



# Simulating the dynamics of inextensible vesicles by the penalty immersed boundary method

Yongsam Kim<sup>a,1</sup>, Ming-Chih Lai<sup>b,\*</sup>

<sup>a</sup> Department of Mathematics, Chung-Ang University, Dongjak-Gu, Heukseok-Dong 221, Seoul 156-756, South Korea

<sup>b</sup> Center of Mathematical Modeling and Scientific Computing, Department of Applied Mathematics, National Chiao Tung University, 1001, Ta Hsueh Road, Hsinchu 300, Taiwan

## ARTICLE INFO

### Article history:

Received 21 September 2009

Received in revised form 24 February 2010

Accepted 12 March 2010

Available online 17 March 2010

### Keywords:

Penalty immersed boundary method

Inextensible vesicles

Binary-component vesicles

## ABSTRACT

In this paper, we develop an immersed boundary (IB) method to simulate the dynamics of inextensible vesicles interacting with an incompressible fluid. In order to take into account the inextensibility constraint of the vesicle, the penalty immersed boundary (pIB) method is used to virtually decouple the fluid and vesicle dynamics. As numerical tests of our current pIB method, the dynamics of single and multiple inextensible vesicles under shear flows have been extensively explored, and compared with the previous literature. The method is also validated by a series of convergence study, which confirms its consistent first-order accuracy on the velocity field, the vesicle configuration, the vesicle area and the perimeter errors. In addition, the method is also applied to study a binary-component vesicle problem.

© 2010 Elsevier Inc. All rights reserved.

## 1. Introduction

We introduce an immersed boundary (IB) method [25] to simulate the dynamics of inextensible vesicles in an incompressible viscous fluid. The vesicle dynamics is important for numerous biophysical systems that involve deforming fibers interacting with a surrounding fluid. The inextensible vesicles have attracted a great deal of attention recently as they are present in many biological systems [18] and can be considered as models of red blood cells [24,29] and drug-carrying capsules [34]. The dynamics of moving vesicles, which are determined by their boundary elasticity, inextensibility, and hydrodynamical force, have been extensively explored both experimentally [7,12] (and the references therein) and computationally [18,38,36].

Vesicle simulations have been conducted mostly by molecular dynamics models [23] or by continuum mechanics models of the fluid and the vesicle membrane using the boundary integral method [36]. Here, we use a Cartesian-grid based IB method [25]. The IB method is generally a useful computational tool for the problems in which a thin elastic boundary is immersed in and interacting with a surrounding fluid [4,3,26,11,13,17]. In [14], the authors have introduced a new version of the IB method, the penalty immersed boundary (pIB) method, which can handle efficiently the case of massive elastic boundary, and have shown that the new method can be applied to many problems in which the mass of the immersed boundary plays a crucial dynamical role, see also [2,9,15,16] for more examples of the application of the pIB method. This method has the virtue of simplicity; one can easily implement it in the context of an existing IB method code for the massless boundary case.

\* Corresponding author. Tel.: +886 3 5731361; fax: +886 3 5724679.

E-mail addresses: [kimy@cau.ac.kr](mailto:kimy@cau.ac.kr) (Y. Kim), [mclai@math.nctu.edu.tw](mailto:mclai@math.nctu.edu.tw) (M.-C. Lai).

<sup>1</sup> Tel.: +82 2 820 5219; fax: +82 2 823 5214.

The basic idea of the pIB method [14] is to use two Lagrangian immersed boundaries to represent the real immersed boundary for different purposes. The first immersed boundary is massless and interacts with the fluid directly as in the traditional IB computation, i.e., it moves at the local fluid velocity and applies the force locally to the fluid. The other immersed boundary is massive which carries the fiber mass and is connected to the previous massless boundary by a system of stiff springs. However, this massive boundary does *not* interact with the fluid directly and it moves according to Newton's law of motion with the forces generated by the system of stiff springs that connects the two boundary representations. Roughly speaking, as the spring stiffness tends to infinity, the massive boundary will track the motion of the massless boundary and endow it with the mass. This behavior has been verified quantitatively in [14].

In order to apply the penalty IB idea to the inextensible vesicle problems, as in [14], we represent the vesicle boundary by two Lagrangian immersed boundaries. 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. This idea of separation of dynamical systems using a penalty idea has also been used in [10].

To show that the present developed pIB method is a robust and efficient numerical tool to handle the inextensible boundary in a fluid, we simulate several benchmark two-dimensional problems with single and multiple vesicles under a shear flow. From these examples, we show that the simulation results obtained by the pIB method are comparable to those obtained from previous literature [7,12,38,36,23]. In order to provide further verification that the pIB method correctly solves the inextensible vesicle dynamics, we perform a convergence study, which confirms the consistent first-order accuracy on the velocity field, the vesicle configuration, the vesicle area error, and the vesicle perimeter error. Finally, we extend the method to solve a binary-component vesicle model which is obtained by simplifying the model in [32].

## 2. Equations of motion

We begin by stating the mathematical formulation of the equations of motion for a system comprised of a two-dimensional viscous incompressible fluid in a domain  $\Omega$  containing an immersed, inextensible, massless vesicle boundary  $\Gamma$ . We assume that the fluid is the same inside and outside of the vesicle boundary, and the equations of motion in immersed boundary formulation are

$$\rho \left( \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) = -\nabla p + \mu \nabla^2 \mathbf{u} + \mathbf{f}, \tag{1}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{2}$$

$$\mathbf{f}(\mathbf{x}, t) = \int_{\Gamma} \mathbf{F}(s, t) \delta(\mathbf{x} - \mathbf{X}(s, t)) ds, \tag{3}$$

$$\frac{\partial \mathbf{X}}{\partial t}(s, t) = \mathbf{U}(s, t) = \int_{\Omega} \mathbf{u}(\mathbf{x}, t) \delta(\mathbf{x} - \mathbf{X}(s, t)) d\mathbf{x}, \tag{4}$$

$$\mathbf{F}(s, t) = \frac{\partial}{\partial s} (\sigma(s, t) \boldsymbol{\tau}(s, t)) - \frac{\partial^2}{\partial s^2} \left( c_b \frac{\partial^2 \mathbf{X}(s, t)}{\partial s^2} \right), \tag{5}$$

$$\boldsymbol{\tau}(s, t) = \frac{\partial \mathbf{X}}{\partial s} / \left| \frac{\partial \mathbf{X}}{\partial s} \right|, \tag{6}$$

$$\frac{\partial \mathbf{U}}{\partial s} \cdot \frac{\partial \mathbf{X}}{\partial s} = 0. \tag{7}$$

Eqs. (1) and (2) are the familiar Navier–Stokes equations for an incompressible viscous fluid. The values  $\rho$  and  $\mu$  denote constant density and viscosity of the fluid, respectively. The unknown variables in the fluid equations are the fluid velocity,  $\mathbf{u}(\mathbf{x}, t)$ ; the fluid pressure,  $p(\mathbf{x}, t)$ , where  $\mathbf{x} = (x, y)$  are fixed Cartesian coordinates, and  $t$  is the time. The other unknown in the above system  $\mathbf{X}(s, t)$  describes the configuration of the vesicle boundary at any given time. Though the variable  $s$  can be naturally chosen as the arc-length parameter since the vesicle is inextensible and thus the total length of vesicle boundary is conserved, we normally use non arc-length parameter  $s$  as in the traditional IB formulation.

Eqs. (3) and (4) both involve the two-dimensional Dirac delta function  $\delta(\mathbf{x}) = \delta(x)\delta(y)$ , which expresses the local character of the interaction between the fluid and the immersed boundary. Eq. (3) simply expresses the relation between the two corresponding force densities  $\mathbf{f}(\mathbf{x}, t) d\mathbf{x}$  and  $\mathbf{F}(s, t) ds$ . Eq. (4) is the equation of motion of the immersed vesicle boundary, i.e., the boundary moves at the local fluid velocity.

Eqs. (5)–(7) are the vesicle boundary equations which are written in Lagrangian form. Eq. (5) is the standard formula for a vesicle under tension (first term, denoted by  $\mathbf{F}_e$ ) and bending force (second term, denoted by  $\mathbf{F}_b$ ), where  $\mathbf{F} ds$  is the force applied by an element of vesicle  $ds$  to the fluid. The function  $\boldsymbol{\tau}(s, t)$  is the unit tangent vector to the vesicle boundary. Notice that, unlike the traditional IB formulation in which the surface tension is either known or a function of immersed boundary configurations, here, the surface tension  $\sigma(s, t)$  is an unknown function which plays the role of a Lagrange multiplier to enforce the inextensibility constraint of the vesicle, Eq. (7). This inextensibility constraint is equivalent to the fact that the local stretching factor of the vessel satisfies  $\frac{d}{dt} \left| \frac{\partial \mathbf{X}}{\partial s} \right| = 0$ . To see how the boundary force can be expressed as Eq. (5), we explain as follows.

The vesicle membrane energy can be modelled by the Helfrich type energy ( $E_b$ ) [8] to resist the bending of the membrane. To enforce the inextensibility constraint, this energy is replaced by a modified form using the method of Lagrangian multiplier. More precisely, the total energy can be written as

$$E = E_e + E_b = \int_{\Gamma} \sigma ds + \int_{\Gamma} \frac{c_b}{2} \left| \frac{\partial^2 \mathbf{X}}{\partial s^2} \right|^2 ds, \tag{8}$$

where  $\sigma(s, t)$  is an unknown surface tension and  $c_b$  is the constant bending rigidity. By taking the variational derivative of the above energy form, the boundary force Eq. (5) is obtained by

$$\mathbf{F} = \mathbf{F}_e + \mathbf{F}_b = - \left( \frac{\delta E_e}{\delta \mathbf{X}} + \frac{\delta E_b}{\delta \mathbf{X}} \right). \tag{9}$$

In [38], Zhou and Pozrikidis consider the deformation of liquid capsules with incompressible (same as inextensible) interfaces in simple shear flow. They assume only the membrane tension force (first term in the right-hand side of Eq. (5)) to enforce the inextensibility. By using the boundary element formulation for Stokes flow, the authors discretize the whole equations by solving nonlinear system of vesicle boundary positions and the associated tension. Recently, Veerapaneni et al. [36] used a similar boundary integral method to simulate the dynamics of 2D [36] and 3D axis-symmetric [37] inextensible vesicles suspended in Stokes fluid. Unlike [38], Veerapaneni et al. [36,37] take the bending force into account. In this paper, we use the immersed boundary method to study the dynamics of inextensible vesicles in Navier–Stokes flow. Furthermore, in a later section, we consider the case of binary-component vesicle in which the bending rigidity  $c_b$  and spontaneous curvature are all dependent on the local heterogeneous concentration of those two components corresponding to liquid-ordered and liquid-disordered surface phases [22]. We note that a thermodynamically consistent model has been proposed to simulate the dynamics of two-dimensional multi-component vesicles in Stokes flow [32].

2.1. Penalty IB method

Since the system of Eqs. (1)–(7) is complete, one can directly solve the whole system using some numerical methods. To enforce the inextensibility constraint, however, the methods should handle this constraint realized by Eqs. (5)–(7) in an implicit way, which requires an iterative method to solve the whole system. To avoid this complication and make the scheme efficient, we modify the formulation of Eqs. (1)–(7) so that the penalty IB (pIB) method [14] can be applied. In the pIB method, we use a dual representation of the immersed boundary. One of its representatives, denoted by  $\mathbf{X}(s, t)$ , interacts with the fluid and moves at the local fluid velocity, just as in the case of an immersed boundary in traditional IB computation. The other representative, denoted by  $\mathbf{Y}(s, t)$ , is elastic and linked to  $\mathbf{X}(s, t)$  by a system of stiff springs. The boundary points of  $\mathbf{Y}(s, t)$  are not coupled directly to the fluid, and they move as if in a vacuum, see Fig. 1.

In mathematical terms, we replace Eqs. (5)–(7) by the following:

$$\mathbf{F} = K(\mathbf{Y}(s, t) - \mathbf{X}(s, t)) + R(\mathbf{V}(s, t) - \mathbf{U}(s, t)), \tag{10}$$

$$K(\mathbf{Y}(s, t) - \mathbf{X}(s, t)) + R(\mathbf{V}(s, t) - \mathbf{U}(s, t)) = \frac{\partial}{\partial s} (\sigma(s, t) \boldsymbol{\tau}(s, t)) - c_b \frac{\partial^4 \mathbf{Y}(s, t)}{\partial s^4}, \tag{11}$$

$$\boldsymbol{\tau}(s, t) = \frac{\partial \mathbf{Y}}{\partial s} / \left| \frac{\partial \mathbf{Y}}{\partial s} \right|, \tag{12}$$

$$\frac{\partial \mathbf{V}}{\partial s} \cdot \frac{\partial \mathbf{Y}}{\partial s} = 0, \tag{13}$$

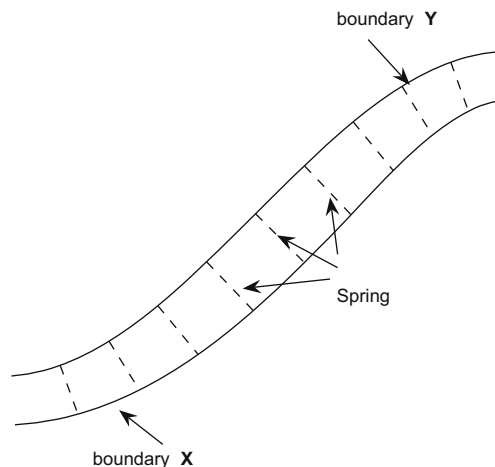


Fig. 1. The two immersed boundaries  $\mathbf{X}(s, t)$  and  $\mathbf{Y}(s, t)$  are linked together by a collection of a very stiff springs with rest length zero.

where  $\mathbf{U} = \partial\mathbf{X}/\partial t$  (defined in Eq. (4)) and  $\mathbf{V} = \partial\mathbf{Y}/\partial t$  are the velocities of the two representations of the immersed boundary.

Eq. (10) defines the force density  $\mathbf{F}$  which is transmitted by the boundary  $\mathbf{X}(s, t)$  to the fluid. It includes only the force density generated by the stiff springs that link the two representations of the immersed boundary and  $K$  and  $R$  are the *penalty* parameters. The larger they are, the greater the energy penalty that must be paid to separate the positions and velocities of the two boundary representations by any given amount. Eqs. (11)–(13) are the equations of motion and velocity of the boundary  $\mathbf{Y}(s, t)$ . Note that the penalty force is instantaneously balanced by the elastic force of the boundary  $\mathbf{Y}(s, t)$  and this elastic force does not act directly on the fluid, i.e., the boundary  $\mathbf{Y}(s, t)$  does not interact with the fluid directly.

Generally speaking, as the penalty parameters  $K$  and  $R$  go to infinity, the boundaries  $\mathbf{X}(s, t)$  and  $\mathbf{Y}(s, t)$  and velocities  $\mathbf{U}(s, t)$  and  $\mathbf{V}(s, t)$  will tend to coincide, and  $\mathbf{F}$  in Eq. (10) approaches the force  $\mathbf{F}$  in Eq. (5). In practice, we choose  $K$  and  $R$  sufficiently large to keep the positions and velocities of the two representations of the boundary close to each other. Notice that, the present pIB idea shares the same spirit with the ‘virtual boundary method’ [5,30] for simulating the fluid interacting with rigid boundaries. Both methods introduce the penalty parameters  $K$  and  $R$  to penalize the difference in the desired and numerical velocities and positions of the immersed boundary. The difference is that, the virtual boundary method is developed to impose the no-slip boundary condition while the present pIB method is to impose the inextensibility condition on the immersed boundaries. Besides, an extra unknown surface tension must be solved in the present approach.

The pIB method is an extension of the original IB method to handle the case where the mass of the immersed boundary plays an important dynamical role; for example, flapping filament in a flowing soap film, falling parachute in a steady flow, and compliant vessel interacting with blood [14,17,16]. The present pIB method is different from the one in [14] in two aspects; namely, (1) no mass is considered here, (2) an additional damping-like force  $R(\mathbf{V}(s, t) - \mathbf{U}(s, t))$  is added to the penalty force. Nevertheless, the massive boundary case can be handled simply by adding the D’Alembert force  $M(s) \frac{\partial^2 \mathbf{Y}(s, t)}{\partial t^2}$  to the left side of Eq. (11) where  $M(s)$  is the mass density of the boundary. Then Eq. (11) simply represents Newton’s law of motion. The addition of  $R(\mathbf{V}(s, t) - \mathbf{U}(s, t))$  to the penalty force makes it possible to apply the inextensibility constraint (Eq. (13)) to the force balance equation (Eq. (11)) by means of the velocity  $\mathbf{V}(s, t)$ . A similar idea was also used in [10]. The advantage of the pIB method is that we only need to solve Eqs. (10)–(13) to enforce the inextensibility constraint, which are decoupled from the fluid equations.

### 3. Numerical implementation of the pIB method

For numerical implementation, we use a first-order IB method, generalized to take into account the equations for the boundary  $\mathbf{Y}(s, t)$  that is linked to the boundary  $\mathbf{X}(s, t)$  by stiff springs. We use a superscript to denote the time level; thus,  $\mathbf{X}^n(s)$  is a shorthand for  $\mathbf{X}(s, n\Delta t)$ , where  $\Delta t$  is the size of the time step, and similarly for all other variables. Our goal is to compute updated  $\mathbf{u}^{n+1}$ ,  $\mathbf{v}^{n+1}$ ,  $\mathbf{x}^{n+1}$ , and  $\mathbf{y}^{n+1}$  from given data  $\mathbf{u}^n$ ,  $\mathbf{v}^n$ ,  $\mathbf{x}^n$  and  $\mathbf{y}^n$ .

The step-by-step procedure of the numerical implementation proceeds as follows:

$$\mathbf{F}^n(s) = K(\mathbf{Y}^n(s) - \mathbf{X}^n(s)) + R(\mathbf{V}^n(s) - \mathbf{U}^n(s)), \tag{14}$$

$$\mathbf{f}^n(\mathbf{x}) = \sum_s \mathbf{F}^n(s) \delta_h(\mathbf{x} - \mathbf{X}^n(s)) \Delta s, \tag{15}$$

$$\rho \left( \frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} + \frac{1}{2} \sum_{i=1,2} (u_i D_i^0 \mathbf{u} + D_i^0 (u_i \mathbf{u}))^n \right) + \mathbf{D}p^{n+1} = \mu \mathbf{L} \mathbf{u}^{n+1} + \mathbf{f}^n, \tag{16}$$

$$\mathbf{D} \cdot \mathbf{u}^{n+1} = 0, \tag{17}$$

$$\mathbf{U}^{n+1}(s) = \sum_{\mathbf{x}} \mathbf{u}^{n+1}(\mathbf{x}) \delta_h(\mathbf{x} - \mathbf{X}^n(s)) h^2, \tag{18}$$

$$\tilde{\mathbf{Y}}^{n+1}(s) = \mathbf{Y}^n(s) + \frac{\Delta t}{2} \mathbf{V}^n(s), \quad \tilde{\tau}^{n+1}(s) = \frac{\partial \tilde{\mathbf{Y}}^{n+1} / \partial s}{|\partial \tilde{\mathbf{Y}}^{n+1} / \partial s|} \tag{19}$$

$$K(\mathbf{Y}^n(s) - \mathbf{X}^n(s)) + R(\mathbf{V}^{n+1}(s) - \mathbf{U}^{n+1}(s)) = \frac{\partial}{\partial s} (\tilde{\sigma}^{n+1}(s) \tilde{\tau}^{n+1}(s)) - c_b \frac{\partial^4 \mathbf{Y}^n(s)}{\partial s^4}. \tag{20}$$

$$\frac{\partial \mathbf{V}^{n+1}}{\partial s} \cdot \frac{\partial \tilde{\mathbf{Y}}^{n+1}}{\partial s} = 0, \tag{21}$$

$$\mathbf{X}^{n+1}(s) = \mathbf{X}^n(s) + \Delta t \mathbf{U}^{n+1}(s), \tag{22}$$

$$\mathbf{Y}^{n+1}(s) = \mathbf{Y}^n(s) + \Delta t \mathbf{V}^{n+1}(s). \tag{23}$$

First, we calculate the force density  $\mathbf{F}^n(s)$  defined on Lagrangian grid points and spread it into the Eulerian grid force  $\mathbf{f}^n(\mathbf{x})$  in the Navier–Stokes equations. These are done by Eqs. (14) and (15) which are the discretizations of Eqs. (10) and (3), respectively. Here and throughout the paper  $\sum_{\mathbf{x}}$  denotes the sum over the rectangular lattice in physical space on which the fluid variables are defined. Similarly,  $\sum_s$  will denote the sum over the boundary nodes on which the two boundary positions  $\mathbf{X}$ ,  $\mathbf{Y}$ , and the force density  $\mathbf{F}$  are defined.

The discrete delta function  $\delta_h$  in Eqs. (15) and (18) is of the following form:

$$\delta_h(\mathbf{x}) = h^{-2} \phi\left(\frac{x_1}{h}\right) \phi\left(\frac{x_2}{h}\right), \tag{24}$$

where  $\mathbf{x} = (x_1, x_2)$  and  $h = \Delta x = \Delta y$  is the mesh width of the uniform square lattice. The function  $\phi$  is determined by the following conditions:

1.  $\phi$  is a continuous function.
2.  $\phi(r) = 0$  for  $|r| \geq 2$ .
3. For all  $r$ ,  $\sum_{j=\text{even}} \phi(r-j) = \sum_{j=\text{odd}} \phi(r-j) = \frac{1}{2}$ .
4. For all  $r$ ,  $\sum_j (r-j)\phi(r-j) = 0$ .
5. For all  $r$ ,  $\sum_j (\phi(r-j))^2 = C$ , where  $C$  is a constant, independent of  $r$ .

The motivation of these postulates is discussed in [26,27]. It follows that  $C = \frac{3}{8}$  and  $\phi$  is uniquely determined as

$$\phi(r) = \begin{cases} \frac{3-2|r|+\sqrt{1+4|r|-4r^2}}{8} & \text{if } |r| \leq 1 \\ \frac{5-2|r|-\sqrt{-7+12|r|-4r^2}}{8} & \text{if } 1 \leq |r| \leq 2 \\ 0 & \text{if } 2 \leq |r|. \end{cases} \tag{25}$$

This is an even, bell-shaped function, which is not only continuous but also has a continuous first derivative.

With the force density  $\mathbf{f}^n(\mathbf{x})$  in hand, we can turn to solving the Navier–Stokes equations. This is done by Eqs. (16) and (17) in which  $D_i^0$  is the standard central difference operator in the spatial direction denoted by  $i$ , where  $i = 1, 2$ , and  $L$  is the standard 5-point discrete Laplacian. Note that the skew-symmetric difference is used for the convection term [19,25]. The vector operator  $\mathbf{D} = (D_1, D_2)$  that is used for the discrete gradient and divergence can be defined as follows [28]:

$$(D_1)\phi(x_1, x_2) = \sum_{x'_1, x'_2} \phi(x'_1, x'_2) \gamma(x_1 - x'_1) \omega(x_2 - x'_2), \tag{26}$$

$$(D_2)\phi(x_1, x_2) = \sum_{x'_1, x'_2} \phi(x'_1, x'_2) \omega(x_1 - x'_1) \gamma(x_2 - x'_2), \tag{27}$$

where  $\gamma(x) = \delta_h(x + X)|_{X=-h/2}^{X=h/2}$  and  $\omega(x) = \int_{-h/2}^{h/2} \delta_h(x + X) dX$ . This operator is designed for “improved volume conservation” and is constructed according to a recipe introduced in [28] which ensures that the value of  $\mathbf{D} \cdot \mathbf{u}$  at any particular grid point is exactly equal to the average over an  $h \times h$  square centered on that grid point of the continuous divergence of the interpolated velocity field in which the immersed boundary points move. This definition “tunes” the operator  $\mathbf{D}$  to the function  $\delta_h$  that defines the interpolation scheme of the immersed boundary method.

Eqs. (16) and (17) form a linear system with constant coefficients in the unknowns  $\mathbf{u}^{n+1}, p^{n+1}$ . We solve this linear system by taking its discrete Fourier transform (implemented by the FFT algorithm), which reduces it to a collection of  $3 \times 3$  uncoupled linear systems, one at each grid point (wave number) in the discrete Fourier space. These  $3 \times 3$  linear systems are easily solved, and then the inverse discrete Fourier transform (again implemented by the FFT algorithm) is used to obtain the solution in the physical space. The above procedure is most naturally applied in the context of periodic boundary conditions, and for that reason we choose our fluid domain as a periodic box. Note, however, that there is no fundamental requirement of using a periodic domain in conjunction with the pIB method. Any fluid solver based on finite element or finite difference methods can be used, with whatever boundary conditions that solver can accommodate. We have emphasized the periodic case because of the efficiency that follows from the usage of the FFT algorithm for solving the linear systems that arise in our numerical scheme.

Next, we update the boundary velocity  $\mathbf{V}^{n+1}$  by solving Eq. (20) which is the discretization of Eq. (11). The boundary velocity  $\mathbf{U}^{n+1}(s)$  and the preliminary boundary position  $\tilde{\mathbf{Y}}^{n+1}(s)$  can be obtained by Eqs. (18) and (19), respectively. Since  $\tilde{\boldsymbol{\tau}}^{n+1}(s)$  is defined as in Eqs. (19)–(21) must be solved simultaneously in order to obtain  $\mathbf{V}^{n+1}(s)$  and  $\tilde{\sigma}^{n+1}(s)$ . To see how this is done, we first define

$$\tilde{\mathbf{F}}^n = K(\mathbf{Y}^n - \mathbf{X}^n) - R\mathbf{U}^{n+1} + c_b \frac{\partial^4 \mathbf{Y}^n}{\partial s^4}, \tag{28}$$

which is a known value. Then we can write Eq. (20) as

$$\frac{\partial}{\partial s} (\tilde{\sigma}^{n+1} \tilde{\boldsymbol{\tau}}^{n+1}) = R\mathbf{V}^{n+1} + \tilde{\mathbf{F}}^n. \tag{29}$$

Taking derivative to both sides of this equation with respect to the variable  $s$  and applying the inner product to the result with  $\tilde{\boldsymbol{\tau}}^{n+1}$ , we obtain

$$\frac{\partial^2}{\partial s^2} (\tilde{\sigma}^{n+1} \tilde{\boldsymbol{\tau}}^{n+1}) \cdot \tilde{\boldsymbol{\tau}}^{n+1} = \left( R \frac{\partial \mathbf{V}^{n+1}}{\partial s} + \frac{\partial \tilde{\mathbf{F}}^n}{\partial s} \right) \cdot \tilde{\boldsymbol{\tau}}^{n+1}. \tag{30}$$

Now, using the discrete version of inextensible constraint Eq. (21), we obtain the following second-order differential equation for  $\tilde{\sigma}^{n+1}$  as

$$\frac{\partial^2}{\partial s^2} (\tilde{\sigma}^{n+1} \tilde{\tau}^{n+1}) \cdot \tilde{\tau}^{n+1} = \frac{\partial \tilde{\mathbf{F}}^n}{\partial s} \cdot \tilde{\tau}^{n+1}. \tag{31}$$

Using the relations  $|\tilde{\tau}^{n+1}| = 1$ ,  $\tilde{\tau}_s^{n+1} \cdot \tilde{\tau}^{n+1} = 0$ , and  $\tilde{\tau}_{ss}^{n+1} \cdot \tilde{\tau}^{n+1} = -|\tilde{\tau}_s^{n+1}|^2$ , we can write the left-hand side of Eq. (31) as  $(\tilde{\sigma}^{n+1} \tilde{\tau}^{n+1})_{ss} \cdot \tilde{\tau}^{n+1} = \tilde{\sigma}_{ss}^{n+1} - |\tilde{\tau}_s^{n+1}|^2 \tilde{\sigma}^{n+1}$ . Thus Eq. (31) is a modified Helmholtz equation, which can be solved by a standard finite difference method. The condition number of the resultant matrix equation is approximately  $O(M^2)$ , where  $M$  is the number of the markers. Once we have obtained  $\tilde{\sigma}^{n+1}$ , the velocity  $\mathbf{V}^{n+1}$  can be obtained using Eq. (29).

Finally we update the configuration of the two representations of the boundary  $\mathbf{X}(s)$  and  $\mathbf{Y}(s)$  by Eqs. (22) and (23), which completes the evolution over one time step.

Note that, the present numerical scheme shown in (14)–(23) is in fact a combination of the forward and backward Euler methods so the accuracy is only first-order. (This is verified in Section 4.2 below.) Even though there are second-order IB methods available in the literature [6,19], they are only second-order accurate for problems with sufficiently smooth solutions, say an immersed structure with no zero thickness. Since this is not the case considered here, we could have no gain by using the second-order methods.

### 4. Numerical results

#### 4.1. Single vesicle in a shear flow

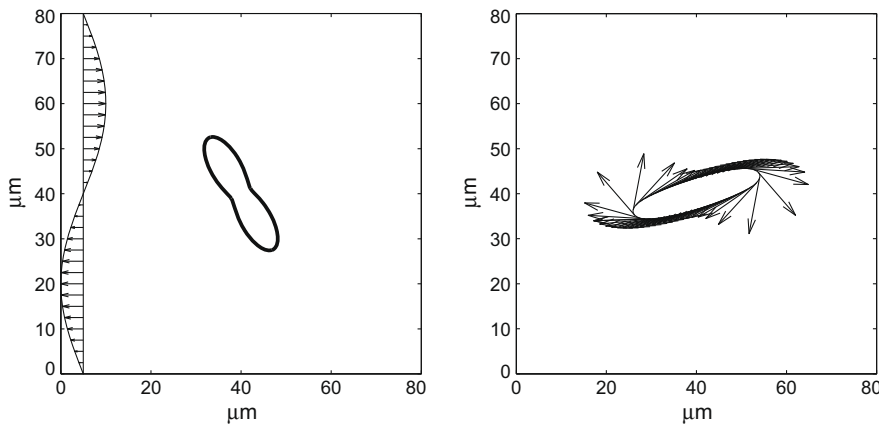
In this section, we introduce a two-dimensional computational model of a single vesicle suspended in a simple shear flow. This problem was explored by several researchers through numerical simulations [18,38,36] in Stokes flow.

We present the initial setting of our model and display the physical and computational parameters which are used in our numerical experiments.

Consider an incompressible viscous fluid in a square box  $(0, D) \times (0, D)$  where  $D = 80 \mu\text{m}$ . The fluid-filled domain contains an immersed elastic vesicle which is a closed curve. The interior and exterior of the vesicle consist of the same fluid. The left panel of Fig. 2 shows the initial configuration of a vesicle of an arbitrary shape. In order to apply a shear flow, it is natural to impose the background velocity field  $\mathbf{u}_0 = \gamma(y - D/2, 0)$  to the whole domain, where  $y$  is from 0 to  $D$  and  $\gamma$  is the shear rate. In this paper, however, since we use the periodic boundary condition for the computational domain, we impose the periodic shear flow field,  $\mathbf{u}_0(y) = \gamma(-\frac{D}{2\pi} \sin(2\pi y/D), 0)$  instead, see the left panel of Fig. 2.

In our simulations, we use the following parameters: the fluid density  $\rho$  is  $1.0 \text{ g/cm}^3$  and the viscosity  $\mu$  is  $0.01 \text{ g/(cm s)}$ , which is the same as those of water, the bending coefficient  $c_b$  is  $10^{-10} \text{ dyne cm}$ , the time step  $\Delta t = 4 \times 10^{-8} \text{ s}$ , and the spatial mesh width  $h = \Delta x = \Delta y = 80/256 \mu\text{m}$ . Following the analysis and scaling in [18,36], we define the length scale  $R_0 = L/2\pi$  where  $L$  is the perimeter of the vesicle boundary and the time scale  $\tau = \mu R_0^3/c_b$ . We generally use  $R_0 = 10 \mu\text{m}$ , which induces an intrinsic time scale  $\tau = 0.1 \text{ s}$ . We vary the initial configuration of the vesicle and the shear rate  $\gamma$  to explore the difference of the motion of a vesicle. These variations introduce two important dimensionless parameters; namely, the reduced area  $V = \frac{A}{\pi R_0^2} = \frac{4A\pi}{L^2}$  where  $A$  is the area of the vesicle, and the dimensionless shear rate  $\chi = \gamma \mu R_0^3/c_b = \gamma \tau = 0.1\gamma$ .

One important computational issue is how to choose the penalty coefficients  $K$  and  $R$  appropriately. Here, we offer a practical guide as follows. First, we choose some allowable maximum deviations for two representations  $|\mathbf{Y}(s, t) - \mathbf{X}(s, t)|$  and



**Fig. 2.** The configuration of a single vesicle suspended in the shear flow  $\mathbf{u}_0 = \gamma(-\frac{D}{2\pi} \sin(2\pi y/D), 0)$  at  $t = 0$  (left) and  $t = 0.04$  (right) in dimensionless time. Here the reduced area of the vesicle is  $V = 0.51$ , and the shear rate is  $\chi = 250$ . The right figure includes the velocity field on the vesicle boundary which shows a tank-treading tangential motion of the steady state vesicle.

$|\mathbf{V}(s, t) - \mathbf{U}(s, t)|$ , respectively. Then, we adjust (increase) the values of  $K$  and  $R$  until they are large enough that these allowable deviations will not be exceeded. However, during each run, the time step size  $\Delta t$  must be adjusted accordingly with the penalty parameters  $K$  and  $R$  to ensure the numerical stability. As discussed in [5,30], a useful time step stability constraint arising from solving Eqs. (14)–(17) can be chosen as

$$\Delta t < \frac{\sqrt{R^2 + 2KC} - R}{K}, \tag{32}$$

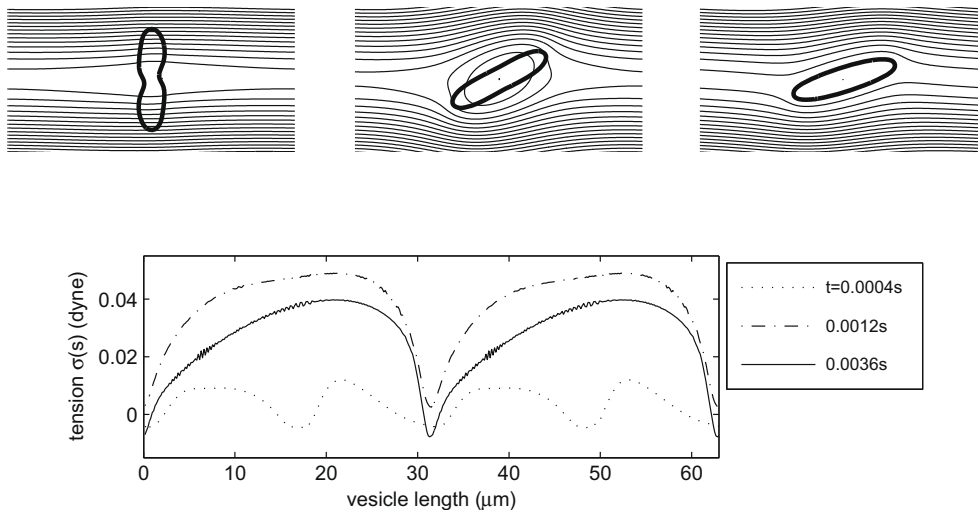
where  $C$  is a problem dependent constant of order one. A more concrete stability analysis has been characterized in [21]. (Notice that, the roles  $K$  and  $R$  played in the present work are similar to the ones  $-\alpha$  and  $-\beta$  in [5,30]. The integral feedback term in [5,30] involving the fluid velocity and desired boundary velocity can lead to the present form of Eq. (10) easily.) One should notice that the present time step constraint is more restrictive than the above criterion since we solve the Eqs. (28)–(31) to obtain the surface tension mainly in an explicit way. Based on the above process, we choose  $K = 2 \times 10^9$  dyne/cm<sup>3</sup> and  $R = 4 \times 10^3$  dyne s/cm<sup>3</sup>.

It is known that the vesicle in a simple shear flow undergoes a tank-treading tangential motion at its equilibrium configuration. The right panel of Fig. 2 shows a simulation result in which the vesicle configuration and the velocity field on the vesicle boundary are drawn at time  $t = 0.04$ . The top panel of Fig. 3 shows a more detailed motion of the single vesicle including the streamlines of the velocity field at certain times. The surface tensions  $\sigma(s, t)$  at the corresponding times, which are computed counterclockwise along the vesicle boundary starting at a boundary point on the long axis, are plotted in the bottom panel. The surface tension also reaches the equilibrium state and remains almost unchanged as time evolves.

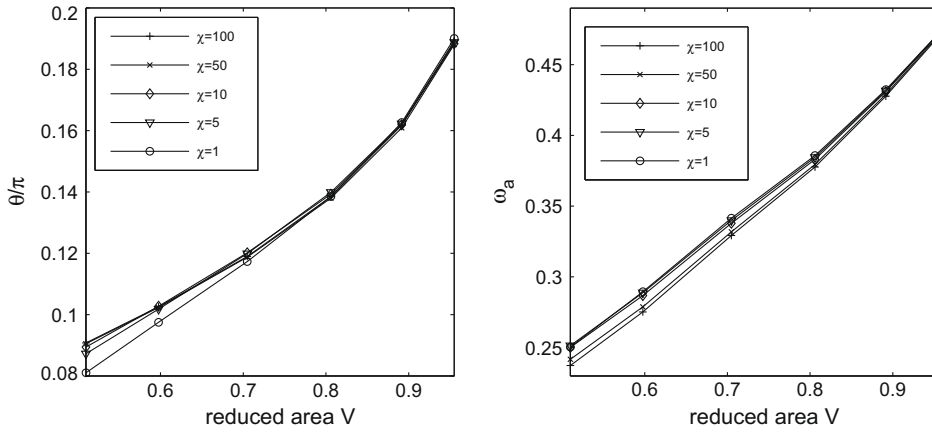
Notice that, in Fig. 3, we have observed that there are some sawtooth oscillations for the surface tension along the vesicle boundary at  $t = 0.036$  when the vesicle configuration is steady. This oscillatory behavior has also been found in previous study [38] where the point-wise collocation method is applied. In order to remove the oscillations, one might apply the numerical smoothing so that the high frequency modes can be eliminated. Here, however, these sawtooth oscillations are generally low amplitudes, and the surface tension remains with sufficient accuracy as long as the vesicle maintains a compact shape. The convergence study shown below strongly suggests that the pIB method is convergent.

The motion of a steady state vesicle can be characterized by both the inclination angle  $\theta$  between the long axis of the vesicle and the flow direction, and the frequency  $\omega$  of the revolution. These two values have been found to be strongly dependent on the reduced area  $V$  but independent of the dimensionless shear rate  $\chi$  [18,36]. This is verified in Fig. 4 which shows the inclination angle (left) and the revolution angular velocity (right) in terms of reduced area when the vesicles are at their steady shapes. We observe that both the inclination angle and angular velocity increase with the reduced area  $V$  but are nearly independent of the shear rate  $\chi$ , which confirms the observation from previous results in the literature [18,36].

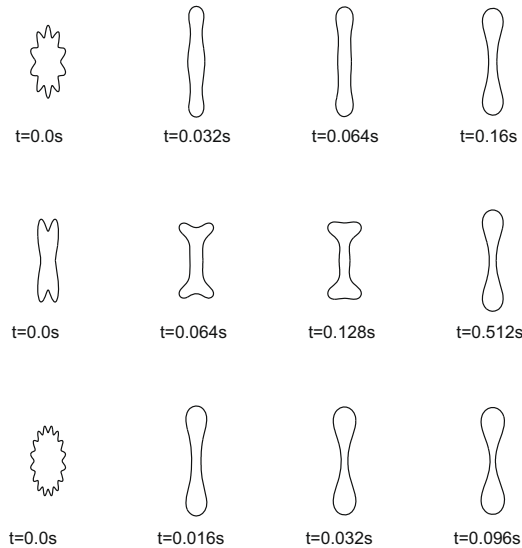
Fig. 5 shows the shape dynamics of a vesicle suspended freely in a quiescent flow. In the absence of the inextensibility constraint, the equilibrium shape of a vesicle is a circle. However, the equilibrium shape of an inextensible vesicle can be different from a circle. It is also known that the equilibrium shape is dependent only on the reduced area  $V$  of the vesicle but independent of the material properties of the vesicle and the surrounding fluid [31,36]. Fig. 5 shows the motion of vesicles which have three different initial shapes (see the first column) but with the same reduced area  $V = 0.33$ . The first row has  $\mu = 0.01$  g/(cm s) and  $c_b = 10^{-10}$  dyne cm, the second row has  $\mu = 0.01$ g/(cm s) and  $c_b = 10^{-11}$  dyne cm, and the third



**Fig. 3.** The motion of a single vesicle and streamlines (top) and the surface tension  $\sigma(s, t)$  (bottom) along the vesicle at times 0.004, 0.012, and 0.036. The reduced area  $V = 0.51$ , and the shear rate  $\chi = 250$ . We can observe that, once the vesicle reaches its steady configuration, it goes through a tank-treading tangential motion.



**Fig. 4.** The inclination angles (left) and the angular velocities (right) of vesicles with different reduced areas in shear flows with different shear rates. We set the angular velocity  $\omega_a = \frac{\sqrt{2}\bar{\omega}}{\gamma\kappa_0}$ , where  $\bar{\omega}$  is the mean angular velocity over the vesicle boundary at the equilibrium configuration. We observe that both  $\theta$  and  $\omega_a$  are nearly independent of the dimensionless shear rate. This result is in agreement with [18,36].



**Fig. 5.** The motion of three vesicles of the same reduced area with different bending rigidities and different fluid viscosities at some chosen times. The equilibrium shapes of the three vesicles are almost the same, i.e., it is independent of fluid viscosity and the material property.

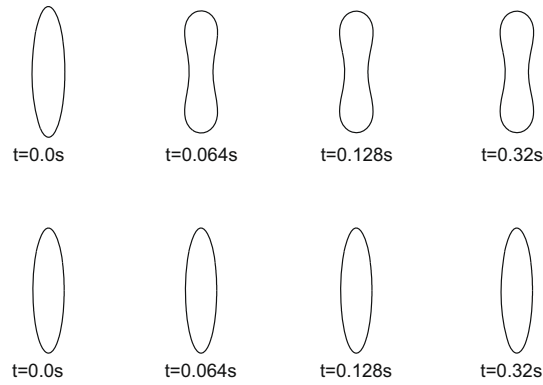
row has  $\mu = 0.001 \text{ g}/(\text{cm s})$  and  $c_b = 10^{-10} \text{ dyne cm}$ . We can observe that, independent of the bending rigidity of the vesicle and the fluid viscosity, the three vesicles have almost the same equilibrium shape. The only difference is the time elapsed to reach the equilibrium configuration.

Even though a vesicle in a quiescent flow has its equilibrium shape similar to those in Fig. 5 depending on the reduced area  $V$ , the equilibrium shape could be an ellipse, if the initial shape is an ellipse with a small aspect ratio and the bending rigidity is small [33]. We verify this fact from Fig. 6 which shows the motion of vesicle with a shape of an ellipse with the aspect ratio 0.25. We use two different bending coefficients:  $c_b = 10^{-10} \text{ dyne cm}$  (top) and  $10^{-14} \text{ dyne cm}$  (bottom). Whereas the ellipse with a larger bending rigidity changes into a shape similar to those in Fig. 5 (top), the ellipse with a smaller bending rigidity does not change its initial shape (bottom).

4.2. Convergence study

In order to provide further verification that the pIB method correctly solves the inextensible vesicle dynamics, we perform a convergence study. We consider a single vesicle in a shear flow with shear rates  $\chi = 10$  and  $\chi = 250$ , see Fig. 2. Now we choose the mesh sizes of the domain  $N = 128, 256, 512,$  and  $1024$  so that the corresponding mesh width is  $h = D/N$ . We also

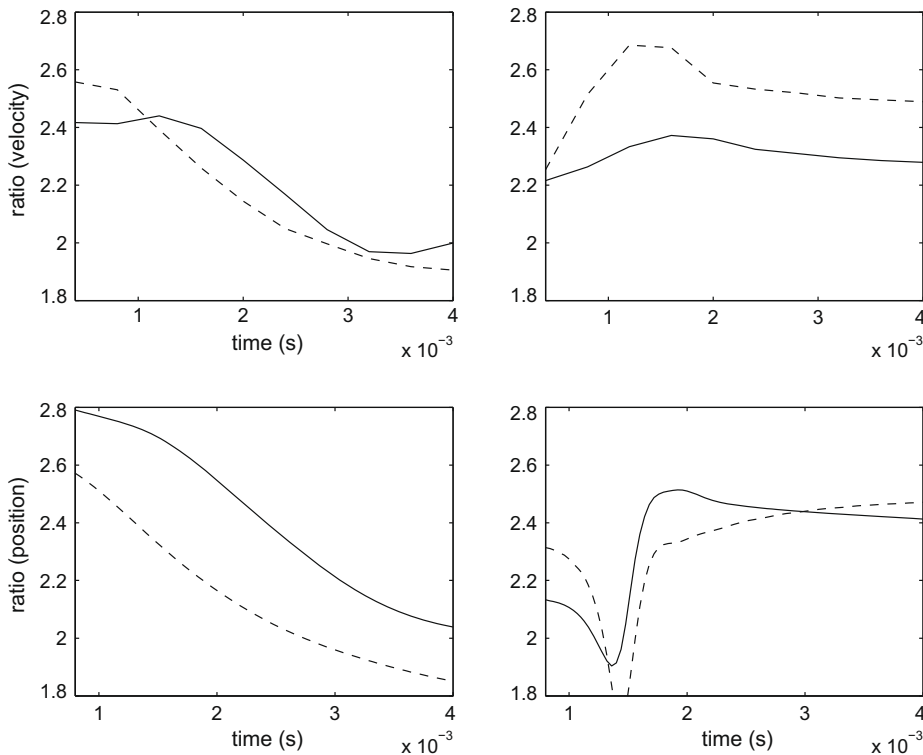




**Fig. 6.** The motion of vesicles with different bending rigidity. The initial ellipse has the aspect ratio 0.25. Whereas the ellipse with a larger bending rigidity  $c_b = 10^{-10}$  dyne cm changes its shape (top), the ellipse with a smaller one  $c_b = 10^{-14}$  dyne cm keeps its initial shape (bottom).

choose  $\Delta s$  and  $\Delta t$  proportional to  $h$ , so that each factor of two in refinement of the fluid mesh width is accompanied by the same factor of refinement for the boundary mesh and the time step duration. The penalty parameters are defined as  $K = 10^4$  N dyne/cm<sup>3</sup> and  $R = 0.005$  N dyne s/cm<sup>3</sup>. i.e., both parameters increase as the mesh width and time step are refined. In order to make both penalty force terms in Eq. (10) equally important, the magnitude of  $K$  is obtained by scaling a time step factor of  $R$ . As we shall see in Fig. 7, the distance between two represented markers  $\mathbf{X}$  and  $\mathbf{Y}$  gets smaller when the grid is refined (the penalty parameters  $K$  and  $R$  thus get larger).

Since the vesicle boundary is inextensible and the fluid is incompressible, the perimeter of the vesicle and the enclosed area should remain as constants as time goes on. Let  $L_0(\mathbf{X})$  and  $L_T(\mathbf{X})$  be the perimeters of the vesicle boundary  $\mathbf{X}$  at the initial time and final time  $T = 0.04$ , respectively. The relative error of the perimeter is defined as  $|L_T(\mathbf{X}) - L_0(\mathbf{X})|/L_0(\mathbf{X})$ , and the



**Fig. 7.** Convergence ratios of the computed velocity field  $\mathbf{u}(\mathbf{x}, t)$  of the fluid (top) and the configuration of the vesicle boundary  $\mathbf{X}(s, t)$  (bottom). The convergence ratios (defined in the text) are plotted as functions of time for two different cases:  $\chi = 10$  (left) and  $\chi = 250$  (right). In each panel, the dashed line is the convergence ratio obtained using the grids  $N = 128, 256, 512$ , and the solid line is the ratio obtained with  $N = 256, 512, 1024$ . The convergence ratio is nearly two (first-order accuracy) for all the different cases.

relative error of the area is  $|A_T(\mathbf{X}) - A_0(\mathbf{X})|/A_0(\mathbf{X})$  where the functional  $A$  assigns to the vesicle  $\mathbf{X}$  its enclosed area. Table 1 shows the relative errors of the perimeter and the area of the region bounded by the vesicle boundary in the two cases:  $\chi = 10$  (left) and  $\chi = 250$  (right). We observe that the two errors in both cases indicate first-order convergence. Besides, those errors are reasonably small and comparable to the results in [36].

Fig. 7 shows the convergence ratios of the computed fluid velocity  $\mathbf{u}(\mathbf{x}, t)$  and the vesicle boundary  $\mathbf{X}(s, t)$ . Since we do not know the exact solution of the problem, the estimation of the convergence ratio requires three numerical solutions for three consecutive  $N$ 's. Let  $(u_N, v_N)$  be the velocity field, and let  $\|\cdot\|_2$  be  $L_2$  norm. The top panels of Fig. 7 shows the convergence ratios  $(\|u_N - u_{2N}\|_2^2 + \|v_N - v_{2N}\|_2^2)^{1/2} / (\|u_{2N} - u_{4N}\|_2^2 + \|v_{2N} - v_{4N}\|_2^2)^{1/2}$  versus time for each of the cases  $N = 128$  (dotted line) and 256 (solid line). The bottom panels of Fig. 7 shows the corresponding convergence ratios of the configuration of the vesicle boundary  $\mathbf{X}(s, t)$  versus time. (Note that, the error estimation is calculated in a similar manner to that of the fluid velocity above.) One can see from this figure that the convergence ratio for the fluid velocity and the vesicle boundary are all around two, which indicates that the present method is first-order accurate. This is the typical behavior in accuracy for the IB method as applied to problems with thin elastic boundaries (for second-order IB method in the case of an immersed elastic structure of finite thickness, see [6,14]).

#### 4.3. Multiple vesicles in a shear flow

As in [36], we now consider the case of multiple vesicles suspended in a shear flow. Since the computational time of our method is mostly consumed by solving the fluid equations, and there is no direct interaction between two different vesicles (the interaction occurs only through the fluid equations), the total computational cost is almost independent of the number of vesicles. However, since the increase of the number of vesicles simulated requires to increase the size of the computational domain, in order to solve the fluid equations with the same resolution as in the case of one vesicle, the computing time will still increase proportionally to the size of the computational domain.

We first consider the dynamics of four vesicles under a shear flow in a fluid domain  $2D \times D$ . Fig. 8 shows the motion of the four vesicles including streamlines of the velocity field at some chosen times. The shear rate is  $\chi = 250$  as in [36]. The group of vesicles turns together like one single vesicle initially (from 0.016 to 0.024), and then they start to separate (from 0.04 to 0.048). Once they separate, the shear flow will drive them farther away from each other. The result is consistent with the one obtained in [36].

We now simulate the dynamics of a larger number (56) of interacting vesicles. As explained before, the computing time of our method is dependent mostly on the number of spatial grid. We use the domain  $(0, 4D) \times (0, 4D)$  which is sixteen times larger than the case shown in Fig. 2 to get the same resolution. This amount of the increasing domain requires about 64 times more computing time than the single vesicle case. The cost of our method is comparable to that of the boundary integral method incorporating fast multiple method which is  $O(mn \log m)$ , where  $n$  is the number of vesicles and  $m$  is the number of the marker points on each vesicle [36].

Fig. 9 shows the motion of the vesicles at several times. While the group of vesicles behaves like a single vesicle and moves following the shear flow, each vesicle changes its shape into a relaxed one. Note that, we allow the vesicles to move out of the domain, in which case we should handle the data outside the domain by duplicating them into the domain in a periodic way, see the vesicles leaving the left (or right) side of the domain and coming into the right (or left) side of the domain.

#### 4.4. Binary-component vesicle

As a final example, we consider a binary-component vesicle suspended freely in a viscous fluid. The binary-component vesicle is also inextensible and the bending rigidity and the spontaneous curvature of the vesicle boundary are not constants, but depend on the concentration of the different lipid phases. We consider the two component system with mass density  $\rho_A$  and  $\rho_B$  for components A and B, respectively. The mass concentration variable is defined as  $\phi = \rho_B / (\rho_A + \rho_B)$  such that  $\phi = 0$  and  $\phi = 1$  represent the component A and B, respectively. In practice, those two components correspond to liquid-ordered and liquid-disordered surface phases [22]. Thus, the bending energy for this two component system in Eq. (8) is given by

$$E_b = \int_{\Gamma} \frac{c_b(\phi)}{2} (\kappa - \kappa_0(\phi))^2 ds, \tag{33}$$

**Table 1**

The relative errors of the perimeter and the enclosed area measured at time 0.04. The left table is for  $\chi = 10$  and the right table is for  $\chi = 250$ .

N	$\chi = 10$		$\chi = 250$	
	$ L_T - L_0 /L_0(\mathbf{X})$	$ A_T - A_0 /A_0(\mathbf{X})$	$ L_T - L_0 /L_0(\mathbf{X})$	$ A_T - A_0 /A_0(\mathbf{X})$
128	1.13e-6	3.14e-5	1.52e-5	5.61e-5
256	7.12e-7	6.21e-6	1.72e-5	3.64e-5
512	4.05e-7	4.61e-7	1.03e-5	1.36e-5
1024	2.21e-7	2.17e-7	5.47e-6	5.38e-6

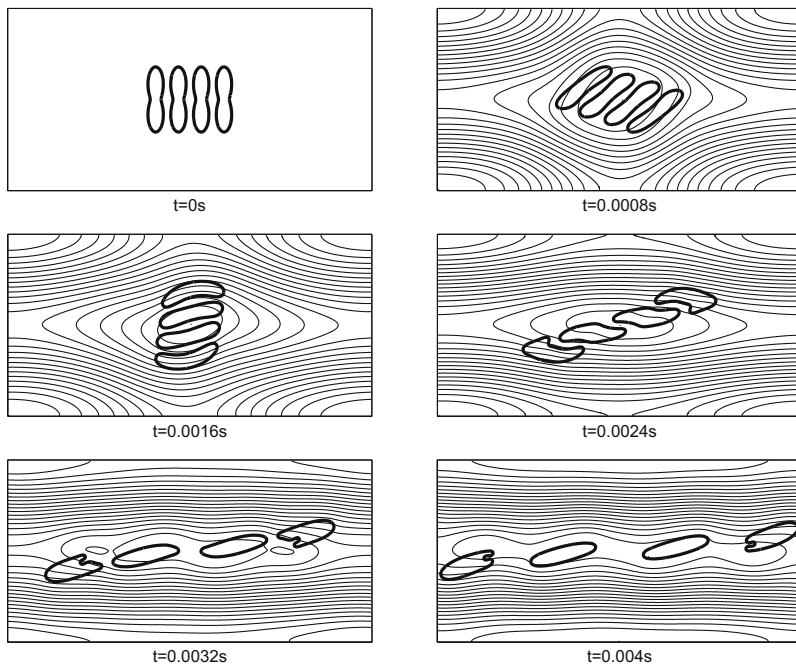


Fig. 8. The motion of four vesicles in a shear flow at some chosen times with streamlines depicting the velocity field.

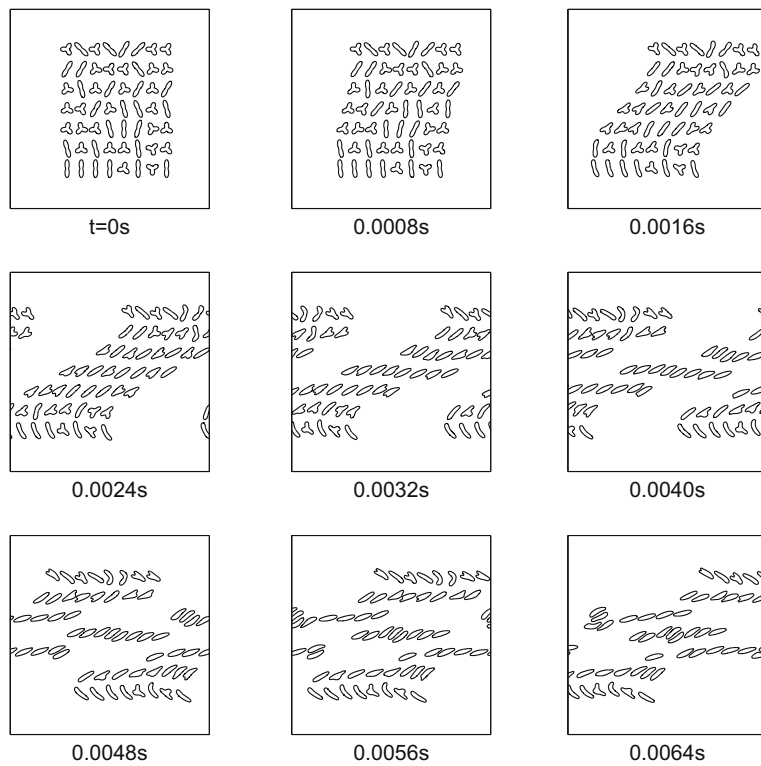


Fig. 9. The motion of 56 vesicles in a shear flow at some chosen times.

where  $\kappa = \left| \frac{\partial^2 \mathbf{X}}{\partial s^2} \right|$  is the curvature, and  $\kappa_0$  is the spontaneous curvature which represents the asymmetry nature of the membrane. Note that here we use the arc-length parameter  $s$ . Taking the variational derivative of the above modified bending energy, we obtain the following bending force

$$\mathbf{F}_b = -\frac{\delta E_b}{\delta \mathbf{X}} = \frac{\partial^2}{\partial s^2} \left( c_b(\phi) \left( \frac{\kappa_0(\phi)}{\left| \frac{\partial^2 \mathbf{X}}{\partial s^2} \right|} - 1 \right) \frac{\partial^2 \mathbf{X}}{\partial s^2} \right) \tag{34}$$

One can easily see that when the case of constant  $c_b$  and zero spontaneous curvature  $\kappa_0 = 0$ , the above bending force recovers to the original one in Eq. (8). So the total force  $\mathbf{F}$  in Eq. (5) now becomes

$$\mathbf{F}(s, t) = \frac{\partial}{\partial s} (\sigma(s, t) \boldsymbol{\tau}(s, t)) + \frac{\partial^2}{\partial s^2} \left( c_b(\phi) \left( \frac{\kappa_0(\phi)}{\left| \frac{\partial^2 \mathbf{X}}{\partial s^2} \right|} - 1 \right) \frac{\partial^2 \mathbf{X}}{\partial s^2} \right). \tag{35}$$

Since the phase concentration  $\phi$  affects the bending rigidity  $c_b(\phi)$  and the spontaneous curvature  $\kappa_0(\phi)$  of the membrane, in this work, we simply adopt the same idea used in interfacial flow in which the surfactant concentration affects the magnitude of surface tension. Since the surfactant stays in the interface (insoluble), one has to solve the surfactant equation along the interface [20,35]. Here, the order parameter  $\phi$  plays a similar role as the surfactant in surface tension driven flows. Thus, one needs to solve the phase concentration equation on the vesicle boundary  $\Gamma$ . In order to separate those two components on the vesicle boundary, we adopt the Cahn–Hilliard type continuum approach [1], meaning that, the time evolution of  $\phi(s, t)$  is governed by

$$\frac{\partial \phi}{\partial t} = M_\phi \Delta_s \psi \tag{36}$$

$$\psi = \frac{1}{\varepsilon} G'(\phi) - \varepsilon \Delta_s \psi, \tag{37}$$

where the surface Laplacian is defined as  $\Delta_s = \frac{\partial}{\partial s} \left( \frac{\partial}{\partial s} / \left| \frac{\partial \mathbf{X}}{\partial s} \right| \right) / \left| \frac{\partial \mathbf{X}}{\partial s} \right|$ . Here,  $\psi$  represents the chemical potential. Notice that, a more complicated chemical potential form has been derived in [32] to preserve the thermodynamically consistent relationship between the mass flux term and the force term. The function  $G(\phi) = \frac{1}{4} \phi^2 (1 - \phi)^2$  is a double well potential that describes the tendency of the phase concentration  $\phi$  into the two stable phase separation  $\phi = 0$  or  $\phi = 1$ , and  $G'(\phi) = \phi(\phi - 1)(\phi - 1/2)$ . The constant  $M_\phi$  is the mobility and  $\varepsilon$  is a small parameter that effectively describes the thickness of the transitional layer on the membrane that separates the components A and B.

To apply pIB method to the binary-component vesicle problem, we use two representations  $\mathbf{X}$  and  $\mathbf{Y}$  of the vesicle boundary and we apply the same idea to the binary-component vesicle case by using  $\mathbf{Y}$  instead of  $\mathbf{X}$  in the above equations and obtain

$$\frac{\partial \phi}{\partial t} = M_\phi \frac{\partial}{\partial s} \left( \frac{\partial \psi}{\partial s} / \left| \frac{\partial \mathbf{Y}}{\partial s} \right| \right) / \left| \frac{\partial \mathbf{Y}}{\partial s} \right|, \tag{38}$$

$$\psi = \frac{1}{\varepsilon} G'(\phi) - \varepsilon \frac{\partial}{\partial s} \left( \frac{\partial \phi}{\partial s} / \left| \frac{\partial \mathbf{Y}}{\partial s} \right| \right) / \left| \frac{\partial \mathbf{Y}}{\partial s} \right|, \tag{39}$$

$$K(\mathbf{Y}(s, t) - \mathbf{X}(s, t)) + R(\mathbf{V}(s, t) - \mathbf{U}(s, t)) = \frac{\partial}{\partial s} (\sigma(s, t) \boldsymbol{\tau}(s, t)) + \frac{\partial^2}{\partial s^2} \left( c_b(\phi) \left( \frac{\kappa_0(\phi)}{\left| \frac{\partial^2 \mathbf{Y}}{\partial s^2} \right|} - 1 \right) \frac{\partial^2 \mathbf{Y}}{\partial s^2} \right). \tag{40}$$

Note that, Eqs. (38) and (39) are additional equations to solve in the governing system, and Eq. (40) replaces Eq. (11).

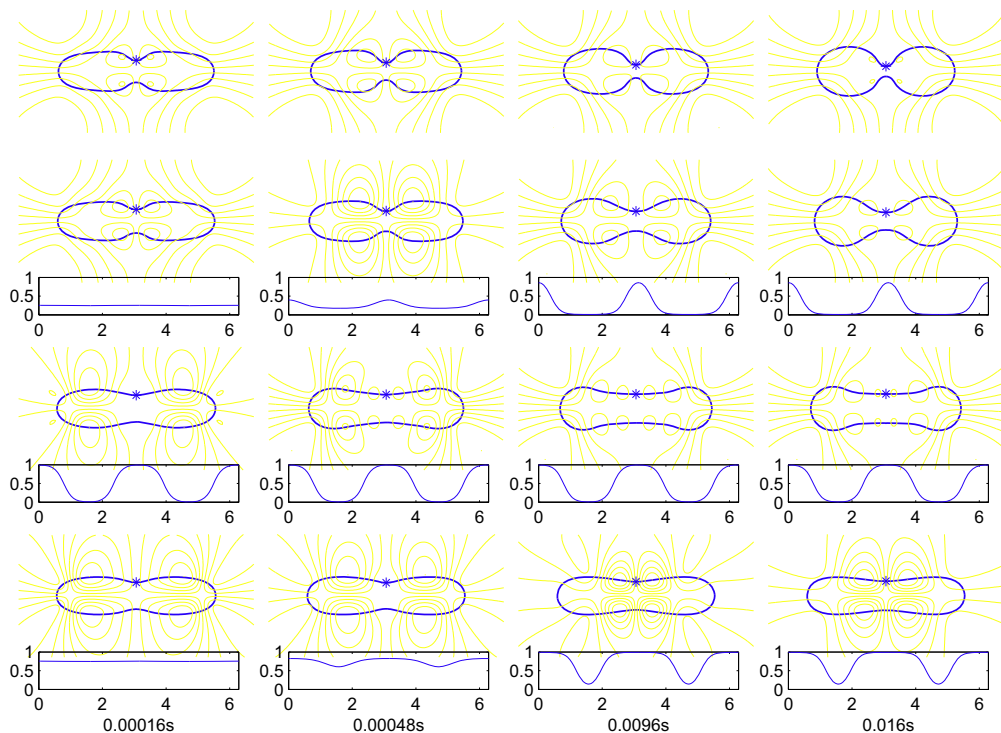
For Eqs. (38) and (39), since the vesicle boundary  $\mathbf{Y}$  is inextensible, we use the initial stretching factor  $\left| \frac{\partial \mathbf{Y}^0}{\partial s} \right|$  for  $\left| \frac{\partial \mathbf{Y}^n}{\partial s} \right|$  in all the time level  $n$  and discretize Eqs. (38) and (39) to obtain

$$\phi^{n+1} = \phi^n + \Delta t M_\phi \frac{\partial}{\partial s} \left( \frac{\partial \psi^{n+1}}{\partial s} / \left| \frac{\partial \mathbf{Y}^0}{\partial s} \right| \right) / \left| \frac{\partial \mathbf{Y}^0}{\partial s} \right|, \tag{41}$$

$$\psi^{n+1} = \frac{1}{\varepsilon} G'(\phi^n) - \varepsilon \frac{\partial}{\partial s} \left( \frac{\partial \phi^{n+1}}{\partial s} / \left| \frac{\partial \mathbf{Y}^0}{\partial s} \right| \right) / \left| \frac{\partial \mathbf{Y}^0}{\partial s} \right|. \tag{42}$$

The spatial discretization of these two equations induces a linear system for  $\phi^{n+1}$  and  $\psi^{n+1}$ . Since  $\left| \frac{\partial \mathbf{Y}^0}{\partial s} \right|$  is initially defined and does not change in time, we only need to invert the resulting matrix once and save it for the later usage. Once we obtain the new  $\phi^{n+1}$ , we can substitute it into the functions of  $c_b(\phi^{n+1})$  and  $\kappa_0(\phi^{n+1})$  so that Eq. (40) can be solved numerically using the same form of discretization as before where  $\mathbf{Y}$  in the right-hand side is treated explicitly.

In our tests, we choose  $M_\phi = 10^{-6}$ , and  $\varepsilon = 10^{-4}$ . The spontaneous curvature is  $\kappa_0(\phi) = \frac{1}{R_0} (5(1 - \phi) + 0.1\phi)$ , where the length scale  $R_0$  is  $10^{-3}$ , and the bending rigidity is  $c_b(\phi) = c_0((1 - \phi) + 0.5\phi)$ , where the coefficient  $c_0$  is  $10^{-10}$ . Notice that, the above special form of  $c_b$  and  $\kappa_0$  are adopted from [32] except for some scaling factors. The initial shape of the binary-component vesicle is the same as the one shown in Fig. 2. As in [32], the initial concentration  $\phi$  is a mixture of two phases with the following perturbed form:



**Fig. 10.** The motion of binary-component vesicles suspended freely with streamlines depicting the velocity field and the concentration of the lipid phase  $\phi(s, t)$  at some chosen times. Four different concentration cases: constant concentration  $\phi = 0$  (first row),  $\bar{\phi} = 0.25$  (2nd–3th rows),  $\bar{\phi} = 0.5$  (4th–5th rows) and  $\bar{\phi} = 0.75$  (bottom two rows).

$$\phi(s, 0) = \bar{\phi} + 0.001(\cos(2s) + \cos(4s) + \cos(6s)), \quad (43)$$

where  $\bar{\phi}$  is the average concentration.

Fig. 10 shows the motion of the binary-component vesicle suspended freely with streamlines depicting the velocity field and the concentration of the lipid phase  $\phi(s, t)$  at some chosen times. The cases of four different average concentrations have been simulated and ordered from top to bottom row as  $\phi = 0$  (first row), and  $\bar{\phi} = 0.25, 0.5$ , and  $\bar{\phi} = 0.75$  (last two rows). Notice that, the case of  $\phi = 0$  (first row) shows the motion of a single component vesicle which has a constant concentration  $\phi = 0$  all the time. The concentration of the lipid phase  $\phi(s, t)$  is drawn in terms of  $s$  starting at the point marked by \* on each vesicle and in the counterclockwise way along the vesicle.

One can see that, as time evolves, the phase separation occurs and yields two larger regions: one region of  $\phi \approx 1$  at the vesicle top and bottom parts where the vesicle boundary has smaller curvature, and the other region  $\phi \approx 0$  phase at the two tips of the vesicle where the vesicle boundary has larger curvature. This is exactly what we expect due to the form of spontaneous curvature. One can conclude from the figure that, as the average concentration gets lower, the curvature of the vesicle gets larger due to the spontaneous curvature. That is, when the average concentration  $\bar{\phi}$  becomes larger, the vesicle becomes flatter in general.

## 5. Summary and conclusions

We have developed a penalty immersed boundary (pIB) method to simulate the dynamics of inextensible vesicle interacting with an incompressible fluid. In order to take into account the inextensibility constraint of the vesicle, we virtually separate the dynamics of the inextensible vesicle from the fluid equations using the penalty force, and then solve the system of equations governing the motion of the inextensible boundary. In this separate system for the vesicle motion, the inextensibility constraint simply produces a simple one-dimensional second-order elliptic equation for the unknown surface tension.

We have validated the method by simulating the motion of a single vesicle both in a shear flow and in a quiescent flow. We have shown that the vesicle in a simple shear flow undergoes a tank-treading tangential motion at its equilibrium state, and that its inclination angle  $\theta$  and frequency  $\omega$  of the revolution are strongly dependent on the reduced area  $V$  but independent of the shear rate  $\chi$ , as shown in other literature. The fact that the shape of a vesicle in a quiescent flow depends only on the reduced area  $V$  of the vesicle but independent of the material properties of the interface and surrounding fluid

has also been verified. Additional validation has been provided in the form of a convergence study, which confirms the expected first-order accuracy of the scheme.

For multiple vesicle cases, the total computing time is dependent mostly on the number of the computational grids but is almost independent of the number of vesicles. The cost of our method is comparable to that of other numerical methods such as the boundary integral method. Moreover, we have also extended the method to solve a simplified binary-component vesicle model.

Future work will include the extension of the methodology introduced here to the study of three-dimensional vesicle dynamics. The application of the present method to some biological systems such as red blood cells and drug-carrying capsules will also be the subject of future work. Unlike the boundary integral method, the present method can be applied to the high Reynolds number flow in which the inertial effect cannot be neglected.

## Acknowledgment

Y Kim is supported by the Chung-Ang University Research Grants in 2010. M.-C. Lai is supported in part by National Science Council of Taiwan under research Grant NSC-97-2628-M-009-007-MY3 and MoE-ATU project.

## References

- [1] J.W. Cahn, J.E. Hilliard, Free energy of a nonuniform system I interfacial free energy, *J. Chem. Phys.* 28 (1958) 258–267.
- [2] C. Duncan, G. Zhai, R. Scherer, Modeling coupled aerodynamics and vocal fold dynamics using immersed boundary methods, *J. Acoust. Soc. Am.* 120 (2006) 2859–2871.
- [3] L.J. Fauci, C.S. Peskin, A computational model of aquatic animal locomotion, *J. Comput. Phys.* 77 (1988) 85–108.
- [4] A.L. Fogelson, C.S. Peskin, A fast numerical method for solving the three-dimensional Stokes equations in the presence of suspended particles, *J. Comput. Phys.* 79 (1988) 50–69.
- [5] D. Goldstein, R. Handler, L. Sirovich, Modeling a no-slip flow with an external force field, *J. Comput. Phys.* 105 (1993) 354–366.
- [6] B.E. Griffith, 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.
- [7] K.H. de Haas, C. Blom, D. van den Ende, M.H.G. Duits, J. Mellema, Deformation of giant lipid bilayer vesicles in shear flow, *Phys. Rev. E* 56 (1997).
- [8] W. Helfrich, Elastic properties of lipid bilayers: theory and possible experiments, *Z. Naturforsch C* 28 (1973) 693–703.
- [9] J.J. Heys, B. Knott, T. Gedeon, Y. Kim, Modeling arthropod filiform hair motion using the penalty immersed boundary method, *J. Biomech. Eng.* 41 (2008) 977–984.
- [10] W. Huang, S. Shin, H. Sung, Simulation of flexible filaments in a uniform flow by the immersed boundary method, *J. Comput. Phys.* 226 (2007) 2206–2228.
- [11] E. Jung, C.S. Peskin, Two-dimensional simulations of valveless pumping using the immersed boundary method, *SIAM J. Sci. Comput.* 23 (2001) 19–45.
- [12] V. Kantsler, V. Steinberg, Orientation and dynamics of a vesicle in tank-treading motion in shear flow, *Phys. Rev. Lett.* 95 (2005).
- [13] Y. Kim, C.S. Peskin, 2-D parachute simulation by the immersed boundary method, *SIAM J. Sci. Comput.* 28 (6) (2006).
- [14] Y. Kim, C.S. Peskin, Penalty immersed boundary method with an elastic boundary with mass, *Phys. Fluids* 19 (5) (2007).
- [15] Y. Kim, C.S. Peskin, Numerical study of incompressible fluid dynamics with nonuniform density by the immersed boundary method, *Phys. Fluids* 20 (2008) 062101.
- [16] Y. Kim, S. Lim, A. Friedman, S. Raman, O. Simonetti, Blood flow in a compliant vessel by the immersed boundary method, *Ann. Biomed. Eng.* 37 (5) (2009).
- [17] Y. Kim, C.S. Peskin, 3-D parachute simulations by the immersed boundary method, *Comput. Fluids* 38 (2009) 1080–1090.
- [18] M. Kraus, W. Wintz, U. Seifert, R. Lipowsky, Fluid vesicles in shear flow, *Phys. Rev. Lett.* 77 (1996).
- [19] M.-C. Lai, C.S. Peskin, An immersed boundary method with formal second-order accuracy and reduced numerical viscosity, *J. Comput. Phys.* 160 (2000) 705–719.
- [20] M.-C. Lai, Y.-H. Tseng, H. Huang, An immersed boundary method for interfacial flows with insoluble surfactant, *J. Comput. Phys.* 227 (2008) 7279–7293.
- [21] C. Lee, Stability characteristics of the virtual boundary method in three-dimensional applications, *J. Comput. Phys.* 184 (2003) 559–591.
- [22] J.S. Lowengrub, A. Ratz, A. Voigt, Phase-field modeling of the dynamics of multi-component vesicles: spinodal decomposition, coarsening, budding and fission, *Phys. Rev. E* 79 (2009) 031926.
- [23] A.J. Markvoort, R.A. Van Santen, P.A.J. Hilbers, Vesicle shapes from molecular dynamics simulations, *J. Phys. Chem. B* 110 (2006).
- [24] H. Noguchi, G. Gompper, Shape transitions of fluid vesicles and red blood cells in capillary flows, *Proc. Natl. Acad. Sci.* 102 (2005).
- [25] C.S. Peskin, The immersed boundary method, *Acta Numer.* 11 (2002) 479–517.
- [26] C.S. Peskin, D.M. McQueen, Three dimensional computational method for flow in the heart: immersed elastic fibers in a viscous incompressible fluid, *J. Comput. Phys.* 81 (1989) 372–405.
- [27] C.S. Peskin, D.M. McQueen, Fluid dynamics of the heart and its valves, in: *Case Studies Math. Model.: Ecol. Physiol. Cell Biol.*, Prentice-Hall, Englewood Cliffs, NJ, 1996, pp. 309–337.
- [28] C.S. Peskin, B.F. Printz, Improved volume conservation in the computation of flows with immersed elastic boundaries, *J. Comput. Phys.* 105 (1993) 33–46.
- [29] C. Pozrikidis, The axis-symmetric deformation of a red blood cell in uniaxial straining Stokes flow, *J. Fluid Mech.* 216 (1990) 231–254.
- [30] E.M. Saiki, S. Biringen, Numerical simulation of a cylinder in uniform flow: application of a virtual boundary method, *J. Comput. Phys.* 123 (1996) 450–465.
- [31] U. Seifert, Adhesion of vesicles in two-dimensions, *Phys. Rev. A* 43 (1991).
- [32] J.S. Sohn, Y.-H. Tseng, S. Li, A. Voigt, J.S. Lowengrub, Dynamics of multi-component vesicles in a viscous fluid, *J. Comput. Phys.* 229 (2010) 119–144.
- [33] P. Song, Numerical Simulations of Cylindrical Membranes, Ph.D. Thesis, Peking University, 2008.
- [34] M.J. Stevens, Coarse-grained simulations of lipid bilayers, *J. Chem. Phys.* 121 (2004).
- [35] H.A. Stone, A simple derivation of the time-dependent convective–diffusion equation for surfactant transport along a deforming interface, *Phys. Fluids A* 2 (1) (1990) 111–112.
- [36] 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.
- [37] S.K. Veerapaneni, D. Gueyffier, G. Biros, D. Zorin, A numerical method for simulating the dynamics of 3D axis-symmetric vesicles suspended in viscous flows, *J. Comput. Phys.* 228 (2009) 7233–7249.
- [38] H. Zhou, C. Pozrikidis, Deformation of liquid capsules with incompressible interfaces in simple shear flow, *J. Fluid Mech.* 283 (1995) 175–200.