Journal of Computational Physics 228 (2009) 7233–7249

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

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

Shravan K. Veerapaneni [a,*], Denis Gueyffier [b], George Biros [c], Denis Zorin [a]

[a] Courant Institute of Mathematical Sciences, New York University, NY 10012, United States
[b] NASA Goddard Institute for Space Studies, New York, NY 10025, United States
[c] College of Computing, Georgia Institute of Technology, Atlanta, GA 30332, United States

ARTICLE INFO

Article history:
Received 18 March 2009
Received in revised form 7 June 2009
Accepted 19 June 2009
Available online 24 June 2009

Keywords:
Particulate flows
Integral equations
Axisymmetric flows
Numerical methods
Fluid membranes
Inextensible vesicles
Moving boundaries

ABSTRACT

We extend [Shravan K. Veerapaneni, Denis Gueyffier, Denis Zorin, George Biros, A boundary integral method for simulating the dynamics of inextensible vesicles suspended in a viscous fluid in 2D, Journal of Computational Physics 228(7) (2009) 2334–2353] to the case of three-dimensional axisymmetric vesicles of spherical or toroidal topology immersed in viscous flows. Although the main components of the algorithm are similar in spirit to the 2D case—spectral approximation in space, semi-implicit time-stepping scheme—the main differences are that the bending and viscous force require new analysis, the linearization for the semi-implicit schemes must be rederived, a fully implicit scheme must be used for the toroidal topology to eliminate a CFL-type restriction and a novel numerical scheme for the evaluation of the 3D Stokes single layer potential on an axisymmetric surface is necessary to speed up the calculations. By introducing these novel components, we obtain a time-scheme that experimentally is unconditionally stable, has low cost per time step, and is third-order accurate in time. We present numerical results to analyze the cost and convergence rates of the scheme. To verify the solver, we compare it to a constrained variational approach to compute equilibrium shapes that does not involve interactions with a viscous fluid. To illustrate the applicability of method, we consider a few vesicle-flow interaction problems: the sedimentation of a vesicle, interactions of one and three vesicles with a background Poiseuille flow.

© 2009 Elsevier Inc. All rights reserved.

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 and to study the properties of biomembranes. 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 limited attention compared to other types of particulate flows. In [25], we introduced an 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

* Corresponding author. Tel.: +1 2129983510.
E-mail addresses: shravan@cims.nyu.edu, shravan.meam@gmail.com (S.K. Veerapaneni), dgueyffier@giss.nasa.gov (D. Gueyffier), gbiros@gmail.com (G. Biros), dzorin@cims.nyu.edu (D. Zorin).

0021-9991/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2009.06.020

$$\begin{aligned} &\frac{\partial \mathbf{x}}{\partial t} = \mathbf{v}_\infty + \mathcal{S}[\mathbf{f}_b + \mathbf{f}_\sigma] \quad \text{(vesicle position evolution)}, \\ &\mathrm{div}_\gamma \frac{\partial \mathbf{x}}{\partial t} = 0 \quad \text{(surface inextensibility)}, \end{aligned} \tag{1}$$

where γ is the vesicle membrane, div_γ is the surface divergence operator, $\mathbf{x}$ is a Lagrangian 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 Stokes operator, defined in Section 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 nature [19]), eliminating CFL-type time-step restrictions requires a fully implicit time-marching scheme. Our main contribution is the development of efficient numerical schemes for (1) for spherical and toroidal topologies with the following components:

- They require a single linear solve per time step for spherical topology and a small number of nonlinear iterations for toroidal topology;
- They include an efficient preconditioner to enforce the surface-incompressibility constraint;
- They are spectrally accurate in space and third-order accurate in time.

Another 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. For verification, we compare equilibrium shapes obtained using the proposed method with shapes obtained using a variational approach that does not involve computing viscous forces. Finally, to illustrate the capabilities of the method, we consider sedimentation of vesicles under gravity and interactions of multiple vesicles with a background Poiseuille flow.

1.1. Limitations

In the current form, our scheme is not adaptive. Incorporating adaptivity in space and time requires suitable error estimators. In addition, while *p*-type spatial adaptivity can be incorporated with less effort, more fundamental changes to the current scheme are required for *h*-type adaptivity because a non-uniform discretization would require a different approach to compute high-order derivatives accurately.

Our scheme has a mild time-step restriction in the case of shear flows: the stable time-step size is inversely proportional to the shear rate. While one would hope that a fully implicit scheme would eliminate or reduce this restriction, our experiments indicate that even a fully implicit Newton scheme (Section 4.3) does not yield noticeable speed-ups. This is because the Newton iterations do not converge for large time-step sizes. The time-steps for which they do converge are very close to the time-steps for which the semi-implicit scheme is stable.

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.

1.2. 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 [16] (Table 1, p. 289; for vesicles see the "liquid capsules" entry).

Several groups have focused on determining stationary shapes of three-dimensional vesicles using semi-analytic [20,3,5], or numerical methods like the phase-field [8,7] and membrane finite element methods [9,13]. 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 [10,21]. A closely related work is also that of [17], in which, a nearly inextensible interface was considered for the axisymmetric motion of red blood cells inside a cylindrical tube.

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

1.3. Contents

In Section 2, we formulate the integro-differential Eq. (1) that govern vesicle dynamics. The spatial and temporal discretizations are described in Sections 3 and 4, respectively. In Section 5, we present numerical results for a number of problems involv-

ing 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. The verification of the solver and several important details (semi-analytic solutions for the quiescent case, expressions for the force and Stokes convolutions in the axisymmetric case and an analysis of the approximation error for high-order derivatives and ways to improve accuracy) 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 vesicle membrane. The motion of the background fluid is described by the Stokes equations,

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

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

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

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 velocity of the background fluid. The membrane forces of magnitude $\mathbf{f}$ are balanced by a traction jump across the interface γ. That is, if Σ denotes the stress tensor, then

$$[[\Sigma\mathbf{n}]]_\gamma = \mathbf{f}, \tag{4}$$

where $\mathbf{n}$ is the normal to the interface. To derive an expression for $\mathbf{f}$ we have to consider the constitutive properties of the vesicle membrane. The standard assumptions for vesicles [19] consider a surface elastic energy that consists of two terms:

$$\mathcal{E}(H,\sigma) = \int_\gamma \frac{1}{2}\kappa_B H^2 + \sigma\, d\gamma, \tag{5}$$

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 [12],

$$E = \mathbf{x}_u \cdot \mathbf{x}_u, \quad F = \mathbf{x}_u \cdot \mathbf{x}_v, \quad G = \mathbf{x}_v \cdot \mathbf{x}_v \quad \text{(first fundamental form)}, \tag{6}$$

$$L = \mathbf{x}_{uu} \cdot \mathbf{n}, \quad M = \mathbf{x}_{uv} \cdot \mathbf{n}, \quad N = \mathbf{x}_{vv} \cdot \mathbf{n} \quad \text{(second fundamental form)}. \tag{7}$$

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

$$\mathbf{n} = (\mathbf{x}_u \times \mathbf{x}_v)/\sqrt{EG - F^2}, \quad dA = \sqrt{EG - F^2}\, du\, dv = W\, du\, dv. \tag{8}$$

We can now define the mean and Gaussian curvatures as

$$H = \frac{1}{2}\frac{EN - 2FM + GL}{W^2}, \quad K = \frac{LN - M^2}{W^2}. \tag{9}$$

Then, following [26], the gradient or first variation of (5) is given by

$$\Delta\mathcal{E} = \int_\gamma (\Delta_S H + 2H(H^2 - K))\mathbf{n} \cdot \Delta\mathbf{x} - (\sigma\Delta_S\mathbf{x} + \nabla_S\sigma) \cdot \Delta\mathbf{x}\, d\gamma, \tag{10}$$

where Δ_S is the Laplace-Beltrami operator defined by

$$\Delta_S\phi = \frac{1}{W}\left(\left(\frac{E\phi_v - F\phi_u}{W}\right)_v + \left(\frac{G\phi_u - F\phi_v}{W}\right)_u\right), \quad \text{for some scalar function } \phi. \tag{11}$$

From (10), we define the bending and tension forces as

$$\mathbf{f}_b = -(\Delta_S H + 2H(H^2 - K))\mathbf{n}, \quad \mathbf{f}_\sigma = \sigma\Delta_S\mathbf{x} + \nabla_S\sigma. \tag{12}$$

These two forces constitute the interfacial force, that is, $\mathbf{f} = \mathbf{f}_b + \mathbf{f}_\sigma$. Using classical potential theory [15], the solution of (2)–(4), combined with the local inextensibility constraint of the membrane can be written as

$$\dot{\mathbf{x}} = \mathbf{v}_\infty(\mathbf{x}) + \mathcal{S}[\mathbf{f}_b + \mathbf{f}_\sigma](\mathbf{x}) \quad \text{and} \quad \operatorname{div}_\gamma(\mathcal{S}[\mathbf{f}_\sigma]) = -\operatorname{div}_\gamma(\mathbf{v}_\infty + \mathcal{S}[\mathbf{f}_b]). \tag{13}$$

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

$$G(\mathbf{x},\mathbf{y}) = \frac{1}{8\pi\mu}\left(\frac{1}{|\mathbf{r}|}\mathbf{I} + \frac{\mathbf{r}\otimes\mathbf{r}}{|\mathbf{r}|^3}\right), \quad \mathbf{r} = \mathbf{x} - \mathbf{y}. \tag{14}$$

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

$$\mathbf{x} = \begin{bmatrix} x_1(u)\cos v \\ x_1(u)\sin v \\ x_2(u) \end{bmatrix}, \quad \mathbf{f} = \begin{bmatrix} f_1(u)\cos v \\ f_1(u)\sin v \\ f_2(u) \end{bmatrix}. \tag{15}$$

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 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 arc-length 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, β given by $\kappa = x_{1s}x_{2ss} - x_{2s}x_{1ss}$ and $\beta = \frac{x_{2s}}{x_1}$; here, we just state the result:

$$\mathbf{f}_b = \frac{1}{2}\left(\Delta_S(\kappa+\beta) + \frac{(\kappa+\beta)(\kappa-\beta)^2}{2}\right)\mathbf{n}, \quad \mathbf{f}_\sigma = (\sigma\mathbf{x}_s)_s - \sigma\beta\mathbf{n}, \tag{16}$$

$$\text{and at the poles, we have } \lim_{x_1\to 0}\mathbf{f}_b = \kappa_{ss}\mathbf{n}, \quad \lim_{x_1\to 0}\mathbf{f}_\sigma = \sigma_s\mathbf{x}_s - 2\sigma\kappa\mathbf{n}. \tag{17}$$

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

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

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

After further simplification, we get

$$\mathcal{S}[\mathbf{f}] = \begin{bmatrix} F_1 \\ F_2 \end{bmatrix} = \int_0^{2\pi} dv \int_0^\pi du \begin{bmatrix} \frac{\cos v}{|\mathbf{r}|} + \frac{(y_1\cos v - x_1)(y_1 - x_1\cos v)}{|\mathbf{r}|^3} & \frac{(y_1\cos v - x_1)(y_2 - x_2)}{|\mathbf{r}|^3} \\ \frac{(y_1 - x_1\cos v)(y_2 - x_2)|}{|\mathbf{r}|^3} & \frac{1}{|\mathbf{r}|} + \frac{(y_2 - x_2)^2}{|\mathbf{r}|^3} \end{bmatrix}\begin{bmatrix} f_1 \\ f_2 \end{bmatrix} y_1|\mathbf{y}_u|. \tag{18}$$

All the integrals with respect to 'v' are computed analytically using Eqs. (56)–(60). In summary, the axisymmetric form of the 3D Stokes operator is given by

$$\mathcal{S}[\mathbf{f}](\mathbf{x}) = \int_0^\pi K(\mathbf{x}, u)\mathbf{f}(u)y_1(u)|\mathbf{y}_u|\,du. \tag{19}$$

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

2.1.1. 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

$$\mathbf{f}_g = (\rho^{\text{in}} - \rho^{\text{out}})(\mathbf{g}\cdot\mathbf{x})\mathbf{n}. \tag{20}$$

Then, the governing equations that include gravitational forces are

$$\dot{\mathbf{x}} = \mathbf{v}_\infty + \mathcal{S}[\mathbf{f}_b + \mathbf{f}_\sigma + \mathbf{f}_g]; \quad \mathrm{div}_\gamma(\mathcal{S}[\mathbf{f}_\sigma]) = -\,\mathrm{div}_\gamma(\mathbf{v}_\infty + \mathcal{S}[\mathbf{f}_b + \mathbf{f}_g]). \tag{21}$$

2.1.2. Scaling

Following [10], 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 vesicle dynamics are characterized by a single parameter [10], the reduced volume $v = \frac{6\sqrt{\pi}V}{A^{3/2}}$.

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 $\mathbf{v}_\infty = c(w^2 - x_1^2(u))$, 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 parametrizes such external flows.

In the presence of gravity, an additional parameter that governs the vesicle dynamics is the nondimensional gravity parameter, given by $\hat{g} = \frac{(\rho^{\mathrm{in}} - \rho^{\mathrm{out}})gR_0^4}{\kappa_B}$.

2.1.3. Multiple vesicles

The governing equations for the jth-vesicle, in a suspension of N_v vesicles, are given by

$$\dot{\mathbf{x}}_j = \mathbf{v}_\infty(\mathbf{x}_j) + \mathcal{S}_j[\mathbf{f}_b + \mathbf{f}_\sigma](\mathbf{x}_j) + \sum_{\substack{k=1\\k\neq j}}^{N_v} \mathcal{S}_k[\mathbf{f}_b + \mathbf{f}_\sigma](\mathbf{x}_j), \tag{22}$$

$$\mathrm{div}_{\gamma j}(\mathcal{S}_j[\mathbf{f}_\sigma]) = -\mathrm{div}_{\gamma j}\left(\mathbf{v}_\infty(\mathbf{x}_j) + \mathcal{S}_j[\mathbf{f}_b](\mathbf{x}_j) + \sum_{\substack{k=1\\k\neq j}}^{N_v} \mathcal{S}_k[\mathbf{f}_b + \mathbf{f}_\sigma](\mathbf{x}_j)\right). \tag{23}$$

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

To summarize, (22) and (23) give the update and the incompressibility constraint, the forces $\mathbf{f}_b$ and $\mathbf{f}_\sigma$ are given by (16) and (17) and the single layer is given by (18). These equations form a closed system of equations for the positions and tensions.

3. Spatial discretization scheme

We have chosen the spatial discretization scheme to enable 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 (18) involving singular kernels. We use the trigonometric polynomial bases to represent the position of the interface and functions defined on it. The coordinate functions x_1 and x_2 are given by the coefficients $\hat{x}_1(k)$ and $\hat{x}_2(k)$:[1]

$$x_1(u) = \sum_{k=1}^{M} \hat{x}_1(k)\,\sin(ku), \quad x_2(u) = \sum_{k=0}^{M} \hat{x}_2(k)\,\cos(ku),\ u \in [0,\pi]. \tag{24}$$

(Recall that for smoothness $x_1(u)$ is required to be odd and $x_2(u)$ even). Similarly, $\sigma(u) = \sum_{k=0}^{M} \hat{\sigma}(k)\,\cos(ku)$. 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:

$$x_{1u}(u) = \sum_{k=1}^{M} k\hat{x}_1(k)\,\cos(ku), \quad x_{2u}(u) = -\sum_{k=1}^{M} k\hat{x}_2(k)\,\sin(ku). \tag{25}$$

Assuming that the shape of the vesicle is smooth, this derivative approximation is spectrally accurate. We make a few more remarks on the derivative accuracy and the effects of round-off error in the Appendix D.

3.1. Quadrature scheme

The kernels in (19) have a logarithmic singularity, which can be verified by examining their asymptotic expansions. Let $z \in (0,1)$, then we have the following expansions around $z = 0$,

$$\mathrm{EllipticK}(1-z) = c_0 - \tfrac{1}{2}\ln z + (c_1 - \tfrac{1}{4}\ln z)z + (c_2 - \tfrac{5}{32}\ln z)z^2 + \mathcal{O}(z^3) \quad (\textit{first kind}),$$
$$\mathrm{EllipticE}(1-z) = d_0 + (d_1 - \tfrac{1}{2}\ln z)z + (d_2 - \tfrac{1}{8}\ln z)z^2 + \mathcal{O}(z^3) \quad (\textit{second kind}),$$

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

for some constants c_k and $d_k, k \geqslant 0$. 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 (19), 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.

3.1.1. Special quadrature for the poles

Substituting $x_1 = 0$ in (18), we get

$$\mathcal{S}[\mathbf{f}] = \begin{bmatrix} F_1 \\ F_2 \end{bmatrix} = \int_0^{2\pi}\int_0^{\pi} \begin{bmatrix} \frac{\cos v}{|\mathbf{r}|} + \frac{y_1 \cos v}{|\mathbf{r}|^3} & \frac{(y_1 \cos v)(y_2 - x_2)}{|\mathbf{r}|^3} \\ \frac{y_1(y_2 - x_2)}{|\mathbf{r}|^3} & \frac{1}{|\mathbf{r}|} + \frac{(y_2 - x_2)^2}{|\mathbf{r}|^3} \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} y_1 |\mathbf{y}_u|\, du\, dv \tag{26}$$

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:

$$F_1 = 0; \quad F_2 = 2\pi \int_0^{\pi} \frac{y_1^2(y_2 - x_2)}{|\mathbf{r}|^3} f_1 |\mathbf{y}_u| + \left(\frac{y_1}{|\mathbf{r}|} + \frac{y_1(y_2 - x_2)^2}{|\mathbf{r}|^3}\right) f_2 |\mathbf{y}_u|\, du. \tag{27}$$

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

As a starting point, we consider the explicit scheme that has been used by several authors [10,21,4] 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 substantially 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. $\mathcal{Q}(\mathbf{x})\mathbf{x}$. Then, $\mathcal{Q}(\mathbf{x})$ is treated explicitly and $\mathbf{x}$ implicitly. 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 scheme performs very well for spherical vesicles, eliminating the numerical stiffness and delivering orders-of-magnitude computational savings compared to the explicit scheme; and (ii) for toroidal vesicles, the semi-implicit may be inadequate and an implicit scheme is required.

4.1. Explicit scheme

Let us introduce a linear operator $\mathcal{L}$ defined by $\mathcal{L}\sigma = \mathrm{div}_\gamma(\mathcal{S}[\mathbf{f}_\sigma])$. Then, given the current position of the membrane, we first compute the tension by inverting $\mathcal{L}$ and then update the position explicitly. More precisely, let Δt be a fixed time-step size and let the position at $n\Delta t$, 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` $r^n = -\mathrm{div}_\gamma(\mathcal{S}[\mathbf{f}_b^n])$, `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` $\mathbf{x}^{n+1} = \mathbf{x}^n + \Delta t\, \mathcal{S}[\mathbf{f}_b^n + \mathbf{f}_\sigma^n]$

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 Fig. 1. Hence, when a Krylov iterative scheme is used to solve for σ^n, the number of iterations grow proportionally to the number of spatial discretization points. For instance, using the GMRES method [18] to invert $\mathcal{L}$ would require $\mathcal{O}\left(\sqrt{M}\right)$ iterations.

We now describe a preconditioner that eliminates this ill-conditioning. In [25] we showed that, on the 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 pre-

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

Fig. 1. In this table, we report the condition number of the operator $\mathcal{L}$, which we have computed numerically as a function of the spatial discretization size (M) for the vesicle geometry depicted to the left.

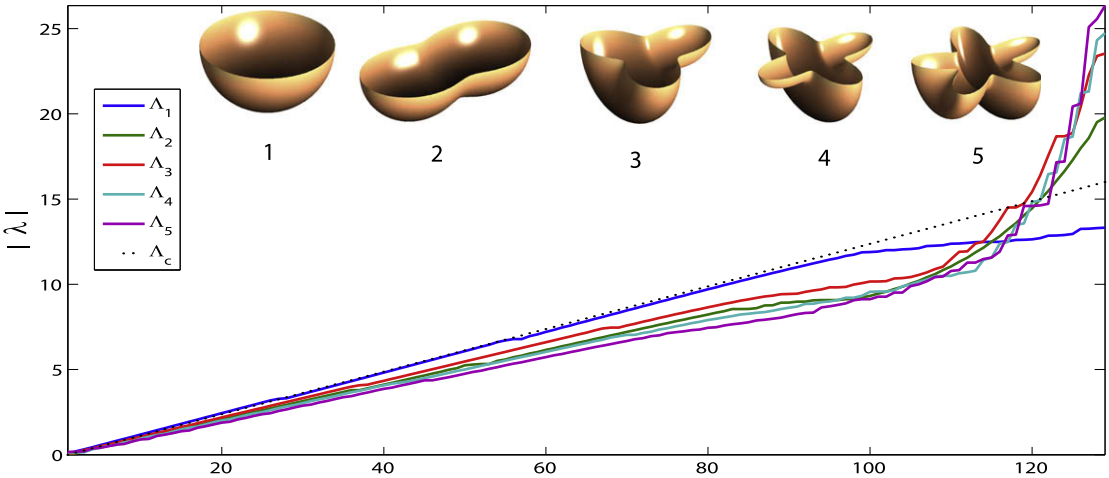


Fig. 2. Plot of the eigenvalue magnitudes of the operator $\mathcal{L}$ defined on vesicle shapes with same surface area. For comparison, we also show the spectrum Λ_c of the corresponding operator defined on the unit circle.

conditioner for solving the inextensibility constraint on a general boundary. Specifically, we proposed a preconditioner P given by (Eq. (22) of [25]):

$$P = \mathcal{F}^{-1}\Lambda_c^{-1}\mathcal{F}, \quad \Lambda_c = \ \mathrm{diag}\left\{\lambda_{-\frac{M}{2}}, \lambda_{-\frac{M}{2}+1}, \ldots, \lambda_{\frac{M}{2}-1}\right\}, \tag{28}$$

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 Fig. 2, we plot the spectrum of the operator $\mathcal{L}$, which we computed numerically for different boundary configurations. We observe that the spectra follow a pattern similar to Λ_c. This motivated the use of P as a preconditioner for the 3D axisymmetric case as well. 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 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 our numerical experiments, we have found that this preconditioner works very well for solving the constraint equation on general axisymmetric 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 time-step size, arising due to the high-order spatial derivatives in the right-hand side of the evolution equation in (1), 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]. Such schemes have been demonstrated to be efficient in many problems of interest in computational physics (e.g. [11,22]). In [25], we proposed two semi-implicit schemes for simulating the 2D vesicle dynamics and showed, using numerical experiments, that they dramat-

[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.

Table 1
Minimum eigenvalues of $Q(\mathbf{x})$ defined on the vesicle geometry shown in Fig. 1 for different spatial discretizations. We computed these values numerically.

M	17	33	65	129	257
λ_{min}	−1.74e+03	−2.25e+04	−2.34e+05	−2.09e+06	−1.74e+07

ically improve the stable time-step sizes over the explicit scheme. The scheme that follows is an extension of those schemes to the axisymmetric case.

The main challenge is to simplify the nonlinear expressions for the forces and to identify an appropriate linearization that is both easy to compute and results in a stable scheme. 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 smooth 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,

$$\mathbf{f}_b(\mathbf{x}) = \frac{1}{2}\Delta_S(\kappa+\beta)\mathbf{n} + \frac{1}{2}(\kappa-\beta)(\kappa-\beta)_s\mathbf{x}_s, \quad \lim_{x_1\to 0}\mathbf{f}_b = \kappa_{ss}\mathbf{n}. \tag{29}$$

Again, we would like to emphasize that, as in the 2D case, we use a modified version of the bending force to simplify the implementation. The force in (29) is composed of a normal and a tangential component. Now, we can easily select the terms with highest order spatial derivatives 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 thus, it has second-order derivatives. Therefore, the two terms in the normal component, $\Delta_S\kappa$ and $\Delta_S\beta$, have fourth-order derivatives in 'u' and hence are candidates for implicit treatment. Similarly, in the tangential component, 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 $n\Delta t$ is known. The goal is to obtain the corresponding positions and tensions, $\{\mathbf{x}^{n+1}(u_k), \sigma^{n+1}(u_k)\}_{k=1}^M$, at $(n+1)\Delta t$. For the simplest (first-order accurate) time discretization, our semi-implicit scheme is

$$\frac{1}{\Delta t}(\mathbf{x}^{n+1} - \mathbf{x}^n) = \mathcal{S}\left[\mathbf{f}_b^{n+1} + \mathbf{f}_\sigma^{n+1}\right](\mathbf{x}^n), \tag{30}$$

$$\mathcal{L}(\mathbf{x}^n)\sigma^{n+1} = -\mathrm{div}_{\gamma^n}\mathcal{S}\left[\mathbf{f}_b^{n+1}\right](\mathbf{x}^n), \tag{31}$$

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.

$$\begin{aligned}
&\mathbf{f}_b^{n+1} = \frac{1}{2}\Delta_S(\kappa^{n+1}+\beta^{n+1})\mathbf{n} + \frac{1}{2}(\kappa-\beta)(\kappa^{n+1}-\beta^{n+1})_s\mathbf{x}_s \quad \textit{bending force},\\
&\mathbf{f}_\sigma^{n+1} = (\sigma^{n+1}\mathbf{x}_s)_s - \sigma^{n+1}\beta\mathbf{n} \quad \textit{tension force},\\
&\kappa^{n+1} = x_{1s}\frac{1}{|\mathbf{x}_u|}\left(\frac{x_{2u}^{n+1}}{|\mathbf{x}_u|}\right)_u - x_{2s}\frac{1}{|\mathbf{x}_u|}\left(\frac{x_{1u}^{n+1}}{|\mathbf{x}_u|}\right)_u, \quad \beta^{n+1} = \frac{1}{x_1}\left(\frac{x_{2u}^{n+1}}{|\mathbf{x}_u|}\right) \quad \textit{curvatures},\\
&\lim_{x_1\to 0}\beta^{n+1} = \frac{1}{x_{1u}}\left(\frac{x_{2u}^{n+1}}{|\mathbf{x}_u|}\right)_u \quad \lim_{x_1\to 0}\Delta_S(\kappa^{n+1}+\beta^{n+1}) = 2(\kappa^{n+1}+\beta^{n+1})_{ss} \quad \textit{pole conditions}.
\end{aligned}$$

Substituting these expressions in (30) and (31), we get a coupled linear system of equations for $\mathbf{x}^{n+1}$ and σ^{n+1}. However, these equations are ill-conditioned and hence are computationally expensive to solve. Next, we discuss preconditioning techniques that can be used to accelerate the linear solves.

Combining (30) and (31), we can write the semi-implicit scheme more compactly as follows

$$\frac{1}{\Delta t}(\mathbf{x}^{n+1} - \mathbf{x}^n) = Q(\mathbf{x}^n)\mathbf{x}^{n+1}, \tag{32}$$

where the operator $Q(\mathbf{x})$ includes all explicit terms from the bending and the inextensibility constraint equations. Because of the bending force term, $Q(\mathbf{x})$ is highly ill-conditioned operator. In Table 1, we list the minimum eigenvalue of Q for a specific boundary configuration. Asymptotically, it grows as $\mathcal{O}(-M^3)$. Therefore, the condition number of the matrix $(I - \Delta t Q)$ grows as $\mathcal{O}(M^3)$ and as a result, the number of GMRES iterations required to solve $(I - \Delta t Q)\mathbf{x}^{n+1} = \mathbf{x}^n$ grow as $\mathcal{O}(M^{3/2})$. To avoid this increase in the number of iterations, we design a preconditioner. While computing the inverse of $(I - \Delta t Q)$ is difficult even for simpler geometries, it turns out that mesh-independent behavior can be achieved using a preconditioner based on the analytic spectrum Λ_c [25] of the two-dimensional bending operator for a unit circle (similar to the preconditioner for the constraint equation). The preconditioner P_t, for (30) is defined as [3]

[3] Again, note that, in order to use P_t, we first need to extend the parametric domain from $[0, \pi]$ to $[0, 2\pi]$.

$$P_t = \begin{bmatrix} \mathcal{F}^{-1}(1-\Delta t \Lambda_c)^{-1}\mathcal{F} & 0 \\ 0 & \mathcal{F}^{-1}(1-\Delta t \Lambda_c)^{-1}\mathcal{F} \end{bmatrix}, \tag{33}$$

$$\text{where } \Lambda_c = \operatorname{diag}\left\{\lambda_{-\frac{M}{2}}, \lambda_{-\frac{M}{2}+1}, \ldots, \lambda_{\frac{M}{2}-1}\right\}, \quad \lambda_k = -\frac{|k|^3}{8}. \tag{34}$$

We use the GMRES method to solve the coupled linear set of Eqs. (30) and (31), iterating on the Schur complement of the position unknown and at each iteration we need to invert the inextensibility operator. We use P (defined in (28)) as a preconditioner for the constraint Eq. (31) and P_t as a preconditioner for the evolution Eq. (30). The total cost per time-step of this scheme exceeds that of an explicit scheme by a factor that is equal to the number of iterations required to solve (30). Through numerical experiments, we observed that this number is nearly mesh-independent. This preconditioner is applicable on spherical vesicles only. We have not derived preconditioners for the case of toroidal vesicles.

The background velocity and the gravitational force are treated explicitly. 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]. In the case of spherical vesicles, this scheme overcomes the 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

Following Eq. (32), we define the Jacobian (J) and residual (R) as follows:

$$J(\mathbf{x}) = 1 - \Delta t\, Q(\mathbf{x}); \quad R^n(\mathbf{x}) = J(\mathbf{x})\mathbf{x} - \mathbf{x}^n. \tag{35}$$

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 $\widetilde{J} = J(\mathbf{x}^n)$	*compute inexact Jacobian*
2.	$\mathbf{x}_0 = \widetilde{J}^{-1}\mathbf{x}^n$	*initial guess*
3.	$k = 0$	
4.	while $\min_\lambda \|R^n[\mathbf{x}_k - \lambda\widetilde{J}^{-1}R^n(\mathbf{x}_k)]\| > \epsilon\|R^n(\mathbf{x}_k)\|$ & $k <$ MaxIts	*check for residual convergence*
5.	$\quad \mathbf{p} = -\widetilde{J}^{-1}R^n(\mathbf{x}_k)$	*determine descent direction*
6.	$\quad \lambda_m = \min_\lambda \|R^n(\mathbf{x}_k + \lambda\mathbf{p})\|$	*line search*
7.	$\quad \mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_m\mathbf{p}$	
8.	$\quad k = k+1$	
9.	end while	
10.	$\mathbf{x}^{n+1} = \mathbf{x}_k$	*update positions*

Here, ϵ is the desired tolerance on the residual and MaxIts are the maximum number of Newton iterations per time-step. For more details on inexact Newton methods, see [14].

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

$$\mathbf{x}(u,v) = \begin{bmatrix} x_1(u)\cos v \\ x_1(u)\sin v \\ x_2(u) \end{bmatrix}, \begin{matrix} x_1(u) = \left(\sqrt{\cos^2 u + 9\sin^2 u} + \cos^2 4u\right)\sin u \\ x_2(u) = -\frac{1}{2}\left(\sqrt{\cos^2 u + 9\sin^2 u} + \cos^2 4u\right)\cos u \end{matrix}, \quad u \in [0,\pi],\ v \in [0, 2\pi]. \tag{36}$$

In Fig. 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 2, we report the errors in computing the single layer potential using a fourth-order quadrature scheme described in Section 3.

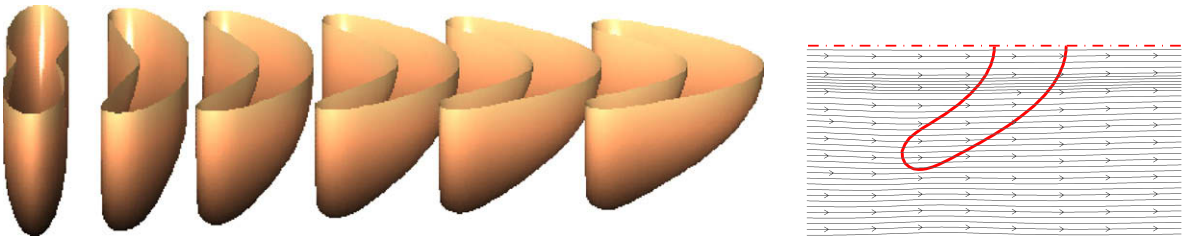
Example 2. In the second example, we consider the motion of a vesicle suspended in a external parabolic flow, shown in Fig. 4. The surface parameters of the initial vesicle shape are given by

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

Fig. 3. Relative errors in computing the principal curvatures κ and β numerically of the shape shown. Specifically, we compute the curvatures at the discrete points $\{u_k = \frac{\pi k}{M-1}\}_{k=1}^{M}$ and report the errors $\frac{\max_k|\kappa(u_k)-\kappa^*(u_k)|}{\max_k|\kappa(u_k)|}$ (similarly errors in β) for different discretization sizes. The reference values (κ^* and β^*) are computed analytically. As expected, we observe spectral convergence.

Table 2
Relative errors in computing $\mathcal{S}[\mathbf{n}]$, defined on the boundary shown in Fig. 3, for different spatial discretization sizes (M). 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 ($M = 513$).

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



(a) (b)

Fig. 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 and (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 3, accuracy in Table 5 and the performance of the preconditioner in Table 6.

$$x_1(u) = \left(\sqrt{\cos^2 u + 9\sin^2 u}\right)\sin u, \quad x_2(u) = -\frac{1}{3}\left(\sqrt{\cos^2 u + 9\sin^2 u}\right)\cos u. \tag{37}$$

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 flow, the resultant equilibrium shape is similar to the ones obtained through numerical experiments in [24] for capillary flows.

We study the stability and convergence properties of various schemes based on this simulation. In Table 3, we list the maximum allowable time-step size $\Delta t_{\max}$ for the explicit and the semi-implicit schemes. We determine $\Delta t_{\max}$ by starting

Table 3
Stable (nondimensional) time-step sizes for first-order explicit and semi-implicit schemes for a vesicle whose initial configuration is shown in Fig. 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	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

from an arbitrarily 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 (this explains the repeated numbers in Table 3). We can infer from the table that the explicit scheme requires $\Delta t_{\max}$ to be inversely proportional to the cube of M (approximately). On the other hand, $\Delta t_{\max}$ is independent of M for the semi-implicit scheme.

Notice, however, that $\Delta t_{\max}$, in the case of the semi-implicit scheme, is inversely proportional to the shear rate. The inexact Newton scheme, described in Section 4.3, has similar behavior (see Table 4). When we tried to use a larger time step, the nonlinear iterations did not converge.

In all our numerical experiments with vesicles of spherical topology, we observed that the semi-implicit scheme performs as good 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 of choice for spherical topology.

Since the interior of the vesicle is filled with an incompressible fluid, the enclosed volume is preserved. As the surface is locally inextensible, the total surface area must also be preserved. In Table 5, we report the relative errors in preserving the total volume and surface area of the vesicle shown in Fig. 4.

Next, for the same simulation, we study the performance of the preconditioner for inverting the inextensibility constraint. In Table 6, 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 $\mathcal{O}(1)$. Finally, in Table 7, we report the performance of the preconditioner for the evolution equation.

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

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 5
Surface area and the enclosed volume must be preserved in a vesicle simulation. In this Table, we report the relative errors in the area (A) and volume (V) conservation for the simulation shown in Fig. 4. Here, A_f and V_f are the area and volume, respectively, measured at the end of the simulation, q is the convergence order of the semi-implicit scheme, M is the number of spatial discretization points and $\Delta t = \frac{1}{M}$.

M	$\frac{\lvert A_f - A\rvert}{A}$		$\frac{\lvert V_f - V\rvert}{V}$	
	$q = 1$	$q = 3$	$q = 1$	$q = 3$
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 6
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.

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 7
Number of GMRES iterations to solve the discrete evolution Eq. (32) for two cases: (i) without using any preconditioner and (ii) using the preconditioner P_t defined in (33). Notice that, asymptotically, they grow superlinearly for the former case and are nearly constant for the later case. These values are for the simulation shown in Fig. 4. The GMRES tolerance, in this example, is set to 10^{-6}.

M	17	33	65	129	257	513
None	5	12	30	72	174	423
P_t	6	11	13	15	15	15

Example 3. We consider a vesicle with oscillatory initial shape, defined in Eq. (36) and simulate its motion to equilibrium in the absence of an external flow. We show snapshots in Fig. 5. The advantage of the semi-implicit scheme is clear in this example: we can simulate the dynamics with a drastically smaller number of time-steps in comparison to an explicit method.

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

$$\mathbf{v}(\mathbf{x}) = \int_{\gamma} G(\mathbf{x},\mathbf{x}')[\mathbf{f}_b(\mathbf{x}') + \mathbf{f}_\sigma(\mathbf{x}')]\, d\gamma(\mathbf{x}'), \quad \mathbf{x} \in \mathbf{R}^3. \tag{38}$$

We plot the streamlines corresponding to this simulation in Fig. 6.

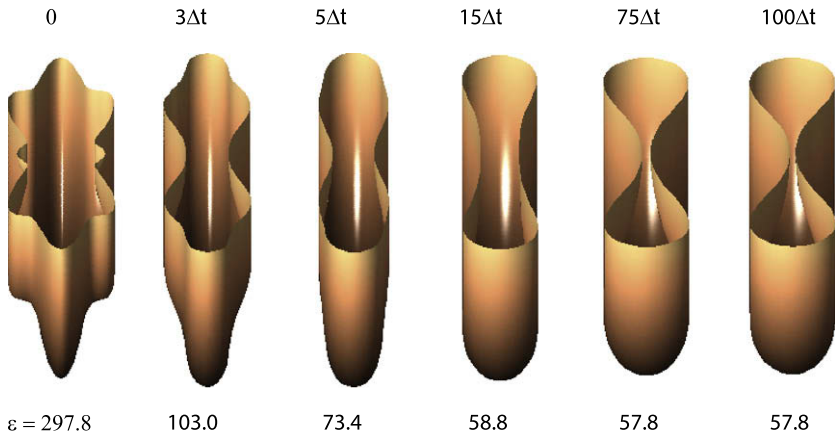


Fig. 5. Snapshots of a freely suspended vesicle, with complicated initial shape, relaxing to equilibrium. Here, $\varepsilon = \frac{\pi\kappa_B}{4}\int_0^{\pi}(\kappa+\beta)^2 x_1 ||\mathbf{x}_u||\, du$, the bending energy. We require $M = 128$ to resolve the initial shape to double precision ($\epsilon = 10^{-12}$). 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.

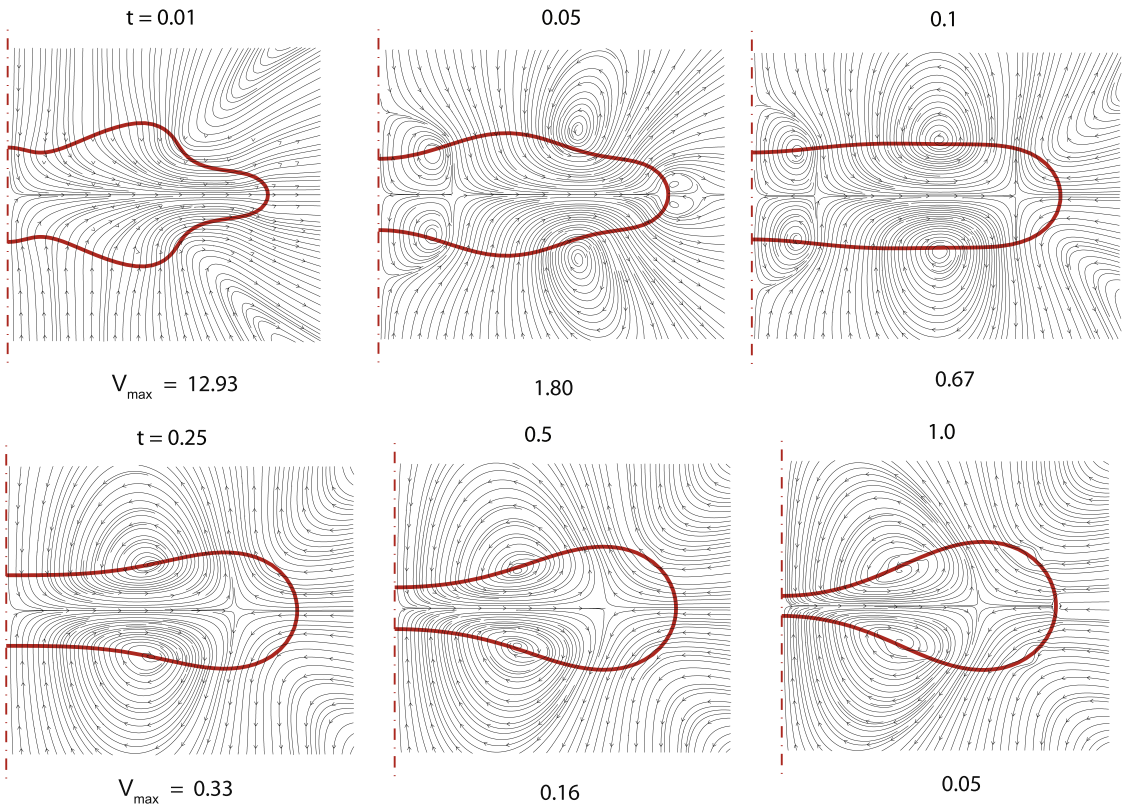


Fig. 6. Snapshots of streamlines in the plane of symmetry for the simulation in Fig. 5.

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

$$x_1(u) = (1 + 0.03\cos 5u)\cos u, \quad x_2(u) = (1 + 0.03\cos 5u)\sin u. \tag{39}$$

In Fig. 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. As mentioned before, the fully implicit scheme outperforms the semi-implicit scheme for toroidal geometries.

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 reaches an equilibrium shape and translates with a constant velocity. These equilibrium shapes are the 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 Fig. 8. The behavior of vesicles in the gravity field will be considered in greater detail in a separate article.

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

Fig. 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.

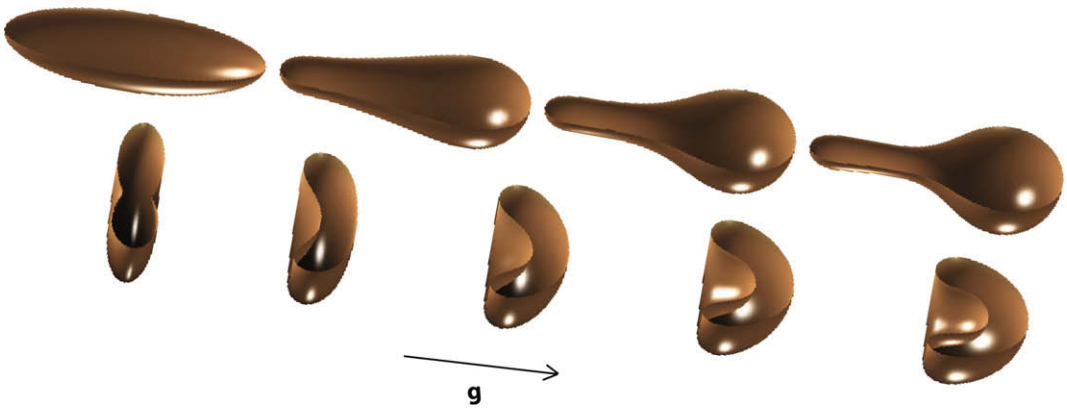


Fig. 8. Snapshots of a deforming prolate and an oblate vesicle suspended in a viscous fluid in the presence of gravity. In this example, the reduced volumes of the prolate of the oblate vesicles are 0.78 and 0.65, respectively; $\hat{g} = 100$.

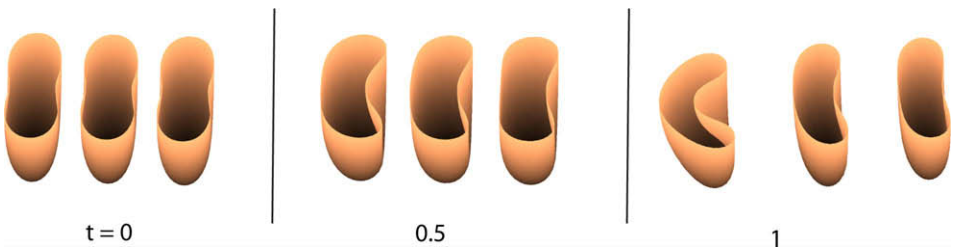


Fig. 9. Snapshots of interacting vesicles suspended in an external parabolic flow.

Finally, we show a multiple vesicle simulation in Fig. 9. The semi-implicit scheme, with the explicit treatment of the interaction forces, has been used for this simulation.

6. Conclusions and future work

We have presented a numerical scheme for simulating the motion of axisymmetric vesicles of spherical and toroidal topologies in viscous fluid flows. Our numerical 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, the method achieves 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 and high-order accurate calculation of derivatives.

Acknowledgements

This work was supported by the US Department of Energy under Grants DEFG0200ER25053 and DE-FG02-04ER25646, and the US National Science Foundation Grants CCF-0427985 and OCI 0749285.

Appendix A. Equilibrium shapes from constrained minimization

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

$$\mathcal{L} = \frac{1}{2}\int_\gamma H^2 d\gamma + \sigma\left(\int_\gamma d\gamma - A\right) + p\left(\int_\Omega d\Omega - V\right). \tag{40}$$

By taking variations, we obtain the forces due to bending and the constraints. To find the equilibrium shape, we can use a steepest descent-like algorithm in which the pseudo-velocity of the boundary is given by

$$\frac{\partial \mathbf{x}}{\partial t} = (v_\kappa - \sigma(\kappa+\beta) + p)\mathbf{n}, \tag{41}$$

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

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

$$\dot{A} = 2\pi\int_0^\pi \dot{x}_1|\mathbf{x}_u| + \frac{x_1}{|\mathbf{x}_u|}(x_{1u}\dot{x}_{1u} + x_{2u}\dot{x}_{2u})\,du \tag{42}$$

$$= 2\pi\int_0^\pi \left(\frac{v_1}{x_1} + \mathbf{x}_s\cdot\mathbf{v}_s\right)x_1|\mathbf{x}_u|\,du = 0. \tag{43}$$

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

$$\dot{V} = 2\pi\int_0^\pi v_n x_1|\mathbf{x}_u|\,du = 0. \tag{44}$$

Substituting the velocity (41) in (43) and (44) and solving for the unknowns σ and p, we get the following

$$\sigma = \frac{\langle v_\kappa\rangle\langle\kappa+\beta\rangle - A\langle(\kappa+\beta)v_\kappa\rangle}{\langle\kappa+\beta\rangle^2 - A\langle(\kappa+\beta)^2\rangle}; \quad p = \frac{\sigma\langle\kappa+\beta\rangle - \langle v_\kappa\rangle}{A}, \qquad \text{where } \langle f\rangle := \int_\gamma f\,d\gamma. \tag{45}$$

We can compute the equilibrium shapes by starting from an arbitrary shape, updating the shape using a time-marching scheme on (41) until the surface velocity vanishes. At every time-step, the spatial constants σ and p are computed using (45). We computed the equilibrium shapes using this approach and compared with those obtained by solving (1) and as expected, they match very well.

Appendix 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 (15), we can reduce the fundamental form coefficients to single variable as follows

$$E = \mathbf{x}_u \cdot \mathbf{x}_u = |\mathbf{x}_u|^2, \quad F = \mathbf{x}_u \cdot \mathbf{x}_v = 0, \quad G = \mathbf{x}_v \cdot \mathbf{x}_v = x_1^2.$$
$$W = \sqrt{EG - F^2} = x_1|\mathbf{x}_u|, \quad \mathbf{n} = (\mathbf{x}_u \times \mathbf{x}_v)/W = \frac{1}{|\mathbf{x}_u|}\begin{bmatrix} -x_{2u}\cos v \\ -x_{2u}\sin v \\ x_{1u} \end{bmatrix} \tag{46}$$
$$L = \mathbf{x}_{uu} \cdot \mathbf{n} = \frac{x_{1u}x_{2uu} - x_{2u}x_{1uu}}{|\mathbf{x}_u|}, \quad M = \mathbf{x}_{uv} \cdot \mathbf{n} = 0, \quad N = \mathbf{x}_{vv} \cdot \mathbf{n} = \frac{x_1 x_{2u}}{|\mathbf{x}_u|}$$

Let κ be the cross-section curvature and s be the arc-length parameter. We have $s_u = |\mathbf{x}_u|$ and $\kappa = x_{1s}x_{2ss} - x_{2s}x_{1ss}$. Let $\beta = \frac{x_{2s}}{x_1}$, then the curvatures can be compactly written as

$$K = \frac{LN - W^2}{W^2} = \kappa\beta, \quad H = \frac{1}{2}\frac{EN - 2FM + GL}{W^2} = \frac{1}{2}(\kappa + \beta). \tag{47}$$

i.e., κ and β are the principal curvatures. Substituting the expressions for the Gauss and mean curvatures in (12), we get the axisymmetric form of the bending force

$$\mathbf{f}_b = \frac{1}{2}\left(\Delta_S(\kappa + \beta) + \frac{(\kappa + \beta)(\kappa - \beta)^2}{2}\right)\mathbf{n}. \tag{48}$$

The Laplace-Beltrami operator, defined in (11) can be simplified to the following form

$$\Delta_S \phi = \frac{1}{W}\left(\frac{G\phi_u}{W}\right)_u = \frac{1}{x_1}(x_1\phi_s)_s \tag{49}$$

Finally, the tension forces become

$$\mathbf{f}_\sigma = \nabla_s \sigma + \sigma\Delta_S \mathbf{x} = \frac{G}{W^2}\sigma_u \mathbf{x}_u - 2H\sigma\mathbf{n} \tag{50}$$
$$= \sigma_s \mathbf{x}_s - \sigma(\kappa + \beta)\mathbf{n} = (\sigma\mathbf{x}_s)_s - \sigma\beta\mathbf{n} \tag{51}$$

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

$$\beta(u) \to \kappa(u), \quad \Delta_S\phi \to 2\phi_{ss} \tag{52}$$
$$\text{and hence } \mathbf{f}_b \to 2\kappa_{ss}\mathbf{n}, \quad \mathbf{f}_\sigma \to (\sigma_s\mathbf{x}_s - 2\sigma\kappa\mathbf{n}). \tag{53}$$

Appendix C. Stokes kernel

The convolution with the Stokes kernel (defined in (18)) can be computed analytically in the 'v' direction. Here, we state the result after introducing the following notation. Let

$$\mathcal{P} = \sqrt{(y_2 - x_2)^2 + (y_1 + x_1)^2}; \quad \mathcal{M} = \sqrt{(y_2 - x_2)^2 + (y_1 - x_1)^2)} \tag{54}$$
$$\text{and } \mathcal{K} = \text{EllipticK}\left(\frac{2\sqrt{y_1x_1}}{\mathcal{P}}\right); \quad \mathcal{E} = \text{EllipticE}\left(\frac{2\sqrt{y_1x_1}}{\mathcal{P}}\right). \tag{55}$$

Recalling that $|\mathbf{r}| = \left[x_1^2 + y_1^2 - 2x_1y_1\cos v + (x_2 - y_2)^2\right]^{1/2}$, we obtain

$$\int_0^{2\pi} \frac{dv}{|\mathbf{r}|} = \frac{4\mathcal{K}}{\mathcal{P}}, \tag{56}$$
$$\int_0^{2\pi} \frac{\cos v\, dv}{|\mathbf{r}|} = \frac{2}{y_1x_1\mathcal{P}}\left(\left(y_1^2 + x_1^2 + (y_2 - x_2)^2\right)\mathcal{K} - P^2\mathcal{E}\right), \tag{57}$$
$$\int_0^{2\pi} \frac{(y_1\cos v - x_1)(y_1 - x_1\cos v)\, dv}{|\mathbf{r}|^3} = \frac{2(y_2 - x_2)^2}{y_1x_1\mathcal{P}\mathcal{M}^2}\left(\mathcal{M}^2\mathcal{K} - \left(y_1^2 + x_1^2 + (y_2 - x_2)^2\right)\mathcal{E}\right), \tag{58}$$
$$\int_0^{2\pi} \frac{(y_1\cos v - x_1)\, dv}{|\mathbf{r}|^3} = \frac{2}{x_1\mathcal{P}\mathcal{M}^2}\left(-\mathcal{M}^2\mathcal{K} + \left(y_1^2 + (y_2 - x_2)^2 - x_1^2\right)\mathcal{E}\right), \tag{59}$$
$$\int_0^{2\pi} \frac{dv}{|\mathbf{r}|^3} = \frac{4\mathcal{E}}{\mathcal{P}\mathcal{M}^2}. \tag{60}$$

The resulting kernels in (56)–(60) are functions of single variable u. Notice that, as $\mathbf{x} \to \mathbf{y}$, $\mathcal{K}$ and $\mathcal{E}$ become singular since $\frac{2\sqrt{y_1x_1}}{\mathcal{P}} \to 1$.

Appendix D. Derivative accuracy

Theoretically, the error in computing the derivative using (25) should decay super-algebraically. However, in practice, the overall error is also dictated by the round-off 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 round-off error growth is $\mathcal{O}(N^k\epsilon)$ in computing a derivative of order k. We illustrate this behavior in Fig. 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 [6].

While we cannot avoid the round-off 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 x_{2s} and x. Let D_h be the discrete Fourier differentiation operator (for odd and even functions defined in Eq. (25)), then we define the expanded form of the scalar function β_s as follows (note that $\beta = \frac{x_{2s}}{x}$)

$$\beta_s = \frac{1}{s_u} D_h\left(\frac{x_{2s}}{x_1}\right) \quad \text{(non-expanded form)} \tag{61}$$

$$= \frac{1}{s_u}\left[\frac{(x_1 s_u) D_h(x_{2u}) - D_h(x_1 s_u) x_{2u}}{(x_1 s_u)^2}\right] \quad \text{(expanded form)}. \tag{62}$$

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 Eq. (62), 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 8.

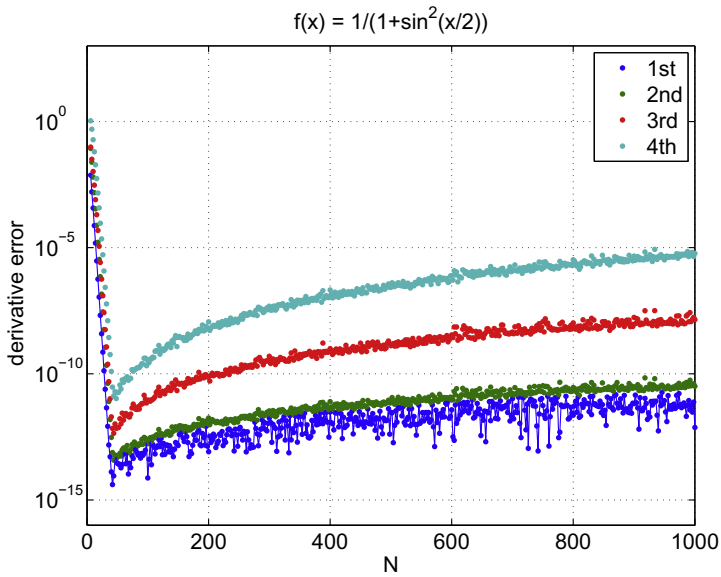


Fig. 10. Relative errors in computing $f(x)$ using Fourier differentiation operator.

Table 8
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 round-off errors. However, the fully expanded force computation improves the errors substantially compared to the non-expanded version.

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

References

[1] Bradley K. Alpert, Hybrid Gauss-trapezoidal quadrature rules, SIAM Journal on Scientific Computing 20 (5) (1999) 1551–1584.
[2] Uri M. Ascher, Linda R. Petzold, Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1998.
[3] Malcolm I.G. Bloor, Michael J. Wilson, Method for efficient shape parametrization of fluid membranes and vesicles, Physical Review E 61 (4) (2000) 4218–4229.
[4] C. Pozrikidis, Effect of membrane bending stiffness on the deformation of capsules in simple shear flow, Journal of Fluid Mechanics 440 (August) (2001) 269–291.
[5] F. Campelo, A. Hernandez-Machado, Dynamic model and stationary shapes of fluid vesicles, The European Physical Journal E: Soft Matter and Biological Physics 20 (2006) 37.
[6] Wai Sun Don, Alex Solomonoff, Accuracy enhancement for higher derivatives using Chebyshev collocation and a mapping technique, SIAM Journal on Scientific Computing 18 (4) (1997) 1040–1055.
[7] Q. Du, J. Zhang, Adaptive finite element method for a phase field bending elasticity model of vesicle membrane deformations, SIAM Journal on Scientific Computing 30 (3) (2007) 1634–1657.
[8] Qiang Du, Chun Liu, Xiaoqiang Wang, A phase field approach in the numerical study of the elastic bending energy for vesicle membranes, Journal of Computational Physics 198 (2) (2004) 450–468.
[9] Feng Feng, William S. Klug, Finite element modeling of lipid bilayer membranes, Journal of Computational Physics 220 (1) (2006) 394–408.
[10] Martin Kraus, Wolfgang Wintz, Udo Seifert, Reinhard Lipowsky, Fluid vesicles in shear flow, Physical Review Letters 77 (17) (1996).
[11] M.C.A. Kropinski, An efficient numerical method for studying interfacial motion in two-dimensional creeping flows, Journal of Computational Physics 171 (2) (2001) 479–508.
[12] Andrei Ludu, Nonlinear Waves and Solitons on Contours and Closed Surfaces, Springer Series in Synergetics, 2007.
[13] L. Ma, W.S. Klug, Viscous regularization and r-adaptive remeshing for finite element analysis of lipid membrane mechanics, Journal of Computational Physics 227 (11) (2008) 5816–5835.
[14] Jorge Nocedal, Stephen J. Wright, Numerical Optimization, Springer Series in Operations Research, second ed., 2006.
[15] Henry Power, L. Wrobel, Boundary Integral Methods in Fluid Mechanics, Computational Mechanics Publications, 1995.
[16] C. Pozrikidis, Interfacial dynamics for Stokes flow, Journal of Computational Physics 169 (2) (2001) 250–301.
[17] C. Pozrikidis, Axisymmetric motion of a file of red blood cells through capillaries, Physics of Fluids 17 (3) (2005) 14.
[18] Yousef Saad, Iterative Methods for Sparse Linear Systems, second ed., SIAM, Philadelphia, PA, 2003.
[19] U. Seifert, Configurations of fluid membranes and vesicles, Advances in Physics 46 (January) (1997) 13–137.
[20] Udo Seifert, Karin Berndl, Reinhard Lipowsky, Shape transformations of vesicles: phase diagram for spontaneous- curvature and bilayer-coupling models, Physical Review A 44 (2) (1991) 1182–1202.
[21] S. Sukumaran, U. Seifert, Influence of shear flow on vesicles near a wall: a numerical study, Physical Review E 64 (2001).
[22] A.-K. Tornberg, M.J. Shelley, Simulating the dynamics and interactions of flexible fibers in Stokes flows, Journal of Computational Physics 196 (May) (2004) 8–40.
[23] L.N. Trefethen, Spectral Methods in Matlab, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2000.
[24] M. Mader, V. Vitkova, T. Podgorski, Deformation of vesicles flowing through capillaries, Europhysics Letters 68 (3) (2004) 398–404.
[25] Shravan K. Veerapaneni, Denis Gueyffier, Denis Zorin, George Biros, A boundary integral method for simulating the dynamics of inextensible vesicles suspended in a viscous fluid in 2D, Journal of Computational Physics 228 (7) (2009) 2334–2353.
[26] J.L. Weiner, On a problem of Chen, Willmore, et al., Indiana University Mathematics Journal 27 (1978) 19–35.