A numerical method for simulating the dynamics of 3D axisymmetric vesicles suspended in viscous flows

February 25, 2009

Abstract

We extend the efficient high-order method of [Veerapaneni et al., 2008] to the axisymmetric flows with immersed vesicles of spherical or toroidal topology. In this case, the bending and fluid forces require a siginficantly different (for bending forces, nonlinear, vs. linear in arclength in the 2D case) computation. The qualitative numerical behavior of the problem is also different: with a nonlinear implicit scheme needed to eliminate the CFL-type restriction in the toroidal case. We present an unconditionally stable scheme with low cost per time step, and spectrally accurate in space and third-order accurate in time. We also present a novel numerical scheme for evaluation of the 3D Stokes single-layer potential on an axisymmetric surface. As an application, we explore the motion of axisymmetric vesicles under gravity.

1 Introduction

Vesicles are closed lipid membranes suspended in a viscous solution. They are common in biological systems, and play an important role in intracellular and intercellular transport; artificial vesicles are used in a variety of drug-delivery systems. The vesicle evolution dynamics are characterized by a competition between membrane elastic energy, nonlinearity, surface inextensibility, and non-local interactions due to the hydrodynamic coupling. The design of efficient computational methods for such flows has received relatively limited attention, compared to other types of particulate flows, due to the difficulty in simulation of large number of deformable vesicles. In [25], we introduced a new algorithm for vesicle simulations in two dimensions. In this paper, we take the first step towards efficient high-order three-dimensional simulations by considering axisymmetric vesicle flows for the case where there is no viscosity contrast across the vesicle membrane. The equations that govern the motion of a single vesicle in three dimensions are

$$\frac{\partial \mathbf{x}}{\partial t} = \mathbf{v}_\infty + \mathcal{S}[\mathbf{f}_b + \mathbf{f}_\sigma] \quad \text{(vesicle position evolution)},$$

$$\operatorname{div}_\gamma \frac{\partial \mathbf{x}}{\partial t} = 0 \quad \text{(surface inextensibility)},$$

(1)

where γ is the vesicle membrane, div_γ is the surface divergence operator, $\mathbf{x}$ is a point on γ, $\mathbf{f}_\sigma$ is a force (tension) due to surface inextensibility, $\mathbf{f}_b$ is a force due to bending, $\mathbf{v}_\infty$ is the far-field velocity of the bulk fluid, and $\mathcal{S}$ is the single-layer potential Stokes operator, defined in § 2. The first equation describes the motion of the vesicle boundary; the second equation expresses the local inextensibility of γ.

Our main goal is to extend the ideas presented in [25]to the axisymmetric case of vesicles with spherical or toroidal topology. The extension is non-trivial because in three dimensions the bending energy has a much more complicated form, and cannot be reduced to a linear expression in arc-length derivatives as in the two-dimensional case. Furthermore, the qualitative numerical behavior of bending forces is also different: an unconditionally stable semi-implicit linearized scheme with no CFL-type restriction on the time step, similar to the two-dimensional case, could only be found for the spherical topology. For vesicles with toroidal topology (admittedly less common, but observed in practice [18]), eliminating CFL-type time-step restrictions requires a nonlinear iteration. Our main contributions is two schemes for solving the system of equations (1) for spherical and toroidal topologies, with the following properties:

- a single linear solve per time step is need for spherical topology, and a small number of iterations for toroidal topology;

- each time-step solve has nearly constant cost per degree of freedom;

- the scheme is spectrally accurate in space and third-order accurate in time.

An important part of the algorithm is a novel numerical scheme for evaluation of the 3D Stokes single-layer potential on an axisymmetric surface, needed to achieve an optimal complexity of the algorithm. As an application, we explore the motion of axisymmetric vesicles under gravity.

Limitations. The main limitations of our scheme are the lack of adaptivity (both in space and time) and the dependence of the stable time step on the shear rate in a shear flow. While one would hope that a fully implicit scheme would eliminate or reduce these time-step restrictions, our experiments indicate that even a fully-implicit Newton scheme (§4.3) does not yield noticeable improvements noticeable gains in the time when we were able to converge the Newton iterations. An additional limitation of the overall scheme is that we do not consider topology changes or vesicles flows with a viscosity contrast across the membrane, which would require solution of an additional boundary integral equation.

Related work. There has been a lot of work on modeling 3D axisymmetric particulate flows. In [25], we discussed vesicle-related algorithms. An excellent review of such methods can be found in [15] (Table 1, pg. 289; for vesicles see the "liquid capsules" entry).

Several groups have focused on determining stationary shapes of three-dimensional vesicles using semi-analytic [19, 4, 6], or numerical methods like the phase-field [9, 8] and membrane finite element methods

[10, 13]. These approaches are based on a constrained variational approach (i.e., minimizing the bending energy subject to area and volume constraints) and cannot be used for interactions of multiple vesicles in shear flows.

A full three-dimensional simulation of a single vesicle incorporating the hydrodynamic coupling, local inextensibility and the bending forces has been reported in [11, 20]. A closely related work is also that of [16], in which, a nearly inextensible interface was considered for the axisymmetric motion of red blood cells inside a cylindrical tube.

In all, however, little work has been done in developing fast algorithms for axisymmetric vesicle flows.

Contents. In §2, we present the integro-differential equations (1) that govern vesicle dynamics. The spatial and temporal discretizations are described in §3 and §4 respectively. In §5, we present numerical results for a number of problems involving single and multiple vesicles suspended in a viscous fluid. We conduct numerical experiments to investigate the stability and convergence order of different time-stepping schemes. Several important details (semi-analytic solutions for the quiescent case, expressions for the force and Stokes convolutions in the axisymmetric and approximation error for high-order derivatives) are presented in the appendix.

2 Problem Formulation

For simplicity, we first discuss the formulation for a single vesicle suspended in an unbounded viscous fluid. Let $p(\mathbf{x})$ and $\mathbf{v}(\mathbf{x})$ denote the fluid pressure and velocity fields, and let γ denote the membrane of the vesicle. The motion of the background fluid is described by the Stokes equations,

$$-\mu\triangle\mathbf{v} + \nabla p = \mathbf{0} \quad \text{and} \quad \operatorname{div}\mathbf{v} = \mathbf{0} \quad \text{in} \quad \mathbb{R}^3\backslash\gamma, \tag{2}$$

where μ is the viscosity of the fluid. The no-slip boundary condition on γ and the free-space boundary condition requires that

$$\mathbf{v} = \dot{\mathbf{x}} \quad \text{on} \quad \gamma, \quad \lim_{\mathbf{x}\to\infty}\mathbf{v}(\mathbf{x}) - \mathbf{v}_{\infty}(\mathbf{x}) = \mathbf{0}, \tag{3}$$

where $\dot{\mathbf{x}}$ is the total derivative of the motion of material point on the vesicle surface (i.e., its velocity) and $\mathbf{v}_{\infty}$ is the far-field fluid velocity. The continuity of the forces across the interface results in a stress vector jump across the interface γ of magnitude $\mathbf{f}$. To derive an expression for $\mathbf{f}$ we have to consider the constitutive properties of the vesicle membrane. The standard assumptions for vesicles consider a surface elastic energy that consists of two terms:

$$\mathcal{E}(H,\sigma) = \int_{\gamma}\frac{1}{2}\kappa_B H^2 + \sigma\,d\gamma, \tag{4}$$

where κ_B is the bending modulus and H is the mean curvature. The first term is the bending energy and the second term is required to enforce the local inextensibility constraint of the surface. In other words, the

tension σ is a Lagrange multiplier that enforces the constraint. The interfacial force can be derived from the surface energy by taking its L^2-gradient

$$\mathbf{f} = -\frac{\delta \mathcal{E}}{\delta \mathbf{x}},$$

In order to derive a formula for $\mathbf{f}$ in terms of the curvature and the parameterization of the surface, we need to introduce a few quantities. Let $\mathbf{x}(u, v) : U \to \gamma$ be a parametrization of the surface. The corresponding fundamental form coefficients are [21],

$$E = \mathbf{x}_u \cdot \mathbf{x}_u, \quad F = \mathbf{x}_u \cdot \mathbf{x}_v, \quad G = \mathbf{x}_v \cdot \mathbf{x}_v \quad \text{(first fundamental form)}, \tag{5}$$

$$L = \mathbf{x}_{uu} \cdot \mathbf{n}, \quad M = \mathbf{x}_{uv} \cdot \mathbf{n}, \quad N = \mathbf{x}_{vv} \cdot \mathbf{n} \quad \text{(second fundamental form)}. \tag{6}$$

The normal to the surface and the area element are defined by

$$\mathbf{n} = (\mathbf{x}_u \times \mathbf{x}_v)/\sqrt{EG - F^2}, \quad dA = \sqrt{EG - F^2}\, du\, dv = W\, du\, dv. \tag{7}$$

We can now define the mean and Gaussian curvatures as

$$H = \frac{1}{2}\frac{EN - 2FM + GL}{W^2}, \quad K = \frac{LN - M^2}{W^2}. \tag{8}$$

Then, following [26], the gradient or first variation of (4) is given by

$$\frac{\delta \mathcal{E}}{\delta \mathbf{x}} = \int_\gamma \left(\Delta_S H + 2H(H^2 - K)\right)\mathbf{n} \cdot \delta\mathbf{x} - (\sigma\Delta_S\mathbf{x} + \nabla_S\sigma) \cdot \delta\mathbf{x}\, d\gamma, \tag{9}$$

where Δ_S is the Laplace-Beltrami operator defined by

$$\Delta_S\phi = \frac{1}{W}\left(\left(\frac{E\phi_v - F\phi_u}{W}\right)_v + \left(\frac{G\phi_u - F\phi_v}{W}\right)_u\right), \quad \text{for some scalar function } \phi. \tag{10}$$

From (9), we define the bending and tension forces as

$$\mathbf{f}_b = -(\Delta_S H + 2H(H^2 - K))\mathbf{n}, \quad \mathbf{f}_\sigma = \sigma\Delta_S\mathbf{x} + \nabla_S\sigma. \tag{11}$$

The force density exerted by the vesicle membrane on the fluid is given by $\mathbf{f} = \mathbf{f}_b + \mathbf{f}_\sigma$. Using classical potential theory [14], the solution of (2, 3), combined with the local inextensibility constraint of the membrane can be written as

$$\dot{\mathbf{x}} = \mathbf{v}_\infty(\mathbf{x}) + \mathcal{S}[\mathbf{f}_b + \mathbf{f}_\sigma](\mathbf{x})$$

$$\text{div}_\gamma\left(\mathcal{S}[\mathbf{f}_\sigma]\right) = -\text{div}_\gamma\left(\mathbf{v}_\infty + \mathcal{S}[\mathbf{f}_b]\right). \tag{12}$$

This is a system of two integro-differential equations for the two unknowns: the position of the membrane $\mathbf{x}$ and the tension σ. The single layer potential operator is defined by $\mathcal{S}[\mathbf{f}](\mathbf{x}) = \int_\gamma G(\mathbf{x}, \mathbf{y})\mathbf{f}(\mathbf{y})\, d\mathbf{y}$, where G is the free-space Green's function for the Stokes operator and is given by

$$G(\mathbf{x}, \mathbf{y}) = \frac{1}{8\pi\mu}\left(\frac{1}{\|\mathbf{r}\|}\mathbf{I} + \frac{\mathbf{r}\otimes\mathbf{r}}{\|\mathbf{r}\|^3}\right), \quad \mathbf{r} = \mathbf{x} - \mathbf{y}. \tag{13}$$

Next, we present the reduction of these equations to one spatial variable in the axisymmetric case.

2.1 Axisymmetric formulation

Assuming symmetry in the 'v' direction, the positions and the interfacial forces take the following form

$$\mathbf{x} = \begin{bmatrix} x_1(u)\cos v \\ x_1(u)\sin v \\ x_2(u) \end{bmatrix}, \quad \mathbf{f} = \begin{bmatrix} f_1(u)\cos v \\ f_1(u)\sin v \\ f_2(u) \end{bmatrix}. \tag{14}$$

The parametric domain $\{u, v\} \in U$ is $[0, 2\pi] \times [0, 2\pi]$ for toroidal topologies; representing all variables in the trigonometric basis guarantees that the resulting functions, are well-defined as a function on the toroidal domain $[(R + \cos u)\cos v, (R + \cos u)\sin v, \sin u]$, with $R = \sqrt{2}$. A sphere can be regarded as a degenerate torus with $R = 0$, with each point of the sphere corresponding to two points on the torus. To make this mapping one-to-one we consider only one half of the parametric domain $[0, \pi] \times [0, 2\pi]$. For $\mathbf{x}$ to be a smooth function on the sphere, it is necessary and sufficient that x_1 is an odd and x_2 is an even periodic function of u; in other words, a trigonometric series for x_1 and x_2 have only nonzero coefficients for sines and cosines respectively. Similarly, any scalar function defined on the surface needs to be even in u.

We can now write the bending and tension forces in terms of u. Let s be the arclength parameter, that is, $s(u) = \int_0^u \|\mathbf{x}(u')\| \, du'$. In the Appendix B, we derive the expressions for the forces in terms of the principal curvatures κ and β; here, we just state the result:

$$\mathbf{f}_b = \frac{1}{2}\left(\triangle_S(\kappa + \beta) + \frac{(\kappa + \beta)(\kappa - \beta)^2}{2}\right)\mathbf{n}, \quad \mathbf{f}_\sigma = (\sigma\mathbf{x}_s)_s - \sigma\beta\mathbf{n}, \tag{15}$$

and at the poles, we have $\quad \lim_{x_1 \to 0} \mathbf{f}_b = \kappa_{ss}\mathbf{n}, \quad \lim_{x_1 \to 0} \mathbf{f}_\sigma = \sigma_s\mathbf{x}_s - 2\sigma\kappa\mathbf{n}.$ \tag{16}

Next, we derive the axisymmetric form of the single layer potential. Without loss of generality, we assume that the targets on the surface are located at the cross-section $v = 0$. Then, the target and source points have the form $\mathbf{x} = [x_1, 0, x_2]^T$ and $\mathbf{y}(u, v) = [y_1\cos v, y_1\sin v, y_2]^T$ respectively (for notational clarity, we drop the explicit dependence of x_i and y_i, $i = 1, 2$, on u). The single layer potential can be written as

$$\mathcal{S}[\mathbf{f}] = \begin{bmatrix} F_1 \\ 0 \\ F_2 \end{bmatrix} = \int_0^{2\pi} dv \int_0^\pi du \left(\frac{1}{|\mathbf{r}|}\mathbf{I} + \frac{\mathbf{r} \otimes \mathbf{r}}{|\mathbf{r}|^3}\right) \begin{bmatrix} f_1\cos v \\ f_1\sin v \\ f_2 \end{bmatrix} y_1\|\mathbf{y}_u\|,$$

$$\text{where} \quad \mathbf{r} = \begin{bmatrix} y_1\cos v - x_1 \\ y_1\sin v \\ y_2 - x_2 \end{bmatrix}; \quad |\mathbf{r}| = \left[x_1^2 + y_1^2 - 2x_1y_1\cos v + (x_2 - y_2)^2\right]^{1/2}.$$

$$\mathcal{S}[\mathbf{f}] = \begin{bmatrix} F_1 \\ F_2 \end{bmatrix} = \int_0^{2\pi} dv \int_0^\pi du \begin{bmatrix} \frac{\cos v}{|\mathbf{r}|} + \frac{(y_1\cos v - x_1)(y_1 - x_1\cos v)}{|\mathbf{r}|^3} & \frac{(y_1\cos v - x_1)(y_2 - x_2)}{|\mathbf{r}|^3} \\ \frac{(y_1 - x_1\cos v)(y_2 - x_2)|}{|\mathbf{r}|^3} & \frac{1}{|\mathbf{r}|} + \frac{(y_2 - x_2)^2}{|\mathbf{r}^3|} \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} y_1\|\mathbf{y}_u\|. \tag{17}$$

All the integrals with respect to 'v' are computed analytically using equations (53–57). In summary, the axisymmetric form of the 3D Stokes operator is given by

$$\mathcal{S}[\mathbf{f}](\mathbf{x}) = \int_0^\pi K(\mathbf{x}, u)\mathbf{f}(u)y_1(u)\|\mathbf{y}_u\|\, du. \tag{18}$$

The kernel K is composed of *elliptic integrals* of first and second kind.

Gravitational force. If there is a density difference across the membrane of a vesicle, then the vesicle experiences an additional force due to gravity given by

$$\mathbf{f}_g = (\rho^{\text{in}} - \rho^{\text{out}})(\mathbf{g} \cdot \mathbf{x})\mathbf{n}. \tag{19}$$

Then, the governing equations that include graviational forces are

$$\dot{\mathbf{x}} = \mathbf{v}_\infty + \mathcal{S}[\mathbf{f}_b + \mathbf{f}_\sigma + \mathbf{f}_g]; \quad \text{div}_\gamma(\mathcal{S}[\mathbf{f}_\sigma]) = -\text{div}_\gamma(\mathbf{v}_\infty + \mathcal{S}[\mathbf{f}_b + \mathbf{f}_g]). \tag{20}$$

Scaling. Following [11], we set the length and time scales as $R_0 = \sqrt{\frac{A}{4\pi}}$ and $\tau = \frac{\mu R_0^3}{\kappa_B}$ respectively, where A is the surface area of the vesicle. In the absence of external flows and gravity, it is known that the dynamics of the vesicle are characterized by a single parameter [11], namely, the reduced volume$(\nu) = \frac{6\sqrt{\pi}V}{A^{3/2}}$.

Since we are dealing with an axisymmetric problem, we must consider an axisymmetric $\mathbf{v}_\infty$, for example a velocity field with parabolic profile that smoothly decays to zero away from the axis of symmetry, to resemble the profile of a Poiseuille flow. Typically, we consider velocity profiles of the form $\mathbf{v}_\infty = c\left(w^2 - x_1^2(u)\right)$, where c and w are constants. Notice that the curvature of this velocity profile is $\frac{\partial^2 \mathbf{v}_\infty}{\partial x_1^2} = -2c$, and the corresponding shear rate is $\frac{\partial \mathbf{v}_\infty}{\partial x_1} = -2cx_1(u)$. We introduce the nondimensional entity $\hat{c} = \frac{c\mu R_0^4}{\kappa_B}$ that parameterizes such external flows.

In the presence of gravity, an additional parameter that governs the vesicle dynamics is the nondimensional gravity parameter, given by $\hat{g} = \frac{(\rho^{\text{in}} - \rho^{\text{out}})g R_0^4}{\kappa_B}$.

Multiple Vesicles. The governing equations in the multiple vesicle case are given by

$$\dot{\mathbf{x}}_j = \mathbf{v}_\infty(\mathbf{x}_j) + \mathcal{S}_j[\mathbf{f}_b + \mathbf{f}_\sigma](\mathbf{x}_j) + \sum_{\substack{k=1 \\ k\neq j}}^K \mathcal{S}_k[\mathbf{f}_b + \mathbf{f}_\sigma](\mathbf{x}_j), \tag{21}$$

$$\text{div}_{\gamma j}\left(\mathcal{S}_j[\mathbf{f}_\sigma]\right) = -\text{div}_{\gamma j}\left(\mathbf{v}_\infty(\mathbf{x}_j) + \mathcal{S}_j[\mathbf{f}_b](\mathbf{x}_j) + \sum_{\substack{k=1 \\ k\neq j}}^K \mathcal{S}_k[\mathbf{f}_b + \mathbf{f}_\sigma](\mathbf{x}_j)\right). \tag{22}$$

where we separate the terms accounting for the interactions with other vesicles.

To summarize, equations (15),(16), (17), (21) and (22) form a complete system of equations for $\mathbf{x}(u)$, $\mathbf{y}(u)$ and $\sigma(u)$.

3 Spatial discretization scheme

Spatial discretization needs to support efficient and high-order computation of derivatives for computing bending and tension forces $\mathbf{f}_b$ and $\mathbf{f}_\sigma$ and accurate computation of integrals (17) involving singular kernels.

We use the trigonometric polynomial bases to represent the position of the interface and functions defined on it.

x_1 and x_2 are given by the coefficients $\hat{x}_1(k)$ and $\hat{x}_2(k)$: [1]

$$x_1(u) = \sum_{k=1}^{M} \hat{x}_1(k) \sin(ku), \quad x_2(u) = \sum_{k=0}^{M} \hat{x}_2(k) \cos(ku). \tag{23}$$

(Recall that for smoothness $x_1(u)$ is required to be odd and $x_2(u)$ even). Similarly, the tension is represented as $\sigma(u) = \sum_{k=0}^{M} \hat{\sigma}(k) \cos(ku)$. The spatial to spectral transform and vice-versa are computed efficiently using the forward and inverse fast sine- and cosine-transforms. This representation allows for an efficient derivative computation:

$$x_{1u}(u) = \sum_{k=1}^{M} k\hat{x}_1(k) \cos(ku), \quad x_{2u}(u) = -\sum_{k=1}^{M} k\hat{x}_2(k) \sin(ku). \tag{24}$$

Since we assume that the shape of the vesicle is smooth, this scheme to compute the derivatives yields spectral accuracy. We make a few more remarks on the dervative accuracy in Appendix D.

3.1 Quadrature Scheme

The kernels in (18) have a logarithmic singularity, which can be verified by examining their asympototic expansions. Let $z \in (0,1)$, then we have the following expansions around $z = 0$,

$$\text{EllipticK}(1-z) = c_0 - \tfrac{1}{2}\ln z + \left(c_1 - \tfrac{1}{4}\ln z\right)z + \left(c_2 - \tfrac{5}{32}\ln z\right)z^2 + \mathcal{O}(z^3) \quad \textit{(first kind)},$$

$$\text{EllipticE}(1-z) = d_0 + \left(d_1 - \tfrac{1}{2}\ln z\right)z + \left(d_2 - \tfrac{1}{8}\ln z\right)z^2 + \mathcal{O}(z^3) \quad \textit{(second kind)},$$

for some constants c_k and $d_k, k \geq 0$. To resolve the logarithmic singularity, we use the high-order Gauss-trapezoidal rules of [1] that compute integrals of the form $\int_0^1 \phi(z) \ln z + \psi(z)\, dz$, where $\phi(z)$ and $\psi(z)$ are smooth functions. To compute (18), we split the interval of integration into two parts: $(0, u')$ and (u', π), where u' is the evaluation point on the boundary, that is, $\mathbf{y}(u') = \mathbf{x}$. In each interval, we use the Gauss-trapezoidal rule to handle the singularity at $u = u'$. To compute the integrand at the Gauss points, we use Fourier interpolation.

Treatment at the poles. Substituting $x_1 = 0$ in (17), we get

$$\mathcal{S}[\mathbf{f}] = \begin{bmatrix} F_1 \\ F_2 \end{bmatrix} = \int_0^{2\pi} \int_0^{\pi} \begin{bmatrix} \frac{\cos v}{|\mathbf{r}|} + \frac{y_1 \cos v}{|\mathbf{r}|^3} & \frac{(y_1 \cos v)(y_2 - x_2)}{|\mathbf{r}|^3} \\ \frac{y_1(y_2 - x_2)}{|\mathbf{r}|^3} & \frac{1}{|\mathbf{r}|} + \frac{(y_2 - x_2)^2}{|\mathbf{r}^3|} \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} y_1 \|\mathbf{y}_u\| \, du dv \tag{25}$$

---

[1] In the case of torus, we use Fourier basis, $\mathbf{x}(u) = \sum_{k=-M/2}^{M/2-1} \hat{\mathbf{x}}(k) e^{-iku}$.

where $|\mathbf{r}| = \sqrt{y_1^2 + (y_2 - x_2)^2}$. Performing analytic integration in the 'v' direction, we get the following expressions for F_1 and F_2:

$$F_1 = 0; \quad F_2 = 2\pi \int_0^\pi \frac{y_1^2(y_2 - x_2)}{|\mathbf{r}|^3} f_1 \|\mathbf{y}_u\| + \left(\frac{y_1}{|\mathbf{r}|} + \frac{y_1(y_2 - x_2)^2}{|\mathbf{r}|^3} \right) f_2 \|\mathbf{y}_u\| \, du. \tag{26}$$

Notice that the kernel is non-singular and hence the quadrature rule is modified accordingly.

Trigonometric discretization of the surface and the high-order quadrature scheme allow us, for a given smooth surface position, to compute the velocities of surface points with spectral accuracy.

4 Time-Stepping scheme

In this section, we describe three time-stepping schemes for (1).

As a starting point, we consider the explicit scheme that has been used by several authors [11, 20, 5] for vesicle simulations. While it has a low cost per time step, we demonstrate that this scheme suffers from severe stability constraints on the time-step size. The maximal stable step size for this scheme is often significantly smaller than the step size needed to resolve the physics of the vesicle motion.

Our second time-stepping scheme is an extension of the semi-implicit scheme that we introduced in [25]. We have shown that in the two-dimensional case, the numerical stiffness can be circumvented by regarding the stiffest terms of the right-hand side of the equations as linear spatially-variant operators acting on the surface point positions and tensions, e.g., $\mathcal{B}(\mathbf{x})\mathbf{x}$. Then $\mathcal{B}(\mathbf{x})$ part is treated explicitly and $\mathbf{x}$ implicity. Such methods are usually referred to as semi-implicit or implicit-explicit methods [2]. Apart from improving the numerical stability, these methods have the advantage that they lead to *linear* algebraic equations at every time-step. Finally, we discuss a third, fully implicit scheme in which the *nonlinear* equations are solved for each time step using an inexact Newton method.

The nonlinearity of the underlying system of equations renders their analysis quite difficult, and we rely on numerical experiments to analyze the behavior of our schemes. Overall, we have observed that **(i)** the semi-implicit schemes perform very well for spherical vesicles, eliminating the numerical stiffness and delivering orders-of-magnitude computational shavings compared to the explicit scheme; and **(ii)** for toroidal vesicles, the semi-implicit may be inadequate and the inexact Newton scheme, is needed.

4.1 Explicit Scheme

We introduce a linear operator $\mathcal{L}$, defined by, $\mathcal{L}\sigma = \text{div}_\gamma (\mathcal{S}[\mathbf{f}_\sigma])$. Given the current position of the membrane, in an explicit scheme, first the tension is computed by inverting $\mathcal{L}$ and then the position is updated. More specifically, let $\triangle t$ be a fixed time-step size and let the position at $n\triangle t$, denoted by $\mathbf{x}^n$, be known. Then, the following steps are performed to compute $\mathbf{x}^{n+1}$:

1. Compute the bending force $\mathbf{f}_b^n$

2. Compute $r^n = \mathtt{div}_\gamma\left(\mathcal{S}[\mathbf{f}_b^n]\right)$, with all the operators defined on $\mathbf{x}^n$

3. Solve $\mathcal{L}\sigma^n = r^n$

4. Compute the tension force $\mathbf{f}_\sigma^n$

5. Update the positions $\mathbf{x}^{n+1} = \mathbf{x}^n + \triangle t\, \mathcal{S}[\mathbf{f}_b^n + \mathbf{f}_\sigma^n]$

Computationally, the most expensive part of this scheme is computing σ^n by inverting $\mathcal{L}$ (Step 3). This step does not scale well with the number of unknowns because the condition number of $\mathcal{L}$ grows linearly with the number of spatial discretization points M, see Figure 1. Hence, when a Krylov iterative scheme is used to

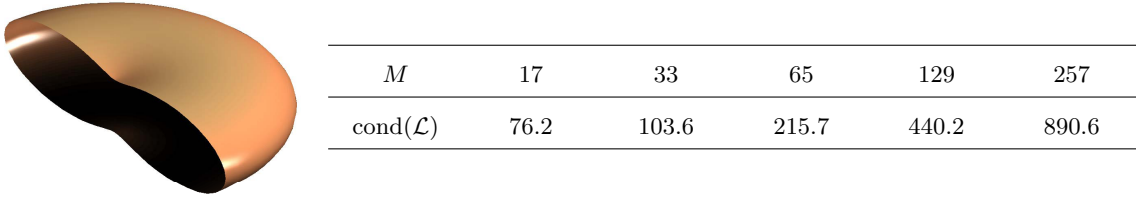| M	17	33	65	129	257
cond($\mathcal{L}$)	76.2	103.6	215.7	440.2	890.6

Figure 1: Condition number of the operator $\mathcal{L}$, computed on the adjacent shape, as a function of the number of spatial discretization points.

solve for σ^n, the number of iterations grow proportional to the number of spatial discretization points. For instance, GMRES [17] requires $\mathcal{O}(\sqrt{M})$ iterations.

We now describe a preconditioner, based on heuristics, that eliminates this mesh-dependence. We have shown in [25] that, on a unit circle, the Fourier transform diagonalizes the operator $\mathcal{L}$. We derived the spectrum Λ_c of $\mathcal{L}$ analytically and used its inverse as a preconditioner for solving the inextensibility constraint on a general boundary. Specifically, the preconditioner (P) is given by the Equation 22 of [25]:

$$P = \mathcal{F}^{-1}\Lambda_c^{-1}\mathcal{F}, \quad \Lambda_c = \mathrm{diag}\left\{\lambda_{-\frac{M}{2}}, \lambda_{-\frac{M}{2}+1}, \ldots, \lambda_{\frac{M}{2}-1}\right\}, \tag{27}$$

where $\mathcal{F}$ is the Fourier transform operator and λ_k is the kth–eigenvalue of $\mathcal{L}$ defined on unit circle and given by[2] $\lambda_k = -\frac{|k|}{8}$.

In the case of a unit sphere, we were unable to compute the spectrum analytically. In Figure 2, we plot the spectrum of the operator $\mathcal{L}$ for different boundary configurations. We note that the spectra roughly follow the same pattern. Moreover, they follow closely the spectrum of the corresponding operator defined on a unit circle in the $v = 0$ plane (also shown in the Figure). Therefore, it is possible to use P itself as a preconditioner. In order to do that, we first need to extend the parametric domain of the constraint equation from $u = [0, \pi]$ to the parametric domain of the unit circle defined in $v = 0$ plane, which is $u = [0, 2\pi]$. Recalling that

---

[2]There is a factor of two difference from the expression in [25]because of the difference in constants multiplying the corresponding Green's functions.
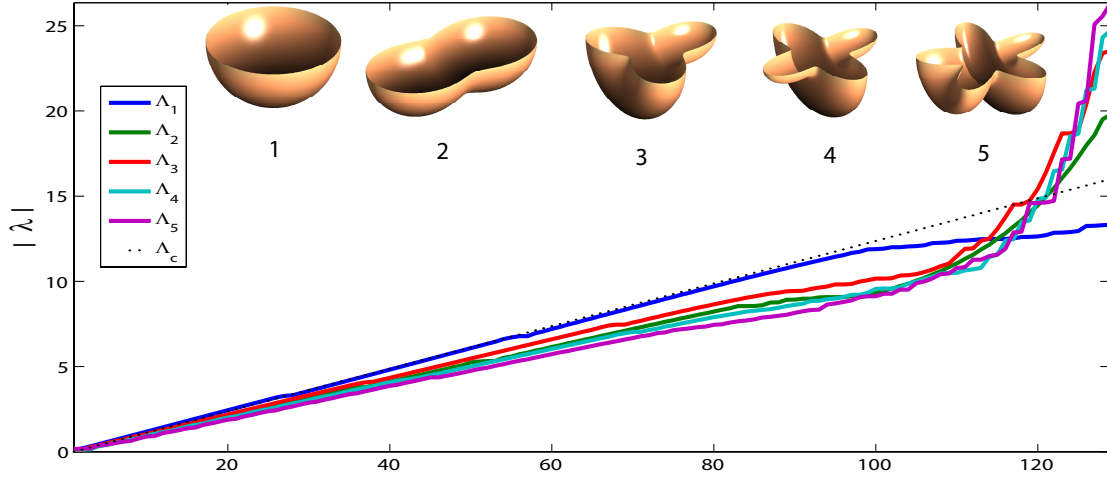
Figure 2: Plot of the eigenvalue magnitudes of the operator $\mathcal{L}$ defined on vesicle shapes with same surface area. For comparision purposes, we also show the spectrum of the corresponding operator defined on the unit circle.

scalar functions are even in u, the extension is simply $\mathcal{L}\sigma\,|_{2\pi-u} = \mathcal{L}\sigma|_u$ and $r(2\pi - u) = r(u)$, where r is the right-hand side of the constraint equation. Now the preconditioner is applied to solve $\mathcal{L}\sigma = r$ for σ at discrete equidistant points in $[0, 2\pi]$ and then only the values at the points within $[0, \pi]$ are retained.

In practice, we have found that this preconditioner works very well for solving the contraint equation on general geometries.

By incorporating spectral representations, high-order quadrature rules, fast spatial transforms and the preconditioner, we have minimized the computational cost per time-step of the explicit scheme. However, the fundamental drawback of the explicit scheme is the severe constraint on the size of the time-step, which still persists. We address this important issue in the next two schemes.

4.2   Semi-Implicit Scheme

In semi-implicit schemes, the linear part of the stiffest terms is treated implicitly [2]. Theoretically, such schemes were proved to be unconditionally stable for advection-diffusion problems [3]. Albeit the lack of theoretical proof, they have also been demonstrated to be efficient in many other problems of interest in computational physics [12, 22]. In our previous work [25], we have proposed two semi-implicit schemes for simulating the 2D vesicle dynamics and showed, through numerical experiments, that they dramatically improve the stable time-step sizes over the explicit scheme. The scheme that follows is an extension of those

schemes to the axisymmetric case.

First, we rewrite the bending force in a different form to facilitate the identification of the stiffest terms. In [25], we argued that, because of the inextensibility constraint, forces of certain form (specifically, aligned with virtual forces corresponding to the constraint) can be added or subtracted from the bending force without altering the dynamics (Appendix A of [25]). This was used to derive a simpler form for the bending force. In the axisymmetric case, similarly, forces in the form $(h\mathbf{x}_s)_s - h\beta\mathbf{n}$, for some scalar field $h(s)$, can be added to $\mathbf{f}_b$, as for any vesicle deformation satisfying the inextensibility constraint, these forces do not do any work. By substituting $h(s) = \frac{(\kappa-\beta)^2}{4}$ and modifying $\mathbf{f}_b$, we get the following form for the bending force,

$$\mathbf{f}_b(\mathbf{x}) = \frac{1}{2}\triangle_S(\kappa+\beta)\mathbf{n} + \frac{1}{2}(\kappa-\beta)(\kappa-\beta)_s\mathbf{x}_s, \quad \lim_{x_1\to 0}\mathbf{f}_b = \kappa_{ss}\mathbf{n}. \tag{28}$$

Again, we would like to emaphasize that, as in the 2D case, rewriting bending force is just for the ease of presentation. The force in (28) is comprised of a normal and a tangential component. Now, we can easily pick the terms with highest order spatial dervatives for implicit treatment. While counting the order of derivatives applied to the coordinate functions in each term, one should be cautious of its behavior at the poles. For instance, although $\beta = \frac{x_{2s}}{x_1}$ may seem to have only a single derivative with respect to s, at the poles we have $\lim_{x_1\to 0}\beta(u) = \kappa(u)$, and hence it has second-order derivatives. Therefore, the two terms in the normal component, $\triangle_S\kappa$ and $\triangle_S\beta$, have fourth-order derivatives in 'u' and hence are candidates for implicit treatment. Similarly, in the tangential component, it is easy to see that the term $(\kappa - \beta)_s$ has the highest (third) order spatial derivatives. Following these observations, we are now ready to propose the semi-implicit scheme. We assume that the position of the spatial points $\{\mathbf{x}^n(u_k)\}_{k=1}^M$ at time $n\triangle t$ is known. The goal is to obtain the corresponding positions and tensions, $\{\mathbf{x}^{n+1}(u_k), \sigma(u_k)\}_{k=1}^M$, at $(n+1)\triangle t$. For the simplest (first-order accurate) time discretization our semi-implicit scheme is

$$\frac{1}{\triangle t}\left(\mathbf{x}^{n+1} - \mathbf{x}^n\right) = \mathcal{S}\left[\mathbf{f}_b^{n+1} + \mathbf{f}_\sigma^{n+1}\right](\mathbf{x}^n), \tag{29}$$

$$\mathcal{L}(\mathbf{x}^n)\sigma^{n+1} = -\text{div}_{\gamma^n}\mathcal{S}\left[\mathbf{f}_b^{n+1}\right](\mathbf{x}^n), \tag{30}$$

where $\mathcal{S}[\mathbf{f}](\mathbf{x}^n) = 2\pi\int_0^\pi G(\mathbf{x}^n(u'), \mathbf{x}^n(u))x_1^n|\mathbf{x}_u^n|\,du$ for any force field $\mathbf{f}$. In the following definition of forces, for notational simplicity, we drop the subscript on the terms that are treated explicitly. For example, we substitute κ^n by κ and so on.

$\mathbf{f}_b^{n+1} = \frac{1}{2}\triangle_S\left(\kappa^{n+1} + \beta^{n+1}\right)\mathbf{n} + \frac{1}{2}(\kappa-\beta)\left(\kappa^{n+1} - \beta^{n+1}\right)_s\mathbf{x}_s$ $\hfill$ *bending force,*

$\mathbf{f}_\sigma^{n+1} = \left(\sigma^{n+1}\mathbf{x}_s\right)_s - \sigma^{n+1}\beta\mathbf{n}$ $\hfill$ *tension force,*

$\kappa^{n+1} = x_{1s}\frac{1}{|\mathbf{x}_u|}\left(\frac{x_{2u}^{n+1}}{|\mathbf{x}_u|}\right)_u - x_{2s}\frac{1}{|\mathbf{x}_u|}\left(\frac{x_{1u}^{n+1}}{|\mathbf{x}_u|}\right)_u, \quad \beta^{n+1} = \frac{1}{x_1}\left(\frac{x_{2u}^{n+1}}{|\mathbf{x}_u|}\right)$ $\hfill$ *curvatures,*

$\lim_{x_1\to 0}\beta^{n+1} = \frac{1}{x_{1u}}\left(\frac{x_{2u}^{n+1}}{|\mathbf{x}_u|}\right)_u \quad \lim_{x_1\to 0}\triangle_S\left(\kappa^{n+1} + \beta^{n+1}\right) = 2\left(\kappa^{n+1} + \beta^{n+1}\right)_{ss}$ $\hfill$ *pole conditions.*

We use GMRES to solve the coupled linear set of equations (29, 30). Similar to the explicit scheme, we use P as a precondtioner for the constraint equation (30). The total cost per time-step of this scheme exceeds that of an explicit scheme by a factor of the number of iterations required to solve (29). If present, the background velocity and the gravitational force are treated explicity. In the case of multiple vesicles, the interaction forces are also treated explicitly.

Versions of this scheme with higher order in time are readily obtained using backward difference formula [2]. As in the 2D case [25], these schemes do yield high-order convergence, but only so in certain flow regimes.

In the case of spherical vesicles, this scheme overcomes the high-order stiffness and allows for stable time-step sizes that are orders of magnitude higher than those allowed by the explicit scheme (see Section 5). Therefore, the semi-implicit scheme yields significant cost savings by not having to take too many unnecessary time-steps. In the case of toroidal vesicles, we observed that, in practice, the semi-implicit scheme still has a stability constraint. Next, we present a time-scheme that has higher computational cost per time-step but performs well in the case of toroidal vesicles.

4.3 Inexact Newton Scheme

In the semi-implicit scheme (29, 30), we extracted a linear part of the stiffest term and treated it implicitly. An compact form of this scheme is

$$\frac{1}{\triangle t}\left(\mathbf{x}^{n+1} - \mathbf{x}^n\right) = Q(\mathbf{x}^n)\mathbf{x}^{n+1}, \tag{31}$$

where the operator $Q(\mathbf{x})$ encodes both (i) the terms in the bending force that were treated explicitly and (ii) the inextensibility constraint. Further, we define the Jacobian (J) and residual (R) as follows

$$J(\mathbf{x}) = 1 - \triangle t\, Q(\mathbf{x}); \quad R^n(\mathbf{x}) = J(\mathbf{x})\mathbf{x} - \mathbf{x}^n. \tag{32}$$

In the semi-implicit scheme, we solved the linear algebraic equation $J(\mathbf{x}^n)\mathbf{x}^{n+1} = \mathbf{x}^n$ to update the positions. On the other hand, in a fully implicit scheme, we solve the nonlinear equation $J(\mathbf{x}^{n+1})\mathbf{x}^{n+1} = \mathbf{x}^n$, typically, by using one of the many variants of Newton scheme. This is computationally expensive but can, in principle, lead to a more stable method.

In an inexact Newton scheme, instead of solving the nonlinear equation, the Jacobian is replaced by an approximation and a search direction that minimizes the residual is found at each Newton iteration:

1. `Set` $\tilde{J} = J(\mathbf{x}^n)$ *compute inexact Jacobian*

2. $\mathbf{x}_0 = \tilde{J}^{-1}\mathbf{x}^n$

3. $k = 0$

4. `while` $\min_\lambda R[\mathbf{x}_k - \lambda \tilde{J}^{-1} R(\mathbf{x}_k)] > \epsilon$ & $k < $ `MaxIts`   *check for residual convergence*

5. $\mathbf{p} = -\tilde{J}^{-1}R(\mathbf{x}_k)$ *determine descent direction*

6. $\lambda_m \leftarrow \min_\lambda R[\mathbf{x}_k + \lambda \mathbf{p}]$ *line search*

7. $\mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_m \mathbf{p}$

8. $k = k + 1$

9. `end while`

10. $\mathbf{x}^{n+1} = \mathbf{x}_k$ *update positions*
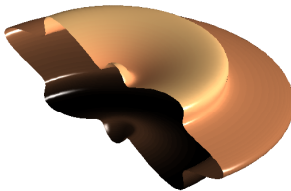
5   Results

In this section, we present numerical experiments to demonstrate the stability and convergence of our numerical scheme.

Example 1.   First, we verify the accuracy of our spatial discretization scheme. Consider a vesicle's surface defined by

$$\mathbf{x}(u,v) = \begin{bmatrix} x_1(u)\cos v \\ x_1(u)\sin v \\ x_2(u) \end{bmatrix}, \qquad \begin{array}{l} x_1(u) = (\sqrt{\cos^2 u + 9\sin^2 u} + \cos^2 4u)\sin u \\ x_2(u) = -\frac{1}{2}(\sqrt{\cos^2 u + 9\sin^2 u} + \cos^2 4u)\cos u \end{array}, \quad u \in [0,\pi],\ v \in [0,2\pi]. \tag{33}$$

In Figure 3, we report the errors in computing the two principal curvatures on this surface. Since we use spectral differentiation, the errors decay rapidly. In Table 1, we report the errors in computing the single layer potential using a fourth order quadrature scheme described in Section 3.

M	κ	β
9	2.52e+03	1.77e+00
17	1.43e-03	1.54e-03
33	1.01e-05	3.42e-06
65	1.08e-10	3.14e-11
129	1.91e-13	2.73e-13

Figure 3: Relative errors in computing the curvatures numerically on the shape shown. The reference values are computed analytically. As expected, we observe spectral convergence.

M	9	17	33	65	129	257
Quadrature error	2.21e-02	1.50e-04	1.10e-05	8.50e-07	5.26e-08	2.99e-09

Table 1: Relative errors in computing $\mathcal{S}[\mathbf{n}]$ defined on the boundary shown in Figure 3. The singular integrals are computed analytically in the v-direction and a fourth-order quadrature rule is used to compute the resulting integrals in the u-direction. Reference values are computed numerically by a finer discretization ($M = 513$).

Example 2. In the second example, we consider the motion of a vesicle suspended in a external parabolic flow, shown in Figure 4. The surface parameters of the initial vesicle shape are given by

$$x_1(u) = (\sqrt{\cos^2 u + 9\sin^2 u})\sin u, \quad x_2(u) = -\frac{1}{3}(\sqrt{\cos^2 u + 9\sin^2 u})\cos u. \tag{34}$$

For any non-zero shear rate, the vesicle undergoes large deformations to reach an equilibrium parachute-like shape and then translates with a constant velocity. Although the vesicle is suspended in an unbounded Poiseuille flow, the resultant equilibrium shape is similar to the ones obtained through numerical experiments in [24] for capillary flows.
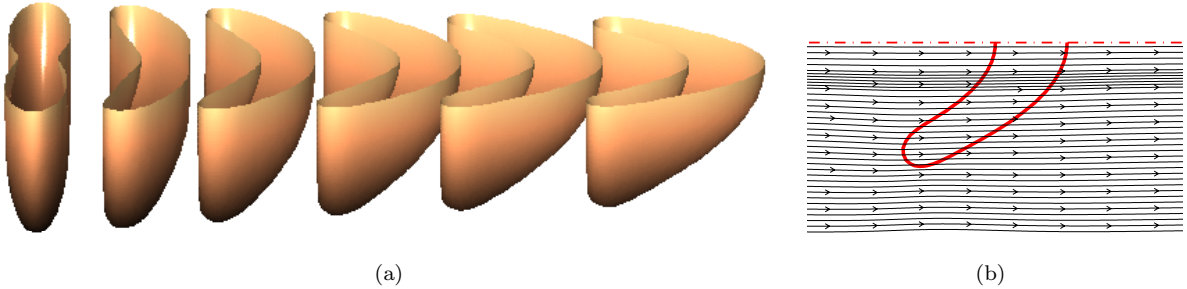

(a)   (b)

Figure 4: (a) Snapshots of an oblate vesicle suspended in an external parabolic flow with $\mathbf{v}_\infty = \left(0, 0, c\left(1 - \frac{x_1^2}{4R_0^2}\right)\right)$, where c is a constant; (b) cross-sectional plot of streamlines at the end of the simulation. The vesicle reaches an equilibrium parachute-like shape and translates with a constant velocity. For this simulation, we report the stability of our numerical scheme in Table 2, accuracy in Table 4, and the performance of the preconditioner in Table 5.

We study the stability and convergence properties of various schemes based on this simulation. In Table 2, we list the maximum allowable time-step size $\triangle t_{\max}$ for the explicit and the semi-implicit schemes. We determine $\triangle t_{\max}$ by starting from an aribitrarily large time-step and checking if the numerical simulation is stable. If not, we reduce the step size by half and repeat the experiment until we get a stable simulation. We can infer from the table that the explicit scheme requires $\triangle t_{\max}$ to be inversely propotional to the cube

14

of M (approximately). On the other hand, $\triangle t_{\max}$ is independent of M for the semi-implicit scheme.

Notice, however, that $\triangle t_{\max}$ is inversely proportional to the shear rate. The inexact Newton scheme also has similar behavior, see Table 3, which suggests that this time-step restriction may be related to the properties of the solution, rather than only be a limitation of the numerical method. In all our numerical experments with spheroidal vesicles, we observed that the semi-implict scheme performs as well as the inexact Newton scheme. Since, relatively, the computational cost per time-step of the semi-implicit scheme is much lower, it is the method choice for spheroids.

M	Explicit scheme		Semi-implicit scheme	
	$\hat{c} = 0$	200	0	200
17	2.75e-03	5.48e-05	2.50e-01	4.88e-04
33	3.44e-04	6.85e-06	2.50e-01	4.88e-04
65	2.15e-05	4.28e-07	2.50e-01	4.88e-04
129	2.69e-06	5.35e-08	2.50e-01	4.88e-04
257	3.36e-07	6.69e-09	2.50e-01	4.88e-04

Table 2: Stable (nondimensional) time-step sizes for first-order explicit and semi-implicit schemes for a spheroidal vesicle whose initial configuration is shown in Figure 4. Here, $\hat{c}$ is the nondimensional curvature of the external parabolic flow profile. While the explicit scheme suffers from a severe stability restriction on the time-step size, the semi-implicit scheme is devoid of such restrictions. However, the semi-implicit scheme requires that the time-step size is inversely proportional to the shear rate.

M	$\hat{c} = 2$	20	200
33	1.56e-02	3.90e-03	4.88e-04
129	1.56e-02	3.90e-03	4.88e-04

Table 3: Stable time-step sizes in the case of the inexact Newton scheme for the simulation in Figure 4.

Since the interior of the vesicle is filled with an incompressible fluid, the volume enclosed by it is preserved. As the surface is locally inextensible, the total surface area must also be preserved. In Table 4, we report the relative errors in preserving the total volume and surface area of the vesicle shown in Figure 4.

Next, for the same simulation , we study the performance of the preconditioner to solve the inextensibility constraint. In Table 5, we list the average number of iterations required for solution using GMRES. The preconditioner reduces the number of iterations from $\mathcal{O}(\sqrt{M})$ to nearly $\mathcal{O}(1)$.

M	$\frac{\|A_f - A\|}{A}$		$\frac{V_f - V}{V}$	
	$q = 1$	$q = 3$	$q = 1$	$q = 3$
17	1.34e-03	6.31e-04	4.50e-04	4.42e-04
33	8.93e-04	1.55e-04	6.10e-05	1.81e-05
65	4.89e-04	3.88e-05	6.65e-06	7.87e-07
129	2.54e-04	9.28e-06	3.66e-06	6.07e-08

Table 4: Surface area and the enclosed volume must be preserved in a vesicle simulation. Here, we report the relative errors in the area and volume measured at the end of the simulation shown in Figure 4. q is the convergence order of the semi-implicit scheme, M is the number of spatial discretization points and $\triangle t = \frac{1}{M}$.

Preconditioner	None		P	
M	$\epsilon = 10^{-6}$	$\epsilon = 10^{-12}$	$\epsilon = 10^{-6}$	$\epsilon = 10^{-12}$
17	9	10	8	10
33	16	19	9	17
65	26	36	8	17
129	40	58	8	18
257	58	86	8	18
513	84	127	8	19

Table 5: Performance of the preconditioner to solve the inextensibility constraint. Here, we report the number of GMRES iterations required to solve the discrete inextensibility constraint equation within a relative tolerance of ϵ. Without a preconditioner, this number increases approximately proportional to $\sqrt{M}$. This is because the condition number of $\mathcal{L}$ increases linearly with M. On the other hand, the preconditioner P yields nearly mesh-independent convergence.

Example 3. We consider the vesicle with oscillatory initial shape, defined in equation (33), and simulate its motion to equilibrium in the absence of an external flow. We show snapshots in Figure 5. The advantage of the semi-implicit scheme is clear from this example: we can simulate the dynamics with fewer number of time-steps.

The flow field around the vesicle can be computed by the following expression

$$\mathbf{v}(\mathbf{x}) = \int_\gamma G(\mathbf{x}, \mathbf{x}')[\mathbf{f}_b(\mathbf{x}') + \mathbf{f}_\sigma(\mathbf{x}')] \, d\gamma(\mathbf{x}'), \quad \mathbf{x} \in \mathbf{R}^3. \tag{35}$$

We plot the streamlines corresponding to this simulation in Figure 6.
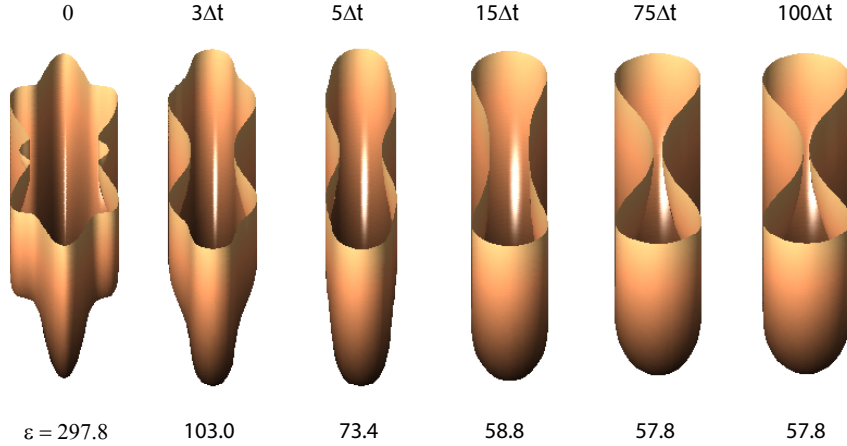
Figure 5: Snapshots of a freely suspended vesicle, with complicated initial shape, relaxing to equilibrium. We require $M = 128$ to resolve the initial shape to double precision ($\epsilon = 10^{-12}$). The advantage of the semi-implicit scheme is conspicuous from this experiment: while a fully explicit demands more than a million time-steps to simulate the dynamics, the semi-implicit scheme requires fewer than hundred time-steps.

Example 4. We consider a toridal vesicle suspended freely in a viscous fluid. The surface parameters are given by

$$x_1(u) = (1 + 0.03 \cos 5u) \cos u, \quad x_2(u) = (1 + 0.03 \cos 5u) \sin u. \tag{36}$$

In Figure 7, we show the snapshots of the vesicle relaxing to an equilibrium Willmore torus. We also list the maximum time-step sizes for different discretizations allowed by the inexact Newton and the semi-implicit schemes.

Simulations in the presence of gravity. Presence of gravitational field alters the dynamics in many interesting ways. We observed that, in the absence of external flow, if the parameter $\hat{g}$ is low, a vesicle
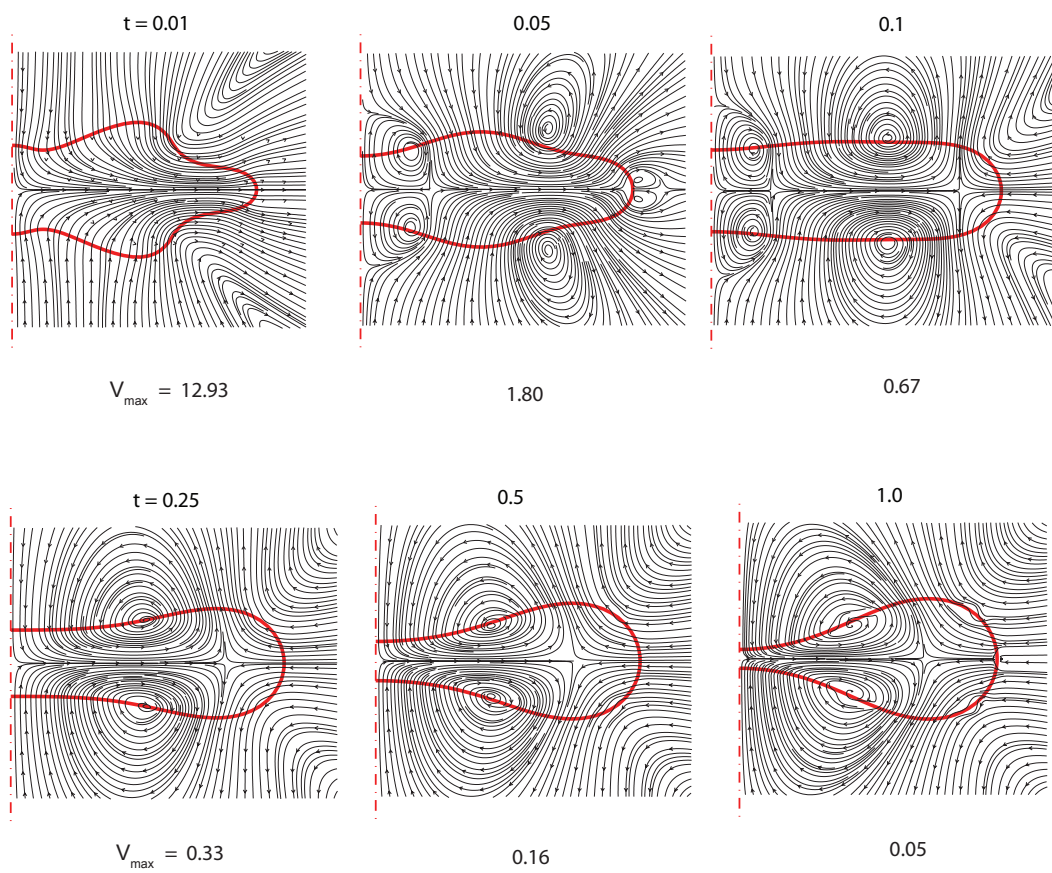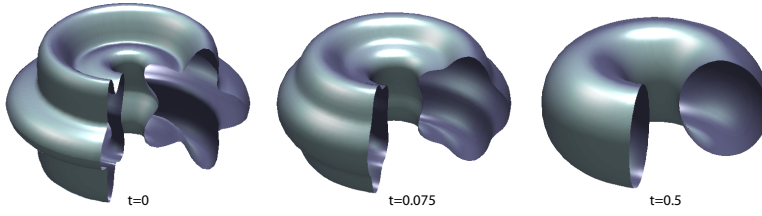
t = 0.01   0.05   0.1

V_{max} = 12.93   1.80   0.67

t = 0.25   0.5   1.0

V_{max} = 0.33   0.16   0.05

Figure 6: Snapshots of streamlines in the plane of symmetry for the simulation in Figure 5.

18

M	S.I.	I.N.
20	1.0e-02	3.0e-03
40	2.5e-03	3.0e-03
60	1.2e-03	3.0e-03
80	6.2e-04	3.0e-03

Figure 7: Toroidal star-shaped vesicle relaxing to a Willmore torus. The difference between the surface at each step and the Willmore torus is magnified by a factor 9. In this case, we have observed that the inexact Newton scheme (I.N.) has better stability properties than the semi-implicit scheme (S.I.). Here, we report the stable time-step sizes, for both the schemes, as a function of the number of spatial discretization points.

reaches an equilibrium shape and translates with a constant velocity. These equilibrium shapes are same as the ones obtained in quiescent fluid suspension.

On the other hand, when $\hat{g}$ is high, the vesicle deforms either to a gourd shape or a stomatocyte-like shape corresponding to prolate and oblate initial shapes respectively. We show two such simulations in Figure 8. The behavior of vesicles in the gravity field will be considered in greater detail in a separate article.
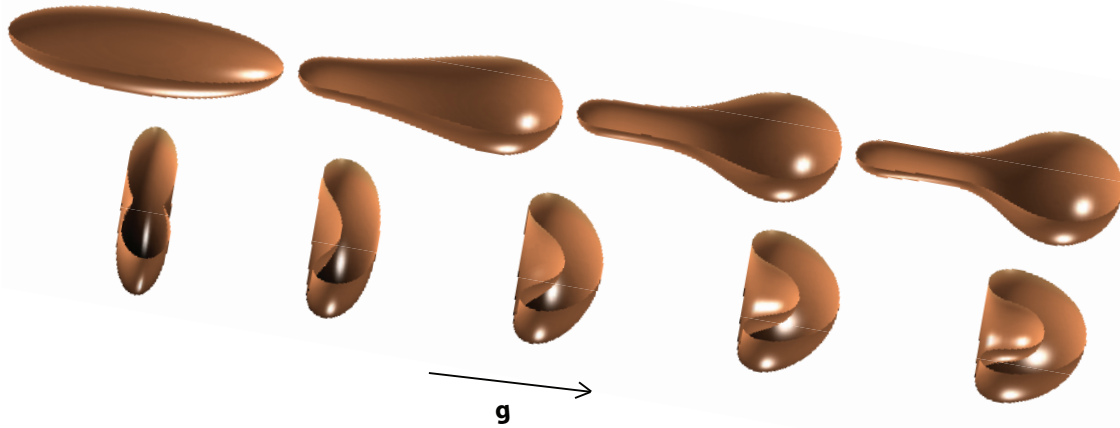

Figure 8: Snapshots of a deforming prolate and an oblate vesicle suspended in a viscous fluid in the presence of gravity. In this example, the reduced volume of the prolate is 0.78, that of the oblate is 0.65 and $\hat{g} = 100$.

Finally, we show a multiple vesicle simulation in Figure 9. The semi-implicit scheme, with the explicit treatment of the interaction forces, has been used for this simmulation.
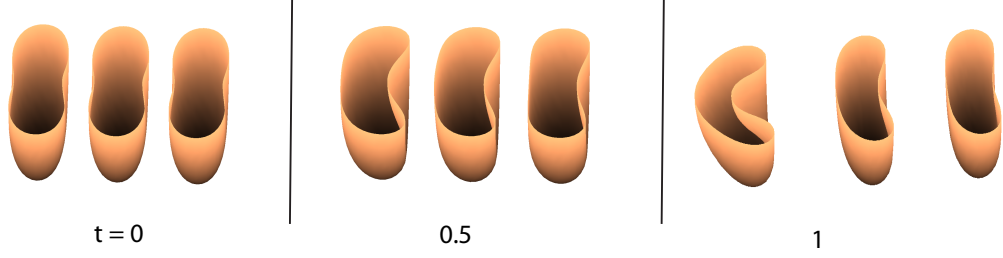
Figure 9: Snapshots of three vesicles suspended in an external parabolic flow.

6   Conclusions and future work

We have presented two numerical schemes for simulating the motion of axisymmetric vesicles of spherical and toroidal topologies in viscous fluid flows. These schemes overcome the stringent restrictions on the time-step size of an explicit scheme with modestly higher computational cost per time-step. We have also introduced a new scheme for computing Stokes potential on an axisymmetric surface. Overall, our schemes achieve high-order accuracy in space and time.

We are currently working on extending these schemes to arbitrary shaped vesicles in 3D. This requires many additional components like accurate surface representations, preconditioners, high-order schemes to compute derivatives and singular quadrature rules for surface integrals, etc.

A   Equilibrium shapes from constrained minimization

Alternative to fluid-formulation, we can obtain the equilibrium shapes by minimizing the bending energy subject to contraints on *global* surface area and volume. Introducing Lagrange multipliers σ and p corresponding to the constraints on area and volume, we can write the Lagrangian as

$$\mathcal{L} = \frac{1}{2} \int_\gamma H^2 d\gamma + \sigma \left(\int_\gamma d\gamma - A \right) + p \left(\int_\Omega d\Omega - V \right). \tag{37}$$

By taking variations, we get the forces due to curvature and constraints. If we start from a non-equilibrium shape, the following is the surface velocity generated by these forces.

$$\dot{\mathbf{x}} = \left(v_\kappa - \sigma(\kappa + \beta) + p \right) \mathbf{n}, \tag{38}$$

$$\text{where} \quad v_\kappa = \frac{1}{2} \left(\triangle_S (\kappa + \beta) + \frac{(\kappa + \beta)(\kappa - \beta)^2}{2} \right).$$

20

The unknowns σ and p, which are functions of time alone, are computed by requiring that the rate of changes in area and volume must vanish.

$$\dot{A} = 2\pi \int_0^\pi \dot{x}_1 |\mathbf{x}_\alpha| + \frac{x_1}{|\mathbf{x}_\alpha|}(x_{1u}\dot{x}_{1u} + x_{2u}\dot{x}_{2u})\, du \tag{39}$$

$$= 2\pi \int_0^\pi \left(\frac{v_1}{x_1} + \mathbf{x}_s \cdot \mathbf{v}_s\right) x_1 |\mathbf{x}_\alpha|\, du = 0. \tag{40}$$

This expression can also be obtained by integrating the surface divergence of the velocity field, that is, $\dot{A} = \int_\gamma \nabla_S \cdot \mathbf{v}_s d\gamma$. If v_n is the component of the velocity normal to the surface, then the rate of change of volume is given by

$$\dot{V} = 2\pi \int_0^\pi v_n x_1 ds = 0. \tag{41}$$

Substituting the velocity (38) in (40) and (41), we get the following expressions for the unknowns σ and p,

$$\sigma = \frac{\langle v_\kappa \rangle \langle \kappa + \beta \rangle - A \langle (\kappa + \beta)v_\kappa \rangle}{\langle \kappa + \beta \rangle^2 - A \langle (\kappa + \beta)^2 \rangle}; \quad p = \frac{\sigma \langle \kappa + \beta \rangle - \langle v_\kappa \rangle}{A}, \quad \text{where} \quad \langle f \rangle = \int_\gamma f\, d\gamma. \tag{42}$$

We can compute the equilibrium shapes by starting from an arbitrary shape, updating the shape using a time-marching scheme on (38) until the surface velocity vanishes. At every time-step, the spatial constants σ and p are computed using (42). We computed the equilibrium shapes using this approach and compared with those obtained by solving (1), and as expected, they match very well.

B Interfacial forces

Here, we give more details on the derivation of the expressions for bending and tension forces in the axisymmetric case. Starting from the special form the positions take in the axisymmetric case (14), we can reduce the fundamental form coefficients to single variable as follows

$$E = \mathbf{x}_u \cdot \mathbf{x}_u = |\mathbf{x}_u|^2, \quad F = \mathbf{x}_u \cdot \mathbf{x}_v = 0, \quad G = \mathbf{x}_v \cdot \mathbf{x}_v = x_1^2.$$

$$W = \sqrt{EG - F^2} = x_1 |\mathbf{x}_u|, \quad \mathbf{n} = (\mathbf{x}_u \times \mathbf{x}_v)/W = \frac{1}{|\mathbf{x}_u|} \begin{bmatrix} -x_{2u}\cos v \\ -x_{2u}\sin v \\ x_{1u} \end{bmatrix} \tag{43}$$

$$L = \mathbf{x}_{uu} \cdot \mathbf{n} = \frac{x_{1u}x_{2uu} - x_{2u}x_{1uu}}{|\mathbf{x}_u|}, \quad M = \mathbf{x}_{uv} \cdot \mathbf{n} = 0, \quad N = \mathbf{x}_{vv} \cdot \mathbf{n} = \frac{x_1 x_{2u}}{|\mathbf{x}_u|}$$

Let κ be the cross-section curvature and s be the arclength parameter. We have $s_u = |\mathbf{x}_u|$ and $\kappa = x_s y_{ss} - y_s x_{ss}$. Let $\beta = \frac{y_s}{x}$, then the curvatures can be compactly written as

$$K = \frac{LN - W^2}{W^2} = \kappa\beta, \quad H = \frac{1}{2}\frac{EN - 2FM + GL}{W^2} = \frac{1}{2}(\kappa + \beta). \tag{44}$$

i.e., κ and β are the principal curvatures. Substituting the expressions for the Gauss and mean curvatures in (11), we get the axisymmetric form of the bending force

$$\mathbf{f}_b = \frac{1}{2}\left(\triangle_S(\kappa + \beta) + \frac{(\kappa + \beta)(\kappa - \beta)^2}{2}\right)\mathbf{n} \tag{45}$$

The Laplace-Beltrami operator, defined in (10) can be simplified to the following form

$$\triangle_S \phi = \frac{1}{W}\left(\frac{G\phi_u}{W}\right)_u = \frac{1}{x_1}(x_1\phi_s)_s \tag{46}$$

$$\mathbf{f}_\sigma = \nabla_s\sigma + \sigma\triangle_S\mathbf{x} = \frac{G}{W^2}\phi_u\mathbf{x}_u - 2H\sigma\mathbf{n} \tag{47}$$

$$= \sigma_s\mathbf{x}_s - \sigma(\kappa + \beta)\mathbf{n} = (\sigma\mathbf{x}_s)_s - \sigma\beta\mathbf{n} \tag{48}$$

Pole conditions. Using the fact that scalar functions in u are even functions and $x_1(u)$ is an odd function, we can compute the limits by Taylor's expansion around zero. As $x_1 \to 0$, we have

$$\beta(u) \to \kappa(u), \quad \triangle_S\phi \to \phi_{ss} \tag{49}$$

and hence $\mathbf{f}_b \to 2\kappa_{ss}\mathbf{n}, \quad \mathbf{f}_\sigma \to (\sigma_s\mathbf{x}_s - 2\sigma\kappa\mathbf{n}).$ (50)

C Stokes kernel

The convolution with the Stokes kernel (defined in (17)) can be computed analytically in the 'v' direction. Here we state the result after introducing the following notation. Let $\mathbf{r} = \begin{bmatrix} y_1\cos v - x_1 \\ y_1\sin v \\ y_2 - x_2 \end{bmatrix}$ and $|\mathbf{r}| = $

$[x_1^2 + y_1^2 - 2x_1 y_1\cos v + (x_2 - y_2)^2]^{1/2}$. Also let,

$$\mathcal{P} = \sqrt{(y_2 - x_2)^2 + (y_1 + x_1)^2}; \quad \mathcal{M} = \sqrt{(y_2 - x_2)^2 + (y_1 - x_1)^2)} \tag{51}$$

$$\text{and} \quad \mathcal{K} = \text{EllipticK}\left(\frac{2\sqrt{y_1 x_1}}{\mathcal{P}}\right); \quad \mathcal{E} = \text{EllipticE}\left(\frac{2\sqrt{y_1 x_1}}{\mathcal{P}}\right) \tag{52}$$

Recalling that $|\mathbf{r}| = [x_1^2 + y_1^2 - 2x_1 y_1\cos v + (x_2 - y_2)^2]^{1/2}$, we have the following,

$$\int_0^{2\pi}\frac{dv}{|\mathbf{r}|} = \frac{4\mathcal{K}}{\mathcal{P}} \tag{53}$$

$$\int_0^{2\pi}\frac{\cos v\, dv}{|\mathbf{r}|} = \frac{2}{y_1 x_1 \mathcal{P}}\left((y_1^2 + x_1^2 + (y_2 - x_2)^2)\mathcal{K} - \mathcal{P}^2\mathcal{E}\right) \tag{54}$$

$$\int_0^{2\pi}\frac{(y_1\cos v - x_1)(y_1 - x_1\cos v)\, dv}{|\mathbf{r}|^3} = \frac{2(y_2 - x_2)^2}{y_1 x_1 \mathcal{P}\mathcal{M}^2}\left(\mathcal{M}^2\mathcal{K} - (y_1^2 + x_1^2 + (y_2 - x_2)^2)\mathcal{E}\right) \tag{55}$$

$$\int_0^{2\pi}\frac{(y_1\cos v - x_1)\, dv}{|\mathbf{r}|^3} = \frac{2}{x_1 \mathcal{P}\mathcal{M}^2}\left(-\mathcal{M}^2\mathcal{K} + (y_1^2 + (y_2 - x_2)^2 - x_1^2)\mathcal{E}\right) \tag{56}$$

$$\int_0^{2\pi}\frac{dv}{|\mathbf{r}|^3} = \frac{4\mathcal{E}}{\mathcal{P}\mathcal{M}^2}. \tag{57}$$

The resulting kernels in (53 - 57) are functions of single variable u. Notice that, as $\mathbf{x} \to \mathbf{y}$, $\mathcal{K}$ and $\mathcal{E}$ become singular since $\frac{2\sqrt{y_1 x_1}}{\mathcal{P}} \to 1$.

D   Derivative accuracy

Theoretically, the error in computing the derivative using (23) decays super-algebraically. However, in practice, the overall error is also dictated by the roundoff errors which grow as $\mathcal{O}(N\epsilon)$, where ϵ is the machine precision. For high-order derivatives, this becomes even more prominent as the roundoff error growth is $\mathcal{O}(N^k\epsilon)$ in computing a derivative of order k. We illustrate this behavior in Figure 10, using the `MATLAB` code of Trefethen [23] for Fourier differentiation (http://www.comlab.ox.ac.uk/nick.trefethen/p7.m).

Since, inherently, the derivative is an ill-conditioned operator, this behavior is typical for most of the numerical methods, in particular, for spectral methods [7].
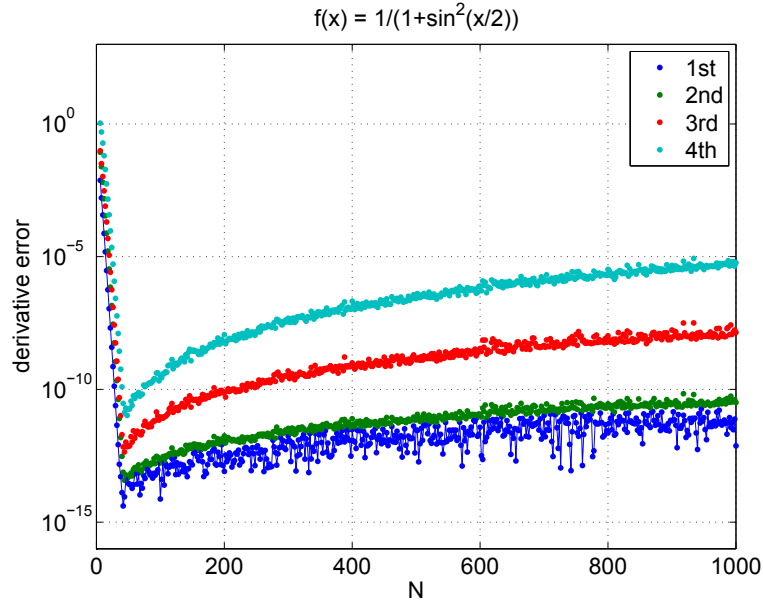

Figure 10: Relative errors in computing $f(x)$ using Fourier differentiation operator.

While we cannot avoid the roundoff errors, we can enhance the accuracy of the force computations by a technique that we shall call *full expansion*. Suppose we numerically computed the functions y_s and x. Let D_h be the discrete Fourier differentiation operator (for odd and even functions defined in equation 23), then we define the expanded form of the scalar function β_s as follows (note that $\beta = \frac{y_s}{x}$)

$$\beta_s \;=\; \frac{1}{s_\alpha} D_h \left(\frac{x_{2s}}{x_1} \right) \quad \text{(non-expanded form)} \tag{58}$$

$$\;=\; \frac{1}{s_\alpha} \left[\frac{(x_1 s_\alpha) D_h(x_{2\alpha}) - D_h(x_1 s_\alpha) x_{2\alpha}}{(x_1 s_\alpha)^2} \right] \quad \text{(expanded form)}. \tag{59}$$

In the expanded form, a Fourier approximation of non-bandlimited functions is minimized. While the gain in accuracy is not substantial for low-order derivatives, it could be significant in computing the bending force

which involves fourth-order derivatives. Similar to equation (59), a fully expanded form of the bending force is obtained by the use of the chain rule to avoid approximating functions of the form $\frac{\phi(u)}{W(u)}$, where $\phi(u)$ is a scalar function. We list the errors in Table 6.

M	non-expanded			fully expanded		
	$\omega = 1$	5	9	1	5	9
20	1.10e-006	8.91e-002	2.65e-001	8.31e-011	3.64e-013	3.10e+000
40	1.20e-010	8.16e-002	7.81e-001	1.32e-009	4.26e-012	3.55e-012
80	1.23e-009	4.05e-003	3.74e-002	4.38e-008	2.44e-011	1.98e-011
160	5.84e-008	1.40e-006	4.29e-003	1.48e-007	8.91e-011	3.14e-011
320	5.03e-007	1.91e-011	4.48e-005	3.42e-005	3.00e-008	1.80e-009

Table 6: Relative errors in computing the bending force using the non-expanded and the fully expanded expressions on a perturbed sphere. The reference values are computed analytically. Here, ω is the order of the perturbation, more specifically, the surface parameters are given by $\mathbf{x}_1(u) = (1+0.1\cos\omega u)\sin u$ and $\mathbf{x}_2(u) = (1 + 0.1\cos\omega u)\cos u$. In both cases, the errors decay rapidly, because of the spectrally convergent scheme, and then start to grow, because of the roundoff errors. However, the fully expanded force computation improves the errors significantly compared to the non-expanded version.

References

[1] Bradley K. Alpert. Hybrid Gauss-trapezoidal quadrature rules. *SIAM Journal on Scientific Computing*, 20(5):1551–1584, 1999.

[2] Uri M. Ascher and Linda R. Petzold. *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1998.

[3] Uri M. Ascher, Steven J. Ruuth, and Brian T. R. Wetton. Implicit-explicit methods for time-dependent partial differential equations. *SIAM Journal on Numerical Analysis*, 32(3):797–823, 1995.

[4] Malcolm I. G. Bloor and Michael J. Wilson. Method for efficient shape parametrization of fluid membranes and vesicles. *Physical Review E*, 61(4):4218–4229, Apr 2000.

[5] Pozrikidis C. Effect of membrane bending stiffness on the deformation of capsules in simple shear flow. *Journal of Fluid Mechanics*, 440:269–291, August 2001.

[6] F. Campelo and A. Hernandez-Machado. Dynamic model and stationary shapes of fluid vesicles. *The European Physical Journal E: Soft Matter and Biological Physics*, 20:37, 2006.

[7] Wai Sun Don and Alex Solomonoff. Accuracy enhancement for higher derivatives using chebyshev collocation and a mapping technique. *SIAM Journal on Scientific Computing*, 18(4):1040–1055, 1997.

[8] Q. Du and J. Zhang. Adaptive finite element method for a phase field bending elasticity model of vesicle membrane deformations. *SIAM Journal On Scientific Computing*, 30(3):1634–1657, 2007.

[9] Qiang Du, Chun Liu, and Xiaoqiang Wang. A phase field approach in the numerical study of the elastic bending energy for vesicle membranes. *Journal of Computational Physics*, 198(2):450–468, 2004.

[10] Feng Feng and William S. Klug. Finite element modeling of lipid bilayer membranes. *Journal of Computational Physics*, 220(1):394–408, 2006.

[11] Martin Kraus, Wolfgang Wintz, Udo Seifert, and Reinhard Lipowsky. Fluid vesicles in shear flow. *Physical Review Letters*, 77(17), 1996.

[12] M. C. A. Kropinski. An efficient numerical method for studying interfacial motion in two-dimensional creeping flows. *Journal of Computational Physics*, 171(2):479–508, 2001.

[13] L. Ma and W. S. Klug. Viscous regularization and r-adaptive remeshing for finite element analysis of lipid membrane mechanics. *Journal Of Computational Physics*, 227(11):5816–5835, 2008.

[14] Henry Power and L. Wrobel. *Boundary Integral Methods in Fluid Mechanics*. Computational Mechanics Publications, 1995.

[15] C. Pozrikidis. Interfacial dynamics for stokes flow. *Journal of Computational Physics*, 169(2):250–301, 2001.

[16] C. Pozrikidis. Axisymmetric motion of a file of red blood cells through capillaries. *Physics Of Fluids*, 17(3):14, 2005.

[17] Yousef Saad. *Iterative Methods for Sparse Linear Systems, 2nd edition*. SIAM, Philadelpha, PA, 2003.

[18] U. Seifert. Configurations of fluid membranes and vesicles. *Advances in Physics*, 46:13–137, January 1997.

[19] Udo Seifert, Karin Berndl, and Reinhard Lipowsky. Shape transformations of vesicles: Phase diagram for spontaneous- curvature and bilayer-coupling models. *Physical Review A*, 44(2):1182–1202, 1991.

[20] S. Sukumaran and U. Seifert. Influence of shear flow on vesicles near a wall: A numerical study. *Physical Review E*, 64, 2001.

[21] Victor A Toponogov. *Differential Geometry of Curves and Surfaces: A Concise Guide*. A Birkhauser book, 2006.

[22] Anna-Karin Tornberg and Leslie Greengard. A fast multipole method for the three-dimensional stokes equations. *Journal of Computational Physics*, 227(3):1613–1619, 2008.

[23] L. N. Trefethen. *Spectral Methods in Matlab*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2000.

[24] M. Mader V. Vitkova and T. Podgorski. Deformation of vesicles flowing through capillaries. *EUROPHYSICS LETTERS*, 68(3):398–404, 2004.

[25] S. K. Veerapaneni, D. Gueyffier, D. Zorin, and G. Biros. A boundary integral method for simulating the dynamics of inextensible vesicles suspended in a viscous fluid in 2d. *Journal of Computational Physics (in print)*, 2008.

[26] J. L. Weiner. On a problem of chen, willmore, et al. *Indiana University Mathematics Journal*, 27:19–35, 1978.