國立交通大學

應用數學系

博士論文

沉浸邊界法在囊泡問題之數值模擬

Immersed Boundary Methods for Simulating
Vesicle Dynamics

研究生：胡偉帆

指導教授：賴明治 教授

沉浸邊界法在囊泡問題之數值模擬
Immersed boundary methods for simulating vesicle dynamics

研　究　生：胡偉帆　　　　Student：Wei-Fan Hu

指導教授：賴明治　教授　　　　Advisor：Ming-Chih Lai

國 立 交 通 大 學
應 用 數 學 系
博 士 論 文

A Dissertation
Submitted to Department of Applied Mathematics
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy
in

Applied Mathematics

July 2014

Hsinchu, Taiwan, Republic of China

中華民國一百零三年七月

沉浸邊界法在囊泡問題之數值模擬

研究生：胡偉帆　　　　　　　　　　　　　　　　指導教授：賴明治 教授

國立交通大學
應用數學系 博士班

摘　　要

過去數十年來，囊泡問題的動態數值問題一直是個熱門的議題。在本論文中，我們透過沉浸邊界法 (immersed boundary method)來描述囊泡的流體數學模型，公式，包含 Eulerian 座標系下的流體方程式以及建立在 Lagrangian 座標系中有關界面的變數，而這兩個座標系之間各個變數的轉換，則是藉由 Dirac delta function 來連結。本論文致力於發展簡易且精確的數值方法來模擬此流體界面問題。

首先，我們提出了一個 fractional step immersed boundary method 用於模擬不可延展界面之問題(不考慮 bending effect)。我們證明了作用在表面張力 spreading operator 是 surface divergence operator 的斜自伴算子 (skew-adjoint)。利用這個特性，對於流體變數之離散我們可獲得一個對稱矩陣，並可使用 fractional step 方法解決此線性系統。我們比較了此數值方法的精確度，以及利用本方法研讀不可延展界面在 shear flow 底下之數值模擬。

再來我們研發出一種 unconditionally stable immersed boundary method 作為二維的囊泡在 Navier-Stokes 流體之模擬。我們用一種半隱示 (semi-implicit)的方法來表示囊泡之界面力，而與介面力相關的 stretching factor 則可透過其他的方程式獲得。我們證明出在本方法中，流體系統的總能量將隨時間而遞減。另外，利用 projection method，對於流場我們推導出一個對稱正定的線性系統，此矩陣可利用多重網格法 (multi-grid method)有效率地計算其解。在數值實驗中，我們驗證本方法之精準度，並在效率上遠勝於傳統之顯示邊界力處裡方法。同樣我們也利用本方法研讀囊泡在二維座標底下之型變動態系統。

最後，我們延伸至三維軸對稱的囊泡問題。與其將表面張力作為一 Lagrange's multiplier 來迫使囊泡之不可延展性，我們定義一種 spring-like 表面張力作為本模型之逼近。囊泡的邊界可利用 Fourier spectral 方法來表示，並且我們可精準地計算出界面上的平均曲率、高斯曲率等等。透過一系列之數值模擬，我們展示出本方法之應用性與可靠性。我們使用本方法研讀囊泡在靜止流、重力場影響及 Poiseuille flow 下之動態型變問題。

誌　　　謝

很幸運的我終於畢業了，說到寫致謝文這檔事每次都讓我頭痛不已，特別不想寫些似流水帳類的文章。不過自詡小文青的我，真正撰寫時，卻突然變得好像失憶般地不知道該寫些什麼了。

當然我不會忘記首先該感謝的人是我的指導教授賴明治大教授，遙想起還在頭髮茂密的碩士班年代，懵懂無知的我修了一堂賴老大開的應用數學課程，當時真覺得原來我所學的數學可以這麼好玩、這麼應用，於是乎在抉擇研究領域上自然往應用數學靠上不少。對於把數值計算跟程式撰寫完全留在大學的我一開始是很慌張的，只好每天跑到系上的圖書館埋頭苦幹（兼吹冷氣），當我靠自己的能力漸漸補足科學計算的基本常識後終於能開始做些研究，這種滿足感是我在人生中少有的。當然，在研究的過程中痛苦時常是大於收穫的，也不知怎麼的，我好像愛上了這種痛苦，不，應該說愛上解決痛苦問題後的收穫吧，我便毅然決然選擇學術圈作為我往後吃飯糊口的地方。

在賴老大的耳濡目染下，我逐漸地能夠掌握到研究正確的方向 （雖然常常有時候偷懶想跳過某些步驟時都一定會被抓包，一次次的被修理中我也慢慢地建立做學問該有的嚴謹態度）。從一開始的在台上演講給寥寥無幾個同學都會雙腿發抖的我，到現在可以從容自在地回應聽眾的問題，這種經驗可說是非常難得可貴的。我很享受在這種被訓練及磨練下能夠逐漸成為一個獨立思考的個體，甚至廢寢忘食也只為了解決某個研究上遇到的問題（記得我的第二篇文章主要的概念還是在夢中想到的）。關於感謝賴老大的詞我想也不必多想了，努力工作吧！

在四年半的修業過程中，我得感謝許多系上的老師們。依稀記得來到交大第一堂課程就是林琦焜老師的偏微分方程，著實給我了個震撼教育（不只在數學方面，在個人方面特別要教導我們該擺脫過往的洗腦教育，關心社會以及土地）。也感謝蔡武廷老師（現任台大）讓我對於流體方程從零開始建立基礎，幽默逗趣的上課過程讓學習更輕鬆（我記得蔡老師很喜歡用一個詞—機車）。當然我必須感謝一起合作的老師們，特別謝謝林文偉老師幫助我解決困難的矩陣問題，讓我的第一篇論文得以發表。感謝我的口試委員老師們，對本論文的指教及提供了許多有關接下來要作的研究的意見。

千里馬計畫是我人生中一個很特別的經驗，本來一直認為自己會到美國東岸或西岸去，沒想到最後到了法國南邊的一個小鎮。在法文只能出口幾句日常問好的狀況下我學習獨立生

活，但也見識到歐洲人是如何熱愛自己的生命（工作時間為早上十點至下午五點，中午還要喝個二小時的咖啡），快速且全面的文化衝擊讓我見識到世界的廣大。我要感謝傳立葉大學物理系的 Misbah 教授，在他的熱情招待下我有個舒適的環境能夠安身立命做研究，也讓我見識到了物理學家們頗析問題的方法及精闢。在法國我認識了交大的學姊 Sharon，不但人美心地善良也很熱心，幫助我在異鄉搞定許多的生活大小事，就好像我另一個親姊。

最後我要感謝我的家人以及多年來陪伴的好朋友們。謝謝我的父母永遠當我的依靠，長久以來包容我的懶散、沒耐心。感謝我親愛的朋友們 Howard、仲尹、Nelson、Jeny、麻將、曾鈺傑、縉祥、叮叮等等，時常一起打球出門玩樂、分享喜怒哀樂的夥伴們。最後我一定要說是何等的幸運，能夠在交大遇到我女朋友，謝謝妳時常帶給我歡笑，讓我能精力飽滿的度過每一天。

Acknowledgment

First of all, I give an immense gratitude to my advisor, Professor Ming-Chih Lai, who has given me invaluable insights, advices and encouragements. His great knowledge, infinite ideas stimulate me to immerse myself in research and lead me to dig more potential from my brain. Choosing him as my advisor is the best decision I have made, I always believe that all the time. I am not sure that I am capable of showing my gratefulness in words, but I guess the best way to express my thanks is - working hard for him.

I would like to express my gratitude to all the professors whom I took classes from at NCTU, and special thanks to Professor C. K. Lin, who has broadened my view not only extracting knowledge in mathematics but also paying more attention for the politics and our homeland. I extend my thanks to Professor Chaouqi Misbah for hosting me in Joseph Fourier University in Grenoble, also I want to thank friends who made my life colorful in France (Sharon, Nicola, Yara and Zaiyi). Thanks for the supports from Ministry of Science and Technology of Taiwan. In addition, I appreciate the defense committees for their stimulating questions and helpful suggestions to complete this dissertation.

Next I want to express my thanks to Howard (maybe I should call him professor Tseng) and Ken, you two always give me a lot of suggestions not merely in researching but in my life. Thank you my brothers. I also want to thank each person I met in NCTU and all my dear friends, Nelson, Jeny, Yu-Jei, Chin-Hsiang, DingDing etc.

Finally, I deeply appreciate my family, my parents and my sister. They always hold my back and give me endless supports. Thanks for your accompanies and let me get through the hardest time in my life. I left my last thank to my girlfriend YY Lu, who makes me happy and enrich my life everyday, and I believe happiness is the best method for doing research, thank you.

Abstract

Numerical simulation of vesicle dynamics has been a popular issue for many decades. In this dissertation, a mathematical formulation for suspension of vesicle in fluid is modeled by immersed boundary method, where a mixture of Eulerian fluid variables and curve-linear Lagrangian interfacial variables are used, and the linkage between these two variables is a smoothed Dirac delta function. The purpose of this dissertation is to develop accurate and efficient numerical schemes for simulating vesicle dynamics through immersed boundary method.

Firstly, we propose a fractional step immersed boundary method to mimic dynamical system of an inextensible interface (vesicle without bending effect). In addition to solving for the fluid variables such as the velocity and pressure, the present problem involves finding an extra unknown elastic tension such that the surface divergence of the velocity is zero along the interface. By taking advantage of skew-adjoint property between force spreading operator and surface divergence operator, the resultant linear system of equations is symmetric and can be solved by fractional steps so that only fast Poisson solvers are involved. The convergent tests for present fluid solver is performed and confirm the desired accuracy. The tank-treading motion for an inextensible interface under a simple shear flow has been studied extensively, and the results are in good agreement with those obtained in literature. This part of work has been published in SIAM Journal of Scientific Computing as in [37].

Secondly, we develop an unconditionally stable immersed boundary method to simulate 2D vesicle under a Navier-Stokes flow. We adopt a semi-implicit boundary forcing approach, where the stretching factor used in the forcing term can be computed from the derived evolutional equation. By using the projection method to solve the fluid equations, the pressure is decoupled and we have a symmetric positive definite system that can be solved efficiently. The method can be shown to be unconditionally stable, in the sense that the total energy of fluid system is decreasing. A resulting modification benefits from this improved numerical stability, as the time step size can be significantly increased. The numerical result shows the severe time step restriction in an explicit boundary forcing scheme is avoided by present method. The part of work has been published in East Asian Journal of Applied Mathematics as in [22].

Lastly, we extend to simulate three-dimensional axisymmetric vesicle suspended in a Navier-Stokes flow. Instead of introducing a Lagrange's multiplier to enforce the vesicle inextensibility constraint, we modify the model by adopting a spring-like tension to make the vesicle boundary nearly inextensible so that solving for the unknown tension can be avoided. We also derive a new elastic force from the modified vesicle energy and obtain exactly the same form as the originally unmodified one. In order to represent the vesicle boundary, we use Fourier spectral approximation so we can compute the geometrical quantities on the interface more accurately. A series of numerical tests on the present schemes have been conducted to illustrate the applicability and reliability of the method. We perform the convergence check for fluid variables for present schemes. Then we study the vesicle dynamics in quiescent flow, Poiseuille and under influence of gravity in detail. The numerical results are shown to be in good agreement with those obtained in literature. The part of work has been published in Journal of Computational Physics as in [23].

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Blood flow is a continuous circulation of blood in cardiovascular system. The main cell in blood is red blood cell, which plays an important role in transporting system in human body. Especially, vesicles share similar features with red blood cells and thus its model can be viewed as a simplification of red blood cell. Vesicle dynamics has been widely investigated in large amount of literatures. The long turn of present work is to understand the mechanism of red blood cells, and the short turn is to fully study the hydrodynamical system of vesicle.

1.1 Blood flow and vesicles

Blood flow has been at the center of the attention of scientists since the 19^{th} century. Blood is circulated around the body through blood vessels by the pumping action of the heart. Blood flow is a bodily flow that transport essential substances such as oxygen and nutrients to the cells and take away metabolic wastes from the same cells. In vertebrates, blood is composed by blood plasma, platelets, red blood cells and white blood cells. The main part of blood is plasma, which is a Newtonian fluid and constitutes around 55% of blood fluid (the large part of it is water). The rest 45% is red blood cells , while white blood cells and platelets only occupy less than 1% in blood. The white blood cells are the cells of the immune system that are involved in defending the body against both infectious disease and foreign materials; the main function of platelets is to stop bleeding. A scanning electron microscope image of a normal red blood cell, a platelet, and a white blood cell is shown in Figure 1.1.

Red blood cells (RBCs), also called erythrocytes, delivering oxygen (O_2) to the body tissues via the blood flow through the circulatory system. They take up oxygen in the lungs or gills and release it by diffusing thorough membrane of RBC into tissues while squeezing through the body's vessels. RBCs are also able to transport 2% carbon dioxide (CO_2) away from organisms while the major part of CO_2 are transported by plasma. The membrane of RBC forms a lipid bilayer, which is constructed

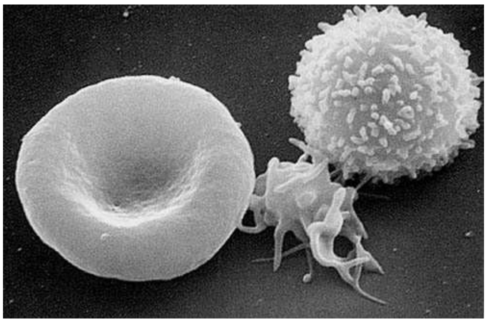


Figure 1.1: The cells in blood. From left to right: red blood cell, platelet and white blood cell.

by a cytoskeleton network embedding different kinds of proteins, as sketched in Figure. 1.2. The proteins of the membrane skeleton are responsible for the deformability, flexibility and durability of the red blood cell. In general, a healthy RBC exhibits in biconcave shape (see Figure. 1.1) to accommodate maximum carry of oxygen; the shape of RBC is flexible, especially it can highly deformed while flowing thorough tiny capillaries with blood.

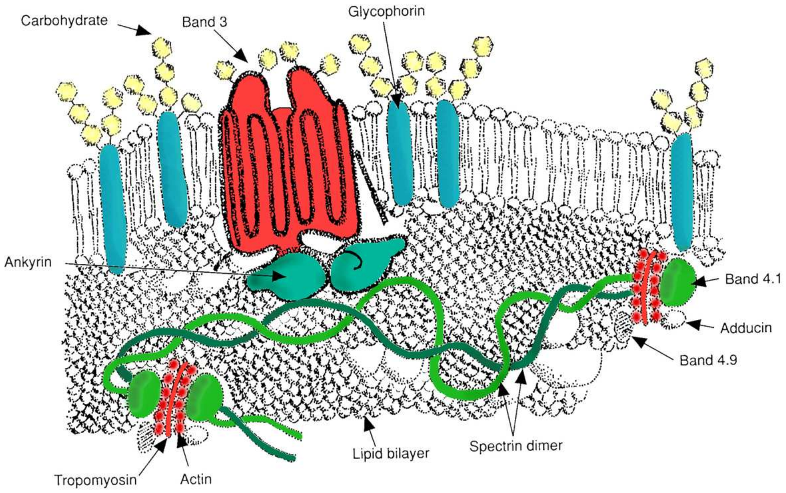


Figure 1.2: A cartoon showing major proteins on the membrane of RBC.

As mentioned in previous paragraph, the model of RBCs is quite complex due to variety features on its cell membrane. In order to grasp the fundamentals of blood flow and link them to the properties of RBCs, we consider a model system of *vesicles*. Vesicles are fluidic drops which is similar to the one surrounding living cells without cytoskeleton network structures (compare to RBCs). The most feature of vesicle is that it can be viewed as an approximation model of RBC. The membrane of vesicles has similar properties to the membrane of RBCs, resisting strongly surface dilatation and exhibits a resistance against bending. Despite the simplicity structure of vesicle, many features observed by RBCs were also observed by vesicles. The mechanical details of vesicles will be introduced later in Chapter 2.

1.2 Existing works

For the past years, the vesicle problems have been extensively explored by the classic [24] and small deformation theories [40, 32, 12], flow experiments [26, 10], and computer simulations [25, 3, 45, 66, 67, 28, 4, 60, 68, 72]; just to name a few recent ones. Readers who are interested in more details on those relevant works can refer to the recent article [72].

The study of vesicle dynamics is a fluid-structure interaction problem. The numerical simulation of the problem consists of not only solving a two-phase flow but also requiring to enforce an inextensibility constraint along the surface which makes the problem challenging. Among the numerical methods for simulating vesicle problems in literature, one can characterize those methods by how the membrane surface (or interface) is represented and what the fluid solver is used. Based on this characterization, several methods have been developed such as boundary integral method [25, 66, 67, 4, 68, 72], level set method [41, 60, 36, 43, 11], phase field method [9, 3, 43], particle collision method [45], immersed interface method [35, 65], and immersed boundary method [28, 37] et. al.

1.3 Immersed boundary method

The immersed boundary (IB) method proposed by Peskin [48] has been successfully applied to many fluid-structure interaction problems, see the review [50]. The IB formulation employs an Eulerian description for the fluid velocity and the pressure, and Lagrangian description for the configuration of the immersed elastic structure (immersed boundary or interface). The immersed structure exerts some force into the fluid that drives the fluid flow and at the same time the fluid flow carries the immersed structure to a new configuration. This interaction between the fluid and the immersed structure is linked through a force spreading and velocity interpolating operator by the usage of smoothed version of Dirac delta function [50]. This method is easy to implement and efficient simply because the immersed structure (no matter how complex) is regarded as a force generator to the fluid so the fluid variables can be solved in a fixed Eulerian domain without generating any structure-fitting grid. Therefore, many fast efficient fluid solvers can be applied through the framework of IB method. We will introduce the numerical part of IB method in detail in Chapter 3.

1.4 Contribution of the present work

In this dissertation, we focus on developing numerical method to mimic vesicle rheology dynamics through IB method. For two-dimensional space, we propose a fractional step IB method and unconditionally energy stable method, then we extend to three-dimensional axial symmetric coordinate by using a nearly inextensible approach. These works are based on our previous work [37, 22, 23]. The brief descriptions of numerical method are illustrated as below.

- **Fractional step immersed boundary method**. The governing equation is an incompressible Stokes equation with a suspension of inextensible interface (that is, the bending rigidity is neglected). We found that the spreading operator of force generator has skew-adjoint property to surface divergence operator both in theoretical continuous and numerical discrete version. By taking advantage of this crucial property, the time-stepping numerical discretization involves solving a symmetric positive semi-definite matrix, and this linear system can be done efficiently by employing a preconditioned conjugate gradient method. We successfully reproduce vesicle-like dynamics such as tank-treading motion under shear flow. As a benchmark, we also measure the inclination angle and tank-treading frequency when reaching steady state. The numerical result shows a highly agreement with those in many other literatures.

- **Unconditionally energy stable immersed boundary method**. The fluid equation is governed by a Navier-Stokes equation with suspension of vesicle. Rather than enforcing purely inextensible constrain, we adopt a nearly inextensible approach to simulate vesicle dynamics. The fluid equations are discretized by projection method so the pressure is decoupled. Again by using skew-adjoint property, it leads to solve a symmetric positive sparse linear system respect to intermediate fluid variable, this can be done efficiently by aggregation-based multigrid. Meanwhile, we prove that the developed scheme shows an unconditional stability in energy sense, which means the total energy of fluid system decreases during time integration. This is a significant step to release restriction of numerical time step due to applying the penalized method. Again we successfully simulated vesicle dynamics, such as a free relaxation to equilibrium state, tank-treading motion under shear flow, the numerical cost is substantial saved by this scheme.

- **Simulating three-dimensional axisymmetric vesicles**. We extend previous work to three-dimensional space to link the real world. An axisymmetric vesicle is suspended in incompressible Navier-Stokes flow in three-dimensional capillary. Again the nearly inextensible approximation in axisymmetric version is applied. We have shown that the modified interfacial force due to nearly inextensible approaching has exactly the same form to the original one. Besides, we adopt a high accuracy spectral method to evaluate interfacial quantities such as mean curvature, Gaussian curvature and surface Laplace of mean curvature. The fluid variables are solved numerically by traditional projection method in which fast Poisson solver can be applied to obtain fluid velocity filed and pressure increment. We investigated behaviors of axisymmetric vesicle in situation of freely suspended, under influence of gravity field and passing through capillary in Poiseuille flow. The numerical result shows a good agreement with what observed in experiment, and this is an evidence showing the reliability of our proposed scheme.

Chapter 2

Mathematical model of vesicle hydrodynamics

The first model used for vesicles can be tracked back to Helfrich (1973) [18], who investigated the rheology dynamics of vesicle through a elasticity theory. In this chapter, we introduce more details about vesicle, such as vesicle mechanism, the vesicle surface energy and the the boundary forces on vesicle membrane. The mathematical formulation of hydrodynamical system for suspended vesicle is given.

2.1 Vesicle mechanics

A vesicle can be visualized as a liquid droplet within another liquid enclosed by a lipid membrane with the size about 100 nm to 10 μm. Such lipid membrane consists of tightly packed lipid molecules with hydrophilic heads facing the exterior and interior fluids and with hydrophobic tails hiding in the middle and thus forms a bilayer phospholipid with thickness about 6 nm (see Figure 2.1).

Vesicle is widely used for mimicking model of biological cells and also gives many features observed by living cells. The shape of vesicle is deformable, in equilibrium state, the vesicle usual appears in shape of biconcave disk because of minimized surface energy (see Figure 2.2). Topological changes (fusion, division, etc.) of vesicle are possible but rarely happened in real world. In principle, the vesicle membrane mainly undergoes two basic deformation factors, which are *inextensibility* and *bending effect*.

At usual room temperature, since the phospholipidic molecules do not bind with each other, the membrane is thus regard as a liquid phase. In this scenario, phospholipids stay very close and tend to approximate a constant density, it would lead to local conservation of vesicle surface. Therefore, the membrane is considered as an inextensible surface. Moreover, because the permeability of vesicle membrane is very small and fluid inside vesicle is Newtonian incompressible fluid, this contributes to

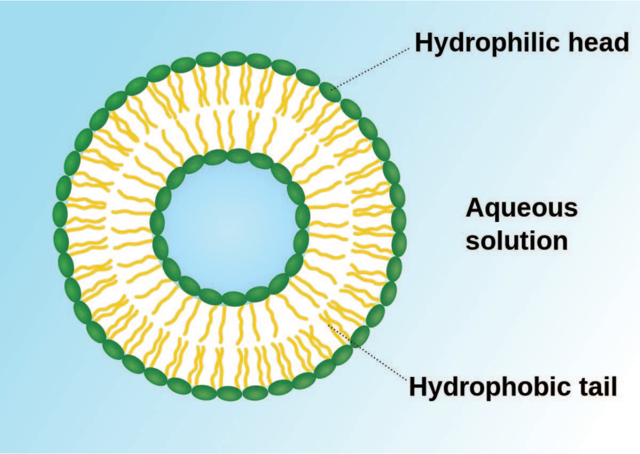


Figure 2.1: A cartoon showing a vesicle and its molecular structure on membrane. Picture is taken from the web site of Wikipedia.

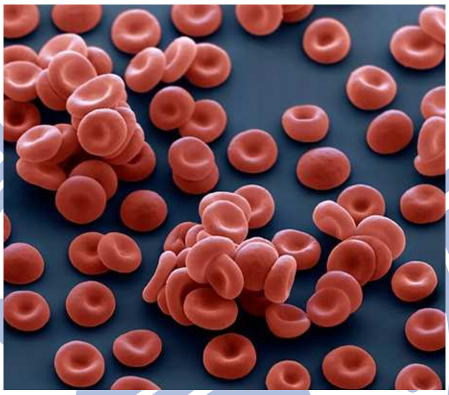


Figure 2.2: Biconcave equilibrium state of vesicles.

total volume of vesicle is conserved as well. The phospholipid membrane is known to exhibit a resistance against bending, we have to take bending effect into account in our mathematical formulation.

In consequence, we list three features of vesicle as below.

- The total surface area of vesicle is conserved because of the inextensibility. This is the intrinsic feature of vesicle which is regarded as a simplified model of red blood cell. The mathematical formulation of inextensibility will be explained later in Section 2.3.

- The total volume of vesicle remains a constant since it contains incompressible Newtonian fluid. The mathematical statement is shown in Section 2.4.

- The bending effect should be considered in modeling vesicle dynamics. The explicit form of bending force is given in Section 2.2.

When it comes to vesicle dynamics, we must mention a dimensionless characteristic variable, the *reduced volume*, which measures the deflation of the vesicle, plays a significant role in the vesicle dynamics. It is defined as

$$\nu^{3D} = \frac{V_0}{\frac{4}{3}\pi(S_0/4\pi)^{3/2}},$$

where V_0 and S_0 represent the volume and the surface area of vesicle, respectively. This dimensionless number is nothing but the volume ratio of the vesicle to a sphere with the same surface area, and thus is equal to 1 for a sphere. In two dimensions a parameter equivalent to reduced volume is defined by *reduced area*, which is defined by

$$\nu^{2D} = \frac{A_0}{\pi(L_0/2\pi)^2},$$

where A_0 and L_0 denote for the area and the arc-length of vesicle respectively. Again this definition follows the ratio of the vesicle to a circle with the same arc-length. In the rest of dissertation, we would use the notation ν to denote reduced volume (3D) or reduced area (2D) for convenience.

2.2 Interfacial forces on vesicle

A vesicle membrane is known to be inextensible and exhibits a resistance against bending. Thus, the membrane energy can be modeled by two parts; namely, the elastic tension energy E_σ to enforce the inextensibility constraint [71], and the Helfrich type energy E_b [18] to resist the bending of the membrane. A way to derive the interfacial forces on vesicle is from the energy point of view, a variational method. Assume $E[\mathbf{X}]$ is an energy functional corresponding to vesicle configuration $\mathbf{X}$ (or interface), we can obtain the interfacial force $\mathbf{F}$ by taking variational derivative as

$$\mathbf{F} = -\frac{\delta E[\mathbf{X}]}{\delta \mathbf{X}},$$

where $\delta\mathbf{X}$ is a perturbation on interface. In next subsection, the mathematical definition of two energies E_σ and E_b are given in both two and three-dimensional space, and the corresponding interfacial force are obtained.

2.2.1 In two dimensions

Let us define the vesicle by $\Gamma(t)$, and its configuration is presented by $\mathbf{X} = \mathbf{X}(s,t) = (X(s,t), Y(s,t))$, $0 \leq s \leq L_b$, where s is a Lagrangian parameter along interface. The unit tangent vector (along the interface counterclockwise) and normal vector (pointed outward of interface) are denoted by $\boldsymbol{\tau}$ and $\mathbf{n}$ respectively, and these two vectors can be obtained by

$$\boldsymbol{\tau} = (\tau_1, \tau_2) = \frac{\mathbf{X}_s}{|\mathbf{X}_s|} \qquad \text{and} \qquad \mathbf{n} = (n_1, n_2) = (\tau_2, -\tau_1).$$

The total vesicle energy $E(t)$ which consists of elastic and bending energy is defined by

$$E(t) = E_\sigma(t) + E_b(t) = \int_\Gamma \sigma \; \mathrm{d}s + \frac{c_b}{2}\int_\Gamma \kappa^2 \; \mathrm{d}s,$$

where σ is the surface tension, κ is the curvature and c_b is the strength of bending rigidity. Let us get start the process of variational derivative. Denote $\widetilde{\mathbf{X}}(s) = (\widetilde{X}(s), \widetilde{Y}(s))$ as a perturbation of the interface $\mathbf{X}$ (omit time variable t) and ε as a small number. Then the perturbed elastic energy becomes

$$E_\sigma(\mathbf{X} + \varepsilon\widetilde{\mathbf{X}}) = \int_0^{L_b} \sigma|\mathbf{X}_s + \varepsilon\widetilde{\mathbf{X}}_s| \; \mathrm{d}s.$$

Taking the derivative of above energy with respect to ε, we obtain

$$\frac{dE_\sigma}{d\varepsilon}(\mathbf{X} + \varepsilon\widetilde{\mathbf{X}}) = \int_0^{L_b} \sigma \frac{(X_s + \varepsilon\widetilde{X}_s)\widetilde{X}_s + (Y_s + \varepsilon\widetilde{Y}_s)\widetilde{Y}_s}{\sqrt{(X_s + \varepsilon\widetilde{X}_s)^2 + (Y_s + \varepsilon\widetilde{Y}_s)^2}} \; \mathrm{d}s.$$

Then we evaluate the above equation at $\varepsilon = 0$ to obtain

$$\begin{aligned} \left.\frac{dE_\sigma}{d\varepsilon}(\mathbf{X} + \varepsilon\widetilde{\mathbf{X}})\right|_{\varepsilon=0} &= \int_0^{L_b} \sigma \frac{\mathbf{X}_s}{|\mathbf{X}_s|} \cdot \widetilde{\mathbf{X}}_s \; \mathrm{d}s = \int_0^{L_b} \sigma\boldsymbol{\tau} \cdot \widetilde{\mathbf{X}}_s \; \mathrm{d}s \\ &= \sigma\boldsymbol{\tau} \cdot \widetilde{\mathbf{X}}\Big|_0^{L_b} - \int_0^{L_b} \frac{\partial}{\partial s}(\sigma\boldsymbol{\tau}) \cdot \widetilde{\mathbf{X}} \; \mathrm{d}s \quad \text{(integration by part)} \\ &= -\int_0^{L_b} \frac{\partial}{\partial s}(\sigma\boldsymbol{\tau}) \cdot \widetilde{\mathbf{X}} \; \mathrm{d}s. \end{aligned}$$

This leads to the corresponding elastic tension force

$$\mathbf{F}_\sigma = \frac{\partial}{\partial s}(\sigma\boldsymbol{\tau}). \tag{2.1}$$

Especially notice that, if tension σ is a constant (for the case of clean droplet), then the elastic tension force would act only along normal direction of interface. On the other hand, for obtaining the bending force we perturb the bending energy by

$$E_b(\mathbf{X} + \varepsilon\widetilde{\mathbf{X}}) = \frac{c_b}{2}\int_0^{L_b} \left|\frac{\partial^2\mathbf{X} + \varepsilon\widetilde{\mathbf{X}}}{\partial s^2}\right|^2 \; \mathrm{d}s.$$

Taking derivative with respect to ε obtains

$$\begin{aligned}\left.\frac{dE_b}{d\varepsilon}(\mathbf{X}+\varepsilon\widetilde{\mathbf{X}})\right|_{\varepsilon=0} &= c_b\int_0^{L_b}\frac{\partial X+\varepsilon\widetilde{X}}{\partial s^2}\frac{\partial^2\widetilde{X}}{\partial s^2}+\frac{\partial Y+\varepsilon\widetilde{Y}}{\partial s^2}\frac{\partial^2\widetilde{Y}}{\partial s^2}\ \mathrm{d}s\\ &= c_b\int_0^{L_b}\frac{\partial^2\mathbf{X}}{\partial s^2}\cdot\frac{\partial^2\widetilde{\mathbf{X}}}{\partial s^2}\ \mathrm{d}s\\ &= -\int_0^{L_b}-c_b\frac{\partial^4\mathbf{X}}{\partial s^4}\cdot\widetilde{\mathbf{X}}\ \mathrm{d}s\quad\text{(integration by part twise)}\end{aligned}$$

Therefore we obtain the explicit form of bending force as

$$\mathbf{F}_b=-c_b\frac{\partial^4\mathbf{X}}{\partial s^4}. \tag{2.2}$$

Notice that there is another remarkable version of bending force as

$$\mathbf{F}_b=c_b\left(\frac{\partial^2\kappa}{\partial s^2}+\frac{\kappa^3}{2}\right)\mathbf{n}. \tag{2.3}$$

One can see the derivation in detail in [27]. Throughout this dissertation, we select the two-dimensional bending force $\mathbf{F}_b$ in Eq. (2.2).

2.2.2 In three dimensions

In three-dimensional case, the vesicle surface is presented by parametric form as $\mathbf{X}(\alpha,\beta,t)=(X(\alpha,\beta,t),Y(\alpha,\beta,t),Z(\alpha,\beta,t))$, $0\le\alpha\le L_\alpha$ and $0\le\beta\le L_\beta$, where α and β are two Lagrangian parameters. The total energy of vesicle is defined by

$$E(t)=E_\sigma(t)+E_b(t)=\int_\Gamma\sigma\ \mathrm{d}S+\frac{c_b}{2}\int_\Gamma H^2\ \mathrm{d}S,$$

where H is the surface mean curvature and $\mathrm{d}S$ is the local surface element. By taking variational derivative to the surface energy, one can derive the vesicle boundary force consisting of the elastic force $\mathbf{F}_\sigma$ and the bending force $\mathbf{F}_b$ as

$$\mathbf{F}_\sigma=\nabla_s\sigma-2H\sigma\mathbf{n}\qquad\text{and}\qquad\mathbf{F}_b=c_b\left(\Delta_s H+2H(H^2-K)\right)\mathbf{n}, \tag{2.4}$$

where K is the Gaussian curvature respectively, ∇_s is the surface gradient operator, and Δ_s is the surface Laplacian operator. The detailed derivation can be found in [69]. The computation of these geometrical quantities and operators can be obtained by geometrical fundamental form, ant they are given in detail in Appendix A.

Remark: Notice that there are terms of derivative of function along interface happened in the interfacial force term, especially fourth derivative of interfacial position is involved when evaluating bending force. These terms shall be treated very carefully in numerical simulation due to there might be a contributed interfacial stiffness.

2.3 Inextensible formulation

As mentioned previously, the vesicle exhibits inextensible property, which means the local surface element conserves as a quantity, and obviously this leads to conservation of global surface area of vesicle. The mathematical description of inextensibility in two and three-dimensional spaces are shown in the following.

2.3.1 In two dimensions

In two-dimensional space, the vesicle is represented by closed curve and its local arc-length is conserved. It is straightforward to write changing rate of local arc-length by

$$\frac{\partial}{\partial t}(|\mathbf{X}_s|) = \frac{\partial}{\partial t}\left(\sqrt{X_s^2+Y_s^2}\right) = \frac{X_sX_{st}+Y_sY_{st}}{|\mathbf{X}_s|} = \frac{\partial \mathbf{U}}{\partial s}\cdot\boldsymbol{\tau} = (\nabla_s\cdot\mathbf{U})\,|\mathbf{X}_s|\,, \tag{2.5}$$

where $\mathbf{U} = \frac{\partial \mathbf{X}}{\partial t}$ denotes for velocity defined on the vesicle (interface). From Eq. (2.5), since local arc-length does not change with time varying, i.e. $\frac{\partial}{\partial t}(|\mathbf{X}_s|) = 0$, we can deduce that surface divergence of velocity must be zero, that is

$$\nabla_s\cdot\mathbf{U} = 0. \tag{2.6}$$

2.3.2 In three dimensions

In three-dimensional case, we start by using the fact

$$|X_\alpha\times X_\beta|\,\mathbf{n} = X_\alpha\times X_\beta.$$

Then taking time derivative to the above equation obtains

$$\frac{\partial}{\partial t}\left(|X_\alpha\times X_\beta|\right)\mathbf{n} + |X_\alpha\times X_\beta|\,\frac{\partial \mathbf{n}}{\partial t} = \frac{\partial}{\partial t}\left(X_\alpha\times X_\beta\right).$$

By taking inner product of $\mathbf{n}$ on the both side of above equation, we have

$$\begin{aligned}
\frac{\partial}{\partial t}\left(|X_\alpha\times X_\beta|\right) &= \frac{\partial}{\partial t}\left(X_\alpha\times X_\beta\right)\cdot\mathbf{n}\\
&= \mathbf{n}\cdot(\mathbf{U}_\alpha\times\mathbf{X}_\beta) + \mathbf{n}\cdot(\mathbf{X}_\alpha\times\mathbf{U}_\beta)\\
&= \mathbf{U}_\alpha\cdot(\mathbf{X}_\beta\times\mathbf{n}) + \mathbf{U}_\beta\cdot(\mathbf{n}\times\mathbf{X}_\alpha)\\
&= \frac{\mathbf{U}_\alpha}{W}\left(\mathbf{X}_\beta\times(\mathbf{X}_\alpha\times\mathbf{X}_\beta)\right) + \frac{\mathbf{U}_\beta}{W}\left((\mathbf{X}_\alpha\times\mathbf{X}_\beta)\times\mathbf{X}_\beta\right)\\
&= \frac{\mathbf{U}_\alpha}{W}\left(G\mathbf{X}_\alpha - F\mathbf{X}_\beta\right) + \frac{\mathbf{U}_\beta}{W}\left(E\mathbf{X}_\beta - F\mathbf{X}_\alpha\right)\\
&= \frac{G\mathbf{U}_\alpha - F\mathbf{U}_\beta}{W}\cdot\mathbf{X}_\alpha + \frac{E\mathbf{U}_\beta - F\mathbf{U}_\alpha}{W}\cdot\mathbf{X}_\beta\\
&= (\nabla_s\cdot\mathbf{U})\,|X_\alpha\times X_\beta|\,.
\end{aligned}$$

Notice that we use the notation of first fundamental form (refer to Appendix A). One can see that, from the above equation, inextensible property again leads zero surface divergence of velocity field as

$$\nabla_s \cdot \mathbf{U} = 0. \tag{2.7}$$

In summary, inextensible property is equivalent zero surface divergence of velocity field as in Eqs. (2.6) and (2.7), and this is regarded as an additional *constraint*. In fact, tension plays a role as Lagrangian's multiplier to enforce inextensibility, we will explain it in detail in later section.

2.4 Hydrodynamical equations

In previous section, the membrane driven force of vesicle is obtained by minimizing total vesicle energy. The next step is to couple the motion of equation for fluid to the membrane force. The framework of fluid vesicle mechanism system will be clearly stated in this section.

2.4.1 Incompressible Navier-Stokes equations

Firstly, we begin by explaining mathematical formulation of fluid incompressibility. Consider a specific system of fluid with control volume V (shape can change with time), but the total mass does not change. Denote density by ρ and velocity field by $\mathbf{u}$, under Lagrangian framework, by tracking total mass of this control volume, the mass conservation law is stated as

$$\frac{D}{Dt}\int_{V(t)} \rho \, \mathrm{d}V = 0, \tag{2.8}$$

where $\frac{D}{Dt} = \frac{\partial}{\partial t} + \mathbf{u}\cdot\nabla$ denotes for material derivative. By Reynolds transport theorem, the above equation can be written by

$$\int_V \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) \, \mathrm{d}V = 0. \tag{2.9}$$

Since control volume is arbitrary chosen, the integrand in Eq. (2.9) is always zero, i.e.,

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0. \tag{2.10}$$

Eq. (2.10) is called *continuity equation* and can be alternatively written in the form of

$$\frac{D\rho}{Dt} + \nabla \cdot \mathbf{u} = 0. \tag{2.11}$$

For incompressible fluid, shape of control volume can change but mass and volume remain, that is, the density of fluid element remains constant. Under this assumption, we can deduce $\frac{D\rho}{Dt} = 0$, thus mass conservation law of Eq. (2.11) is reduced by

$$\nabla \cdot \mathbf{u} = 0. \tag{2.12}$$

The Navier-Stokes equation is govern by conservation law of momentum. In classical mechanics, momentum is the product of the mass and velocity of an object. The conservation of momentum describes the total amount of momentum within a control volume keeps as a constant. That is to say, momentum is neither created nor destroyed, but it can be changed by external forces. Actually, the conservation of momentum is an application of Newton's second law of motion, which states the rate change of momentum of a fluid mass is equal to net external forces acting one the mass. Such external forces are consisted of two classes. One is the body force (body force per unit mass), gravitational force or electromagnetic force for instance; the other one is surface force (surface force per unit area), such as pressure forces or viscous stress.

With basic physical property of fluid mechanics, now we begin to introduce mathematical formulation of Navier-Stokes equation. We assume the fluid is Newtonian, based on constitutive law, the expression for stress tensor is

$$\sigma_{ij} = -p\delta_{ij} + \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right), \tag{2.13}$$

where δ_{ij} presents Kronecker delta function ($\delta_{ij} = 1$ if $i = j$; $\delta_{ij} = 0$ if $i \neq j$), p is the pressure and μ is the viscosity. Throughout this dissertation, the density and viscosity are both assumed to be uniform constant. Coupling the fluid incompressibility constrain Eq. (2.12), the mathematical equations of motion consist of a viscous incompressible fluid in a domain Ω can be written as follows.

$$\begin{aligned} &\rho \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\, \mathbf{u} \right) = \nabla \cdot \sigma_{ij} = -\nabla p + \mu \Delta \mathbf{u} \qquad \text{in } \Omega, \\ &\nabla \cdot \mathbf{u} = 0 \qquad \text{in } \Omega. \end{aligned}$$

Notice that there is no equation describing evolution of pressure p. In fact, pressure p plays a role as Lagrange's multiplier to enforce the fluid incompressibility constraint. On the other hand, for well-poseness of Navier-Stokes equation, the velocity field is equipped with boundary condition, and we name several types of boundary condition as follows.

- No-slip boundary condition: There is no fluid which penetrates the boundary, the fluid is at rest status on boundary, that is

$$\mathbf{u}|_{\partial\Omega} = \mathbf{0}.$$

- Inflow boundary condition: The velocity field is given specifically on the boundary, that is

$$\mathbf{u}|_{\partial\Omega} = \mathbf{u}_b.$$

- Outflow boundary condition: Neither velocity component changes in normal direction to the boundary, that is

$$\frac{\partial \mathbf{u}}{\partial n} = \mathbf{0}.$$

2.4.2 Coupling the fluid equations to vesicle forces

In this section, the governing equation of fluid motion and force acting on vesicle membrane are combined by Immersed boundary formulation. In immersed boundary framework, the fluid variables (velocity field, pressure, etc.) are defined in Eulerian coordinate and interfacial variables (membrane position, membrane force, etc.) are defined in Lagrangian coordinate. These two types of variable are linked by interaction equations in which the Dirac delta function plays a prominent role. Now we come to state the whole fluid system of vesicle dynamics through immersed boundary method in two-dimensional space. The mathematical equations of motion consist of a viscous incompressible fluid in a domain Ω containing an immersed, inextensible, massless vesicle boundary (or interface) $\Gamma(t)$ which can be written in an immersed boundary formulation as

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u}\right) = -\nabla p + \mu\Delta\mathbf{u} + \int_\Gamma \mathbf{F}(s,t)\delta(\mathbf{x}-\mathbf{X}(s,t))\,\mathrm{d}s \quad \text{in } \Omega, \tag{2.14}$$

$$\nabla\cdot\mathbf{u} = 0 \quad \text{in } \Omega, \tag{2.15}$$

$$\nabla_s\cdot\mathbf{U} = 0 \quad \text{on } \Gamma, \tag{2.16}$$

$$\frac{\partial \mathbf{X}}{\partial t}(s,t) = \mathbf{U}(s,t) = \int_\Omega \mathbf{u}(\mathbf{x},t)\delta(\mathbf{x}-\mathbf{X}(s,t))\,\mathrm{d}\mathbf{x} \quad \text{on } \Gamma. \tag{2.17}$$

The interfacial force $\mathbf{F}$ consists of tension and bending force as

$$\mathbf{F} = \mathbf{F}_\sigma + \mathbf{F}_b = \frac{\partial}{\partial s}(\sigma\boldsymbol{\tau}) - c_b\frac{\partial^4\mathbf{X}}{\partial s^4}. \tag{2.18}$$

Eqs. (2.14)-(2.15) are the incompressible Navier-Stokes equations with a singular force term $\mathbf{F}$ arising from the vesicle membrane force. Eq. (2.16) represents the inextensibility constraint of the vesicle surface which is equivalent to the zero surface divergence of the velocity along the interface. Eq. (2.17) simply explains that the interface moves along with the local fluid velocity (the interfacial velocity). Here, the interfacial velocity $\mathbf{U}$ is the interpolation of the fluid velocity at the interface defined as in traditional IB formulation. The interaction between the fluid and the interface is linked by the two-dimensional Dirac delta function $\delta(\mathbf{x}) = \delta(x)\delta(y)$. One can easily extend to fully three-dimensional Navier-Stokes equation by taking three-dimensional vesicle membrane force mentioned previously.

It is worthy to mention that, in fact, the tension σ acts like a Lagrange's multiplier function to enforce the local inextensibility constraint along the surface which is exactly the same role played by the pressure to enforce the fluid incompressibility in Navier-Stokes equations. Therefore, the development of an efficient and accurate algorithm for vesicle dynamics remains quite challenging; not to mention the vesicle surface moves along with the surrounding fluid as well.

Chapter 3

Immersed boundary method

The immersed boundary (IB) method was firstly proposed by Peskin [48] to simulate blood heart value problems. The IB method has found a wide variety of applications in biological mechanics and has been successfully applied to study fluid-structure interaction problems. The main idea of IB method is to treat Lagrangian material as a part of fluid in which arises additional interfacial forces. The fluid variables are solved in regular Cartesian domain (uniform rectangular lattice) without any modification of fluid equation; the interface is tracked by a set of discrete Lagrangian markers which can freely move in Eulerian domain and the shape of interface can has complex geometry. The interfacial forces can be computed through spatial position of Lagrangian markers and only has effect in its vicinity fluid, i.e. these forces are spread into its surrounding fluid lattices. With presence of these forces, the new fluid velocity can be obtained. The interface position is then advanced by this new fluid velocity through an interpolation.

3.1 Connection between fluids and interfaces

In this section, we introduce the linkage between fluid equation and Lagrangian interface. By taking advantage of Dirac delta function, a value of function can be obtained through product with a shifted delta source function, and then integrate over the whole space (thus the shifted delta function is regarded as a kernel of integral). We clarify this definition by writing the spreading forces on Cartesian grid by

$$\mathbf{f}(\mathbf{x},t) = \int_\Gamma \mathbf{F}(s,t)\,\delta(\mathbf{x}-\mathbf{X}(s,t))\,\mathrm{d}s, \tag{3.1}$$

and the interpolation of fluid velocity field by

$$\mathbf{u}(\mathbf{X}(s,t),t) = \mathbf{U}(s,t) = \int_\Omega \mathbf{u}(\mathbf{x},t)\,\delta(\mathbf{x}-\mathbf{X}(s,t))\,\mathrm{d}\mathbf{x}. \tag{3.2}$$

Notice that Eqs. (3.1) and (3.2) represent for a line integral along the interface and an area integral over whole fluid domain respectively (in three-dimensional case, they are surface and volume integral). It is interesting to see that the total amount of singular force in Eq. (3.1), which is an integral of $\mathbf{f}(\mathbf{x}, t)$ over whole domain, is equal to the total force on the immersed interface. This means although the interfacial forces (force per length) are converted to body force (force per area), the total amount of applied body force still preserves to act on the fluids. The important feature of IB method is Eq. (3.1) converts Lagrangian to Eulerian coordinate while Eq. (3.2) does in opposite direction.

3.2 Discrete delta function

As mentioned in previous section, the major novelty of IB method is introducing a delta function, which plays an important role converting coordinate between Eulerian and Lagrangian manner. Since delta function is in a view point of distribution under generalized function theorem, in the aspect of numerics, the discrete version of delta function shall be acquired.

The discrete version of delta function $\delta_h(\mathbf{x})$ should be smooth enough to guarantee the smoothness of distribution of Eqs. (3.1) and (3.2), and has compact support in a narrow region to ensure the numerical efficiency when computing spreading and interpolation. A two-dimensional delta function is required to be the form of

$$\delta_h(\mathbf{x}) = \frac{1}{h^2}\phi\left(\frac{x}{h}\right)\phi\left(\frac{y}{h}\right),$$

where $\mathbf{x} = (x, y)$, $\phi(r)$ is a real function, and $\delta_h \to \delta$ as $h \to 0$. We say $\phi(r)$ has *moment condition* of order q if

$$\sum_i (\alpha - i)^r \phi(\alpha - i) = \begin{cases} 1, & r = 0 \\ 0, & 1 \leq r \leq q-1 \end{cases} \tag{3.3}$$

for all values of α. When $r = 0$, ensures total mass of the discrete delta function is identically one. The moment condition (3.3) represents for the numerical accuracy of interpolation. For instance of one-dimensional case, suppose ϕ has moment condition q, let f be a real smooth function and α be an arbitrary real number, we have

$$f(\alpha) - h\sum_i f(x_i)\delta_h(x_i - \alpha) = O(h^q),$$

where $x_i = ih$. The above equation can be done directly by using Taylor expansion on $f(x_i)$ at the point $x = \alpha$. Now we list several regular discrete functions which are popular and widely used in lot of literatures. They are the 2-point hat function ϕ_1 [29], the 4-point cosine function ϕ_2 [48], the 3-point discrete function ϕ_3 [54], and the 4-point piecewise function ϕ_4 [50] as follows (with moment condition of order 2,1,2 and 2 respectively).

- A 2-point hat function ϕ_1:

$$\phi_1(r) = \begin{cases} 1 - |r|, & |r| \leq 1, \\ 0, & 1 \leq |r|. \end{cases}$$

- A 4-point cosine function ϕ_2:

$$\phi_2(r) = \begin{cases} \frac{1}{4}\left(1 + \cos(\frac{\pi r}{2})\right), & |r| \leq 2, \\ 0, & 2 \leq |r|. \end{cases}$$

- A 3-point function ϕ_3:

$$\phi_3(r) = \begin{cases} \frac{1}{3}(1 + \sqrt{-3r^2 + 1}), & |r| \leq 0.5, \\ \frac{1}{6}\left(5 - 3|r| - \sqrt{-3(1 - |r|)^2 + 1}\right), & 0.5 \leq |r| \leq 1, \\ 0, & 1.5 \leq |r|. \end{cases}$$

- A 4-point function ϕ_4:

$$\phi_4(r) = \begin{cases} \frac{1}{8}\left(3 - 2|r| + \sqrt{1 + 4|r| - 4r^2}\right), & |r| \leq 1, \\ \frac{1}{8}\left(5 - 2|r| - \sqrt{-7 + 12|r| - 4r^2}\right), & 1 \leq |r| \leq 2, \\ 0, & 2 \leq |r|. \end{cases}$$

However, it has been found that with these regular discrete functions, a non-physical oscillations might happened in numerical simulation which is mainly due to the derivative of regular functions do not satisfy certain moment condition. In [70], the authors developed an simple smoothing technique for discrete delta function $\phi^*(r)$ by extending a half mesh wider support as

$$\phi^*(r) = \int_{r-0.5}^{r+0.5} \phi(r')\, \mathrm{d}r'. \tag{3.4}$$

In terms of the definition (3.4), the smoothed functions ϕ_1^*, ϕ_2^*, ϕ_3^* and ϕ_4^* corresponding to ϕ_1, ϕ_2, ϕ_3 and ϕ_4 can be formulated as follows.

- A smoothed 2-point hat function ϕ_1^*:

$$\phi_1^*(r) = \begin{cases} \frac{3}{4} - r^2, & |r| \leq 0.5, \\ \frac{9}{8} - \frac{3}{2}|r| + \frac{r^2}{2}, & 0.5 \leq |r| \leq 1.5, \\ 0, & 1.5 \leq |r|. \end{cases}$$

- A smoothed 4-point cosine function ϕ_2^*:

$$\phi_2^*(r) = \begin{cases} \frac{1}{4\pi}\left[\pi + 2\sin\left(\frac{\pi}{4}(2r+1)\right) - 2\sin\left(\frac{\pi}{4}(2r-1)\right)\right], & |r| \leq 1.5, \\ -\frac{1}{8\pi}\left[-5\pi + 2\pi|r| + 4\sin\left(\frac{\pi}{4}(2|r|-1)\right)\right], & 1.5 \leq |r| \leq 1.5, \\ 0, & 2.5 \leq |r|. \end{cases}$$

- A smoothed 3-point function ϕ_3^*:

$$\phi_3^*(r) = \begin{cases} \frac{17}{48} + \frac{\sqrt{3}\pi}{108} + \frac{|r|}{4} - \frac{r^2}{4} + \frac{1-2|r|}{16}\sqrt{-12r^2+12|r|+1} & \\ -\frac{\sqrt{3}}{12}\sin^{-1}\left(\frac{\sqrt{3}}{2}(2|r|-1)\right), & |r| \leq 1, \\ \frac{55}{48} - \frac{\sqrt{3}\pi}{108} - \frac{13|r|}{12} + \frac{r^2}{4} + \frac{2|r|-3}{48}\sqrt{-12r^2+36|r|-23} & \\ +\frac{\sqrt{3}}{36}\sin^{-1}\left(\frac{\sqrt{3}}{2}(2|r|-3)\right), & 1 \leq |r| \leq 2, \\ 0, & 2 \leq |r|. \end{cases}$$

- A smoothed 4-point function ϕ_4^*:

$$\phi_4^*(r) = \begin{cases} \frac{3}{8} + \frac{\pi}{32} - \frac{r^2}{4}, & |r| < 0.5, \\ \frac{1}{4} + \frac{1-|r|}{8}\sqrt{-2+8|r|-4r^2} - \frac{1}{8}\sin^{-1}\left(\sqrt{2}(|r|-1)\right), & 0.5 \leq |r| \leq 1.5, \\ \frac{17}{16} - \frac{\pi}{64} - \frac{3|r|}{4} + \frac{r^2}{8} + \frac{|r|-2}{16}\sqrt{-14+16|r|-4r^2} & \\ +\frac{1}{16}\sin^{-1}\left(\sqrt{2}(|r|-2)\right), & 1.5 \leq |r| \leq 2.5, \\ 0, & 2.5 \leq |r|. \end{cases}$$

A comparative plot between regular function ϕ and corresponding smoothed function ϕ^* are depicted in Figure. 3.1. To summarize, we list the advantages of smoothed function below.

- Through Eq. (3.4), the smoothed function has a half mesh wider support and gains one higher derivative than regular one

- The moment conditions of smoothed function are preserved from original regular function except ϕ_2^* gains moment condition of order 2.

- The derivative of smoothed functions $\phi'(r)$ have one order higher moment condition than $\phi(r)$.

One can refer the details of proof for last two points in [70]. In our numerical experiments, the smoothed delta function indeed benefits not only improving the stability of whole numerical scheme, but also removing nonphysical interfacial oscillation. We take ϕ_4^* as a standard to construct discrete delta function throughout all numerical simulations in this dissertation.

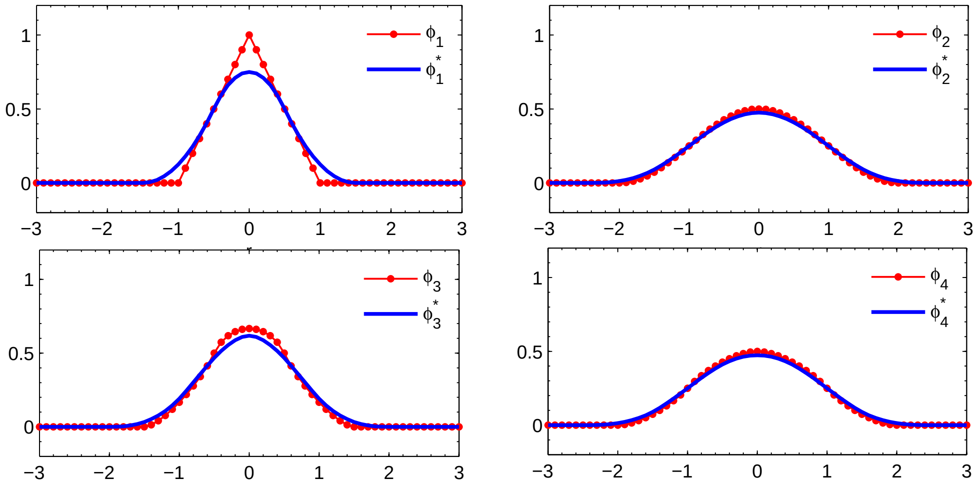


Figure 3.1: The figure of discrete function ϕ_i $(i = 1, 2, 3, 4)$ and correspond smoothed functions ϕ_i^* $(i = 1, 2, 3, 4)$.

3.3 Numerical setup

In this section, we introduce a numerical setup for spatial discretization based on standard finite difference method in detail. Firstly, a famous and traditional staggered grid is applied for discretization of fluid equations. Under layout of staggered grid, fluid equations can be solved accurately by some existing fluid solver, such as projection method. Secondly, the usage of boundary conditions and approximation of values at ghost point will be carried out. Lastly, the numerical setting of interfacial values are presented.

3.3.1 Staggered grid

The staggered marker-and-cell (MAC) grid was proposed by Harlow and Welsh [17] to simulate viscous impressible flow with free surface. The rectangular computational domain Ω is set by $\Omega = [a, b] \times [c, d]$, speaking in detail, the pressure is defined on the grid points labeled as $x = (x_i, y_j) = (a + (i - 1/2)\triangle x, c + (j - 1/2)\triangle y)$, $i = 1, 2, \cdots, m$ and $j = 1, 2, \cdots, n$, while the velocity components u and v are defined at $(x_i, y_{j-1/2}) = (a + (i - 1)\triangle x, c + (j - 1/2)\triangle y)$ and $(x_{i-1/2}, y_j) = (a + (i - 1/2)\triangle x, c + (j - 1)\triangle y)$, respectively. Here, we assume that a uniform mesh width $h = \triangle x = \triangle y$ is used, although that is not necessary. One can see the sketch of staggered grid from Figure 3.2.

Let us demonstrate a standard second-order finite difference method based on staggered grid for fluid equation. In two-dimensional space, the incompressible Navier-

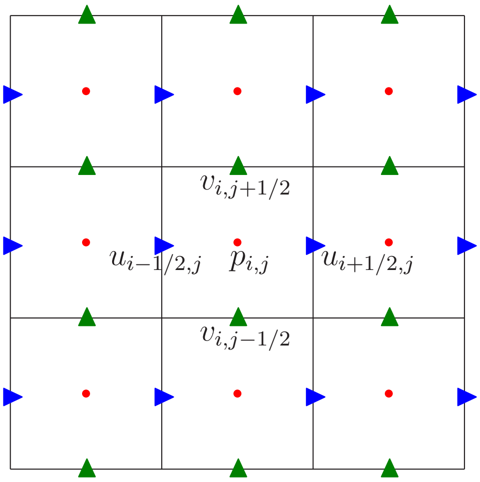


Figure 3.2: The layout of staggered grid in two-dimensional rectangular domain. The red dots indicate location of pressure p, while the green and blue triangles represent for velocity component u and v respectively.

Stokes equation is expressed explicitly by

$$\rho\left(\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}\right)=-\frac{\partial p}{\partial x}+\mu\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right)+f, \tag{3.5}$$

$$\rho\left(\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}\right)=-\frac{\partial p}{\partial y}+\mu\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\right)+g, \tag{3.6}$$

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}=0, \tag{3.7}$$

where f and g represent external forces. Let start with discretization of fluid incompressibility in Eq. (3.7), a second-order accurate finite difference based on Figure. 3.2 gives

$$\frac{u_{i+1/2,j}-u_{i-1/2,j}}{h}+\frac{v_{i,j+1/2}-v_{i,j-1/2}}{h}=0. \tag{3.8}$$

In Eqs. (3.5) and (3.6), the standard second-order finite difference expressions for the derivatives with respect to space for (i,j)-cell are computed by

$$\begin{aligned}
\left(u\frac{\partial u}{\partial x}\right)_{i+1/2,j} &= u_{i+1/2,j}\frac{u_{i+3/2,j}-u_{i-1/2,j}}{2h},\\
\left(v\frac{\partial u}{\partial y}\right)_{i+1/2,j} &= v_{i+1/2,j}\frac{u_{i+1/2,j+1}-u_{i+1/2,j-1}}{2h},\\
\left(\frac{\partial p}{\partial x}\right)_{i+1/2,j} &= \frac{p_{i+1,j}-p_{i,j}}{h},\\
\left(\frac{\partial^2 u}{\partial x^2}\right)_{i+1/2,j} &= \frac{u_{i-1/2,j}-2u_{i+1/2,j}+u_{i+3/2,j}}{h^2},\\
\left(\frac{\partial^2 u}{\partial y^2}\right)_{i+1/2,j} &= \frac{u_{i+1/2,j-1}-2u_{i+1/2,j}+u_{i+1/2,j+1}}{h^2},
\end{aligned}$$

$$\begin{aligned}
\left(u\frac{\partial v}{\partial x}\right)_{i,j+1/2} &= u_{i,j+1/2}\frac{v_{i+1/2,j+1}-u_{i+1/2,j-1}}{2h},\\
\left(v\frac{\partial v}{\partial y}\right)_{i,j+1/2} &= v_{i,j+1/2}\frac{v_{i,j+3/2}-v_{i,j-1/2}}{2h},\\
\left(\frac{\partial p}{\partial y}\right)_{i,j+1/2} &= \frac{p_{i,j+1}-p_{i,j}}{h},\\
\left(\frac{\partial^2 v}{\partial x^2}\right)_{i,j+1/2} &= \frac{v_{i-1,j+1/2}-2v_{i,j+1/2}+v_{i+1,j+1/2}}{h^2},\\
\left(\frac{\partial^2 v}{\partial y^2}\right)_{i,j+1/2} &= \frac{v_{i,j-1/2}-2v_{i,j+1/2}+v_{i,j+3/2}}{h^2}.
\end{aligned}$$

Notice that the terms $u_{i,j+1/2}$ and $v_{i+1/2,j}$ do not show up in staggered grid formation, and these two terms can be evaluated by a second-order linear interpolation as

$$\begin{aligned}
u_{i,1+1/2} &= \frac{1}{4}\left(u_{i-1/2,j}+u_{i+1/2,j}+u_{i-1/2,j+1}+u_{i+1/2,j+1}\right),\\
v_{i+1/2,j} &= \frac{1}{4}\left(v_{i,j-1/2}+v_{i+1,j-1/2}+v_{i,j+1/2}+v_{i+1,j+1/2}\right).
\end{aligned}$$

In further chapters, we will use the notations ∇_h, $\nabla_h\cdot$ and Δ_h to denote for numerical gradient, divergence and Laplace mentioned above.

3.3.2 Ghost values from boundary conditions

In this section, we demonstrate the evaluation of approximation for ghost values due to they are used in fluid solver. In staggered grid formation, the pressure is define on a cell center, while velocity component u and v are on cell face.

In our numerical simulation, we always assume that u is given on $x = a$ and $x = b$, and v is also known on $y = c$ and $y = d$. That is, Dirichlet boundary conditions (either no-slip or inflow) are assigned to $u(a, y)$, $u(b, y)$, $v(x, c)$ and $v(x, d)$. For the rest boundary conditions ,$u(x, c)$, $u(x, d)$, $v(a, y)$ and $v(b, y)$, can be set as either Dirichlet or Neumann boundary type. Under this assumption, the boundary values of $u_{1/2,j}$ and $u_{m+1/2,j}$ can be obtained exactly corresponding to its boundary conditions $u(a, y)$ and $u(b, y)$, as well as boundary values of $v_{i,1/2}$ and $v_{i,n+1/2}$ to $v(x, c)$ and $v(x, d)$. Therefore, the ghost points are $u_{i+1/2,0}$, $u_{i+1/2,m+1}$, $v_{0,j+1/2}$ and $v_{m+1,j+1/2}$. We separate two types of boundary condition to approximate those ghost values bellow.

- **Dirichlet boundary condition**: The value of ghost point is evaluated by a second-order extrapolation as follows.

$$\begin{aligned}
u_{i+1/2,0} &= 2u(x_{i+1/2}, c) - u_{i+1/2,1},\\
u_{i+1/2,n+1} &= 2u(x_{i+1/2}, d) - u_{i+1/2,n},\\
v_{0,j+1/2} &= 2v(a, y_{j+1/2}) - v_{1,j+1/2},\\
v_{m+1,j+1/2} &= 2v(b, y_{j+1/2}) - v_{m,j+1/2}.
\end{aligned}$$

- **Neumann boundary condition**: Suppose the value of $\frac{\partial u}{\partial y}(x, c)$ is given, then the ghost value $u_{i+1/2,0}$ can be obtained by a central difference with local truncation error $O(h^2)$ by

$$\frac{u_{i+1/2,1} - u_{i+1/2,0}}{h} = \frac{\partial u}{\partial y}(x_{i+1/2}, c).$$

  Rearranging the above equation gives

$$u_{i+1/2,0} = h\frac{\partial u}{\partial y}(x_{i+1/2}, c) - u_{i+1/2,1}.$$

  The other ghost values are computed in the similar way as

$$\begin{aligned}
u_{i+1/2,n+1} &= h\frac{\partial u}{\partial y}(x_{i+1/2}, d) + u_{i+1/2,n},\\
v_{0,j+1/2} &= h\frac{\partial v}{\partial x}(a, y_{j+1/2}) - v_{1,j+1/2},\\
v_{m+1,j+1/2} &= h\frac{\partial v}{\partial x}(b, y_{j+1/2}) + v_{m,j+1/2}.
\end{aligned}$$

3.3.3 Lagrangian manners

For the immersed interface $\mathbf{X}$, we use a collection of discrete points $s_k = k\Delta s$, $k = 0, 1, \cdots, M$, with the interface mesh width Δs so the Lagrangian markers of the interface are represented by $\mathbf{X}_k = \mathbf{X}(s_k)$. Other interfacial quantities, such as force density $\mathbf{F}$ and velocity $\mathbf{U}$ are defined on position $\mathbf{X}_k$ so the notations $\mathbf{F}_k = \mathbf{F}(s_k)$ and $\mathbf{U}_k = \mathbf{U}(s_k)$ are used. One can see the illustration of IB framework of regular uniform grid and set of discrete Lagrangian markers in Figure 3.3.

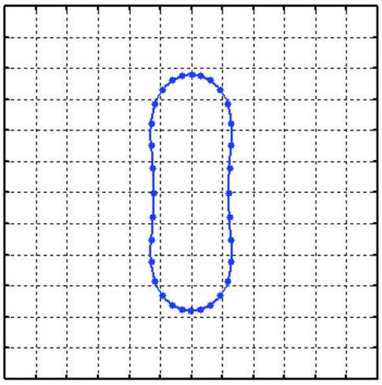


Figure 3.3: The basic framework of IB method, spreading discrete Lagrangian markers in a computational domain with uniform mesh.

Especially pay attention that we often have to compute derivatives of interfacial quantities. For instance, the vesicle membrane forces which consist of tension and bending force involve evaluating derivatives of position $\mathbf{X}_k$. Therefore calculation of derivative along the interface plays an important role to the dynamics of the interface. Here we provide two ways to compute derivative of a function along interface. Without loss of generality, for any function defined on the interface $\psi(s)$, we approximate the partial derivative $\frac{\partial \psi}{\partial s}$ by

- **Finite difference method**: The value of $\frac{\partial \psi}{\partial s}$ can be approached by standard second-order central difference scheme as

$$D_f\psi(s) = \frac{\psi(s + \Delta s/2) - \psi(s - \Delta s/2)}{\Delta s}.$$

- **Spectral method**: Since ψ is assumed to be periodic along interface, we can adopt the spectral Fourier discretization to achieve higher-order of accuracy. The interface is represented by the discrete Fourier series expansion as

$$\psi(s) = \sum_{k=-M/2}^{M/2-1} \widehat{\psi_k} e^{iks},$$

where $\widehat{\psi_k}$ is the corresponding Fourier coefficients for $\psi(s)$. This Fourier coefficients can be performed very efficiently by using the Fast Fourier Transform (FFT). Then the Fourier coefficients of P-th derivative of $\psi(s)$ can be obtained by taking an inverse FFT to $(ik)^P\widehat{\psi_k}$.

Chapter 4

Fractional step immersed boundary method

In this chapter, we develop a fractional step method based on IB formulation for Stokes flow with an inextensible interface (vesicle without bending stiffness). Solving the fluid variables such as the velocity and pressure, the present problem involves finding extra unknown elastic tension such that the surface divergence of the velocity is zero along the interface. Once the velocity is found, the interface is moving according to the local fluid velocity. Since the fluid is incompressible and the interface is inextensible, both the area enclosed by the interface and the total length of the interface should be conserved. Notice that, the dynamics of vesicles are determined by their boundary rigidity, inextensibility, and the hydrodynamical forces. Our present model for inextensible interface can be regarded as a simplified vesicle model without bending effect.

The rest of the chapter are organized as follows. In the next section, we describe our governing equations for the Stokes flow with an inextensible interface based on IB formulation. We also show the skew-adjoint property between the spreading operator acting on the tension and the surface divergence of the velocity. In Section 4.2, the symmetry of the resultant matrix equation is provided first, and then a numerical algorithm based on the fractional step method is developed. Finally, the numerical results including the convergence tests and the tank-treading motion for an inextensible interface under a simple shear flow are shown in detail in Section 4.4.

4.1 Governing equations

We begin by stating the mathematical formulation of the Stokes flow with an inextensible interface. Consider there is a moving, immersed, inextensible interface $\Gamma(t)$ in the fixed fluid domain Ω. We assume that the fluids inside and outside of the interface are the same so the governing equations in IB formulation can be written as follows.

$$-\nabla p + \mu \Delta \mathbf{u} + \int_\Gamma \frac{\partial}{\partial s}(\sigma \boldsymbol{\tau}) \delta(\mathbf{x} - \mathbf{X}(s,t)) \, \mathrm{d}s = 0 \text{ in } \Omega, \tag{4.1}$$

$$\nabla \cdot \mathbf{u} = 0 \text{ in } \Omega, \tag{4.2}$$

$$\nabla_s \cdot \mathbf{U} = 0 \text{ on } \Gamma, \quad \text{where } \mathbf{U}(s,t) = \int_\Omega \mathbf{u}(\mathbf{x},t) \delta(\mathbf{x} - \mathbf{X}(s,t)) \, \mathrm{d}\mathbf{x}, \tag{4.3}$$

$$\frac{\partial \mathbf{X}}{\partial t}(s,t) = \mathbf{U}(s,t) \text{ on } \Gamma. \tag{4.4}$$

Eqs. (4.1) and (4.2) are the familiar incompressible Stokes equations with a singular force term arising from the interface. Eq. (4.3) represents the inextensibility constraint of the interface which is equivalent to the zero surface divergence of the velocity along the interface. Here, the interfacial velocity $\mathbf{U}$ is simply the interpolation of the fluid velocity at the interface which is defined as in traditional IB formulation. Eq. (4.4) simply represents that the interface moves along with the local fluid velocity (the interfacial velocity). The interaction between the fluid and the interface is linked by the two-dimensional Dirac delta function $\delta(\mathbf{x}) = \delta(x)\delta(y)$. Unlike the traditional IB formulation in which the elastic tension $\sigma(s,t)$ is either known or a function of immersed boundary configurations, here, the tension is a part of solution needed to be determined. In this model, we consider the Stokes flow; however, the numerical method can be extended to Navier-Stokes flow straightforwardly by treating the nonlinear advection terms explicitly during the time evolution.

The difficulties in solving the above interfacial problem are as follows. Firstly, since the fluid is incompressible and the interface is inextensible, both the area enclosed by the interface and total length of the interface should be conserved simultaneously. Furthermore, the local inextensibility constraint (4.3) is more stringent than the conservation of the total interfacial length since the latter one is a global constraint. Secondly, the elastic tension must be treated as an unknown function which is needed to be solved with the fluid variables simultaneously. In previous literature, most of related work is based on boundary integral methods, see for example, [71, 66, 59] and the references therein. However, boundary integral methods generally assume infinite domains, and cannot be generalized to full Navier-Stokes equations since there is no corresponding Green function. Until recently, Kim and Lai [28] have applied a penalty IB method to simulate the dynamics of inextensible vesicles. By introducing two different kind of Lagrangian markers, the authors are able to decouple the fluid and vesicle dynamics so that the computation can be performed more efficiently. One potential problem of this approach is that the time step depends on the penalty number and must be chosen smaller as the penalty number becomes larger. In [35], a new finite difference scheme based on immersed interface method (IIM) has been developed for solving the present problem in Navier-Stokes flow. The authors treat the unknown elastic tension as an augmented variable so that the augmented IIM can be applied. In this chapter, we discretize the equations (4.1)-(4.3) directly without decoupling and use a fractional step method to solve the resultant linear system of equations. The numerical algorithm will be given in next section.

Before to continue, we first prove that spreading operator acting on the function σ and the surface divergence operator of the velocity are skew-adjoint with each other. To proceed, let us define the spreading operator S of σ and the surface divergence operator ∇_s of $\mathbf{U}$ as follows.

$$S(\sigma) = \int_\Gamma \frac{\partial}{\partial s}(\sigma \boldsymbol{\tau}) \delta(\mathbf{x} - \mathbf{X}(s)) \, \mathrm{d}s. \tag{4.5}$$

$$\nabla_s \cdot \mathbf{U} = \frac{\partial \mathbf{U}}{\partial \boldsymbol{\tau}} \cdot \boldsymbol{\tau} = \frac{\partial \mathbf{U}}{\partial s} \cdot \boldsymbol{\tau} \Big/ |\mathbf{X}_s| = \frac{\partial}{\partial s} \left(\int_\Omega \mathbf{u}(\mathbf{x}) \delta(\mathbf{x} - \mathbf{X}(s)) \, \mathrm{d}\mathbf{x} \right) \cdot \boldsymbol{\tau} \Big/ |\mathbf{X}_s| . \tag{4.6}$$

We also define the inner product of functions on Ω and Γ in the following.

$$\langle \mathbf{u}, \mathbf{v} \rangle_\Omega = \int_\Omega \mathbf{u}(\mathbf{x}) \cdot \mathbf{v}(\mathbf{x}) \, \mathrm{d}\mathbf{x}, \quad \langle f, g \rangle_\Gamma = \int_\Gamma f(l)\, g(l)\, dl, \tag{4.7}$$

where l in Eq. (4.7) is the arc-length parameter. Then we have

$$\begin{aligned} \langle S(\sigma), \mathbf{u} \rangle_\Omega &= \int_\Omega \left(\int_\Gamma \frac{\partial}{\partial s}(\sigma \boldsymbol{\tau}) \delta(\mathbf{x} - \mathbf{X}(s)) \, \mathrm{d}s \right) \cdot \mathbf{u}(\mathbf{x}) \, \mathrm{d}\mathbf{x} \\ &= \int_\Gamma \frac{\partial}{\partial s}(\sigma \boldsymbol{\tau}) \cdot \left(\int_\Omega \mathbf{u}(\mathbf{x}) \delta(\mathbf{x} - \mathbf{X}(s)) \, \mathrm{d}\mathbf{x} \right) \mathrm{d}s \\ &= -\int_\Gamma \sigma \left(\boldsymbol{\tau} \cdot \frac{\partial \mathbf{U}}{\partial s} \right) \mathrm{d}s \quad \text{(intergartion by parts and the closed interface)} \\ &= \int_\Gamma \sigma \left(-\frac{\partial \mathbf{U}}{\partial s} \cdot \boldsymbol{\tau} \Big/ |\mathbf{X}_s| \right) \left| \frac{\partial \mathbf{X}}{\partial s} \right| \mathrm{d}s \\ &= \langle \sigma, -\nabla_s \cdot \mathbf{U} \rangle_\Gamma = \langle \sigma, S^*(\mathbf{U}) \rangle_\Gamma \end{aligned} \tag{4.8}$$

By comparison, it leads that the spreading operator and the surface divergence operator are skew-adjoint.

The reason for showing the skew-adjointness of those two operators are two fold. Firstly, since the surface divergence of velocity is zero in Eq. (4.8), from the above derivation, we have $\langle S(\sigma), \mathbf{u} \rangle_\Omega = 0$. That is, the present elastic tension does not do work to the fluid which is not surprising since it is the Lagrange multiplier for the inextensible constraint. However, if we add the bending force along the interface as the one in vesicle problems [66, 28], then the bending force does do work to the fluid. Secondly, the skew-adjointness is also satisfied in discrete sense (see in next section) so that the resultant matrix equation is symmetric; therefore, some efficient iterative solvers such as conjugate gradient (CG) method can be easily applied.

4.2 Numerical algorithm

We now are ready to discretize Eq. (4.1)-(4.4) by the IB method. Let Δt be the time step size, and the superscript of the variables denote the time step index. At the beginning of each time step n, the interface configuration $\mathbf{X}_k^n$, and the unit tangent $\boldsymbol{\tau}_{k-1/2}^n$ are all given. The numerical algorithm can be written as follows.

$$-\nabla_h p^{n+1} + \mu \Delta_h \mathbf{u}^{n+1} + \sum_{k=0}^{M-1} D_s \left(\sigma^{n+1} \boldsymbol{\tau}^n\right)_k \delta_h(\mathbf{x} - \mathbf{X}_k^n) \Delta s = 0, \tag{4.9}$$

$$\nabla_h \cdot \mathbf{u}^{n+1} = 0, \tag{4.10}$$

$$\nabla_{s_h} \cdot \mathbf{U}_k^{n+1} = \frac{\mathbf{U}_k^{n+1} - \mathbf{U}_{k-1}^{n+1}}{\Delta s} \cdot \boldsymbol{\tau}_{k-1/2}^n \Big/ \left|D_s \mathbf{X}^n\right|_{k-1/2} = 0, \tag{4.11}$$

$$\mathbf{U}_k^{n+1} = \sum_{\mathbf{x}} \mathbf{u}(\mathbf{x})^{n+1} \delta_h(\mathbf{x} - \mathbf{X}_k^n)\, h^2. \tag{4.12}$$

Once we obtain the new velocity field $\mathbf{u}^{n+1}$ on the fluid grid, we can interpolate the new velocity to the marker points by Eq. (4.12), and move the Lagrangian markers to new positions. That is,

$$\mathbf{X}_k^{n+1} = \mathbf{X}_k^n + \Delta t\, \mathbf{U}_k^{n+1}. \tag{4.13}$$

Therefore, we have

$$\frac{\mathbf{X}_k^{n+1} - \mathbf{X}_{k-1}^{n+1}}{\Delta s} = \frac{\mathbf{X}_k^n - \mathbf{X}_{k-1}^n}{\Delta s} + \Delta t \frac{\mathbf{U}_k^{n+1} - \mathbf{U}_{k-1}^{n+1}}{\Delta s}.$$

By multiplying the above equation by itself and using the zero discrete surface divergence (4.11), we obtain the following quality

$$\left|\frac{\mathbf{X}_k^{n+1} - \mathbf{X}_{k-1}^{n+1}}{\Delta s}\right|^2 = \left|\frac{\mathbf{X}_k^n - \mathbf{X}_{k-1}^n}{\Delta s}\right|^2 + (\Delta t)^2 \left|\frac{\mathbf{U}_k^{n+1} - \mathbf{U}_{k-1}^{n+1}}{\Delta s}\right|^2,$$

which leads to

$$|D_s \mathbf{X}^{n+1}|_{k-1/2}^2 = |D_s \mathbf{X}^n|_{k-1/2}^2 + (\Delta t)^2\, |D_s \mathbf{U}^{n+1}|_{k-1/2}^2. \tag{4.14}$$

Thus, we conclude the point-wise error for the local stretching factor is first-order accurate which is comparable to the accuracy of the IB method.

4.3 Implementation details

4.3.1 Discrete skew-adjoint operators

In this subsection, we show that the spreading operator S acting on the elastic tension and the surface divergence operator ∇_s acting on the velocity are also skew-adjoint in the discrete sense. That is, we shall prove the numerical identity for Eq. (4.8). To proceed, we first define the corresponding discrete inner product on the fluid grid Ω_h and the interfacial grid Γ_h in the following

$$\langle \mathbf{u}, \mathbf{v}\rangle_{\Omega_h} = \sum_{\mathbf{x}} \mathbf{u}(\mathbf{x})\cdot\mathbf{v}(\mathbf{x})\, h^2, \qquad \langle \phi, \psi\rangle_{\Gamma_h} = \sum_{k=1}^{M} \phi_{k-1/2}\, \psi_{k-1/2}\, |D_s \mathbf{X}|_{k-1/2}\, \Delta s, \quad (4.15)$$

where the second summation is nothing but the mid-point rule for the second integral of (4.7). We also define the discrete spreading operator S_h acting on the discrete elastic tension σ_h as

$$S_h(\sigma_h) = \sum_{k=0}^{M-1} D_s\, (\sigma \boldsymbol{\tau})_k\, \delta_h(\mathbf{x} - \mathbf{X}_k)\Delta s. \quad (4.16)$$

Then we have

$$\begin{aligned}
\langle S_h(\sigma_h), \mathbf{u}\rangle_{\Omega_h} &= \sum_{\mathbf{x}} \left(\sum_{k=0}^{M-1} D_s\, (\sigma\boldsymbol{\tau})_k\, \delta_h(\mathbf{x} - \mathbf{X}_k)\Delta s \right) \cdot \mathbf{u}(\mathbf{x})\, h^2 \\
&= \sum_{k=0}^{M-1} D_s\, (\sigma\boldsymbol{\tau})_k \cdot \left(\sum_{\mathbf{x}} \mathbf{u}(\mathbf{x}) \delta_h(\mathbf{x} - \mathbf{X}_k)\, h^2 \right) \Delta s = \sum_{k=0}^{M-1} D_s\, (\sigma\boldsymbol{\tau})_k \cdot \mathbf{U}_k \Delta s \\
&= \sum_{k=0}^{M-1} \frac{\sigma_{k+1/2}\, \boldsymbol{\tau}_{k+1/2} - \sigma_{k-1/2}\, \boldsymbol{\tau}_{k-1/2}}{\Delta s} \cdot \mathbf{U}_k \Delta s \\
&= -\sum_{k=1}^{M} \sigma_{k-1/2} \left(\frac{\mathbf{U}_k - \mathbf{U}_{k-1}}{\Delta s} \right) \cdot \boldsymbol{\tau}_{k-1/2}\, \Delta s \quad \text{(summation by parts)} \\
&= \langle \sigma_h, -\nabla_{s_h} \cdot \mathbf{U}\rangle_{\Gamma_h} = \langle \sigma_h, S_h^*(\mathbf{U})\rangle_{\Gamma_h}.
\end{aligned}$$

One should notice that this discrete skew-adjoint property is crucial to our IB formulation for solving Eqs. (4.9) to (4.11). Due to the fact that discrete surface divergence of the velocity is zero in Eq. (4.11), we can re-scale this constraint to make the resultant matrix obtained by Eq. (4.11) is the transpose of the resultant matrix obtained by the discrete spreading operator of the tension. One can also verify this symmetric property by expressing the terms explicitly. The detail is given in the Appendix B. We now are ready to write down the linear system of equations for Eqs. (4.9) to (4.11), and develop a numerical algorithm to solve the system.

4.3.2 Existence of the solution

By using the staggered grid for the discretization of fluid variables, it is well-known that the matrix obtained by the discrete divergence operator of the fluid velocity can be written as the transpose of the discrete gradient operator of the pressure. As discussed in previous subsection, the resultant matrix obtained by the discrete surface divergence of velocity can be written as the transpose of the matrix obtained by the discrete spreading operator of the tension. Thus, the linear system for Eqs. (4.9)-(4.11) is symmetric and can be written as

$$\begin{bmatrix} L & G & S \\ G^T & 0 & 0 \\ S^T & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ p \\ \sigma \end{bmatrix} = \begin{bmatrix} bc_1 \\ bc_2 \\ bc_3 \end{bmatrix}, \tag{4.17}$$

where the sub-matrix L, G and S are represented the discrete Laplacian $\mu\Delta_h$, discrete gradient $-\nabla_h$, and the discrete spreading operator S_h. The sub-matrix size of L, G and S are $((m-1)n+m(n-1))\times((m-1)n+m(n-1))$, $((m-1)n+m(n-1))\times(mn)$ and $(m-1)n+m(n-1)\times M$, respectively. The right-hand side vector $[bc_1, bc_2, bc_3]^T$ of Eq. (4.17) consists only of the velocity boundary conditions since the pressure does not need the boundary condition in staggered grid formulation.

Let us discuss the existence of the linear system of Eq. (4.17). From now on, we denote the matrix in (4.17) by A. As is known, without the effect of the inextensible interface, the linear system becomes pure Stokes flow as

$$\begin{bmatrix} L & G \\ G^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ p \end{bmatrix} = \begin{bmatrix} bc_1 \\ bc_2 \end{bmatrix}. \tag{4.18}$$

Let us denote the matrix in Eq. (4.18) by $\widetilde{A}$. It is also well-known that the nullity of $\widetilde{A}$ equals to one since the pressure is unique up to a constant, and the existence of a solution can be verified by using the discrete incompressible constraint (4.10). To be precisely, since the rank of deficiency of $\widetilde{A}$ is only one, based on the algebraic structure of the sub-matrix G, the kernel of $\widetilde{A}$ is

$$\ker(\widetilde{A}) = \operatorname{span}\{[\ \underbrace{0\cdots0}_{(m-1)n+m(n-1)}\ \ \underbrace{1\cdots1}_{mn}\ ]^T\}. \tag{4.19}$$

And for any vector $z \in \ker(\widetilde{A}^T) = \ker(\widetilde{A})$, we have

$$\begin{aligned} z^T \begin{bmatrix} bc_1 \\ bc_2 \end{bmatrix} &= \sum_{k=1}^{mn} (bc_2)_k = \sum_{j=1}^{n} \frac{u_{0,j}}{h} - \sum_{j=1}^{n} \frac{u_{m,j}}{h} + \sum_{i=1}^{m} \frac{v_{i,0}}{h} - \sum_{i=1}^{m} \frac{v_{i,n}}{h} \\ &= -\left(-\sum_{j=1}^{n} u_{0,j}h + \sum_{j=1}^{n} u_{m,j}h - \sum_{i=1}^{m} v_{i,0}h + \sum_{i=1}^{m} v_{i,n}h \right) h^{-2} \\ &= \left(\sum_{i=1}^{m} \sum_{j=1}^{n} \left(\frac{u_{i,j} - u_{i-1,j}}{h} + \frac{v_{i,j} - v_{i,j-1}}{h} \right) h^2 \right) h^{-2} \\ &= 0, \quad \text{(by the discrete incompressibility (4.10))} \end{aligned} \tag{4.20}$$

which shows the compatibility condition for the existence of a solution.

If the effect of the inextensible interface is added, the matrix $\widetilde{A}$ is augmented by S and S^T to become A as in (4.17). Since the matrix S comes from the discrete spreading operator of the tension, the entries of S depend on the number of moving Lagrangian markers, their positions and tangents. It is unlikely to show rigorously that the nullity of A is exactly equal to one; however, we have checked the above statement to be true in our numerical experiments. So the apparent kernel will be

$$\ker(A) = \operatorname{span}\{[\ \underbrace{0 \cdots 0}_{(m-1)n+m(n-1)}\ \underbrace{1 \cdots 1}_{mn}\ \underbrace{0 \cdots 0}_{M}\ ]^T\}, \tag{4.21}$$

and the existence of a solution for the linear system (4.17) follows the equality of (4.20) immediately.

4.3.3 Fractional step method

In this subsection, we follow the idea of fractional step method developed by Taira and Colonius [64] to solve the resultant linear system of equations (4.17). In [64], the authors applied the IB method to simulate the incompressible flow over solid bodies with prescribed body surface motion. Unlike the previous approaches in [13, 57, 30], they introduce the boundary force as another Lagrange multiplier to enforce the no-slip constraint for the velocity at the immersed boundary. From this point of view, the present approach shares the similar spirit as in [64] by introducing the elastic tension as a new Lagrange multiplier to enforce the surface divergence free constraint (4.11) along the interface. Since the pressure can be regarded as a Lagrange multiplier for the fluid divergence free constraint (4.10), one can group those two Lagrange multipliers as a new column vector $\phi = [p, \sigma]^T$ and combine the sub-matrices G and S as $Q = [G, S]$, the linear system (4.17) now becomes

$$\begin{bmatrix} L & Q \\ Q^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \phi \end{bmatrix} = \begin{bmatrix} bc_1 \\ \widetilde{b} \end{bmatrix}, \text{ where } \widetilde{b} = \begin{bmatrix} bc_2 \\ bc_3 \end{bmatrix}. \tag{4.22}$$

As in [64], we perform a block **LU** decomposition to the above equation to obtain

$$\begin{bmatrix} L & 0 \\ Q^T & -Q^TL^{-1}Q \end{bmatrix} \begin{bmatrix} I & L^{-1}Q \\ 0 & I \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \phi \end{bmatrix} = \begin{bmatrix} bc_1 \\ \widetilde{b} \end{bmatrix}. \tag{4.23}$$

Note that, the above decomposition is possible (L^{-1} exists) since the matrix L arising from the discrete Laplacian operator is symmetric and negative definite. This matrix decomposition is also referred to Uzawa method. One can further split the above matrix equation into following steps by introducing an intermediate velocity vector $\mathbf{u}^*$ as

$$L\mathbf{u}^* = bc_1, \tag{4.24}$$

$$(-Q^TL^{-1}Q)\phi = \widetilde{b} - Q^T\mathbf{u}^*, \tag{4.25}$$

$$\mathbf{u} = \mathbf{u}^* - L^{-1}Q\phi. \tag{4.26}$$

Recall that the original matrix denoted by A in Eq. (4.17) is singular due to the pressure value is unique up to a constant, thus the singularity cannot be removed from applying the block LU decomposition. In fact, the matrix $(-Q^TL^{-1}Q)$ in Eq. (4.25) is symmetric and singular with rank $mn+M-1$ since G is singular. We now provide an existence of solution for Eq. (4.25) as follows.

For any vector $y \in \ker((-Q^TL^{-1}Q)^T) = \ker(-Q^TL^{-1}Q)$, then the vector y satisfies $y^T(-Q^TL^{-1}Q)y = (Qy)^T(-L^{-1})(Qy) = 0$. This implies that $Qy = 0$ since $-L^{-1}$ is a positive definite matrix; thus, $y \in \ker(Q)$. By writing $L \cdot 0 + Qy = 0$, we can immediately obtain the vector $[0\ y]^T \in \ker(A) = \ker(A^T)$. Notice that, such y also satisfies $y^T(\widetilde{b} - Q^T\mathbf{u}^*) = 0$ since $y^T\widetilde{b} = 0$ due to the discrete incompressible constraint (4.10). Therefore, the right-hand side vector of Eq. (4.25) belongs to the range of the matrix $(-Q^TL^{-1}Q)$ in which a solution exists. One can also immediately see from the structure of sub-matrix G that the solution in Eq. (4.25) is unique up to a constant.

Now we are ready to describe the detailed numerical implementation for solving Eqs. (4.24)-(4.26). It is a common practice to avoid the direct computation of the inverse of the matrix L since it is too expensive. In [64], a second-order approximation for L^{-1} based on Taylor expansion is implemented for solving similar equations as our Eqs. (4.25)-(4.26), and conjugate gradient method is applied to solve those equations iteratively. However, this leads to another time step constraint related to the viscosity and the eigenvalues of the discrete Laplacian. In this chapter, since we are working on the Stokes flow rather than the Navier-Stokes, we are unable to approximate L^{-1} using Taylor's expansion. Although we do not approximate the L^{-1} directly, we still can solve Eqs. (4.25)-(4.26) efficiently thanks to the fast Poisson solver developed in public software package FISHPACK [1]. (The present matrix L is nothing but the discrete Laplacian operator.) The detailed steps for solving Eqs. (4.24)-(4.26) are as follows.

Step 1. Solve Eq. (4.24) by two fast Poisson solvers to obtain intermediate velocity field $\mathbf{u}^*$.

Step 2. Since the matrix $(-Q^T L^{-1} Q)$ is symmetric and positive semi-definite, the conjugate gradient method can be applied. In each iteration, a matrix-vector product $(-Q^T L^{-1} Q)\,\varphi$ is needed; fortunately, this can be done by letting $z = L^{-1}Q\varphi$, and solving $Lz = Q\varphi$. Once it is done, we multiply z by $-Q^T$ to obtain the product needed. Again, the time-consuming cost in each iteration is one fast Poisson solver.

Step 3. Find the velocity field $\mathbf{u}$ from Eq. (4.26). Since ϕ is solved via Step 2, by solving $Lw = Q\phi$, we then obtain $\mathbf{u} = \mathbf{u}^* - w$. Again, this involves applying one fast Poisson solver.

Therefore, the overall cost in Step 1-3 for our present numerical algorithm can be counted in terms of the number of fast Poisson solver applied. In next section, we shall show the numbers of fast Poisson solver used in the Stokes flow for different grid resolutions.

4.4 Numerical results

In this section, we perform a series of numerical tests for the present scheme. We first provide the convergence and efficiency tests for the Stokes flow without interfaces. We then perform the convergence test for the Stokes flow with an inextensible interface. Finally, we simulate single interface problems in a shear flow to mimic the vesicle problems in [28] without the bending effect. As expected, one can still see the tank-treading motion along the interface. Throughout this section, the computational domain is chosen as $\Omega = [-1, 1] \times [-1, 1]$.

4.4.1 Convergence and efficiency tests

In this subsection, we perform the convergence test and evaluate the efficiency for the present Stokes solver without an interface. The numerical algorithm for solving this problem is exactly same as Step 1-3 described in previous section but with a simpler version; that is, $Q = G$ and $\phi = p$. The different efficient Stokes solver can be found in [55, 2, 53] and the references therein.

Here, we use the following analytic solution so we can easily compute the errors between the exact and the numerical solutions.

$$u_e(x, y) = \sin x \cos y, \quad v_e(x, y) = -\cos x \sin y \quad \text{and} \quad p_e(x, y) = e^x \sin y.$$

Note that, the above solution does not satisfy the pure Stokes equations so we need to add some external force field (can be easily computed) into the equations. However,

$m = n$	32	64	128	256	512
$\|u_e - u_h\|_\infty$	1.578e-4	4.481e-5	1.206e-5	3.153e-6	8.120e-7
rate	-	1.82	1.89	1.94	1.96
$\|v_e - v_h\|_\infty$	1.578e-4	4.481e-5	1.206e-5	3.153e-6	8.120e-7
rate	-	1.82	1.89	1.94	1.96
$\|p_e - p_h\|_\infty$	9.615e-4	4.286e-4	2.052e-4	1.005e-4	4.970e-5
rate	-	1.17	1.06	1.03	1.02

Table 4.1: Numerical accuracy of Stokes solver.

$m = n$	32	64	128	256	512
iterations	12	14	15	16	18
CPU time(sec)	0.02	0.05	0.20	0.93	4.97

Table 4.2: The cost of CPU time and iterations.

it does not change the method or algorithm since the extra force term appears in the right-hand side of equations. Along the boundary of computational domain, the Dirichlet boundary conditions for the velocity are provided while no pressure boundary condition is needed in our setting.

It is also worth mentioning that the pressure is unique up to a constant in Stokes equations. Rather than pinning a certain value to a particular discrete pressure as in [64], the uniqueness can be guaranteed by setting up a constraint for the discrete pressure as

$$\sum_{i,j} p_{i,j}\, h^2 = \int_\Omega p_e(\mathbf{x})\, d\mathbf{x}. \tag{4.27}$$

So it is straightforward that our initial guess p^0_{ij} in conjugate gradient iteration can be chosen as $p^0_{ij} = \int_\Omega p_e(\mathbf{x})\, d\mathbf{x}/|\Omega|$. By using the mathematical induction, one can easily show that during the conjugate gradient iteration, the discrete pressure does satisfy the constraint (4.27). In those tests, the tolerance of residual is chosen as 10^{-8}.

Table 4.1 shows the maximum errors between the exact and numerical solutions for different grid resolutions. One can see that the velocity field has clean second order accuracy while the pressure has clean first-order accuracy. Table 4.2 shows the efficiency of present Stokes solver. One can see that the number of conjugate gradient iterations increase slightly even we double the grid sizes and the CPU time for 512 × 512 mesh is just a few seconds.

4.4.2 Convergence test for an inextensible interface

In this subsection, we perform the convergence study for the present numerical algorithm to the Stokes flow with an inextensible interface. Here, we put an inextensible interface Γ with initial configuration $(X(s), Y(s)) = (0.2\cos(s), 0.5\sin(s))$ under a shear flow $(u, v) = (\gamma\, y, 0)$ in a fluid domain Ω, see Figure 4.1 in detail. The shear rate γ is chosen to be $\gamma = 1$ and the fluid viscosity is $\mu = 1$. We choose the different mesh sizes as $m = n = 64, 128, 256, 512$ so the corresponding mesh is $h = 2/m$. We also set the Lagrangian mesh width as $\Delta s \approx h/2$ and the time step duration for interface evolution as $\Delta t = h/4$.

Since the analytical solution is not available in this test, we choose the result obtained from the finest mesh $m = n = 512$ as our reference solution, and compute the maximum error between the reference solution and the numerical solution. All the numerical solutions are computed up to time $T = 0.5$. Since the interface is inextensible and the fluid is incompressible, the perimeter of the interface and the enclosed area by the interface should remain constants theoretically as time goes on. Let L_0 and L_h be the perimeters of the interface at the initial time and final time $T = 0.5$, respectively. The relative error of the perimeter is defined as $|L_h - L_0|/L_0$, and the relative error of the area enclosed is $|A_h - A_0|/A_0$. Table 4.3 shows the relative errors of the perimeter, the area of the region bounded by the interface, the maximum error of the interface configuration, and the maximum error for the fluid velocity field. Note that, the fluid variables are defined at the staggered grid so when we refine the mesh, the numerical solutions will not coincide with the same grid locations. In these runs, we simply use a linear interpolation to compute the solutions at the desired locations. Due to the fact that the IB formulation has the singular forcing term in the equations, regularizing the singular term by smoothing discrete delta function causes the method to be first-order accurate. The numerical results shown in Table

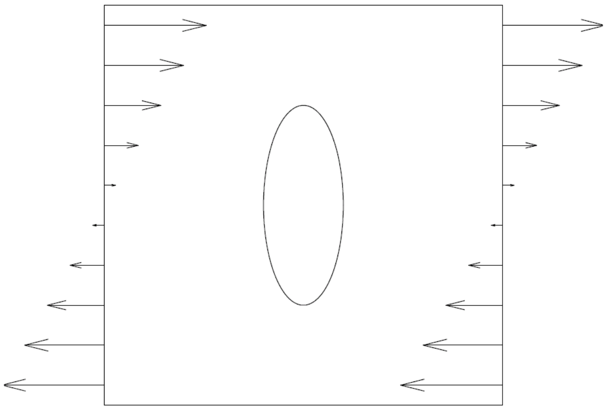


Figure 4.1: A diagram of an inextensible interface in a shear flow.

	$m = n = 64$	$m = n = 128$	rate	$m = n = 256$	rate
$\lvert L_h - L_0\rvert / L_0$	1.349e-03	7.201e-04	0.91	3.364e-04	1.10
$\lvert A_h - A_0\rvert / A_0$	9.069e-04	4.132e-04	1.14	2.010e-04	1.04
$\lVert \mathbf{X}_h - \mathbf{X}_{\text{ref}}\rVert_\infty$	6.878e-03	2.498e-03	1.46	7.808e-04	1.68
$\lVert u_h - u_{\text{ref}}\rVert_\infty$	4.110e-02	1.675e-02	1.30	5.589e-03	1.46
$\lVert v_h - v_{\text{ref}}\rVert_\infty$	4.050e-02	1.672e-02	1.28	6.086e-03	1.58

Table 4.3: The mesh refinement results for the perimeter of the interface L_h, the enclosed area A_h, the interface configuration $\mathbf{X}_h$, and the velocity u_h and v_h.

4.3 are consistent with what we expect from theory.

4.4.3 Tank-treading motion under shear flow

Unlike the previous subsection that we focus on the numerical convergence test for our present scheme, in this subsection, we consider the physical transient deformation of an inextensible interface subject to a simple shear flow. As mentioned before, the motivation of this test is to mimic the simulation of the vesicle dynamics which has a lot of applications in bio-fluid problems.

As in previous test, we put an inextensible interface Γ with initial configuration $(X(s), Y(s)) = (0.18\cos(s), 0.5\sin(s))$ under a shear flow $(u, v) = (\gamma y, 0)$ in a fluid domain $\Omega = [-1, 1] \times [-1, 1]$, see Figure 4.1 in detail. The shear rate γ is chosen to be $\gamma = 1$ and the fluid viscosity is $\mu = 1$. The mesh used is 128×128 and the residual tolerance for is 10^{-4}. It is worth mentioning that the elastic tension σ computed in previous numerical experiments [71, 28, 35] tends to oscillate along the interface which makes the conjugate gradient method for solving in Eq. (4.25) difficult to converge if the residual tolerance is too large. It will be more appealing if we can find a good preconditioner for the matrix in Eq. (4.25) so that PCG method converges faster. This issue should be investigated further in later work. Figure 4.2 shows the interface configuration at three different times $t = 0.0625, 1.25, 3.125$, while Figure 4.3 shows the corresponding elastic tension σ plotted counterclockwise along the interface. (The starting point of zero length in Figure 4.3 is marked by x in Figure 4.2.) One can indeed see some slight oscillations of the tension along the interface as seen in previous literature [71, 28, 35]. Furthermore, one can see that the tension has smallest values at the interface positions where the curvatures are largest (both tips) which also in agreement with those in previous literature.

To be more physically realistic, we now run the problem to a longer time. It is well-known that the equilibrium dynamics of inextensible interface or vesicle under a simple shear flow undergoes a tanking-treading motion if the viscosity contrast under a certain threshold [24]. Here, by tank-treading motion we mean that the

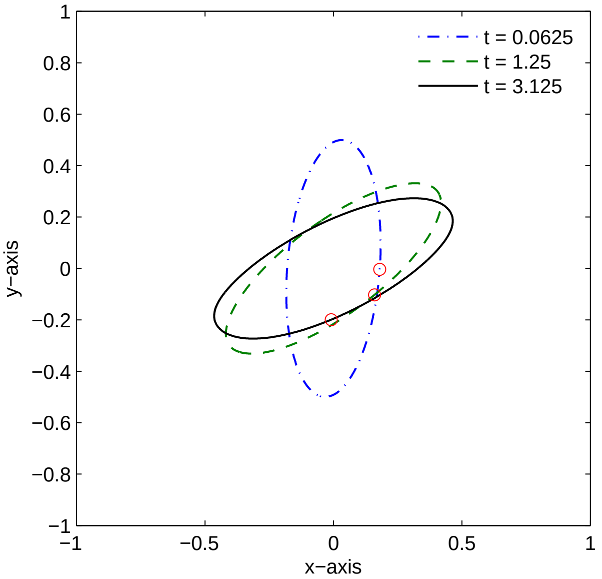


Figure 4.2: The motion of an inextensible interface in a shear flow with initial configuration $(X(s), Y(s)) = (0.18\cos(s), 0.5\sin(s))$.

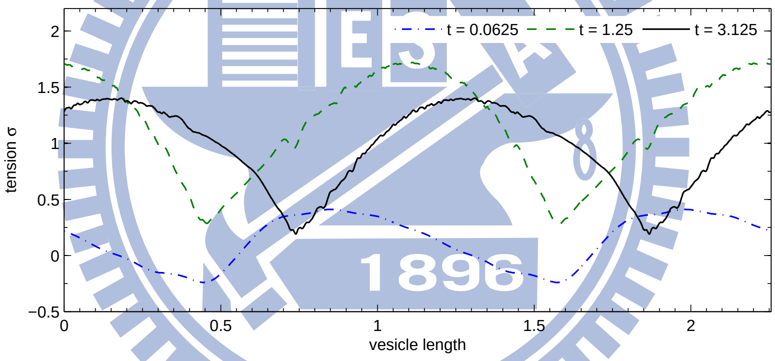


Figure 4.3: The plots of tension σ along the interface.

configuration of the interface or vesicle remains stationary while there is a tangential motion along the interface. Figure 4.4 shows the evolutional motion of the interface (bi-concave shape initially) at different times. We can see that after some time, the interface shape does not seem to change at all; however, the Lagrangian point along the interface marked by ”$*$” moves along with its tangential velocity. The streamlines at three different chosen times are shown in Figure 4.5 in which we can see no normal motion in equilibrium. This tank-treading motion is good agreement in previous studies [71, 66, 28, 35].

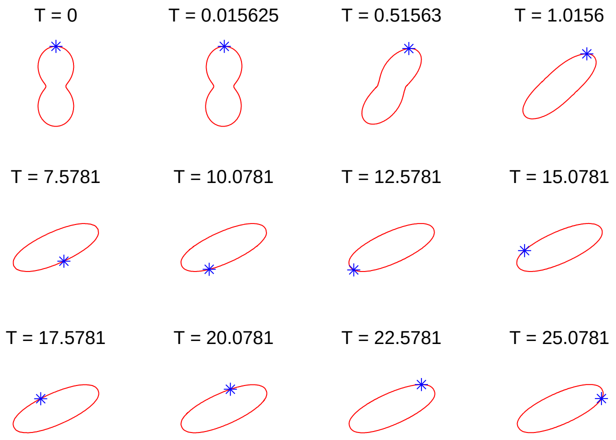


Figure 4.4: The tank-treading motion of an inextensible interface under a shear flow.

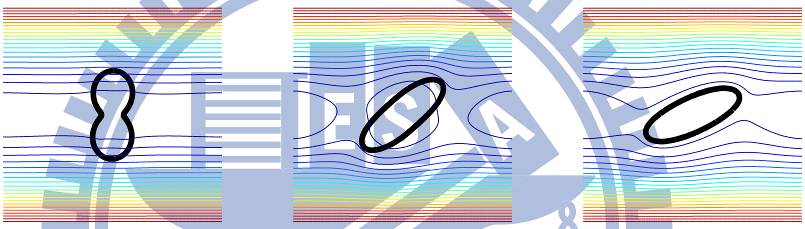


Figure 4.5: The streamlines of the flow along with an interface.

As discussed in previous literature [71, 26, 25, 66, 28], the motion of a steady or an equilibrium inextensible interface (or vesicle) can be characterized by both the inclination angle θ between the long axis of interface and the flow direction, and the tank-treading frequency $f = 2\pi / \int_\Gamma \frac{dl}{u_\tau}$ of the revolution, where u_τ is the tangential velocity component. The inclination angle has been founded to be strongly dependent on the reduced area ν. However, the angle is independent of the shear rate γ. This behavior has been verified in the left panel of Figure 5.2 and is in good agreement with previous studies [71, 26, 25, 66, 28] which shows the steady inclination angle (θ/π) versus the reduced area (ν) with different shear rates $\gamma = 1, 5, 10$. We have observed that the inclination angle increases with the reduced area but is nearly independent of the shear rate. The right panel of Figure 5.2 shows that the tank-treading frequency f versus the reduced area ν for different shear rates. One can see that as the shear

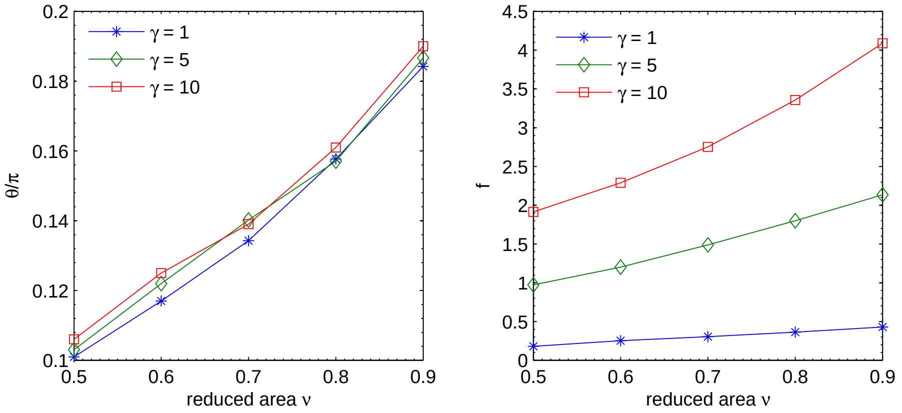


Figure 4.6: The inclination angles θ/π (left) and the tank-treading frequency $f = 2\pi/\int_\Gamma \frac{\mathrm{d}l}{\mathbf{u}\cdot\boldsymbol{\tau}}$ (right) versus reduced areas ν with different shear rates for the tank-treading motions of an inextensible interface in a shear flow.

rate increases, the tangential motion becomes stronger; thus, the frequency becomes larger. Moreover, by fixing the shear rate, if the interface has larger reduced area then it has larger frequency as well. Again, our numerical results are in a good agreement with previous studies in literature.

Chapter 5

Unconditionally energy stable immersed boundary method

In this chapter, we propose an unconditionally stable IB method to simulate vesicle dynamics. The vesicle is assumed to be nearly inextensible, that is, the magnitude of time changing of stretching factor is not exactly zero but quite small. Under this penalty method, the tension of vesicle has an explicit form so we do not need to solve fluid variables and surface tension simultaneously (differ form previous chapter). However, since the cost of penalty method contributes to interfacial stiffness, there is a harsh restriction on choosing time step Δt to maintain numerical stability.

As known in literature [62, 38, 58, 50], the IB method suffers a time step restriction to maintain numerical stability. This restriction becomes more stringent when the elastic force is stiff and the force spreading occurs at the beginning of each time step (an explicit scheme). Note that, such time step restriction cannot be elevated even the fluid solver is discretized in a semi-implicit manner (the explicit differencing of the advection term and implicit differencing of the diffusion term). Rather than performing the force spreading at the beginning of the time step, one might consider to perform the procedure at the intermediate (semi-implicit scheme) or even at the end of time step (implicit scheme). There have been many attempts to reduce the stiffness or to overcome this difficulty of time step restriction in the past decade [44, 46, 20, 21, 6, 7, 16]. However, there is always a trade-off between the stability and efficiency of those algorithms. In this chapter, we shall propose a new semi-implicit scheme that can be solved quite efficiently (the resultant linear system is symmetric positive definite) and the time step size can be significantly increased.

This chapter is organized as follows. In Section 5.1, we introduce the formulation of incompressible Navier-Stokes equations with an ordinary immersed elastic interface (tension has explicit formulation). We then develop semi-implicit IB schemes based on the projection method for the fluid solver in Section 5.2. We also show those developed schemes to be unconditionally energy stable. Then we modify those semi-implicit schemes to be more efficient so that the resultant linear system is symmetric positive definite. Finally we apply present method to simulation of vesicle dynamics, the numerical results are given in Section 5.6.

5.1 Governing equations

We begin by stating the mathematical formulation of the Navier-Stokes flow with an immersed boundary (or an interface). We consider a moving, immersed, elastic boundary $\Gamma(t)$ which exerts forces into an incompressible fluid in a fixed fluid domain Ω. We assume that the fluids inside and outside of the boundary are the same so the governing equations in immersed boundary formulation can be written as follows.

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u}\right) + \nabla p = \mu\Delta\mathbf{u} + \int_\Gamma \mathbf{F}(s,t)\delta(\mathbf{x}-\mathbf{X}(s,t))\,\mathrm{d}s \ \text{ in } \Omega, \tag{5.1}$$

$$\nabla\cdot\mathbf{u} = 0 \ \text{ in } \Omega, \tag{5.2}$$

$$\frac{\partial \mathbf{X}}{\partial t}(s,t) = \mathbf{U}(s,t) = \int_\Omega \mathbf{u}(\mathbf{x},t)\delta(\mathbf{x}-\mathbf{X}(s,t))\,\mathrm{d}\mathbf{x} \ \text{ on } \Gamma. \tag{5.3}$$

Eqs. (5.1) and (5.2) are the familiar incompressible Navier-Stokes equations with a singular force arising from the immersed boundary. Eq. (5.3) simply represents that the immersed boundary moves along with the local fluid velocity (the interfacial velocity). Here, the interfacial velocity $\mathbf{U}$ is simply an interpolation of the fluid velocity at the immersed boundary. The interaction between the fluid and the immersed boundary is linked by the two-dimensional Dirac delta function $\delta(\mathbf{x}) = \delta(x)\delta(y)$. In this chapter, the elastic force mainly comes from the tension which satisfies the Hooke's law as

$$\mathbf{F}(s,t) = \frac{\partial}{\partial s}(\sigma\boldsymbol{\tau}), \quad \sigma = \sigma_0(|\mathbf{X}_s| - r_0), \tag{5.4}$$

where σ_0 is the elastic coefficient, and r_0 is the rest length. Note that, the rate of local stretching factor can be derived to satisfy the following equation [33]

$$\frac{\partial}{\partial t}|\mathbf{X}_s| = (\nabla_s\cdot\mathbf{U})\,|\mathbf{X}_s| = \frac{\partial \mathbf{U}}{\partial s}\cdot\boldsymbol{\tau}. \tag{5.5}$$

5.2 Numerical algorithm

Let Δt be the time step size and the superscript index be the time step level. At the beginning of each time level n, the boundary configuration $\mathbf{X}_k^n$, and the unit tangent $\boldsymbol{\tau}_{k-1/2}^n$ are all given. To simplify our notations, we define the discrete spreading operator acting on the tension and discrete surface divergence operator acting on the velocity as follows.

$$\mathcal{S}_h^m[\sigma](\mathbf{x}) = \mathcal{S}_h^m[\sigma_0(|\mathbf{X}_s|-r_0)](\mathbf{x}) = \sum_{k=1}^{M} D_s\left(\sigma_0(|\mathbf{X}_s|-r_0)\boldsymbol{\tau}^m\right)_k \delta_h(\mathbf{x}-\mathbf{X}_k^m)\Delta s, \quad (5.6)$$

$$\mathcal{T}_h^m[\mathbf{u}](s_{k-1/2}) = \frac{\mathbf{U}_k-\mathbf{U}_{k-1}}{\Delta s}\cdot\boldsymbol{\tau}^m_{k-1/2} \quad \mathbf{U}_k = \sum_{\mathbf{x}} \mathbf{u}(\mathbf{x})\delta_h(\mathbf{x}-\mathbf{X}_k^m)h^2. \quad (5.7)$$

The superscript m stands for the information of immersed boundary position adopted at time level $m\Delta t$. Note that, the above discrete operators are skew-adjoint as $\langle \mathcal{S}_h^m[\sigma], \mathbf{u}\rangle_{\Omega_h} = \langle \sigma, -\mathcal{T}_h^m[\mathbf{u}]\rangle_{\Gamma_h}$. It is also worthy mentioning that the skew-adjointness property plays an important role in our energy stability analysis later.

In present chapter, our numerical discretization and energy stability analysis are based on unsteady Stokes equations instead of Navier-Stokes. For the latter, the nonlinear advection term can be treated explicitly during the time evolution with moderate CFL condition. Alternatively, one can split the Navier-Stoke equations into an advection part and unsteady Stokes part in which the advection equation is solved by an alternating direction implicit (ADI) method to maintain the unconditionally numerical stability [21].

Here we introduce two time-integration schemes; namely, the backward Euler (BE) and Crank-Nicholson (CN) scheme in conjunction with the projection method [31] as follows.

- Backward Euler (BE) scheme

$$\rho\frac{\mathbf{u}^*-\mathbf{u}^n}{\Delta t} = \mu\Delta_h\mathbf{u}^* + \mathcal{S}_h^n[\sigma_0(|\mathbf{X}_s|^{n+1}-r_0)] \quad \text{in } \Omega_h \quad (5.8)$$

$$\Delta_h\phi = \frac{1}{\Delta t}\nabla_h\cdot\mathbf{u}^* \quad \text{in } \Omega_h; \quad \left.\frac{\partial\phi}{\partial\mathbf{n}}\right|_{\partial\Omega_h} = 0, \quad (5.9)$$

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \Delta t\nabla_h\phi \quad \text{in } \Omega_h \quad (5.10)$$

$$\nabla_h p^{n+1} = \rho\nabla_h\phi - \mu\nabla_h(\nabla_h\cdot\mathbf{u}^*) \quad \text{in } \Omega_h \quad (5.11)$$

$$\frac{|\mathbf{X}_s|^{n+1}-|\mathbf{X}_s|^n}{\Delta t} = \mathcal{T}_h^n[\mathbf{u}^{n+1}] \quad \text{on } \Gamma_h \quad (5.12)$$

$$\mathbf{U}_k^{n+1} = \sum_{\mathbf{x}} \mathbf{u}(\mathbf{x})^{n+1}\delta_h(\mathbf{x}-\mathbf{X}_k^n)\, h^2 \quad \text{on } \Gamma_h, \quad (5.13)$$

$$\mathbf{X}_k^{n+1} = \mathbf{X}_k^n + \Delta t\,\mathbf{U}_k^{n+1} \quad \text{on } \Gamma_h. \quad (5.14)$$

- Crank-Nicholson (CN) scheme

$$\rho\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} = \frac{\mu}{2}\Delta_h(\mathbf{u}^* + \mathbf{u}^n) + \mathcal{S}_h^n\left[\sigma_0\left(\frac{|\mathbf{X}_s|^{n+1} + |\mathbf{X}_s|^n}{2} - r_0\right)\right] \quad \text{in } \Omega_h \tag{5.15}$$

$$\Delta_h\phi = \frac{1}{\Delta t}\nabla_h\cdot\mathbf{u}^* \quad \text{in } \Omega_h \quad \left.\frac{\partial\phi}{\partial\mathbf{n}}\right|_{\partial\Omega_h} = 0, \tag{5.16}$$

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \Delta t\nabla_h\phi \quad \text{in } \Omega_h \tag{5.17}$$

$$\nabla_h p^{n+1/2} = \rho\nabla_h\phi - \mu\nabla_h(\nabla_h\cdot\mathbf{u}^*) \quad \text{in } \Omega_h \tag{5.18}$$

$$\frac{|\mathbf{X}_s|^{n+1} - |\mathbf{X}_s|^n}{\Delta t} = \mathcal{T}_h^n[(\mathbf{u}^{n+1} + \widehat{\mathbf{u}}^n)/2] \quad \text{on } \Gamma_h \tag{5.19}$$

$$\mathbf{U}_k^{n+1} = \sum_{\mathbf{x}}\mathbf{u}(\mathbf{x})^{n+1}\delta_h(\mathbf{x}-\mathbf{X}_k^n)\,h^2, \quad \widehat{\mathbf{U}}_k^n = \sum_{\mathbf{x}}\mathbf{u}(\mathbf{x})^n\delta_h(\mathbf{x}-\mathbf{X}_k^n)\,h^2 \quad \text{on } \Gamma_h, \tag{5.20}$$

$$\mathbf{X}_k^{n+1} = \mathbf{X}_k^n + \Delta t(\mathbf{U}_k^{n+1} + \widehat{\mathbf{U}}_k^n)/2 \quad \text{on } \Gamma_h. \tag{5.21}$$

The spatial operators ∇_h and Δ_h are the standard second-order central difference approximations to the gradient and Laplacian. Here, δ_h is a smoother version of discrete delta function developed in [70]. Notice that, unlike the explicit scheme, both backward Euler and Crank-Nicholson schemes treat the stretching factor $|\mathbf{X}_s|^{n+1}$ as an unknown solution which is closed by the discretization of Eq. (6.14). Thus, in terms of immersed boundary force computation, the above two schemes are semi-implicit.

5.3 Energy stability analysis

In this subsection, we perform the energy stability analysis for our present schemes. We shall show that both methods are unconditionally stable in the sense that total energy is decreasing. Here, we follow the similar energy analysis proposed in [44, 21].

To proceed, we define the kinetic energy K and potential energy P of the system as follows.

$$K = \frac{\rho}{2}\|\mathbf{u}\|_{\Omega_h}^2 = \frac{\rho}{2}\langle\mathbf{u},\mathbf{u}\rangle_{\Omega_h}, \qquad P = \frac{\sigma_0}{2}\|\,|\mathbf{X}_s| - r_0\|_{\Gamma_h}^2 = \frac{\sigma_0}{2}\langle|\mathbf{X}_s| - r_0, |\mathbf{X}_s| - r_0\rangle_{\Gamma_h}, \tag{5.22}$$

where the associated discrete inner products are defined as

$$\langle\mathbf{u},\mathbf{v}\rangle_{\Omega_h} = \sum_{\mathbf{x}}\mathbf{u}(\mathbf{x})\cdot\mathbf{v}(\mathbf{x})\,h^2, \qquad \langle\phi,\psi\rangle_{\Gamma_h} = \sum_{k=1}^{M}\phi_{k-1/2}\,\psi_{k-1/2}\,\Delta s, \tag{5.23}$$

respectively. Thus, the total energy is $E = K + P$.

Theorem 1. *The backward Euler (BE) scheme of Eqs. (5.8)-(5.14) is unconditionally energy stable; that is, the scheme satisfies $E^{n+1} \leq E^n$ for each time step n.*

Proof: We first consider the stability analysis for BE scheme. For convenience, let us denote the singular force term in Eq.(5.8) by $\mathcal{S}_h^n[\sigma^{n+1}]$. We first substitute $\mathbf{u}^*$ in Eq. (5.10) into Eq. (5.8) to get the equation for $\rho(\mathbf{u}^{n+1} - \mathbf{u}^n)$, and then take the discrete inner product with $\mathbf{u}^{n+1} + \mathbf{u}^n$ to obtain

$$
\begin{aligned}
&K^{n+1} - K^n = \frac{\rho}{2}\langle \mathbf{u}^{n+1}, \mathbf{u}^{n+1}\rangle_{\Omega_h} - \frac{\rho}{2}\langle \mathbf{u}^{n}, \mathbf{u}^{n}\rangle_{\Omega_h} \\
&= \frac{\rho}{2}\langle \mathbf{u}^{n+1} + \mathbf{u}^n, \mathbf{u}^{n+1} - \mathbf{u}^n\rangle_{\Omega_h} \\
&= \frac{\rho}{2}\left(-\langle \mathbf{u}^{n+1} - \mathbf{u}^n, \mathbf{u}^{n+1} - \mathbf{u}^n\rangle_{\Omega_h} + 2\langle \mathbf{u}^{n+1}, \mathbf{u}^{n+1} - \mathbf{u}^n\rangle_{\Omega_h}\right) \\
&= -\frac{\rho}{2}\|\mathbf{u}^{n+1} - \mathbf{u}^n\|^2_{\Omega_h} + \langle \mathbf{u}^{n+1}, \rho\left(\mathbf{u}^{n+1} - \mathbf{u}^n\right)\rangle_{\Omega_h} \\
&= -\frac{\rho}{2}\|\mathbf{u}^{n+1} - \mathbf{u}^n\|^2_{\Omega_h} + \Delta t\langle \mathbf{u}^{n+1}, -\rho\nabla_h\phi + \mu\Delta_h\mathbf{u}^{n+1} + \mu\Delta t\nabla_h\Delta_h\phi + \mathcal{S}_h^n[\sigma^{n+1}]\rangle_{\Omega_h} \\
&= -\frac{\rho}{2}\|\mathbf{u}^{n+1} - \mathbf{u}^n\|^2_{\Omega_h} - \Delta t\rho\langle \mathbf{u}^{n+1}, \nabla_h\phi\rangle_{\Omega_h} + \mu\Delta t\langle \mathbf{u}^{n+1}, \Delta_h\mathbf{u}^{n+1}\rangle_{\Omega_h} \\
&+ \mu\Delta t^2\langle \mathbf{u}^{n+1}, \nabla_h\Delta_h\phi\rangle_{\Omega_h} + \Delta t\langle \mathbf{u}^{n+1}, \mathcal{S}_h^n[\sigma^{n+1}]\rangle_{\Omega_h}
\end{aligned}
$$

The successive difference of potential energy can be obtained as

$$
\begin{aligned}
&P^{n+1} - P^n = \frac{\sigma_0}{2}\left(\langle |\mathbf{X}_s|^{n+1} - r_0, |\mathbf{X}_s|^{n+1} - r_0\rangle_{\Gamma_h} - \langle |\mathbf{X}_s|^{n} - r_0, |\mathbf{X}_s|^{n} - r_0\rangle_{\Gamma_h}\right) \\
&= \frac{\sigma_0}{2}\langle |\mathbf{X}_s|^{n+1} + |\mathbf{X}_s|^{n} - 2r_0, |\mathbf{X}_s|^{n+1} - |\mathbf{X}_s|^{n}\rangle_{\Gamma_h} \\
&= \frac{\sigma_0}{2}\langle 2\,|\mathbf{X}_s|^{n+1} - 2r_0 - \Delta t\mathcal{T}_h^n[\mathbf{u}^{n+1}], \Delta t\mathcal{T}_h^n[\mathbf{u}^{n+1}]\rangle_{\Gamma_h} \;\; \left(\text{use } |\mathbf{X}_s|^{n} = |\mathbf{X}_s|^{n+1} - \Delta t\mathcal{T}_h^n[\mathbf{u}^{n+1}]\right) \\
&= \langle \sigma_0(|\mathbf{X}_s|^{n+1} - r_0) - \frac{\sigma_0\Delta t}{2}\mathcal{T}_h^n[\mathbf{u}^{n+1}], \Delta t\mathcal{T}_h^n[\mathbf{u}^{n+1}]\rangle_{\Gamma_h} \\
&= \Delta t\langle \sigma^{n+1}, \mathcal{T}_h^n[\mathbf{u}^{n+1}]\rangle_{\Gamma_h} - \frac{\sigma_0\Delta t^2}{2}\|\mathcal{T}_h^n[\mathbf{u}^{n+1}]\|^2_{\Gamma_h}.
\end{aligned}
$$

Thus, the total energy between two successive time steps can be written as

$$
\begin{aligned}
&E^{n+1} - E^n \\
&= -\frac{\rho}{2}\|\mathbf{u}^{n+1} - \mathbf{u}^n\|^2_{\Omega_h} - \Delta t\rho\langle \mathbf{u}^{n+1}, \nabla_h\phi\rangle_{\Omega_h} + \mu\Delta t\langle \mathbf{u}^{n+1}, \Delta_h\mathbf{u}^{n+1}\rangle_{\Omega_h} + \mu\Delta t^2\langle \mathbf{u}^{n+1}, \nabla_h\Delta_h\phi\rangle_{\Omega_h} \\
&+ \Delta t\langle \mathcal{S}_h^n[\sigma^{n+1}], \mathbf{u}^{n+1}\rangle_{\Omega_h} + \Delta t\langle \sigma^{n+1}, \mathcal{T}_h^n[\mathbf{u}^{n+1}]\rangle_{\Gamma_h} - \frac{\sigma_0\Delta t^2}{2}\|\mathcal{T}_h^n[\mathbf{u}^{n+1}]\|^2_{\Gamma_h}.
\end{aligned}
$$

The second and fourth terms vanish due to the orthogonality of discrete divergence-free velocity and the gradient, and the commutativity of ∇_h and Δ_h. The fifth and sixth terms are cancelled out due to both discrete operators are skew-adjoint as $\langle \mathcal{S}_h^n[\sigma], \mathbf{u}\rangle_{\Omega_h} = \langle \sigma, -\mathcal{T}_h^n[\mathbf{u}]\rangle_{\Gamma_h}$. The third term is always negative by the negative definiteness of discrete Laplace operator. Therefore, we have

$$E^{n+1} - E^n = -\frac{\rho}{2}\|\mathbf{u}^{n+1} - \mathbf{u}^n\|^2_{\Omega_h} - \mu\Delta t\|\nabla_h \mathbf{u}^{n+1}\|^2_{\Omega_h} - \frac{\sigma_0 \Delta t^2}{2}\|\mathcal{T}_h^n[\mathbf{u}^{n+1}]\|^2_{\Gamma_h}.$$

Thus, the total energy is decreasing which shows the present BE scheme is unconditionally energy stable.

Theorem 2. *The Crank-Nicholson (CN) scheme of Eqs. (5.15)-(5.21) is unconditionally energy stable; that is, the scheme satisfies $E^{n+1} \le E^n$ for each time step n.*

Proof: For the CN scheme, one can also do similar stability analysis. Again, let us denote the singular force term in Eq.(5.15) by $\mathcal{S}_h^n[(\sigma^{n+1} + \sigma^n)/2]$. By substituting $\mathbf{u}^*$ in Eq. (5.17) into Eq. (5.15) to get the term $\rho(\mathbf{u}^{n+1} - \mathbf{u}^n)$ and then taking the discrete inner product with $\mathbf{u}^{n+1} + \mathbf{u}^n$, we obtain the successive difference of kinetic energy as

$$\begin{aligned} K^{n+1} - K^n &= \frac{\rho}{2}\langle \mathbf{u}^{n+1}, \mathbf{u}^{n+1}\rangle_{\Omega_h} - \frac{\rho}{2}\langle \mathbf{u}^n, \mathbf{u}^n\rangle_{\Omega_h} \\ &= \frac{\rho}{2}\langle \mathbf{u}^{n+1} + \mathbf{u}^n, \mathbf{u}^{n+1} - \mathbf{u}^n\rangle_{\Omega_h} \\ &= \frac{\Delta t}{2}\langle \mathbf{u}^{n+1} + \mathbf{u}^n, -\rho\nabla_h \phi + \frac{\mu}{2}\Delta_h(\mathbf{u}^{n+1} + \mathbf{u}^n) + \frac{\mu\Delta t}{2}\nabla_h\Delta_h\phi + \mathcal{S}_h^n[(\sigma^{n+1} + \sigma^n)/2]\rangle_{\Omega_h}. \end{aligned}$$

The successive difference of potential energy can be written as

$$\begin{aligned} P^{n+1} - P^n &= \frac{\sigma_0}{2}\left(\langle |\mathbf{X}_s|^{n+1} - r_0, |\mathbf{X}_s|^{n+1} - r_0\rangle_{\Gamma_h} - \langle |\mathbf{X}_s|^n - r_0, |\mathbf{X}_s|^n - r_0\rangle_{\Gamma_h}\right) \\ &= \frac{\sigma_0}{2}\langle |\mathbf{X}_s|^{n+1} + |\mathbf{X}_s|^n - 2r_0, |\mathbf{X}_s|^{n+1} - |\mathbf{X}_s|^n\rangle_{\Gamma_h} \\ &= \frac{\Delta t}{4}\langle \sigma_0(|\mathbf{X}_s|^{n+1} - r_0 + |\mathbf{X}_s|^n - r_0), \mathcal{T}_h^n[\mathbf{u}^{n+1} + \widehat{\mathbf{u}}^n]\rangle_{\Gamma_h} \text{ (using Eq. (5.19))} \\ &= \frac{\Delta t}{4}\langle \sigma^{n+1} + \sigma^n, \mathcal{T}_h^n[\mathbf{u}^{n+1} + \widehat{\mathbf{u}}^n]\rangle_{\Gamma_h}. \end{aligned}$$

Thus, the successive difference of total energy is

$$\begin{aligned} E^{n+1} - E^n = \frac{-\rho\Delta t}{2}\langle \mathbf{u}^{n+1} + \mathbf{u}^n, \nabla_h\phi\rangle_{\Omega_h} + \frac{\mu\Delta t}{4}\langle \mathbf{u}^{n+1} + \mathbf{u}^n, \Delta_h(\mathbf{u}^{n+1} + \mathbf{u}^n)\rangle_{\Omega_h} \\ + \frac{\mu\Delta t^2}{4}\langle \mathbf{u}^{n+1} + \mathbf{u}^n, \nabla_h\Delta_h\phi\rangle_{\Omega_h} + \frac{\Delta t}{4}\langle \mathbf{u}^{n+1} + \mathbf{u}^n, \mathcal{S}_h^n[\sigma^{n+1} + \sigma^n]\rangle_{\Omega_h} \\ + \frac{\Delta t}{4}\langle \sigma^{n+1} + \sigma^n, \mathcal{T}_h^n[\mathbf{u}^{n+1} + \widehat{\mathbf{u}}^n]\rangle_{\Gamma_h} = -\frac{\mu\Delta t}{4}\|\nabla_h(\mathbf{u}^{n+1} + \mathbf{u}^n)\|^2_{\Omega_h}, \end{aligned}$$

The first and third terms after the first equality vanish due to the orthogonality of discrete divergence-free velocity and the gradient, and the commutativity of ∇_h and

Δ_h. The fourth and last terms are cancelled out due to both discrete operators are skew-adjoint. The third term is always negative by the negative definiteness of discrete Laplace operator. Thus, the total energy is decreasing which shows the scheme to be unconditionally energy stable.

5.4 Modified projection method

To avoid solving a coupled linear system of equations in Eqs. (5.8)-(5.12) or Eqs. (5.15)-(5.19), we can modify the above BE and CN schemes slightly in the following. The key idea is to march the time integration of the stretching factor $|\mathbf{X}_s|^{n+1}$ by the intermediate velocity $\mathbf{u}^*$ rather than $\mathbf{u}^{n+1}$, and substitute it into the discrete spreading operator $\mathcal{S}_h^n$ so that $|\mathbf{X}_s|^{n+1}$ is no long part of unknowns.

- Modified BE scheme

$$\rho\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} = \mu\Delta_h\mathbf{u}^* + \mathcal{S}_h^n[\sigma^n] + \sigma_0\Delta t\mathcal{S}_h^n\mathcal{T}_h^n[\mathbf{u}^*] \quad \text{in } \Omega_h, \tag{5.24}$$

$$\Delta_h\phi = \frac{1}{\Delta t}\nabla_h\cdot\mathbf{u}^* \quad \text{in } \Omega_h, \quad \left.\frac{\partial\phi}{\partial\mathbf{n}}\right|_{\partial\Omega_h} = 0, \tag{5.25}$$

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \Delta t\nabla_h\phi \quad \text{in } \Omega_h, \tag{5.26}$$

$$\nabla_h p^{n+1} = \rho\nabla_h\phi - \mu\nabla_h(\nabla_h\cdot\mathbf{u}^*) - \sigma_0\Delta t^2\mathcal{S}_h^n\mathcal{T}_h^n[\nabla_h\phi] \quad \text{in } \Omega_h, \tag{5.27}$$

$$\mathbf{U}_k^{n+1} = \sum_{\mathbf{x}}\mathbf{u}(\mathbf{x})^{n+1}\delta_h(\mathbf{x}-\mathbf{X}_k^n)\,h^2 \quad \text{on } \Gamma_h, \tag{5.28}$$

$$\mathbf{X}_k^{n+1} = \mathbf{X}_k^n + \Delta t\,\mathbf{U}_k^{n+1} \quad \text{on } \Gamma_h. \tag{5.29}$$

- Modified CN scheme

$$\begin{aligned}\rho\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} &= \frac{\mu}{2}\Delta_h(\mathbf{u}^* + \mathbf{u}^n) + \mathcal{S}_h^n[\sigma^n] \\ &\quad + \frac{\sigma_0\Delta t}{4}\mathcal{S}_h^n\mathcal{T}_h^n[\widehat{\mathbf{u}}^n] + \frac{\sigma_0\Delta t}{4}\mathcal{S}_h^n\mathcal{T}_h^n[\mathbf{u}^*] \quad \text{in } \Omega_h,\end{aligned} \tag{5.30}$$

$$\Delta_h\phi = \frac{1}{\Delta t}\nabla_h\cdot\mathbf{u}^* \quad \text{in } \Omega_h \quad \left.\frac{\partial\phi}{\partial\mathbf{n}}\right|_{\partial\Omega_h} = 0, \tag{5.31}$$

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \Delta t\nabla_h\phi \quad \text{in } \Omega_h \tag{5.32}$$

$$\nabla_h p^{n+1/2} = \rho\nabla_h\phi - \frac{\mu}{2}\nabla_h(\nabla_h\cdot\mathbf{u}^*) - \frac{\sigma_0\Delta t^2}{4}\mathcal{S}_h^n\mathcal{T}_h^n[\nabla_h\phi] \quad \text{in } \Omega_h, \tag{5.33}$$

$$\mathbf{U}_k^{n+1} = \sum_{\mathbf{x}}\mathbf{u}(\mathbf{x})^{n+1}\delta_h(\mathbf{x}-\mathbf{X}_k^n)\,h^2, \quad \widehat{\mathbf{U}}_k^n = \sum_{\mathbf{x}}\mathbf{u}(\mathbf{x})^{n}\delta_h(\mathbf{x}-\mathbf{X}_k^n)\,h^2, \quad \text{on } \Gamma_h, \tag{5.34}$$

$$\mathbf{X}_k^{n+1} = \mathbf{X}_k^n + \Delta t(\mathbf{U}_k^{n+1} + \widehat{\mathbf{U}}_k^n)/2 \quad \text{on } \Gamma_h. \tag{5.35}$$

Now, let us discuss the computational cost of the above numerical schemes. As mentioned before, the discrete operators $\mathcal{S}_h^n$ and $\mathcal{T}_h^n$ are skew-adjoint in which the matrix forms can be related by $\mathcal{T}_h^n = -\frac{h^2}{\Delta s}(\mathcal{S}_h^n)^T$. Thus, the resultant matrix equations for $\mathbf{u}^*$ in Eq. (5.24) and Eq. (5.30) are both symmetric and positive definite. Thus, it can be solved efficiently by the aggregation-based multigrid (AGMG) method recently developed by Y. Notay [47]. In Eq. (5.25) and (5.31), Poisson equation for pressure increment ϕ is solved by public software package FISHPACK [1]. One should notice that, despite the fact that the modified BE and CN schemes are not exactly unconditionally energy stable as the original schemes, the time step size Δt can be tremendously alleviated in practical computation as shown in our numerical results.

5.5 Applications to simulating vesicle dynamics

In [28], Kim and Lai developed a penalty IB method to study the dynamics of inextensible vesicles by introducing a dual representation (denoted by $\mathbf{X}(s,t)$ and $\mathbf{Y}(s,t)$) of the immersed boundary in which one of its representatives $\mathbf{X}(s,t)$ interacts with the fluid directly as in traditional IB computation, and another one $\mathbf{Y}(s,t)$ follows the equations of vesicle dynamics including the inextensibility constraint without a direct interaction with the fluid dynamics. Both IB representatives are linked by stiff springs. The advantage of this penalty idea is to decouple the solutions of fluid and vesicle dynamics at each time step so that the traditional IB implementation can be simply applied without much extra effort. However, the tension must be solved in this approach and the penalty numbers must be chosen sufficiently large to keep those two IB representations close enough. In practice, one should choose the time step size sufficiently small because of the usage of explicit forcing computation. Recently, we simplify the penalty approach by introducing a spring-like tension to keep the vesicle boundary nearly inextensible. As a result, the tension (Lagrange's multiplier for inextensibility) is no longer to be solved as part of solutions. We apply this simplified penalty approach to simulate the dynamics of three-dimensional axisymmetric vesicles in Navier-Stokes flows [23]. However, one need to choose the time step size sufficiently small because the IB force is computed at the beginning of each time step. Here, unlike those above mentioned explicit schemes, we use the developed semi-implicit schemes to simulate the vesicle dynamics in Navier-Stokes flows.

Vesicles are closed lipid membranes suspended in a viscous fluid. The membrane force consists of the elastic and bending parts as

$$\mathbf{F}(s,t) = \frac{\partial}{\partial s}(\sigma \boldsymbol{\tau}) - c_b \frac{\partial^4 \mathbf{X}}{\partial s^4}, \tag{5.36}$$

where c_b is the bending rigidity. Here, the tension σ is introduced as an unknown function to act as the Lagrange's multiplier for enforcing the local inextensibility of the membrane as

$$\nabla_s \cdot \mathbf{U} = \frac{\partial \mathbf{U}}{\partial \boldsymbol{\tau}} \cdot \boldsymbol{\tau} = 0, \quad \text{on } \Gamma. \tag{5.37}$$

As in [5, 23], we replace the unknown tension σ by

$$\sigma = \sigma_0 \left(|\mathbf{X}_s| - |\mathbf{X}_s|^0 \right), \tag{5.38}$$

where the elastic coefficient $\sigma_0 \gg 1$ and $|\mathbf{X}_s|^0$ is the local stretching factor of initial vesicle configuration. This spring-like tension intends to keep the stretching factor $|\mathbf{X}_s|$ close to its initial counterpart $|\mathbf{X}_s|^0$. (Note that, the exact inextensibility means $|\mathbf{X}_s| = |\mathbf{X}_s|^0$ for all time.) The numerical schemes presented in Section 3 can be straightforwardly extended to handle the tension computation by replacing the constant rest length r_0 with the initial stretching factor $|\mathbf{X}_s|^0$. We leave the computational detail for the bending force in the Appendix C.

Throughout this section, we set the fluid density $\rho = 1$, viscosity $\mu = 1$, and the computational domain $\Omega = [-1, 1] \times [-1, 1]$; unless otherwise stated. The stopping tolerance of the iterative method AGMG is 10^{-6}.

5.6 Numerical results

5.6.1 Convergence study

We first conduct a convergency study for the present modified BE and CN schemes. We put a vesicle with an elliptical shape $\mathbf{X}(s, 0) = (0.2\cos(s), 0.5\sin(s))$ in quiescent flow initially. The elastic coefficient is chosen as $\sigma_0 = 10^5$ and the bending coefficient $c_b = 0.01$. We use different mesh sizes as $m = n = 64, 128, 256, 512$ with the corresponding mesh $h = 2/m$. We also set the Lagrangian mesh width as $\Delta s \approx h/2$ and the time step $\Delta t = h/4$. Since the fluid is incompressible and the vesicle boundary is inextensible, the enclosed area and the total perimeter of the vesicle should be conserved theoretically as time evolves. For the fluid variables, we choose the result obtained from the finest mesh $m = 512$ as our reference solution and compute the maximal error between the reference solution and the numerical solution. All numerical solutions are computed up to $T = 0.125$.

Table 5.1 shows the relative errors of the vesicle area and the perimeter, the maximum errors of the vesicle boundary configuration, and the fluid velocity field. Here, we list the results obtained by modified BE and CN schemes. Since the fluid variables are defined at the staggered grid, when we refine the mesh, the numerical solution on the refined mesh do not coincide with the one obtained from the coarser mesh. In these runs, we simply use linear interpolation to compute the solutions at the same grid locations. One can see from Table 5.1 that the present numerical results show roughly first-order convergence for all solution variables.

BE	$m=n=64$	$m=n=128$	rate	$m=n=256$	rate
$\lvert A_h - A_0\rvert/A_0$	1.470e-3	1.252e-3	0.23	7.162e-4	0.81
$\lvert L_h - L_0\rvert/L_0$	7.444e-3	2.684e-3	1.47	1.154e-4	1.22
$\lVert \mathbf{X}_h - \mathbf{X}_{\text{ref}}\rVert_\infty$	4.209e-3	8.477e-4	2.31	4.762e-4	0.83
$\lVert u_h - u_{\text{ref}}\rVert_\infty$	5.453e-2	8.110e-3	2.75	1.578e-3	2.36
$\lVert v_h - v_{\text{ref}}\rVert_\infty$	5.628e-2	1.098e-2	2.36	3.203e-3	1.78
CN	$m=n=64$	$m=n=128$	rate	$m=n=256$	rate
$\lvert A_h - A_0\rvert/A_0$	2.688e-3	2.082e-3	0.37	1.148e-4	0.86
$\lvert L_h - L_0\rvert/L_0$	6.899e-3	2.447e-3	1.50	1.052e-4	1.22
$\lVert \mathbf{X}_h - \mathbf{X}_{\text{ref}}\rVert_\infty$	3.296e-3	9.643e-4	1.77	6.821e-4	0.50
$\lVert u_h - u_{\text{ref}}\rVert_\infty$	7.318e-2	4.919e-2	0.57	2.744e-2	0.84
$\lVert v_h - v_{\text{ref}}\rVert_\infty$	7.561e-2	5.790e-2	0.39	1.935e-2	1.58

Table 5.1: The mesh refinement results for the vesicle area A_h, the vesicle perimeter L_h, the boundary configuration $\mathbf{X}_h$, and the velocity u_h and v_h.

5.6.2 Maximal time step comparison

Now, let us investigate the numerical stability of the present schemes by testing the maximal time steps for the explicit scheme (EP), the modified BE, and CN scheme. Here, we use three different elastic coefficients $\sigma_0 = 10^7, 10^8, 10^9$ to study the numerical stability by comparing the maximal time step that can be used in each scheme. To determine the maximal time step, we make sure that the vesicle boundary behaviors are reasonable in which the relative errors of area and perimeter are both within 1% and no numerical instability occurs. The numerical parameters are chosen as same as in previous convergence study except the initial vesicle configuration is chosen as $\mathbf{X}(s) = (0.1\sin(s), 0.5\cos(s))$. Table 5.2 shows the maximal time steps for three different schemes. One can see that the time step for both modified BE and CN schemes can be chosen as 3-4 order larger than the time step used in the explicit scheme. For explicit scheme, as the grid becomes finer or the elastic coefficient σ_0 becomes larger, the time step should become smaller accordingly in order to maintain numerical stability. However, for both BE or CN schemes, we can always set $\Delta t = O(h)$ to maintain the desired numerical stability.

Table 5.3 shows the average CPU time (in seconds) for each time step and total CPU time for the computation up to $T = 1$. One can see that, the present modified BE and CN schemes outperform the explicit scheme in terms of total CPU time.

m, n	$\sigma_0 = 10^7$			$\sigma_0 = 10^8$			$\sigma_0 = 10^9$		
	EP	BE	CN	EP	BE	CN	EP	BE	CN
128	2.22e-5	1.56e-2	1.56e-2	4.36e-6	1.56e-2	1.56e-2	1.02e-6	1.56e-2	1.56e-2
256	1.03e-5	7.81e-3	7.81e-3	1.74e-6	7.81e-3	7.81e-3	3.81e-7	7.81e-3	7.81e-3
512	4.74e-6	3.91e-3	3.91e-3	7.44e-7	3.91e-3	3.91e-3	1.53e-7	3.91e-3	3.91e-3

Table 5.2: Maximum time steps for the explicit(EP), modified BE and CN schemes with different elastic coefficients σ_0.

m, n	EP			BE			CN		
	Δt	Average	Total	Δt	Average	Total	Δt	Average	Total
128	1.02e-6	0.07	68627	1.56e-2	0.13	8	1.56e-2	0.13	8
256	3.81e-7	0.16	419947*	7.81e-3	0.39	50	7.81e-3	0.43	55
512	1.53e-7	0.45	2941176*	3.91e-3	1.25	320	3.91e-3	1.45	371

Table 5.3: The average CPU time (in seconds) of each time step for different schemes. The total time with ”*” means the estimated value.

5.6.3 Tank-treading motion under shear flow

To be more physically realistic, we now consider the transient motion of a vesicle with the initial configuration $\mathbf{X}(s) = (0.1\sin(s), 0.5\cos(s))$ under shear flow $\mathbf{u} = (\gamma y, 0)$. The elastic coefficient is chosen as $\sigma_0 = 10^5$ and the bending coefficient $c_b = 0.01$. It is well-known that the equilibrium dynamics of vesicle under a simple shear flow undergoes tanking-treading motion if the viscosity contrast is under a certain threshold [24]. Here, by tank-treading motion we mean that the configuration of the vesicle remains stationary while there is a tangential motion along the vesicle boundary. Figure 5.1 shows the evolutional motion of the vesicle at different times. One can see that after some time, the vesicle shape does not change; however, the Lagrangian point (marked by $*$) along the interface moves along with its tangential velocity. This tank-treading motion is in good agreement with previous studies [71, 66, 28, 35].

As discussed in literature [71, 26, 25, 66, 28], the motion of a steady vesicle under shear flow can be characterized by the inclination angle θ between the long axis of vesicle and the flow direction, and the tank-treading frequency $f = 2\pi / \int_\Gamma \frac{dl}{u_\tau}$, where u_τ is the tangential velocity component. The inclination angle has been found to be strongly dependent on the reduced area ν. As the reduced area increases, the inclination angle will increase too. However, the angle is independent of the shear rate γ. This behavior has been verified in the left panel of Figure 5.2 which shows the steady inclination angle (θ/π) versus the reduced area (ν) with different shear rates $\gamma = 1, 5, 10$. Our numerical results are again in good agreement with those

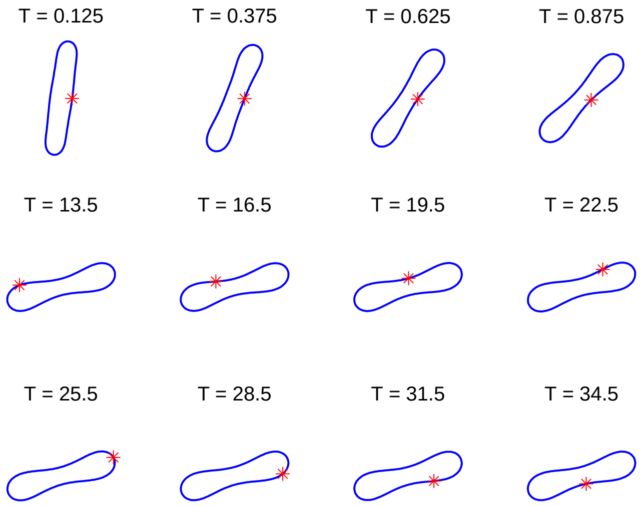


Figure 5.1: The tank-treading motion of a vesicle under shear flow.

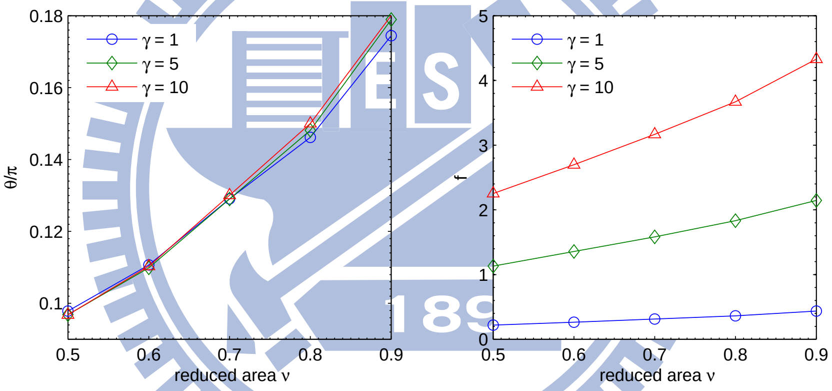


Figure 5.2: The inclination angles θ/π (left) and the tank-treading frequency $f = 2\pi / \int_\Gamma \frac{\mathrm{d}l}{\mathbf{u}\cdot\boldsymbol{\tau}}$ (right) versus reduced areas ν with different shear rates for the tank-treading motion of an inextensible vesicle in a shear flow.

obtained in previous studies [66, 28]. The right panel of Figure 5.2 shows that the tank-treading frequency f versus the reduced area ν for different shear rates. One can see that as the shear rate increases, the tangential motion becomes stronger; thus, the frequency becomes larger. Moreover, by fixing the shear rate, if the vesicle has larger reduced area then it has larger frequency as well. Again, our numerical results are in good agreement with those obtained in previous studies [66, 28].

Chapter 6

Simulating three-dimensional axisymmetric vesicles

There are two different approaches to enforce the local inextensibility constraint in literature. The first one needs to discretize the whole equations first (regardless of using boundary integral, finite element or finite difference method) and then to solve the tension and fluid variables simultaneously. Based on different time stepping schemes, this kind of approach can be explicit or semi-implicit depending on how we treat the force computations. There usually exists a trade-off between the time-step stability and efficiency in those algorithms. Some recent works using boundary integral method [66, 67, 68, 4, 72], level set method [60, 36] et. al. fall into this category, just to name a few. Recently, we have also introduced a linearly semi-implicit scheme to solve a 2D Stokes flow with an inextensible interface enclosing a solid particle [37]. The scheme solves the fluid variables, the tension and the particle surface force altogether. There are two advantages of the method as follows: (1) the scheme preserves some operators symmetrically so the resultant linear system is symmetric; (2) the local inextensibility error can be estimated analytically. Although one can increase the time step significantly, more improvements have to be done on the iterative methods for solving such resultant linear system, especially finding good pre-conditioners.

Another approach is called penalty method. In [28], the authors introduced a dual Lagrangian immersed boundaries to represent the vesicle boundary. One vesicle boundary interacts with the fluid dynamics directly and the other vesicle boundary follows the equations of the vesicle dynamics, including the inextensibility constraint, without a direct interaction with the fluid dynamics. The two boundaries are connected by the penalty forces which act on both boundaries by Newton's third law. The tension must still be solved in above approach.

However, one can even simplify the above penalty approach by introducing a spring-like force to replace the tension and to keep the surface area from dilating. Thus, the tension or Lagrange's multiplier for inextensibility is no need to be solved as part of the solution. Although the inextensibility constraint is not exactly satisfied, we can get physically reasonable results as long as the penalty number (stiffness number) is chosen large enough. This approach has been adopted successfully by incorporating the level set method to study 2D or 3D [39, 41, 11] vesicle dynamics in finite element framework. A partially implicit sharp interface method in [5] also used the above approach to test a 2D case with an inextensible interface. In this chapter, we use the above similar idea to study 3D axisymmetric vesicle dynamics in immersed boundary framework. One should notice that all the calculations about the spatial derivatives, geometrical quantities (mean and Gaussian curvatures), the bending and elastic forces, and the Navier-Stokes solver have non-trivial differences comparing with our previous 2D works.

The rest of the chapter are organized as follows. In Section 2, we will describe the fluid-vesicle governing equations in axisymmetric form based on immersed boundary formulation. Instead of introducing a Lagrange's multiplier to enforce the vesicle inextensibility constraint, we modify the model by adopting a spring-like tension to make the vesicle boundary nearly inextensible so that solving the unknown tension can be avoided. Since higher-order spatial derivatives are needed in computations of interfacial geometrical quantities, we use Fourier spectral approximation to represent the interface. A detailed numerical algorithm is described in Section 6.2. A series of numerical tests to demonstrate the accuracy and reliability of the present scheme are presented in Section 6.3.

6.1 Governing equations

We consider a single 3D axisymmetric inextensible vesicle $\Gamma(t)$ suspended in a viscous incompressible Navier-Stokes fluid domain Ω. Here, the vesicle is time-dependent and is symmetry in the θ direction so that the vesicle surface has the parametric form as

$$\mathbf{X}(s,\theta,t) = (R(s,t)\cos\theta, R(s,t)\sin\theta, Z(s,t)), \tag{6.1}$$

where the parameters (s,θ) are in $[0,\pi]\times[0,2\pi]$. One should also note that the vesicle membrane force which is defined later has the same symmetric form as in Eq. (6.1). Under such symmetry, one can regard the vesicle boundary as the surface of revolution around the z-axis with the radius $R(s,t)$ so that it can be simplified into a two-dimensional immersed boundary $\mathbf{X}(s,t) = (R(s,t), Z(s,t)), s \in [0,\pi]$ in the 2D meridian $r - z$ domain.

Under the same assumption of axisymmetry, the 3D Navier-Stokes equations can be simply written in a 2D manner using the axisymmetric cylindrical coordinates $\mathbf{x} = (r,z)$. In the following, we denote $\mathbf{u} = (u,w)$ as the velocity field defined on a 2D meridian domain $\Omega = \{(r,z)|0 < r \leq a,\ c \leq z \leq d\}$, where u and w are the radial (r-coordinate) and axial (z-coordinate) velocity components, respectively. We also denote $\mathbf{U} = (U,W)$ as the corresponding velocity components on the vesicle boundary. The mathematical equations of motion consist of a viscous incompressible fluid in a domain Ω containing an immersed, inextensible, massless vesicle boundary (or interface) Γ which can be written in an immersed boundary formulation as

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t}+(\mathbf{u}\cdot\nabla)\mathbf{u}\right)=-\nabla p+\mu\widetilde{\Delta}\mathbf{u}+\int_0^{\pi}\mathbf{F}(s,t)\delta(\mathbf{x}-\mathbf{X}(s,t))\ \mathrm{d}s\quad \text{in } \Omega, \tag{6.2}$$

$$\nabla\cdot\mathbf{u}=0\quad \text{in } \Omega, \tag{6.3}$$

$$\nabla_s\cdot\mathbf{U}=0\quad \text{on } \Gamma, \tag{6.4}$$

$$\frac{\partial \mathbf{X}}{\partial t}(s,t)=\mathbf{U}(s,t)=\int_{\Omega}\mathbf{u}(\mathbf{x},t)\delta(\mathbf{x}-\mathbf{X}(s,t))\ \mathrm{d}\mathbf{x}\quad \text{on } \Gamma. \tag{6.5}$$

Here, we assume the fluid has the same viscosity and density inside and outside of the vesicle boundary. For axisymmetry, the gradient and divergence operators are defined as

$$\nabla=\left(\frac{\partial}{\partial r},\frac{\partial}{\partial z}\right),\qquad \nabla\cdot=\left(\frac{1}{r}\frac{\partial}{\partial r}r,\frac{\partial}{\partial z}\right)\cdot, \tag{6.6}$$

thus, the Laplace operator is

$$\Delta=\nabla\cdot\nabla=\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial}{\partial r}\right)+\frac{\partial^2}{\partial z^2}, \tag{6.7}$$

and $\widetilde{\Delta}\mathbf{u}=(\Delta u-\frac{u}{r^2},\Delta w)$.

Eqs. (6.2)-(6.3) are the familiar incompressible Navier-Stokes equations with a singular force term $\mathbf{F}$ arising from the vesicle membrane force. Eq. (6.4) represents the inextensibility constraint of the vesicle surface which is equivalent to the zero surface divergence of the velocity along the interface, see the explanation later. Eq. (6.5) simply explains that the interface moves along with the local fluid velocity (the interfacial velocity). Here, the interfacial velocity $\mathbf{U}$ is the interpolation of the fluid velocity at the interface defined as in traditional IB formulation. The interaction between the fluid and the interface is linked by the two-dimensional Dirac delta function $\delta(\mathbf{x})=\delta(r)\delta(z)$.

6.1.1 Vesicle boundary forces in axisymmetric coordinate

Vesicle membrane is known to be inextensible and exhibits a resistance against bending. Thus, the membrane energy can be modelled by two parts; namely, the Helfrich type energy E_b [18] to resist the bending of the membrane, and the tension energy E_σ to enforce the inextensibility constraint [71]. So the total energy is

$$E(t)=E_b(t)+E_\sigma(t)=\int_{\Gamma}\left(\frac{c_b}{2}H^2+\sigma\right)\ \mathrm{d}S, \tag{6.8}$$

where c_b is the bending rigidity, H is the surface mean curvature, and σ is the tension which acts as a Lagrange multiplier to enforce the inextensibility constraint Eq. (6.4). By taking the variational derivative from the above surface energy, one can derive the vesicle boundary force $\mathbf{F}_\Gamma = \mathbf{F}_b + \mathbf{F}_\sigma$ consisting of the bending force $\mathbf{F}_b$ and the elastic force $\mathbf{F}_\sigma$ as

$$\mathbf{F}_b = c_b \left(\Delta_s H + 2H(H^2 - K)\right) \mathbf{n}, \qquad \mathbf{F}_\sigma = \nabla_s \sigma - 2H\sigma\mathbf{n}, \tag{6.9}$$

where K is the membrane Gaussian curvature, $\mathbf{n}$ is the unit outward normal vector, ∇_s is the surface gradient operator, and Δ_s is the surface Laplacian operator. The detailed derivation of Eq. (6.9) can be found in [69].

Under axisymmetric assumption, one can further write down H, K, and $\Delta_s H$ in terms of R and Z explicitly (see the Appendix in detail) as

$$H = \frac{1}{2}\left(\frac{Z_s}{R\,|\mathbf{X}_s|} + \frac{R_s Z_{ss} - R_{ss} Z_s}{|\mathbf{X}_s|^3}\right), \quad K = \frac{Z_s(R_s Z_{ss} - R_{ss} Z_s)}{R\,|\mathbf{X}_s|^4}, \tag{6.10}$$

$$\Delta_s H = \frac{1}{R\,|\mathbf{X}_s|}\frac{\partial}{\partial s}\left(\frac{R}{|\mathbf{X}_s|}\frac{\partial H}{\partial s}\right). \tag{6.11}$$

The unit tangent $\boldsymbol{\tau}$ and the unit outward normal vector $\mathbf{n}$ can be hereby defined as

$$\boldsymbol{\tau}(s,t) = \frac{\mathbf{X}_s}{|\mathbf{X}_s|} = \frac{(R_s, Z_s)}{\sqrt{R_s^2 + Z_s^2}}, \qquad \mathbf{n}(s,t) = \frac{(Z_s, -R_s)}{\sqrt{R_s^2 + Z_s^2}}. \tag{6.12}$$

Thus, by substituting the above surface geometrical quantities into Eq. (6.9), an explicit form for the bending force $\mathbf{F}_b$ can be derived easily. Similarly, the tension force $\mathbf{F}_\sigma$ can also be explicitly written as

$$\begin{aligned}
\mathbf{F}_\sigma &= \nabla_s \sigma - 2H\sigma\mathbf{n} \\
&= \left(\frac{1}{|\mathbf{X}_s|}\frac{\partial \sigma}{\partial s}\right)\boldsymbol{\tau} - \left(\frac{Z_s}{R\,|\mathbf{X}_s|} + \frac{R_s Z_{ss} - R_{ss} Z_s}{|\mathbf{X}_s|^3}\right)\sigma\mathbf{n} \\
&= \left(\frac{R_{ss} Z_s - R_s Z_{ss}}{|\mathbf{X}_s|^3}\sigma\mathbf{n} + \frac{1}{|\mathbf{X}_s|}\frac{\partial \sigma}{\partial s}\boldsymbol{\tau}\right) - \frac{Z_s}{R\,|\mathbf{X}_s|}\sigma\mathbf{n} \\
&= \frac{1}{|\mathbf{X}_s|}\left(\sigma\frac{\partial \boldsymbol{\tau}}{\partial s} + \frac{\partial \sigma}{\partial s}\boldsymbol{\tau}\right) - \frac{Z_s}{R\,|\mathbf{X}_s|}\sigma\mathbf{n} = \frac{1}{|\mathbf{X}_s|}\frac{\partial}{\partial s}(\sigma\boldsymbol{\tau}) - \frac{Z_s}{R\,|\mathbf{X}_s|}\sigma\mathbf{n}.
\end{aligned} \tag{6.13}$$

The delta function in 3D Cartesian coordinates and axisymmetric cylindrical coordinates can be related as $\delta^3 = \delta^2/r$ where δ^2 is the two-dimensional delta function in $r-z$ plane. Therefore, the singular force term can be written as

$$\int_\Gamma \mathbf{F}_\Gamma \delta^3(\mathbf{x} - \mathbf{X})\,\mathrm{d}S = \int_0^\pi \mathbf{F}_\Gamma \frac{\delta^2(\mathbf{x} - \mathbf{X})}{R} R\,|\mathbf{X}_s|\,\,\mathrm{d}s = \int_0^\pi \mathbf{F}_\Gamma\,|\mathbf{X}_s|\,\delta^2(\mathbf{x} - \mathbf{X})\,\mathrm{d}s.$$

We thus obtain the immersed boundary force $\mathbf{F}(s,t) = \mathbf{F}_\Gamma\,|\mathbf{X}_s|$.

6.1.2 Nearly inextensible approach

As mentioned before, to avoid solving the extra unknown tension $\sigma(s,t)$, we adopt an idea that is simpler than our previous work [28] to study 2D vesicle dynamics. Let us start with the derivation of inextensibility constraint (6.4) in axisymmetric coordinates. The rate of change of the surface dilating factor $R\left|\mathbf{X}_s\right|$ can be derived as follows [34].

$$\begin{aligned}
\frac{\partial}{\partial t}\left(R\left|\mathbf{X}_s\right|\right) &= \frac{\partial R}{\partial t}\left|\mathbf{X}_s\right| + R\frac{\partial\left|\mathbf{X}_s\right|}{\partial t} = U\left|\mathbf{X}_s\right| + R\frac{\mathbf{X}_s\cdot\mathbf{X}_{st}}{\left|\mathbf{X}_s\right|} \\
&= U\left|\mathbf{X}_s\right| + R\frac{\mathbf{X}_s\cdot\mathbf{U}_s}{\left|\mathbf{X}_s\right|} = U\left|\mathbf{X}_s\right| + R\frac{\partial\mathbf{U}}{\partial s}\cdot\boldsymbol{\tau} \\
&= \left(\frac{U}{R} + \frac{\partial\mathbf{U}}{\partial\boldsymbol{\tau}}\cdot\boldsymbol{\tau}\right)R\left|\mathbf{X}_s\right| = (\nabla_s\cdot\mathbf{U})R\left|\mathbf{X}_s\right|.
\end{aligned} \tag{6.14}$$

The above derivation shows the expression of the surface divergence $\nabla_s\cdot\mathbf{U}$ explicitly, and also shows that if the surface is locally inextensible (the rate of change of surface dilating factor equals to zero), then we obtain $\nabla_s\cdot\mathbf{U} = 0$ which is exactly the constraint shown in Eq. (6.4). In order to relax the constraint a bit, one can replace the unknown tension by a spring-like elastic force as

$$\sigma = \sigma_0(R\left|\mathbf{X}_s\right| - R^0\left|\mathbf{X}_s\right|^0) \tag{6.15}$$

with a large stiffness constant σ_0, where $R^0\left|\mathbf{X}_s\right|^0$ is the initial surface dilating factor. Therefore, the interface now is nearly inextensible. Consequently, the elastic energy E_σ is modified by

$$E_\sigma(t) = \int_0^\pi \frac{\sigma_0}{2}\left(R\left|\mathbf{X}_s\right| - R^0\left|\mathbf{X}_s\right|^0\right)^2\,\mathrm{d}s, \tag{6.16}$$

and the total membrane energy becomes

$$E(t) = \int_0^\pi \left(\frac{\sigma_0}{2}\left(R\left|\mathbf{X}_s\right| - R^0\left|\mathbf{X}_s\right|^0\right)^2 + \frac{c_b}{2}H^2R\left|\mathbf{X}_s\right|\right)\,\mathrm{d}s. \tag{6.17}$$

6.1.3 Elastic boundary force

Since the elastic energy is modified, the resultant tension force $\mathbf{F}_\sigma$ must be modified as well. This can be done by taking the variational derivative of new energy E_σ as follows. Let $\widetilde{\mathbf{X}}(s) = (\widetilde{R}(s),\widetilde{Z}(s))$ be a perturbation of the interface $\mathbf{X}$, and ε be a small number. Then the perturbed energy becomes

$$E_\sigma(\mathbf{X}+\varepsilon\widetilde{\mathbf{X}}) = \int_0^\pi \frac{\sigma_0}{2}\left[(R+\varepsilon\widetilde{R})\left|\mathbf{X}_s+\varepsilon\widetilde{\mathbf{X}}_s\right| - R^0\left|\mathbf{X}_s\right|^0\right]^2\,\mathrm{d}s.$$

Taking the derivative of above energy with respect to ε, we obtain

$$\begin{aligned}
&\frac{\mathrm{d}E_\sigma}{\mathrm{d}\varepsilon}\left(\mathbf{X}+\varepsilon\widetilde{\mathbf{X}}\right)\\
&=\int_0^\pi \sigma_0\left[(R+\varepsilon\widetilde{R})|\mathbf{X}_s+\varepsilon\widetilde{\mathbf{X}}_s| - R^0\,|\mathbf{X}_s|^0\right]\left[\widetilde{R}|\mathbf{X}_s+\varepsilon\widetilde{\mathbf{X}}_s| + \left(R+\varepsilon\widetilde{R}\right)\right]\frac{\mathbf{X}_s+\varepsilon\widetilde{\mathbf{X}}_s}{|\mathbf{X}_s+\varepsilon\widetilde{\mathbf{X}}_s|}\cdot\widetilde{\mathbf{X}}_s\ \mathrm{d}s.
\end{aligned}$$

Then we evaluate the above equation at $\varepsilon = 0$ to obtain

$$\begin{aligned}
&\left.\frac{\mathrm{d}E_\sigma}{\mathrm{d}\varepsilon}\left(\mathbf{X}+\varepsilon\widetilde{\mathbf{X}}\right)\right|_{\varepsilon=0}\\
&=\int_0^\pi \sigma_0\left(R\,|\mathbf{X}_s| - R^0\,|\mathbf{X}_s|^0\right)\left(\widetilde{R}\,|\mathbf{X}_s| + R\boldsymbol{\tau}\cdot\widetilde{\mathbf{X}}_s\right)\ \mathrm{d}s\\
&=\int_0^\pi \sigma\widetilde{R}\,|\mathbf{X}_s|\ \mathrm{d}s + \int_0^\pi \sigma R\boldsymbol{\tau}\cdot\widetilde{\mathbf{X}}_s\ \mathrm{d}s \quad \left(\text{since}\sigma = \sigma_0\left(R\,|\mathbf{X}_s| - R^0\,|\mathbf{X}_s|^0\right)\right)\\
&=\int_0^\pi \sigma\widetilde{R}\,|\mathbf{X}_s|\ \mathrm{d}s - \int_0^\pi \frac{\partial}{\partial s}\left(R\sigma\boldsymbol{\tau}\right)\cdot\widetilde{\mathbf{X}}\ \mathrm{d}s \quad (\text{integration by parts and } R=0 \text{ at end points})\\
&=\int\left(\frac{\sigma}{R}\mathbf{e}_1\right)\cdot\widetilde{\mathbf{X}}\ \mathrm{d}S - \int\left[\frac{1}{|\mathbf{X}_s|}\frac{\partial}{\partial s}(\sigma\boldsymbol{\tau}) + \frac{R_s\sigma\boldsymbol{\tau}}{R\,|\mathbf{X}_s|}\right]\cdot\widetilde{\mathbf{X}}\ \mathrm{d}S \quad (\text{using } \mathrm{d}S = R\,|\mathbf{X}_s|\ \mathrm{d}s)\\
&=\int\left[\frac{\sigma}{R\,|\mathbf{X}_s|^2}\left(R_s^2+Z_s^2,0\right) - \frac{\sigma}{R\,|\mathbf{X}_s|^2}\left(R_s^2,R_sZ_s\right) - \frac{1}{|\mathbf{X}_s|}\frac{\partial}{\partial s}(\sigma\boldsymbol{\tau})\right]\cdot\widetilde{\mathbf{X}}\ \mathrm{d}S\\
&=\int\left[\frac{\sigma Z_s}{R\,|\mathbf{X}_s|}\left(\frac{Z_s}{|\mathbf{X}_s|},-\frac{R_s}{|\mathbf{X}_s|}\right) - \frac{1}{|\mathbf{X}_s|}\frac{\partial}{\partial s}(\sigma\boldsymbol{\tau})\right]\cdot\widetilde{\mathbf{X}}\ \mathrm{d}S\\
&=-\int\left[\frac{1}{|\mathbf{X}_s|}\frac{\partial}{\partial s}(\sigma\boldsymbol{\tau}) - \frac{Z_s}{R\,|\mathbf{X}_s|}\sigma\mathbf{n}\right]\cdot\widetilde{\mathbf{X}}\ \mathrm{d}S\\
&=-\int\mathbf{F}_\sigma\cdot\widetilde{\mathbf{X}}\ \mathrm{d}S.
\end{aligned}$$

This leads to the corresponding elastic force

$$\mathbf{F}_\sigma = \frac{1}{|\mathbf{X}_s|}\frac{\partial}{\partial s}(\sigma\boldsymbol{\tau}) - \frac{Z_s}{R\,|\mathbf{X}_s|}\sigma\mathbf{n}. \tag{6.18}$$

It is very interesting to conclude that the above form is exactly same as the one in Eq. (6.13) despite the fact that the definition of tension σ is different.

6.2 Numerical discretization

6.2.1 Fourier representation of the interface and computation of geometrical quantities

We adopt the spectral Fourier discretization to achieve higher order of accuracy as in [67]. Since the interface is axisymmetric and is defined in 2D meridian domain

($r - z$ plane with $r > 0$), one can extend the interface representation $\mathbf{X}(s) = (R(s), Z(s)), s \in [0, \pi]$ to be in another half of meridian domain ($r < 0$). More precisely, we perform the radial component $R(s)$ by the odd extension and the axial component $Z(s)$ by the even extension so that this extending interfacial representation is periodic in $[0, 2\pi]$ in the $r - z$ plane. Under such extension, one can choose an even number of grid points M such that the interface is discretized by $\mathbf{X}(s_k) = (R(s_k), Z(s_k)), s_k = k\Delta s = 2k\pi/M, k = 0, 1, \ldots M$. (Note that, the spectral representation for the interface adopted in [67] used the sine and cosine expansions for the radial and axial components, respectively.) Thus, the interface is represented by the discrete Fourier series expansion as

$$R(s) = \sum_{k=-M/2}^{M/2-1} \widehat{R_k} e^{i\,k\,s}, \quad \text{and} \quad Z(s) = \sum_{k=-M/2}^{M/2-1} \widehat{Z_k} e^{i\,k\,s}, \tag{6.19}$$

where $\widehat{R_k}$ and $\widehat{Z_k}$ are the corresponding Fourier coefficients for $R(s)$ and $Z(s)$, respectively. Those Fourier coefficients can be performed very efficiently by using the Fast Fourier Transform (FFT).

As described in previous section, in order to obtain the interfacial bending force, one need to compute the mean curvature H and its surface Laplacian $\Delta_s H$, and the Gaussian curvature K. Those geometrical quantities are explicitly expressed before as in Eqs. (6.10)-(6.11). One can see the computations of those geometrical quantities need the calculations of up to fourth-order derivatives of interface components $R(s)$ and $Z(s)$. However, these can be done quite easily by using the pseudospectral method [63] in which the Fourier coefficients of p-th derivative of $R(s)$ can be obtained as $(il)^p \widehat{R_l}$ so by taking one more inverse FFT, we can obtain the derivative in physical space. (Similar procedures should be done for the derivatives of $Z(s)$). Notice that, we should expand all the derivatives involving $R(s)$ and $Z(s)$ explicitly for those geometrical quantities in Eq. (6.10)-(6.11) and compute them to enhance the accuracy.

Remark: It is important to mention that in the above spectral computations for Eqs. (6.10)-(6.11), special care must be taken at the points of $s = 0$ and $s = \pi$ where the radial component $R = 0$. However, due to the odd extension of $R(s)$ across at $s = 0, \pi$, one can conclude that the second derivative R_{ss} and the fourth derivative R_{ssss} are both zero at those two points. Similarly, due to the even extension of the axial component $Z(s)$, the first derivative Z_s and the third derivative Z_{sss} are both zero at $s = 0, \pi$. By applying the the l'Hospital's rule and using those conditions mentioned above, one can evaluate the geometrical quantities at $s = 0, \pi$ as follows.

$$
\begin{aligned}
&\lim_{s \to 0,\pi} H(s) = \left. \frac{Z_{ss}}{|R_s| R_s} \right|_{s=0,\pi}, \quad \lim_{s \to 0,\pi} H_s(s) = 0, \\
&\lim_{s \to 0,\pi} H_{ss}(s) = \left. \frac{2R_s^2 Z_{ssss} - 6Z_{ss}(R_s R_{sss} + Z_{ss}^2) - 2R_s R_{sss} Z_{ss}}{3R_s |R_s|^3} \right|_{s=0,\pi}, \\
&\lim_{s \to 0,\pi} K(s) = \left. \frac{Z_{ss}^2}{R_s^4} \right|_{s=0,\pi}, \quad \lim_{s \to 0,\pi} \Delta_s H(s) = \left. \frac{2H_{ss}}{R_s^2} \right|_{s=0,\pi}.
\end{aligned}
$$

6.2.2 Time-stepping scheme

Let us denote the time step size by Δt and the superscript of the variables as the time step index. At the beginning of each time step n, the interface configuration $\mathbf{X}^n(s) = \mathbf{X}(s, n\Delta t) = (R^n(s), Z^n(s))$ is given so the unit tangent vector $\boldsymbol{\tau}^n(s) = \mathbf{X}_s^n / |\mathbf{X}_s|^n$ can be evaluated accordingly. Here and the following, the subscript s-index denotes the spectral derivative with respect to s as described in previous subsection. The unit outward normal $\mathbf{n}^n(s)$ can be obtained easily from $\boldsymbol{\tau}^n(s)$. The steps of numerical time integration are as follows.

1. Compute the interfacial tension and bending force.

$$
\begin{aligned}
\sigma^n &= \sigma_0 \left(R^n |\mathbf{X}_s|^n - R^0 |\mathbf{X}_s|^0 \right), \quad \mathbf{F}_\sigma^n = \sigma_s^n \boldsymbol{\tau}^n + \sigma^n \boldsymbol{\tau}_s^n - \frac{Z_s^n}{R^n} \sigma^n \mathbf{n}^n, \\
\mathbf{F}_b^n &= c_b(\Delta_s H^n + 2H^n((H^n)^2 - K^n)) |\mathbf{X}_s|^n \, \mathbf{n}^n, \\
\mathbf{F}^n &= \mathbf{F}_\sigma^n + \mathbf{F}_b^n
\end{aligned}
$$

   where the mean curvature H and its surface Laplacian $\Delta_s H$, and the Gaussian curvature K are all computed in the way described in previous subsection. One should also notice that the term Z_s^n / R^n in $\mathbf{F}_\sigma$ at the collocation points $s = 0, \pi$ must be modified to Z_{ss}^n / R_s^n due to the fact of $R = 0$ at those two points.

2. Distribute the interfacial force from the interface to the fluids.

$$
\mathbf{f}^n(\mathbf{x}) = \sum_s \mathbf{F}^n(s) \, \delta_h^2 \left(\mathbf{x} - \mathbf{X}^n(s) \right) \Delta s.
$$

   Note that, when a marker $\mathbf{X}^n$ is close to the z-axis ($r = 0$), the spreading region will cover some grid points in another half of meridian domain ($r < 0$) due to the support of discrete delta function. In that case, we should make the same supplement to the symmetric grid points in the right half domain.

3. Solve the Navier-Stokes equations. This can be done by the following semi-implicit second-order projection method, where the nonlinear term is approximated by the Adams-Bashforth scheme.

$$\rho\frac{3\mathbf{u}^* - 4\mathbf{u}^n + \mathbf{u}^{n-1}}{2\Delta t} + \big(2(\mathbf{u}^n\cdot\nabla_h)\mathbf{u}^n - (\mathbf{u}^{n-1}\cdot\nabla_h)\mathbf{u}^{n-1}\big) = -\nabla_h p^n + \mu\widetilde{\Delta}_h\mathbf{u}^* + \mathbf{f}^n,$$

$$\Delta_h\phi = \frac{3}{2\Delta t}\nabla_h\cdot\mathbf{u}^*, \quad \frac{\partial\phi}{\partial\mathbf{n}} = 0 \text{ on } \partial\Omega,$$

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \frac{2}{3}\Delta t\nabla_h\phi,$$

$$\nabla_h p^{n+1} = \rho\nabla_h\phi + \nabla_h p^n - \frac{2}{3}\Delta t\mu\widetilde{\Delta}_h(\nabla_h\phi).$$

   The spatial operators ∇_h and $\nabla_h\cdot$ are the standard second-order central difference approximations to the gradient and divergence operators in Eq. (6.6). Also, the spatial operator Δ_h is the standard second-order central difference approximation to the Laplace operator in Eq. (6.7).

   Notice that, due to the axisymmetric property, the boundary conditions for velocity and pressure at $r = 0$ are $u = 0$, $\partial w/\partial r = 0$ and $\partial p/\partial r = 0$. One can see that the above Navier-Stokes solver involves solving two Helmholtz-type equations for the velocity $\mathbf{u}^*$ and one Poisson equation for the pressure increment ϕ. Those elliptic equations can be solved efficiently by the public software FISHPACK [1].

4. Interpolate the new velocity from the fluid lattice points to the Lagrangian markers and then advance the markers to new positions as

$$\mathbf{X}^{n+1}(s) = \mathbf{X}_k^n + \Delta t\sum_{\mathbf{x}}\mathbf{u}^{n+1}\delta_h^2(\mathbf{x} - \mathbf{X}^n(s))h^2.$$

6.3 Numerical results

In this section, we perform a series of numerical tests for the present scheme developed in the previous section. We first verify the spectral accuracy for the spatial discretization of the interfaces by calculating their geometrical quantities such as mean curvature, Gaussian curvature and the surface Laplacian of mean curvature. Then we will choose different stiffness numbers to study the effect on the resulting flows. The mesh refinement study will be performed to check the accuracy of the scheme and an equilibrium biconcave shape of vesicle in quiescent flow will be demonstrated. We also study the vesicle deformations under the gravitational force and conclude that even with the same reduced volume but different initial shapes can cause very different final stationary shapes. The above results are consistent with those used either 3D axisymmetric [67] or 3D simulations [4, 68]. We conclude our numerical tests by simulating the vesicles in Poiseuille flows and study the equilibrium shapes under the effects of different reduced volume, the confinement, and the mean flow velocity.

6.3.1 Accuracy check for the interfacial geometrical quantities

In the following, we check the accuracy of the spectral scheme for the computations of the geometrical quantities of the interface under the Fourier spatial representations. Here, we test four different axisymmetric surfaces depicted in Figure 6.1 in which $(R(s), Z(s))$ are listed as follows.

- Spherical surface: $(0.5\cos s, 0.5\sin s)$
- Prolate surface: $(0.1\cos s, 0.5\sin s)$
- Oscillatory surface: $(\frac{3}{20}r(s)\cos s, \frac{3}{40}r(s)\sin s)$, where $r(s) = \sqrt{\sin^2 s + 9\cos^2 s + \cos^2 4s}$
- Peanut-like surface: $((0.0414 + 0.4006\sin^2 s - 0.2246\sin^4 s)\cos s, 0.5\sin s)$

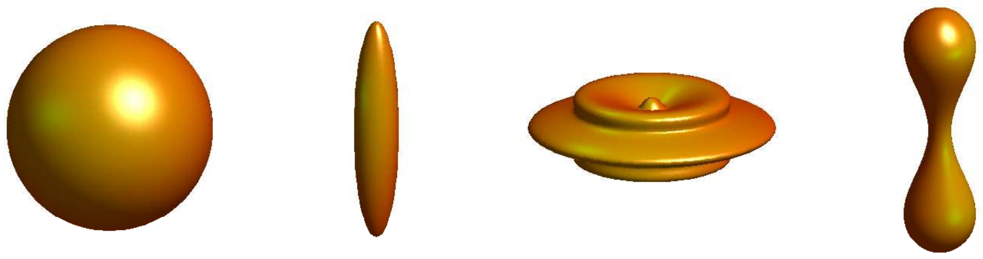


Figure 6.1: Four different surfaces: Spherical, Prolate, Oscillatory and Peanut-like surface (from left to right).

Table 6.1 show the maximum absolute errors (not relative errors) between the numerical results obtained by present spectral method and the analytical results which can be computed easily from their parametric forms. Here, we show the errors for mean curvature H, Gaussian curvature K and the surface Laplacian of mean curvature $\Delta_s H$ as well. One can see that for smooth surface such as spherical, prolate or peanut-like surface, using the grid number $M = 32$ to compute H and K has already achieved the accuracy close to the machine error. As the mesh is refined, one can see the absolute errors slightly increase due to the round-off error effect which is quite common in practical spectral computations. This effect becomes even more prominent when we compute $\Delta_s H$ since higher derivatives cause larger conditioned numbers. The spectral method used in [67] shows the same error behaviors.

For the oscillatory surface, as expected, using a little finer grid point $M = 256$ can achieve better accuracy as previously mentioned cases. Meanwhile, in this case, one can see the error decreases exponentially which shows the spectral accuracy of the method. We also show the errors obtained from the second-order finite difference scheme for the oscillatory surface case in Table 6.2. It is apparently to see that the spectral method outperforms the finite difference method for the computations of those geometrical quantities in terms of the magnitude of absolute errors. Nevertheless, the finite difference method still shows the convergence in a lower order way.

M	Spherical surface			Prolate surface		
	H	K	$\Delta_s H$	H	K	$\Delta_s H$
32	1.643E−14	6.528E−14	1.284E−11	3.339E−13	3.365E−11	1.909E−08
64	6.662E−13	2.646E−13	1.787E−10	7.176E−13	6.207E−11	8.812E−08
128	3.586E−13	1.434E−12	6.741E−09	8.647E−12	8.649E−10	3.949E−06

M	Oscillatory surface			Peanut-like surface		
	H	K	$\Delta_s H$	H	K	$\Delta_s H$
32	4.600E−02	1.440E+00	1.907E+03	7.993E−14	1.705E−12	5.575E−10
64	3.336E−04	5.181E−03	2.799E+01	2.398E−13	4.874E−12	4.561E−09
128	3.572E−09	5.304E−08	9.222E−04	1.877E−12	3.969E−11	1.823E−07
256	6.068E−12	1.511E−10	1.475E−06			

Table 6.1: The mesh refinement study for the computations of H, K and $\Delta_s H$ by the spectral method. The maximum absolute error is defined as $\|H - H_e\|_\infty$, where H_e is exact value of mean curvature. Similar notations for other quantities.

6.3.2 Study on different stiffness number σ_0

As mentioned before, the stiffness parameter σ_0 in Eq. (6.15) must be chosen large enough to keep the nearly inextensible property. In this subsection, we test three different numbers to see how the vesicle behaves on those choices. We put a vesicle with elliptical shape $\mathbf{X}_0(s) = (0.5\cos s, 0.1\sin s)$ in quiescent flow. Three different stiffness parameters $2 \times 10^3, 2 \times 10^4$ and 2×10^5 are chosen, and the time steps are chosen as $\Delta t = h/16, h/32, h/256$, respectively. We set the bending coefficient $c_b = 2 \times 10^{-2}$ and run the simulations up to time $T = 2$. As shown in Figure 6.2(a), all vesicles coincide with each others (better view from cross-section) at different times

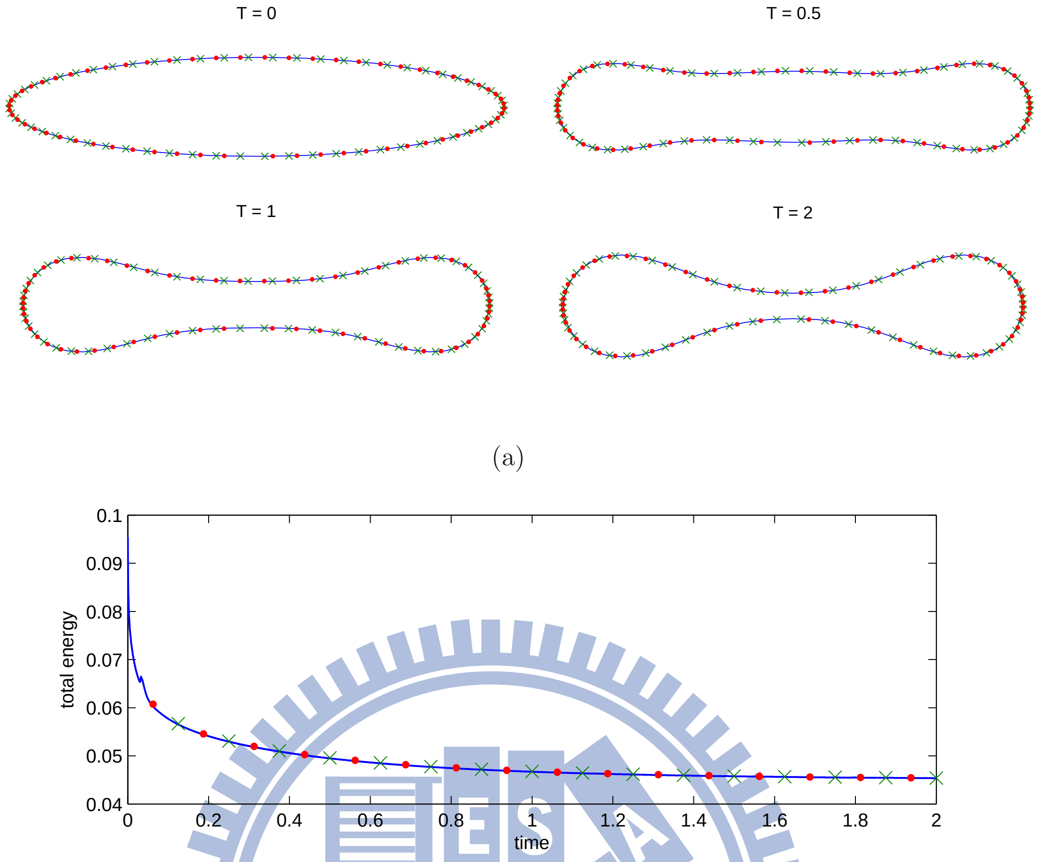
M	H	K	$\Delta_s H$
32	6.359E+00	1.703E+02	1.956E+05
64	3.061E+00	2.529E+02	1.197E+05
128	1.787E+00	4.163E+01	7.090E+04
256	5.464E−01	1.145E+01	3.499E+04

Table 6.2: The mesh refinement study for the computations of H, K and $\Delta_s H$ by the second-order finite difference method. Only the results of the oscillatory surface are listed.

and they all reach the bi-concave shape eventually. We also plot the time evolution of total energy computed by Eq. (6.17) in Figure 6.2(b), once again, the total energies are also coincided. This consistence indicates that as long as the parameter σ_0 is large enough, the flow remains the same. Table 6.3 shows the maximum errors of area dilating factor, the relative errors of the total surface area and the volume of the vesicle at $T = 2$. Notice that, the total surface area and the volume are calculated in the axisymmetric way. It is interesting to see that as the stiffness increases an order of magnitude, the maximum error of area dilation will be reduced, which results the surface area error is reduced by the same order of magnitude. This kind of error behavior is quite physically reasonable since the larger σ_0 is, the stronger spring force is provided to keep $R^n |\mathbf{X}_s|^n$ closer to the initial one. Although the volume relative error does not show the same asymptotic behavior as the surface error, both relative errors are within 0.03% when σ_0 is equal to or greater than 2×10^4. We also test the case of a vesicle under shear flow and obtain almost the same error behaviors so we omit here. Based on this test, we choose $\sigma_0 = 2 \times 10^4$ in the remainder of this chapter.

σ_0	Δt	$\Vert R \vert\mathbf{X}_s\vert - R^0 \vert\mathbf{X}_s\vert^0 \Vert_\infty$	$\vert A_h - A_0\vert/A_0$	$\vert V_h - V_0\vert/V_0$
2×10^3	$h/16$	2.988E−04	2.431E−03	9.391E−04
2×10^4	$h/32$	6.551E−05	2.060E−04	2.865E−04
2×10^5	$h/256$	2.903E−05	2.105E−05	2.657E−04

Table 6.3: The errors of the area dilating factor, the total surface area, and the volume.



(a)

(b)

Figure 6.2: (a) A freely suspended vesicle with different stiffness numbers σ_0. $\sigma_0 = 2\times10^3$: '-'; $\sigma_0 = 2\times10^4$: '×'; $\sigma_0 = 2\times10^5$: '·'. (b) The corresponding time evolutions of total energy.

6.3.3 Convergence test

In this subsection, we perform the convergence study for the present numerical algorithm. Again, we put a vesicle with elliptical shape $\mathbf{X}_0(s) = (0.5\cos s, 0.15\sin s)$ in quiescent flow initially. The computational domain is chosen as $[0, 1] \times [-0.5, 0.5]$. We choose the different mesh sizes as $m = 64, 128, 256$ and 512 so the corresponding mesh is $\Delta r = \Delta z = h = 1/m$. Since the analytical solution is not available in this test, we choose the result obtained from the finest mesh $m = 512$ as our reference solution, and compute the maximum error between the reference solution and the numerical solution. The bending coefficient is chosen as $c_b = 2 \times 10^{-2}$ as before, and all the numerical solutions are computed up to time $T = 0.5$. Table 6.4 shows the relative errors of the vesicle surface area and the vesicle volume, the maximum error of the vesicle interface configuration, and the fluid velocity field. Note that, the fluid variables are defined at the staggered grid so when we refine the mesh, the numerical solutions will not coincide with the same grid locations. In these runs, we simply use a cubic spline interpolation to compute the solutions at the desired locations. Due to the fact that the immersed boundary formulation has the singular forcing term in the equations, regularizing the singular term by smoothing discrete delta function causes the method to be first-order accurate. The numerical results shown in Table 6.4 are consistent with what we expect from theory.

	$m = 64$	$m = 128$	rate	$m = 256$	rate
$\lvert A_h - A_0\rvert / A_0$	4.032E−04	2.024E−04	0.99	1.009E−04	1.00
$\lvert V_h - V_0\rvert / V_0$	6.434E−04	1.505E−04	2.10	3.170E−05	2.25
$\lVert \mathbf{X}_h - \mathbf{X}_{\text{ref}}\rVert_\infty$	1.689E−03	4.625E−04	1.87	9.937E−05	2.22
$\lVert u_h - u_{\text{ref}}\rVert_\infty$	2.363E−03	1.162E−03	1.02	5.321E−04	1.13
$\lVert w_h - w_{\text{ref}}\rVert_\infty$	1.876E−03	7.652E−04	1.29	7.069E−04	0.11

Table 6.4: The mesh refinement results for the surface area A_h, the enclosed volume V_h, the configuration $\mathbf{X}_h$, and the velocity field u_h and w_h.

6.3.4 A suspended vesicle in quiescent flow

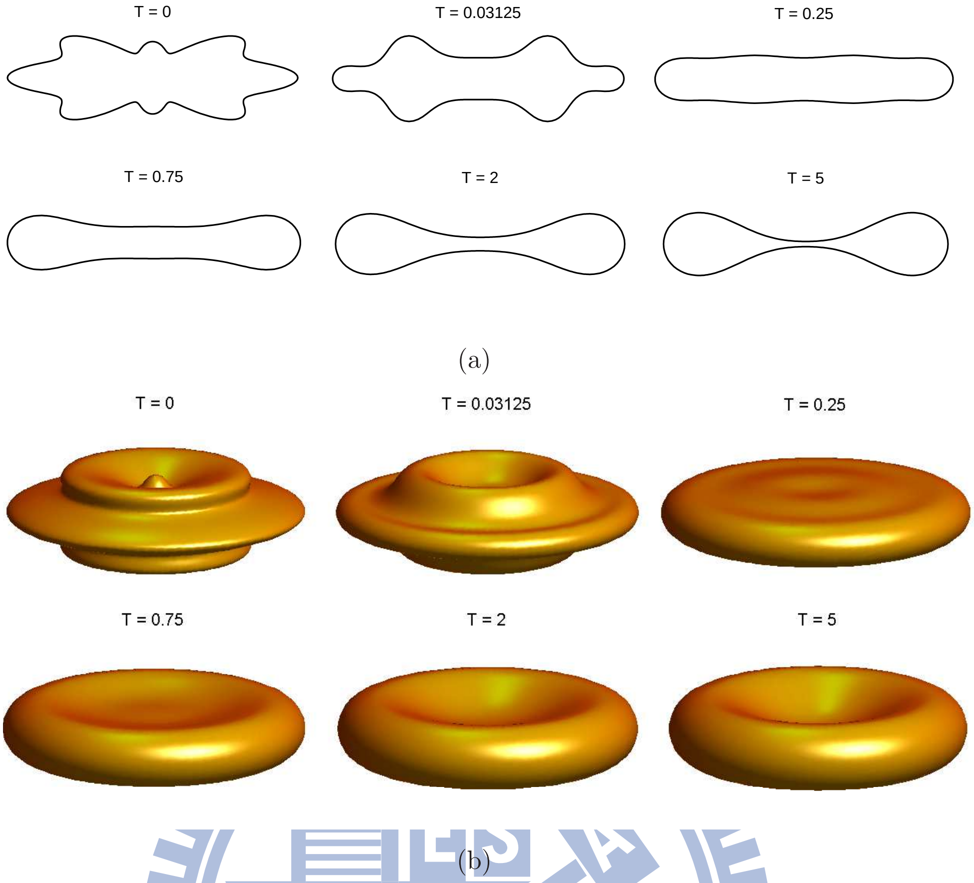
In this test, we put a freely suspended vesicle with an oscillatory initial configuration as described in subsection 6.3.1 in a quiescent flow. The computational domain is the same as previous test and bending coefficient is chosen as $c_b = 5 \times 10^{-2}$. Figure 6.3 shows the snapshots of the vesicle shapes in both cross-sectional and three-dimensional visualizations. One can see that the initially oscillatory interface reaches a stationary bi-concave shape which is consistent with those results in literature [4, 67]. The evolutional plot of the corresponding energy described in Eq. (6.17) is shown in Figure 6.4 so one can see that the total membrane energy is decreasing and tends to a constant indicating that the vesicle is in equilibrium shape eventually.

6.3.5 Vesicles under the gravity

As in [4, 67], we also study the gravitational effect on the shapes of the vesicle by considering different density across the interface. To simulate this problem, we simply add an extra interfacial force $\mathbf{F}_g$ as

$$\mathbf{F}_g = \left(\rho^i - \rho^o\right) (\mathbf{g} \cdot \mathbf{X}) \left|\mathbf{X}_s\right| \mathbf{n},$$

into the vesicle boundary force [49]. Here, ρ^i and ρ^o represent the fluid density inside and outside of the membrane, respectively, and $\mathbf{g}$ is the gravitational field . With



(a)

(b)

Figure 6.3: (a) Snapshots of a freely suspended vesicle in quiescent flow. (b) The corresponding plots result in 3D view.

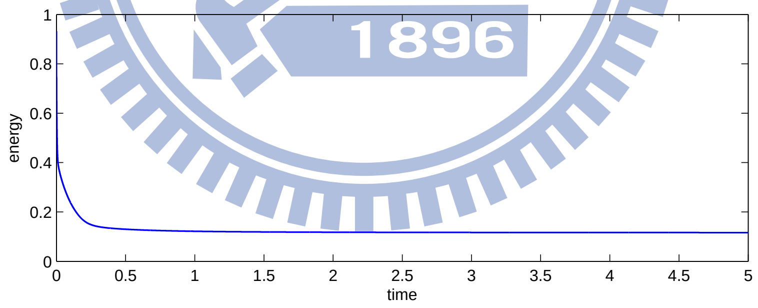


Figure 6.4: The membrane energy evolution of the freely suspended vesicle.

presence of the gravitation, a vesicle drops (since $\rho^i > \rho^o$) and reaches a terminal velocity with an equilibrium shape. Figure 6.5 shows the snapshots of two different initial shapes; namely, an oblate vesicle $\mathbf{X}_0(s) = (0.25 \cos s, 0.125 \sin s)$ and prolate vesicle $\mathbf{X}_0(s) = (0.1531 \cos s, 0.3333 \sin s)$. Notice that, both vesicles have the same

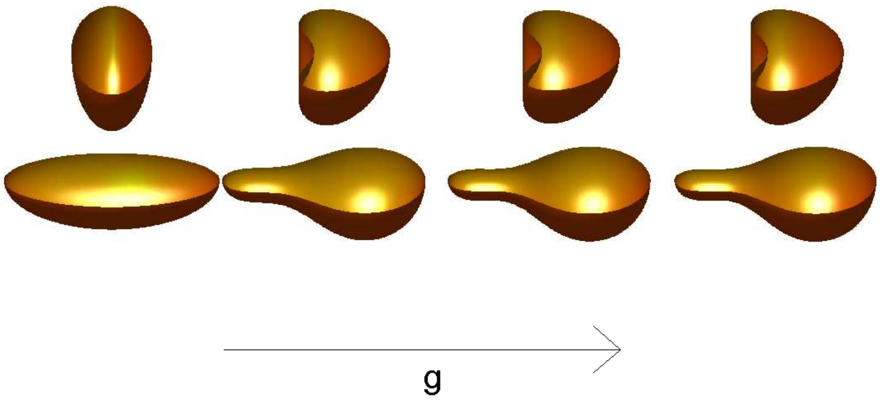


Figure 6.5: Snapshots for the vesicles under the gravity. The gravitational force is pointing into negative z direction. Top: initial oblate shape; bottom: initial prolate shape

volume and surface area. The oblate one (top) converges toward to a "parachute-like" shape while the prolate vesicle (bottom) will converge toward to a "pear-like" shape. Those results are qualitatively consistent with those results obtained in [4, 67].

6.3.6 Vesicles in Poiseuille flow

As mentioned before, vesicles have the similar mechanical properties that mimic RBCs in flows. There have been widely investigated in the rheology of red blood cells passing through capillaries in Poiseuille flows numerically such as [56, 52, 61] and the references therein. Therefore, it will be quite natural to study the vesicle dynamics in Poiseuille flows. Recently, Coupier et. al. [8] study the problem through experiments, numerical, and theoretical computations to characterize the shape diagram of vesicles in Poiseuille flow. In the following, we shall also investigate the problem and compare our axisymmetric results with the ones shown in [8].

In this subsection, we set up the Poiseuille flow as illustrated in Figure 6.6 by

$$w = W_m \left(1 - \frac{r^2}{L^2}\right), \quad u = 0,$$

where L is the capillary radius, and W_m is mean flow velocity indicating the centerline velocity. Throughout this subsection, as in Figure 6.6, we draw the z-axis along the horizontal direction (different from those in previous subsections) since the flow is along the axial direction.

It is known that, in Poiseuille flow, a vesicle reaches its equilibrium shape and then moves with a constant velocity. As discussed in literature [8], the reduced volume ν

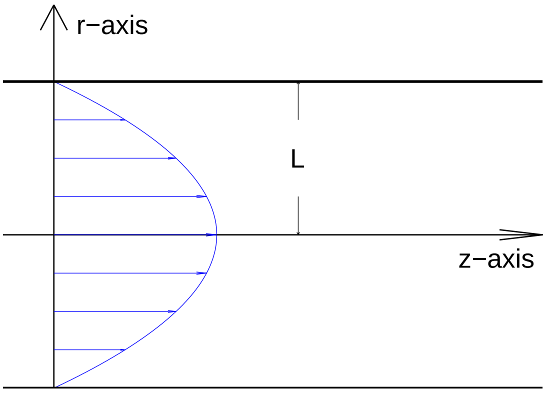


Figure 6.6: The velocity profile of Poiseuille flow.

measured the deflation of the vesicle plays a significant role in vesicle dynamics which is defined as

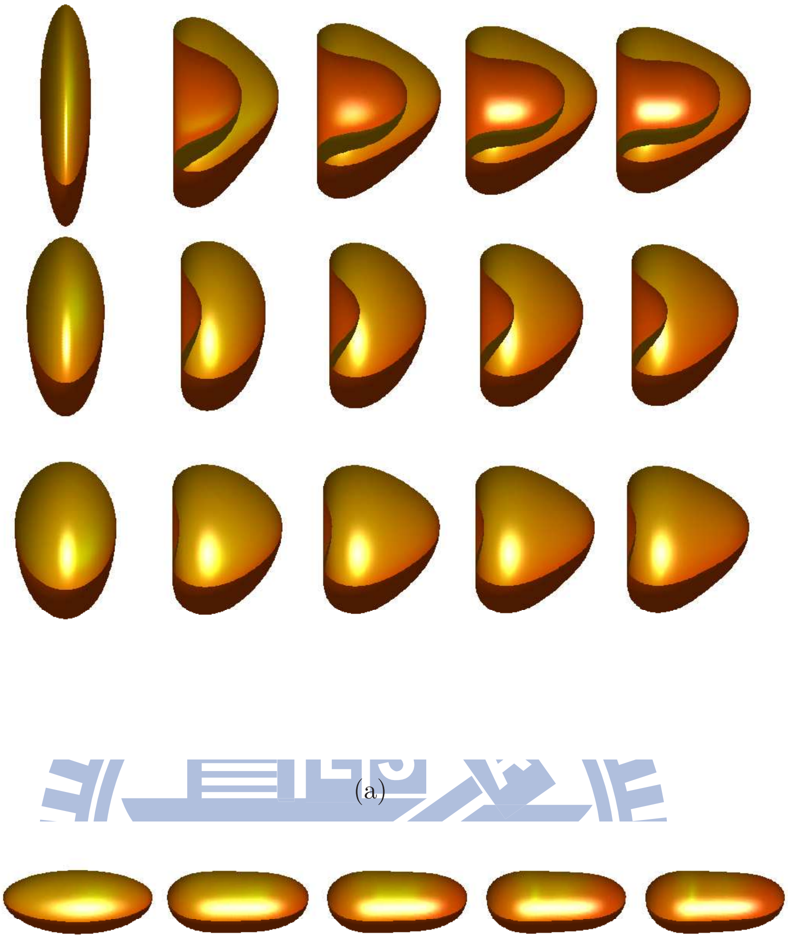
$$\nu = \frac{V_0}{\frac{4}{3}\pi(A_0/4\pi)^{3/2}},$$

where V_0 and A_0 represent the volume and the surface area, respectively. This dimensionless number is nothing but the volume ratio of the vesicle to a sphere with the same surface area. Thus, for a sphere, the reduced volume equals to one. In addition, we define a characteristic non-dimensional parameter $\hat{R}$ to indicate the vesicle confinement in the flow by

$$\hat{R} = \frac{R_0}{L},$$

where the effect radius can be computed by $R_0 = (3V_0/4\pi)^{1/3}$. Thus, a larger $\hat{R}$ means a stronger confinement. In this subsection, we shall investigate three different effects by varying the reduced volume ν, the confinement $\hat{R}$, and the mean velocity W_m individually. The computational $r-z$ domain is chosen as $\Omega = [0, L] \times [-5L, 5L]$. Unless otherwise stated, we all use oblate vesicles as our initial shapes in the flows.

- *Effect of the reduced volume.* In this test, we choose an oblate vesicle with three different reduced volumes $\nu = 0.48, 0.75, 0.9$ but the same volume in a Poiseuille flow with a weaker confinement $\hat{R} = 0.3$ and $W_m = 1$. We run the simulations until the equilibrium shapes are obtained. Figure 6.7(a) shows the snapshots toward to equilibrium shapes for those different reduced volumes. One can see the parachute-like shapes are observed in all three cases while the smaller reduced volume deforms significantly more since it is thinner. This behavior is



(a)

(b)

Figure 6.7: (a) Snapshots of the vesicle in Poiseuille flow with different reduced volume $\nu = 0.48$ (top), $\nu = 0.75$ (middle), and $\nu = 0.9$ (bottom). The flow comes from left to right. (b) A vesicle with initial prolate shape with reduced volume $\nu = 0.9$ results in the bullet-like shape eventually.

consistent with the results obtained in [8]. On the other hand, it is interesting to see that if we choose the prolate vesicle initially (with same surface area and volume as the oblate one with $\nu = 0.9$), we can obtain the bullet-like shape as shown in Figure 6.7(b). Notice that, the parachute-like shape has a concave rear part while the bullet-like shape has a convex one instead.

- *Effect of the confinement.* To study the confinement effect, we keep the mean velocity $W_m = 1$ and reduced volume $\nu = 0.95$ fixed but vary the confinement as $\hat{R} = 0.3, 0.375, 0.5$, respectively. Figure 6.8 shows the cross-sectional view of the equilibrium shapes with different confinements in which the confinement gets stronger from left to right. One can see that the equilibrium shape turns from a parachute-like shape into bullet-like one as the confinement gets stronger. This is a physically interesting result which can be explained as the confinement is sufficiently large, its effect will dominate the other flow effects. This is also in a good agreement with the result obtained in [8].

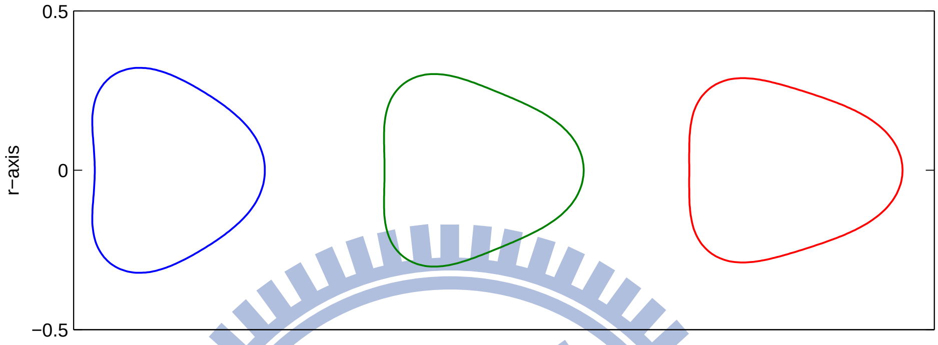


Figure 6.8: Cross-sectional view of the equilibrium shapes with different confinements Left: $\hat{R} = 0.3$; middle: $\hat{R} = 0.375$ and right: $\hat{R} = 0.5$. As the confinement increases from left to right, the shape turns from parachute-like to bullet-like.

- *Effect of the mean velocity.* In the final test, we keep the reduced volume $\nu = 0.95$ and the weaker confinement $\hat{R} = 0.3$ fixed but vary the mean velocity as $W_m = 1, 10, 100$. Notice that, varying the mean velocity alone (but keeping other parameters fixed) means to vary the capillary number $Ca = \frac{\mu R_0^4 W_m}{c_b L^2}$. Thus, the effect can be regarded as the effect of capillary number. Figure 6.9 shows the cross-sectional view of the equilibrium shapes with different mean velocity. By increasing the mean flow velocity, the equilibrium vesicle will turn from parachute-like to an unexpected bullet-like shape. This interesting result has been obtained from the experimental observations [8] (the authors used ν between 0.95 and 0.97) and our result in Figure 6.9 is qualitatively consistent with theirs.

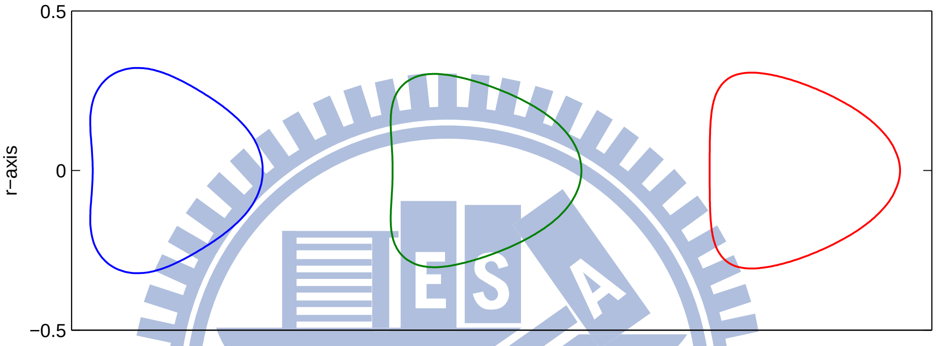


Figure 6.9: Cross-sectional view of the equilibrium shapes with different mean velocity. Left: $W_m = 1$; middle: $W_m = 10$ and right: $W_m = 100$. As the mean flow velocity increases, the shape turns from parachute-like to bullet-like.

Chapter 7

Conclusions and future works

In this thesis, we developed three kinds of numerical methods to simulate vesicle dynamics. The governing equations are formulated based on immersed boundary framework where a mixture of Eulerian fluid variables and Lagrangian interfacial variables is used, with the linkage of these two kinds of variable is by smoothed Dirac delta function. Firstly, We state the model of an inextensible interface (vesicle without bending effect) suspended in Stokes flow. By taking advantage of the property of skew-adjointness between spreading operator and surface divergence operator, the resultant linear system can be solved efficiently by iterative conjugate gradient method. Secondly, we turn our direction to employ the nearly inextensible approach to mimic vesicle dynamics. We proposed an unconditionally energy stable scheme so that the time restriction can be released substantially. This scheme involves solving a positive definite linear system which can be done efficiently by multigrid method. Lastly, we consider the realistic case of three-dimensional vesicle. The vesicle model is approximated by penalized nearly inextensible approach. Moreover, differ to our previous work, the derivative of interfacial variables are evaluated by high accuracy spectral method which outperforms finite difference method. The fluid equation is discretized by projection method and then fast Poisson solver can be used. We investigated vesicle dynamics numerically, such as relaxation to equilibrium state, tank-treading motion under shear flow, with presence of gravity field and shape rheology under Poiseuille flow.

As a next step, we will generalized the present model to with consideration of viscosity contrast which is close to the realistic world. In particular, we plan to extend our work to real three-dimensional space, while the presentation of vesicle shall be treated carefully when it deforms.

Bibliography

[1] J. Adams, P. Swarztrauber, and R. Sweet, Fishpack – a package of Fortran subprograms for the solution of separable elliptic partial differential equations, 1980.

[2] J. H. Bramble, J. E. Pasciak, and A. T. Vassilev, Analysis of the inexact Uzawa algorithm for saddle point problems, SIAM J. Numer. Anal., 34 (1997), 1072-1092.

[3] T. Biben, K. Kassner, C. Misbah, Phase-field approach to three-dimensional vesicle dynamics, Phys. Rev. E, 72 (2005), 041921.

[4] G. Boedec, M. Leonetti, M. Jaeger, 3D vesicle dynamics simulations with a linearly traingulated suface, J. Comput. Phys., 230 (2011), 1020-1034.

[5] J. T. Beale, Partially implicit motion of a sharp interface in Navier-Stokes flow, J. Comput. Phys., 231 (2012), 6159-6172.

[6] H. D. Ceniceros, J. E. Fisher, and A. M. Roma, Efficient solutions to robust, semi-implicit discretizations of the immersed boundary method, J. Comput. Phys., 228 (2009), 7137-7158.

[7] H. D. Ceniceros and J. E. Fisher, A fast, robust, and non-stiff Immersed Boundary Method, J. Comput. Phys., 230 (2011), 5133-5153.

[8] C. Coupier, A. Farutin, C. Minetti, T. Podgorski, C. Misbah, Shape diagram of vesicles in Poiseuille flow, Phys. Rev. Lett., 108 (2012), 178106.

[9] Q. Du, C. Liu, X. Wang, A phase field approach in the numerical study of the elastic bending energy for vesicle membranes, J. Comput. Phys., 198 (2004), 450-468.

[10] J. Deschamps, V. Kantsler, E. Segre, V. Steinberg, Dynamics of a vesicle in general flow, PNAS, 106 (2009), 11444-11447.

[11] V. Doyeux, V. Chabannes, C. Prud'Homme, M. Ismail, Simulation of vesicle using level set method solved by high order finite element, ESIAM, 38 (2013), 335-347.

[12] A. Farutin, T. Biben, C. Misbah, Analytical progress in the theory of vesicles under linear flow, Phys. Rev. E, 81 (2010) 061904.

[13] D. Goldstein, R. Handler and L. Sirovich, Modeling a no-slip flow boundary with an external force field, J. Comput. Phys., 105 (1993) 354-366.

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

[15] B.E. Griffith, R.D. Hornung, D.M. McQueen, and C.S. Peskin, An adaptive, formally second order accurate version of the immersed boundary method, J. Comput. Phys., 223 (2007), 10-49.

[16] R. D. Guy and B. Philip, A multigrid method for a model of the implicit immersed boundary equations, Commun. Comput. Phys., 12 (2012), pp. 378-400.

[17] F. H. Harlow, J. E. Welsh, Numerical calculation of time-dependent viscous incompressible flow of fluid with a free surface, Phys. Fluids, 8 (1965), 2181-2189.

[18] W. Helfrich, Elastic properties of lipid bilayers: Theory and possible experiments, Z Naturforsch C, 28 (1973), 693-703.

[19] K.H. de Haas, C. Blom, D. van den Ende, M.H.G. Duits, and J. Mellema, Deformation of giant lipid bilayer vesicles in shear flow, Physical Review E, 56 (1997).

[20] T.Y. Hou and Z. Shi, Removing the stiffness of elastic force from the immersed boundary method for the 2D Stokes equations, J. Comput. Phys., 227 (2008), 9138-9169.

[21] T. Y. Hou and Z. Shi, An efficient semi-implicit immersed boundary method for the Navier-Stokes equations, J. Comput. Phys., 227 (2008), 8968-8991.

[22] W.-F. Hu and M.-C. Lai, Unconditionally energy stable immersed boundary method with application to vesicle dynamics, East Asian Journal of Applied Mathematics, 3 (2013), 247-262.

[23] W.-F. Hu, Y. Kim and M.-C. Lai, An immersed boundary method for simulating the dynamics of three-dimensional axisymmetric vesicles in Navier-Stokes flows, J. Comput. Phys., 257 (2014), 670-686 .

[24] S.R. Keller and R. Skalak, Motion of a tank-treading ellipsoldal particle in a shear flow, Journal of Fluid Mechanics, 120:27-47, 1982.

[25] M. Kraus, W. Wintz, U. Seifert, and R. Lipowsky, Fluid vesicles in shear flow, Phys. Rev. Lett., 77 (1996).

[26] V. Kantsler and V. Steinberg, Orientation and dynamics of a vesicle in tank-treading motion in shear flow, Phys. Rev. Lett., 95 (2005).

[27] B. Kaoui, G. H. Ristow, I. Cantat, C. Misbah, and W. Zimmermann, Lateral migration of a 2D vesicle in unbounded Poiseuille fliw, Phys. Rev. E 77, 021903 (2008).

[28] Y. Kim and M.-C. Lai, Simulating the dynamics of inextensible vesicles by the penalty immersed boundary method, J. Comput. Phys., 229 (2010), 4840-4853.

[29] R. J. Leveque and Z. Li, The Immersed Interface Method for Elliptic Equations with Discontinuous Coefficients and Singular Sources, SIAM Journal on Numerical Analysis, 31 (1994), 1019-1044.

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

[31] J. L. Guermond, P. Minev, and J. Shen, An overview of projection methods for incompressible flows, Comput. Methods Appl. Mech. Engrg., 195 (2006), 6011-6045.

[32] V. V. Lebedev, K. S. Turitsyn, S. S. Vergeles, Dynamics of nearly spherical vesicles in an external flow, Phys. Rev. Lett., 99 (2007), 218101.

[33] M.-C. Lai, Y.-H. Tseng, and H. Huang, An immersed boundary method for interfacial flow with insoluble surfactant, J. Comput. Phys., 227 (2008) 7279-7293.

[34] M.-C. Lai, C.-Y. Huang, Y.-M. Huang, Simulating the axisymmetric interfacial flows with insoluble surfactant by immered boundary method, International Journal of Numerical Analysis and Modeling, 8 (2011), No 1, 105-117.

[35] Z. Li and M.-C. Lai, New finite difference methods based on IIM for inextensible interfaces in incompressible flows, East Asian Journal of Applied Mathematics, 1 (2011), 155-171 .

[36] A. Laadhari, P. Saramito, C. Misbah, Computing the dynamics of biomembranes by combining conservative level set and adaptive finite element methods, J. Comput. Phys., 263 (2014) 328-352.

[37] M.-C. Lai, W.-F. Hu, W.-W. Lin, A fractional step immersed boundary method for Stokes flow with an inextensible interface enclosing a solid particle, SIAM J. Sci. Comput., 34, No 5 (2012), 692-710.

[38] A.A. Mayo and C.S. Peskin, An implicit numerical method for fluid dynamics problems with immersed elastic boundaries, Contemp. Math., 141 (1993), 261-277.

[39] E. Maitre, G.-H. Cottet, A level set method for fluid-structure interactions with immersed surfaces, Math. Models Meth. Appl. Sci., 16 (2006), 415.

[40] C. Misbah, Vacillating breathing and tumbling of fluid vesicles under shear flow, Phys. Rev. Lett., 96 (2006), 028104.

[41] E. Maitre, T. Milcent, G.-H. Cottet, A. Raoult, Y. Usson, Applications of level set methods in computational biophysics, Math. Comput. Model., 49 (2009), 2161-2169.

[42] Y. Mori and C.S. Peskin, Implicit second-order immersed boundary methods with boundary mass, Comput. Methods Appl. Mech. Engrg., 197 (2008), 2049-2067.

[43] E. Maitre, C. Misbah, P. Peyla, A. Raoult, Comparison between advected-field and level-set methods in the study of vesicle dynamics, Physica D: Nonlinear Phenomena, 241 (2012), 1146-1157.

[44] E.P. Newren, A.L. Fogelson, R.D. Guy, and R.M. Kirby, Unconditionally stable discretizations of the immersed boundary equations, J. Comput. Phys., 222 (2007), 702-719.

[45] H. Noguchi, G. Gompper, Swinging and tumbling of fluid vesicles in shear flow, Phys. Rev. Lett., 98 (2007), 128103.

[46] E.P. Newren, A.L. Fogelson, R.D. Guy, and R.M. Kirby, A comparison of implicit solvers for the immersed boundary equations, Comput. Methods Appl. Mech. Engrg., 197 (2008), 2290-2304.

[47] Y. Notay, An aggregation-based algebraic multigrid method, Electronic Transactions on Numerical Analysis, 37 (2010), 123-146.

[48] C. Peskin, Flow patterns around heart valves: A numerical method. J. Comput. Phys., 10 (1972), 252-271.

[49] C. Pozrikidis, Introduction to Theoretical and Computational Fluid Dynamics, Oxford University Press, 1997.

[50] C. Peskin, The immersed boundart method, Acta Numer., 11 (2002), 479-517.

[51] A. Pressley, Elementary Differential Geometry, Springer 2000.

[52] C. Pozrikidis, Axisymmetric motion of a file of red blood cells through capillaries, Physics of fluid, 17 (2005), 031503.

[53] J. Peters, V. Reichelt, and A. Reusken, Fast iterative solvers for discrete Stokes equations, SIAM J. Sci. Comput., 27 (2005), 646-666.

[54] A. Roma, C. Peskin, M. Berger, An Adaptive Version of the Immersed Boundary Method, J. Comput. Phys., 153 (1999), 509-534.

[55] J. C. Strikwerda, An iterative method for solving finite difference approximations to the Stokes equations, SIAM J. Numer. Anal., 21 (1984), 447-458.

[56] T. W. Secomb, R. Skalak, N. Ozkaya, J. F. Gross, Flow of axisymmetric red blood cells in narror capillaries, J. Fluid. Mech., 163 (1986), 405-423.

[57] E. M. Saiki and S. Biringen, Numerical simulation of a cylinder in uniform flow: application of a virtual boundary method, J. Comput. Phys., 123 (1996), 450.

[58] J.M. Stockie and B.R. Wetton, Analysis of stiffness in the immersed boundary method and implications for time-stepping schemes, J. Comput. Phys., 154 (1999), 41-64.

[59] J.S. Sohn, Y.-H. Tseng, S. Li, A. Voigt and J.S. Lowengrub, Dynamics of multicomponent vesicles in a viscous fluid, J. Comput. Phys., 229 (2010), 119-144.

[60] D. Salac, M. Miksis A level set projection model of lipid vesicles in general flows, J. Comput. Phys, 230 (2011), 8192-8215.

[61] L. Shi, T. W. Pan, R. Glowinski, Deformation of a single red blood cell in bounded Poiseuille flows, Phys. Rev. E, 85 (2012), 016307.

[62] C. Tu and C.S. Peskin, Satbility and instability in the computation of flows with moving immersed boundaries: a comparison of three methods, SIAM J. Sci. Stat. Comput., 13 (1992), 1361-1376.

[63] L. N. Trefethen, Spectral Methods in MATLAB, SIAM 2000.

[64] K. Taira and T. Colonius, The immersed boundary mehtod: A projection approach, J. Comput. Phys., 225 (2007), 2118-2137.

[65] Z. Tan, Duc-Vinh Le, K.M. Lim, B.C. Khoo, An Immersed Interface Method for the Simulation of Inextensible Interfaces in Viscous Fluids, Commun. Comput. Phys. 11 (2012), 925-950.

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

[67] S.K. Veerapaneni, D. Gueyffier, D. Zorin, G. Biros, A numerical method for simulating the dynamics of 3D axisymmetric vesicles suspended in viscous flows, J. Comput. Phys., 228 (2009), 7233-7249.

[68] S.K. Veerapaneni, A. Rahimian, G. Biros, D. Zorin, A fast algorithm for simulating vesicle flows in three dimensions, J. Comput. Phys., 230 (2011), 5610-5634.

[69] J. L. Weiner, On a problem of Chen, Willmore, et. al., Indiana University Mathematical Journal, 27 (1978) 19-35.

[70] X. Yang, X. Zhang, Z. Li, G.-W. He, A smoothing technique for discrete delta functions with application to immersed bounary method in moving boundary simulations, J. Comput. Phys., 228 (2009), 7821-7836.

[71] H. Zhou, C. Pozrikidis, Deformation of liquid capsules with incompressible interfaces in simple shear flow, J. Fluid Mech., 283 (1995), 175-200.

[72] H. Zhao, E.S. G. Shaqfeh, The dynamics of a vesicle in simple shear flow, J. Fluid Mech., 674 (2011), 578-604.

Appendix A

Geometrical operators and quantities on a surface

In this appendix, we provide computation process of mean curvature H and Gaussian curvature K which can be obtained by first and second fundamental forms [51] and geometrical differential operators on surface.

A three dimensional surface can be presented by parametric form as $\mathbf{X}(\alpha, \beta) = (X(\alpha, \beta), Y(\alpha, \beta), Z(\alpha, \beta))$, $0 \leq \alpha \leq L_\alpha$ and $0 \leq \beta \leq L_\beta$, where α and β are two Lagrangian parameters. From first fundamental form, we have

$$E = \mathbf{X}_\alpha \cdot \mathbf{X}_\alpha, \quad F = \mathbf{X}_\alpha \cdot \mathbf{X}_\beta, \quad G = \mathbf{X}_\beta \cdot \mathbf{X}_\beta, \quad W = \sqrt{EG - F^2};$$

by second fundamental form, we have

$$L = \mathbf{X}_{\alpha\alpha} \cdot \mathbf{n}, \quad M = \mathbf{X}_{\alpha\beta} \cdot \mathbf{n}, \quad N = \mathbf{X}_{\beta\beta} \cdot \mathbf{n},$$

where unit normal vector can be obtained by $\mathbf{n} = \mathbf{X}_\alpha \times \mathbf{X}_\beta / W$. With these elementary geometrical quantities, the mean curvature H and Gaussian curvature K is computed by

$$H = -\frac{1}{2} \frac{EN - 2FM + GL}{W^2}, \quad K = \frac{LN - M^2}{W^2}.$$

Suppose ϕ is a scalar function and $\mathbf{f}$ is a vector function, then we have surface gradient ∇_s, surface divergence $\nabla_s \cdot$ and surface Laplace operator Δ_s (or Laplace-Beltrami operator) as follows.

$$\nabla_s \phi = \frac{G\mathbf{X}_\alpha - F\mathbf{X}_\beta}{W^2} \phi_\alpha + \frac{E\mathbf{X}_\beta - F\mathbf{X}_\alpha}{W^2} \phi_\beta,$$

$$\nabla_s \cdot \mathbf{f} = \frac{G\mathbf{f}_\alpha - F\mathbf{f}_\beta}{W^2} \cdot \mathbf{X}_\alpha + \frac{E\mathbf{f}_\beta - F\mathbf{f}_\alpha}{W^2} \cdot \mathbf{X}_\beta,$$

$$\Delta_s \phi = \frac{1}{W} \left[\left(\frac{G\phi_\alpha - F\phi_\beta}{W} \right)_\alpha + \left(\frac{E\phi_\beta - F\phi_\alpha}{W} \right)_\beta \right].$$

Appendix B

Discrete skew-adjoint operators

In this appendix, we give a direct derivation to the matrix obtained from the discrete spreading operator S_h of σ_h and the matrix obtained from discrete surface divergence operator ∇_{s_h} of $\mathbf{f}U$ are transpose with each other. First, let us rewrite the operator $S_h(\sigma_h)$ by

$$\begin{aligned}
S_h(\sigma_h) &= \sum_{k=0}^{M-1} D_s\left(\sigma\boldsymbol{\tau}\right)_k \delta_h(\mathbf{x}-\mathbf{X}_k)\,\Delta s \\
&= \sum_{k=0}^{M-1} \frac{\sigma_{k+1/2}\,\boldsymbol{\tau}_{k+1/2}-\sigma_{k-1/2}\,\boldsymbol{\tau}_{k-1/2}}{\Delta s}\delta_h(\mathbf{x}-\mathbf{X}_k)\Delta s \\
&= \sum_{k=1}^{M} \sigma_{k-1/2}\,\boldsymbol{\tau}_{k-1/2}\,\delta_h(\mathbf{x}-\mathbf{X}_{k-1}) - \sum_{k=1}^{M} \sigma_{k-1/2}\boldsymbol{\tau}_{k-1/2}\delta_h(\mathbf{x}-\mathbf{X}_k) \\
&- \;\sigma_{-1/2}\boldsymbol{\tau}_{-1/2}\delta_h(\mathbf{x}-\mathbf{X}_0)+\sigma_{M-1/2}\boldsymbol{\tau}_{M-1/2}\delta_h(\mathbf{x}-\mathbf{X}_M) \\
&= \sum_{k=1}^{M} \left(\delta_h(\mathbf{x}-\mathbf{X}_{k-1})-\delta_h(\mathbf{x}-\mathbf{X}_k)\right)\boldsymbol{\tau}_{k-1/2}\,\sigma_{k-1/2}.
\end{aligned}$$

Note that, the last two terms are cancelled out due to the periodicity of the interface. Now we can write down the discrete operator ∇_{s_h} as

$$\begin{aligned}
\nabla_{s_h}\cdot\mathbf{U}_k &= \frac{\mathbf{U}_k-\mathbf{U}_{k-1}}{\Delta s}\cdot\boldsymbol{\tau}_{k-1/2}/\left|D_s\mathbf{X}\right|_{k-1/2} \\
&= -\frac{h^2}{\Delta s\left|D_s\mathbf{X}\right|_{k-1/2}}\sum_{\mathbf{x}}\left(\delta_h(\mathbf{x}-\mathbf{X}_{k-1})-\delta_h(\mathbf{x}-\mathbf{X}_k)\right)\boldsymbol{\tau}_{k-1/2}\cdot\mathbf{u}_{i,j}.
\end{aligned}$$

Since the discrete surface divergence operator of the velocity is zero as described in Eq. (4.11), we can scale out the coefficient $-\frac{h^2}{\Delta s\left|D_s\mathbf{X}\right|_{k-1/2}}$ so that the resultant matrices obtained from S_h and $\nabla_{s_h}\cdot$ are transpose to each other.

Appendix C

Unconditionally stable IB method

In this appendix, we provide the numerical detail on how we compute the bending force in Eq. (5.36). To proceed, we first define the discrete force spreading operator $\mathcal{F}_h^n$, and the velocity interpolating operator $\mathcal{I}_h^n$ by

$$\mathcal{F}_h^n[\mathbf{F}](\mathbf{x}) = \sum_{k=1}^{M} \mathbf{F}(s_k)\delta_h(\mathbf{x}-\mathbf{X}_k^n)\Delta s, \qquad \mathcal{I}_h^n[\mathbf{u}](s_k) = \sum_{\mathbf{x}} \mathbf{u}(\mathbf{x})\delta_h(\mathbf{x}-\mathbf{X}_k^n)\, h^2, \tag{C.1}$$

respectively. It is well-known [50, 44] that the above both operators are adjoint with each other as

$$\begin{aligned}\langle \mathcal{F}_h^n[\mathbf{F}], \mathbf{u}\rangle_{\Omega_h} &= \sum_{\mathbf{x}} \left(\sum_{k=1}^{M} \mathbf{F}(s_k)\delta_h(\mathbf{x}-\mathbf{X}_k^n)\Delta s \right) \cdot \mathbf{u}(\mathbf{x})\, h^2 \\ &= \sum_{k=1}^{M} \mathbf{F}(s_k) \cdot \left(\sum_{\mathbf{x}} \mathbf{u}(\mathbf{x})\delta_h(\mathbf{x}-\mathbf{X}_k^n)\, h^2 \right) \Delta s = \langle \mathbf{F}, \mathcal{I}_h^n[\mathbf{u}]\rangle_{\Gamma_h}.\end{aligned}$$

The singular immersed boundary force arising from the bending is written as

$$\mathbf{f}_b(\mathbf{x}) = -c_b \int_{\Gamma} \frac{\partial^4 \mathbf{X}}{\partial s^4} \delta(\mathbf{x} - \mathbf{X}(s,t))\, \mathrm{d}s. \tag{C.2}$$

To simplify our notations, we define the discrete fourth-order centered difference operator $\mathcal{A}_h$ as an approximation to the fourth-order derivative. Thus, the discretization for Eqs. (C.2) can be written as

$$\mathbf{f}_b^{n+1}(\mathbf{x}) = -c_b \mathcal{F}_h^n[\mathcal{A}_h \mathbf{X}^{n+1}](\mathbf{x}).$$

By substituting $\mathbf{X}^{n+1} = \mathbf{X}^n + \Delta t \mathcal{I}_h^n[\mathbf{u}^{n+1}]$ into the above equation, we have

$$\begin{aligned}\mathbf{f}_b^{n+1}(\mathbf{x}) &= -c_b \mathcal{F}_h^n[\mathcal{A}_h \mathbf{X}^n + \Delta t \mathcal{A}_h \mathcal{I}_h^n[\mathbf{u}^{n+1}]](\mathbf{x}) \\ &= -c_b \mathcal{F}_h^n[\mathcal{A}_h \mathbf{X}^n](\mathbf{x}) - c_b \Delta t \mathcal{F}_h^n \mathcal{A}_h \mathcal{I}_h^n[\mathbf{u}^{n+1}](\mathbf{x}).\end{aligned} \tag{C.3}$$

Since the discrete operators $\mathcal{F}_h^n$ and $\mathcal{I}_h^n$ are both adjoint to each other and the discrete fourth differential operator $\mathcal{A}_h$ is symmetric positive definite, we can conclude that the above composite operator $\mathcal{F}_h^n \mathcal{A}_h \mathcal{I}_h^n$ is also symmetric positive definite. So in our modified BE scheme, we only need to add the additional singular force $\mathbf{f}_b^*(\mathbf{x})$ as

$$\mathbf{f}_b^*(\mathbf{x}) = -c_b \mathcal{F}_h^n[\mathcal{A}_h \mathbf{X}^n](\mathbf{x}) - c_b \Delta t \mathcal{F}_h^n \mathcal{A}_h \mathcal{I}_h^n[\mathbf{u}^*](\mathbf{x}). \tag{C.4}$$

One should note that, the symmetric positive definite matrix structure for $\mathbf{u}^*$ in Eq. (5.24) does not change at all even we add this extra bending force term. The bending term for modified CN scheme can be similarly derived so we omit the detail here.

Appendix D

Geometrical differential operators in axisymmetric coordinate

In this Appendix, we derive the mean curvature H, Gaussian curvature K, and the surface Laplacian of mean curvature $\Delta_s H$ in axisymmetric coordinates as shown in Eq. (6.10). The mean curvature H and Gaussian curvature K can be computed by the the first and second fundamental forms [51]. As before, under axisymmetric assumption, the interface can be parameterized as

$$\mathbf{X}(s,\theta) = (R(s)\cos\theta, R(s)\sin\theta, Z(s)).$$

Here, we omit the dependence of time t for simplicity. The coefficients E, F, G of the first fundamental form for the above surface, and the surface area dilating factor W can be obtained as

$$\begin{aligned}
E &= \mathbf{X}_s \cdot \mathbf{X}_s = |\mathbf{X}_s|^2, \\
F &= \mathbf{X}_s \cdot \mathbf{X}_\theta = 0, \\
G &= \mathbf{X}_\theta \cdot \mathbf{X}_\theta = R^2, \\
W &= \sqrt{EG - F^2} = R\,|\mathbf{X}_s|.
\end{aligned}$$

The unit outward normal vector is

$$\mathbf{n} = \frac{\mathbf{X}_\theta \times \mathbf{X}_s}{W} = \frac{1}{|\mathbf{X}_s|}(Z_s\cos\theta, Z_s\sin\theta, -R_s).$$

The coefficients of the second fundamental form L, M and N can be obtained as

$$\begin{aligned}
L &= \mathbf{X}_{ss} \cdot \mathbf{n} = \frac{1}{|\mathbf{X}_s|}(R_{ss}Z_s - R_s Z_{ss}), \\
M &= \mathbf{X}_{s\theta} \cdot \mathbf{n} = 0, \\
N &= \mathbf{X}_{\theta\theta} \cdot \mathbf{n} = \frac{-RZ_s}{|\mathbf{X}_s|}.
\end{aligned}$$

Therefore, the mean curvature H and Gaussian curvature K can be computed explicitly as

$$\begin{aligned} H &= -\frac{1}{2}\frac{EN-2FM+GL}{W^2} = \frac{1}{2}\left(\frac{Z_s}{R\,|\mathbf{X}_s|} + \frac{R_sZ_{ss}-R_{ss}Z_s}{|\mathbf{X}_s|^3}\right), && \text{(D.1)} \\ K &= \frac{LN-M^2}{W^2} = \frac{Z_s(R_sZ_{ss}-R_{ss}Z_s)}{R\,|\mathbf{X}_s|^4}. && \text{(D.2)} \end{aligned}$$

It is interesting to note that there is another way to derive the above curvatures H and K. Using the formula of the surface divergence operator derived in Eq. (6.14) and substituting the normal vector into the definition of $2H = \nabla_s \cdot \mathbf{n}$, we obtain the mean curvature H by

$$2H = \nabla_s \cdot \mathbf{n} = \frac{Z_s}{R\,|\mathbf{X}_s|} + \left(\frac{\partial \mathbf{n}}{\partial s}\frac{1}{|\mathbf{X}_s|}\right)\cdot \boldsymbol{\tau} = \frac{Z_s}{R\,|\mathbf{X}_s|} + \frac{R_sZ_{ss}-R_{ss}Z_s}{|\mathbf{X}_s|^3}. \qquad \text{(D.3)}$$

One can immediately see that those two terms in above equation comprise two principal curvatures, thus their product leads to the Gaussian curvature as shown in (D.2).

Note that, the surface gradient operator in axisymmetric coordinates is defined as

$$\nabla_s\, H = \frac{\partial H}{\partial \boldsymbol{\tau}}\boldsymbol{\tau} = \left(\frac{1}{|\mathbf{X}_s|}\frac{\partial H}{\partial s}\right)\boldsymbol{\tau},$$

where the tangent vector is defined in Eq. (6.12). By applying the surface divergence operator (as defined in Eq. (6.14)) to the above equation, one can obtain

$$\Delta_s H = \nabla_s \cdot (\nabla_s H) = \frac{1}{R\,|\mathbf{X}_s|}\frac{\partial}{\partial s}\left(\frac{R}{|\mathbf{X}_s|}\frac{\partial H}{\partial s}\right). \qquad \text{(D.4)}$$