Journal of Computational Physics 408 (2020) 109277

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

An immersed boundary projection method for simulating the inextensible vesicle dynamics

Kian Chuan Ong [a,*], Ming-Chih Lai [b]

[a] National Center for Theoretical Sciences, No. 1, Sec. 4, Roosevelt Road, Astronomy-Mathematics Building, National Taiwan University, Taipei 10617, Taiwan
[b] Department of Applied Mathematics, National Chiao Tung University, 1001, Ta Hsueh Road, Hsinchu 300, Taiwan

ARTICLE INFO

Article history:
Received 7 July 2019
Received in revised form 14 January 2020
Accepted 18 January 2020
Available online 22 January 2020

Keywords:
Immersed boundary projection method
Fractional step method
Unconditionally energy stable scheme
Inextensible interface
Bending force
Vesicle dynamics

ABSTRACT

We develop an immersed boundary projection method (IBPM) based on an unconditionally energy stable scheme to simulate the vesicle dynamics in a viscous fluid. Utilizing the block LU decomposition of the algebraic system, a novel fractional step algorithm is introduced by decoupling all solution variables, including the fluid velocity, fluid pressure, and the elastic tension. In contrast to previous works, the present method preserves both the fluid incompressibility and the interface inextensibility at a discrete level simultaneously. In conjunction with an implicit discretization of the bending force, the present method alleviates the time-step restriction, so the numerical stability is assured by non-increasing total discrete energy during the simulation. The numerical algorithm takes a linearithmic complexity by using preconditioned GMRES and FFT-based solvers. The grid convergence studies confirm the solution variables exhibit first-order convergence rate in L^2-norm. We demonstrate the numerical results of the vesicle dynamics in a quiescent fluid, Poiseuille flow, and shear flow, which are congruent with the results in the literature.

1. Introduction

The immersed boundary (IB) method [1] is a classical numerical method for simulating the fluid-structure interaction that has been applied to problems ranging from locomotion in an organism to dynamics of soft matters including droplets, bubbles, and vesicles. The fundamental principle of the IB method is based on a Lagrangian description of the interface immersed in an Eulerian description of the fluid domain. The interaction between the fluid and the interface is coupled through the convolution with a regularized delta function. Hence, the IB method requires minimal geometric information to handle complex moving boundary without the need for sophisticated mesh-generation.

The mathematical modeling of the morphological dynamics of biological cell membranes has received increased attention in biology, biophysics, and bioengineering community. In particular, inextensible vesicle has been considered as a viable model of biological cells for simulating various biophysical phenomena such as red blood cells [2] and drug-delivery capsules [3]. The transient dynamics are characterized by the interaction between membrane elastic energy, inextensibility, bending force, and hydrodynamic force, which pose a range of challenges to develop robust numerical schemes in terms of computational cost, accuracy, and numerical stability. One of the major numerical difficulties is to impose the local inextensibility constraint. In contrast to the regular multiphase flow problems, the elastic tension of an inextensible interface is

[*] Corresponding author.
E-mail addresses: kcong@ncts.ntu.edu.tw (K.C. Ong), mclai@math.nctu.edu.tw (M.-C. Lai).

https://doi.org/10.1016/j.jcp.2020.109277
0021-9991/© 2020 Published by Elsevier Inc.

unknown a priori and acts as a Lagrange multiplier to enforce the local inextensibility constraint on the interface which is analogous to the role of pressure to enforce the fluid incompressibility constraint. The local inextensibility constraint of the interface has been considered within the boundary integral method [4–10], phase field method [11–14], IB method [15–20], immersed interface method [21,22], and level set method [13,23–25,27].

A gradient-augmented level set method was proposed by [23,24] to model the inextensible vesicle by regularizing the fluid stress tensor with a delta function to localize the force at the interface. An iterative four-step projection method is used to enforce the fluid incompressibility and local inextensibility constraints. However, the projection method does not exactly preserve the fluid incompressibility constraint due to the explicit treatment of the projection step to determine the Lagrange multiplier. In [25,26], a saddle-point approach is used to compute the level set equation within the finite element framework. Additional ad-hoc Lagrange multipliers that do not generate additional forces in the fluid are incorporated into the level set equation to enhance the conservation of area and arc-length. On the contrary, higher order polynomial approximations are used in [27] instead of additional Lagrange multipliers within a similar framework. In [13], a penalty formulation is used in conjunction with level set formulation to penalize the local stretching of the interface in order to maintain a nearly inextensible constraint of the membrane. Consequently, an empirical parameter is introduced, and a severe time-step restriction must be imposed to ensure numerical stability.

In the context of phase field method, a Lagrange multiplier was introduced by [11,12] to satisfy a time-dependent advection-diffusion-reaction equation of the tension where the temporal derivative is proportional to the surface divergence of the velocity. The diffusion term is included and acts as a regularization term to enhance numerical stability. Nevertheless, the local inextensibility constraint is not fully preserved for a finite thickness of the diffuse interface as justified by the asymptotic analysis [12]. Instead of solving the dynamics of the tension, the authors introduced an equation for the local Lagrange multiplier whose solution provides a harmonic extension of the local Lagrange multiplier adjacent to the interface while maintaining the local inextensibility constraint [14]. A novel feature is that the accuracy of the local inextensibility constraint is explicitly controlled by a regularization parameter that scales with the square of the interface thickness for which the asymptotic analysis can justify the convergence to the sharp interface model. Besides, a local relaxation scheme is also proposed to correct the local stretching/compression errors dynamically.

A two-dimensional boundary integral method was proposed by [5] to simulate the motion of inextensible interface in a viscous fluid. The formulation adopts a spectral approximation in space and a semi-implicit time-stepping scheme in time to circumvent the severe time-step restriction. The work is extended to axisymmetric case [6] and to the general three-dimensional case of vesicles [8]. On the contrary to 2D, the three-dimensional boundary integral formulation is based on Lagrangian spectral collocation points defined on the interface so the spectral accuracy in space can be achieved by using spherical harmonics discretization. Despite the high accuracy obtained, the spectral representation of the interface possess several disadvantages due to the non-locality of the representation. In practice, an interface may be deformed to a dumbbell shape in a shear flow or slipper shape in Poiseuille flow which rather inefficient to be represented by the spherical harmonics since the interface contains a small region of high curvatures. As an alternative, the triangular mesh has been employed by [7,9,10] in the context of boundary integral method for the interface parametrization. In [7], the local inextensibility constraint is satisfied by a Lagrange multiplier which acts as an elastic tension. Subsequently, the Lagrange multiplier is eliminated from the equations by constructing a projection operator acting on the velocity field, and the numerical stability restriction is circumvented. On the other hand, an optimized penalization formulation is used by [10] to solve for the elastic tension. By systematically selecting the optimal values for the local and global strain penalization parameters at each time-step, it preserves the inextensibility constraint better than the conventional penalization formulation while circumvents the severe time-step restriction due to the numerical stiffness of the elastic tension force.

The penalty IB formulation has been adopted to impose the local inextensibility constraint in 2D [15–17,20] and 3D [18,19]. By modifying the elastic tension energy, the unknown elastic tension is replaced by a Hookean elastic tension based on the local interface configuration. The formulation avoids solving the coupled system directly and simplifies the overall numerical algorithm. However, there exist several drawbacks in the penalty IB formulation. Firstly, the ad-hoc modification of elastic tension energy inevitably introduces an empirical penalization parameter of σ_0. The parameter highly depends on the problem, and its optimal value may be varied in space and time throughout the simulation. Therefore the accuracy of the local inextensibility constraint is challenging to control explicitly and uniformly. In principle, the inextensibility constraint should be fully recovered as the parameter σ_0 tends to be infinity. Although the numerical algorithm is simplified and the elastic tension can be computed explicitly in a segregated manner, the numerical stability is severely hindered due to the large value of σ_0 and the explicit discretization of bending force. As a consequence, the computational time-step is restricted to $O(h^2)$ or higher in practice. Hence, satisfying the discrete local inextensibility constraint consistently up to the machine precision or a prescribed tolerance while maintaining numerical stability remains a critical numerical issue in the context of the IB method.

The main objective of the present study is to develop an immersed boundary projection method (IBPM) for simulating the inextensible vesicle dynamics in a viscous fluid. To ensure the numerical stability, we discretize the governing equations using the unconditionally energy stable scheme recently developed by the authors in [28]. Based on the previous analysis in [29,30], we propose a novel numerical algorithm to decouple all the solution variables while maintaining the numerical stability, accuracy, and efficiency. The computational framework is compelling such that it offers excellent versatility in the implementation and future extension. The remainder of the paper is organized as follows. In Section 2, we define the governing equations of an inextensible vesicle immersed in an incompressible viscous fluid domain. In Section 3, we describe the details of the staggered finite difference framework and the interface representation by Lagrangian markers, followed by the numerical discretization of the governing equations. In Section 4, the detailed numerical algorithm of the IBPM for the inextensible vesicle is presented. We demonstrate the discrete energy stability and the grid convergence rate under different flow configurations in Section 5. The conservation of the enclosed area and arc-length are investigated quantitatively, and the performance of the iterative solver is also evaluated carefully in this section. We further explore the dynamics of a vesicle in a quiescent fluid, Poiseuille flow, and shear flow by varying the critical parameters related to the reduced area and characteristic velocity. The numerical results are justified by comparing to the existing analytical and numerical results obtained from the literature. Finally, we summarize the conclusion of the present study in Section 6.

2. Governing equations of motion

The practical interest of the present study lies in the low-Reynolds number Stokes flow, where the convective term is negligible in this regime. Consider the governing equations for a vesicle immersed in a two-dimensional incompressible viscous fluid domain Ω,

$$\frac{\partial \mathbf{u}}{\partial t} - \nu \Delta \mathbf{u} + \nabla p = S(\mathbf{F}_\sigma) + S(\mathbf{F}_b) \quad \text{in } \Omega, \tag{1}$$

$$S(\mathbf{F}_\sigma) = \int_\Gamma \frac{\partial}{\partial s}(\sigma \boldsymbol{\tau}) \delta(\mathbf{x} - \mathbf{X}(s,t)) \, ds, \tag{2}$$

$$S(\mathbf{F}_b) = -c_b \int_\Gamma \frac{\partial^4 \mathbf{X}}{\partial s^4} \delta(\mathbf{x} - \mathbf{X}(s,t)) \, ds, \tag{3}$$

$$\nabla \cdot \mathbf{u} = 0, \quad \text{in } \Omega, \tag{4}$$

$$\nabla_s \cdot \mathbf{U} = \frac{\partial \mathbf{U}}{\partial s} \cdot \boldsymbol{\tau} = 0 \quad \text{on } \Gamma, \tag{5}$$

$$\frac{\partial \mathbf{X}}{\partial t}(s,t) = \mathbf{U}(s,t) = \int_\Omega \mathbf{u}(\mathbf{x},t) \delta(\mathbf{x} - \mathbf{X}(s,t)) \, d\mathbf{x}, \quad \text{on } \Gamma, \tag{6}$$

where $\mathbf{u}$ is the fluid velocity, p the fluid pressure, and ν the kinematic viscosity. The enclosed interface (or immersed vesicle boundary) Γ is represented by a set of Lagrangian markers $\mathbf{X}(s,t)$ with the arc-length parameter s. The notations $\boldsymbol{\tau}(s,t) = \partial \mathbf{X}/\partial s$ and c_b represent the unit tangent vector and the constant bending rigidity, respectively. There exists two types of forcing arising from the vesicle boundary Γ, i.e., elastic tension force $\mathbf{F}_\sigma$ in Equation (2) and bending force $\mathbf{F}_b$ in Equation (3), derived from the variational derivative of Helfrich type energy [31].

3. Numerical discretization

The fluid variables are defined on the staggered finite difference grid in a computational rectangular domain $\Omega = [a,b] \times [c,d]$ as illustrated in Fig. 1. The pressure p is defined at the cell-center $\mathbf{x} = (x_i, y_j) = (a + (i-1/2)\Delta x, c + (j-1/2)\Delta y)$, $i = 1, 2, \cdots, m$ and $j = 1, 2, \cdots, n$, while the velocity components u and v are defined at cell-edges $(x_{i-1/2}, y_j) = (a + (i-1)\Delta x, c + (j-1/2)\Delta y)$ and $(x_i, y_{j-1/2}) = (a + (i-1/2)\Delta x, c + (j-1)\Delta y)$, respectively. For brevity, we assume a uniform grid size of $h = \Delta x = \Delta y$. To represent the immersed vesicle boundary, we employ M discrete points $s_k = k\Delta s, k = 0, 1, \cdots M-1$ with the marker size Δs comparable to the grid size h. The Lagrangian markers are therefore denoted by $\mathbf{X}_k = \mathbf{X}(s_k)$, and $\mathbf{X}_M = \mathbf{X}_0$ since the vesicle boundary is closed. The values of elastic tension σ are defined at the half-integer points between the Lagrangian markers $s_{k-1/2} = (k - 1/2)\Delta s$ and they are represented as $\sigma_{k-1/2}$.

In the present study, we discretize Equations (1) to (6) by the backward Euler scheme developed in [28] as

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} - \nu \Delta_h \mathbf{u}^{n+1} + \nabla_h p^{n+1} = S_h^n\left(\mathbf{F}_\sigma^{n+1}\right) + S_h^n\left(\mathbf{F}_b^{n+1}\right) \quad \text{in } \Omega_h, \tag{7}$$

$$S_h^n\left(\mathbf{F}_\sigma^{n+1}\right) = \sum_{k=1}^{M} D_s\left(\sigma^{n+1}\boldsymbol{\tau}^n\right)_k \delta_h\left(\mathbf{x} - \mathbf{X}_k^n\right)\Delta s, \tag{8}$$

$$S_h^n\left(\mathbf{F}_b^{n+1}\right) = -c_b \sum_{k=1}^{M} D_s^+ D_s^- D_s^+ D_s^- \mathbf{X}_k^{n+1} \delta_h\left(\mathbf{x} - \mathbf{X}_k^n\right)\Delta s, \tag{9}$$

$$\nabla_h \cdot \mathbf{u}^{n+1} = 0 \quad \text{in } \Omega_h, \tag{10}$$

$$\nabla_{s_h} \cdot \mathbf{U}_k^{n+1} = D_s^- \mathbf{U}_k^{n+1} \cdot \boldsymbol{\tau}_{k-1/2}^n = 0 \quad \forall k, \tag{11}$$

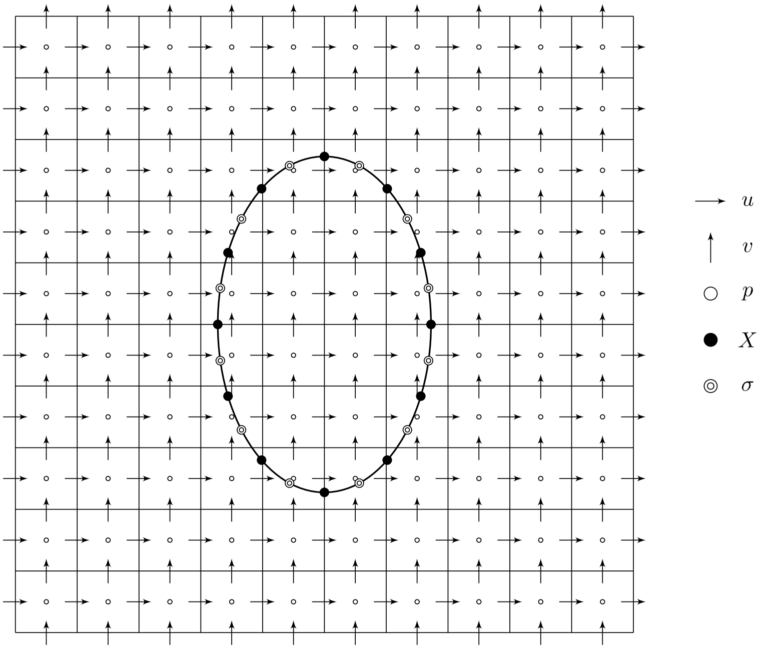


Fig. 1. Staggered finite difference grid in a computational rectangular domain. The pressure p is defined at the cell-center while the velocity u and v are defined at cell-edges. The Lagrangian markers $\mathbf{X}$ represent the IB and σ are defined at the half-integer points between the Lagrangian markers.

$$\frac{\mathbf{X}_k^{n+1}-\mathbf{X}_k^n}{\Delta t}=\mathbf{U}_k^{n+1}=\sum_{\mathbf{x}}\mathbf{u}(\mathbf{x})^{n+1}\delta_h\left(\mathbf{x}-\mathbf{X}_k^n\right)h^2 \quad \forall k. \tag{12}$$

The spatial operators ∇_h and Δ_h are the second-order central difference approximations to the gradient and Laplacian on the staggered finite difference grid, respectively. $\nabla_h\cdot$ and $\nabla_{s_h}\cdot$ are the divergence and surface divergence operators, respectively. We apply the forward, backward, and central difference schemes to approximate the partial derivative of an arbitrary function $\Phi(s)$ defined on the IB,

$$D_s^+\Phi(s)=\frac{\Phi(s+\Delta s)-\Phi(s)}{\Delta s}, \quad D_s^-\Phi(s)=\frac{\Phi(s)-\Phi(s-\Delta s)}{\Delta s}, \tag{13}$$

$$D_s\Phi(s)=\frac{\Phi(s+\Delta s/2)-\Phi(s-\Delta s/2)}{\Delta s}. \tag{14}$$

Subsequently, the unit tangent $\boldsymbol{\tau}=D_s\mathbf{X}$ is approximated discretely at the half-integer points. The operator δ_h is the discrete delta function expressed by a tensor product of a kernel $\phi(r)$,

$$\delta_h(\mathbf{x})=\frac{1}{h^2}\phi\left(\frac{x}{h}\right)\phi\left(\frac{y}{h}\right), \tag{15}$$

where the kernel $\phi(r)$ is the 6-point supported C^3 function developed in [32]. In practice, the distribution of discrete points s_k with uniform arc-length may not be sustainable. Consequently, we define the non-uniform fourth derivative operator based on Taylor series expansion [28],

$$\frac{\partial^4\Phi_k}{\partial s^4}\approx\sum_{i=-2}^{2}c_{k+i}\Phi_{k+i}. \tag{16}$$

It is important to mention that above numerical scheme (7)-(12) is a linearly semi-implicit scheme despite the fact that the original governing equations (1)-(6) are nonlinear in $\mathbf{X}$. The spreading operator S_h^n used in the scheme is the linearization of S_h^{n+1} by lagging one discrete time-step Δt for the Lagrangian markers from $t=(n+1)\Delta t$ to $t=n\Delta t$. Consistently, the discrete inextensibility constraint in the scheme also applies one time-step lagging of the interface tangent for linearization. Therefore, by lagging the spreading and interpolation operators in the IB method yields a linear system of equations instead of the nonlinear system. Nevertheless, one can still rigorously prove that the present scheme is unconditionally energy stable [28]. For the Navier-Stokes equations, the nonlinear convective term can be treated explicitly during the time evolution with a moderate CFL condition. In preference, the Navier-Stokes equations can be split into a convection part and unsteady Stokes part in which the former part can be solved by the alternating direction implicit (ADI) method to maintain the unconditionally numerical stability [33]. However, it is still practically challenging to directly solve the system (7)-(12) numerically. Applying sparse iterative solver such as Krylov subspace method requires a large number of iterations at each time-step, and it may fail to converge without proper preconditioners. On the other hand, applying the sparse direct solver requires the explicit matrix entries at each time-step, and a large amount of fill-ins will cost a lot of computer memory. For the present study, we focus on the development of an efficient fractional step method to solve the system without jeopardizing the numerical accuracy and stability.

4. Immersed boundary projection method

The main objective of the present study is to develop an IBPM for the aforementioned discretized system of equations (7)-(12). The present method follows the algebraic approach used by Taira and Colonius [30], based on the fractional step analysis of Perot [29], for simulating incompressible flows over moving solid objects. The idea is to rewrite the whole linear system into a block-LU decomposition and then make appropriate approximations to obtain several fractional steps that can be efficiently solved. To facilitate the numerical algorithm development, we consider first the inextensible interface problem in the absence of the bending force. Then the extension to model the vesicle dynamics by incorporating the bending force will be addressed subsequently.

4.1. Inextensible interface

In the absence of the bending force term Equation (9) in Equation (7), the rest of equations can be written in a matrix operator form as

$$H\mathbf{u}^{n+1} + Gp^{n+1} + S\sigma^{n+1} = \mathbf{r} + bc_1, \tag{17}$$

$$D\mathbf{u}^{n+1} = -G^{\intercal}\mathbf{u}^{n+1} = bc_2, \tag{18}$$

$$-S^{\intercal}\mathbf{u}^{n+1} = 0, \tag{19}$$

$$E\mathbf{u}^{n+1} + \mathbf{X}^{n+1} = \mathbf{X}^{n}, \tag{20}$$

where $H = \frac{1}{\Delta t}I - \nu L$ is the discrete modified Helmholtz operator, L is the discrete Laplacian operator, G is the discrete gradient operator, $D \equiv -G^{\intercal}$ is the discrete divergence operator, and $\mathbf{r}$ is the explicit source term of the momentum equation. The term bc_1 is the boundary condition for the momentum equation resulting from the Laplacian operator, whereas bc_2 is the boundary condition resulting from the divergence operator. E is the discrete interpolation operator that interpolates velocity components defined at the cell-edges onto the Lagrangian markers. S is the discrete spreading operator that regularizes the singular force due to the elastic tension σ at time-level $n+1$ over the adjacent Eulerian grid, i.e.,

$$S\sigma^{n+1} = \sum_{k=1}^{M} D_s\left(\sigma^{n+1}\boldsymbol{\tau}^{n}\right)_k \delta_h\left(\mathbf{x}-\mathbf{X}_k^n\right)\Delta s = \sum_{k=1}^{M} \sigma_{k-1/2}^{n+1}\boldsymbol{\tau}_{k-1/2}^{n}\left[\delta_h\left(\mathbf{x}-\mathbf{X}_{k-1}^n\right) - \delta_h\left(\mathbf{x}-\mathbf{X}_k^n\right)\right]. \tag{21}$$

Note that the discrete surface divergence of velocity can be cast as $-S^{\intercal}\mathbf{u}^{n+1}$ [17], and it is regarded as a projection operator acting on the Eulerian velocity field at time-level $n+1$. Henceforth, Equations (17) to (19) produce a skew-symmetric algebraic system of equations as

$$\begin{bmatrix} H & G & S \\ -G^{\intercal} & 0 & 0 \\ -S^{\intercal} & 0 & 0 \end{bmatrix}\begin{bmatrix} \mathbf{u}^{n+1} \\ p^{n+1} \\ \sigma^{n+1} \end{bmatrix} = \begin{bmatrix} \mathbf{r}+bc_1 \\ bc_2 \\ 0 \end{bmatrix}. \tag{22}$$

To begin with, some augmented operators are first introduced; $\tilde{G} \equiv [G \quad S]$, $\tilde{G}^{\intercal} \equiv [G^{\intercal} \quad S^{\intercal}]^{\intercal}$, $\lambda^{n+1} \equiv \left[p^{n+1} \quad \sigma^{n+1}\right]^{\intercal}$, and $\tilde{bc}_2 \equiv [bc_2 \quad 0]^{\intercal}$. Thus, Equation (22) can be written as

$$\begin{bmatrix} H & \tilde{G} \\ -\tilde{G}^{\intercal} & 0 \end{bmatrix}\begin{bmatrix} \mathbf{u}^{n+1} \\ \lambda^{n+1} \end{bmatrix} = \begin{bmatrix} \mathbf{r}+bc_1 \\ \tilde{bc}_2 \end{bmatrix}. \tag{23}$$

It is realized that λ acts as a set of Lagrange multipliers to enforce the fluid incompressibility and the interface inextensibility. Subsequently, the block LU decomposition of Equation (23) reads

$$\begin{bmatrix} H & 0 \\ -\tilde{G}^{\intercal} & \tilde{G}^{\intercal}H^{-1}\tilde{G} \end{bmatrix}\begin{bmatrix} I & H^{-1}\tilde{G} \\ 0 & I \end{bmatrix}\begin{bmatrix} \mathbf{u}^{n+1} \\ \lambda^{n+1} \end{bmatrix} = \begin{bmatrix} \mathbf{r}+bc_1 \\ \tilde{bc}_2 \end{bmatrix} \tag{24}$$

which is in a form similar to Uzawa method [34]. As in [30], it is practically feasible to approximate H^{-1} by Q^N,

$$\begin{bmatrix} H & 0 \\ -\tilde{G}^{\intercal} & \tilde{G}^{\intercal}Q^N\tilde{G} \end{bmatrix}\begin{bmatrix} I & Q^N\tilde{G} \\ 0 & I \end{bmatrix}\begin{bmatrix} \mathbf{u}^{n+1} \\ \lambda^{n+1} \end{bmatrix} = \begin{bmatrix} \mathbf{r}+bc_1 \\ \tilde{bc}_2 \end{bmatrix} + \begin{bmatrix} \left(HQ^N - I\right)\tilde{G}\lambda^{n+1} \\ 0 \end{bmatrix} \tag{25}$$

where Q^N is the N-th order Neumann series expansion of H^{-1},

$$H^{-1} \approxeq Q^N = \Delta t \left(I + \Delta t \nu L + (\Delta t \nu L)^2 + \cdots + (\Delta t \nu L)^{N-1} \right) = \Delta t \sum_{k=0}^{N-1} (\Delta t \nu L)^k . \tag{26}$$

Let us define

$$\begin{bmatrix} \mathbf{u}^* \\ \lambda^{n+1} \end{bmatrix} = \begin{bmatrix} I & Q^N \tilde{G} \\ 0 & I \end{bmatrix} \begin{bmatrix} \mathbf{u}^{n+1} \\ \lambda^{n+1}, \end{bmatrix}$$

where $\mathbf{u}^* = \mathbf{u}^{n+1} + Q^N \tilde{G} \lambda^{n+1}$ is the intermediate velocity. Then, Equation (25) can be written in the following form,

$$\begin{aligned} H\mathbf{u}^* &= \mathbf{r} + bc_1 + \left(HQ^N - I \right) \tilde{G} \lambda^{n+1}, \\ \tilde{G}^\intercal Q^N \tilde{G} \lambda^{n+1} &= \tilde{G}^\intercal \mathbf{u}^* + \tilde{bc}_2, \\ \mathbf{u}^{n+1} &= \mathbf{u}^* - Q^N \tilde{G} \lambda^{n+1}. \end{aligned}$$

Dropping the last term $\left(HQ^N - I \right) \tilde{G} \lambda^{n+1}$ at the righthand side of the momentum equation results in a fractional step procedure in which $\mathbf{u}^*$ and λ^{n+1} are decoupled. Hence, the term $\left(HQ^N - I \right) \tilde{G} \lambda^{n+1}$ is the corresponding splitting error occurring in the momentum equation while there is no splitting error for the constraint equation as we can easily obtain $-\tilde{G}^\intercal \mathbf{u}^{n+1} = \tilde{bc}_2$. By setting $Q^N = I\Delta t (N = 1)$, the above scheme becomes

$$\begin{aligned} H\mathbf{u}^* &= \mathbf{r} + bc_1 + \underbrace{(H\Delta t - I) \tilde{G} \lambda^{n+1}}_{\text{first-order splitting error}} + O\left(\Delta t^2 \right), \\ \tilde{G}^\intercal \tilde{G} \lambda^{n+1} &= \frac{1}{\Delta t} \left(\tilde{G}^\intercal \mathbf{u}^* + \tilde{bc}_2 \right), \\ \mathbf{u}^{n+1} &= \mathbf{u}^* - \Delta t \tilde{G} \lambda^{n+1}. \end{aligned}$$

Thus, the leading first-order splitting error has the form $(H\Delta t - I)\tilde{G}\lambda^{n+1} = -\nu \Delta t L \tilde{G} \lambda^{n+1}$. Besides, $\tilde{G}^\intercal \tilde{G}$ can be formed to reduce the computational effort significantly. Meanwhile, if the first-order splitting error is neglected, then the classical projection method proposed in [35,36] is recovered.

By augmenting an explicit term at the righthand side of the momentum equation, the splitting error can be reduced to second-order [29], i.e.,

$$H\mathbf{u}^* = \mathbf{r} + bc_1 + \underbrace{(H\Delta t - I) \tilde{G} \lambda^n}_{\text{augmented explicit term}} + \underbrace{(H\Delta t - I) \tilde{G} \left(\lambda^{n+1} - \lambda^n \right)}_{\text{second-order splitting error}} + O\left(\Delta t^2 \right). \tag{27}$$

Therefore, we obtain a numerical algorithm with second-order splitting error as follows,

$$\begin{aligned} H\mathbf{u}^* &= \mathbf{r} + (H\Delta t - I) \tilde{G} \lambda^n + bc_1, \\ \tilde{G}^\intercal \tilde{G} \lambda^{n+1} &= \frac{1}{\Delta t} \left(\tilde{G}^\intercal \mathbf{u}^* + \tilde{bc}_2 \right), \\ \mathbf{u}^{n+1} &= \mathbf{u}^* - \Delta t \tilde{G} \lambda^{n+1}. \end{aligned}$$

Notice that, solving the equation of $\mathbf{u}^*$ requires to compute the explicit term $(H\Delta t I - I)\tilde{G}\lambda^n$ at the righthand side. To avoid such explicit term computation, we introduce another intermediate velocity $\mathbf{u}^{**} = \mathbf{u}^* - \Delta t \tilde{G} \lambda^n$ and the incremental variable $\delta\lambda = \lambda^{n+1} - \lambda^n$, so the algorithm can be further simplified to

$$\begin{aligned} H\mathbf{u}^{**} &= \mathbf{r} - \tilde{G} \lambda^n + bc_1, \\ \tilde{G}^\intercal \tilde{G} \delta\lambda &= \frac{1}{\Delta t} \left(\tilde{G}^\intercal \mathbf{u}^{**} + \tilde{bc}_2 \right), \\ \mathbf{u}^{n+1} &= \mathbf{u}^{**} - \Delta t \tilde{G} \delta\lambda. \end{aligned}$$

The first equation involves solving the modified Helmholtz equation about $\mathbf{u}^{**}$ which can be done very efficiently using Fast Fourier Transform (FFT)-based solver. Once $\mathbf{u}^{**}$ is obtained, the remaining question is how to solve the equation of the incremental variable $\delta\lambda = [\delta p \quad \delta\sigma]^\intercal$, i.e.,

$$\begin{bmatrix} G^\intercal G & G^\intercal S \\ S^\intercal G & S^\intercal S \end{bmatrix} \begin{bmatrix} \delta p \\ \delta \sigma \end{bmatrix} = \frac{1}{\Delta t} \begin{bmatrix} G^\intercal \mathbf{u}^{**} + bc_2 \\ S^\intercal \mathbf{u}^{**} \end{bmatrix}. \tag{28}$$

By applying the block LU decomposition again, the above equation can be written as

$$\begin{bmatrix} -L & 0 \\ S^\intercal G & S^\intercal S + S^\intercal G L^{-1} G^\intercal S \end{bmatrix} \begin{bmatrix} I & -L^{-1} G^\intercal S \\ 0 & I \end{bmatrix} \begin{bmatrix} \delta p \\ \delta\sigma \end{bmatrix} = \frac{1}{\Delta t} \begin{bmatrix} G^\intercal \mathbf{u}^{**} + bc_2 \\ S^\intercal \mathbf{u}^{**} \end{bmatrix}, \tag{29}$$

where $L \equiv -G^\intercal G$ is the discrete Laplacian operator. Let us introduce $\delta p^* = \delta p - L^{-1} G^\intercal S \delta\sigma$, the incremental variables δp and $\delta\sigma$ can be solved in a decoupled way as follows,

$$\begin{aligned} L\delta p^* &= -\frac{1}{\Delta t}\left(G^\intercal \mathbf{u}^{**} + bc_2\right), \\ \left(S^\intercal S + S^\intercal G L^{-1} G^\intercal S\right)\delta\sigma &= \frac{1}{\Delta t} S^\intercal \mathbf{u}^{**} - S^\intercal G \delta p^*, \\ \delta p &= \delta p^* + L^{-1} G^\intercal S \delta\sigma. \end{aligned}$$

Thus, δp and $\delta\sigma$ can be computed sequentially without compromising the accuracy since the performed block LU decomposition is exact. A key feature is that both the discrete fluid incompressibility and discrete interface inextensibility are satisfied simultaneously as we will see later.

In summary, the time advancement (from time level $n\Delta t$ to $(n+1)\Delta t$) of IBPM for the inextensible interface problem can be outlined in the following steps.

$$\begin{array}{lcr} (1) & H\mathbf{u}^{**} = \mathbf{r} - Gp^n - S\sigma^n + bc_1, & \text{(FFT-based solver)} \\ (2) & L\delta p^* = -\dfrac{1}{\Delta t}\left(G^\intercal \mathbf{u}^{**} + bc_2\right), & \text{(FFT-based solver)} \\ (3) & \left(S^\intercal S + S^\intercal G L^{-1} G^\intercal S\right)\delta\sigma = \dfrac{1}{\Delta t} S^\intercal \mathbf{u}^{**} - S^\intercal G\delta p^*, & \text{(GMRES + FFT-based solver)} \\ (4) & \delta p = \delta p^* + L^{-1} G^\intercal S\delta\sigma, & \text{(FFT-based solver)} \\ (5) & \mathbf{u}^{n+1} = \mathbf{u}^{**} - \Delta t\left(G\delta p + S\delta\sigma\right), & \\ (6) & \mathbf{X}^{n+1} = \mathbf{X}^n - E\mathbf{u}^{n+1}. & \end{array}$$

The pressure and the elastic tension are then updated by $p^{n+1} = p^n + \delta p$ and $\sigma^{n+1} = \sigma^n + \delta\sigma$, respectively. We now analyze the computational efficiency for each step in the above algorithm. In the present implementation, the discrete Laplacian operator L and discrete modified Helmholtz operator H are diagonalizable in Fourier space, we therefore utilize the FFT-based modified Helmholtz and Poisson solvers to compute the intermediate velocity $\mathbf{u}^{**}$ and the intermediate pressure increment δp^* in Steps (1) and (2), respectively. Similarly, solving the pressure increment in Step (4), we do not compute the inverse of Laplacian explicitly but simply applying FFT-based Poisson solver once. To compute the elastic tension increment $\delta\sigma$ in Step (3), we apply the GMRES iterative solver [38]. Each GMRES iteration requires to apply one FFT-based Poisson solver and sparse matrix-vector products. Notice that, the size of the sparse matrix-vector product corresponding to the number of Lagrangian markers is one dimension lower than the Eulerian grid number. In practice, the average number of GMRES iterations required to converge to a prescribed error tolerance of h^2 is proportional to the grid resolution. To accelerate the iterative convergence in Step (3), we utilize the incomplete LU factorization [39] of the approximated Schur complement $\hat{C} \equiv S^\intercal S + S^\intercal G(\text{diag}(L))^{-1} G^\intercal S$ as a simple preconditioner for the GMRES iteration. The cost of generating such preconditioner is insignificant as compared to the cost of matrix-vector multiplications. The performance of the preconditioned GMRES solver will be evaluated in Section 5. Finally, Steps (5) and (6) are simply variable substitutions. The overall computational cost can be evaluated in terms of the number of FFT-based fast solvers applied.

We conclude this subsection by proving that the discrete fluid incompressibility and discrete interface inextensibility are satisfied simultaneously at every time-step, i.e., $-G^\intercal \mathbf{u}^{n+1} = bc_2$ and $-S^\intercal \mathbf{u}^{n+1} = 0$. Applying the discrete divergence operator $-G^\intercal$ on $\mathbf{u}^{n+1}$ in Step (5), we have

$$\begin{aligned} -G^\intercal \mathbf{u}^{n+1} &= -G^\intercal \mathbf{u}^{**} + \Delta t\left(G^\intercal G\delta p + G^\intercal S\delta\sigma\right) \\ &= \Delta t L\delta p^* + bc_2 + \Delta t\left(G^\intercal G\delta p + G^\intercal S\delta\sigma\right) \quad \text{by Step (2)} \\ &= -\Delta t\left(G^\intercal G\delta p + G^\intercal S\delta\sigma\right) + bc_2 + \Delta t\left(G^\intercal G\delta p + G^\intercal S\delta\sigma\right) \quad \text{by } L \equiv -G^\intercal G \text{ and Step (4)} \\ &= bc_2. \end{aligned}$$

Hence, the discrete fluid incompressibility is satisfied. Similarly, by applying the discrete surface divergence operator $-S^\intercal$ on $\mathbf{u}^{n+1}$ in Step (5) yields

$$\begin{aligned} -S^\intercal \mathbf{u}^{n+1} &= -S^\intercal \mathbf{u}^{**} + \Delta t\left(S^\intercal G\delta p + S^\intercal S\delta\sigma\right) \\ &= -\Delta t\left(S^\intercal G\delta p^* + S^\intercal G L^{-1} G S\delta\sigma + S^\intercal S\delta\sigma\right) + \Delta t\left(S^\intercal G\delta p + S^\intercal S\delta\sigma\right) \quad \text{by Step (3)} \end{aligned}$$

$$
\begin{aligned}
&= -\Delta t\left(S^\top G\delta p + S^\top S\delta\sigma\right) + \Delta t\left(S^\top G\delta p + S^\top S\delta\sigma\right) \quad \text{by Step (4)}\\
&= 0.
\end{aligned}
$$

Hence, the discrete interface inextensibility is satisfied.

4.2. Inextensible interface with bending force

We have developed an algorithm to solve the inextensible interface problem in previous subsection. In this subsection, we extend the development of the algorithm to compute the inextensible interface with bending force for simulating the vesicle dynamics. As before, the resultant matrix equations of (7)-(12) can be written in the following operator form

$$H\mathbf{u}^{n+1} + Gp^{n+1} + S\sigma^{n+1} + B\mathbf{X}^{n+1} = \mathbf{r} + bc_1, \tag{30}$$

$$D\mathbf{u}^{n+1} = -G^\top\mathbf{u}^{n+1} = bc_2, \tag{31}$$

$$-S^\top\mathbf{u}^{n+1} = 0, \tag{32}$$

$$E\mathbf{u}^{n+1} + \mathbf{X}^{n+1} = \mathbf{X}^n, \tag{33}$$

where B is the discrete bending force operator. By substituting Equation (33) into Equation (30), we have

$$\tilde{H}\mathbf{u}^{n+1} + Gp^{n+1} + S\sigma^{n+1} = \mathbf{r} + bc_1 - B\mathbf{X}^n, \tag{34}$$

where $\tilde{H} \equiv H - BE$ is the modified Helmholtz operator H subtracting a low rank perturbation BE due to the bending force. Clearly, the resultant matrix form is similar to Equation (23) except that the operator H is replaced by the operator $\tilde{H}$, and the righthand side is modified, i.e.,

$$\begin{bmatrix} \tilde{H} & \tilde{G} \\ -\tilde{G}^\top & 0 \end{bmatrix}\begin{bmatrix} \mathbf{u}^{n+1} \\ \lambda^{n+1} \end{bmatrix} = \begin{bmatrix} \mathbf{r} + bc_1 - B\mathbf{X}^n \\ \tilde{bc}_2 \end{bmatrix}, \tag{35}$$

where the augmented matrix $\tilde{G} = [G \quad S]$ and augmented variable $\lambda^{n+1} = [p^{n+1} \quad \sigma^{n+1}]^\top$ are defined as before. Again, by introducing the incremental variable $\delta\lambda = [\delta p \quad \delta\sigma]^\top$ and the intermediate velocity $\mathbf{u}^{**} = \mathbf{u}^{n+1} + \Delta t\,(G\delta p + S\delta\sigma)$, and applying the preceding splitting technique in the previous subsection, the resultant matrix equation (35) can be written into following steps,

$$
\begin{aligned}
\tilde{H}\mathbf{u}^{**} &= \mathbf{r} - Gp^n - S\sigma^n - B\mathbf{X}^n + bc_1,\\
L\delta p^* &= -\frac{1}{\Delta t}\left(G^\top\mathbf{u}^{**} + bc_2\right),\\
\left(S^\top S + S^\top G L^{-1} G^\top S\right)\delta\sigma &= \frac{1}{\Delta t}S^\top\mathbf{u}^{**} - S^\top G\delta p^*,\\
\delta p &= \delta p^* + L^{-1}G^\top S\delta\sigma,\\
\mathbf{u}^{n+1} &= \mathbf{u}^{**} - \Delta t\,(G\delta p + S\delta\sigma).
\end{aligned}
$$

Notice that, in this case, we can not apply the FFT-based solver directly to solve the momentum equation of $\mathbf{u}^{**}$, since the operator $\tilde{H} = H - BE$ is perturbed by a low rank matrix. However, the inversion of $\tilde{H}$ can be expressed by the Sherman-Morrison-Woodbury formula [37],

$$\tilde{H}^{-1} \equiv (H - BE)^{-1} \equiv H^{-1} + H^{-1}B(I - EH^{-1}B)^{-1}EH^{-1}. \tag{36}$$

The primary advantage of applying Equation (36) is that the formula dramatically simplifies the computation of $\tilde{H}^{-1}$ since it is expressed in terms of H^{-1} for which the FFT-based solver can be applied. By introducing an intermediate variable $\mathbf{u}^{***} = H^{-1}(\mathbf{r} - Gp^n - S\sigma^n - B\mathbf{X}^n + bc_1)$, the IBPM for the inextensible interface with bending rigidity to simulate the vesicle dynamics can be summarized in the following segregated steps,

$$
\begin{aligned}
&(1) & H\mathbf{u}^{***} &= \mathbf{r} - Gp^n - S\sigma^n - B\mathbf{X}^n + bc_1, & &\text{(FFT-based solver)}\\
&(2) & \mathbf{u}^{**} &= \mathbf{u}^{***} + H^{-1}B(I - EH^{-1}B)^{-1}E\mathbf{u}^{***}, & &\text{(GMRES + FFT-based solver)}\\
&(3) & L\delta p^* &= -\frac{1}{\Delta t}\left(G^\top\mathbf{u}^{**} + bc_2\right), & &\text{(FFT-based solver)}\\
&(4) & \left(S^\top S + S^\top G L^{-1} G^\top S\right)\delta\sigma &= \frac{1}{\Delta t}S^\top\mathbf{u}^{**} - S^\top G\delta p^*, & &\text{(GMRES + FFT-based solver)}\\
&(5) & \delta p &= \delta p^* + L^{-1}G^\top S\delta\sigma, & &\text{(FFT-based solver)}\\
&(6) & \mathbf{u}^{n+1} &= \mathbf{u}^{**} - \Delta t\,(G\delta p + S\delta\sigma), & &\\
&(7) & \mathbf{X}^{n+1} &= \mathbf{X}^n - E\mathbf{u}^{n+1}. & &
\end{aligned}
$$

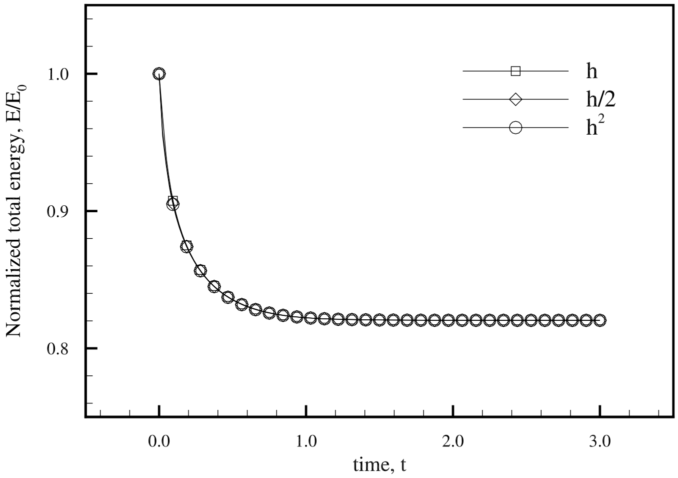


Fig. 2. Normalized total energy E/E_0 dissipation profiles from time $t = 0$ to $t = 3.0$ using time-step sizes $\Delta t = h$, $h/2$, and h^2 with a fixed Eulerian grid resolution of $h = 1/256$.

The pressure and the elastic tension are then updated by $p^{n+1} = p^n + \delta p$ and $\sigma^{n+1} = \sigma^n + \delta\sigma$, respectively. One should note that, Steps (3)-(7) are same as the ones in the previous subsection. The intermediate velocity variable $\mathbf{u}^{***}$ in Step (1) is efficiently solved by the FFT-based modified Helmholtz solver. Computing $\mathbf{u}^{**}$ in Step (2) can be interpreted as an "implicit bending force update" of $\mathbf{u}^{***}$, and it requires a few numbers of GMRES iterations. Again, the GMRES iteration does not need to form the matrix explicitly, but only requires applying the FFT-based modified Helmholtz solver and sparse matrix-vector products. In practice, Step (2) requires only an average number of 2-3 GMRES iterations to converge to the prescribed error tolerance of order h^2. So again, the entire algorithm is matrix-free and takes a linearithmic complexity as the GMRES solver only requires matrix-vector products, and the inversion of Laplacian and modified Helmholtz operators are computed directly using FFT-based fast solvers.

One of the appealing features of the present method is its modularity. In particular, the freedom to use a conventional Navier-Stokes or Stokes solver with minor modifications in Steps (1), (3), and (6). Such modularity is achieved by solving the momentum equation in Step (1) with explicit IB elastic tension force $S\sigma^n$ and bending force $B\mathbf{X}^n$. Step (3) involves solving the Poisson equation by replacing δp by δp^* in a conventional fluid solver. Lastly, updating the velocity components in Step (6) involves an incremental pressure gradient $G\delta p$, and an incremental elastic tension $S\delta\sigma$ additionally.

5. Numerical results

5.1. Discrete energy stability

We conduct an energy stability verification for the present IBPM developed in Section 4. A vesicle is initialized as an ellipse shape $\mathbf{X}(\theta) = (0.2\cos\theta, 0.5\sin\theta)$ using uniform arc-length parametrization [28], and is immersed in a quiescent fluid domain $\Omega = [-1, 1] \times [-1, 1]$. For other parameters, $\nu = 1$, and $c_b = 0.01$ are applied. The simulations are time-marching from time $t = 0$ to $t = 3.0$, using time-step sizes $\Delta t = h$, $h/2$, and h^2, with a fixed Eulerian grid resolution of $h = 1/256$. The Lagrangian markers size is chosen as $\Delta s = O(h)$.

The total energy of the system E is defined as the sum of kinetic energy E_K and bending energy E_B, where

$$E_K = \frac{1}{2}\|\mathbf{u}\|^2_{\Omega_h} = \frac{1}{2}\langle \mathbf{u}, \mathbf{u}\rangle_{\Omega_h}, \quad E_B = \frac{c_b}{2}\|D_s^+ D_s^- \mathbf{X}\|^2_{\Gamma_h} = \frac{c_b}{2}\langle D_s^+ D_s^- \mathbf{X}, D_s^+ D_s^- \mathbf{X}\rangle_{\Gamma_h}, \tag{37}$$

with the associated discrete inner products defined as

$$\langle \mathbf{u}, \mathbf{v}\rangle_{\Omega_h} = \sum_{\mathbf{x}} (\mathbf{u}\cdot\mathbf{v})\, h^2, \quad \langle \Phi, \Psi\rangle_{\Omega_h} = \sum_{k=1}^{M} \Phi_k \Psi_k \Delta s. \tag{38}$$

Fig. 2 shows the transient dissipation profiles of the total energy E normalized by the initial total energy E_0 at $t = 0$. The discrete energy profiles are monotonically decreasing, which substantiates that the discrete energy of the system is bounded and the present numerical algorithm does not affect the unconditionally energy stability of the underlying scheme. We examine the convergence rate of the solution variables quantitatively in the next subsection.

Table 1
Grid convergence study 1: grid refinement analysis on the Eulerian variables u, v, and p.

h	$\|u_h - u_{\text{ref}}\|_2$	Rate	$\|v_h - v_{\text{ref}}\|_2$	Rate	$\|p_h - p_{\text{ref}}\|_2$	Rate
1/32	8.607×10^{-3}	-	6.878×10^{-3}	-	1.079×10^{-2}	-
1/64	3.649×10^{-3}	1.24	3.649×10^{-3}	0.91	6.405×10^{-3}	0.75
1/128	1.698×10^{-3}	1.10	1.489×10^{-3}	1.29	3.808×10^{-3}	0.75
1/256	6.923×10^{-4}	1.29	6.296×10^{-4}	1.24	1.928×10^{-3}	0.98

Table 2
Grid convergence study 1: grid refinement analysis on the Lagrangian variables X, Y, and σ.

h	$\|X_h - X_{\text{ref}}\|_2$	Rate	$\|Y_h - Y_{\text{ref}}\|_2$	Rate	$\|\sigma_h - \sigma_{\text{ref}}\|_2$	Rate
1/32	2.696×10^{-3}	-	1.235×10^{-3}	-	9.463×10^{-2}	-
1/64	6.229×10^{-4}	2.11	3.045×10^{-4}	2.02	4.700×10^{-2}	1.01
1/128	1.565×10^{-4}	1.99	8.664×10^{-5}	1.81	2.336×10^{-2}	1.01
1/256	3.603×10^{-5}	2.12	1.899×10^{-5}	2.19	1.162×10^{-2}	1.01

Table 3
Grid convergence study 1: grid refinement analysis on the enclosed area A and total arc-length L.

h	$\|A_h - A_0\|/A_0$	Rate	$\|L_h - L_0\|/L_0$	Rate
1/32	2.075×10^{-4}	-	1.132×10^{-4}	-
1/64	1.055×10^{-4}	0.98	1.061×10^{-4}	0.09
1/128	5.363×10^{-5}	0.98	6.964×10^{-5}	0.61
1/256	2.884×10^{-5}	0.89	3.989×10^{-5}	0.80

5.2. Grid convergence study 1

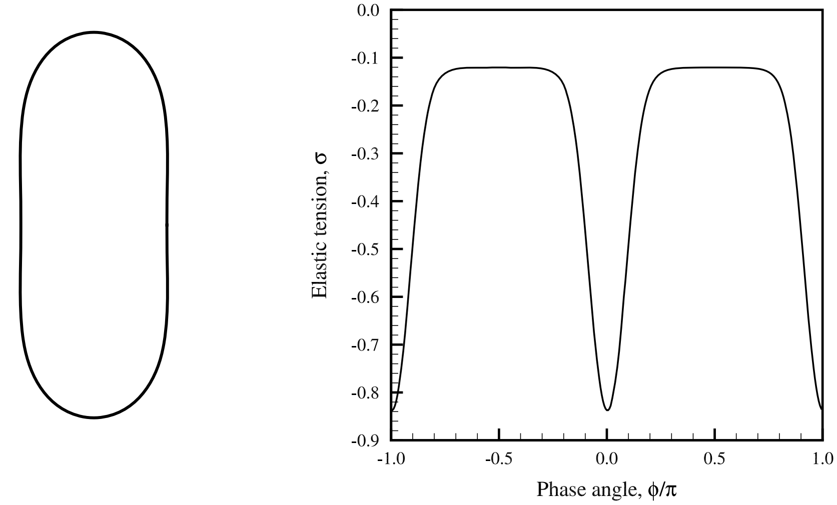
We conduct a grid convergence study for a vesicle in a quiescent fluid. An initial vesicle shape is defined by an ellipse $\mathbf{X}(\theta) = (0.2\cos\theta, 0.5\sin\theta)$, and placed in a quiescent fluid domain $\Omega = [-1, 1] \times [-1, 1]$. For a linear refinement, a time-step of $\Delta t = h/2$ is employed in conjunction with a series of grid resolution $h = 1/32, 1/64, 1/128, 1/256$. The stopping tolerance of all GMRES method is set to be h^2. All the simulations are time-marching from $t = 0$ to $t = 1.0$. Since an analytical solution is not available, the numerical solution of a fine grid resolution $h = 1/512$ is taken as the reference solution. To quantify the grid convergence rate of the solution variables, the L^2-norm errors are evaluated by taking the difference between the numerical solutions and interpolated reference solutions.

Table 1 shows the grid refinement analysis of the Eulerian variables, i.e., velocity components u and v, and pressure p. The numerical results indicate that the Eulerian variables exhibit a first-order convergence rate in L^2-norm. Table 2 tabulates the grid refinement analysis of the Lagrangian variables, i.e., position vectors X and Y, and elastic tension σ. A better than first-order convergence rate is observed for the position vectors, while the elastic tension converges at a first-order rate in L^2-norm. The capability of the present method in preserving the enclosed area A and total arc-length L is investigated and summarized in Table 3. A_0 and L_0 are the initial enclosed area and initial total arc-length of the vesicle, respectively. As the grid is linearly refined, these relative errors are gradually decreasing at a first-order rate approximately. When the grid resolution of $h = 1/256$ is used, the relative errors of the area conservation and total arc-length conservation are about 0.003% and 0.004%, respectively. Hence, the present method preserves the enclosed area and total arc-length excellently without any ad-hoc modification. Fig. 3 depicts the vesicle shape and its elastic tension profile in terms of phase angle ϕ/π at $t = 1.0$ using the grid resolution of $h = 1/256$. Notably, no numerical wiggle is detected in the elastic tension profile. In Section 5.4, we further explore the deformation of the vesicle to its equilibrium shape systematically.

In order to investigate the performance of the preconditioned GMRES to solve $\delta\sigma$, we repeat the simulation by comparing with un-preconditioned GMRES using the same stopping tolerance h^2 and preconditioned GMRES with a fixed stopping tolerance of 10^{-12}. Table 4 summarizes the performance of the GMRES in terms of average iteration counts for different grid resolutions. For all the grid resolutions, the corresponding average numbers of GMRES iteration tabulated for the case with and without preconditioner, and the prescribed stopping tolerance are indicated in the bracket. It is observed that, without a preconditioner, the average GMRES iteration scales linearly with the linear grid refinement. By applying the simple preconditioner proposed in the present study, the average number of GMRES iteration with tolerance h^2 is maintained at 2-3 iterations for all the grid resolutions, which is essentially grid-independent. Meanwhile, the preconditioner works effectively in conjunction with a fixed stopping tolerance of 10^{-12}, which also outperforms the case without preconditioner.

5.3. Grid convergence study 2

To further justify the numerical accuracy, we conduct a grid convergence study for a vesicle in a shear flow. An initial vesicle shape is expressed by an ellipse $\mathbf{X}(\theta) = (0.2\cos\theta, 0.5\sin\theta)$, and immersed in a fluid domain $\Omega =$



(a) Vesicle shape.

(b) Elastic tension profile.

Fig. 3. Deformation of a vesicle in a quiescent fluid at $t = 1.0$. The corresponding elastic tension σ profile is plotted in terms of phase angle ϕ/π.

Table 4
Grid convergence study 1: performance of the preconditioned GMRES for solving the incremental elastic tension $\delta\sigma$. The prescribed stopping tolerance is indicated in the bracket.

h	Average GMRES iteration		
	Un-preconditioned (h^2)	Preconditioned (h^2)	Preconditioned (10^{-12})
1/32	4.6	2.4	8.6
1/64	10.1	2.7	13.5
1/128	20.6	3.0	16.1
1/256	40.8	3.2	16.7

Table 5
Grid convergence study 2: grid refinement analysis of the Eulerian variables u, v, and p.

h	$\|u_h - u_{\text{ref}}\|_2$	Rate	$\|v_h - v_{\text{ref}}\|_2$	Rate	$\|p_h - p_{\text{ref}}\|_2$	Rate
1/32	1.657×10^{-2}	-	1.287×10^{-2}	-	1.672×10^{-4}	-
1/64	6.636×10^{-3}	1.32	5.215×10^{-3}	1.30	3.671×10^{-5}	2.19
1/128	2.640×10^{-3}	1.33	2.087×10^{-3}	1.32	2.253×10^{-5}	0.70
1/256	8.427×10^{-4}	1.65	6.895×10^{-4}	1.60	1.065×10^{-5}	1.08

Table 6
Grid convergence study 2: grid refinement analysis of the Lagrangian variables X, Y, and σ.

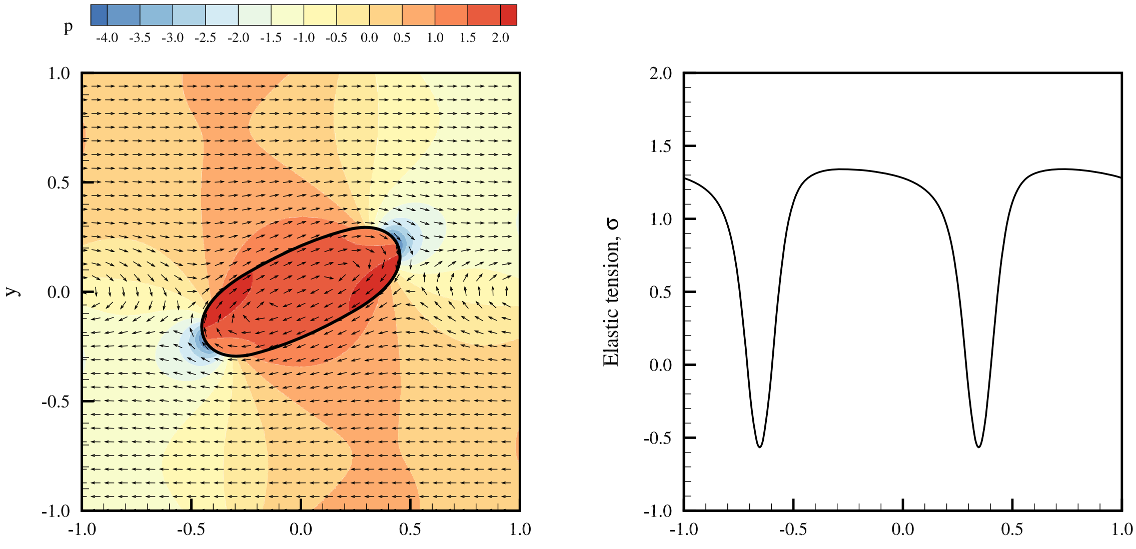
h	$\|X_h - X_{\text{ref}}\|_2$	Rate	$\|Y_h - Y_{\text{ref}}\|_2$	Rate	$\|\sigma_h - \sigma_{\text{ref}}\|_2$	Rate
1/32	1.239×10^{-2}	-	1.279×10^{-2}	-	3.555×10^{-1}	-
1/64	5.046×10^{-3}	1.30	5.358×10^{-3}	1.25	1.421×10^{-1}	1.32
1/128	2.078×10^{-3}	1.28	2.205×10^{-3}	1.28	5.859×10^{-2}	1.28
1/256	6.881×10^{-4}	1.59	7.060×10^{-4}	1.64	2.224×10^{-2}	1.40

$[-1, 1] \times [-1, 1]$ with a shear rate of $\dot{\gamma} = 1$. We employ a time-step of $\Delta t = h/4$, and with a series of grid resolution $h = 1/32, 1/64, 1/128, 1/256$ to conduct the linear refinement analysis. The stopping tolerance of all GMRES is set to h^2. All the simulations are time-marching from time $t = 0$ to $t = 5.0$. In a similar manner, we utilize the numerical solution evaluated from a fine grid resolution of $h = 1/512$ as the reference solution. Then, the L^2-norm errors are computed by evaluating the difference between the computed and interpolated reference solutions.

Table 5 shows the grid refinement analysis of the Eulerian variables, i.e., velocity components u and v, and pressure p, while Table 6 tabulates the grid refinement analysis of the Lagrangian variables, i.e. position vectors X and Y, and elastic tension σ. Similar to the prior grid convergence study, the numerical results demonstrate a first-order convergence rate in L^2-norm for all the solution variables. Since the vesicle exhibits a tank-treading motion in a shear flow, we examine the convergence rate of the evolution frequency $\omega = 2\pi / \int_\Gamma \frac{dl}{\mathbf{u} \cdot \boldsymbol{\tau}}$ in addition to the enclosed area A and total arc-length. L. Table 7 tabulates the grid refinement analysis of these computed quantities. The conservation of the enclosed area and total arc-length of the vesicle are gradually improved at a first-order rate, as the grid is linearly refined, The evolution frequency

Table 7
Grid convergence study 2: grid refinement analysis of the enclosed area A, total arc-length L, and evolution frequency ω.

h	$\lvert A_h - A_0\rvert/A_0$	Rate	$\lvert L_h - L_0\rvert/L_0$	Rate	$\lvert\omega_h - \omega_{\mathrm{ref}}\rvert$	Rate
1/32	4.733×10^{-3}	-	5.362×10^{-3}	-	1.236×10^{-2}	-
1/64	2.695×10^{-3}	0.81	2.723×10^{-3}	0.98	4.706×10^{-3}	1.39
1/128	1.410×10^{-3}	0.93	1.371×10^{-3}	0.99	1.867×10^{-3}	1.33
1/256	7.185×10^{-4}	0.97	6.880×10^{-4}	0.99	5.879×10^{-4}	1.67



(a) Flow topology.

(b) Elastic tension profile.

Fig. 4. A vesicle undergoes tank-treading motion in a shear flow at $t = 5.0$. The associated flow topology is illustrated by the pressure contour and velocity vectors. The resultant elastic tension σ profile is plotted in term of phase angle ϕ/π. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

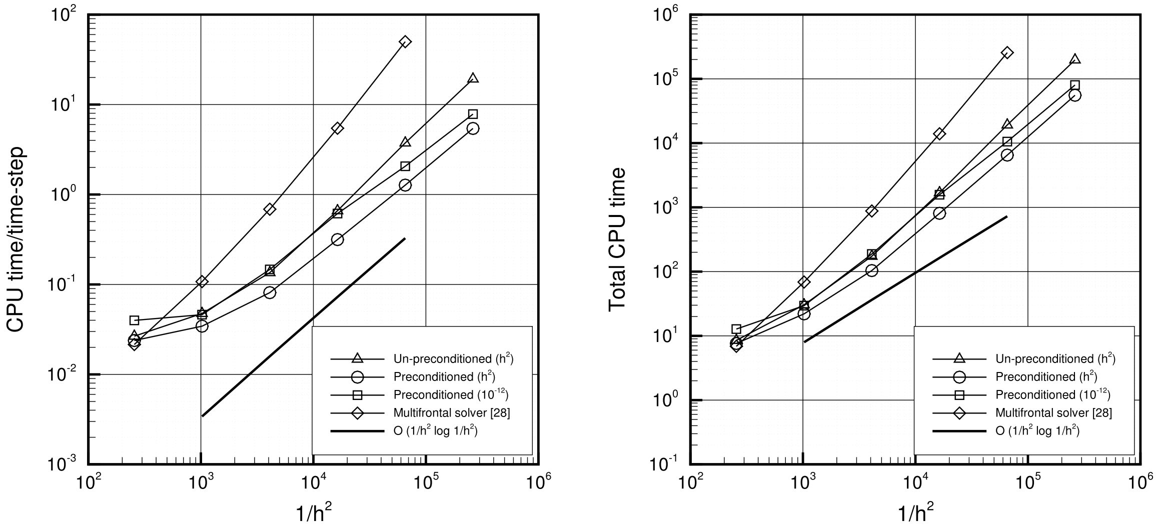
Table 8
Grid convergence study 2: performance of the preconditioned GMRES for solving the incremental elastic tension $\delta\sigma$. The prescribed stopping tolerance is indicated in the bracket.

h	Average GMRES iteration		
	Un-preconditioned (h^2)	Preconditioned (h^2)	Preconditioned (10^{-12})
1/32	5.2	1.6	14.1
1/64	11.4	2.2	15.5
1/128	26.6	3.6	18.8
1/256	53.6	5.0	20.0

ω also converges at first-order rate. Fig. 4 depicts the vesicle shape and its elastic tension profile at $t = 5.0$ using the grid resolution of $h = 1/256$. The pressure contour and velocity vectors illustrate the complex flow topology. The present method is capable of computing the elastic tension profile without numerical wiggles under an external hydrodynamical force. In Section 5.6, we further explore the tank-treading motion of the vesicle systematically.

We also repeat the simulation to investigate further the performance of the preconditioned GMRES to solve $\delta\sigma$ by comparing with un-preconditioned GMRES using the same stopping tolerance h^2 and preconditioned GMRES with tight stopping tolerance of 10^{-12}. In a similar manner, Table 8 summarizes the performance of the GMRES method in terms of average iteration counts for different grid resolutions. We confirm that the preconditioner is still very effective (an only slight increase of iteration number as the grid is linearly refined) even when an external hydrodynamical force is applied.

The previous IB formulations [15–20] have suffered a time-step restriction of $O(h^2)$ or higher in practice. In contrast, the present method allows a larger time-step restriction of $O(h)$. Given simulation time T, the number of time-steps required by the previous IB methods exhibits quadratic or higher growth with respect to the grid resolution. As a result, the total CPU time for the same simulation time T required by the previous IB methods would greatly exceed the one required by the present method. Comparing to the IB method with multifrontal solver [28], the present IBPM avoids explicit matrix entries and factorization at each time-step. Also, the expensive matrix inversion is replaced by FFT-based solvers. As shown



(a) CPU time per time-step

(b) Total CPU time

Fig. 5. CPU time complexity of IBPM with un-preconditioned and preconditioned GMRES method using stopping tolerance of h^2 and preconditioned GMRES method using stopping tolerance of 10^{-12}. The simulations are time-marching from time $t = 0$ to $t = 5.0$ using $\Delta t = h/4$.

in Fig. 5, we report the CPU time complexity of IBPM with un-preconditioned and preconditioned GMRES method using the stopping tolerance of h^2, and preconditioned GMRES method using stopping tolerance of 10^{-12}. The computational cost required by IBPM is significantly lower than the previous IB method [28]. Fig. 5(a) and Fig. 5(b) show that both the CPU time per time-step and total CPU time of all the cases are indeed linearithmic $O(1/h^2 \log 1/h^2)$, which confirms that the FFT-based solvers are the dominant part of the computational cost.

5.4. Equilibrium shape in a quiescent fluid

We simulate the dynamics of a vesicle with an initial ellipse shape immersed in a quiescent fluid domain. Since the initial shape of the vesicle does not correspond to its equilibrium shape, it deforms to minimize the Helfrich energy until reaching its equilibrium state. We define the reduced area $V = A/(\pi R_0)^2$ to classify the vesicle configuration with the length-scale $R_0 = L/2\pi$. We perform the simulations of a vesicle with unit area, immersed in a quiescent fluid domain $\Omega = [-2, 2] \times [-2, 2]$, by varying its reduced area $V = 0.5, 0.6, 0.7, 0.8, 0.9$. The simulations are conducted using a time-step of $\Delta t = 0.25h$ and a fine grid resolution of $h = 1/128$. All the simulations are computed up to the steady state occurs.

Fig. 6 illustrates the initial states and equilibrium states of the vesicle with different reduced area $V = 0.5, 0.6, 0.7, 0.8, 0.9$. We also use color contour to show the elastic tension distribution $\sigma R_0^2/c_b$ along the interface. We observe that the results are congruent with the literature [23], which indicate that the reduced area V dictates the equilibrium shape of vesicle dynamics.

5.5. Vesicle dynamics in Poiseuille flow

We explore the vesicle dynamics in Poiseuille flow imposed with a parabolic distribution of the velocity $u = u_{\max}\left[1 - (y/L_y)^2\right]$ where $u_{\max}$ is the maximum velocity at the center line located at $y = 0$ and $2L_y$ is the width of the channel. Fig. 7 is a schematic diagram of the initial placement of a vesicle in Poiseuille flow. When the vesicle is initially released at a position symmetric about the centerline, it develops a symmetric parachute shape. However, with a small offset from the centerline c_0, the vesicle develops an asymmetric slipper shape and migrates. We define the dimensionless velocity $w = u_{\max}\lambda_t/R_0$ with the time-scale $\lambda_t = \nu R_0^3/c_b$ to characterize the vesicle dynamics in Poiseuille flow. We perform a series of simulations of a vesicle with unit area and reduced area $V = 0.7$ by varying the offset from the centerline $c_0 = 0$, R_0, $-R_0$, and $w = 60$ in a fluid channel $\Omega = [0, 384] \times [0, 6]$. The confinement ratio is $L_y/R_0 \approx 5$. The simulations are time-marching from $t = 0$ to $t = 96$ using a time-step of $\Delta t = 0.125h\lambda_t/(wR_0)$ and a grid resolution of $h = 1/32$.

Fig. 8 depicts the transient evolution of a vesicle in Poiseuille flow initially placed at (a) $c_0 = 0$, (b) R_0, and (c) $-R_0$. The contour plot represents the elastic tension distribution $\sigma R_0^2/c_b$ along the interface. When $c_0 = 0$, the vesicle develops a parachute shape with negative curvature at the rear due to the symmetric distribution of the velocity. The symmetric distribution of elastic tension is identified on the interface with the lowest value occurring at the tips of the vesicle. Since the shear rate is minimal at the centerline, we notice higher elastic tension occurs at the region away from the centerline due to the increasing shear rate. When the vesicle is initially released at an offset from the centerline $c_0 = R_0$, the vesicle migrates to the centerline gradually while undergoes deformations; from an initially elliptical, to an asymmetric slipper

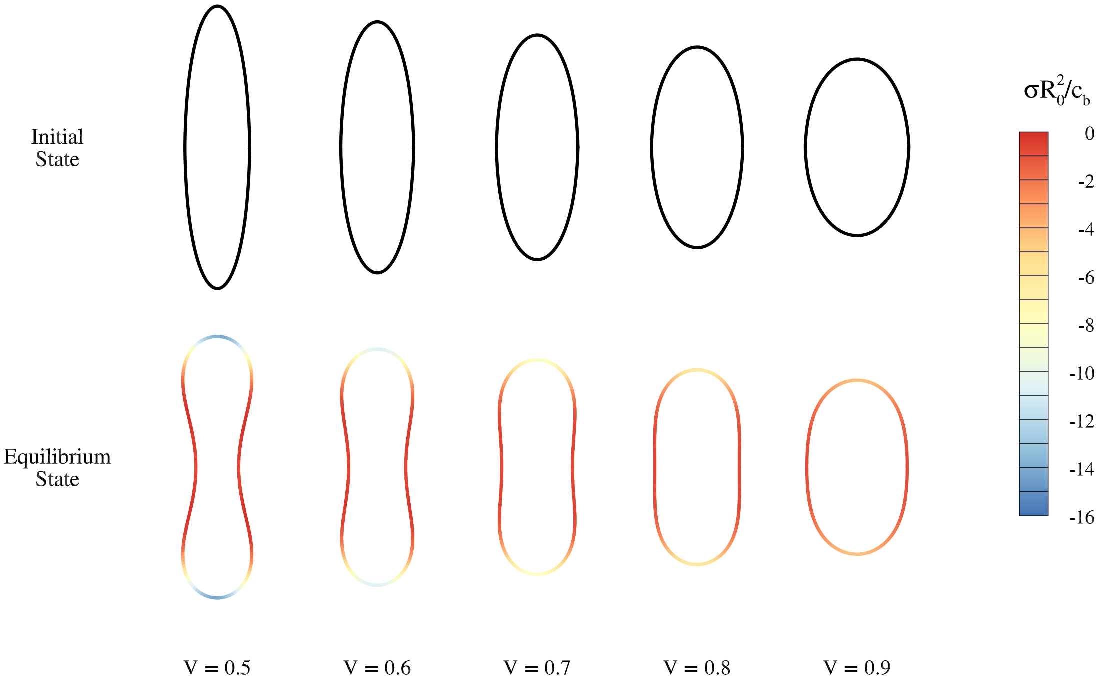


Fig. 6. Initial states and equilibrium states of a unit area vesicle with different reduced area $V = 0.5$, 0.6, 0.7, 0.8, 0.9, in conjunction with contour plot of the elastic tension distribution $\sigma R_0^2/c_b$ along the interface.

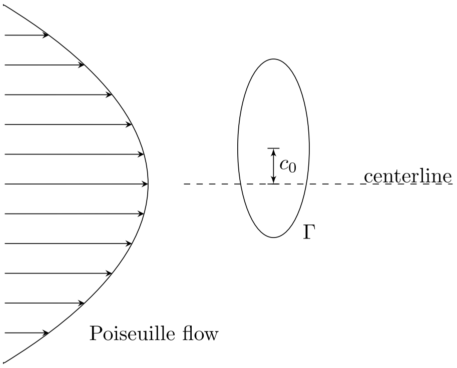
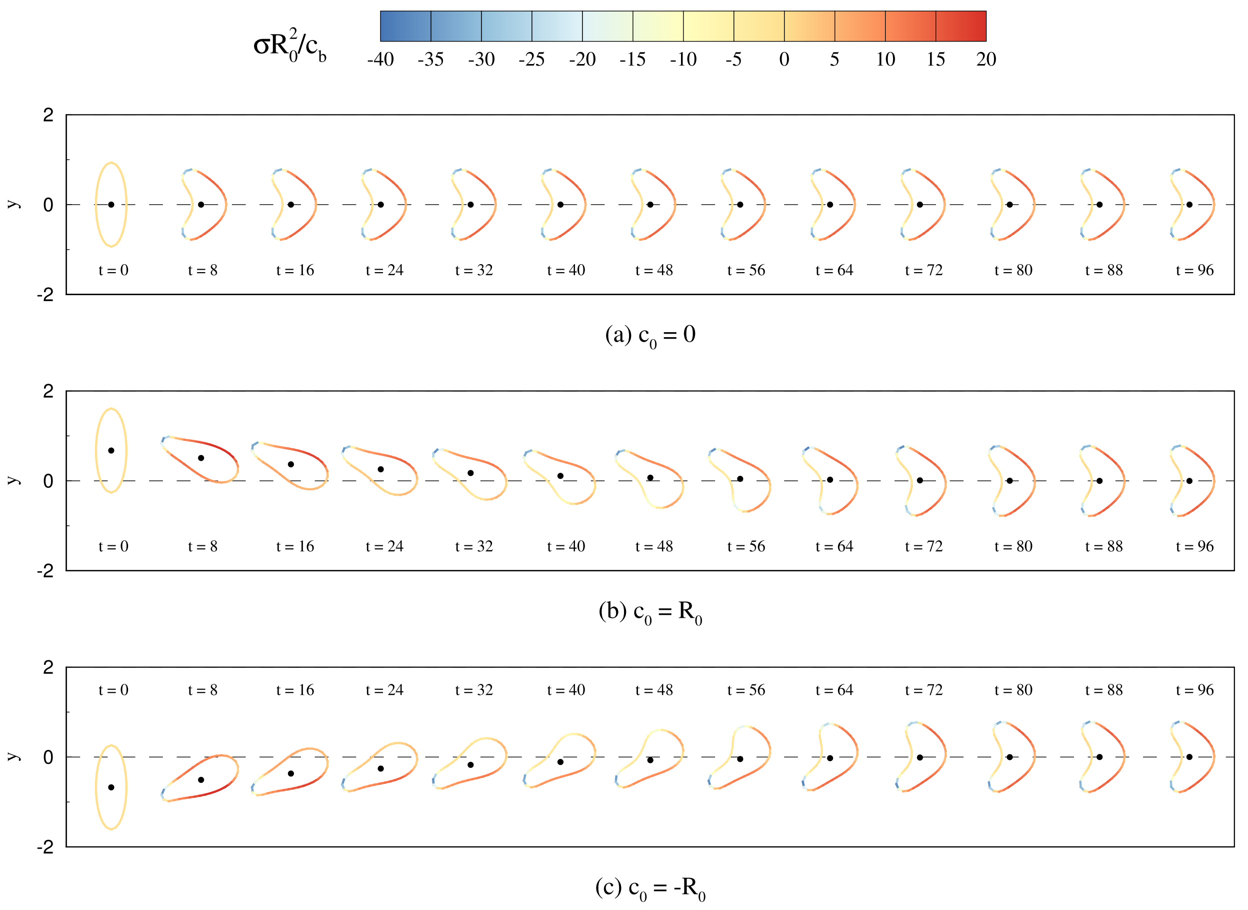


Fig. 7. Schematic diagram of the initial placement of a vesicle in Poiseuille flow where c_0 is an offset from the centerline.

shape, and finally to a parachute shape settles at the centerline. The formation of the asymmetric slipper shape is attributed to the unequal shear rate imposed by the Poiseuille flow on the vesicle, which is responsible for the lateral migration. As a result, asymmetric distribution of elastic tension is observed on the interface, with the highest value occurring in the top-front region and the lowest value occurring in the top region. Similar transient behaviors are also observed in the case $c_0 = -R_0$ in terms of lateral migration and the elastic tension distribution. In general, c_0 appears to be a critical parameter that determines the vesicle shape evolution and lateral migration in Poiseuille flow. The shape evolution is qualitatively congruent with the numerical results from the literature [40,41].

5.6. *Vesicle dynamics in a shear flow*

We explore the vesicle dynamics in an applied shear flow. A vesicle undergoes a tank-treading motion once it reaches the equilibrium under the condition of the matched viscosity between the exterior and the interior fluids. The dimensionless shear rate is defined as $\chi \equiv \dot{\gamma}\lambda_t$ with the time-scale $\lambda_t = \nu R_0^3/c_b$ to characterize the vesicle dynamics in shear flow. Systematically, we perform the simulations of a vesicle with unit area by varying its reduced area $V = 0.5$, 0.6, 0.7, 0.8, 0.9, and the dimensionless shear rate $\chi = 1$, 10, 100 in a fluid domain $\Omega = [0, 8] \times [0, 8]$. The simulations are computed up to the steady state using a time-step of $\Delta t = 0.125h\lambda_t/\chi$ and a grid resolution of $h = 1/64$.



(a) $c_0 = 0$

(b) $c_0 = R_0$

(c) $c_0 = -R_0$

Fig. 8. Transient evolution of a vesicle with reduced area of $V = 0.7$ in Poiseuille flow with initial placement (a) $c_0 = 0$, (b) $c_0 = R_0$, and (c) $c_0 = -R_0$. The dashed line represents the centerline of Poiseuille flow, and the solid dot represents the center of mass of the vesicle. For the initial placement $c_0 = 0$, the vesicle develops a parachute shape. For the initial placement $c_0 = R_0$, and $-R_0$, the vesicle migrates to the centerline while deforms to an asymmetric slipper shape, and finally to a parachute shape settles at the centerline.

Fig. 9 depicts the transient behavior of a vesicle with a reduced area of $V = 0.5$ under a shear flow with a dimensionless shear rate of $\chi = 1$. In the beginning, the vesicle deforms, and orients itself to a fixed inclination angle with the flow direction. After that, it undergoes tank-treading motion with a constant frequency proportional to the shear rate. Notably, the elastic tension distribution $\sigma R_0^2/c_b$ is steady in terms of the phase angle ϕ/π in the steady tank-treading regime. Fig. 10 shows the equilibrium vesicle shapes with reduced area of $V = 0.5$, 0.6, 0.7, 0.8, 0.9 and with dimensionless shear rate of $\chi = 1$, 10, 100. The elastic tension distributions are also depicted in the figure by contour plots and scaled individually for each case with the positive value (red-end) representing tension and the negative value (blue-end) representing compression. One can see that, the highest values occur in the middle of the vesicle, while the lowest values occur at both ends of the vesicle coinciding to the inclination angle.

To quantify the vesicle dynamics in a shear flow, Fig. 11 plots the inclination angle θ/π and the scaled tank-treading evolution frequency $\omega\lambda_t/\chi$ of the vesicle in terms of reduced area V at their equilibrium shape under the shear flow. Both the quantities are observed to be independent of the dimensionless shear rate χ but directly proportional to the reduced area V. As shown in Fig. 11, the present numerical results are in a good agreement with Keller-Skalak analytical model [42] and the numerical results obtained from the penalty IB method [20].

6. Conclusion

We have developed an IBPM based on an unconditionally energy stable scheme for modeling the vesicle dynamics in a viscous fluid. In light of the fractional step analysis of the previous works [29,30], we have exploited the algebraic structure of the underlying scheme and utilized the block LU decomposition technique to develop the numerical algorithm. Both the discrete fluid incompressibility and discrete interface inextensibility are satisfied simultaneously. This is in contrast to the previous approaches adopted by [15–20] in the context of IB method where a penalty formulation is applied by modifying the elastic tension energy, and a severe time-step must be imposed due to the stiffness of the penalization parameter. In

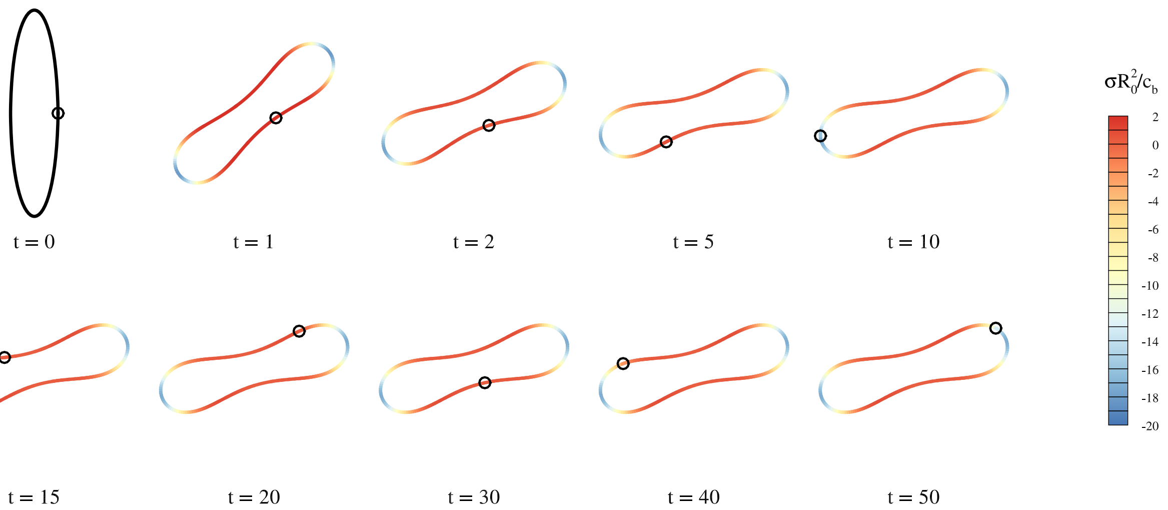


Fig. 9. Transient evolution of a vesicle with reduced area of $V = 0.5$, in a shear flow with dimensionless shear rate of $\chi = 1$. The marker acts as a reference point to track the tank-treading motion of the vesicle.

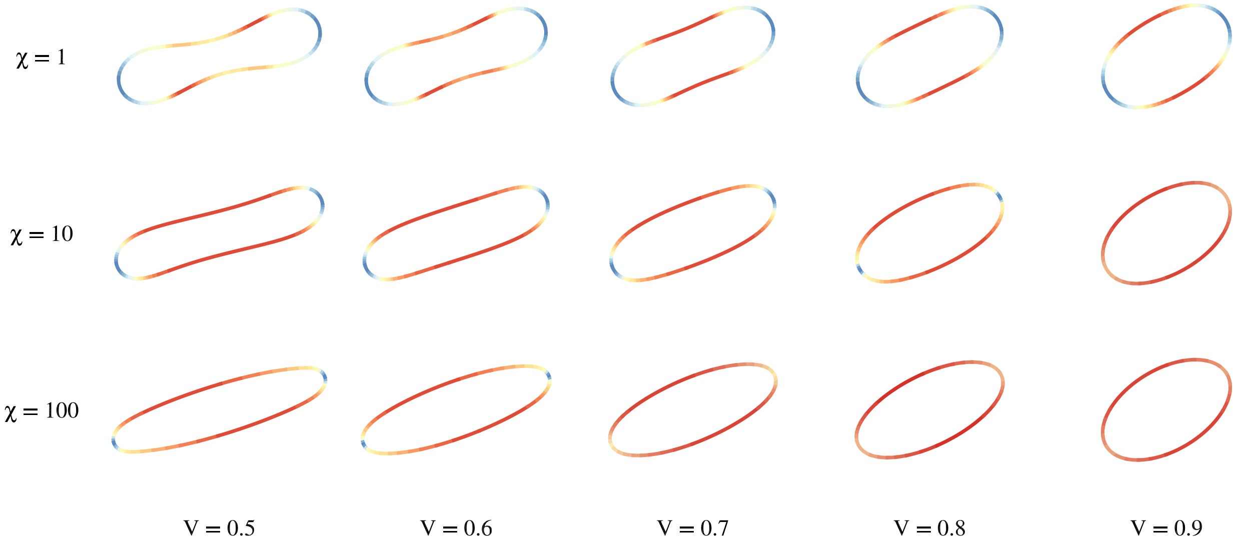
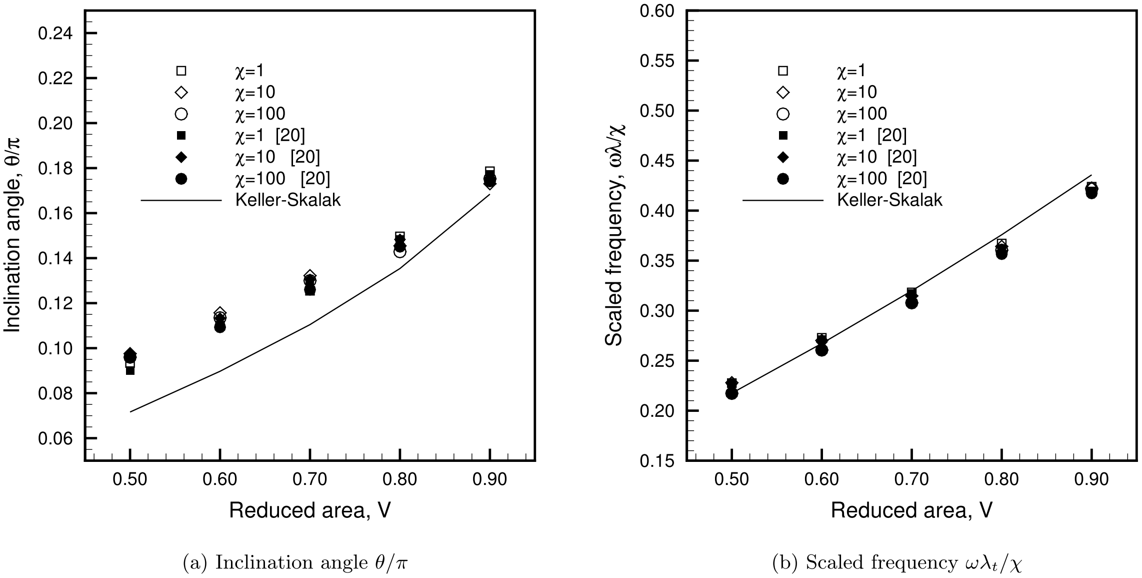


Fig. 10. The equilibrium vesicle shapes of $V = 0.5$, 0.6, 0.7, 0.8, 0.9 under shear rate of $\chi = 1$, 10, 100. The elastic tension distributions are represented by contour plots which are scaled individually for each case with the positive value (red-end) representing tension and the negative value (blue-end) representing compression.

conjunction with the implicit discretization of bending force, the present numerical algorithm significantly alleviates the severe time-step restriction, and the numerical stability is assured by the non-increasing of total discrete energy during the simulation. Another appealing advantage of IBPM is its computational efficiency. All the solution variables are decoupled while maintaining numerical stability and accuracy, which allows us to solve each of the variables independently.

For the present implementation, the overall algorithm is matrix-free and takes a linearithmic complexity. We utilize the FFT-based Poisson and modified Helmholtz solvers to compute the inverse of the Laplacian and modified Helmholtz operators, respectively. The implicit treatment of bending force and the solution of elastic tension are computed iteratively using the GMRES method, while each GMRES iteration requires a FFT-based solver and sparse matrix-vector products. To further accelerate the computation, we have constructed a simple and efficient GMRES preconditioner based on an incomplete LU factorization of the approximated Schur complement. The preconditioner has proved effective and shown essentially grid-independent for the GMRES iteration. We have demonstrated the accuracy and robustness of the present algorithm by simulating the vesicle dynamics in a quiescent fluid, Poiseuille flow, and shear flow. The numerical results are congruent with the analytical and other numerical results obtained from the literature.

The present IBPM can be extended to 3D. This extension is highly non-trivial due to the discretization of elastic tension and bending force terms on an evolving closed surface triangulation. The primary challenge is to develop a 3D uncondition-



(a) Inclination angle θ/π

(b) Scaled frequency $\omega\lambda_t/\chi$

Fig. 11. A comparison of the present inclination angle and tank-treading frequency with the results obtained by the Keller-Skalak theory [42] and the penalty IB method [20]. Inclination angle θ/π and the scaled tank-treading evolution frequency $\omega\lambda_t/\chi$ of the vesicle in terms of reduced area V at their equilibrium tank-treading regime. The solid line represents Keller-Skalak analytical model [42].

ally energy stable scheme (see the 2D counterpart in [28]) as in Eqs. (7)-(12). Currently, we had made progress in dealing with the elastic tension discretization in 3D. The discrete skew-adjoint property for the elastic tension and velocity on a triangulated closed surface can be proved and verified numerically. The aforementioned property is important to establish a discrete energy estimate. (More precisely, it can be used to prove that the elastic tension has no contribution on the decaying energy rate when the surface incompressibility is imposed in 3D). In addition, to handle the bending force, it is required to approximate the mean curvature, Gaussian curvature, and also surface Laplacian of mean curvature on a triangulated closed surface, which poses another challenging task when the vesicle surface is not known a priori. The second author and his coworkers have published a 3D vesicle simulation [19] based on a nearly incompressible approach (without solving the elastic tension) in which the elastic tension and bending force are computed explicitly. Certainly, the scheme in [19] is not unconditionally energy stable and has a rather restrictive time-step. Based on those challenges mentioned above, we will leave the full 3D extension to future work. We envision a long-term study by utilizing IBPM to simulate more physiologically complex dynamics, including the effect of membrane elasticity; the dependence on the physical properties such as viscosity ratio and density ratio; as well as the external effect of gravity and electrohydrodynamics.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

The work of M.-C. Lai was supported in part by Ministry of Science and Technology of Taiwan under research grant MOST-107-2115-M-009-016-MY3 and National Center for Theoretical Sciences. We are grateful to the National Center for High-performance Computing for computer time and facilities.

References

[1] C.S. Peskin, The immersed boundary method, Acta Numer. 11 (2002) 1–39.
[2] H. Noguchi, G. Gompper, Shape transitions of fluid vesicles and red blood cells in capillary flows, Proc. Natl. Acad. Sci. USA 102 (2005) 14159–14164.
[3] M.J. Stevens, Coarse-grained simulations of lipid bilayers, J. Chem. Phys. 121 (2004) 11942–11948.
[4] M. Kraus, W. Wintz, U. Seifert, R. Lipowsky, Fluid vesicles in shear flow, Phys. Rev. Lett. 77 (1996) 3685–3688.
[5] S.K. Veerapaneni, D. Gueyffier, D. Zorin, 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.
[6] 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.
[7] G. Boedec, M. Leonetti, M. Jaeger, 3D vesicle dynamics simulations with a linearly triangulated surface, J. Comput. Phys. 230 (2011) 1020–1034.
[8] S.K. Veerapaneni, A. Rahimian, G. Biros, D. Zorina, A fast algorithm for simulating vesicle flows in three dimensions, J. Comput. Phys. 2230 (2011) 5610–5634.

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

[10] A. Farutin, T. Biben, C. Misbah, 3D numerical simulations of vesicle and inextensible capsule dynamics, J. Comput. Phys. 275 (2014) 539–568.

[11] T. Biben, C. Misbah, Tumbling of vesicles under shear flow within an advected-field approach, Phys. Rev. E 67 (2003) 031908.

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

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

[14] S. Aland, S. Egerer, J. Lowengrub, A. Voigt, Diffuse interface models of locally inextensible vesicles in a viscous fluid, J. Comput. Phys. 277 (2014) 32–47.

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

[16] Y. Kim, M.-C. Lai, Numerical study of viscosity and inertial effects on tank-treading and tumbling motions of vesicles under shear flow, Phys. Rev. E 86 (2012) 066321.

[17] 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 (2012) B692–B710.

[18] W.-F. Hu, Y. Kim, 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.

[19] Y. Seol, W.-F. Hu, Y. Kim, M.-C. Lai, An immersed boundary method for simulating vesicle dynamics in three dimensions, J. Comput. Phys. 322 (2016) 125–141.

[20] Y. Seol, Y.-H. Tseng, Y. Kim, M.-C. Lai, An immersed boundary method for simulating Newtonian vesicles in viscoelastic fluid, J. Comput. Phys. 376 (2019) 1009–1027.

[21] Z. Li, M.-C. Lai, New finite difference methods based on IIM for inextensible interfaces in incompressible flows, East Asian J. Appl. Math. 1 (2011) 155–171.

[22] Z. Tan, D.-V. 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.

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

[24] D. Salac, M.J. Miksis, Reynolds number effects on lipid vesicles, J. Fluid Mech. 711 (2012) 122–146.

[25] A. Laadhari, P. Saramito, C. Misbah, Vesicle tumbling inhibited by inertia, Phys. Fluids 24 (2012) 031901.

[26] 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.

[27] V. Doyeux, Y. Guyot, V. Chabannes, C. Prud'homme, M. Ismail, Simulation of two-fluid flows using a finite element/level set method. Application to bubbles and vesicle dynamics, J. Comput. Appl. Math. 246 (2013) 251–259.

[28] M.-C. Lai, K.C. Ong, Unconditionally energy stable schemes for the inextensible interface problem with bending, SIAM J. Sci. Comput. 41 (2019) B649–B668.

[29] J.S. Perot, An analysis of the fractional step method, J. Comput. Phys. 108 (1993) 51–58.

[30] K. Taira, T. Colonius, The immersed boundary method: a projection approach, J. Comput. Phys. 225 (2007) 2118–2137.

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

[32] Y. Bao, J. Kaye, C.S. Peskin, A Gaussian-like immersed-boundary kernel with three continuous derivatives and improved translational invariance, J. Comput. Phys. 316 (2016) 139–144.

[33] T.Y. Hou, 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.

[34] H. Uzawa, Iterative methods for concave programming, in: K.J. Arrow, L. Hurwicz, H. Uzawa (Eds.), Studies in Linear and Nonlinear Programming, Stanford University Press, 1958.

[35] A.J. Chorin, Numerical solution of the Navier-Stokes equations, Math. Comput. 22 (1968) 745–762.

[36] R. Temam, Sur l'approximation de la solution des équations de Navier-Stokes par la méthode des pas fractionnaires, Arch. Ration. Mech. Anal. 33 (1969) 377–385.

[37] W.W. Hager, Updating the inverse of a matrix, SIAM Rev. 31 (1989) 221–239.

[38] Y. Saad, M.H. Schultz, GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Stat. Comput. 7 (1986) 856–869.

[39] Y. Saad, ILUT: a dual threshold incomplete LU factorization, Numer. Linear Algebra Appl. 1 (1994) 387–402.

[40] G. Danker, P.M. Vlahovska, C. Misbah, Vesicles in Poiseuille flow, Phys. Rev. Lett. 102 (2009) 148102.

[41] B. Kaoui, G. Biros, C. Misbah, Why do red blood cells have asymmetric shapes even in a symmetric flow?, Phys. Rev. Lett. 103 (2009) 188101.

[42] S.R. Keller, R. Skalak, Motion of a tank-treading ellipsoidal particle in a shear flow, J. Fluid Mech. 120 (1982) 27–47.