicine technology. Several of them, for example, liposome-encapsulated doxorubicin, liposomal amphotericin B, liposomal morphine have been approved by US Food and Drug Administration (FDA)[111]. Properties of NP such as size, shape and surface chemistry play an important role in NP binding and clearance. Generally, particles bigger than 200 nm are efficiently filtered by liver, spleen and bone marrow, while particles smaller than 10 nm can be quickly cleared by the kidney or though extravasation[111–113]. To achieve targeted delivery, NPs are usually coated with ligands that bind specifically to a particular type of receptor on the cell surface [114]. NP targeted delivery in a vascular system involves the interplay of transport, hydrodynamic force, and multivalent interactions with targeted biosurfaces. First, NPs are marginated from the blood stream to the vicinity of the vascular wall. The dispersion of particles in a fluid flowing through a tube can be described by the Taylor-Aris theory [115, 116]. Then, the interaction

of NPs with the vessel wall occurs when their minimum distance is below 20 nm[117, 118]. The ligands on the NP surface bind with receptors on the vessel wall, leading to large adhesive forces. After initial binding, the NP may get firmly adhered or may be washed away, depending on the strength of binding, flow conditions, etc.

4.1 Nanoparticle transport in flow

When transported in fluid flow, NPs are subjected to long range hydrodynamics and Brownian dynamics, similar to polymers and colloids in a solution. Brownian dynamics is a mesocopic method to model polymers through replacing the explicit solvent molecules interactions with stochastic forces. The reasoning behind this technique is that there is a large time scale separation between solvent molecule motion and the slow motion of the immersed polymers. Thus, Brownian dynamics is able to simulate polymer motion using much larger time step compared with molecular dynamics. However, simple Brownian dynamics does not contain long range hydrodynamic effects. It is possible to include hydrodynamics in Brownian dynamics through adding an interaction tensor as part of the diffusion tensor[119]. There are also many methods proposed to simulate nanoparticle, polymers, colloids transport in a solution. For example, Dissipative particle dynamics (DPD) is a coarse grained molecular dynamics that uses soft potentials and can reproduce correct Navier-stokes hydrodynamics[6, 120]. The soft potential enables a larger time step compared with molecular dynamics. To make further simplifications, the stochastic rotation dynamics (SRD) method directly updates particle velocity and positions through a streaming and collision process[5, 72]. It eliminates the force calculation procedure but the collision step is designed in such a way that it conserves mass, momentum and energy.

The interaction between NPs and the fluid can be found in the previous litera-

ture of DPD and SRD. Ref [72, 121] gave excellent discussion on how to separate different time and length scales in colloidal systems. Here we only focus on the nanoparticle modeling. The basic idea of NP modeling can be directly borrowed from molecular dynamics. A set of bonds, angle, and dihedral potential will be used to define the structrual response between different atoms(or coarse grained beads) NPs were modeled as rigid bodies with motion governed by hydrodynamic forces and Brownian dynamics [122, 123]. Langevin dynamics was used to simulate the motion of particles.

$$m\frac{d\boldsymbol{u}}{dt} = -\zeta\boldsymbol{u} + \boldsymbol{F}_c + \boldsymbol{F}_r \tag{4.1}$$

where ζ is the friction coefficient defined as $\zeta = 6\pi\mu r$, $\boldsymbol{F}_c$ is the conservative force, and $\boldsymbol{F}_r$ is the random force that satisfies the fluctuation dissipation theorem.

$$\begin{aligned} < \boldsymbol{F}_r(t) > &= 0 \\ < \boldsymbol{F}_r(t)\boldsymbol{F}_r(t') > &= 2k_BT\zeta\delta(t-t')\boldsymbol{I} \end{aligned} \tag{4.2}$$

where k_BT is the thermal energy, $\delta(t-t')$ is the Dirac delta function, $\boldsymbol{I}$ is the unit-second order tensor. The conservative force $\boldsymbol{F}_c$ could be pair wise interaction force between two particles, angle based potential force between three particles, dihedral bending force between four particles, and area or volume conservation force. Here we only focus on individual NPs that can be treated as a single point. Note that this approach is readily extended to model polymer chain, membrane network or any other particle based model.

The solution of the above equation 4.1 gives

$$\boldsymbol{u}(t) = \frac{\boldsymbol{F}_c + \boldsymbol{F}_r}{\zeta} + Cexp(-\frac{\zeta t}{m}) \tag{4.3}$$

where C is a constant. When the time step dt ($4.2\times10^{-8}\,s$ in the present study) is

much larger than the relaxation time $\tau_R = \frac{m}{\zeta}(2.2 \times 10^{-9}\, s)$, the particle position can be updated with the terminal velocity as $\boldsymbol{u}(t) = \frac{\boldsymbol{F}_c + \boldsymbol{F}_r}{\zeta} + \boldsymbol{u}_f$, where the $\boldsymbol{u}_f$ is the contribution from the fluid. This approach is suitable for the immersed boundary method coupling scheme used in this work. If we are interested in small time scale effect, e.g., $dt < \tau_R = 2.2 \times 10^{-9} s$, then molecular dynamics based approach would be a better choice.

4.2 Nanoparticle adhesion modeling

The interaction of NPs with the vessel wall occurs when their minimum distance is below 20 nm [117, 118]. New Ligand-receptor (LR) bonds will be formed between NP surface and cell membrane. The bonds will generate force to resist flow induced drag on NPs. Whether the NPs are binded or not depends on the interplay between different forces. An illustration of ligand receptor interaction when NPs are contact the substrate is shown in Fig.4.1. An excellent review on Ligand receptor interactions can be found in Ref. [124]. Mathematical models[125, 126] and computer simulations[127–129] have been widely used to study the drug delivery mechanism. Previous models that have been proposed to study NP adhesion kinetics are mainly based on either the equilibrium approach [130, 131] or kinetics approach[132]. Evans et al. [131] have developed a model to study binding dynamics of NPs by considering equilibrium force required to separate adhesive bonds while Haun et al. [132] considered the association and dissociation constants of particles as a function of receptor and ligand density. Ferrari and Decuzzi [133] created a NP adhesion model to demonstrate that NP binding probability decreases with size and shear rate. Complex vascular environments such as erythrocyte and vessel geometries on NP delivery have been shown an important impact on NP distribution[24–26, 134].

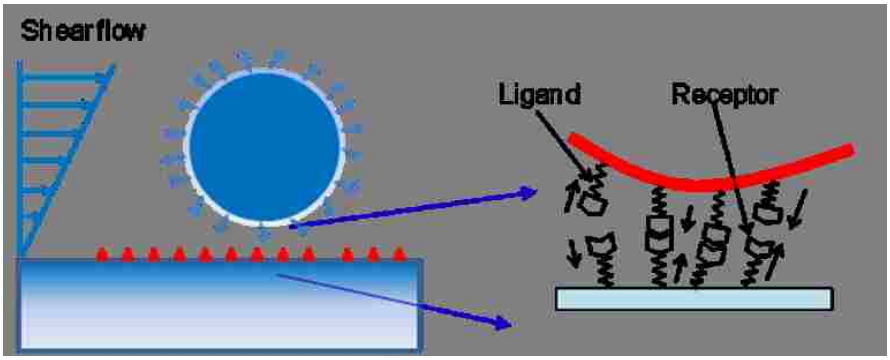


Figure 4.1: An illustration of ligand-receptor binding kinetics between ligand-coated Nanoparticle surface and receptor coated vascular wall surface

On the Ligand receptor scale, an adhesion kinetic equation is used to calculate the bond density N_b [117, 130]

$$\frac{\partial N_b}{\partial t} = k_f(N_l - N_b)(N_r - N_b) - k_r N_b \tag{4.4}$$

where N_l and N_r are the ligand and receptor densities; k_f and k_r are the forward (association) and reverse (dissociation) reaction rates, respectively. This interaction model represents a conservation equation of the different species (ligands, receptors, and bonds). The k_f and k_r are functions of bond length

$$k_f = k_f^0 exp\left(-\frac{\sigma_f(l-l_0)^2}{2k_BT}\right), \qquad k_r = k_r^0 exp\left(-\frac{\sigma_r(l-l_0)^2}{2k_BT}\right) \tag{4.5}$$

where σ_f and σ_r is the elastic constant for LR bond under association and dissociation; k_f^0 and k_r^0 are the the forward and reverse reaction rates at the zero load of ligand-receptor pair. l and l_0 are the bond length and equilibrium bond length. k_B is the Boltzmann constant, T is the temperature. The Eqn.4.4 is a differential equation that used to describe the evolution of bond density. On the other hand, a probability based method of bond formation and dissociation was introduced[102, 135]. The probability of bond formation and dissociation are related to forward

and reverse reaction rate as,

$$P_f = \begin{cases} 1 - e^{-k_f \Delta t}, \text{for } l < d_f \\ 0, \quad \text{for } l \geq d_f \end{cases}, \quad P_r = \begin{cases} 1 - e^{-k_r \Delta t}, \text{for } l < d_r \\ 0, \quad \text{for } l \geq d_r \end{cases} \tag{4.6}$$

where d_f and d_r are the cut off distances for bond formation and dissociation, Δt is the time step used in the simulation. During the simulation, all existing bonds are checked first for a potential dissociation according to probability P_r. A bond is ruptured if $\xi < P_r$ and left unchanged otherwise, where ξ is a random variable uniformly distributed on [0, 1]. The ligand is available for new bonding after breaking. Then, all free ligands are examined for possible bond formations. For each free ligand we loop over the receptors within distance d_f, and bond formation is attempted for each found receptor according to the probability P_f, e.g., A ligand receptor bond is formed if $\xi < P_f$ and remained free otherwise, which is similar to the approach as checking rupture in the first step. On the NP scale, NPs binding is characterized by a concentration c over the binding surface. The material balance on the reactive surface for NP bonding is given by

$$\frac{\partial c_s}{\partial t} = k_a c_w - k_d c_s \tag{4.7}$$

where c_s is the bonded NPs concentration on the surface, c_w is the free NP concentration near the reacting wall, k_a and k_d are the attachment and detachment rate, respectively. It is important to note that the kinetic rates (attachment rate k_a and detachment rate k_d) describing the NP binding in terms of concentration are different from the ligand-receptor reaction rates (forward binding rate k_f and reverse binding rate k_r) that characterize the ligand-receptor interaction at nanoscale. At the ligand-receptor level, Bell gave an analytical formula of k_f and k_r for a diffusion limited ligand-receptor binding process[130, 136]. Both deterministic models and probability based approaches have been proposed to study ligand-receptor bond

formation [117, 137]. To link the ligand receptor bond formation with particle adhesion, Liu et al. [138] proposed a mesoscale modeling method to estimate the binding affinity between a nanocarieer and endothelial cells starting with absolute binding free energies. A continuum-kinetics approach was also used to study cell adhesion and movement where communication between microscale and nanoscale is facilitated numerically through bond force and interface deformation [139]. Fogelson et al. [140] proposed a multiscale model to link ligand-receptor binding with platelet aggregation through a combined elastic inter-platelet link function and immersed boundary method where platelet convection-diffusion equations, and bond formation equations are solved interactively.

4.3 Implementation and benchmark

4.3.1 Cell linked list algorithm

Pairwise potentials are widely used in modeling the interaction between NPs. e.g., the Lennard-Jones (LJ) potential could be used to model the interaction between a pair of atoms, the volume exclusion effect between NPs, and between NPs and the cell membrane. The simplest way to calculate the pairwise interaction force is a so called double-loop algorithm. First, we loop over all the atoms. Then, for each atom, we have to check the distance between the current atom and the rest of the atoms. If the distance is within the cut off distance of the potential, the force term will be calculated. Let us denote the number of atoms as N, the algorithm's complexity is $O(N^2)$. As the number of atoms gets bigger, the algorithm is not efficient. This section will explain the cell linked-list algorithm of which the computational time scales as $O(N)$[141, 142]. Other fast algorithms of short range molecular dynamics algorithms can be found in Ref. [142]. In the cell linked-list algorithm, the whole simulation box is binned into 3D cells of side length of cut off distance (r_c) of the potential. Thus, the distance calculation

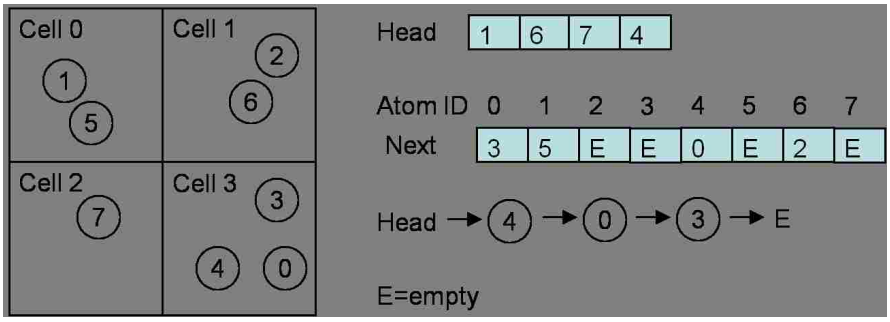


Figure 4.2: A illustration of the cell linked-list algorithm in 2D. The domain is divided into 4 cells. There is a *Head* pointer for each cell that stores the first atom in the cell. Another pointer *Next* is used to store the next atom within the same cell as the current atom with atom ID as the address of pointer *Next*. The linked list for cell 3 is shown in the figure as well.

only has to be performed over the near by 27 cells, the cell that the current atom resides and the surrounding 26 cells. The atoms within a cell is organized using linked list. First, we have a *Head* pointer that points to the first atom within the cell. Then, we have another *Next* pointer with size of the total number of the atoms. The *Next* pointer points to the next atom within the same cell. The *Next* pointer for the last atoms within a cell points to empty. A piece of the code to build the linked list and perform the force calculations is shown in the Appendix A.

4.3.2 Diffusion benchmark case

Due to the countless collision from surrounding fluid molecules, NPs will undergo Brownian motion and move randomly in the quiescent fluid solution. The random motion of NPs can be described as a diffusion process. The theoretical diffusion coefficient for small particles can be given by [143]

$$D = \frac{k_B T}{6\pi\mu r} \tag{4.8}$$

where k_B is the Boltzmann constant, T is the temperature, μ is the fluid viscosity, r is the radius of the NPs, assuming the NPs have a spherical shape. Mean-

while, the simulated diffusion coefficient can be calculated from the mean square displacement $< \xi(t, \tau) >$, given by

$$< \xi(t) >= \frac{1}{N} \sum_{i=1}^{N} \Delta r_i(t)^2 \tag{4.9}$$

where N is the number of particles, $\Delta r_i(t)$ is the distance of the i th particle compared with its initial position, t is the time. The diffusion coefficient can be calculated using

$$D = \frac{1}{2Dim} \frac{d}{dt} < \xi(t) > \tag{4.10}$$

where Dim is the dimension. It could be $\{1, 2, 3\}$, depending how $\Delta r_i(t)$ is calculated. It should be close to the thermal diffusion coefficient calculated by Einstein's formula Eqn. 4.8. In our model settings, the fluid domain was $25\mu m$ by $50\mu m$. The fluid was taken as water with a density of $1000kg/m^3$ and a viscosity of $1 \times 10^{-3} pa.s$. The lattice size δx was $5 \times 10^{-7} m$, time step δt and the relaxation parameter τ were $4.2 \times 10^{-8} s$ and 1.0, respectively. 378 nanoparticles of size of 100 nm were randomly positioned in the fluid domain. The temperature was set at $300K$. The time history of the mean square displacement in one dimension is plotted in Fig.4.3. The diffusion coefficient given by the half of the slop is $4.317 \times 10^{-12} m^2/s$, which is very close to the value given by Einstein's formula $4.39 \times 10^{-12} m^2/s$.

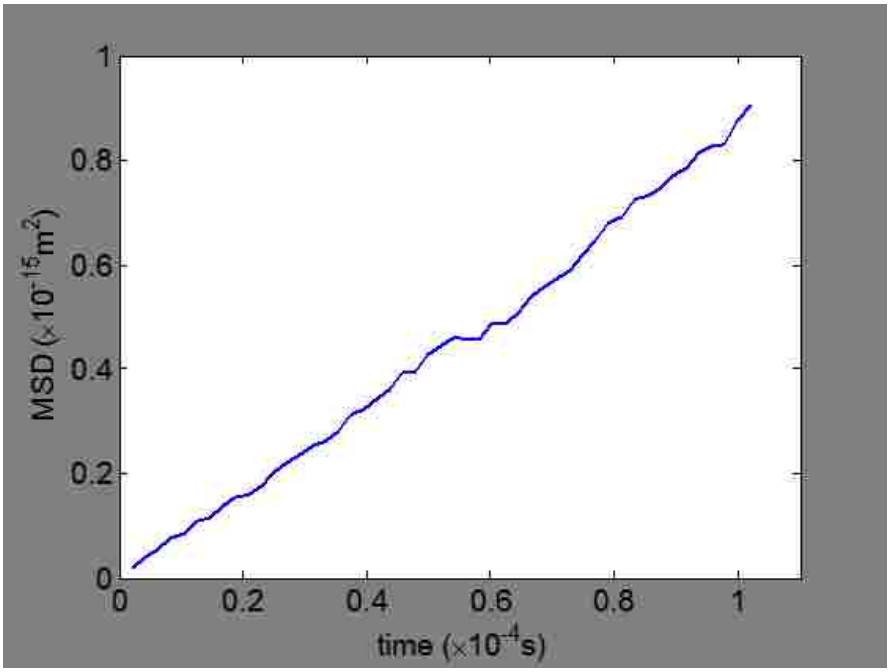


Figure 4.3: Mean square displacement (MSD) time history plot in the simulation

Chapter 5

Fluid structure interactions

As a challenging multiphysics problem, fluid structure interaction (FSI) has been the focus of computational mechanics for many years. FSI problems are generally driven by the interaction of two distinctly different components: fluid and solid. FSI problems is very important in many scientific and engineering applications, such as aircraft design, bridge design, wind turbine, blood flow in heart[144]. However, FSI problems are very challenging due to the nature of nonlinearity and multiphysics. Generally an analytical solution to the coupled system is impossible, while experimental study is limited by the scale of the system. Thus, numerical method has been developed to solve the fluid and structure dynamics simultaneous. The interface boundary conditions are crucial to the coupled FSI problems. Currently there are two general approaches to model FSI problems. The first one is Arbitrary Lagrangian Eulerian method[144, 145]. In this approach, the mesh used for the fluid is neither fixed in space as the Eulerian description of the fluid flow does, nor moved with the fluid as the Lagrangian description does. Instead, the mesh of the fluid can move in any other prescribed way. On the interface between fluid and solid, the velocity and stress should be continuous over the interface. The mesh of fluid conforms to the mesh of solid. Thus, it is straightforward to impose boundary conditions on the fluid solid interface. It is accurate and

efficient in computation. However, as solids are subjected to large translation and rotations, re-meshing is usually used to avoid mesh distortion or entanglement. However, the mesh regeneration is cumbersome during the computation. As opposed the accurately captured fluid solid boundaries in ALE method, Immersed boundary method (IBM) was introduced by Peskin[146, 147] using non boundary fitting method. Specifically, two independent meshes are used for the fluid and solid. The existence of solids is represented by the solid boundary force spread out into the fluid. The IBM approach eliminates the re-meshing procedure in ALE, thus it is very efficient in FSI modeling.

This chapter will mainly discuss the technical details of IBM and its discretization in lattice Boltzmann units. Other coupling techniques such as stress integration approach and friction coupling approach will be introduced. A few benchmark cases have been created to validate the coupling scheme.

5.1 The immersed boundary method

The Immersed Boundary method was selected to model the interaction between the fluid and the immersed solids due to the algorithm's efficiency. The IBM was first proposed by Peskin to study blood flow in the heart[146, 148]. The fluid is solved on a spatially fixed Eulerian grid, while the immersed solids are modeled using a moving Lagrangian mesh, which is not constrained to the geometric layout of the Eulerian fluid grid. Data is exchanged between the two domains through nodal interpolation. The coupling scheme enforces velocity continuity at the fluid-structure boundary, and transfers forces from the structure back into the fluid through an effective force density. This two-way coupling automatically handles immersed body contact and prevents solid penetration through the development of restoring forces in the fluid. The approach has been used for a variety of fluid-structure interaction problems, including the simulation of jelly fish[149],

blood flow[24, 94, 150, 151], platelet migration[152]. Comprehensive reviews of the IBM and its applications can be found in [104, 147]. The immersed structure can be viewed as a parametric surface $X(p, q, r, t)$, where p, q, r are curvilinear coordinates for the structure in Lagrangian description, t is the time. The fluid domain are described by Eulerian coordinates $\boldsymbol{x}$. The the force $\boldsymbol{f}(\boldsymbol{x}, t)$ exerted by the structure on the fluid is interpolated as a source term in the momentum equation using

$$\boldsymbol{f}(\boldsymbol{x}, t) = \int \boldsymbol{F}(p, q, r, t)\delta(\boldsymbol{x} - \boldsymbol{X}(p, q, r, t))dpdqdr \tag{5.1}$$

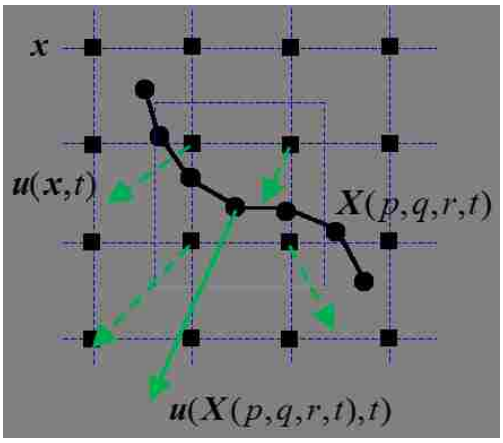
where $\boldsymbol{F}(p, q, r, t)$ is the force density for the structure. Typically it is derived from energy density functions. $\delta(\boldsymbol{x})$ is the three dimensional delta functions $\delta(x_1)\delta(x_2)\delta(x_3)$ where x_1, x_2, x_3 are the Cartesian components of position vector $\boldsymbol{x}$.

Similarly, the structure moving velocity is updated based on the local fluid velocity through interpolation using

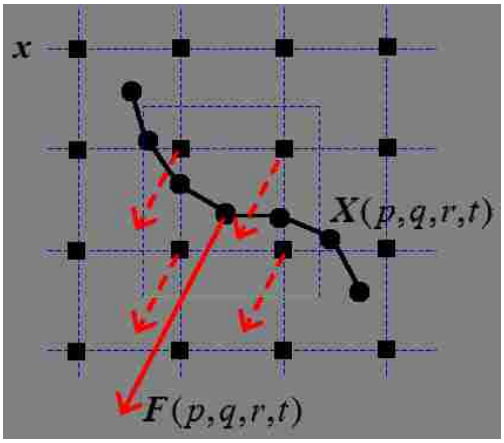
$$\boldsymbol{u}(\boldsymbol{X}(p, q, r, t), t) = \int \boldsymbol{u}(\boldsymbol{x}, t)\delta(\boldsymbol{x} - \boldsymbol{X}(p, q, r, t))d\boldsymbol{x} \tag{5.2}$$

where $\boldsymbol{u}(\boldsymbol{X}(p, q, r, t), t)$ is the structural moving velocity, $\boldsymbol{u}(\boldsymbol{x}, t)$ is the fluid velocity over the fluid domain $\boldsymbol{x}$ at time t. $\delta(\boldsymbol{x})$ is the same function as used in Eqn 5.1. Eqn 5.2 essentially is the velocity continuity conditions on the fluid structure interface.

The concept of IBM coupling scheme is also illustrated in Fig. 5.1. The square dots represent fluid nodes, while circle dots represent solid structure. The solid velocity is interpreted from local fluid velocity. The contribution of each neighbor fluid nodes and how many nodes should be selected is determined by $\delta(\boldsymbol{x})$. For example, Fig.5.1a shows that the velocity of the structural central node is interpreted from nearby four nodes within the dashed square frame. Similarly,



(a) solid velocity is interpreted from local fluid velocity



(b) Structural force is spread into local fluid as force density

Figure 5.1: The two way coupling between the fluid and the structure in the Immersed boundary method. (a) The structural velocity $\boldsymbol{u}(\boldsymbol{X}, t)$ is interpolated from neighbor fluid nodes within the dashed square box. The contribution from fluid node is weighted by the $\delta(\boldsymbol{x})$ function. (b) The structural force will be spread out into local fluid nodes as a force density. The distribution of the force among fluid nodes are determined by the $\delta(\boldsymbol{x})$ function as well.

the structural force would be spread to the four neighbor fluid nodes, as shown in Fig.5.1b.

5.2 Spatial and temporal discretization

Once the coupling scheme has been established, the spatial and temporal discretization of the IBM equations, Eqn. (5.1 and 5.2), is needed in the numerical implementation of IBM. Following Peskin's work[147], we will discuss the spatial discretization first, the choice of $\delta(\boldsymbol{x})$ will be presented next, and the temporal integration scheme will be given last. The spatial discretization of Eulerian grid, denoted g_x is a set of points with equal spacing in Cartesian coordinates, e.g., $\boldsymbol{x} = (x_j, y_j, z_j)\Delta x$, where Δx is the spacing, (x_j, y_j, z_j) are the position components in each direction. This spatial discretization is also consistent with Lattice Boltzmann fluid solver. Similarly, the spatial discretization of the Lagrangian grid, denoted G_s, is the set of (p, q, r) of the form $(p_k\delta p, q_k\delta q, r_k\delta r)$, where (p_k, q_k, r_k)

are integers. Ref. [147] also suggested that the $\delta s < \frac{\Delta x}{2}, s \in \{p, q, r\}$ should be required to avoid fluid leaking. Ref. [64, p.57] shows that the mesh ratio between solid and fluid for small deformations, e.g., $\frac{\delta s}{\Delta x}$, can be safely chosen within range (0.5, 1.5) without significantly influencing the physical results. However, the membrane resolution must be sufficiently high to handle regions with large local curvature. For example, during our test, leaking is observed when the $\frac{\delta s}{\Delta x} = 3.03$ for the cell squeezing through a narrow channel. Here the δs is calculated as the average length of the cell bond during our cell simulation model. The force spreading equation 5.1 becomes

$$\boldsymbol{f}(\boldsymbol{x}, t) = \sum_{(p,q,r)\in G_s} \boldsymbol{F}(p, q, r, t)\delta_{\Delta}(\boldsymbol{x} - \boldsymbol{X}(p, q, r, t))\Delta p\Delta q\Delta r \tag{5.3}$$

Notice that $\boldsymbol{F}(p, q, r, t)$ is the force density among solid structures. Let us define $\mathcal{F} = \boldsymbol{F}(p, q, r, t)\Delta p\Delta q\Delta r$. $\mathcal{F}$ can be viewed as the integration of force density $\boldsymbol{F}$ over the an element volume $dv = \Delta p\Delta q\Delta r$, which is the force term applied to each node. The Lagrangian nodes of solid can be indexed as i without losing any generality. Eqn.5.3 can be reduced as

$$\boldsymbol{f}(\boldsymbol{x}, t) = \sum_{i\in G_s} \mathcal{F}_i\delta_{\Delta}(\boldsymbol{x} - \boldsymbol{X}_i) \tag{5.4}$$

The velocity interpolation formula can be discretized as

$$\boldsymbol{u}(\boldsymbol{X}_i, t) = \sum_{x\in g_x} \boldsymbol{u}(\boldsymbol{x}, t)\delta_{\Delta}(\boldsymbol{x} - \boldsymbol{X}_i)\Delta x^3 \tag{5.5}$$

In lattice Boltzmann method, spatial step and time step in LB units are usually assumed to be unity. e.g., $\Delta x = 1, \Delta t = 1$. Thus, Eqn. 5.5 can be reduced as

$$\boldsymbol{u}(\boldsymbol{X}_i, t) = \sum_{x\in g_x} \boldsymbol{u}(\boldsymbol{x}, t)\delta_{\Delta}(\boldsymbol{x} - \boldsymbol{X}_i) \tag{5.6}$$

Ref. [147] showed that the discretized $\delta_\Delta(\boldsymbol{x})$ function has to meet some restrictions and properties to make sure that mass, force, and torque are the same no matte they are evaluated from Eulerian or Lagrangian variables. One of the assumptions made here is that $\delta_\Delta(\boldsymbol{x})$ can be given by the triple product of a scalar functions $\phi(x)$.

$$\delta_\Delta(\boldsymbol{x}) = \phi(x)\phi(y)\phi(z) \tag{5.7}$$

where (x, y, z) are the three components of position vector $\boldsymbol{x}$. Here we don't go details on analyze the $\phi(x)$ function. Instead, we just want to point out the most widely used four point interpolation function.

$$\phi(x) = \begin{cases} 0, & |x| \geq 2 \\ \frac{1}{8}\left(5 - 2|x| - \sqrt{-7 + 12|x| - 4x^2}\right), & 1 \leq |x| \leq 2 \\ \frac{1}{8}\left(3 - 2|x| + \sqrt{1 + 4|x| - 4x^2}\right), & 0 \leq |x| \leq 1 \end{cases} \tag{5.8}$$

Ref. [147] pointed out that $\phi(x)$ can also be extremely well approximated by a simple formula

$$\phi(x) = \begin{cases} 0, & \text{otherwise} \\ \frac{1}{4}\left(1 + cos\left(\frac{\pi x}{2}\right)\right), & |x| \leq 2 \end{cases} \tag{5.9}$$

However, we suggest using Eqn.5.8 because it is much faster in evaluating polynomial function than $cos(x)$ function. Since the fluid and solid are solved alternatively, this IBM coupling is a partitioned approach. To obtain a second order accurate scheme in temporal discretization, Peskin came up with a time integration scheme based on midpoint rule[147]. For the easiness of discussion, the solution at time step n will be denoted as a superscript over the variable. Assume currently we have the structural position X_i^nat time step n, we need to calculate

the intermediate position at time step $n+\frac{1}{2}$ using

$$\boldsymbol{X}_i^{n+\frac{1}{2}} = \boldsymbol{X}_i^n + \frac{\Delta t}{2}\sum_{\boldsymbol{x}\in g_x} \boldsymbol{u}^n \delta_\Delta(\boldsymbol{x}-\boldsymbol{X}_i)\Delta x^3 \tag{5.10}$$

With the newly obtained solid position at $n+\frac{1}{2}$ time step, the structural force can be evaluated as

$$\mathcal{F}_i^{n+\frac{1}{2}} = \boldsymbol{F}_i^{n+\frac{1}{2}}\Delta p\Delta q\Delta r = StructureSolver(\boldsymbol{X}_i^{n+\frac{1}{2}}) \tag{5.11}$$

where *StructureSolver* is the procedure to calculate all the structure force applied to each structural nodes. Next, the structure force $\mathcal{F}_i^{n+\frac{1}{2}}$ will be spread out into fluid through

$$\boldsymbol{f}^{n+\frac{1}{2}}(\boldsymbol{x}) = \sum_{\boldsymbol{X}\in G_s} \mathcal{F}_i^{n+\frac{1}{2}}\delta_\Delta(\boldsymbol{x}-\boldsymbol{X}_i) \tag{5.12}$$

With the force density, the fluid solver will update the fluid velocity $\boldsymbol{u}^{n+1}$ through

$$\boldsymbol{u}^{n+1} = FluidSolver(\boldsymbol{f}^{n+\frac{1}{2}}) \tag{5.13}$$

The *FluidSolver* could be any fluid solver. The Lattice Boltzmann fluid solver is used in this dissertation. Finally, the solid position at time $n+1$ is updated as

$$\boldsymbol{X}_i^{n+1} = \boldsymbol{X}_i^n + \Delta t\sum_{\boldsymbol{x}\in g_x} \boldsymbol{u}^{n+1} \delta_\Delta(\boldsymbol{x}-\boldsymbol{X}_i)\Delta x^3 \tag{5.14}$$

In this work, Eqn 5.10 and 5.14 are used with the Lattice Boltzmann lattice space $\Delta x = 1$ and time step $\Delta t = 1$.

5.3 Benchmark case

Sphere settling in a viscous fluid is widely used as a benchmark for fluid structure interaction (FSI) simulation[153]. The process involves placing a solid sphere in a

static fluid and allowing it to accelerate downward under gravity loading until it reaches a steady velocity, where the resultant drag force balances the gravity load. In our FSI benchmark, the sphere is modeled as a rigid 2D ring structure. The motion of the ring is interpolated from the local fluid velocity. A rigid boundary, however, is very difficult to achieve in the immersed boundary method. Therefore, the modeling approach developed by Fogelson[154] and Feng[155] was adopted. The model produces an effectively rigid particle surface using stiff elastic fibers. This is accomplished by using virtual images of the surface nodes undergoing rigid motion. A restorative force is applied to the nodes that deviate from the position of the virtual image. Additional details for the approach can be found in the referenced studies.

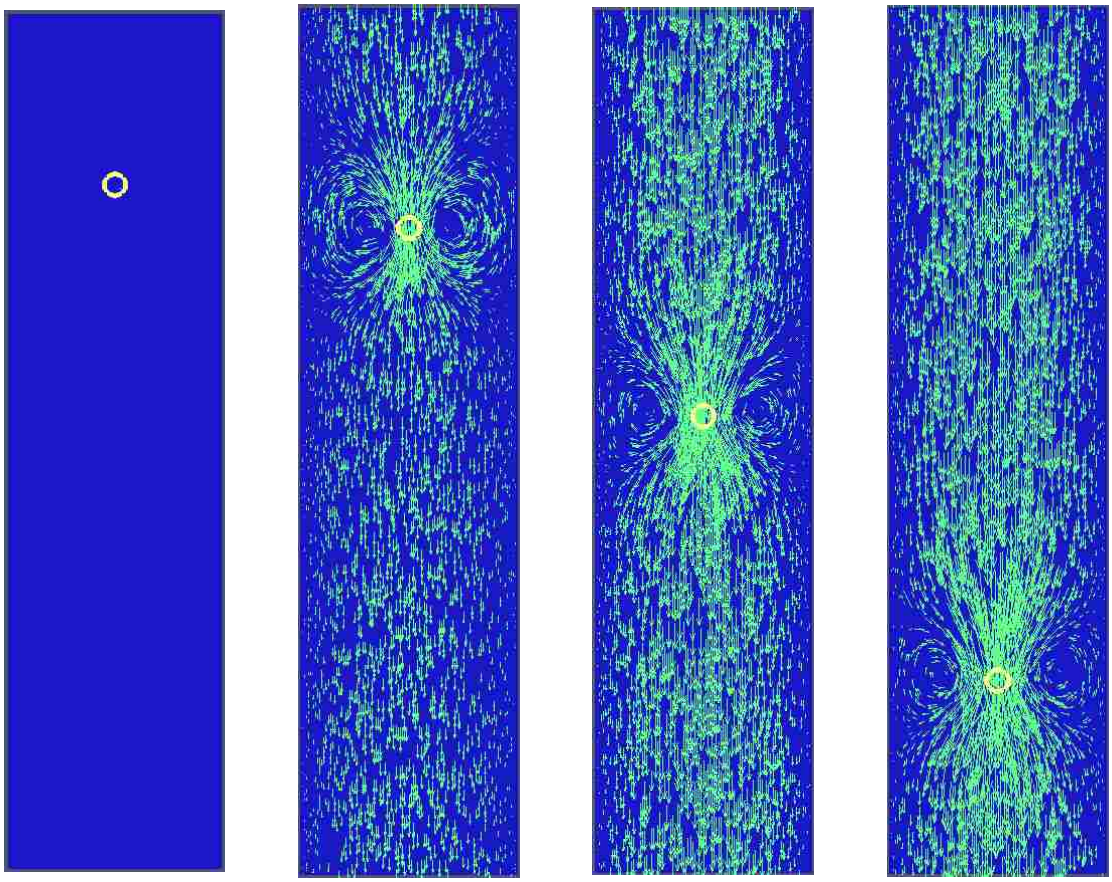
The fluid channel for the benchmark simulation was 4 cm in length and 1 cm in width. The sphere had a diameter of 0.1 cm, and was placed in the fluid at approximately 0.8 cm away from the top, along the channel center line. The density of the sphere was 1001 kg/m^3. The restorative stiffness used for the effective rigid boundary was 1×10^{-4} N/m. The fluid was taken as water with a density of 1000 kg/m^3 and a viscosity of 1×10^{-3} $pa.s$. The lattice size dx was 1×10^{-4} m, and the time step dt is 1.667×10^{-3} s and lattice Boltzmann relaxation parameter τ is 1.0, respectively. The whole set up of the system is shown in Fig. 5.2a and snap shots of the simulation are shown Fig. 5.2. Following reference[156], the theoretical terminal velocity of a cylinder (2D) is

$$V_s = \sqrt{\frac{\pi g D(\rho_s - \rho_f)}{2 C_d \nu \rho_f}} \tag{5.15}$$

where the drag coefficient

$$C_D = \frac{8\pi}{R_e log(7.4/R_e)} \tag{5.16}$$

where D is the diameter, ν is viscosity, R_e is the Reynolds number, ρ is the density,



(a) (b) (c) (d)

Figure 5.2: Snapshots of a dropping cylinder in a quiescent fluid in 2D. The cylinder is driven by a constant gravity. (a) t=0 s, (b) t=6.67 s, (c) t=26.67 s, (d) t=53.33 s. The yellow ring represents a 2D cylinder.

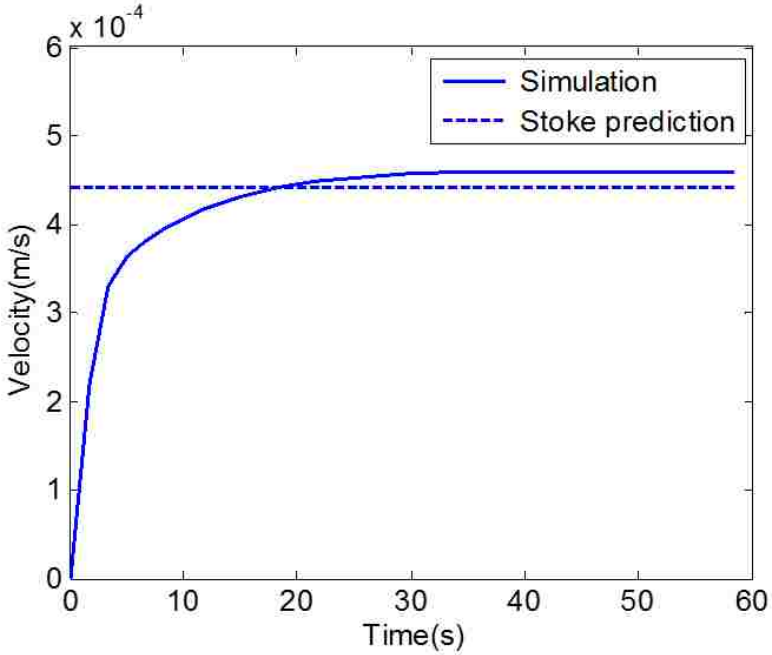


Figure 5.3: Particle settling velocity in simulation compared with theoretical terminal velocity

and s and f stand for solid and fluid, respectively. In order to convert from a node connected ring to a disk, an associated nodal tributary area had to be defined so that the distributed gravitational force could be treated as an equivalent nodal force system. For this study, the associated tributary area was approximated as $dA = \left(\frac{\pi D}{n_s}\right)$, where n_s is the total number of nodes. The predicted terminal velocity from the simulation was $4.6 \times 10^{-4}\ m/s$, which compares very well with the theoretical stokes formula prediction of $4.42 \times 10^{-4}\ m/s$ based on Eqn (5.15), as shown in Fig. 5.3. The difference between the theoretical and simulation based terminal velocity prediction is within 3.8%, which indicates that the FSI code correctly reproduces the kinematics of the sphere in a viscous fluid.

It is well known that red blood cells and droplets will undergo tumbling or tank treading motion under shear, depending on the shear rate and membrane stiffness. Capillary number is typically used to characterize the cell motion. It is defined as $C_a = \nu\eta r/k_s$, where ν is the reference viscosity, η is the shear rate, r is the cell radius, and k_s is the stretching resistance. We performed a 2D simulation

of shear flow with cells. The cell parameters for the simulations are shown in the Table 6.1. The fluid domain was $20\mu m$ by $20\mu m$. The fluid was taken as water with a density of $1000kg/m^3$ and a viscosity of $1 \times 10^{-3}pa.s$. The lattice size dx was $5 \times 10^{-7}m$, and the time step δt and the relaxation parameter τ were $4.2 \times 10^{-8}s$ and 1.0, respectively. The typical cell diameter was $8\mu m$ modeled with 52 nodes.

5.4 Other coupling schemes

5.4.1 Stress integration approach

The fluid structure interaction involving flow separation is beyond the scope of this work. Interested readers may refer to [157–159] for flow separation modelings. Here we only discuss the nonslip boundary conditions between fluid and structure. The nonslip boundary basically is to impose velocity continuity and traction continuity on the interface

$$\begin{aligned} u_f &= u_s \quad \text{on } \Gamma \\ \sigma_{ij}^f n_j &= \sigma_{ij}^s n_j \quad \text{on } \Gamma \end{aligned} \tag{5.17}$$

where u_f and u_s are the fluid and solid velocity, Γ is the interface between the solid and the fluid. σ_{ij} is the stress tensor with superscript f, s representing the fluid and solid, respectively. n_j is the surface norm. In IBM, the solid velocity is interporlated from the fluid, as shown 5.2, while the solid the force is spread into fluid, as indicated by 5.1. The reverse approach can also be used[160, 161]. i.e., first, we apply the fluid stress to the structure, and then solve the structural response, and finally impose the structural velocity as a boundary condition to the fluid. This method is called stress integration approach[161–163]. The total

force acting on the structure is

$$T_i = \int \sigma_{ij} n_j dA \tag{5.18}$$

where dA is the differential area over the interface. The stress tensor σ_{ij} is given by

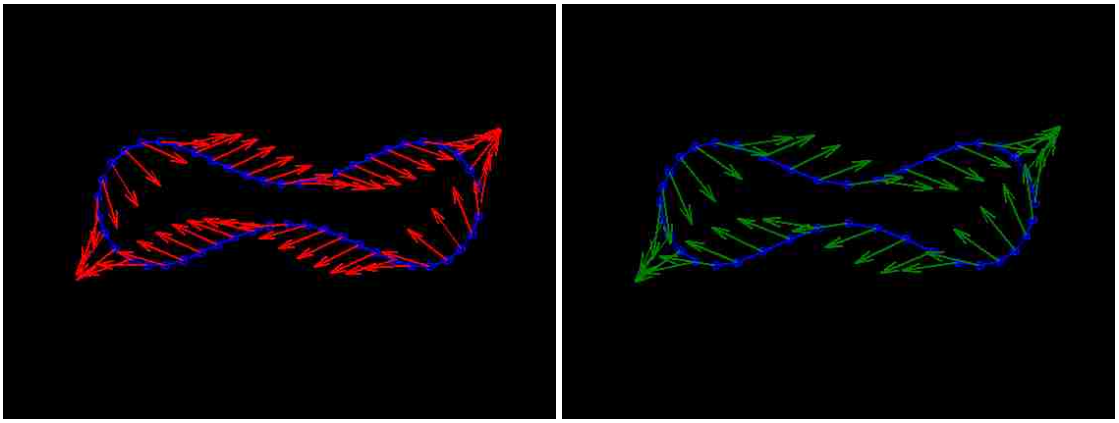
$$\sigma_{ij} = -p\delta_{ij} + \rho\nu \left(u_{i,j} + u_{j,i}\right) \tag{5.19}$$

The pressure term p can be evaluated using Eqn. 2.25. Following Ref. [70], the deviatoric shear stress in LBM $\tau_{ij} := \rho\nu \left(u_{i,j} + u_{j,i}\right)$ can be evaluated as

$$\tau_{ij} = -\left(1 - \frac{\omega}{2}\right) \sum_{\alpha} \left(\xi_{\alpha i}\xi_{\alpha j} - \frac{\delta_{ij}}{D}\boldsymbol{\xi_\alpha} \cdot \boldsymbol{\xi_\alpha}\right) f_\alpha^{neq} \tag{5.20}$$

where $\omega = \dfrac{1}{\tau}$, τ is the relaxation parameter for LBM. α is the index of all the discretized velocity vector, f_α^{neq} is the non-equilibrium part of the density distribution defined as $f_\alpha^{neq} = f_\alpha - f_\alpha^0$ with f_α^0 calculated from Eqn.2.28. $\boldsymbol{\xi_\alpha}$ is the discretized velocity vector as shown in Table 2.1 and 2.2 for 2D and 3D lattices. With the stress calculated from LBM using Eqn. 5.19 and 5.20, the traction applied to the structure can be evaluated by multiplying the surface norm and the stress using the second equation in 5.17. For example, the traction induced by the shear flow on a 2D cell membrane is shown in Fig.5.4a, while the traction evaluated using stress tensor $\tau = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ which is the stress tensor for linear shear flow $u_x = y, u_y = 0$. Compare Fig. 5.4a and 5.4b we can see that Eqn. 5.20 can correctly evaluate the flow induced stress.

With the traction and pressure applied to the structure, the structural analysis is performed to determine the dynamic response. The structural velocity will be treated as a boundary condition for the LBM. There is no standard way to convert boundary velocity to density distribution in LBM. Momentum exchange method



(a) (b)

Figure 5.4: The cell traction induced by the shear flow. (a) the stress is evaluated using Eqn. 5.20. (b) the stress is assumed as [0 1;1 0]

proposed Ladd[54, 55] and a modified bounce back scheme in [160, 161, 164] could be used. We used the approach presented in [160] for the benchmark case of flow induced beam bending as shown in the next section.

5.4.2 Flow induced bending of a beam

In order to validate the code for large-displacement FSI using stress integration approach, a numerical model was developed for the microchannel flow-induced bending simulation, with geometries and physical properties shown in Fig.5.5. Steady state lateral displacement of the beam tip was compared with predictions from an equivalent arbitrary Lagranian-Eularian (ALE) FEM simulation developed in the commercial software COMSOL[165]. It is noted that this simulation is included in the COMSOL documentation as a benchmark case, including recommendations regarding spatial and temporal discretization, and model settings. In the LB-XFEM simulation, the $100\mu m \times 300\mu m$ fluid domain was discretized into a regular lattice with a spatial resolution of $1\mu m$. For computational efficiency, the single relaxation parameter BGK algorithm was utilized with $\tau = 1$. The corresponding time step $\delta t = 0.16\mu s$. The structure is solved through FEM with 2D linear isoparametric quadrilateral elements. The structural mesh has 250

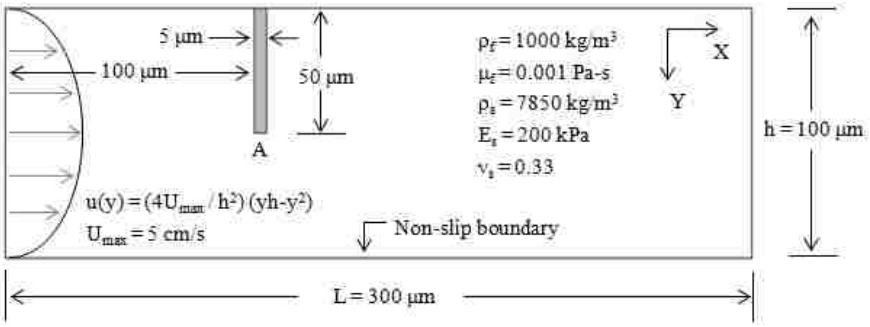


Figure 5.5: Microchannel flow-induced bending simulation for validation of large-displacement FSI response

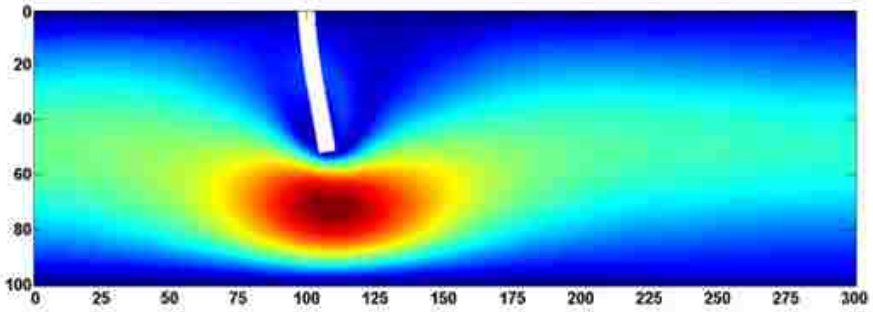


Figure 5.6: Microchannel flow-induced bending simulation: steady state velocity field and beam deformation

elements. The structural model included Rayleigh viscous damping with 5% of critical damping in the first two modes of response in order to speed up the rate of convergence on the steady state displacement. The snapshot of the bending of the beam in the steady state is shown in Fig.5.6, which is exactly similar to the deformation from COMSOL. The tip displacement from both simulations are shown in Fig.5.7. The initial fluctuation is due to the dynamic nature of LBM. The fluctuation is damped out after 2 ms. The steady displacement agree well with the one calculated from the commercial software COMSOL.

5.4.3 Friction based coupling methods

For soft matter system, the flow is usually isothermal, incompressible and with low *Re* number. Thus, it is natural to use stokes friction force to coupling the

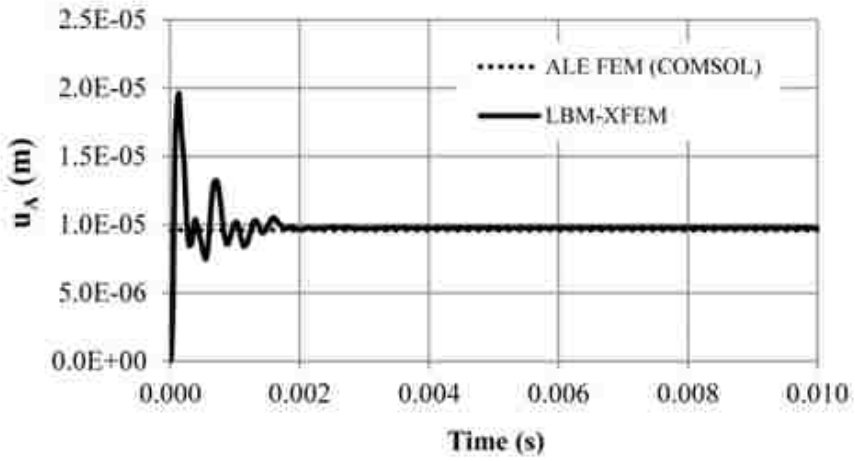


Figure 5.7: Microchannel flow-induced bending simulation: beam tip displacement predictions using ALE FEM and LBM-XFEM

immersed solids and fluids[166–169]. In Ref.[167], the polymer monomers are treated as point with friction force

$$F = -\zeta(u_s - u_f) \tag{5.21}$$

where ζ is the friction coefficient, which may be different from Einstein relation for diffusion. u_s, u_f are solid point velocity and local fluid velocity. The local fluid velocity u_f could be interpolated from nearby fluid nodes. The friction force F will be applied to the solid, together with other conservative forces, random forces, etc. to determine the motion of the particles. Meanwhile, the same magnitude of the force but with reversed direction $-F$ would be applied to the local fluid nodes. For example, the force density $-F/\delta x^3$ will be spread to local 8 fluid nodes, as shown in Ref. [166, 167].

Noted that the nonslip boundary at the solid-fluid interface is not exactly satisfied using the friction based coupling. As pointed out in [167], the microscopic details of the coupling should not play a role as long as the hydrodynamics evolves in fluid on the time scale faster than the diffusion scale of the solid. It correctly reproduced the hydrodynamics in the far field. Thus, the friction coupling ap-

proach sacrifices the resolution of the fluid field near the solid boundary but with the advantage of reducing the number of grid size in LB solver[169].

Chapter 6

Nanoparticle delivery in blood flow

Accurately predicting drug delivery is a critical task in drug development research and clinic trials[170, 171]. It requires careful consideration of physiological conditions such as hematocrit leve[24, 25], vessel geometry and flow conditions[172–174], drug carrier size and shape[18, 25], dissolution rate[175], and external stimuli[176, 177]. For small particles in red blood cell (RBC) suspensions, such as nanoparticles (NP) and platelets, recent studies have demonstrated that local flow field disturbances caused by RBC translation and deformation can enhance particle dispersion[25, 152, 178–180]. The migration of particles in RBC suspensions under shear has been shown to behave like a random walk process[181, 182], with a dispersion rate much larger than thermal diffusion. Therefore, accurate predictions of NP dispersion in RBC suspensions must consider fluid-structure interaction between the immersed solid bodies (particles and cells) and the surrounding fluid. Previously developed models for predicting NP dispersion in RBC suspensions have relied primarily on empirical data fitting. Aarts et. al. experimentally studied shear induced platelet diffusivity (D) which was fitted with shear rate (η) as a power law $D = k\eta^n$, where k is a constant and n is a func-

tion of hematocrit[183]. However, the model parameters are obtained empirically rather than predicted from the underlying physics. Decuzzi et. al. extended the Taylor-Aris theory to calculate an effective NP diffusion rate that considers wall permeability and blood rheology, idealizing the RBCs as Casson fluid with a blunted velocity profile at the vessel core[184, 185]. However, the analysis was relatively complex and the influence of RBCs was oversimplified. In order to address the deficiencies in previously developed models for predicting NP dispersion, this paper presents a numerical study on NP dispersion in RBC suspensions that considers the effects of local flow field disturbances due to RBC motion. This study provides insight into the underlying physics driving NP dispersion in these systems, and develops simple, yet effective, formulae for predicting dispersion rate as a function of characteristic physiological parameters. These simple predictive formulae will provide an efficient approach for assessing NP dispersion under different flow conditions and hematocrit level, thereby facilitating practical modeling of NP transport and distribution in large scale vascular systems[21].

Research has shown that particles in the core region of the vessel migrate toward the cell free layer regions, where the migration process can be modeled as diffusion [181, 182]. This migration is influenced by physical conditions, such as hematocrit level (H_t), cell membrane stiffness (k_s), particle size (r), shear rate (η), fluid viscosity (υ), and cell size (d_c). In this study, two parameters (hematocrit level and shear rate) are considered, while the other parameters are kept constant. NP dispersion is first studied under pure shear flow conditions at different shear rates for a given hematocrit level. Then, the study is extended to investigate NP dispersion in channel flow at different hematocrit levels.

In the study, RBCs were assumed to be healthy with typical physic parameters and with a size of 8 μm. NPs were assumed to be spherical with a typical size of 100 nm. For simplicity, surface charges were neglected so that NPs did not adhere to other NPs or to the RBCs. It is noted that the NP concentration was kept

Parameters	Specified Value	Recommended Range	Reference
Strecthing coefficient k_{s0}	$5\mu N/m$	$5 \sim 12\mu N/m$	[92, 96]
Bending coefficient k_b	$8 \times 10^{-19} J$	$2 \times 10^{-19} \sim 1 \times 10^{-17} J$	[92, 96]

Table 6.1: Cell membrane model parameters

relatively low in order to more readily ascertain the effect of RBC motion on NP dispersion. Dynamic viscosity of the fluid was fixed at $1 \times 10^{-3} Pa \cdot s$. Through dimensional analysis, an empirical function between diffusion coefficient, shear rate, cell size, and hematocrit was defined as: $\frac{D}{d_c^2 \eta} = f(H_t)$, where D is the dispersion rate. This formula was validated through simulations presented in later sections. The test case consisted of a rectangular fluid domain with a 50 μm length, a 25 μm width, and a lattice grid size of 0.5 μm, as shown in 6.1.

In the shear flow case, the top and bottom surfaces were defined as velocity boundaries, while the left and right edges of the domain were modeled as periodic boundaries. In the channel flow case, a parabolic velocity profile was applied at the left inlet boundary and the right outlet boundary was modeled as an open condition. Non-slip boundaries were defined along the upper and lower surfaces. A time step of $4.2 \times 10^{-8} s$ and a relaxation time τ of 1.0 were used for all simulations.

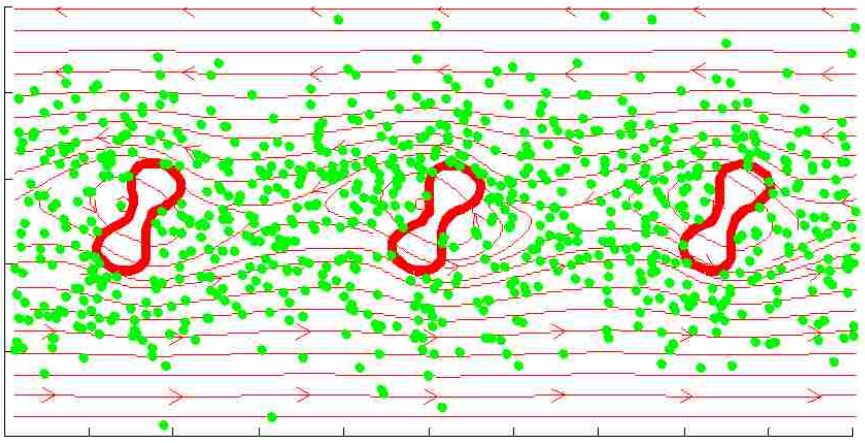
RBC membranes were modeled as bi-concave curves with the dimensions shown in Fig.3.1. A single RBC was composed of 52 nodes. The cell parameters were selected based on recommended values reported in the literature[92, 96], as listed in Table 6.1. The artificial area constraint $k_a = 1$ was selected so that the area change was within 1%. Periodic boundary conditions (PBC) were applied to the left and right boundaries of the fluid domain for both RBCs and NPs. The area ratio between RBCs and the fluid domain was defined as the hematocrit level. The simulation results were collected after the system reached equilibrium, i.e., when RBCs reached steady tumbling or tank treading motion, or when they were relatively uniformly distributed along the channel.

6.1 NP dispersion under pure shear flow

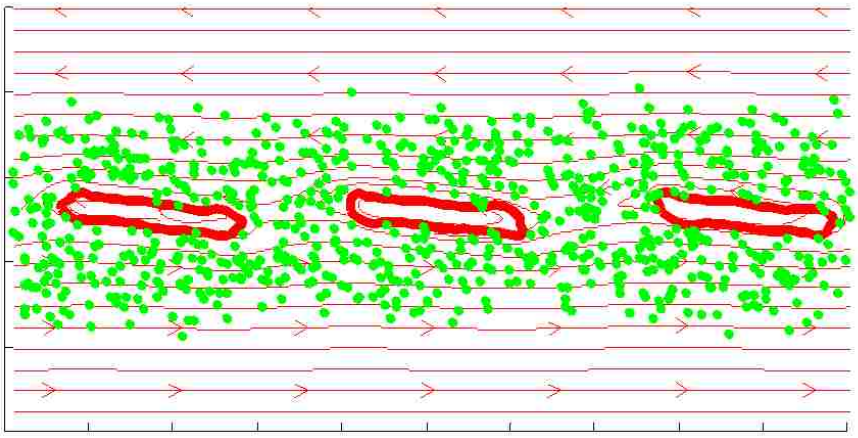
NP dispersion rate was studied over a range of shear rates for a single layer of 3 cells. This set up was designed to eliminate the cell-cell interaction between different layers so that we can focus on the shear rate effect on NP dispersions. Shear rates ranging from 0 to 500 s^{-1} were selected in order to cover both the RBC tumbling and RBC tank treading regions of the flow regime. For all shear rates investigated in the study, the dimensionless number was held at 25. So the simulation time is longer for lower shear rate case. Three RBCs and 792 NPs were considered for each simulation. Snapshots of the interaction between NPs and RBCs at shear rates of 40 s^{-1} and 200 s^{-1} , representative of RBC tumbling and RBC tank treading regions of the flow regime, respectively, are shown in Fig. 6.1.

The mean square displacement over the y direction at different shear rates was calculated to obtain the NP dispersion rates, as shown in Fig. 6.2. It shows that the dispersion rate is strongly influenced by cell motion. In the RBC tumbling ($\eta < 40s^{-1}$) and RBC tank treading ($\eta > 200s^{-1}$) regions of the flow regime, NP dispersion rate is approximately linear with shear rate. Between $40s^{-1}$ and $200s^{-1}$, there is a region where RBC motion transits from tumbling to tank treading motion. In this transition region, there is a drop in NP dispersion with increased shear rate. For the range of shear rates investigated in the study, the dispersion rate initially increases in the tumbling region, then decreases in the transition region, and increases again with the shear rate in the tank treading region. A linear regression model was used to fit both the tumbling(first 3 data points at low shear rate) and tank treading data(last 3 data points at high shear rate)

$$D = \begin{cases} 7.8 \times 10^{-14}\eta + 4.7 \times 10^{-12} & \text{tumbling} \\ 8.5 \times 10^{-15}\eta + 4.0 \times 10^{-12} & \text{tank treading} \end{cases} \tag{6.1}$$



(a) Shear rate of 40 s^{-1}



(b) Shear rate of 200 s^{-1}

Figure 6.1: Interaction between NP and RBC at different shear rates. The bold red lines outline the RBC membranes, while the green markers denote NPs. Flow streamlines are shown in the background. The channel size is 25 μm by 50 μm.

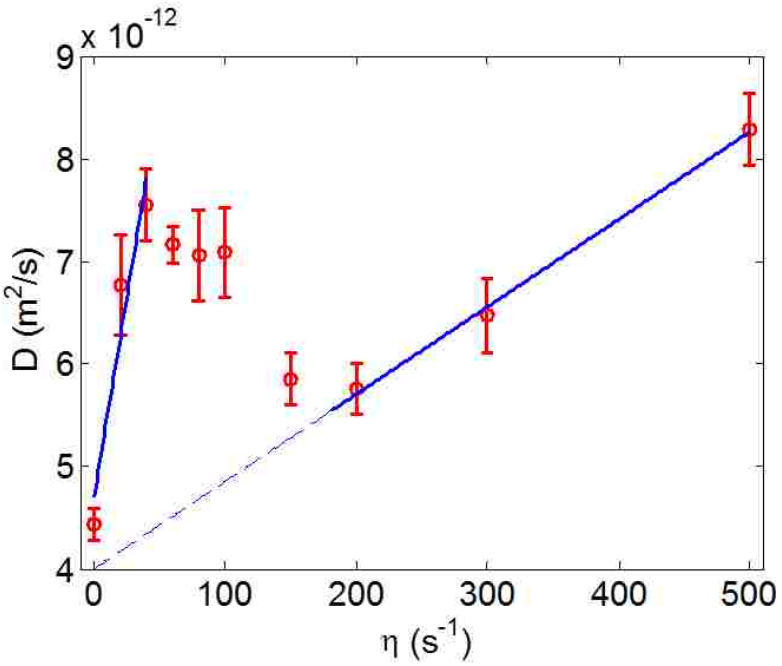


Figure 6.2: NP dispersion rate as a function of shear rate. Error bars indicate the standard variance for three simulations. RBCs undergo tumbling motion at low shear rate($\eta < 40s^{-1}$) and tank treading motion at high shear rate($\eta > 200s^{-1}$). In between, there is a transition region. Linear regression lines for the tumbling and tank treading regions are shown as well.

where η is the shear rate. The formulae indicate that the effect of shear rate on NP dispersion in the tumbling region is roughly an order of magnitude larger than that in the tank treading region. This can be attributed to larger RBC motions in the tumbling region, where RBCs undergo full body rotations that trigger larger local flow disturbances that promote the dispersion of adjacent NP away from the cell. It is also worth noting that the constant terms in the formulae are close to the NP thermal diffusion coefficient. The theoretical diffusion rate for 100 nm particles at a temperature of $300K$ is about $4.4 \times 10^{-12} m^2/s$. This observation agrees with the physical requirement that dispersion rate should be close to thermal diffusion in the absence of shear flow. Therefore, for a given hematocrit level, and with $\eta < 40s^{-1}$ and $\eta > 200s^{-1}$, the dispersion rate D can be written as

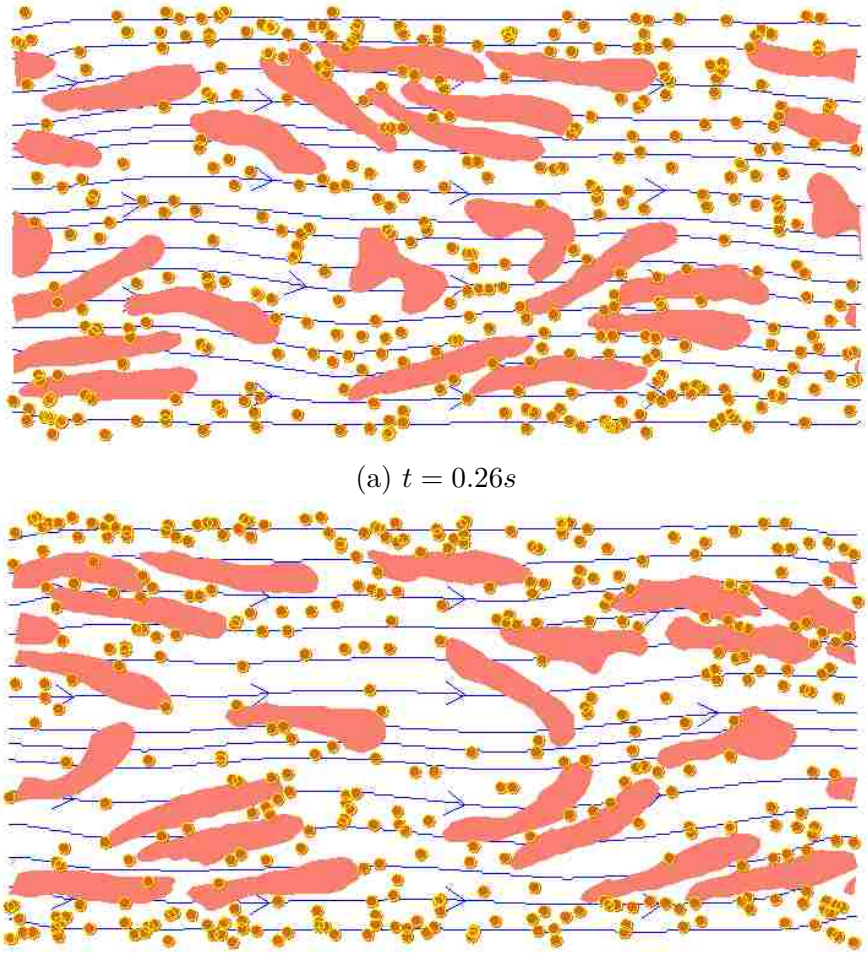
$$D = k\eta + D_0 \tag{6.2}$$

where D_0 is the thermal diffusion coefficient and k is a constant that depends on the hematocrit level. It is noted that this formula is readily adaptable to different particle sizes because the constant term D_0 already contains the particle size effect. The contribution of RBC motion is represented in the constant k. It is noted that the influence of particle concentration on dispersion rate has been neglected.

6.2 Nanoparticle dispersion under channel flow

The previous study of NP dispersion under shear flow provide a general function form between shear rate and dispersion rate. We hypothesis that the function form also applies to the case in channel flow. If the vessel pliancy is neglected, capillaries can be modeled as a rigid cylinder. In 2D case a channel flow profile is assumed. Fig.6.3a and 6.3b present snapshots of NP dispersion in a channel flow simulation with a hematocrit of 23.5% and a shear rate of $200s^{-1}$ at 0.26s and 0.46s, respectively. The channel width is $25\mu m$. For these simulations, the specified shear rate was measured as the shear rate at the wall, unless noted otherwise. Due to the increased cell volume, compared with the pure shear flow simulations, the number of NP was reduced to 378. The NPs were initially positioned in the core region of the channel. Since the shear rate is linearly changing across the channel, RBCs did not exhibit distinctive motions such as tumbling or tank treading as shown in the previous pure shear flow. The higher hematocrit and cell-cell interaction also confined the cell motion in the flow. The majority of the cells behaved like a tank treading motion, while some cells in the core region was bended or folded in the channel due to the symmetry of the velocity near the center line of the channel. The RBC motion under other hematocrit levels and shear rates were similar to Fig.6.3. However, they were not shown here.

As illustrated in Fig.6.3a and 6.3b, the NPs tend to migrate toward the wall.



(a) $t = 0.26s$

(b) $t = 0.46s$

Figure 6.3: Snapshots from a channel flow simulation for a cell-particle mixture with a hematocrit of 23.5% and a shear rate of $200s^{-1}$ at 0.26 s (a) and 0.46 s (b). Fluid streamlines are shown in the background, while the yellow markers represent 100 nm nanoparticles. For illustration purposes, cells and particles are not shown to scale.

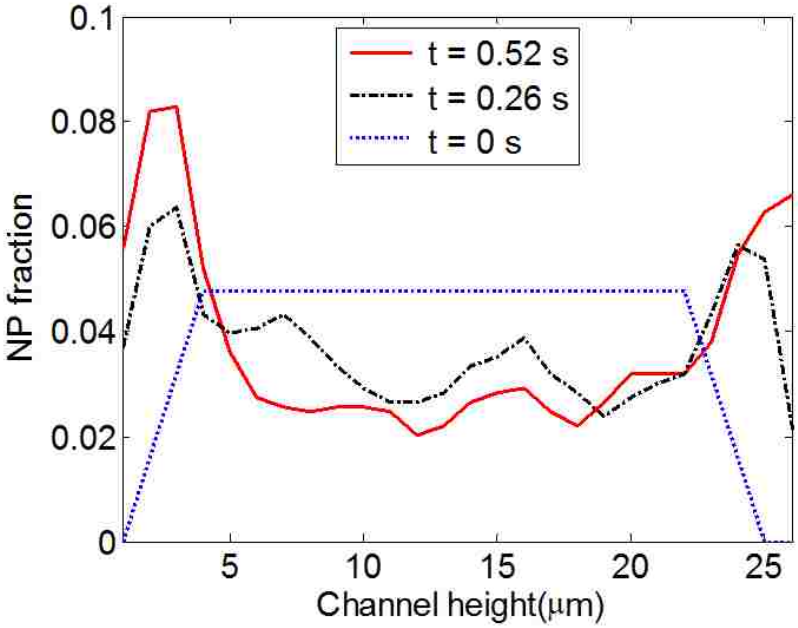
In order to characterize the NP distribution across the channel, the channel width was divided into bins of 1 μm. The number of NPs within each bin was counted and divided by the total number of NPs to obtain the NP fraction within each bin. NP fraction across the channel height is plotted in Fig.6.4a at time points of 0, $0.26s$, and $0.52s$, for a characteristic shear rate of $200s^{-1}$. Fig.6.4b presents NP fraction across the channel height at $t = 0.52s$ under shear rates of $100s^{-1}$, $200s^{-1}$, $300s^{-1}$, and $500s^{-1}$. The NP fractional values shown in the figure are the average of three sample runs using different random seeds for the NP Brownian motion model. Fig.6.4b clearly demonstrates that particle migration speed toward the channel walls increases with shear rate. In order to characterize NP migration speed, the dispersion rate was calculated from the NP mean square displacement. The dispersion rates for different hematocrit, and at various shear rates, are shown in Fig.6.5a. From the pure shear simulation results shown in Eqn6.2, a modified dimensionless dispersion rate was developed

$$D_r = \frac{D - D_0}{d_c^2 \eta} = f(H_t) \tag{6.3}$$

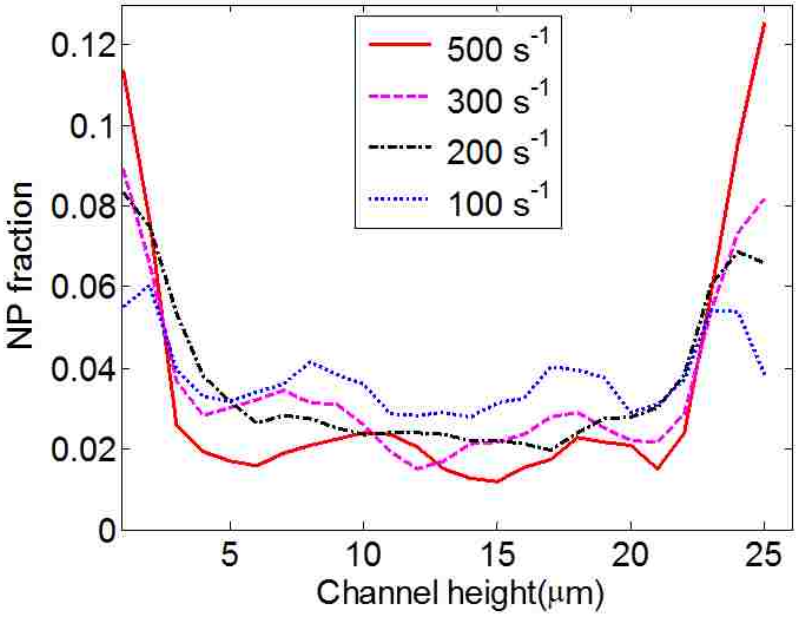
The dimensionless dispersion rate D_r is plotted in Fig.6.5b, where the error bars show standard variance for three sample runs.

Fig.6.5 shows that the lateral dispersion of NP (i.e. migration of NP toward the vessel walls) is much larger than what is predicted by thermal diffusion alone. This migration is influenced by both the hematocrit level and the shear rate. While the relationship between dispersion rate and shear rate is approximately linear (Fig.6.5a), the relationship between dispersion and hematocrit is not fully linear (Fig.6.5b). Nevertheless, a best fit line with reasonable approximation can be written as

$$D_r = \frac{D - D_0}{d_c^2 \eta} = 4.643 \times 10^{-3} H_t + 5.834 \times 10^{-4} \tag{6.4}$$

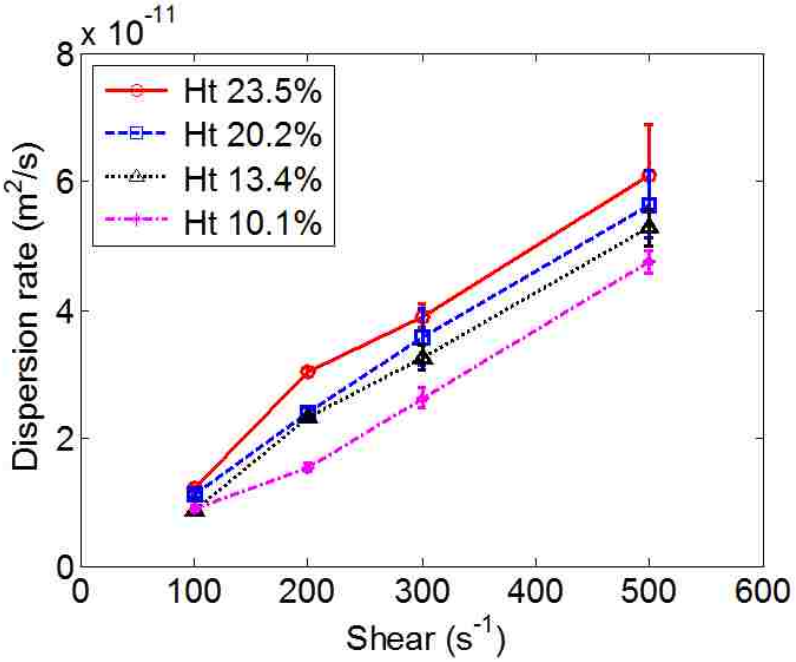


(a) Nanoparticle distribution across the channel height at $t = 0, 0.26s, 0.52s$ at shear rate of $200s^{-1}$

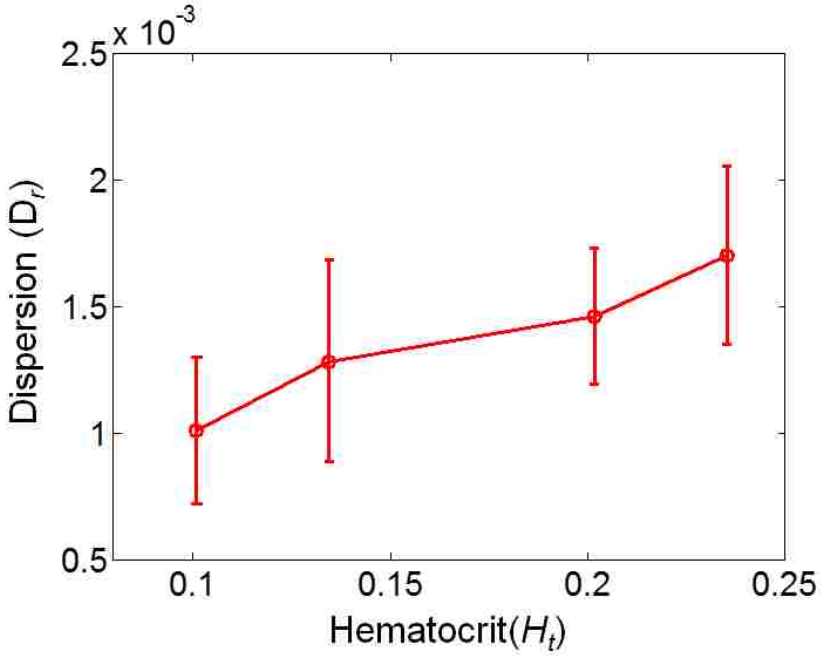


(b) Nanoparticle distribution across the channel height at t=0.52s for shear rates of $100s^{-1}$, $200s^{-1}$, $300s^{-1}$, and $500s^{-1}$

Figure 6.4: The NP fraction across the channel height for a hematocrit level of 23.5%. (a) NP fraction at t=0, 0.26s and 0.52s for a shear rate of 200 s-1, and (b) NP fraction at t=0.52s for shear rates of 100 s-1, 200 s-1, 300 s-1, and 500 s-1.



(a) Nanoparticle dispersion rate at different hematocrit (H_t) and shear rates



(b) Relationship between dimensionless dispersion rate (D_r) and hematocrit (H_t)

Figure 6.5: Dispersion rate of NPs in blood flow. (a) NP dispersion rate at different hematocrit and shear rates. (b) Relationship between dimensionless dispersion rate (D_r) and hematocrit. Error bars show the standard variance from three samples.

Hematocrit	Shear$[s^{-1}]$	Dispersion rate$[cm^2/s]$	Prediction$[cm^2/s]$	Reference
$[0.2, 0.4]$	400	$[0.5, 0.68] \times 10^{-6}$	$[0.39, 0.63] \times 10^{-6}$	[186]
$[0.2, 0.4]$	1100	$[1.5, 2.1] \times 10^{-6}$	$[1.1, 1.7] \times 10^{-6}$	[186]
$[0.1, 0.15]$	44.8	$[8.2, 11.9] \times 10^{-9}$	$[31.3, 37.9] \times 10^{-9}$	[179]
$[0.2]$	44.8	$[17.2] \times 10^{-9}$	$[44.6] \times 10^{-9}$	[179]
$[0.1, 0.2]$	804	$[0.9, 1.4] \times 10^{-7}$	$[5.4, 7.8] \times 10^{-7}$	[180]

Table 6.2: Comparison of particle dispersion rate predictions from Eqn.6.4 with dispersion rates reported in the literature.

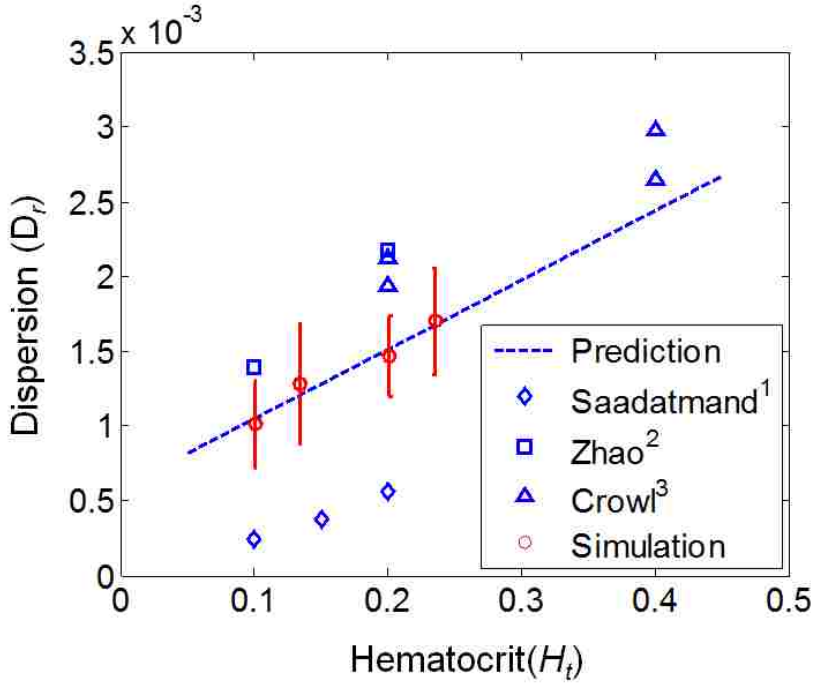


Figure 6.6: Comparison of particle dispersion rate predicted from Eqn.6.4 with the data reported in the literature(Saadatmand[1][179], Zhao[2][180], Crowl[3][186]). The dash line is the prediction from Eqn.6.4

In order to test the accuracy of Eqn.6.4, particle dispersion rate predictions from Eqn.6.4 were compared with dispersion rates published in literature for platelets[180, 186] and $1\mu m$ particles[179]. It is noted that the comparison of Eqn.6.4 with platelets and microparticles was considered due to the lack of NP dispersion rate in the literature. The predictions and measured dispersion rates are summarized in Table 6.2. The dimensionless dispersion rate is also plotted in Fig.6.6 As shown in both the table and the figure, the dispersion rate predictions using Eqn.6.4 are in good agreement with measured rates reported in the

literature. Discrepancies between the predicted and measured dispersion rates may be due to the linear correlation assumption for hematocrit, and/or the effect of particle concentration, which was very low for this study and was assumed to have a negligible effect on NP dispersion. Nevertheless, the order of magnitude of dispersion rate as well as its trend with hematocrit are correctly predicted by Eqn.6.4.

Chapter 7

Cell seperation based on deformability

7.1 Numerical methods for cancer cells

The cancer cell model is very similar to RBC model presented in Chapter 3. In this section, we discuss how to model cancer cells and their difference from RBCs. Cancer cells have a lot in common compared to white blood cells. The cell bending stiffness for white blood cells is $500k_BT$, about $2 \times 10^{-18}J$ in temperature of $300K$[187]. The stretching modulus for white blood cells is $18.9\mu N/m$. The deformability of cancer cells varies a lot. CTCs usually present a dormant state[188] where CTC is relatively stiffer. When they become malignant ones, the stiffness decreases significantly[46]. Many literature reported that the Young's modulus for cancer cells. Some data has been summarized in Table 7.1. Those data are

Cell line	Young's modulus(kPa)	Young's modulus(2D, $\mu N/m$)	Reference
BC3726	1.4($\pm$0.7)	14($\pm$7)	[189]
T24	1.0($\pm$0.5)	10($\pm$5)	[189]
Hu456	0.4($\pm$0.3)	4($\pm$3)	[189]
HeyA8	0.71($\pm$0.53)	7.1($\pm$5.3)	[37]
MLCs	N/A	413($\pm$15.2)	[190]

Table 7.1: Young's modulus for different cancer cell types

measured using atomic force microscopy(AFM). Interested readers can find how to use AFM and its application to cancer cell research in Ref.[191]. The Young's modulus for other cell types has been summarized in Ref. [192]. The Young's modulus can be used to determine the cell parameters in cell membrane nonlinear potentials. Combine Eqn. 3.10 and 3.11, we can get the in-plane potential. We only use the WLC potential as an example here.

$$U_{in-plane} = U_{WLC} + U_{POW} = \sum_{j\in 1...N_s} \left[\frac{k_B T l_m}{4p} \frac{3x_j^2 - 2x_j^3}{1 - x_j} + \frac{k_p}{l_j} \right] \tag{7.1}$$

where the $n = 2$ is used for Eqn.3.11.The derivation process is exactly the same for FENE potential. Young's modulus can be derived through the Cauchy stress and strain relationship. The method to calculating the stress from particle based method is through the virial theorem[193]. The shear modulus and Young's modulus for cell network model are summarized in Ref.[98, 102]. For example, for the combination of WLC and POW potential, the shear modulus μ is

$$\mu = \frac{\sqrt{3} k_B T}{4 p l_m x_0} \left(\frac{x_0}{2(1 - x_0)^3} - \frac{1}{4(1 - x_0)^2} + \frac{1}{4} \right) + \frac{3\sqrt{3} k_p}{4 l_0^3} \tag{7.2}$$

The shear modulus μ for the combination of FENE and POW potential is

$$\mu = \frac{\sqrt{3}}{4} \left(\frac{2 k_s x_0^2}{(1 - x_0^2)^2} + \frac{3 k_p}{l_0^3} \right) \tag{7.3}$$

The Poisson ratio ν of an isotropic triangular mesh is 1/3[98], thus, the Young's modulus E is

$$E = \frac{8}{3} \mu \tag{7.4}$$

The membrane thickness is about 100 Angstrom[93]. The thickness of the membrane should multiply Young's modulus to convert it into a 2D membrane elastic modulus. The corresponding Young's modulus for 2D membrane is shown in the

third column in Table 7.1. It shows that the Young's modulus for cancer cells is very similar to Red blood cells, e.g., $E \in (15 - 36\mu N/m)$ [102, p.53].

However, cancel cell modeling is different from red blood cells. Red blood cell has a biconcave shape which enables it undergo large deformation without increase its surface area. However, cancer cells typically have a spherical shape which has the minimum surface volume ratio. When cancer cells are squeezed through a narrow channel or through capillary, the spherical shape is transited into a sausage shape(i.e., a cylinder with hemispherical caps at both ends). The cell volume is conserved but the surface is increased. However, the cell membrane is typically inextensible. How does the cancer cell do that? The reason is the because the numerical microvilli on the cell surface. It has been found that the number of microvilli has been reduced during the cell squeezing process. An average of area increase about 52% was observed during the cancer cell sphere to sausage transition[194]. It is interesting that most of the cancer cells were dead after deformed from spheres with an average diameter of 16.5 μm to cylinders having a mean length of 53 μm in 7 μm capillaries[194]. That leads to the interesting future study in membrane rupture in both RBCs and cancers. A short introduction to cell damage was presented in Section 8.2.1. The surface compressibility is used to characterize the deformability of the cells. The surface compression modulus K, defined as

$$K = -\frac{\partial P}{\partial \log(A)}\bigg|_{A=A_0} \tag{7.5}$$

where P is the in-plane pressure given by $P = -\frac{1}{2}(\tau_{xx} + \tau_{yy})$, A is the surface area. Fedosov analyzed different spring connected network models and gave the relationship between K and other potential constants[102, p.43]. For example, for both WLC and POW potential and FENE and POW potential,

$$K = 2\mu_0 + k_g + k_l \tag{7.6}$$

where μ_0 is the shear modulus, and k_g,k_l are area conservation constants used in Eqn.3.15. Generally, $k_g + k_l \gg \mu_0$ is required for nearly incompressible spring network. In our model, different compression modulus were used to study the cell squeezing process, e.g., healthy cells with less deformability and cancer cells with more deformability.

Another difference between cancer cell model and RBC model is that there is a nucleus within the cell. The size of a cell's nucleus is usually proportional to the size of the cell itself[195, 196], with nucleus volume occupying 7% of the cell volume[197]. The nucleus is about 5 to 10 times stiffer than the cytoskeleton[198]. Thus, nucleus is a barrier for 3D migration[199]. However, for highly deformable cells, such as cancer cells, their nuclei are more flexible so that they can migrate through narrow regions. We did not explicitly model the nucleus. Instead, we treated the nucleus exactly the same as cytoplasm. This assumption was also used by other references[200].

7.2 Lubrication force

One of the challenges in modeling cell squeezing is to correctly capture the thin fluid behavior when the cell membrane moves close to the wall. When the gap between the cell membrane and the wall is very small, e.g., the gap is smaller than a lattice space, the LBM fluid solver can not resolve the thin fluid flow motion. One approach is to use a finer mesh for the whole fluid domain or refine the mesh near the boundary layers[201–203]. This approach would increase the LBM simulation time as the lattice space is reduced. It also requires some efforts to handle the density distribution passage over the interface if a multgrid is used in LBM. Another approach is to introduce an lubrication force to repel the cell membrane so that there are enough fluid within the gap. The physics behind the lubrication force model is that the fluid within the gap has to be displaced if the

cell membrane moves close to the wall. As the gap gets smaller, there is less space for fluid displacement, thus, requiring a larger force to drive the fluid away. This results a repelling force on the cell membrane. The faster the cell moves close to the wall, the bigger the repelling force generated by the fluid. The lubrication repelling force was introduced to the lattice Boltzmann method in Ref[204, 205]. Following their work, the lubrication force derived from the lubrication theory between two identical spheres is

$$\boldsymbol{F}_{ij}^{lub} = -\frac{3\pi\mu r}{s}\hat{\boldsymbol{x}}_{ij}\hat{\boldsymbol{x}}_{ij} \cdot (\boldsymbol{u}_i - \boldsymbol{u}_j) \tag{7.7}$$

where r is the spherical radius, μ is the fluid dynamic viscosity, s is dimensionless gap $s = R/r - 2$ where d is the R is the central distance between two spheres. $\boldsymbol{x}_{ij}$ is the position vector difference between sphere i and j, defined as $\boldsymbol{x}_{ij} = \boldsymbol{x}_i - \boldsymbol{x}_j$, $\hat{\boldsymbol{x}}_{ij}$ is the unit vector. $\boldsymbol{u}$ is the spherical velocity. Eqn7.7 can also be extended to the case where a sphere approach a stationary wall by setting $\boldsymbol{u}_j = 0$.

7.3 Model setup

In our model, the geometry of the microfluidic channel is $43\mu m \times 30\mu m \times 30\mu m$, with a narrow pore of diameter of $10\mu m$. The side view and left view of the channel is shown in Fig.7.1. The pore size is comparable to other narrow channel size used in microfluidics. e.g., a square cross section of $10\mu m \times 10\mu m$ was used in Ref.[38], or a cross section of $6\mu m \times 15\mu m$ in Ref.[46]. The critical channel diameter to filter CTC from RBCs in microfluidic channel is reported between 5 and 12 μm[200]. Bounce back boundary conditions were applied at the walls. The inlet was applied a pressure of 15 Pa, while the outlet pressure was set at 0. The initial set up of the system is shown in Fig.7.2. The lattice size for the fluid is $0.5\mu m$. The cell has 2562 nodes and 5120 triangle surfaces with mean bond length of 0.56 μm. The relaxation parameter $\tau = 0.9$.

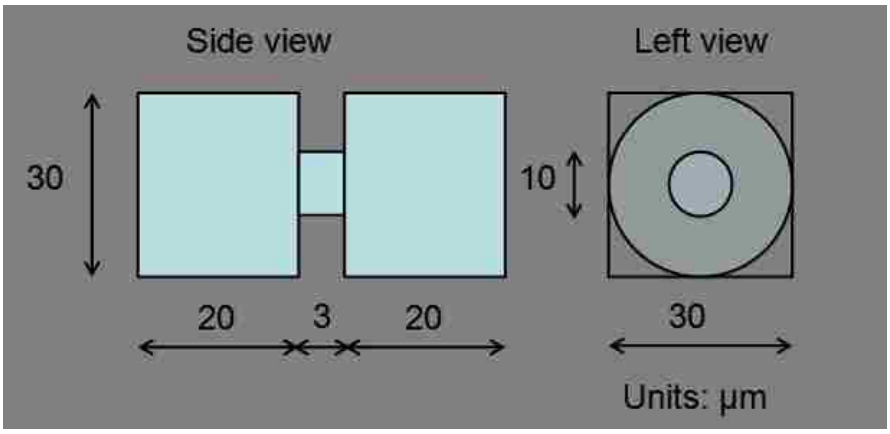


Figure 7.1: The geometry of the channel with narrow pore for cell squeezing test

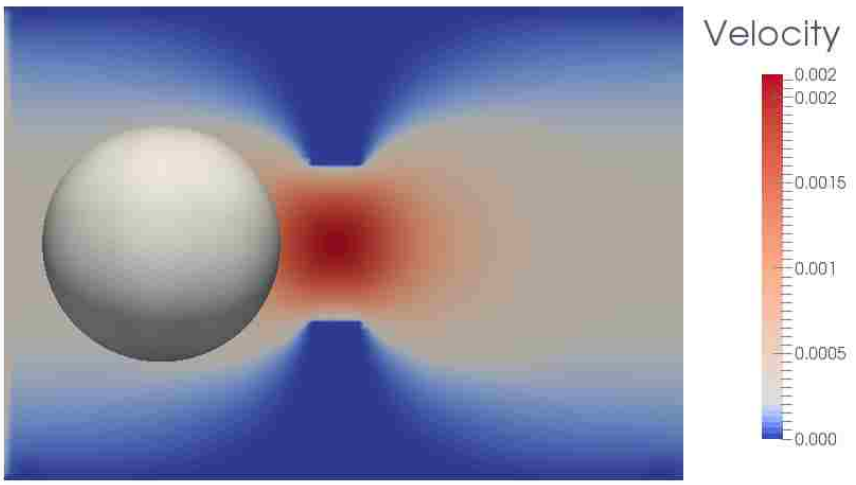


Figure 7.2: The initial setup of the cancer cell squeezing through a narrow pore. The cell with a diameter of 15 μm was positioned at 10 μm away from the inlet. The pore size is 10 μm in diameter and 3 μm in length located at x= 20 μm. A 15 Pa pressure was applied at the left inlet. The right outlet pressure was set to 0. The background color shows the velocity distribution.

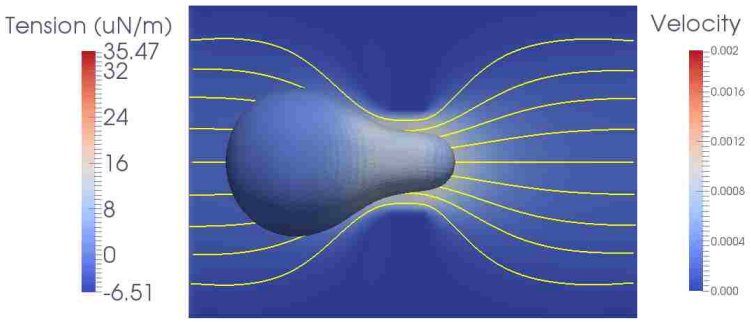
The physical properties of the membrane are listed in Table.7.2. Here the FENE and POW potential were used.

ID	$k_s(\mu N/m)$	$k_b(J)$	$k_g/k_l(\mu N/m)$	K $(\mu N/m)$	k_v
K5	5	2×10^{-19}	0	20	1×10^4
K4	5	2×10^{-19}	2	24	1×10^4
K3	5	2×10^{-19}	20	60	1×10^4
K2	5	2×10^{-19}	200	420	1×10^4
K1	5	2×10^{-19}	2000	4020	1×10^4

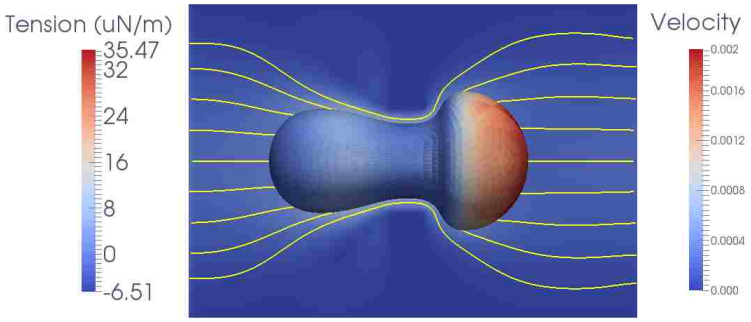
Table 7.2: Cell membrane properties for all the simulations performed in cells squeezing test

7.4 Numerical results

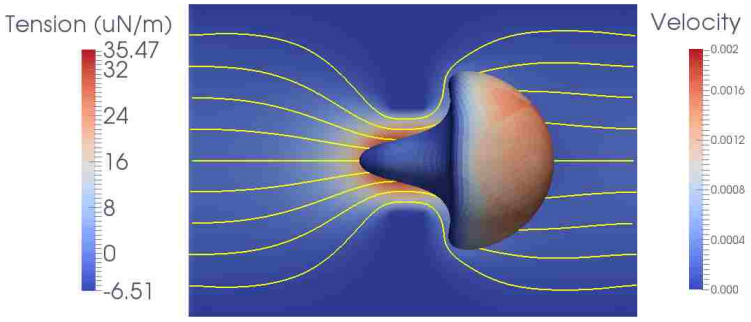
Cells with five different deformability properties(see Table 7.2) were studied in this work. Except the case $K1$, the cell translocation patterns were very similar. Here only the $K3$ cell was selected for analysis. The time sequence of the simulation were shown in Fig.7.3. As shown in the figure, the cell membrane would form a protrusion at the heading membrane due to the convergent flow, as indicated by the streamlines. Once the leading membrane passed through the micropore, it would expand at the other side of the pore. The tension distribution of the cell membrane is not uniform, with higher stress at the heading membrane and the membrane within the micropore. The tail of the membrane were shrunk first before it entered the pore, and then quickly stretched out and bend in the reverse direction once it reached at the other side of the pore. After the tailing membrane left the micropore, the cell would keep expanding toward the wall until it reached a steady state, as shown in Fig.7.3d. The streamlines bended toward the wall sharply after the tailing membrane left the micropore. The changes of the streamlines can be seen from the comparison between Fig.7.3a and 7.3d.



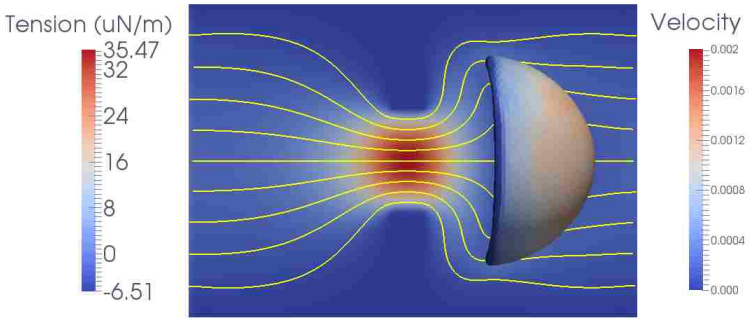
(a) $t = 0.33ms$



(b) $t = 1.67ms$



(c) $t = 2.67ms$



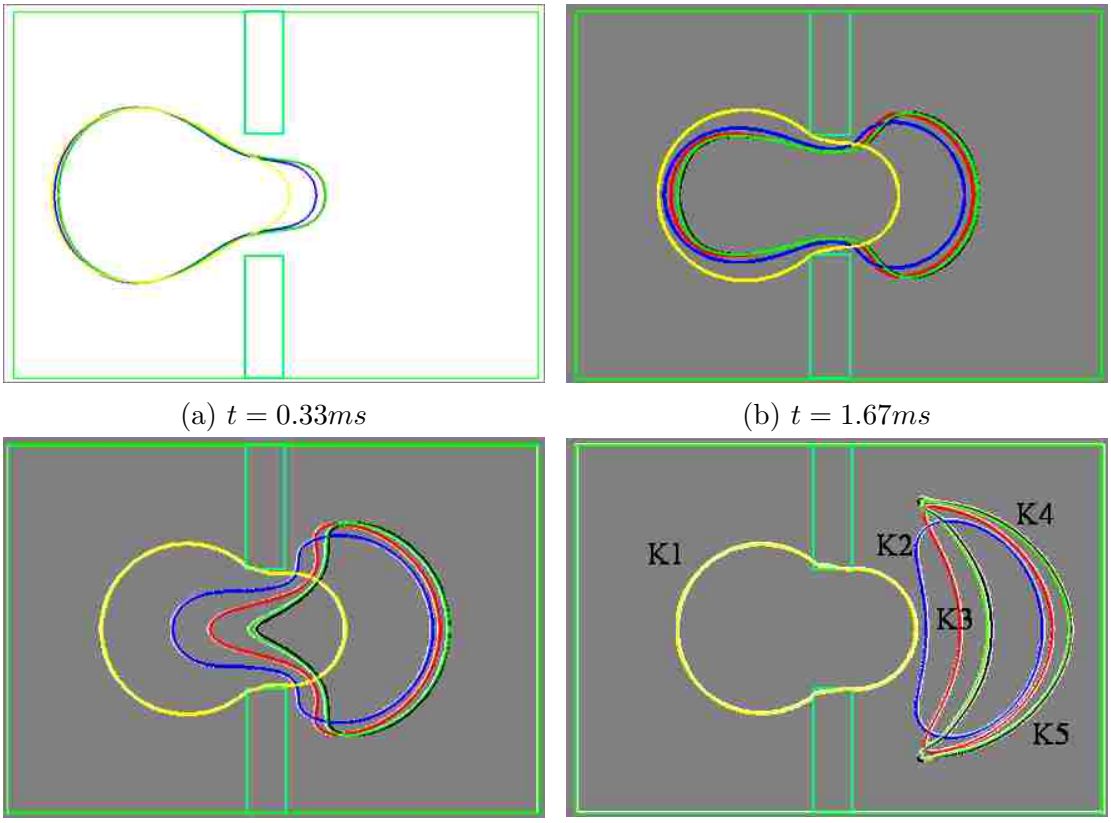
(d) $t = 4.33ms$

Figure 7.3: Snapshots of cell squeezing through the micropore at different times. Velocity magnitude (background, in LB dimensionless units), streamlines (yellow), membrane tension are shown in the figure.

7.4.1 Deformability effect on cell translocation

The snapshots of the simulation of cell with different area compressibility modulus squeezing through a narrow pore is shown in Fig.7.4. The gap formed between the cell membrane and the pore wall depends on the cell deformability, with larger gap for smaller membrane compressibility modulus. It is interesting to see that the cell with $K = 4020\mu N/m$ could not pass through the pore, as indicated by the yellow line in Fig.7.4c and 7.4d. The deformability also influence the curvature of the membrane for the steady parachute shape, with larger curvature for flexible membranes. Notice that the case $K4$ and $K5$ did not show too much difference during the translocation process. This indicates that the nonlinear in plane stretching potential contribution is dominant compared with area conservation potential.

During the cell translocation process, the flow volume rate passed the cross section of the micropore can be used to characterize the translocation time. The volume rate is also related to the ionic current that can be directly measured through electronic nodes[206]. The conductivity of the micropore depends on the physical blockage by cells, thus it is proportional to the volume rate of the solution. The volume rate through the micropore ($x = 43$) was measured in all the simulations performed, as shown in Fig.7.5. As shown from the figure, the volume flow rate decreases more than 80% as the cell passes the pore, resulting a sharp dip in the volume rate time history curve. The flow rate returns to the original level after the cell leaves the pore. The time of the blockage depends on the cell deformability, with shorter translocation time for softer cancer cells. This is consistent with the experimental data reported in Ref[206]. It is interesting to see that there is a second dip in the flow rate for less deformable cells. That is due to the tension increase in tail membrane of the cell. For example, for the cell case $K3$, the tension of the tail membrane at the peak volume rate (the data point at $t = 2.99ms$) is shown in Fig.7.6a, while the tension of the membrane at the dip in the flow rate(the point at $t = 3.67ms$) is shown in Fig.7.6b



(a) $t = 0.33ms$ (b) $t = 1.67ms$

(c) $t = 2.67ms$ (d) $t = 4.33ms$

Figure 7.4: Snapshots of cell squeezing through a narrow pore at different time. Cell profiles in the plane sliced through the cell center are shown in the figures. Cells with different deformability (area compressibility modulus K, in units of $\mu N/m$) are shown in different colors. Yellow: $K1 = 4020$, Blue: $K2 = 420$, Red: $K3 = 420$, Green: $K4 = 60$, Black: $K5 = 20$. The background color shows the velocity distribution for the $K2$ case. The fluid distribution for cells with different deformability shows similar pattern. They are not shown here.

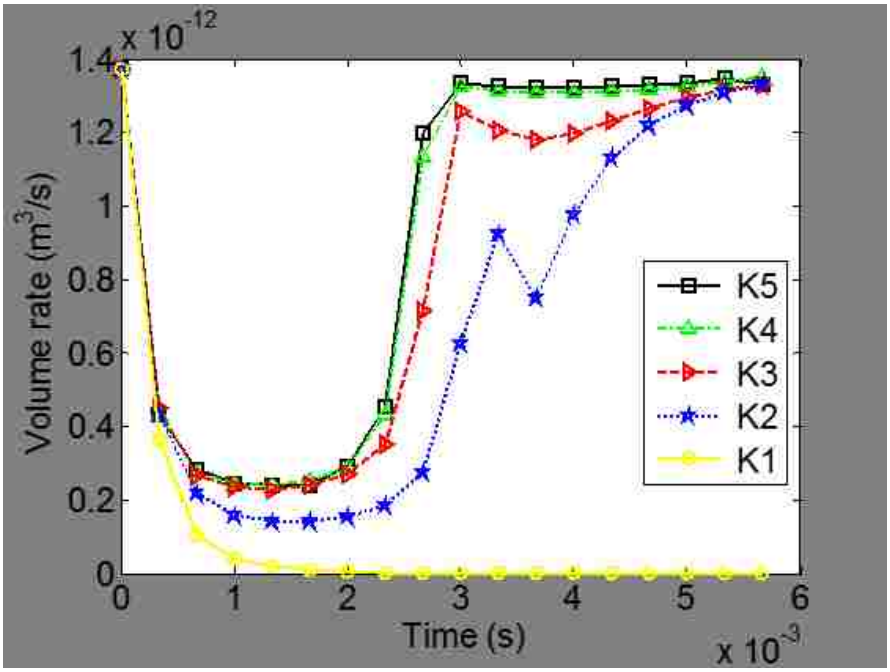
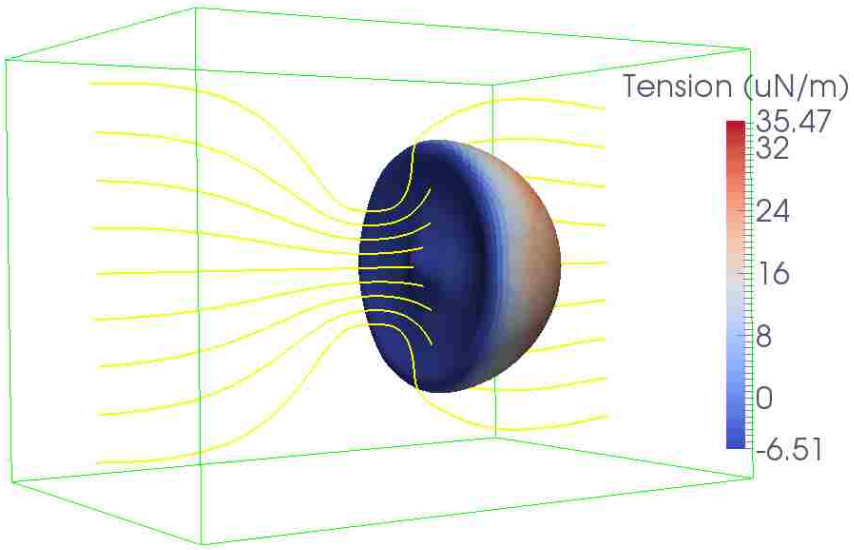


Figure 7.5: The volume rate passing through the micropore for cells with different deformabilities.

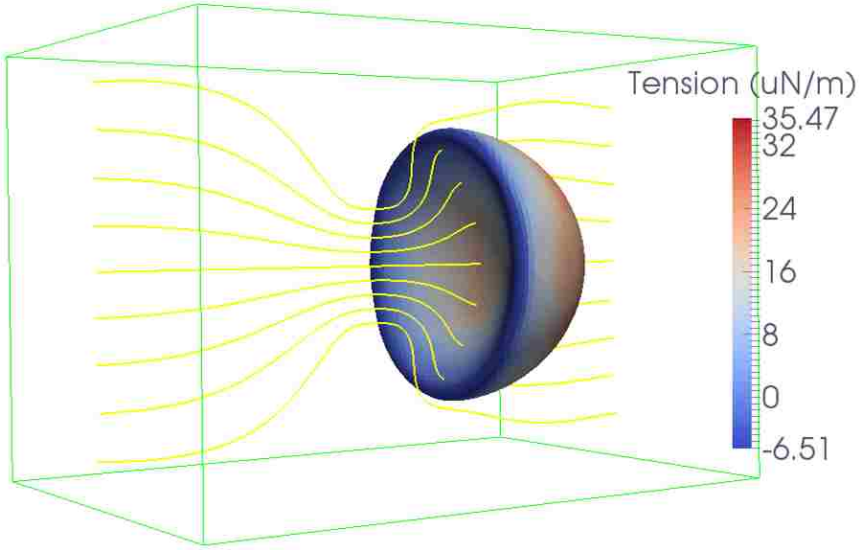
To characterize the easiness of cells squeezing through the micropore, the translocation time of the cells were recorded. The translocation time is obtained from the volume rate time history at the middle section of the micropore(see Fig.7.5). The translocation time for different cell deformability is shown in Fig.7.7 The time for $K1 = 4020\mu N/m$ is not shown in the figure, because the cell is blocked at the entrance of the pore, the translocation time is infinity. It can be seen from the figure that the translocation time is not sensitive to the cell deformability at the pore size of $10\mu m$ for cell size of $15\mu m$.

7.4.2 Pressure and pore size effect on cell translocation

Different pressure difference for cell with deformability $20\mu N/m$ (the case $K5$) squeezing through a $10\mu m$ micropore was studied. The pressure difference applied at the inlet is $5pa, 10pa, 15pa$, and $25pa$. The volume rate at the middle section of the micropore is shown in Fig.7.8 It can be seen that the volume rate at different pressure almost all decreased by 80%. The pressure change does not



(a) $t = 2.99ms$



(b) $t = 3.67ms$

Figure 7.6: Tension changes in the tail membrane after the cell exits the micropore.

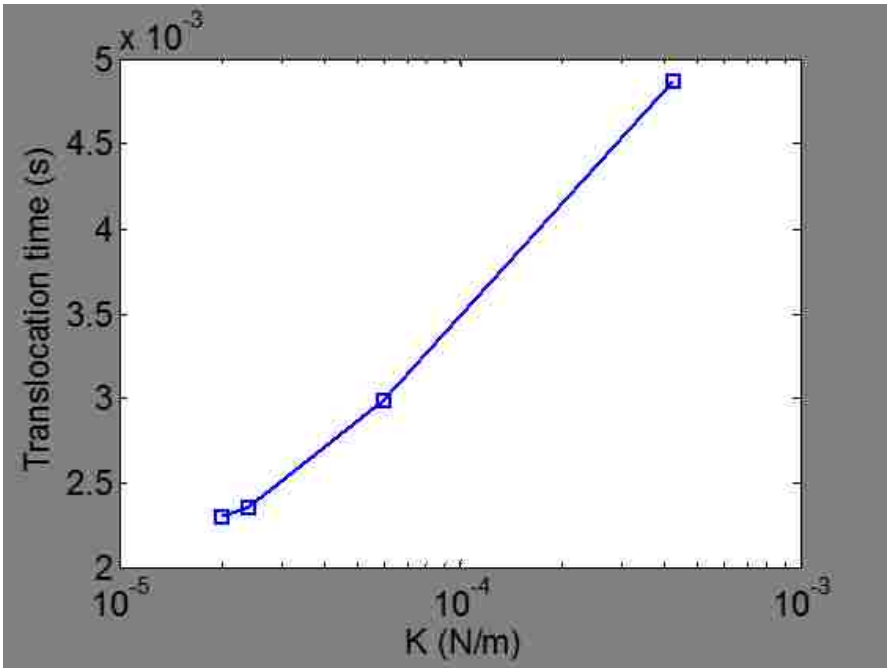
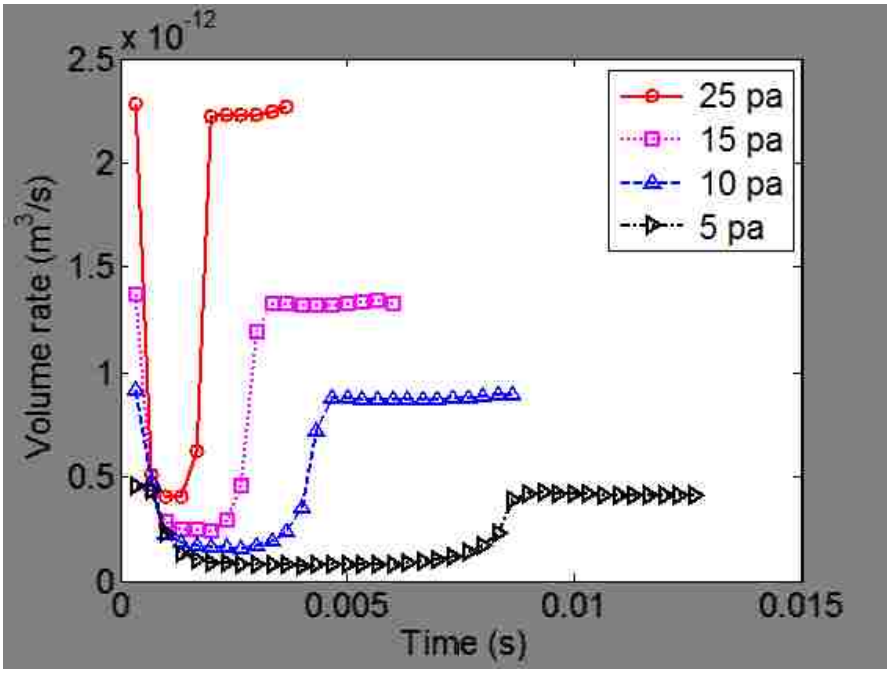


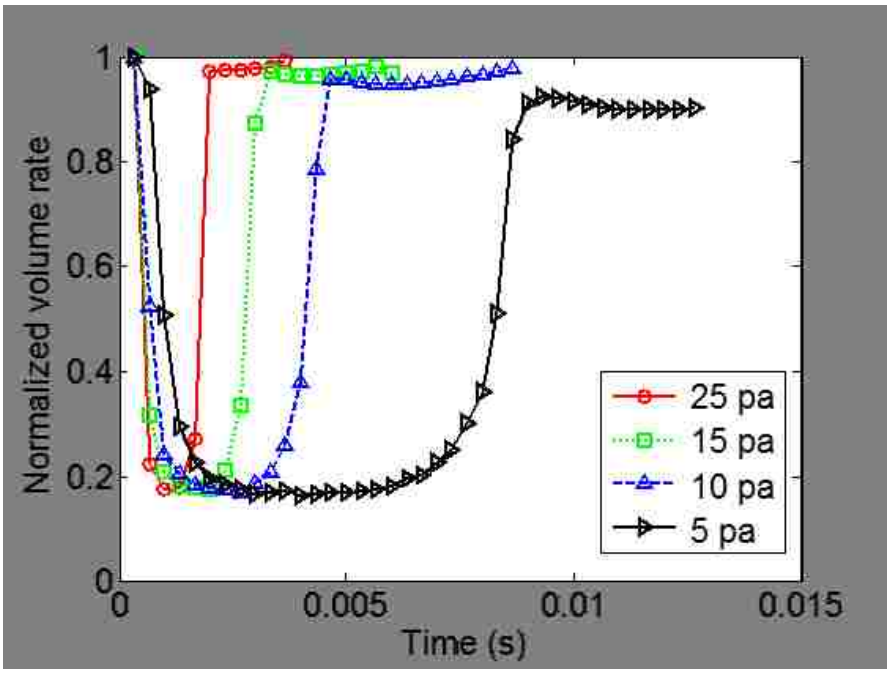
Figure 7.7: The translocation time for cells passing through a micropore with different deformabilities.

influence the percentage of decreased volume rate. However, it does delay the translocation process. The translocation time of the cells under different pressure is show in Fig.7.9 As shown from the figure, the translocation time decreases with the increasing pressure. The translocation time decreases exponentially with pressure increase.

Similarly, the influence of micropore size on cell tanslocation time was also studied. The cell has an area compressibility modulus of $20\mu N/m$. The applied pressure difference is $15pa$. Three difference micropore diameters are $8\mu m$, $10\mu m$, and $12\mu m$ in the study. The time history of the volume rate at the middle of the micropore is shown in Fig.7.10. The dip depth of volume rate curve increases as the diameter of the micropore decreases. The pattern of the volume rate for cell passing through the micropore is similar to the cases with different pressure differences. The translocation time for different micropore size is shown in Fig.7.11.



(a)



(b)

Figure 7.8: The volume rate at the middle section of the micropore under different pressure difference for cells with deformability of $20\mu N/m$ squeezing through a micropore with diameter of $10\mu m$. (a) volume rate; (b) normalized volume rate.

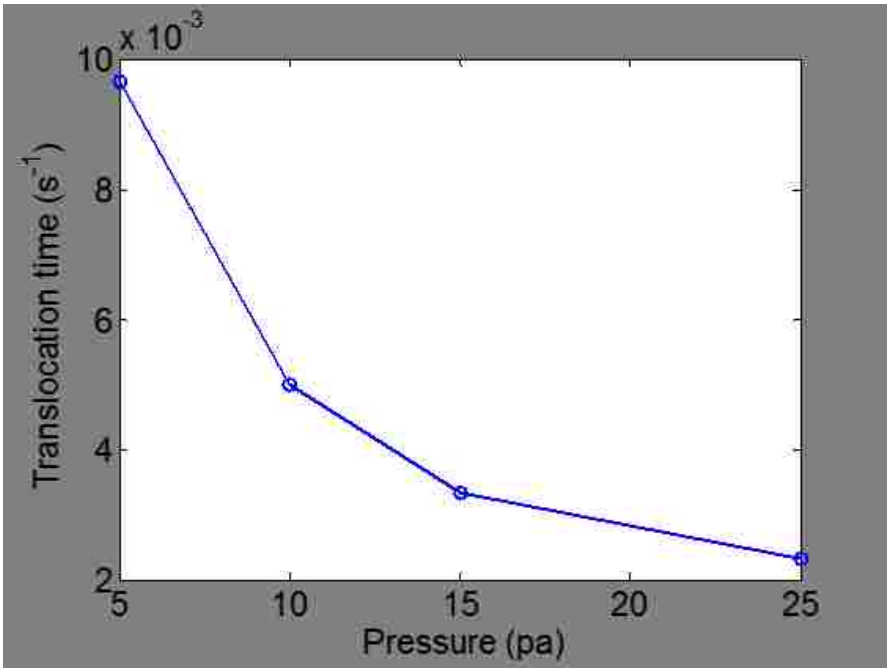
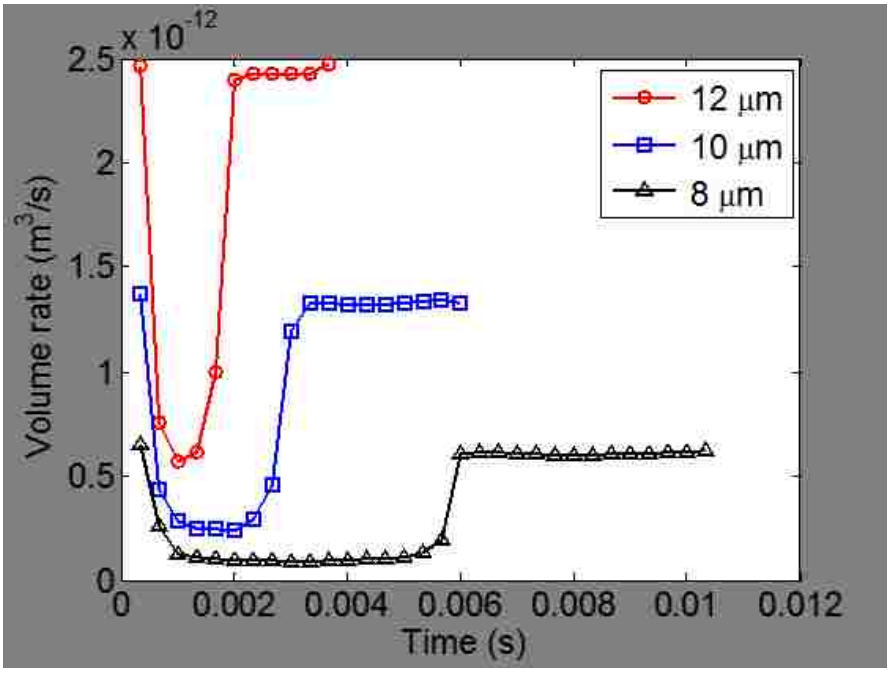


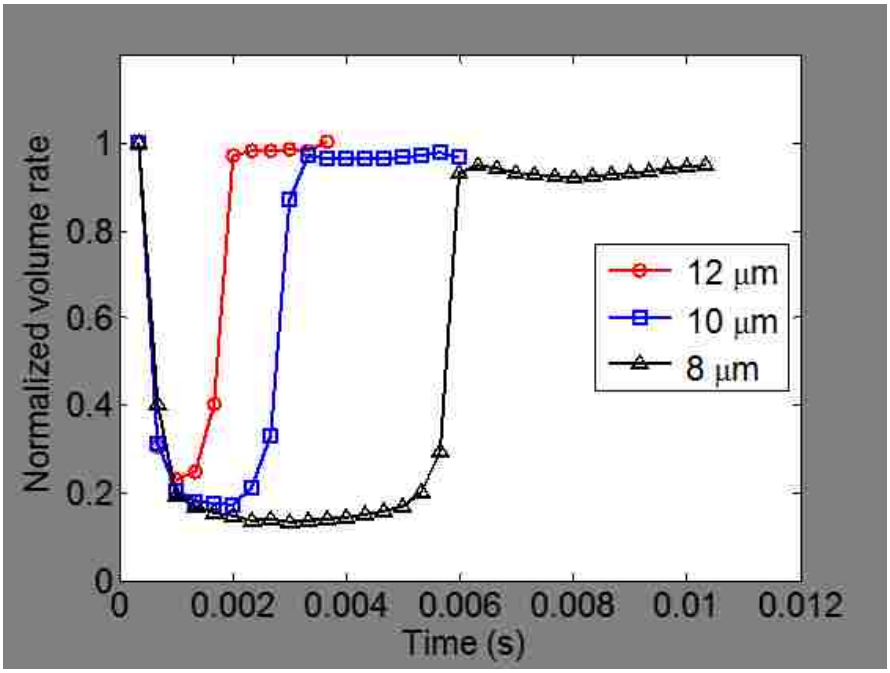
Figure 7.9: The translocation time under different pressure for cells with deformability of $20\mu N/m$ squeezing through a micropore with diameter of $10\mu m$.

7.4.3 Rational design of microfluidics for cell separation

The ultimate goal of the study is to provide rational design principles of microfluidics so that they could be used to efficiently to separate cancer cell from other cells in the blood sample. Two very important parameters in cell separation design are the pressure difference ΔP and the diameter of micropore d. The following section is trying to propose a simple model to design microfluidics. For example, what is the minimal pressure difference ΔP required to squeeze the cells given the cell physical parameters and micropore size d ? What is the minimal pore size under given pressure difference? When the micropore is blocked by the cell, the leading and tailing membrane could be treated as spherical caps, while the middle part of the membrane could be treated as a cylinder, as shown in Fig.7.12. The cell bending stiffness is much smaller than the stretching stiffness, thus the membrane equilibrium conditions could be reasonably assumed to follow Laplace-Young equation. Using a modified Laplace-Young equation[190, 207], we



(a)



(b)

Figure 7.10: The volume rate at the middle section of the micropore for different pore size for cells with deformability of $20\mu N/m$ squeezing through a micropore under pressure difference of $15pa$. (a) volume rate; (b) normalized volume rate.

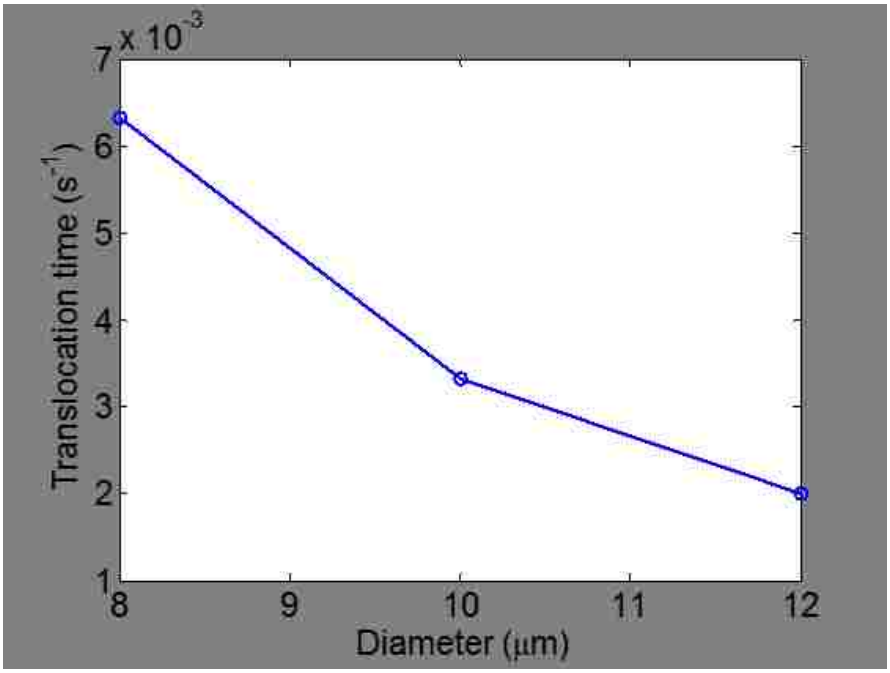


Figure 7.11: The translocation time through different micropore size for cells with deformability of $20\mu N/m$ under the pressure of $15pa$.

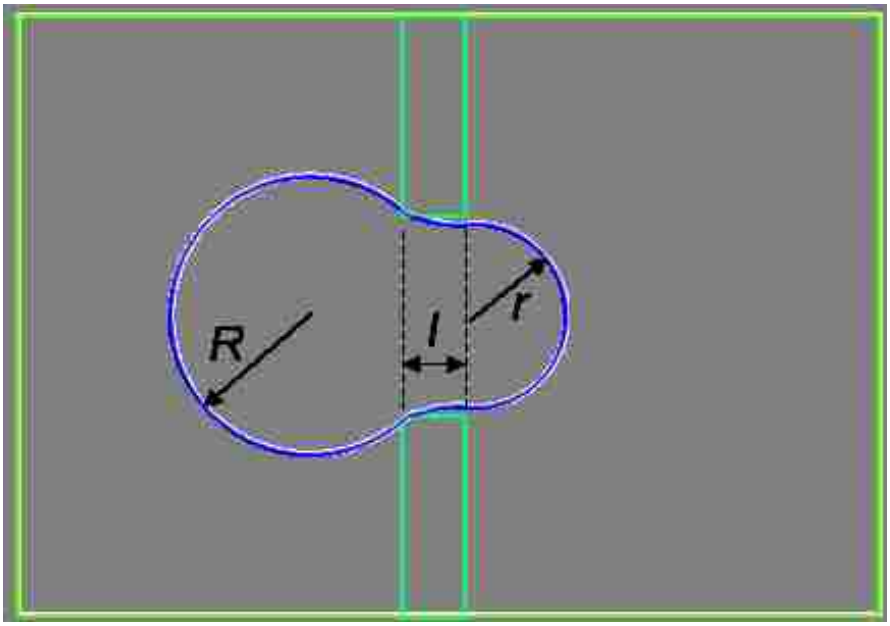


Figure 7.12: The micropore is blocked by the cells. The head and tail membrane form two curved surfaces(curves shown in the figure) that can be approximated as spherical caps with different radius r and R. The middle part of the membrane could be approximated as a cylinder with length l and radius r.

have

$$\Delta P = 2\gamma(\frac{1}{r} - \frac{1}{R}) \tag{7.8}$$

where r and R are the smaller hemisphere radius and bigger hemisphere radius, respectively. γ is the mean tension over the membrane. Noted here that the stress distribution over our cell model is not exactly uniform. For simplicity, the tension is assumed to be uniformly distributed over the whole membrane. From the conservation of volume, we have

$$\frac{\pi(R+\sqrt{R^2-r^2})}{6}(3r^2 + (R+\sqrt{R^2-r^2})^2) + \pi r^2 l + \frac{2}{3}\pi r^3 = V_0 \tag{7.9}$$

The surface area can be calculated as

$$2\pi R(R+\sqrt{R^2-r^2}) + 2\pi r l + 2\pi r^2 = A \tag{7.10}$$

The ideal case is considered here that the smaller spherical cap is exactly a half sphere with radius $r = d/2$ where d is the micropore diameter. From Eqn. 7.9 and 7.10, with the assumption $r = d/2$, we can find the solution for R and l. For example, in the $K1$ case shown in Fig.7.4, we find that $R = 7\mu m$, $l = 2.3\mu m$, the minimal required pressure difference $\Delta P = 25pa$, which is about 67% higher than the pressure difference specified at the inlet $\Delta P_0 = 15pa$. The difference could be several reasons. First, the cell membrane model used here does resist bending, which is not considered in the Laplace-Young equation (7.8). Second, the assumption of a uniform tension distribution over the surface is not exactly correct. The tension distribution of the cell membrane is shown in Fig.7.13. It clearly shows a high stress region in the middle part of the membrane within the micropore. The heading and tailing membrane have relatively small tension. Thus, the uniform tension distribution assumption is not correct. To consider the nonuniform distribution of the tension, the local tension were used to predict the

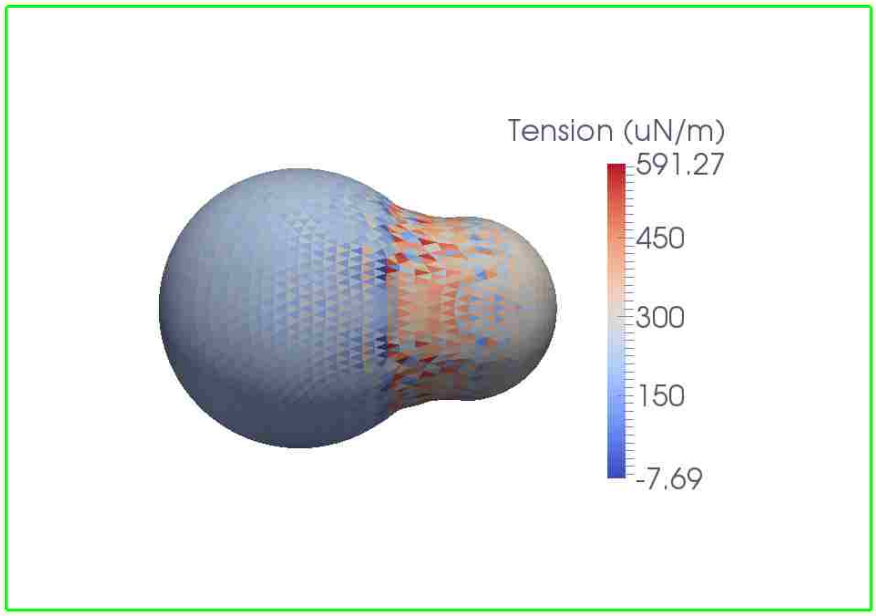


Figure 7.13: Nonuniform tension distribution over the cell membrane. Concentrated high tension was observed on the membrane within the pore. The tension in $\mu N/m$ are shown in the color bar.

pressure difference. The local regions used to calculate the tension were defined as two hemispheres at the heading and tailing membrane. This approach is also consistent with the conditions that Eqn.7.8 derived from.

With local surface averaged tension $\gamma = 128.2\mu N/m$, the predicted pressure $\Delta P = 14.4pa$, which is reasonably close to the applied pressure $15pa$. Thus, through the FSI code, we could calculate the minimal pressure required to squeeze through the micropore. Similarly, the micropore size could also be optimized under given pressure difference and cell physical properties.

Chapter 8

Conclusion and future work

8.1 Conclusion

This dissertation was conducted in the background of modeling of biological flow and soft matter where the interaction between cells and local fluid is very important and has to be resolved. Lattice Boltzmann method was selected as the fluid solver due to its capabilities in multiphase modeling, parallel computing, and easy implementation of the algorithm and complex geometries. The dissertation summarized the techniques of Lattice Boltzmann method as a fluid solver, including the theoretical background, discretization, nondimensionalization, collision schemes and boundary conditions. A network based cell model (coarse grained molecular dynamics) was selected due to its capabilities in handling large deformation and nonlinearity of solids. The cell model is also promising in modeling cell damage which would be one of the future studies. The network model was largely followed Fedosov's dissertation[102], including nonlinear potentials, bending energy, and surface area and volume conservation constraints. The numerical procedure to calculate cell nodal force was given in details. To couple the fluid flow with the cell motion, the Immersed Boundary Method (IBM) was selected as the coupling scheme. IBM removes the burden of mesh updating in tradi-

tional Arbitrary Lagrangian Eulerian (ALE) approach through a force density that represents the effect of immersed solid boundaries. Thus it is very efficient in modeling fluid structure interaction (FSI) problems in soft matter. Nanoparticle (NP) model was also introduced into the coupling system to study the NP delivery in microcirculation. The NP model is directly derived from molecular dynamics (MD) approach. A significant time was devoted to develop the code in 2D and 3D.

The developed code was benchmarked for fluid solver through a lid driven cavity flow, the cell behavior was benchmarked with stretching tests. The FSI was benchmarked through a dropping sphere in a quiescent channel. All the benchmark work confirmed that the fluid flow and cell mechanics can correctly capture the hydrodynamics and biological cell behavior. As applications to the developed code, we first studied the NP dispersion rate under the influence of cells and shear rates; then, we also studied the cancer cell translocation process through a micropore. Particularly, it demonstrated that how this code can be used to optimize the key parameters such as pressure difference and micropore size in microfludics that can efficiently separate cells based on the stiffness.

The contributions of the dissertation work are summarized below.

8.1.1 Nanoparticle dispersion rate

This first part of the dissertation presents a numerical study on NP dispersion in blood flow considering the influence of RBC motion and deformation. NP dispersion rate is found to be strongly influenced by RBC motion, and to have an approximately linear relationship with shear rate in the RBC tumbling ($\eta < 40s^{-1}$) and RBC tank treading ($\eta > 200s^{-1}$) regions of the flow regime. Between these two regions, the relationship between NP dispersion rate and shear rate is not as well defined. More importantly, a general formula to estimate NP dispersion rate for a given shear rate was then developed as $D = k\eta + D_0$ where D_0 is the

thermal diffusion coefficient, k is a constant that depends on the hematocrit and capillary number. NP dispersion with cell suspensions in channel flow was also studied. These simulations were used to characterize NP migration to the vessel walls as a function of shear rate at different hematocrit levels. The predictions given by the proposed empirical formulae agree well with data reported in the literature. These simple predictive analytical formulae provide an efficient approach for assessing NP dispersion under various flow conditions and hematocrit levels, thereby facilitating practical modeling of NP transport and distribution in large scale vascular systems. Compared to this study, the previous modeling results in literature[25, 26] did not provide an analytical formula to predict the dispersion rate under shear and channel flow. In the future, the model presented in this paper will be extended to study 3D cell-particle interaction and transport behavior, and a NP binding model will be developed to explore NP binding and distribution in capillary vessels. The proposed formula for the dispersion rate will also be used to evaluate NP transport and distribution in a large scale vascular network.

8.1.2 Microfluidic design

Deformability could be used as a biomarker to detect circulating cancer cells. The more deformable the cells, the easier they translocate through a micropore. Applied pressure difference and the micropore size are the two key parameters in microfluidics design for cell separation. This work presents 3D modeling results of cells with different deformability squeezing through a micropore. The numerical results indicate that the translocation time of cells is not sensitive to the cell deformability. For example, the translocation time only varies by a few times while the cell surface compressibility modulus changes by orders of magnitude. However, the translocation time strongly depends on the applied pressure difference and the diameter of the micropore. The translocation time increases exponentially as the pressure or the diameter of the micropore decreases. During the cell

squeezing process, the membrane tension is not uniform over the cell surface, with high tension at the leading part of the membrane. When the cell is in contact with the wall, such as the case when cell is too stiff to be blocked from passing the pore, a concentrated high stress was observed for the contacted membrane. The dissertation also provide a method to predict the minimal applied pressure required to squeeze the cell through a certain size of a micropore. A modified Laplace-Young equation with local membrane tension combined with volume conservation and surface area constraint could be used to predict the critical pressure that the cell is exactly stuck at the micropore. During the calculation process, the membrane tension and surface area are provided from the simulation results. The bending contribution is not considered in the Laplace-Young equation. This assumption works for highly deformable cancer cells, however, it may not work for relatively rigid cells.

8.2 Future work

The future work includes two part. The first part is to explore the new physics in cell damage through introducing a bond breaking mechanism in the coarse grained cell membrane model. The second part is to implement parallel computing features into the code.

8.2.1 Red blood cell damage

Cell transport in flow is important for various medical devices such as syringe pump [208], artificial heart[209], heart valve[210], and bio-printer[211]. Cell damage usually happens when they are in direct contact with the surfaces of these devices or when they experience extremely high shear stress in complex geometries such as a converging channel, rotating blade, biodispenser, or an injection nozzle[212, 213]. In patients, hemolysis resulting from blood-device interactions

has been linked to renal failure, anemia, arrhythmias, and death[214, 215]. For instance, Red blood cells (RBCs) damage induced by ventricular assist devices (VADs) over a long period of time is still an unsolved issue[216]. Besides medical devices, cell damage is also a big concern in biofabrication and microfluidic lab-on-chip devices [217–219], where individual or groups of cells are manipulated in a small channel such as syringe-based cell deposition for tissue constructs [220] and inkjet-based cell printing[217, 221]. In these examples, cells are subjected to combined effects of shear, tension and compression. Prolonged contact and collision between blood cells and device surfaces and regions of high shear stress contribute to cell damage[222, 223]. The consequence of RBC damage can be sudden, and potentially fatal. The damage induced nitric oxide depletion results in pulmonary hypertension, abdominal pain, and some other physiological dysfunctions[224]. Thus, it is very important to evaluate the blood damage for safety evaluation of blood-wetted medical devices. The network based cell model is very useful in modeling cell damage. For example, it is natural to introduce a bond breaking mechanism to model the membrane rupture under high stress.

8.2.2 Parallel computing features for the code

One of the promising features of LBM is its capabilities of taking advantage of parallel computing. The local streaming and collision nature enables us to run the fluid solver based on geometric decomposition. There are some available open source codes such as graphic processing units (GPU) based Sailfish[225], and message passing interface (MPI) based Palabos[226]. Some trials have been carried out in graphic processing units (GPU) for the LBM part with a speed up of 30 times in Matlab. However, it still needs a significant work on implementing a GPU version of the FSI code. Multiphase fluid models can also be included in the code as well. Relative references can be found in Section 2.9.

Bibliography

[1] John D Anderson Jr. *A history of aerodynamics: and its impact on flying machines*. Vol. 8. Cambridge University Press, 1997.

[2] Hans J Lugt. "Vortex flow in nature and technology". In: *New York, Wiley-Interscience, 1983, 305 p. Translation.* 1 (1983).

[3] Michael Griebel, Thomas Dornseifer, and Tilman Neunhoeffer. *Numerical simulation in fluid dynamics: a practical introduction*. Vol. 3. Siam, 1997.

[4] Joe J Monaghan. "Smoothed particle hydrodynamics". In: *Annual review of astronomy and astrophysics* 30 (1992), pp. 543–574.

[5] Anatoly Malevanets and Raymond Kapral. "Mesoscopic model for solvent dynamics". In: *The Journal of chemical physics* 110.17 (1999), pp. 8605–8613.

[6] PJ Hoogerbrugge and JMVA Koelman. "Simulating microscopic hydrodynamic phenomena with dissipative particle dynamics". In: *EPL (Europhysics Letters)* 19.3 (1992), p. 155.

[7] YH Qian, Dominique d'Humières, and Pierre Lallemand. "Lattice BGK models for Navier-Stokes equation". In: *EPL (Europhysics Letters)* 17.6 (1992), p. 479.

[8] S Succi. *The Lattice Boltzmann Equation for Fluid Dynamics and Beyond, Clarendon*. 2001.

[9] Nathalie Lemonne et al. “Impaired blood rheology plays a role in the chronic disorders associated with sickle cell-hemoglobin C disease”. In: *Haematologica* 99.5 (2014), pp. 74–75.

[10] Gordon Lowe et al. “Blood rheology, cardiovascular risk factors, and cardiovascular disease: the West of Scotland Coronary Prevention Study”. In: *Thrombosis and haemostasis* 84.4 (2000), pp. 553–558.

[11] M Kaibara. “Rheology of blood coagulation”. In: *Biorheology* 33.2 (1996), pp. 101–117.

[12] C Pozrikidis. “Effect of membrane bending stiffness on the deformation of capsules in simple shear flow”. In: *Journal of Fluid Mechanics* 440 (2001), pp. 269–291.

[13] Timm Krüger et al. “Crossover from tumbling to tank-treading-like motion in dense simulated suspensions of red blood cells”. In: *Soft Matter* 9.37 (2013), pp. 9008–9015.

[14] Xiaohua Huang et al. “Cancer cell imaging and photothermal therapy in the near-infrared region by using gold nanorods”. In: *Journal of the American Chemical Society* 128.6 (2006), pp. 2115–2120.

[15] Rajesh Sardar and Jennifer S Shumaker-Parry. “Asymmetrically functionalized gold nanoparticles organized in one-dimensional chains”. In: *Nano letters* 8.2 (2008), pp. 731–736.

[16] Ji-Ho Park et al. “Systematic surface engineering of magnetic nanoworms for in vivo tumor targeting”. In: *Small* 5.6 (2009), pp. 694–700.

[17] Qiu Dai et al. “A “nanonecklace” synthesized from monofunctionalized gold nanoparticles”. In: *Journal of the American Chemical Society* 127.22 (2005), pp. 8008–8009.

[18] Yan Geng et al. “Shape effects of filaments versus spherical particles in flow and drug delivery”. In: *Nature Nanotechnology* 2.4 (2007), pp. 249–255.

[19] Yuan-cheng Fung. *Biomechanics: circulation*. Springer Science & Business Media, 1997.

[20] M Craig Miller, Gerald V Doyle, and Leon WMM Terstappen. “Significance of circulating tumor cells detected by the CellSearch system in patients with metastatic breast colorectal and prostate cancer”. In: *Journal of oncology* 2010 (2009).

[21] Shaolie S Hossain et al. “In silico vascular modeling for personalized nanoparticle delivery”. In: *Nanomedicine* 8.3 (2013), pp. 343–357.

[22] Ufuk Olgac et al. “Patient-specific three-dimensional simulation of LDL accumulation in a human left coronary artery in its healthy and atherosclerotic states”. In: *American Journal of Physiology-Heart and Circulatory Physiology* 296.6 (2009), H1969–H1982.

[23] D Kim Stangeby and C Ross Ethier. “Computational analysis of coupled blood-wall arterial LDL transport”. In: *Journal of biomechanical engineering* 124.1 (2002), pp. 1–8.

[24] Jifu Tan, Antony Thomas, and Yaling Liu. “Influence of red blood cells on nanoparticle targeted delivery in microcirculation”. In: *Soft matter* 8.6 (2012), pp. 1934–1946.

[25] Kathrin Müller, Dmitry A Fedosov, and Gerhard Gompper. “Margination of micro-and nano-particles in blood flow and its effect on drug delivery”. In: *Scientific reports* 4 (2014).

[26] Tae-Rin Lee et al. “On the near-wall accumulation of injectable particles in the microcirculation: smaller is not better”. In: *Scientific reports* 3 (2013).

[27] Renee Twombly. “Cancer surpasses heart disease as leading cause of death for all but the very elderly”. In: *Journal of the National Cancer Institute* 97.5 (2005), pp. 330–331.

[28] Klaus Pantel and Catherine Alix-Panabières. “Circulating tumour cells in cancer patients: challenges and perspectives”. In: *Trends in molecular medicine* 16.9 (2010), pp. 398–406.

[29] Sungyoung Choi et al. “Continuous blood cell separation by hydrophoretic filtration”. In: *Lab on a Chip* 7.11 (2007), pp. 1532–1538.

[30] Elodie Sollier et al. “Size-selective collection of circulating tumor cells using Vortex technology”. In: *Lab on a Chip* 14.1 (2014), pp. 63–77.

[31] Peng Chen et al. “Microscale Magnetic Field Modulation for Enhanced Capture and Distribution of Rare Circulating Tumor Cells”. In: *Scientific reports* 5 (2015).

[32] Kazunori Hoshino et al. “Microchip-based immunomagnetic detection of circulating tumor cells”. In: *Lab on a Chip* 11.20 (2011), pp. 3449–3457.

[33] Peter RC Gascoyne et al. “Isolation of rare cells from cell mixtures by dielectrophoresis”. In: *Electrophoresis* 30.8 (2009), pp. 1388–1398.

[34] Mark M Wang et al. “Microfluidic sorting of mammalian cells by optical force switching”. In: *Nature biotechnology* 23.1 (2005), pp. 83–87.

[35] Jinjie Shi et al. “Continuous particle separation in a microfluidic channel via standing surface acoustic waves (SSAW)”. In: *Lab on a Chip* 9.23 (2009), pp. 3354–3359.

[36] Tohid Fatanat Didar and Maryam Tabrizian. “Adhesion based detection, sorting and enrichment of cells in microfluidic Lab-on-Chip devices”. In: *Lab on a Chip* 10.22 (2010), pp. 3043–3053.

[37] Gonghao Wang et al. “Stiffness dependent separation of cells in a microfluidic device”. In: *PloS one* 8.10 (2013), e75901.

[38] Han Wei Hou et al. “Deformability study of breast cancer cells using microfluidics”. In: *Biomedical microdevices* 11.3 (2009), pp. 557–564.

[39] Denis Wirtz, Konstantinos Konstantopoulos, and Peter C Searson. “The physics of cancer: the role of physical interactions and mechanical forces in metastasis”. In: *Nature Reviews Cancer* 11.7 (2011), pp. 512–522.

[40] Andrew E Ekpenyong et al. “Viscoelastic properties of differentiating blood cells are fate-and function-dependent”. In: *PLoS One* 7.9 (2012), e45237.

[41] Jochen Guck et al. “Optical deformability as an inherent cell marker for testing malignant transformation and metastatic competence”. In: *Biophysical journal* 88.5 (2005), pp. 3689–3698.

[42] Sarah E Cross et al. “Nanomechanical analysis of cells from cancer patients”. In: *Nature nanotechnology* 2.12 (2007), pp. 780–783.

[43] Ju Li et al. “Cytoskeletal dynamics of human erythrocyte”. In: *Proceedings of the National Academy of Sciences* 104.12 (2007), pp. 4937–4942.

[44] Brian M Cooke, Narla Mohandas, and Ross L Coppel. “The malaria-infected red blood cell: structural and functional changes”. In: *Advances in parasitology* 50 (2001), pp. 1–86.

[45] Subra Suresh et al. “Connections between single-cell biomechanics and human disease states: gastrointestinal cancer and malaria”. In: *Acta biomaterialia* 1.1 (2005), pp. 15–30.

[46] Sangwon Byun et al. “Characterizing deformability and surface friction of cancer cells”. In: *Proceedings of the National Academy of Sciences* 110.19 (2013), pp. 7580–7585.

[47] B Hasslacher Y Pomeau and U Frisch. "Lattice-gas automata for the Navier-Stokes equation". In: *Phys. Rev. Lett* 56.14 (1986), p. 1505.

[48] Stephen Wolfram. "Cellular automaton fluids 1: Basic theory". In: *Journal of Statistical Physics* 45.3-4 (1986), pp. 471–526.

[49] Xiaoyi He and Li-Shi Luo. "Lattice Boltzmann model for the incompressible Navier–Stokes equation". In: *Journal of Statistical Physics* 88.3-4 (1997), pp. 927–944.

[50] Andrew K Gunstensen et al. "Lattice Boltzmann model of immiscible fluids". In: *Physical Review A* 43.8 (1991), p. 4320.

[51] Shiyi Chen and Gary D Doolen. "Lattice Boltzmann method for fluid flows". In: *Annual review of fluid mechanics* 30.1 (1998), pp. 329–364.

[52] Cyrus K Aidun and Jonathan R Clausen. "Lattice-Boltzmann method for complex flows". In: *Annual Review of Fluid Mechanics* 42 (2010), pp. 439–472.

[53] Michael C Sukop and Daniel T Thorne. *Lattice Boltzmann modeling: an introduction for geoscientists and engineers*. Springer, 2007.

[54] Anthony JC Ladd. "Numerical simulations of particulate suspensions via a discretized Boltzmann equation. Part 1. Theoretical foundation". In: *Journal of Fluid Mechanics* 271 (1994), pp. 285–309.

[55] Anthony JC Ladd. "Numerical simulations of particulate suspensions via a discretized Boltzmann equation. Part 2. Numerical results". In: *Journal of Fluid Mechanics* 271 (1994), pp. 311–339.

[56] Xiaoyi He, Shiyi Chen, and Raoyang Zhang. "A lattice Boltzmann scheme for incompressible multiphase flow and its application in simulation of Rayleigh–Taylor instability". In: *Journal of Computational Physics* 152.2 (1999), pp. 642–663.

[57] Xiaoyi He, Shiyi Chen, and Gary D Doolen. “A novel thermal model for the lattice Boltzmann method in incompressible limit”. In: *Journal of Computational Physics* 146.1 (1998), pp. 282–300.

[58] Stewart Harris. *An introduction to the theory of the Boltzmann equation*. Courier Dover Publications, 2004.

[59] Jonas Latt. “Hydrodynamic limit of lattice Boltzmann equations”. PhD thesis. University of Geneva, 2007.

[60] Erlend Magnus Viggen. “The lattice Boltzmann method with applications in acoustics”. In: *Master’s Thesis, NTNU* (2009).

[61] Chen Peng. “The Lattice Boltzmann Method for Fluid Dynamics: Theory and Applications”. In: *M. Math, Department of Mathematics, Ecole Polytechnique Federale de Lausanne* (2011).

[62] Luo Li-shi. “The lattice-gas and lattice Boltzmann methods: past, present, and future”. In: (2000).

[63] Jonas Latt. *Technical report: How to implement your DdQq dynamics with only q variables per node (instead of 2q)*. Tech. rep. Technical report, Tufts University, 2007.

[64] Timm Krüger. *Computer simulation study of collective phenomena in dense suspensions of red blood cells under shear*. Springer, 2012.

[65] Jonas Latt and Bastien Chopard. “Lattice Boltzmann method with regularized pre-collision distribution functions”. In: *Mathematics and Computers in Simulation* 72.2 (2006), pp. 165–168.

[66] Jonas Latt et al. “Straight velocity boundaries in the lattice Boltzmann method”. In: *Physical Review E* 77.5 (2008), p. 056703.

[67] Dominique d'Humières. "Multiple–relaxation–time lattice Boltzmann models in three dimensions". In: *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* 360.1792 (2002), pp. 437–451.

[68] Pierre Lallemand and Li-Shi Luo. "Theory of the lattice Boltzmann method: Dispersion, dissipation, isotropy, Galilean invariance, and stability". In: *Physical Review E* 61.6 (2000), p. 6546.

[69] Jonas Latt. "Choice of units in lattice Boltzmann simulations". In: *Freely available online at http://lbmethod. org/_media/howtos: lbunits. pdf* (2008).

[70] Timm Krüger, Fathollah Varnik, and Dierk Raabe. "Shear stress in lattice Boltzmann simulations". In: *Physical Review E* 79.4 (2009), p. 046704.

[71] ME Cates et al. "Physical and computational scaling issues in lattice Boltzmann simulations of binary fluid mixtures". In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 363.1833 (2005), pp. 1917–1935.

[72] JT Padding and AA Louis. "Hydrodynamic interactions and Brownian forces in colloidal suspensions: Coarse-graining over time and length scales". In: *Physical Review E* 74.3 (2006), p. 031402.

[73] Qisu Zou and Xiaoyi He. "On pressure and velocity boundary conditions for the lattice Boltzmann BGK model". In: *Physics of Fluids (1994-present)* 9.6 (1997), pp. 1591–1598.

[74] Martin Hecht and Jens Harting. "Implementation of on-site velocity boundary conditions for D3Q19 lattice Boltzmann simulations". In: *Journal of Statistical Mechanics: Theory and Experiment* 2010.01 (2010), P01018.

[75] Bo-Nan Jiang, TL Lin, and Louis A Povinelli. "Large-scale computation of incompressible viscous flow by least-squares finite element method". In:

Computer Methods in Applied Mechanics and Engineering 114.3 (1994), pp. 213–231.

[76] Olga Filippova and Dieter Hänel. "Acceleration of lattice-BGK schemes with grid refinement". In: *Journal of Computational Physics* 165.2 (2000), pp. 407–427.

[77] Daniel Lagrava et al. "Advances in multi-domain lattice Boltzmann grid refinement". In: *Journal of Computational Physics* 231.14 (2012), pp. 4808–4822.

[78] Alexandre Dupuis and Bastien Chopard. "Theory and applications of an alternative lattice Boltzmann grid refinement algorithm". In: *Physical Review E* 67.6 (2003), p. 066707.

[79] Xiaowen Shan and Hudong Chen. "Lattice Boltzmann model for simulating flows with multiple phases and components". In: *Physical Review E* 47.3 (1993), p. 1815.

[80] Merlin Ngachin. "Simulation of rising bubbles dynamics using the lattice Boltzmann method". In: (2011).

[81] Haihu Liu et al. "Multiphase lattice Boltzmann simulations for porous media applications–a review". In: *arXiv preprint arXiv:1404.7523* (2014).

[82] R Allen and T Reis. "A lattice Boltzmann model for natural convection in cavities". In: *International Journal of Heat and Fluid Flow* (2013).

[83] Frank J Alexander, Shiyi Chen, and JD Sterling. "Lattice boltzmann thermohydrodynamics". In: *Physical Review E* 47.4 (1993), R2249.

[84] YH Qian. "Simulating thermohydrodynamics with lattice BGK models". In: *Journal of scientific computing* 8.3 (1993), pp. 231–242.

[85] Zhaoli Guo and TS Zhao. "A lattice Boltzmann model for convection heat transfer in porous media". In: *Numerical Heat Transfer, Part B* 47.2 (2005), pp. 157–177.

[86] Keijo Mattila et al. "An efficient swap algorithm for the lattice Boltzmann method". In: *Computer Physics Communications* 176.3 (2007), pp. 200–210.

[87] Keijo Mattila et al. "Comparison of implementations of the lattice-Boltzmann method". In: *Computers & Mathematics with Applications* 55.7 (2008), pp. 1514–1524.

[88] Sean Murphy. "Performance of Lattice Boltzmann kernels". In: *Master's thesis, University of Edinburgh, this document is available at http://www2. epcc. ed. ac. uk/msc/dissertations/dissertations-0405/0762240-9j-dissertation1* 1 (2005).

[89] Jonas Tölke. "Implementation of a Lattice Boltzmann kernel using the Compute Unified Device Architecture developed by nVIDIA". In: *Computing and Visualization in Science* 13.1 (2010), pp. 29–39.

[90] Massimo Bernaschi et al. "A flexible high-performance Lattice Boltzmann GPU code for the simulations of fluid flows in complex geometries". In: *Concurrency and Computation: Practice and Experience* 22.1 (2010), pp. 1–14.

[91] Wei Li et al. "Flow simulation with complex boundaries". In: *GPU Gems* 2 (2005), pp. 747–764.

[92] M Dao, J Li, and S Suresh. "Molecularly based analysis of deformation of spectrin network and human erythrocyte". In: *Materials Science and Engineering: C* 26.8 (2006), pp. 1232–1244.

[93] R Skalak et al. "Strain energy function of red blood cell membranes". In: *Biophysical Journal* 13.3 (1973), pp. 245–264.

[94] Timm Krüger, Fathollah Varnik, and Dierk Raabe. "Efficient and accurate simulations of deformable particles immersed in a fluid using a combined immersed boundary lattice Boltzmann finite element method". In: *Computers & Mathematics with Applications* 61.12 (2011), pp. 3485–3505.

[95] Y Sui et al. "Dynamic motion of red blood cells in simple shear flow". In: *Physics of Fluids (1994-present)* 20.11 (2008), p. 112106.

[96] Dmitry A Fedosov, Bruce Caswell, and George Em Karniadakis. "A multiscale red blood cell model with accurate mechanics, rheology, and dynamics". In: *Biophysical journal* 98.10 (2010), pp. 2215–2225.

[97] Ken-ichi Tsubota, Shigeo Wada, and Takami Yamaguchi. "Particle method for computer simulation of red blood cell motion in blood flow". In: *Computer methods and programs in biomedicine* 83.2 (2006), pp. 139–146.

[98] T Omori et al. "Comparison between spring network models and continuum constitutive laws: Application to the large deformation of a capsule in shear flow". In: *Physical Review E* 83.4 (2011), p. 041918.

[99] Tenghu Wu and James J Feng. "Simulation of malaria-infected red blood cells in microfluidic channels: Passage and blockage". In: *Biomicrofluidics* 7.4 (2013), p. 044115.

[100] Masanori Nakamura, Sadao Bessho, and Shigeo Wada. "Spring-network-based model of a red blood cell for simulating mesoscopic blood flow". In: *International journal for numerical methods in biomedical engineering* 29.1 (2013), pp. 114–128.

[101] Masanori Nakamura, Sadao Bessho, and Shigeo Wada. "Analysis of Red Blood Cell Deformation under Fast Shear Flow for Better Estimation of Hemolysis". In: *International journal for numerical methods in biomedical engineering* 30.1 (2014), pp. 42–54.

[102] Dmitry A Fedosov. *Multiscale modeling of blood flow and soft matter*. Brown University, 2010.

[103] Evan Evans and Yuan-Cheng Fung. "Improved measurements of the erythrocyte geometry". In: *Microvascular research* 4.4 (1972), pp. 335–347.

[104] Rajat Mittal and Gianluca Iaccarino. "Immersed boundary methods". In: *Annu. Rev. Fluid Mech.* 37 (2005), pp. 239–261.

[105] Timm Krueger, David Holmes, and Peter V Coveney. "Deformability-based red blood cell separation in deterministic lateral displacement devices—A simulation study". In: *Biomicrofluidics* 8.5 (2014), p. 054114.

[106] Cédric Chauvierre et al. "Novel polysaccharide-decorated poly (isobutyl cyanoacrylate) nanoparticles". In: *Pharmaceutical research* 20.11 (2003), pp. 1786–1793.

[107] Omid C Farokhzad and Robert Langer. "Nanomedicine: developing smarter therapeutic and diagnostic modalities". In: *Advanced drug delivery reviews* 58.14 (2006), pp. 1456–1459.

[108] Edith Mathiowitz et al. "Biologically erodable microspheres as potential oral drug delivery systems". In: *Nature* 386.6623 (1997), pp. 410–414.

[109] Norased Nasongkla et al. "Multifunctional polymeric micelles as cancer-targeted, MRI-ultrasensitive drug delivery systems". In: *Nano letters* 6.11 (2006), pp. 2427–2430.

[110] Celeste Roney et al. "Targeted nanoparticles for drug delivery through the blood–brain barrier for Alzheimer's disease". In: *Journal of Controlled Release* 108.2 (2005), pp. 193–214.

[111] Robby A Petros and Joseph M DeSimone. "Strategies in the design of nanoparticles for therapeutic applications". In: *Nature Reviews Drug Discovery* 9.8 (2010), pp. 615–627.

[112] Frank Alexis et al. “Factors affecting the clearance and biodistribution of polymeric nanoparticles”. In: *Molecular pharmaceutics* 5.4 (2008), pp. 505–515.

[113] Jin Wang et al. “More effective nanomedicines through particle design”. In: *Small* 7.14 (2011), pp. 1919–1931.

[114] Bianca CH Lutters et al. “Blocking endothelial adhesion molecules: a potential therapeutic strategy to combat atherogenesis”. In: *Current opinion in lipidology* 15.5 (2004), pp. 545–552.

[115] Rutherford Aris. “On the dispersion of a solute in a fluid flowing through a tube”. In: *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*. Vol. 235. 1200. The Royal Society. 1956, pp. 67–77.

[116] Geoffrey Taylor. “Dispersion of soluble matter in solvent flowing slowly through a tube”. In: *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*. Vol. 219. 1137. The Royal Society. 1953, pp. 186–203.

[117] M Dembo et al. “The reaction-limited kinetics of membrane-to-surface adhesion and detachment”. In: *Proceedings of the Royal Society of London. Series B. Biological Sciences* 234.1274 (1988), pp. 55–83.

[118] Cheng Dong et al. “Mechanics of leukocyte deformation and adhesion to endothelium in shear flow”. In: *Annals of biomedical engineering* 27.3 (1999), pp. 298–312.

[119] Hans Christian Öttinger. *Stochastic processes in polymeric fluids: tools and examples for developing simulation algorithms*. Springer Verlag, 1996.

[120] Pep Espanol and Patrick Warren. “Statistical mechanics of dissipative particle dynamics”. In: *EPL (Europhysics Letters)* 30.4 (1995), p. 191.

[121] Martin Hecht et al. “Simulation of claylike colloids”. In: *Physical Review E* 72.1 (2005), p. 011408.

[122] AC Brańka and DM Heyes. “Algorithms for Brownian dynamics simulation”. In: *Physical Review E* 58.2 (1998), p. 2611.

[123] R Radhakrishnan et al. “Temporal multiscale approach for nanocarrier motion with simultaneous adhesion and hydrodynamic interactions in targeted drug delivery”. In: *Journal of computational physics* 244 (2013), pp. 252–263.

[124] Pierre Bongrand. “Ligand-receptor interactions”. In: *Reports on Progress in Physics* 62.6 (1999), p. 921.

[125] Davis Yohanes Arifin, Lai Yeng Lee, and Chi-Hwa Wang. “Mathematical modeling and simulation of drug release from microspheres: implications to drug delivery systems”. In: *Advanced drug delivery reviews* 58.12 (2006), pp. 1274–1325.

[126] Andrew D Grief and Giles Richardson. “Mathematical modelling of magnetically targeted drug delivery”. In: *Journal of Magnetism and Magnetic Materials* 293.1 (2005), pp. 455–463.

[127] Samar Shah et al. “Modeling particle shape-dependent dynamics in nanomedicine”. In: *Journal of nanoscience and nanotechnology* 11.2 (2011), p. 919.

[128] Ralf W Glaser. “Antigen-antibody binding and mass transport by convection and diffusion to a surface: a two-dimensional computer model of binding and dissociation kinetics”. In: *Analytical biochemistry* 213.1 (1993), pp. 152–161.

[129] Hermann B Frieboes et al. “Nanotechnology in cancer drug therapy: a biocomputational approach”. In: *BioMEMS and biomedical nanotechnology*. Springer, 2006, pp. 435–460.

[130] George I Bell. "Models for the specific adhesion of cells to cells". In: *Science* 200.4342 (1978), pp. 618–627.

[131] EVAN A Evans. "Detailed mechanics of membrane-membrane adhesion and separation. I. Continuum of molecular cross-bridges." In: *Biophysical journal* 48.1 (1985), p. 175.

[132] Jered B Haun and Daniel A Hammer. "Quantifying nanoparticle adhesion mediated by specific molecular interactions". In: *Langmuir* 24.16 (2008), pp. 8821–8832.

[133] P Decuzzi and M Ferrari. "The adhesive strength of non-spherical particles mediated by specific interactions". In: *Biomaterials* 27.30 (2006), pp. 5307–5314.

[134] Balabhaskar Prabhakarpandian et al. "Synthetic microvascular networks for quantitative analysis of particle adhesion". In: *Biomedical microdevices* 10.4 (2008), pp. 585–595.

[135] Daniel A Hammer and Sachin M Apte. "Simulation of cell rolling and adhesion on surfaces in shear flow: general results and analysis of selectin-mediated neutrophil adhesion." In: *Biophysical Journal* 63.1 (1992), p. 35.

[136] GEORGE I Bell, MICAH Dembo, and PIERRE Bongrand. "Cell adhesion. Competition between nonspecific repulsion and specific bonding." In: *Biophysical journal* 45.6 (1984), p. 1051.

[137] Cindi Cozens-Roberts, Douglas A Lauffenburger, and John A Quinn. "Receptor-mediated cell attachment and detachment kinetics. I. Probabilistic model and analysis." In: *Biophysical journal* 58.4 (1990), p. 841.

[138] Jin Liu et al. "Computational model for nanocarrier binding to endothelium validated using in vivo, in vitro, and atomic force microscopy experiments". In: *Proceedings of the National Academy of Sciences* 107.38 (2010), pp. 16530–16535.

[139] NA N'dri, W Shyy, and R Tran-Son-Tay. "Computational modeling of cell adhesion and movement using a continuum-kinetics approach". In: *Biophysical journal* 85.4 (2003), pp. 2273–2286.

[140] Aaron L Fogelson and Robert D Guy. "Immersed-boundary-type models of intravascular platelet aggregation". In: *Computer methods in applied mechanics and engineering* 197.25 (2008), pp. 2087–2104.

[141] RW Hockney, SP Goel, and JW Eastwood. "Quiet high-resolution computer models of a plasma". In: *Journal of Computational Physics* 14.2 (1974), pp. 148–158.

[142] Steve Plimpton. "Fast parallel algorithms for short-range molecular dynamics". In: *Journal of computational physics* 117.1 (1995), pp. 1–19.

[143] Albert Einstein. *Investigations on the Theory of the Brownian Movement*. Courier Corporation, 1956.

[144] In:

[145] CW Hirt, Anthony A Amsden, and JL Cook. "An arbitrary Lagrangian-Eulerian computing method for all flow speeds". In: *Journal of Computational Physics* 14.3 (1974), pp. 227–253.

[146] Charles S Peskin. "Flow patterns around heart valves: a numerical method". In: *Journal of computational physics* 10.2 (1972), pp. 252–271.

[147] Charles S Peskin. "The immersed boundary method". In: *Acta numerica* 11 (2002), pp. 479–517.

[148] Charles S Peskin and David M McQueen. "A three-dimensional computational method for blood flow in the heart I. Immersed elastic fibers in a viscous incompressible fluid". In: *Journal of Computational Physics* 81.2 (1989), pp. 372–405.

[149] Gregory Herschlag and Laura Miller. "Reynolds number limits for jet propulsion: a numerical study of simplified jellyfish". In: *Journal of theoretical biology* 285.1 (2011), pp. 84–95.

[150] Junfeng Zhang, Paul C Johnson, and Aleksander S Popel. "Red blood cell aggregation and dissociation in shear flows simulated by lattice Boltzmann method". In: *Journal of biomechanics* 41.1 (2008), pp. 47–55.

[151] Yaling Liu and Wing Kam Liu. "Rheology of red blood cell aggregation by computer simulation". In: *Journal of Computational Physics* 220.1 (2006), pp. 139–154.

[152] Lindsay M Crowl and Aaron L Fogelson. "Computational model of whole blood exhibiting lateral platelet motion induced by red blood cells". In: *International journal for numerical methods in biomedical engineering* 26.3-4 (2010), pp. 471–487.

[153] Lucy Zhang et al. "Immersed finite element method". In: *Computer Methods in Applied Mechanics and Engineering* 193.21 (2004), pp. 2051–2067.

[154] Aaron L Fogelson and Charles S Peskin. "A fast numerical method for solving the three-dimensional Stokes' equations in the presence of suspended particles". In: *Journal of Computational Physics* 79.1 (1988), pp. 50–69.

[155] Zhi-Gang Feng and Efstathios E Michaelides. "The immersed boundary-lattice Boltzmann method for solving fluid–particles interaction problems". In: *Journal of Computational Physics* 195.2 (2004), pp. 602–628.

[156] Sudeshna Ghosh and John M Stockie. "Numerical simulations of particle sedimentation using the immersed boundary method". In: *arXiv preprint arXiv:1304.0804* (2013).

[157] M Sadeghi and Feng Liu. "Coupled fluid-structure simulation for turbomachinery blade rows". In: *43rd AIAA Aerospace Sciences Meeting and Exhibit*. 2005, pp. 2005–0018.

[158] Hirofumi Doi and Juan J Alonso. "Fluid/structure coupled aeroelastic computations for transonic flows in turbomachinery". In: *ASME Turbo Expo 2002: Power for Land, Sea, and Air*. American Society of Mechanical Engineers. 2002, pp. 787–794.

[159] Pinunta Rojratsirikul, Zhijin Wang, and Ismet Gursul. "Unsteady fluid–structure interactions of membrane airfoils at low Reynolds numbers". In: *Experiments in fluids* 46.5 (2009), pp. 859–872.

[160] YW Kwon. "Development of coupling technique for LBM and FEM for FSI application". In: *Engineering computations* 23.8 (2006), pp. 860–875.

[161] S Kollmannsberger et al. "Fixed-grid fluid–structure interaction in two dimensions based on a partitioned Lattice Boltzmann and p-FEM approach". In: *International journal for numerical methods in engineering* 79.7 (2009), pp. 817–845.

[162] Stuart R Blair. *Lattice Boltzmann Methods for Fluid Structure Interaction*. Tech. rep. DTIC Document, 2012.

[163] ROBERT M MacMECCAN et al. "Simulating deformable particle suspensions using a coupled lattice-Boltzmann and finite-element method". In: *Journal of Fluid Mechanics* 618 (2009), pp. 13–39.

[164] Sebastian Geller, Jonas Tölke, and Manfred Krafczyk. "Lattice-Boltzmann method on quadtree-type grids for fluid-structure interaction". In: *Fluid-Structure Interaction*. Springer, 2006, pp. 270–293.

[165] COMSOL Multiphysics. "COMSOL multiphysics user guide (Version 4.3 a)". In: *COMSOL, AB* (2012).

[166] Patrick Ahlrichs and Burkhard Dünweg. "Lattice-Boltzmann simulation of polymer-solvent systems". In: *International Journal of Modern Physics C* 9.08 (1998), pp. 1429–1438.

[167] Patrick Ahlrichs and Burkhard Dünweg. "Simulation of a single polymer chain in solution by combining lattice Boltzmann and molecular dynamics". In: *The Journal of chemical physics* 111.17 (1999), pp. 8225–8239.

[168] Chia Wei Hsu and Yeng-Long Chen. "Migration and fractionation of deformable particles in microchannel". In: *The Journal of chemical physics* 133.3 (2010), p. 034906.

[169] Burkhard Dünweg and Anthony JC Ladd. "Lattice Boltzmann simulations of soft matter systems". In: (2008).

[170] Wendy R Sanhai et al. "Seven challenges for nanomedicine". In: *Nature nanotechnology* 3.5 (2008), pp. 242–244.

[171] Jin Liu et al. "Multiscale modeling of functionalized nanocarriers in targeted drug delivery". In: *Current nanoscience* 7.5 (2011), p. 727.

[172] Nishit Doshi et al. "Flow and adhesion of drug carriers in blood vessels depend on their shape: a study using model synthetic microvascular networks". In: *Journal of Controlled Release* 146.2 (2010), pp. 196–200.

[173] Jifu Tan et al. "The influence of size, shape and vessel geometry on nanoparticle distribution". In: *Microfluidics and nanofluidics* 14.1-2 (2013), pp. 77–87.

[174] Jifu Tan et al. "Coupled particulate and continuum model for nanoparticle targeted delivery". In: *Computers & structures* 122 (2013), pp. 128–134.

[175] Yaling Liu, Samar Shah, and Jifu Tan. "Computational modeling of nanoparticle targeted drug delivery". In: *Reviews in Nanoscience and Nanotechnology* 1.1 (2012), pp. 66–83.

[176] Shunqiang Wang et al. "Computational modeling of magnetic nanoparticle targeting to stent surface under high gradient field". In: *Computational Mechanics* 53.3 (2014), pp. 403–412.

[177] Rakesh Guduru et al. “Magneto-electric nanoparticles to enable field-controlled high-specificity drug delivery to eradicate ovarian cancer cells”. In: *Scientific reports* 3 (2013).

[178] Daniel A Reasor Jr et al. “Determination of critical parameters in platelet margination”. In: *Annals of biomedical engineering* 41.2 (2013), pp. 238–249.

[179] Maryam Saadatmand et al. “Fluid particle diffusion through high-hematocrit blood flow within a capillary tube”. In: *Journal of biomechanics* 44.1 (2011), pp. 170–175.

[180] Hong Zhao, Eric SG Shaqfeh, and Vivek Narsimhan. “Shear-induced particle migration and margination in a cellular suspension”. In: *Physics of Fluids (1994-present)* 24.1 (2012), p. 011902.

[181] Eugene C Eckstein, Douglas G Bailey, and Ascher H Shapiro. “Self-diffusion of particles in shear flow of a suspension”. In: *Journal of Fluid Mechanics* 79.01 (1977), pp. 191–208.

[182] David Leighton and Andreas Acrivos. “The shear-induced migration of particles in concentrated suspensions”. In: *Journal of Fluid Mechanics* 181 (1987), pp. 415–439.

[183] Piet AMM Aarts et al. “Fluid shear as a possible mechanism for platelet diffusivity in flowing blood”. In: *Journal of biomechanics* 19.10 (1986), pp. 799–805.

[184] Francesco Gentile, Mauro Ferrari, and Paolo Decuzzi. “The transport of nanoparticles in blood vessels: The effect of vessel permeability and blood rheology”. In: *Annals of biomedical engineering* 36.2 (2008), pp. 254–261.

[185] Francesco Gentile, Paolo Decuzzi, et al. “Time dependent dispersion of nanoparticles in blood vessels”. In: *Journal of Biomedical Science and Engineering* 3.05 (2010), p. 517.

[186] L Crowl and Aaron L Fogelson. "Analysis of mechanisms for platelet near-wall excess under arterial blood flow conditions". In: *Journal of Fluid Mechanics* 676 (2011), pp. 348–375.

[187] Dmitry A Fedosov, Julia Fornleitner, and Gerhard Gompper. "Margination of white blood cells in microcapillary flow". In: *Physical review letters* 108.2 (2012), p. 028104.

[188] MJ Alvarez-Cubero et al. "Dormant Circulating Tumor Cells In Prostate Cancer: Biological, Clinical And Therapeutic Implications." In: *Current drug targets* (2015).

[189] M Lekka et al. "Elasticity of normal and cancerous human bladder cells studied by scanning force microscopy". In: *European Biophysics Journal* 28.4 (1999), pp. 312–316.

[190] Sarah M McFaul, Bill K Lin, and Hongshen Ma. "Cell separation based on size and deformability using microfluidic funnel ratchets". In: *Lab on a chip* 12.13 (2012), pp. 2369–2376.

[191] Igor Sokolov. "Atomic force microscopy in cancer cell research". In: *Cancer Nanotechnology* 1 (2007), pp. 1–17.

[192] Tatyana G Kuznetsova et al. "Atomic force microscopy probing of cell elasticity". In: *Micron* 38.8 (2007), pp. 824–833.

[193] Aidan P Thompson, Steven J Plimpton, and William Mattson. "General formulation of pressure and stress tensor for arbitrary many-body interaction potentials under periodic boundary conditions". In: *The Journal of chemical physics* 131.15 (2009), p. 154107.

[194] L Weiss et al. "Lethal deformation of cancer cells in the microcirculation: a potential rate regulator of hematogenous metastasis". In: *International journal of cancer* 50.1 (1992), pp. 103–107.

[195] Orna Cohen-Fix. "Cell biology: Import and nuclear size". In: *Nature* 468.7323 (2010), pp. 513–516.

[196] Michael D Huber and Larry Gerace. "The size-wise nucleus: nuclear volume control in eukaryotes". In: *The Journal of cell biology* 179.4 (2007), pp. 583–584.

[197] Paul Jorgensen et al. "The size of the nucleus increases as yeast cells grow". In: *Molecular biology of the cell* 18.9 (2007), pp. 3523–3532.

[198] Peter Friedl, Katarina Wolf, and Jan Lammerding. "Nuclear mechanics during cell migration". In: *Current opinion in cell biology* 23.1 (2011), pp. 55–64.

[199] Takamasa Harada et al. "Nuclear lamin stiffness is a barrier to 3D migration, but softness can limit survival". In: *The Journal of cell biology* 204.5 (2014), pp. 669–682.

[200] Zhifeng Zhang et al. "The effects of 3D channel geometry on CTC passing pressure–towards deformability-based cancer cell separation". In: *Lab on a Chip* 14.14 (2014), pp. 2576–2584.

[201] Dimitri J Mavriplis and Antony Jameson. "Multigrid solution of the Navier-Stokes equations on triangular meshes". In: *AIAA journal* 28.8 (1990), pp. 1415–1425.

[202] Thomas JR Hughes, John A Cottrell, and Yuri Bazilevs. "Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement". In: *Computer methods in applied mechanics and engineering* 194.39 (2005), pp. 4135–4195.

[203] Boyce E Griffith et al. "Parallel and adaptive simulation of cardiac fluid dynamics". In: *Advanced computational infrastructures for parallel and distributed adaptive applications* (2010), p. 105.

[204] Anthony JC Ladd. "Sedimentation of homogeneous suspensions of non-Brownian spheres". In: *Physics of Fluids (1994-present)* 9.3 (1997), pp. 491–499.

[205] AJC Ladd and R Verberg. "Lattice-Boltzmann simulations of particle-fluid suspensions". In: *Journal of Statistical Physics* 104.5-6 (2001), pp. 1191–1251.

[206] Waseem Asghar et al. "Electrical fingerprinting, 3D profiling and detection of tumor cells with solid-state micropores". In: *Lab on a Chip* 12.13 (2012), pp. 2345–2352.

[207] E Evans and W Rawicz. "Entropy-driven tension and bending elasticity in condensed-fluid membranes". In: *Physical Review Letters* 64.17 (1990), p. 2094.

[208] JT Schumacher et al. "System development for generating homogeneous cell suspensions and transporting them in microfluidic components". In: *Engineering in Life Sciences* 8.1 (2008), pp. 49–55.

[209] Amy L Throckmorton et al. "Numerical, hydraulic, and hemolytic evaluation of an intravascular axial flow blood pump to mechanically support Fontan patients". In: *Annals of biomedical engineering* 39.1 (2011), pp. 324–336.

[210] Paolo Castellini, Marcos Pinotti, and Lorenzo Scalise. "Particle image velocimetry for flow analysis in longitudinal planes across a mechanical artificial heart valve". In: *Artificial organs* 28.5 (2004), pp. 507–513.

[211] Kalyani Nair et al. "Characterization of cell viability during bioprinting processes". In: *Biotechnology journal* 4.8 (2009), pp. 1168–1177.

[212] Juntao Zhang et al. "ESTIMATION OF FLOW-INDUCED BLOOD DAMAGE IN BIOMEDICAL DEVICES". In: *ASAIO Journal* 52.2 (2006), 11A.

[213] Miquel Trias et al. “FDA’s Nozzle Numerical Simulation Challenge: Non-Newtonian Fluid Effects and Blood Damage”. In: *PloS one* 9.3 (2014), e92638.

[214] Patrick M McCarthy. “HeartMate implantable left ventricular assist device: bridge to transplantation and future applications”. In: *The Annals of thoracic surgery* 59 (1995), S46–S51.

[215] AS Al Jarallah et al. “The Hemopump as a left ventricular assist device in pediatric applications: initial Canadian applications.” In: *The Canadian journal of cardiology* 13.5 (1997), pp. 489–494.

[216] Hans-Dietrich Polaschegg. “Red blood cell damage from extracorporeal circulation in hemodialysis”. In: *Seminars in dialysis*. Vol. 22. 5. Wiley Online Library. 2009, pp. 524–531.

[217] Barbara Lorber et al. “Adult rat retinal ganglion cells and glia can be printed by piezoelectric inkjet printing”. In: *Biofabrication* 6.1 (2014), p. 015001.

[218] Karen Chang Yan, Kalyani Nair, and Wei Sun. “Three dimensional multi-scale modelling and analysis of cell damage in cell-encapsulated alginate constructs”. In: *Journal of biomechanics* 43.6 (2010), pp. 1031–1038.

[219] Jessica Hultström et al. “Proliferation and viability of adherent cells manipulated by standing-wave ultrasound in a microfluidic chip”. In: *Ultrasound in medicine & biology* 33.1 (2007), pp. 145–151.

[220] TH Ang et al. “Fabrication of 3D chitosan–hydroxyapatite scaffolds using a robotic dispensing system”. In: *Materials Science and Engineering: C* 20.1 (2002), pp. 35–42.

[221] Thomas Boland et al. “Cell and organ printing 2: Fusion of cell aggregates in three-dimensional gels”. In: *The Anatomical Record Part A: Discoveries*

in Molecular, Cellular, and Evolutionary Biology 272.2 (2003), pp. 497–502.

[222] RL Kormos et al. "Rheologic abnormalities in patients with the Jarvik-7 total artificial heart." In: *ASAIO Journal* 33.3 (1987), pp. 413–417.

[223] Ting-Cheng Hung et al. "Effects of long-term Novacor artificial heart support on blood rheology." In: *ASAIO transactions/American Society for Artificial Internal Organs* 37.3 (1990), pp. M312–3.

[224] Russell P Rother et al. "The clinical sequelae of intravascular hemolysis and extracellular plasma hemoglobin: a novel mechanism of human disease". In: *Jama* 293.13 (2005), pp. 1653–1662.

[225] Michal Januszewski and Marcin Kostur. "Sailfish: a flexible multi-GPU implementation of the lattice Boltzmann method". In: *Computer Physics Communications* 185.9 (2014), pp. 2350–2368.

[226] J Latt. *Palabos, parallel lattice Boltzmann solver*. 2009.

Appendix A

Cell Linked-List algorithm

```
1  void Chain::initLJ(){
     if (lx==0 || ly==0)
3      std::cout<<"particle doesn't know fluid geometry size"<<std::endl
       ;
     cx = floor(lx/rCut);
5    cy = floor(ly/rCut);
     std::cout<<"cx,cy "<<cx<<" "<<cy<<std::endl;
7    lscl=new int[nn];
     head=new int[cx*cy];
9    if (pCell){
       lscl_c=new int[pCell->nn];
11     head_c=new int[cx*cy];
     }
13   rrCut=rCut*rCut;
     sig2=sigma*sigma;
15   sig6=sig2*sig2*sig2;
   }
17
   void Chain::buildLinkList(){
19   int cellSize=cx*cy;
     int idx,idy,idc;
21   double dx,dy;
```

```
    dx = double(lx)/double(cx);
23  dy = double(ly)/double(cy);
    for (int i=0;i<cellSize;i++){
25    head[i]=EMPTY;
      if (pCell) head_c[i]=EMPTY;
27  }
    //particles
29  for (int i=0;i<nn;i++){
      idx = floor(x[2*i]/dx);
31    idy = floor(x[2*i+1]/dy);
      idc = idy*cx + idx;
33    lscl[i]=head[idc];
      head[idc]=i;
35  }
    // cell structure nodes
37  if(pCell){
    for (int i=0;i<pCell->nn;i++){
39    idx = floor(pCell->x[2*i]/dx);
      idy = floor(pCell->x[2*i+1]/dy);
41    idc = idy*cx + idx;
      lscl_c[i]=head_c[idc];
43    head_c[idc]=i;
      if (idc >cx*cy) std::cout<<"cell idc "<<idc<<" "<<cx*cy<<std::
     endl;
45  }
    }
47}

49void Chain::pairWiseInteraction(){
    int idc,idc_nb;
51  int i,j,jc;
    // scan all the cells through x, and y direction
53  for (int idcy=0;idcy<cy;idcy++){
      for (int idcx=0;idcx<cx;idcx++){
```

```
55      idc = idcy*cx + idcx;
        if (head[idc]==EMPTY) continue;
57      //scan neighbor cells
        for (int nby=idcy-1;nby<=idcy+1;nby++){
59        for (int nbx=idcx-1;nbx<=idcx+1;nbx++){
            //when cell is on the edge, -1 could lead to out of
    boundary
61          if (periodicX && periodicY){
              idc_nb = ((nby+cy)%cy)*cx+((nbx+cx)%cx);
63          }else if (periodicX){
              if (nby== -1 || nby == cy)
65              continue;
              else
67              idc_nb = nby*cx + ((nbx+cx)%cx);
            }else if (periodicY){
69            if (nbx == -1 || nbx == cx)
                continue;
71            else
                idc_nb = ((nby+cy)%cy)*cx + nbx;
73          }else{
              if (nby== -1 || nby == cy)
75              continue;
              if (nbx == -1 || nbx == cx)
77              continue;
              idc_nb = nby*cx + nbx;
79          }
            if (head[idc_nb] !=EMPTY){
81            i=head[idc];
              while(i!=EMPTY){
83              j=head[idc_nb];
                  while(j!=EMPTY){
85                  if (i<j) LJForce(i,j);
                    j=lscl[j];
87                }//end of loop j
```

```
                i=lscl[i];
89            }
            }
91          if(pCell){
            if (head_c[idc_nb] !=EMPTY){
93            i=head[idc];
              while(i!=EMPTY){
95              jc=head_c[idc_nb];
                  while(jc!=EMPTY){
97                  particleCellLJForce(i,jc);
                    jc=lscl_c[jc];
99                }//end of loop j
                i=lscl[i];
101           }
            }
103         }
          }//eof nbx
105     }//eof nby
      }//eof idcx
107 }//eof idcy
}
```

Appendix B

Vita

Jifu Tan was born on November, 21, 1986 at Chizhou, Anhui Province of China. He is the elder son of Chuanming Tan and Pingying Pan. He has a young brother Jigui Tan. He received his Bachelor Degree in Civil Engineering in 2007 and Master Degree in Mechanical engineering from Lehigh University at 2012. He has published 10 papers as a graduate student. He is a Rossin Doctor Fellow in Lehigh University. He has a lot of teaching experience in numerical methods in engineering.