

# A fast algorithm for simulating vesicle flows in three dimensions

Shravan K. Veerapaneni\*, Abtin Rahimian†, George Biros‡ and Denis Zorin§

November 12, 2010

## Abstract

Vesicles are locally-inextensible fluid membranes that can sustain bending. In this paper, we extend “*A numerical method for simulating the dynamics of 3D axisymmetric vesicles suspended in viscous flows*”, Veerapaneni et al. *Journal of Computational Physics*, 228(19), 2009 to general non-axisymmetric vesicle flows in three dimensions.

Although the main components of the algorithm are similar in spirit to the axisymmetric case (spectral approximation in space, semi-implicit time-stepping scheme), important new elements need to be introduced for a full 3D method. In particular, spatial quantities are discretized using spherical harmonics, and quadrature rules for singular surface integrals need to be adapted to this case; an algorithm for surface reparameterization is needed to ensure sufficient of the time-stepping scheme, and spectral filtering is introduced to maintain reasonable accuracy while minimizing computational costs. To characterize the stability of the scheme and to construct preconditioners for the iterative linear system solvers used in the semi-implicit time-stepping scheme, we perform a spectral analysis of the evolution operator on the unit sphere.

By introducing these algorithmic components, we obtain a time-stepping scheme that, in our numerical experiments, is unconditionally stable. We present results to analyze the cost and convergence rates of the overall scheme. To illustrate the applicability of the new method, we consider a few vesicle-flow interaction problems: a single vesicle in relaxation, sedimentation, shear flows, and many-vesicle flows.

## 1 Introduction

Vesicles (also known as fluid membranes) are closed phospholipid membranes suspended in a viscous solution. They are found 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 in the study biomembrane mechanics. Vesicle-inspired mechanical models can be used to approximate red blood cell mechanics. For example, at equilibrium (i.e., in a quiescent fluid), healthy red blood cells have a biconcave shape that corresponds to a minimal membrane bending energy. Under nonequilibrium conditions, as experienced in a simple shear flow, the best-studied features of red cell dynamics, formation of tank-treading ellipsoids and tumbling motion, are shared with vesicles [6, 33, 35].

---

\*Courant Institute of Mathematical Sciences, New York University, New York 10012, [shravan@cims.nyu.edu](mailto:shravan@cims.nyu.edu)

†College of Computing, Georgia Institute of Technology, Atlanta GA 30332, [rahimian@gatech.edu](mailto:rahimian@gatech.edu)

‡College of Computing, Georgia Institute of Technology, Atlanta GA 30332, [gbiros@gmail.com](mailto:gbiros@gmail.com)

§Courant Institute of Mathematical Sciences, New York University, New York 10012, [dzorin@cims.nyu.edu](mailto:dzorin@cims.nyu.edu)

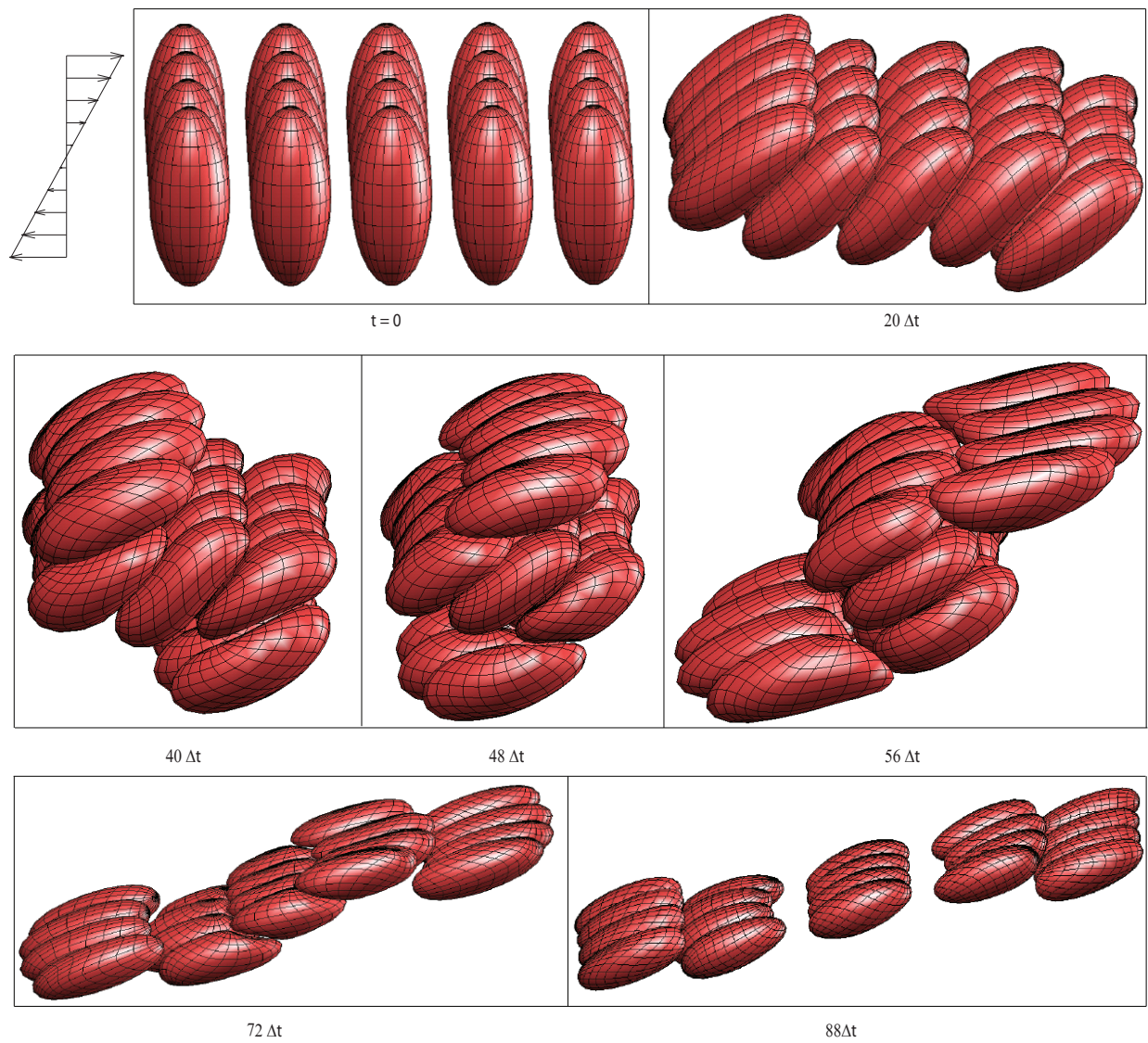


Figure 1: Snapshots of twenty vesicles suspended in a simple shear flow with  $\chi = 18$ . Initially, each vesicle has a non-equilibrium 2-1 ellipsoidal shape and they are arranged in a rectangular lattice. The number of spatial discretization points is 338 ( $p = 12$ ) and the average wall-clock time is 110 seconds per time-step on a 2.33 GHz Xeon workstation with 4GB of RAM.

The vesicle evolution dynamics is characterized by a competition between membrane elastic energy, surface inextensibility, vanishing in-plane shear resistance, and non-local hydrodynamic interactions. Simulation of vesicles is a challenging nonlinear free boundary value problem, not amenable to analytical solutions in all but a few simple cases; numerical simulations and experiments are the only options for the quantitative characterization of vesicle flows.

In this paper, we present an algorithm for the simulation of general three-dimensional vesicle flows, extending our recent work on 2D [62] and 3D axisymmetric vesicles [61]. (To demonstrate the capabilities of our code, we depict a few time-snapshots from a twenty-vesicle simulation in Figure 1.)

To establish notation, we start with the standard PDE formulation of the problem:

$$\begin{aligned}
-\Delta \mathbf{v} + \nabla p &= \mathbf{0} \text{ in } \mathbb{R}^3 \setminus \gamma && \text{(conservation of momentum in bulk fluid),} \\
\operatorname{div} \mathbf{v} &= 0 \text{ in } \mathbb{R}^3 \setminus \gamma && \text{(conservation of mass),} \\
\operatorname{div}_\gamma \mathbf{v} &= 0 \text{ on } \gamma && \text{(surface inextensibility),} \\
\llbracket -p\mathbf{n} + (\nabla \mathbf{v} + \nabla \mathbf{v}^T)\mathbf{n} \rrbracket &= \mathbf{f}_b + \mathbf{f}_\sigma \text{ on } \gamma && \text{(balance of momentum on the membrane),} \\
\mathbf{v} - \mathbf{v}_\infty &\rightarrow 0, \text{ if } \|\mathbf{x}\| \rightarrow \infty && \text{(far field condition),} \\
\frac{\partial \mathbf{x}}{\partial t} &= \mathbf{v} \text{ on } \gamma && \text{(membrane material point motion).}
\end{aligned} \tag{1}$$

In the first equation,  $\mathbf{v}$  is the fluid velocity,  $p$  is the pressure, and  $\gamma$  is the vesicle membrane. In the third equation,  $\operatorname{div}_\gamma$  is the surface divergence operator. In the fourth equation,  $\llbracket q \rrbracket$  denotes the jump of a quantity  $q$  across the vesicle membrane,  $\mathbf{n}$  is the normal to the vesicle membrane,  $\mathbf{f}_b$  is a force due to bending resistance of the vesicle membrane, and  $\mathbf{f}_\sigma$  is a force (tension) that enforces the surface inextensibility constraint. In the momentum balance equation, we assume that the mass of the membrane is negligible. In the last equation,  $\mathbf{x}$  is the position of a material point on the membrane. In addition, the last equation encapsulates the fact that  $\llbracket \mathbf{v} \rrbracket = 0$  due to conservation of mass (normal direction) and the non-slip condition (tangent plane).

A mathematically-equivalent formulation is based on an explicit form of the solution of Stokes equation in the free space:

$$\begin{aligned}
\mathbf{v} &= \mathbf{v}_\infty + \mathcal{S}[\mathbf{f}_b + \mathbf{f}_\sigma] \text{ on } \gamma, \\
\operatorname{div}_\gamma \mathbf{v} &= 0 \text{ on } \gamma, \\
\frac{\partial \mathbf{x}}{\partial t} &= \mathbf{v} \text{ on } \gamma.
\end{aligned} \tag{2}$$

where  $\mathcal{S}$  is the single-layer Stokes operator, defined in Section 3. The first equation encapsulates five equations of the PDE formulation: conservation of momentum in the bulk and on the membrane, conservation of mass, and the far-field boundary conditions for the fluid.

## 1.1 Contributions

We extend the ideas presented in [62] to the general case of vesicles in three dimensions. The main features of the method of [62] are an integral equation formulation, spectral discretization in space, and a semi-implicit time-stepping scheme. In [61], we described an extension of [62] to the axisymmetric case. There are several challenges specific to numerical simulation of non-axisymmetric vesicles flows in 3D: (i) What should the spatial discretization scheme be to maximize

accuracy, computational efficiency, and numerical stability? (ii) Unlike 2D, using a Lagrangian frame of reference for spatial discretization of vesicle boundary results in extreme distortions, leading to numerical instability; how can this instability be controlled? (iii) Bending and tension forces in the 3D case have a more complex nonlinear form; can we design a time-stepping scheme that addresses the severe stiffness while having similar per-time-step complexity to the explicit scheme (similar to our schemes for 2D and axisymmetric flows [61, 62])?

We summarize our contributions below.

- We develop a high-order spatial discretization for inextensible vesicles based on spherical harmonics (Section 4), combining and extending a number of previously proposed techniques (see Section 2), including extending the quadrature scheme of [30] to the Stokes kernel (Section 4.1);
- We propose a particular linearization for a semi-implicit time-stepping scheme along with a preconditioning scheme (Section 5.3);
- We analyze the stiffness of equation 2 (Section 5) and use it to precondition the linear solves in the semi-implicit time scheme;
- We propose a reparametrization scheme for stabilization of time-stepping, and analyze its accuracy and stability (Section 6);
- We verify the numerical scheme by comparing vesicle dynamics obtained with the proposed method against shapes obtained using our axisymmetric solver (which is based on an entirely different discretization scheme) (Section 7);
- We present results on computing the equilibrium shapes of dilute suspension under shear flow, examine a two-vesicle interaction problem, study the sedimentation of a vesicle, and provide an example of a simulation with multiple vesicles (Section 7).

**Synopsis of our method.** Our method is based on Lagrangian tracking of spectral collocation points placed on the membrane of the vesicle combined with a surface reparametrization scheme. We achieve spectral accuracy in space by discretizing using spherical harmonics, which we use for the representation of the membrane and its spatial derivatives, for quadratures, and for antialiasing (Section 4). For weakly-singular integrals, we use the scheme proposed in [30], which enables high-accuracy simulations with a small (compared to low-order schemes) number of points per vesicle. For the position update in time, we use two variants of a semi-implicit marching scheme first derived for advection-diffusion equations [2] and then applied on integral-equation based fluid-structure interaction problems in [58].

The time-marching scheme requires the solution a linear system of equations at each time step, which we perform using a Krylov iterative method (GMRES [54]). The problem of poor conditioning is addressed by a preconditioner based on the analytically obtained spectrum of the operators in (2) for the special case of a unit sphere. Vesicle-vesicle interactions can be carried out using the kernel independent fast multipole method [66]. In this paper, we focus on the mathematical formulation and not on performance and parallelization. We report, preliminary results on a high-performance parallel implementation of this method in [51].



In all, we are able to achieve high accuracy while using a small number of unknowns per vesicle for the spatial discretization and taking large time steps with a relatively low computational cost per time step.

Pseudocode that summarizes the scheme can be found in Section 5.3.

## 1.2 Limitations

We restrict our attention to suspensions of vesicles in fluids with unbounded domains. We have ignored inertial terms, so the overall method is restricted to low Reynolds numbers. We only consider spherical-topology vesicles and we do not allow for topological changes, which are present in many biophysical phenomena involving vesicles.

In our examples, we assume that the interior and exterior of the vesicles are filled with the same liquid. The algorithm extends to the case of viscosity contrast with the introduction of double layer potentials [14]. Additional convolution with a *stresslet* is required which can be computed using the singular quadrature rule described in this paper. The detailed algorithm for two-dimensional flow can be found in [52], in which in addition to viscosity contrast we introduce confined flows with Dirichlet boundary conditions. The formulation for the three dimensional case is similar. We will report the extension in future work.

An important limitation of our scheme is the lack of adaptivity (both in space and time). This lack of adaptivity can cause vesicle-vesicle collisions, which are not possible in the mathematical model we use. Indeed, one can easily construct simulations where without a significant increase of the surface discretization size, our code fails to resolve inter-vesicle interactions accurately. This is an open problem and we are currently working on addressing this issue.

We have also observed a dependence of the stable time step size on the shear rate.

## 1.3 Outline of the paper

In the next section, we review the related work on numerical methods for vesicle flows. In Section 3, we present the overall integral equation formulation. In Sections 4 and 5, we present the spatial and temporal discretization schemes respectively. In Section 6, we discuss the reparametrization scheme and we conclude in Section 7 with results from numerical experiments.

## 2 Related work

Several of the algorithmic components of the proposed method have appeared in the literature. A few closely related papers are our work on 2D and 3D axisymmetric flows ([62], [61]), Graham and Sloan's singular quadratures (for the scalar Helmholtz operator, but are directly applicable to our case) [30], Zinchenko and Davis' surface reparametrization schemes for drops and deformable particles [72], and the very recent work of Zhao et al. [68] on boundary integral equation based simulations of red blood cells in shear flow, presenting a spherical harmonic discretization of membranes with bending and shear resistance and explicit time discretization for particulate flows of this type. We discuss these papers in more detail below. To our knowledge, there is no prior work on implicit or semi-implicit schemes for locally inextensible vesicles in three dimensions.

We focus our discussion of related work on *three-dimensional numerical methods*; we do not attempt to perform a comprehensive review of experimental and numerical studies of vesicle flows.

**Three-dimensional vesicle flows.** Stokesian particulate flow problems are solved using a variety of methods, including unstructured finite element methods and Cartesian grids (immersed boundary/interface, fictitious domain, phase-field and level sets). For a brief review of these methods, in comparison with integral equation formulations, see [9]. Several groups have considered stationary shapes of three-dimensional vesicles using semi-analytic [10, 15, 56], or numerical methods like the phase-field [8, 22, 23] and finite element methods [25, 43]. These approaches are based on a constrained variational approach (i.e., minimizing the bending energy subject to area and volume constraints) and have not been used for resolving fluid-structure interactions.

We restrict our discussion to integral equation formulations, which, for certain classes of problems, offer certain computational advantages compared to stencil-based methodologies (finite differences or finite elements). Pozrikidis [48] reviews the work on boundary integral formulations for particulate flows prior to 2000.

The main features of an integral-equation vesicle flow solver are the underlying formulation (direct vs indirect), the discretization of the surface, the quadrature scheme, and the time-stepping scheme. For Lagrangian vesicle surface discretization, a form of adapting the sampling of the surface to the deformed shape (reparametrization) is needed. To our knowledge the majority of particulate flow solvers in three dimensions use explicit schemes<sup>1</sup> with the exception of the work of Dimitrakopoulos et al. [20]. In that work, a Jacobian-free, finite-difference based Newton method was used. Despite its good convergence, the cost per iteration is somewhat higher than our approach as the method requires multiple evaluations for each matrix-vector multiplication and no preconditioning was used. We discuss the related work for each of the other features in more detail below.

**Integral equation formulation.** Several authors [6, 7, 16, 17, 35, 57] have used direct integral equation formulation for simulating vesicle flows. This formulation results in a single-layer potential for flows with no viscosity contrast and a combination of a single-layer and a double-layer for flows with viscosity contrast [35, 48, 57, 69]. A particular feature of vesicle flows is the local surface inextensibility constraint. In [35, 57] the constraint is treated by a penalty formulation. The penalty parameter results in artificial stiffness and a loss in accuracy in enforcing the constraint. Imposing the constraint using a Lagrange multiplier removes the stiffness and resolves the inextensibility constraint with spectral accuracy [50, 62, 69].

**Surface discretization.** In particulate flows, the surface is typically represented by a triangulation. Once such triangulation is available, either collocation or Galerkin schemes can be used to discretize the integral equation [4]. Examples include [35, 49, 53, 57]. High-order B-spline methods were used in [69]. Pozrikidis [14] proposes a boundary element scheme for deformable capsules with bending and in-plane shear resistance based on six-node quadratically curved triangles and a low-order accurate integration method. This discretization requires tracking of 1026 points per surface to get 0.01% error in the total volume of the particle<sup>2</sup>. Spherical harmonics have been used frequently in boundary integral equations [4]. It enables spectrally accurate integration and

---

<sup>1</sup>In two dimensions, there is a lot of work on semi-implicit schemes, see [62] for more details.

<sup>2</sup>Pozrikidis provides evidence on the significance of including bending even in the case of high-shear flows. He concludes that bending resistance restricts the overall capsule deformations and prevents the development of highly curved shapes, thus, having significant effect even on time scales that are much larger than the bending relaxation time.

differentiation. Zhao et al. [68] uses a spherical harmonic discretization for deformable capsules with bending and shear resistance, but no inextensibility constraint.

**Singular quadratures.** Bruno and Kunyansky [13] proposed a spectral integration scheme for weakly-singular integrals. Ying et al. [67] extended it to arbitrary-geometry smooth surfaces. Although asymptotically optimal, this scheme is rather expensive as it requires the use of partition of unity functions, for which derivative magnitudes rapidly increase with order and as a consequence, a relatively large number of points is needed for good approximation. This scheme is used in [68]. To reduce the discretization size without compromising accuracy, we use the scheme proposed by Graham and Sloan [30]. It is only applicable on smooth surfaces of genus zero but it is quite fast. In [28], the schemes of [13] and [30] are compared and it is shown that the latter scheme is more accurate for small numbers of discretization points.

Simulation of concentrated suspensions of vesicles requires evaluation of nearly-singular integrals. Although we could use the integration scheme in [67], we have found that, for the problem sizes we are targeting, it is cheaper to use a simple upsampling-based quadrature (see discussion in Section 4.1).

**Reparametrization.** In the context of deformable surfaces, the need and methods for maintaining grid quality is ubiquitous. All the resampling methods known to us in this context focus on mesh-based surface representations (primarily piecewise-linear, but also higher-order, e.g., [36]) and we are not aware of any methods designed for spectral discretizations.

Many approaches are based on various types of tangential mesh smoothing, often combined with connectivity adaptation. For simulation of multiple interacting drops, Loewenberg and Hinch [41] proposed a heuristic formula for artificial tangential velocity reducing mesh distortion. Zinchenko et al. [76] observed that the technique of [41] leads to instabilities in simulations with gravity induced motions, and constructed a tangential velocity field by global minimization of the sum of squares of the rates of change of the distances between adjacent vertices (tangency of the field is enforced as a constraint), the first instance of the (*passive stabilization*) approach. This method was further developed by introducing different objectives for minimization in [70, 77], which adapted the sampling density to curvature and controlled triangle quality. More complex versions of passive stabilization methods these methods are quite expensive as relatively complex nonlinear energies are minimized using gradient descent. A simpler version of passive stabilization suitable for moderate deformations was used in [71–73], including a distance-preserving term, and a triangle “compactness” term penalizing elongated triangles. In [75] the same simplified energy is supplemented by node redistribution, aiming to equalize mesh edge lengths. A modification of this method reducing maximal deviation from average is described in [74]. In Cristini et al. [19] a damped relaxation with the energy gradient projected to the surface is used. Additionally, nodes are added by triangle quadrisection (also to minimize the energy) and removed by edge collapse.

An advancing-front method for remeshing of quadratic triangular elements (originally proposed in [39]) adapting triangle size to local criteria such as curvature is described in [36] and is extended in a number subsequent papers to surfaces, in particular in Lohner [42]. Tryggvason et al. [60] briefly describe an algorithm for adapting a mesh to an evolving fluid interface, which uses edge length as a criterion for bisection and edge collapses to remove small elements.

A set of algorithms for anisotropic mesh adaptation, including anisotropic smoothing, applicable to the problem of approximating deforming surfaces, was presented in [32].

Feng and Klug [26] use subdivision finite elements, and demonstrate that local inextensibility combined with a sufficiently accurate Galerkin method makes it possible to simulate moderate deformations with fluid membranes (without fluid interaction). Ma and Klug [43] mention that local inextensibility combined with large deformations still leads to instability, and describe how viscous stabilization of the mesh can be achieved by minimizing an energy measuring the deviation of edge length from previous values. This approach is suitable for computing equilibrium shapes in the absence of external forces, but not for dynamic simulations.

**Aliasing errors.** While various nonlinear quantities (e.g., surface mean curvature and its derivatives used for forces) can be evaluated accurately pointwise, they may contain high frequencies which cannot be resolved properly with a small number of sampling points, leading to *aliasing errors* ([59], [5]), i.e., discretization errors that can be resolved by increasing the resolution. In our previous work [62], in two dimensions 64 points were sufficient to resolve the vesicle interactions and aliasing was not an issue. In three dimensions, however, we need to consider aliasing errors for two reasons: first, we would like to enable accurate simulations with very few spectral collocation points (say 98 points) to allow simulations with very large number of vesicles. In this regime, filtering is required to deal with unresolved frequencies. Second, unlike capsules endowed with strong shear resistance, vesicles are fluid membranes and as a result they experience excessive deformations that lead to significant amplification of aliasing errors. The latter cannot be addressed with simple linear filtering. Nonlinear reparametrization schemes are required. We present one such scheme in this paper tailored for spectral discretizations.

### 3 Formulation

In the rest of the paper we assume that the fluids in the interior and exterior of the vesicle have the *same* dynamic viscosity  $\mu$ .

**Notation.** Before we state the overall formulation of the problem, let us set the notation. We consider a single vesicle first.

- With  $\gamma$  we denote the vesicle membrane (or vesicle boundary).
- For a vector field  $\mathbf{f}(\mathbf{y})$ ,  $\mathbf{y} \in \gamma$ , and an arbitrary point  $\mathbf{x} \in \mathbb{R}^3$  we define the single-layer Stokes operator as

$$\mathcal{S}_\gamma[\mathbf{f}](\mathbf{x}) := \int_\gamma G(\mathbf{x}, \mathbf{y}) \mathbf{f}(\mathbf{y}) d\gamma(\mathbf{y}), \quad G(\mathbf{x}, \mathbf{y}) := \frac{1}{8\pi\mu} \left( \frac{1}{\|\mathbf{r}\|} \mathbf{I} + \frac{\mathbf{r} \otimes \mathbf{r}}{\|\mathbf{r}\|^3} \right), \quad \mathbf{r} = \mathbf{x} - \mathbf{y}, \quad (3)$$

where  $G$  is the free-space Green's function for the Stokes equation,  $\|\cdot\|$  is the Euclidean norm in  $\mathbb{R}^3$ , and  $\mathbf{I}$  is the  $3 \times 3$  identity operator.

- With  $\text{div}_\gamma(\mathbf{x})$ ,  $\nabla_\gamma(\mathbf{x})$ , and  $\Delta_\gamma(\mathbf{x})$  we denote the surface divergence, surface gradient, and surface Laplacian (Laplace-Beltrami operator) respectively evaluated on  $\mathbf{x} \in \gamma$ .
- With  $W(\mathbf{x})$ ,  $\mathbf{n}(\mathbf{x})$ ,  $H(\mathbf{x})$  and  $K(\mathbf{x})$  we denote the area element, the unit normal to  $\gamma$ , the mean curvature, and the Gaussian curvature defined on a point  $\mathbf{x} \in \gamma$ . We give explicit formulas in the Appendix D.

- Finally, let us define the bending  $\mathbf{f}_b$  and tension  $\mathbf{f}_\sigma$  forces on a point  $\mathbf{x} \in \gamma$ :

$$\begin{aligned}\mathbf{f}_b(\mathbf{x}) &= -\kappa_B \left( \Delta_\gamma(\mathbf{x})H(\mathbf{x}) + 2H(\mathbf{x})(H(\mathbf{x})^2 - K(\mathbf{x})) \right) \mathbf{n}(\mathbf{x}), \\ \mathbf{f}_\sigma(\mathbf{x}, \sigma) &= \sigma \Delta_\gamma(\mathbf{x})\mathbf{x} + \nabla_\gamma(\mathbf{x})\sigma,\end{aligned}\tag{4}$$

where  $\kappa_B$  is the surface's bending modulus. These are obtained by taking the  $L^2$ -gradient of the surface energy  $\mathcal{E} = \int_\gamma \frac{1}{2}\kappa_B H^2 + \sigma d\gamma$ . See [65] for the derivation.

In the rest of the paper, we will frequently suppress the dependence of  $\text{div}_\gamma, \nabla_\gamma, \Delta_\gamma, H, K, \mathbf{f}_b, \mathbf{f}_\sigma, \mathbf{n}$  on the evaluation point, which should be clear from the context.

Following [62], the evolution of a point  $\mathbf{x}$  on the vesicle's boundary  $\gamma$  is governed by

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{v}_\infty(\mathbf{x}) + \mathcal{S}[\mathbf{f}_b + \mathbf{f}_\sigma](\mathbf{x}) \\ \text{div}_\gamma(\mathcal{S}[\mathbf{f}_\sigma]) &= -\text{div}_\gamma(\mathbf{v}_\infty + \mathcal{S}[\mathbf{f}_b]),\end{aligned}\tag{5}$$

where  $\mathbf{v}_\infty$  is a specified far-field velocity.

## 4 Spatial Scheme

We use spherical harmonic expansions to represent the surface and the interfacial forces. For force evaluation we need to apply forward and inverse harmonic transforms, as well as differentiate and integrate functions on the surface. In this section, we describe our surface representation, spectral differentiation and quadratures used to integrate functions on the surface, both smooth and singular, with singularity due to multiplication by the Stokes kernel. We conclude with a brief discussion on quadratures for near-singular vesicle-vesicle interactions.

**Spherical harmonics.** Let  $\gamma$  be a smooth surface of spherical topology and let  $\mathbf{x} : U \rightarrow \gamma$  be a parametrization of  $\gamma$ . The domain  $U$  is the rectangle  $\{(u, v) | u \in [0, \pi], v \in [0, 2\pi)\}$ ;  $u$  parametrizes the latitude and  $v$  parametrizes the longitude. A *scalar spherical harmonic* function of degree  $n$  and order  $m$  is given by [12]

$$Y_n^m(u, v) = \frac{1}{\sqrt{2\pi}} \bar{P}_n^m(\cos u) e^{imv}, \quad \text{where} \quad \bar{P}_n^m(t) := \sqrt{\binom{n+\frac{1}{2}}{n-m} \frac{(n-m)!}{(n+m)!}} P_n^m(t).\tag{6}$$

Here,  $|m| \leq n$ ,  $P_n^m$  denote the associated Legendre functions of degree  $n$  and order  $m$ , and  $\bar{P}_n^m$  denote the corresponding normalized associated Legendre functions. Spherical harmonics form an orthonormal basis for square-integrable functions defined on the unit sphere: any scalar function  $\phi \in L^2(\mathbb{S}^2)$  can be expanded as

$$\phi(u, v) = \sum_{n=0}^{\infty} \sum_{m=-n}^n \hat{\phi}_n^m Y_n^m(u, v),\tag{7}$$

$$\text{where} \quad \hat{\phi}_n^m = \int_0^{2\pi} \int_0^\pi \phi(u, v) \overline{Y_n^m(u, v)} \sin u \, du \, dv.\tag{8}$$

For  $\phi \in C^\infty(\mathbb{S}^2)$ , the finite-term approximation

$$\phi(u, v) \approx \sum_{n=0}^p \sum_{m=-n}^n \hat{\phi}_n^m Y_n^m(u, v)\tag{9}$$

is spectrally convergent [47].



**Surface representation.** The vesicle boundary  $\gamma$  is described by a set of spherical harmonic coefficients  $\{\hat{\mathbf{x}}_n^m | n = 0, \dots, p; m = -n, \dots, n\}$  so that for all  $\mathbf{x} \in \gamma$  we have:

$$\mathbf{x}(u, v) = \sum_{n=0}^p \sum_{m=-n}^n \hat{\mathbf{x}}_n^m Y_n^m(u, v), \quad u \in [0, \pi], \quad v \in [0, 2\pi). \quad (10)$$

Each Cartesian component of the position vector  $\mathbf{x}$  is expanded as a real-valued function. In total, we need  $(p+1)^2$  coefficients for each Cartesian component.

**Forward and inverse transforms.** A standard choice for spectrally accurate integration of a function represented by an order  $p$  spherical harmonic expansion is to use the trapezoidal rule for  $v$  and Gaussian quadrature rule for  $u$ : we choose  $2p+2$  equispaced nodes in the  $v$ -direction  $\{v_k = \frac{\pi k}{p}\}_{k=0}^{2p+1}$  and  $p+1$  nodes along the  $u$ -direction  $\{u_j\}_{j=0}^p$  given by

$$u_j = \cos^{-1}(t_j), \quad \text{where } t_j \in [-1, 1]. \quad (11)$$

Here  $t_j$ 's are  $(p+1)$ -point Gauss-Legendre quadrature nodes. On this grid a standard convergence estimate holds:

**Theorem 4.1.** (Quadrature rule for regular integrals) *For any smooth function  $f$  defined on a  $C^\infty$  surface  $\gamma$  of spherical topology, the quadrature rule*

$$\int_{\gamma} f d\gamma = \sum_{j=0}^p \sum_{k=0}^{2p+1} w_{jk} f(u_j, v_k) W(u_j, v_k), \quad \text{where } w_{jk} = \frac{\pi}{p} \frac{\lambda_j}{\sin u_j} \quad (12)$$

*is superalgebraically convergent with  $p$ . Here,  $\lambda_j$  are the Gaussian quadrature weights and  $W(u_j, v_k)$  is the infinitesimal area element of  $\gamma$ .*

One can use fast transforms for both longitude (Fast Fourier Transform) and latitude (Fast Legendre Transform [FLT]) so that the spherical harmonic coefficients can be computed in  $\mathcal{O}(p^2(\log p)^2 + p^2 \log p)$  (see [44])<sup>3</sup>. The pseudocode for the forward spherical harmonic transform is given in Appendix C. A similar algorithm can be used for the inverse transform.

**Computing derivatives.** We compute derivatives of functions defined by samples on  $\gamma$  by applying the forward spherical harmonic transform and using standard differentiation formulas in the spectral domain:

$$\mathbf{x}_u = \sum_{n=0}^p \sum_{m=-n}^n \hat{\mathbf{x}}_n^m (Y_n^m)_u, \quad \mathbf{x}_v = \sum_{n=0}^p \sum_{m=-n}^n \hat{\mathbf{x}}_n^m (Y_n^m)_v \quad (13)$$

where the derivatives of spherical harmonics are:

<sup>3</sup>In our implementation, we use direct transforms since  $p$  is relatively small. As we will see, the main cost of the computation is resolving the weakly singular Stokes single-layer potential.

$$\frac{\partial^k Y_n^m}{\partial v^k} = (im)^k Y_n^m. \quad (14)$$

$$(Y_n^m)_u = \sqrt{(n-m)(n+m+1)} e^{-iv} Y_n^{m+1} + m \cot u Y_n^m, \quad (15)$$

$$(Y_n^m)_{uu} = (n+m+1)\sqrt{(n-m)(2+m+n)} e^{-2iv} Y_n^{m+2} + (2m+1)\sqrt{(n-m)(n+m+1)} e^{-iv} \cot u Y_n^{m+1} - m^2 Y_n^m. \quad (16)$$

For smooth  $\mathbf{x}$  this numerical differentiation scheme is spectrally accurate because the only approximation made is the  $p_{th}$ -order spherical harmonic transform.

Computing differential quantities accurately requires *filtering* for antialiasing. For instance, the mean curvature,  $H(\mathbf{x})$ , of a smooth surface is a smooth function which can be computed *at sample locations* from the derivatives of  $\mathbf{x}$  (see Appendix D). However, even if  $\mathbf{x}$  is bandlimited and exactly representable with the expansion of order  $p$ , this could no longer be true for the mean curvature, and its reconstruction from the values at sample points used for  $\mathbf{x}$  suffers from aliasing. To reduce these errors, we upsample  $\mathbf{x}$  using spherical harmonics interpolation, perform differentiation, filter out the high-frequency components and then restrict the result to the original grid. The required upsampling order depends on the surface and also the differentiation order and  $p$  (for high-enough  $p$ , both  $\mathbf{x}$  and  $H$  are resolved and no filtering is necessary). In our numerical experiments, we found that upsampling to a resolution two times higher than the original grid works well. We demonstrate one such experiment in Figure 2.

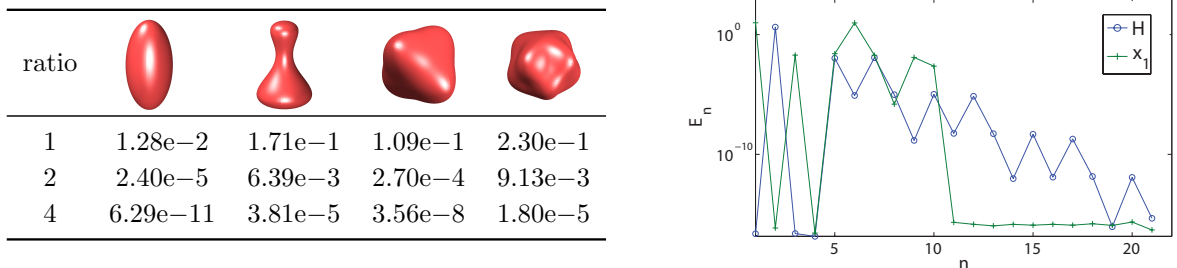


Figure 2: In the left table, we report the relative errors  $\|H_{q/p} - H^*\|_\infty / \|H^*\|_\infty$  in the computation of the mean curvature. By  $H_{q/p}$ , we denote the mean curvature obtained by upsampling a  $p$ th order representation of the  $\mathbf{x}$  to  $q$ th order, computing the curvature and then restricting it to the original grid. The reference function  $H^*$  is computed analytically. Each row corresponds to a particular ratio  $q/p$  and for all shapes we have set  $p = 6$ . In the right figure, we plot the spectrum  $E_n[f] = \sum_{m=-n}^n |\hat{f}_n^m|$  of the functions  $x_1(u, v)$  and  $H(u, v)$  corresponding to the rightmost shape in the table. The bandwidth of the curvature, as expected, is higher than that of the positions justifying the need for upsampling.

## 4.1 Singular Integration

For integrals involving the weakly singular Stokes kernel, we use a superalgebraically convergent integration scheme. There are many techniques that can be used to resolve the singularity accurately. As discussed in the introduction, our main goal in choosing the quadrature scheme is to achieve spectral accuracy while maintaining good computational efficiency for small values of  $p$  ( $p = 8 - 32$ ). For smooth surfaces that admit global parametrization, there are two main approaches to singular

integration: one proposed by Bruno and Kunyansky [13] based on partitions of unity and the other proposed by Graham and Sloan [30] based on rotations and special quadratures. The former scheme has better asymptotic complexity  $\mathcal{O}(p^3)$  as opposed to the  $\mathcal{O}(p^5)$  of the latter. However, for the values of  $p$  that are of interest to us ( $p \leq 48$ ), the Graham-Sloan scheme performs much better [28]: multiplication by the partition of unity used in the scheme of [13] results in substantial increase in the integrand derivative magnitudes resulting in lower accuracy for a given number of samples. Here, we summarize the Graham-Sloan scheme for convolutions with the Laplace kernel  $1/\|\mathbf{r}\|$  and then discuss its extension to the Stokes kernel.

Consider a smooth scalar function  $f(\mathbf{x}(u, v)) : \mathbb{S}^2 \rightarrow \mathbb{R}$ . We compute the integral

$$\mathcal{A}[f](u_t, v_t) := \int_0^{2\pi} \int_0^\pi \frac{f(u, v)W(u, v)}{\|\mathbf{r}\|} du dv \quad \text{where} \quad \mathbf{r} = \mathbf{x}(u_t, v_t) - \mathbf{x}(u, v) \quad (17)$$

and  $W(u, v)$  is the surface area term. First, we rotate the coordinate system  $(u, v) \rightarrow (u_R, v_R)$  so that fixed evaluation point  $(u_t, v_t)$  becomes  $(0, 0)$  (the north pole). The positions are modified to  $\mathbf{x}(u_R, v_R)$ , the density to  $f(u_R, v_R)$  and the surface area term to  $W(u_R, v_R)$ . In subsequent expressions, we drop the the subscripts from  $(u_R, v_R)$  and the coordinate system should be clear from the context. Applying the smooth quadrature rule (12), however, may still not be efficient. As an illustration, consider the unit sphere. In this case,  $\mathbf{x}_{\mathbb{S}^2}(u, v) = (\sin u \cos v, \sin u \sin v, \cos u)$ ,  $W = \sin u$  and

$$\mathcal{A}_{\mathbb{S}^2}[1](0, 0) = \int_0^{2\pi} \int_0^\pi \frac{\sin u}{\sqrt{2 - 2 \cos u}} du dv = \int_0^{2\pi} \int_0^\pi \cos \frac{u}{2} du dv = \int_0^{2\pi} dv \int_{-1}^1 dt \frac{1}{\sqrt{1 - t^2}}, \quad (18)$$

by change of variables  $t = \cos u$ . The integral has a square-root singularity at  $t = \pm 1$ , and hence a Gaussian quadrature rule in the  $t$ -domain (as applied in (12)), is not efficient.

The Graham-Sloan scheme makes use of three properties: (1) since  $\gamma$  is diffeomorphic to a sphere, it reduces the problem of computing the harmonic potential on  $\gamma$  to a problem of computing the harmonic potential on  $\mathbb{S}^2$ . Defining  $s(u, v) = (W/\|\mathbf{r}\|)_{\mathbb{S}^2} = \cos(u/2)$ , we can write s (17) as

$$\mathcal{A}[f](0, 0) := \int_0^{2\pi} \int_0^\pi s(u, v) \left( \frac{f(u, v)W(u, v)}{s(u, v)\|\mathbf{r}\|} \right) du dv;$$

(2) Although the function  $\frac{fW}{s\|\mathbf{r}\|}$  can be discontinuous at the north pole, it has sufficiently well-behaved spherical harmonic expansions; (3) The harmonic potential on  $\mathbb{S}^2$  can be computed analytically for spherical harmonics [45]:

$$\int_0^{2\pi} \int_0^\pi s(u, v) Y_n^m(u, v) du dv = \frac{4\pi}{2n + 1} Y_n^m(0, 0). \quad (19)$$

Based on these properties, we can construct modified quadrature weights to compute the harmonic potentials. The result can be summarized as follows:

**Theorem 4.2.** (Quadrature rule for singular integrals at poles) [30] *For any smooth boundary  $\mathbf{x}$  and for any smooth function  $f$ , the quadrature rule for computing the harmonic potential at the*

north-pole given by

$$\mathcal{A}[f](0, 0) = \sum_{j=0}^p \sum_{k=0}^{2p+1} \frac{w_{jk}^s}{\|\mathbf{x}(0, 0) - \mathbf{x}(u_j, v_k)\|} f(u_j, v_k) W(u_j, v_k), \quad \text{with} \quad (20)$$

$$w_{jk}^s = w_{jk} \sum_{n=0}^p \frac{4\pi}{\sqrt{2(2n+1)}} \frac{P_n(\cos u_j)}{\cos(u_j/2)}$$

is superalgebraically convergent<sup>4</sup> with  $p$ .

**Remark 1.** As has been observed by several authors [3, 13, 30], the function  $\frac{fW}{\|\mathbf{r}\|}$  in (17) is smooth in the rectangular domain  $U$  because the vanishing surface area term  $W$  counteracts the singularity due to  $1/\|\mathbf{r}\|$ . Therefore, applying the Gaussian quadrature along  $u$ -direction (as opposed to Theorem 4.1 which applies Gaussian quadrature in  $t$ -direction where  $t = \cos u$ ) and the trapezoidal rule along  $v$ -direction yields spectral convergence. However, the errors for a specific  $p$  would be slightly higher using this scheme compared to the scheme described in Theorem 4.2. The reason for this is the clustering of the sampling points near the poles, which results in less accurate approximations for a fixed  $p$ .

A significant part of the overall complexity of (20) is due to the rotation of the coordinate system needed for every evaluation point on the vesicle surface. Since we use  $p^2$  evaluation points on the surface, the overall cost of the rotation for just one evaluation point is  $\mathcal{O}(p^4)$  (mapping  $p^2$  points to  $p^2$  points). (The rotations must be applied for  $f$  and the three coordinates of the points.)

However, the  $\mathcal{O}(p^4)$  cost per point can be reduced to  $\mathcal{O}(p^3)$  per point by rotating in the spherical harmonic representation. Invoking the addition theorem [12], one can show that the harmonic coefficients of degree  $n$  in the rotated coordinate system depend only on the coefficients of degree  $n$  in the original coordinate system.

More precisely, let  $\mathbf{F} := \{f(u_j, v_k) | j = 0, \dots, p; k = 0, \dots, 2p + 1\}$ . The rotations can be treated sequentially, i.e., we can first rotate the grid in the  $v$ -direction in the spherical harmonics space (which corresponds to a simple shift requiring  $\mathcal{O}(1)$  operations) and then use a precomputed rotation in the  $u$ -direction. Given a target point  $(u, 0)$ ,  $(0, 0)$  in the rotated system), and for each degree  $n$  we use a  $n \times n$  matrix  $R_n(u, 0)$  such that

$$\hat{\mathbf{F}}'_n(u, 0) = R_n(u, 0) \hat{\mathbf{F}}_n.$$

We represent the combined rotations in both  $u$  and  $v$  more compactly as

$$\hat{\mathbf{F}}'(u, v) = R(u, v) \hat{\mathbf{F}}.$$

Since  $\hat{\mathbf{F}}'_n$  can be computed with  $\mathcal{O}(p^2)$  work,  $\hat{\mathbf{F}}'$  can be computed in  $\mathcal{O}(p^3)$  work. The matrices  $R_n$  are precomputed either using recurrences or directly by the recently proposed algorithm of Gimbutas and Greengard [29]. Due to symmetry, we only need a  $p/2$  set of rotation matrices.

Let  $P$  be the forward spherical harmonics transform and  $P^*$  the inverse spherical harmonics transform. Algorithm 1 summarizes the computation of an integral with harmonic potential.

<sup>4</sup>The proofs outlined in [30] require that the functions need to be oversampled for spectral convergence. However, here we have assumed no oversampling. Empirically we have found that spectral convergence can be observed even without oversampling (eg., Table 3).

---

**Algorithm 1** Computing the harmonic potential.

Input:  $\mathbf{X} := \{\mathbf{x}(u_j, v_k) \mid j = 0, \dots, p, k = 0, \dots, 2p + 1\}$ ,  $\mathbf{F} := \{f(u_j, v_k) \mid j = 0, \dots, p, k = 0, \dots, 2p + 1\}$

Output:  $\{A[f](u_j, v_k) \mid j = 0, \dots, p, k = 0, \dots, 2p + 1\}$

---

```

 $\hat{\mathbf{X}} = P\mathbf{X}$  and  $\hat{\mathbf{F}} = P\mathbf{F}$  //  $\mathcal{O}(p^2 \log^2 p)$ 
for  $k = 0$  to  $2p + 1$  do
  for  $j = 0$  to  $p$  do
     $\hat{\mathbf{X}}' = R(u_j, v_k)\hat{\mathbf{X}}$  //  $R$  is applied to each component of every point in  $\mathbf{X}$ 
     $\hat{\mathbf{F}}' = R(u_j, v_k)\hat{\mathbf{F}}$  //  $\mathcal{O}(p^3)$ 
    EVALUATE THE HARMONIC POTENTIAL USING (20)
     $\mathbf{Y}(u_j, v_k) = P^{-1}\hat{\mathbf{X}}'$  and  $\mathbf{Q}(u_j, v_k) = P^{-1}\hat{\mathbf{F}}'$  //  $\mathbf{Y}, \mathbf{Q}$ , used to simplify notation
     $A[f](u_j, v_k) = \sum_{n=0}^p \sum_{m=0}^{2p+1} \frac{w_{nm}^s}{\|\mathbf{Y}(u_n, v_m) - \mathbf{X}(u_j, v_k)\|} \mathbf{Q}(u_n, v_m) W'(u_n, v_m)$  //  $\mathcal{O}(p^2)$ 
  end for
end for

```

---

**Extension to Stokes kernel.** Let us assume the evaluation point to be the north pole. The Stokes kernel (3) has an additional factor

$$K(u, v) = \mathbf{I} + \frac{\mathbf{r} \otimes \mathbf{r}}{\|\mathbf{r}\|^2}, \quad \mathbf{r} = \mathbf{x}(0, 0) - \mathbf{x}(u, v)$$

multiplying the Laplace kernel  $1/\|\mathbf{r}\|$ . At the north-pole,  $K$  is continuous but non-smooth. Hence, it is not clear if we can use (20). It turns out that the following two properties of the Stokes kernel suffice to extend the theoretical proofs of [30].

- $K(u, v)$  is *reflectionally symmetric*, that is,  $K(u, v) = K(-u, v + \pi)$ .
- $K(u, v)$  is an infinitely differentiable function and all its derivatives are  $2\pi$ -periodic in both  $u$  and  $v$ .

For more formal descriptions, see Definitions 4.3 and 4.5 of [30]. We use the formula

$$\mathcal{S}[f](0, 0) = \sum_{j=0}^p \sum_{k=0}^{2p+1} w_{jk}^s K(u_j, v_k) f(u_j, v_k) W(u_j, v_k) / \|\mathbf{x}(0, 0) - \mathbf{x}(u_j, v_k)\| \quad (21)$$

analogous to (20) for evaluating the single layer potential with spectral accuracy. At each evaluation point, we rotate the coordinate system so that it is mapped to the north-pole and then use (21). The overall complexity of the algorithm (for a single vesicle) is  $\mathcal{O}(p^5)$ -work and  $\mathcal{O}(p^3)$ -storage.

**Nearly singular integrals.** Vesicle-vesicle hydrodynamic interactions are not singular, so formally the smooth integration rule (12) can be used. However, for close interactions the accuracy of these rules deteriorates as the integrals become nearly singular. A robust, fourth-order accurate, near-singular integration scheme for smooth surfaces is presented in [67]. However, the constant in the complexity estimate of that scheme is rather high since it involves multiple evaluations and interpolations, and requires finding nearest points on the surface.



For this paper—motivated by our empirical observation that vesicles in the parameter ranges that we have studied tend to be well-separated<sup>5</sup> even in relatively concentrated suspensions, we have opted for a simpler scheme that uses upsampling and anti-aliasing. We spectrally upsample<sup>6</sup> all vesicle surfaces by a factor of two (similar to the medium-distance part of the evaluation in [67]), then we compute the interactions at this higher resolution and then downsample to the original resolution using spectral cutoff. This computation can be done in a per-vesicle adaptive way, but in our implementation we have used uniform upsampling across all vesicles. Let us emphasize that in our simulations, oversampling and filtering are necessary: if they are not invoked one observes numerical blow-up (Figure 8). For more concentrated suspensions, the more expensive scheme of [67] is likely to be required for robustness.

## 5 Time Scheme

In this section, we discuss an explicit and a new semi-implicit scheme for the vesicle shape evolution. With very few exceptions (e.g., [20]), explicit schemes have been the method of choice for three-dimensional particle simulations.

Explicit schemes have low computational cost per time step but are known to suffer from stability restrictions on size of the time step. This restriction arises due the numerical stiffness from the high-order spatial derivatives in the bending force. Consequently, the overall computational cost of the simulating in a fixed time horizon tends to be high, especially with increasing spatial resolution. For vesicles, the inextensibility constraint is typically enforced with an explicit penalty parameter [8, 23, 35] that introduces additional stiffness. In our explicit scheme, the inextensibility constraint is enforced using the tension as a Lagrange multiplier.

Semi-implicit schemes offer a trade-off between the per-time step computational cost and stability requirement on the time-step size. However, they are more difficult to analyze, especially for nonlinear systems. For further discussion on implicit schemes for vesicle simulations, see [62]. Another option is to use a fully implicit scheme. In our previous work on 3D axisymmetric vesicle flows [61], we observed experimentally that a line-search, single-level, Newton scheme in which the Jacobian is approximated using the semi-implicit scheme linearization scheme does not result in computational savings.

An important part of both of our explicit and semi-implicit schemes is the need for reparameterization of the surface to minimize aliasing errors. At every time step, an auxiliary evolution equation

$$\dot{\mathbf{x}} = F(\mathbf{x}) \tag{22}$$

is solved. The choice of the smoothing function  $F$  and the time-stepping scheme for (22) is discussed in Section 6.

Before we describe the time-stepping schemes, we state three spectral properties of the Stokes, bending, and surface inextensibility operators. We can use these results to (1) characterize the

---

<sup>5</sup>We consider vesicles well-separated if the minimum inter-vesicle distance is greater than the distance between sample points on the sphere. For example, two spheres in shear flow can come arbitrarily close whereas for droplets the distance tends to stay relatively large [40]. We observe a similar behavior for the case of vesicles; for example, see Figure 9.

<sup>6</sup>Upsampling (or spherical harmonic interpolation) refers to mapping the function to a finer grid by computing the spherical harmonic coefficients (using the given function values at the coarser grid) and then evaluating the spherical harmonic expansion (9) at the finer grid points. Similarly, downsampling (or spherical harmonic filtering) refers to mapping the function to a coarser grid.

stiffness of the problem, (2) construct preconditioners for the constraint and evolution equations (§5.4), and (3) derive analytical solutions for various external flows, see Appendix B.

## 5.1 Spectral analysis on the unit sphere

To characterize the stiffness of the equation (5), we consider the case of a spherical vesicle ( $\gamma \equiv \mathbb{S}^2$ ) and we derive the spectrum of  $\mathcal{S}$ ,  $\text{div}_\gamma \mathcal{S}[\mathbf{f}_\sigma]$ , and a simplified form of  $\mathcal{S}[\mathbf{f}_b]$ . To analyze the spectral properties, it is convenient to use *vector spherical harmonics* that form an orthogonal basis for all vectorial functions  $\mathbf{f} \in L^2(\mathbb{S}^2)$ . They are defined in terms of the scalar spherical harmonics and their derivatives by

$$\mathbf{V}_n^m(u, v) := \nabla_\gamma Y_n^m(u, v) - (n+1)Y_n^m(u, v)\mathbf{n}(u, v) \quad (23a)$$

$$\mathbf{W}_n^m(u, v) := \nabla_\gamma Y_n^m(u, v) + nY_n^m(u, v)\mathbf{n}(u, v) \quad (23b)$$

$$\mathbf{X}_n^m(u, v) := \mathbf{n}(u, v) \times \nabla_\gamma Y_n^m(u, v) \quad (23c)$$

where  $\mathbf{n}$  is the normal to the unit sphere.<sup>7</sup>

For notational convenience, we suppress the  $(u, v)$  dependence. Using these definitions, we now derive the spectra of various operators.

**Theorem 5.1.** (Stokes operator) *On the unit sphere, the vector spherical harmonic functions  $\mathbf{V}_n^m$  and  $\mathbf{W}_n^m$  are the eigenfunctions of the Stokes single-layer operator  $\mathcal{S}$  and*

$$\mathcal{S}[\mathbf{V}_n^m] = \frac{n}{(2n+1)(2n+3)}\mathbf{V}_n^m, \quad (24a)$$

$$\mathcal{S}[\mathbf{W}_n^m] = \frac{n+1}{(2n-1)(2n+1)}\mathbf{W}_n^m. \quad (24b)$$

**Theorem 5.2.** (Inextensibility operator) *On the unit sphere, the spherical harmonic functions are eigenfunctions of the inextensibility operator  $\mathcal{L}$  defined as*

$$\mathcal{L}\sigma = \text{div}_\gamma \mathcal{S}[\sigma \Delta_\gamma \mathbf{x} + \nabla_\gamma \sigma]. \quad (25)$$

and

$$\mathcal{L}Y_n^m = -\frac{n(n+1)(2n^2+2n-1)}{(2n-1)(2n+1)(2n+3)}Y_n^m. \quad (26)$$

**Theorem 5.3.** (Bending operator) *On the unit sphere, the vector spherical harmonic functions  $\mathbf{V}_n^m$  and  $\mathbf{W}_n^m$  are eigenfunctions of the bending operator defined by  $\mathcal{B}[\mathbf{f}] = \mathcal{S}[\Delta_\gamma^2 \mathbf{f}]$  and*

$$\mathcal{B}[\mathbf{V}_n^m] = \frac{n^3(n-1)^2}{(2n+1)(2n+3)}\mathbf{V}_n^m, \quad \mathcal{B}[\mathbf{W}_n^m] = \frac{n(n+1)^2(n+2)^2}{(2n+1)(2n+3)}\mathbf{W}_n^m. \quad (27)$$

<sup>7</sup>Various other sets of vector spherical harmonics are used in the literature, tailored for different kinds of problems (e.g., [45, 64]), specifically, for Stokesian flows in [18, 63]. The ones we defined here are first proposed in Hill [31]. They are traces on  $\mathbb{S}^2$  of a corresponding basis set in  $\mathbb{R}^3$ , usually termed as *solid vector spherical harmonics*, defined as [24]

$$\begin{aligned} \mathcal{V}_n^m(r, u, v) &= r\nabla Y_n^m(u, v) - (n+1)Y_n^m(u, v)\mathbf{e}_r, \\ \mathcal{W}_n^m(r, u, v) &= r\nabla Y_n^m(u, v) + nY_n^m(u, v)\mathbf{e}_r, \\ \mathcal{X}_n^m(r, u, v) &= \mathbf{e}_r \times (r\nabla Y_n^m(u, v)), \end{aligned}$$

where  $(r, u, v)$  are the coordinates of a point in the spherical coordinate system with unit vectors  $(\mathbf{e}_r, \mathbf{e}_u, \mathbf{e}_v)$ .

Note that we have used a simplified expression ( $\mathbf{f}_b = \Delta_\gamma^2 \mathbf{x}$ ) that retains only the dominant term in the bending force. The proofs follow directly from the *antenna theorems* of [55], we outline them in Appendix A.

From Theorems 5.1, 5.2 and 5.3, we conclude that (i)  $\mathcal{S}$  is a smoothing operator, (ii)  $\mathcal{L}$  and  $\mathcal{B}$  are ill-conditioned operators, and (iii) the condition number of  $\mathcal{L}$  grows as  $\mathcal{O}(n)$  and the condition number of  $\mathcal{B}$  grows as  $\mathcal{O}(n^3)$ . Therefore, we expect severe time-step restrictions for explicit schemes. Next, we discuss such a scheme for the vesicle evolution equations.

## 5.2 Explicit Scheme

We begin our discussion by considering the time-stepping scheme for a single vesicle. Let  $\Delta t$  be a fixed time-step size. Given the positions of the discrete points on the vesicle surface  $\{\mathbf{x}^n(u_j, v_k) : 0 \leq j \leq p, 0 \leq k \leq 2p - 1\}$  at time  $n\Delta t$ , the goal is to solve (5) for the new positions at  $(n+1)\Delta t$ . A first-order, explicit time-stepping scheme for (5) reads

$$\mathcal{L}(\mathbf{x}^n)\sigma^{n+1} = -\text{div}_\gamma(\mathcal{S}[\mathbf{f}_b^n](\mathbf{x}^n)) \quad (28a)$$

$$\frac{1}{\Delta t}(\mathbf{x}^{n+1} - \mathbf{x}^n) = \mathcal{S}[\mathbf{f}_b^n + \mathbf{f}_\sigma^{n+1}](\mathbf{x}^n), \quad (28b)$$

where all the differentiation operators are evaluated at  $\mathbf{x}^n$  and  $\mathbf{f}_\sigma^{n+1} := \sigma^{n+1}\Delta_\gamma \mathbf{x}^n + \nabla_\gamma \sigma^{n+1}$ . In the first step, we solve a linear system defined by the discretization of (28a) for the tension  $\sigma^{n+1}$ , given the positions at  $n\Delta t$ . In the second step (28b),  $\sigma^{n+1}$  is used for updating the positions. To compute the bending force  $\mathbf{f}_b^n$ , we compute the curvatures  $H^n$  and  $K^n$  of the surface  $\mathbf{x}^n$  and substitute in (4). The explicit treatment of the single layer potential means that we evaluate it by the following formula

$$\mathcal{S}[\mathbf{f}_b^n](\mathbf{x}^n) = \int_0^{2\pi} \int_0^\pi G(\mathbf{x}^n(u, v), \mathbf{x}^n(u', v')) \mathbf{f}_b^n(u', v') du' dv', \quad (29)$$

which is computed using the quadrature rule of Theorem 4.2. For the spatial discretization of (28a) and (28b) we use a collocation method (Fourier-Legendre quadrature points). Finally, we use a Krylov iterative solver (GMRES [54]) combined with a preconditioner to solve the constraint equation.

This explicit scheme requires the solution of one linear system at each time step and resolves the numerical instability due to tension. However, the bending term is treated explicitly. Based on the spectral analysis of §5.1, we conjecture that  $\Delta t \sim \mathcal{O}(p^{-3})$ , which is also corroborated by our numerical tests.

The extension of the scheme (28) to multiple vesicle suspensions is done by treating the vesicle-vesicle interactions explicitly. Given the individual vesicle positions  $\{\mathbf{x}_k^n\}_{k=1}^K$  at time  $n\Delta t$ , the update of positions and tensions to  $(n+1)\Delta t$  is done in three steps:

STEP 1: EVALUATE BENDING FORCES

```
for  $k = 1$  to  $K$  do
  Compute  $\mathbf{f}_b^{k,n}$ ,  $\mathbf{f}_\sigma^{k,n}$ 
end for
```

STEP 2: COMPUTE NEW TENSIONS

Compute  $\{\sigma_j^{n+1}\}_{j=1}^K$  by solving

$$\mathcal{L}(\mathbf{x}_k^n)\sigma_k^{n+1} = -\operatorname{div}_\gamma \left( \sum_{\substack{j=1 \\ j \neq k}}^K \mathcal{S}[\mathbf{f}_\sigma^{j,n}](\mathbf{x}_k) \right) - \operatorname{div}_\gamma \left( \mathbf{v}_\infty + \sum_{j=1}^K \mathcal{S}[\mathbf{f}_b^{j,n}](\mathbf{x}_k) \right), \quad k = 1, \dots, K$$

STEP 3: EXPLICITLY UPDATE POSITIONS

for  $k = 1$  to  $K$  do

$$\mathbf{x}_{n+1}^k = \mathbf{x}_n^k + \Delta t \sum_{j=1}^K \mathcal{S}[\mathbf{f}_b^{j,n} + \mathbf{f}_\sigma^{j,n+1}](\mathbf{x}_n^k)$$

SURFACE REPARAMETERIZATION AND TENSION ADVECTION

Solve:

$$\begin{aligned} \hat{\mathbf{x}}_k^{n+1} &= F(\mathbf{x}_k^{n+1}) && // \text{apply tangential smoothing} \\ \hat{\sigma}_k^{n+1} + \operatorname{div}_\gamma(\sigma_k^{n+1} F(\mathbf{x}_k^{n+1})) &= 0 && // \text{advect the tension} \end{aligned}$$

end for

Notice that after the surface reparameterization we need to advect the tension to the new positions so that it is available for the next time step. The advection velocity is defined by the reparameterization of the surface.

Next, we propose a new semi-implicit scheme for which we have observed experimentally that the time step for stability is nearly independent of the spatial discretization size  $p$ .

### 5.3 Semi-implicit Scheme

In a semi-implicit scheme, the linear parts of the stiffest terms are treated implicitly [1]. For example if we have a linear dynamical system of the form  $\dot{x} = Ax + Bx$ , in which  $A$  is the stiff operator, we can discretize it using an implicit scheme for  $A$  and an explicit scheme for  $B$ . Such schemes have been analyzed in [2]. Our system however, is more complex: it is of the form  $\dot{\mathbf{x}} = \mathbf{Q}(\mathbf{x})\mathbf{x}$ , where  $\mathbf{Q}$  is a nonlinear operator; to our knowledge, there is no analysis of semi-implicit schemes for such dynamical systems. Our semi-implicit approach is, roughly speaking, to discretize it by  $\mathbf{x}^{n+1} - \mathbf{x}^n = \Delta t \mathbf{Q}(\mathbf{x}^n)\mathbf{x}^{n+1}$ .

Because of its constituent fourth-order spatial derivatives, the bending force is the leading order term that induces stiffness into the evolution equation. Therefore, we look for a linearization of the bending force that results in a stable scheme:

$$\mathcal{L}(\mathbf{x}^n)\sigma^{n+1} = -\operatorname{div}_\gamma \mathcal{S}[\mathbf{f}_b^n](\mathbf{x}^n), \quad (30a)$$

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

where all the differentiation operators are evaluated at  $\mathbf{x}^n$  and  $\mathbf{f}_\sigma^{n+1} := \sigma^{n+1} \Delta_\gamma \mathbf{x}^n + \nabla_\gamma \sigma^{n+1}$ . Suppressing the superscripts on explicitly treated terms for notational convenience, the bending force is defined as

$$\mathbf{f}_b^{n+1} = -(\Delta_\gamma H^{n+1} + 2H^{n+1}(H^2 - K))\mathbf{n}, \quad (31)$$

$$\text{and } H^{n+1} = \frac{1}{2W^2} (E\mathbf{x}_{vv}^{n+1} - 2F\mathbf{x}_{uv}^{n+1} + G\mathbf{x}_{uu}^{n+1}) \cdot \mathbf{n}. \quad (32)$$

In the time-stepping scheme (30), we first solve (30a) for the tension force starting from the positions at time  $n\Delta t$  and then we solve (30b) for the bending force using the linearization defined in (31)<sup>8</sup>.

The advantage of this scheme over the explicit scheme discussed in §5.2 is that the time-step restriction is overcome with only a modest increase in computational cost. Fully implicit schemes, on the other hand, require solution of nonlinear equations at every time-step making them potentially more expensive. We update  $\{\mathbf{x}_n^k, \sigma_n^k\}_{k=1}^K$  as follows:

INTERFACIAL FORCES

```
for  $k = 1$  to  $K$  do
  Compute  $\mathbf{f}_b^{k,n}$  and  $\mathbf{f}_\sigma^{k,n}$ 
end for
```

INTERACTION FORCES

```
for  $k = 1$  to  $K$  do
   $F_k^I = 0$ 
  for  $j \in \{(1, \dots, K) \setminus k\}$  do
     $F_k^I = F_k^I + \mathcal{S}[\mathbf{f}_b^{j,n} + \mathbf{f}_\sigma^{j,n}](\mathbf{x}_k)$ 
  end for
end for
```

UPDATE POSITIONS AND TENSIONS

```
for  $k = 1$  to  $K$  do
  Solve :
     $\mathcal{L}(\mathbf{x}_k^n)\sigma_k^{n+1} = -\text{div}_\gamma(\mathcal{S}[\mathbf{f}_b^{k,n}] + F_k^I)$  tension
     $\mathbf{x}_k^{n+1} - \Delta t \mathcal{S}[\mathbf{f}_b^{k,n+1}] = \mathbf{x}_k^n + \Delta t (\mathcal{S}[\mathbf{f}_\sigma^{k,n}] + F_k^I)$  positions
end for
```

SURFACE REPARAMETERIZATION AND TENSION ADVECTION

```
Solve:
   $\hat{\mathbf{x}}^{n+1}_k = F(\mathbf{x}_k^{n+1})$  apply tangential smoothing
   $\hat{\sigma}_k^{n+1} + \text{div}_\gamma(\sigma_k^{n+1} F(\mathbf{x}_k^{n+1})) = 0$  advect the tension
end for
```

High-order variants of the explicit (28) and the semi-implicit (30) schemes are readily obtained using the backward difference formula [1].

Both the explicit and the semi-implicit schemes suffer from mesh distortion in longer time simulations. In 2D, the local inextensibility constraint prevents the mesh distortion to a large extent. Unfortunately, this is not the case in 3D. In §6, we propose a new reparameterization scheme that preserves the quality of the mesh in a dynamic simulation. But first, let us discuss how the spectral analysis of §5.1 can be used for preconditioning.

---

<sup>8</sup>Since the tension force is linear in  $\sigma$  we can easily treat both the tension and bending forces implicitly. In 2D [62] such kind of schemes have superior stability properties at low shear rates. We have implemented this scheme in 3D and noticed (via numerical experiments) no improvements over (30). Compared to (30), these schemes have higher computational cost because of the need to solve coupled linear system of equations and hence are less desirable.



## 5.4 Preconditioners

**Tension solver.** Given the configuration of the vesicle, the inextensibility operator  $\mathcal{L}$ , defined in equation (25), is solved to get the corresponding tension. Depending on the time marching scheme, the right hand side would be some smooth function, denoted here by  $b$ . Therefore, the equation for the tension is  $\mathcal{L}\sigma = b$ . A discrete analogue of this equation is ill-conditioned because by Theorem 5.2, the condition number of  $\mathcal{L}$  on  $\mathbb{S}^2$  grows as  $\mathcal{O}(p)$ . Following our method in [62], we propose using  $\mathcal{P}^{-1}\mathcal{L}^{-1}(\mathbb{S}^2)\mathcal{P}$  as a low cost preconditioner for the solving the inextensibility constraint on general surfaces, where  $\mathcal{P}$  is the projection operator that maps  $\phi$  to  $\hat{\phi}$  for any scalar function  $\phi$ . The action of  $\mathcal{L}(\mathbb{S}^2)$  on spherical harmonics is given in equation (26). In Table 1 we list the number of required GMRES iterations with and without using the preconditioner. We note that the gain is significant only for large values of  $p$ .

$p$	8	16	24	32	48
Preconditioned	3.6	8.8	14	20	39
Non-preconditioned	3.5	6.8	10	14	20

Table 1: Average number of GMRES iterations for the tension solver. The average is taken over five time steps. GMRES tolerance is  $1e-8$  for all cases.

**Position solver.** Combining equations (30), (31), and (32) one can see that the linear equation which is solved for the new position  $\mathbf{x}^{n+1}$  is in the form  $(I - \Delta t\mathcal{B})\mathbf{x}^{n+1} = \mathbf{b}$ , where  $\mathbf{b}$  is some known vector and  $\mathcal{B}$  is the bending operator. According to the spectral analysis in Section 5.1 and specifically equation set (27), the bending operator’s condition number grows cubically in the spherical harmonics’ order. To subdue its ill-conditioning, we devise a diagonal operator in the spherical harmonic space. The preconditioner is defined as  $\mathcal{P}^{-1}\Lambda\mathcal{P}$  where  $\mathcal{P}$  the projection operator, and  $\Lambda := \text{diag}\{(1 - \Delta tn^3)^{-1}\}$ . In Table 2 we report the number of GMRES iterations for the position solver with and without applying the preconditioner. We note that while the performance of the preconditioner is extremely well for low shear rates, it deteriorates for higher shear rates.

$p$	Preconditioned			Non-preconditioned		
	$\chi = 0$	15	150	0	15	150
12	2.7	12	17	12	5	3
16	2.7	12	17	21	12	3.3
24	3.6	13	19	36	23	7.4
32	3.6	13	17	55	37	9.2
48	3.6	13	17	98	45	9.4

Table 2: The average number of GMRES iterations for the position solver in the semi-implicit scheme. The step sizes are the largest stable time steps from Table 4 and the average is taken over ten time steps. The GMRES tolerance is set to  $10^{-6}$ .

## 6 Reparameterization

When membrane shear elasticity is present, as for example in Neo-Hookean constitutive laws (e.g., [68]), there is a built-in mechanism that prevents material points on the membrane from clustering together. The local surface inextensibility of vesicles prevents the extreme deformations that can be observed in drops. Nevertheless, the lack of in-plane shear resistance causes significant distortions of the point distribution, which in turn introduces unresolvable high-frequency components, excessive aliasing errors, and numerical instability (see Figure 4). To address these errors and enable long-time accurate simulations using a small number of spherical harmonics coefficients, we reparameterize the surface at each time step through a redistribution of points that seeks to minimize the high-frequency component of the spherical harmonics expansion of the surface parametrization. *Unlike previous work, this antialiasing is nonlinear: it is not done by a linear spectral projection but by solving a nonlinear variational problem.*

We define the surface  $\gamma$  as the image of an embedding of the sphere  $\mathbf{x}(\mathbf{q}) : \mathbb{S}^2 \rightarrow \mathbb{R}^3$  where  $\mathbf{q}$  is a point on  $\mathbb{S}^2$ . We assume that  $\mathbf{x}(\mathbf{q}) \in \mathcal{X}$ , a space of sufficiently smooth functions on  $\mathbb{S}^2$ . The choice of space depends on the energy; for the energy we consider below any function in  $L^2$  can be used. Many different embeddings correspond to the same surface, and our goal is to choose one that minimizes a quality measure  $E(\mathbf{x}) : \mathcal{X} \rightarrow \mathbb{R}$ . To characterize all embeddings corresponding to  $\gamma$ , we use its implicit representation, a smooth function  $F : \mathbb{R}^3 \rightarrow \mathbb{R}$ , such that  $F(\gamma) = 0$  and  $\nabla F$  does not vanish at  $\mathbf{x}$  on  $\gamma$ . Then for any parametrization  $\mathbf{x}$ ,  $F(\mathbf{x}(\mathbf{q})) = 0$ , for all  $q \in \mathbb{S}^2$ . The unit normal to  $\gamma$  can be computed as  $\mathbf{n} = \nabla F / \|\nabla F\|$ . Using this notation our problem can be formulated as a constrained optimization problem:

$$\begin{aligned} \min_{\mathbf{y} \in \mathcal{X}} E(\mathbf{y}(\mathbf{q})) \\ \text{subject to} \\ F(\mathbf{y}(\mathbf{q})) = 0, \quad \forall q \in \mathbb{S}^2, \end{aligned} \tag{33}$$

that is, find a parametrization  $\mathbf{y}(\mathbf{q})$  of  $\gamma$  minimizing the quality measure  $E$ .

In general, this may be a highly nonlinear problem requiring a computationally expensive method. At the same time, it is generally unnecessary to obtain a precise solution. We present a simple inexpensive method that yields a sufficiently accurate approximation at a low cost.

By introducing a Lagrangian  $E(\mathbf{y}) + \int_{\mathbb{S}^2} \mu F(\mathbf{y})$  and taking variations with respect to  $\mathbf{y}$  and the Lagrange multiplier  $\mu := \mu(\mathbf{q})$ , with  $\mu$  restricted to a space of sufficiently smooth functions on  $\mathbb{S}^2$ , we obtain the (strong form of the) first-order optimality conditions

$$\nabla_{\mathbf{y}} E(\mathbf{y}(\mathbf{q})) + \mu(\mathbf{q}) \nabla_{\mathbf{y}} F(\mathbf{y}(\mathbf{q})) = 0, \quad \forall \mathbf{q} \in \mathbb{S}^2 \tag{34a}$$

and

$$F(\mathbf{y}(\mathbf{q})) = 0, \quad \forall \mathbf{q} \in \mathbb{S}^2. \tag{34b}$$

Due to the special form of these equations, it is possible to eliminate  $\mu$  by taking the inner product of (34a) with  $\nabla F$ , so that

$$\nabla E \cdot \nabla F + \mu \|\nabla F\|^2 = 0, \quad \text{or} \quad \mu = -\nabla E \cdot \nabla F / \|\nabla F\|^2,$$

where we drop the subscript  $\mathbf{y}$  to simplify the notation. Substituting  $\mu$  into (34), we obtain

$$(I - \mathbf{n}(\mathbf{y}) \otimes \mathbf{n}(\mathbf{y})) \nabla E(\mathbf{y}) = 0 \quad \text{and} \quad F(\mathbf{y}) = 0. \tag{35}$$

We solve (35) using pseudo-transient continuation [34] by introducing a parameter  $\tau$ . That is, we solve

$$\dot{\mathbf{y}} + (I - \mathbf{n}(\mathbf{y}) \otimes \mathbf{n}(\mathbf{y})) \nabla E(\mathbf{y}) = 0, \quad \mathbf{y}(0) = \mathbf{x}, \quad \text{and} \quad F(\mathbf{y}) = 0, \quad (36)$$

where  $\dot{\mathbf{y}} := \frac{\partial \mathbf{y}}{\partial \tau}$ . This evolution of  $\mathbf{y}$  cannot increase the energy, since we are moving in the constraint manifold-projected steepest descent direction.<sup>9</sup> Our parametrization optimization can be viewed as a discretization of this flow using an explicit scheme:

$$\boxed{\mathbf{y}_{k+1} = \mathbf{y}_k + \Delta\tau (I - \mathbf{n}_k \otimes \mathbf{n}_k) \nabla E(\mathbf{y}_k)}. \quad (37)$$

Notice that we do not explicitly impose the constraint  $F = 0$ . In the continuous case, it is satisfied for all values of  $\tau$  if the initial value  $\mathbf{x}$  satisfies  $F(\mathbf{x}) = 0$ . Indeed, by substituting (37) to  $F(y_{k+1})$  one can show that the former is zero to first-order approximation and  $F \rightarrow 0 \forall \tau$  with  $\Delta\tau \rightarrow 0$ . Of course, one needs to control the size of  $\Delta\tau$  to avoid excessive discretization errors.

**Choosing the surface parametrization quality metric  $E$ .** Let  $Y_n^m$  ( $n = 0, 1, \dots, |m| \leq n$ ) be the spherical harmonics basis. We define a discrete filter by the attenuation factors  $a_{n,m}$  for  $(n, m)$ -th harmonic. Let  $\mathbf{y} = \sum_{n,m} \langle Y_n^m, \mathbf{y} \rangle Y_n^m$ . Then we define the energy as

$$E(\mathbf{y}) := \sum_{n,m} a_{n,m}^2 \langle Y_n^m, \mathbf{y} \rangle^2. \quad (38)$$

The variation of  $E$  with respect to  $\mathbf{y}$  yields

$$\nabla E = \sum_{n,m} a_{n,m}^2 \langle Y_n^m, \mathbf{y} \rangle Y_n^m.$$

Since we want  $E$  to penalize the high frequencies in order to minimize aliasing errors,  $a_{n,m}$  should be small for low frequencies and should grow for high frequencies. In the special case of perfect low-pass filter ( $a_{n,m} = 0$  for  $n < n_{cutoff}$  and  $a_{n,m} = 1$  for high),

$$\nabla E = \sum_{n > cutoff, m} \langle Y_n^m, \mathbf{y} \rangle Y_n^m, \quad (39)$$

We penalize the high frequency part of  $\mathbf{y}$ .<sup>10</sup>

<sup>9</sup>This can be seen as follows. Let  $\delta\mathbf{y}$ , be the change of  $\mathbf{y}$  along the projected steepest-descent direction scaled by some  $\zeta > 0$ . That is,

$$\delta\mathbf{y} = -\zeta (I - \mathbf{n} \otimes \mathbf{n}) \nabla E = -\zeta \nabla E + \zeta (\nabla E \cdot \mathbf{n}) \mathbf{n}, \quad \zeta > 0.$$

Then the change in the energy

$$\delta E = E(\mathbf{y} + \delta\mathbf{y}) - E(\mathbf{y}) = -\zeta (\|\nabla E\|_2^2 - \|\nabla E \cdot \mathbf{n}\|_2^2).$$

Since  $\nabla E = (\nabla E \cdot \mathbf{n}) \mathbf{n} + \nabla E^{\text{tangent}}$  at every point and  $\nabla E^{\text{tangent}} \cdot \mathbf{n} = 0$ , it follows that  $\delta E = -\zeta \|\nabla E^{\text{tangent}}\|_2^2$  is nonpositive. Therefore, we are guaranteed to reach a stationary point.

<sup>10</sup>We also experimented with filters associated with the inverse spectrum of the Laplace Beltrami operator on the sphere. In the spectral domain,  $a_i$  is proportional to the frequency, leading, as expected, to attenuation with coefficients proportional to the frequency squared. There is no significant difference between the two reparametrization schemes, but the antialiasing properties of the scheme that we use produced slightly better results.

**Integrating the solver with vesicle simulations.** In the context of vesicle simulations at the end of each time step, we perform reparametrization to improve the quality of the surface representation, using the scheme (37) with the quality measure  $E$  given by (39). Since the objective of reparametrization is to maximize the decay of spherical harmonic coefficients, we choose  $n_0$  in Equation (39) to be  $p/3$  where  $p$  is the order of truncated spherical harmonics series of the surface.

In Algorithm 2, we give the pseudocode for the reparametrization step. In the algorithm,

---

**Algorithm 2** Explicit reparametrization

---

**Require:**  $\mathbf{x}$

choose  $\Delta\tau$

$\mathbf{y}_0 = \mathbf{x}$

$\mathbf{g}_0 = -(I - \mathbf{n}(\mathbf{y}_0) \otimes \mathbf{n}(\mathbf{y}_0)) \nabla E(\mathbf{y}_0)$

**while**  $\|\mathbf{y} - \mathbf{y}_0\| > \rho_y$  and  $\|\mathbf{g}\| > \rho_g \|\mathbf{g}_0\|$  **do**

$\mathbf{g} = -(I - \mathbf{n}(\mathbf{y}) \otimes \mathbf{n}(\mathbf{y})) \nabla E(\mathbf{y})$

*projected gradient*

$\mathbf{y}^+ \leftarrow \mathbf{y} + \Delta\tau \mathbf{g}$

$\mathbf{y} \leftarrow \mathbf{y}^+$

**end while**

---

$\nabla E$  is given by (39). Notice, that in our implementation we use upsampling (by a factor of two). That is we first upsample  $\mathbf{x}$ , then we pseudo-time march to obtain the new points  $\mathbf{y}$ , and finally we downsample to the original resolution of  $\mathbf{x}$ . This upsampling significantly improves the quality and effectiveness of the reparametrization.

Algorithm 2 corresponds to a sequence of steepest descent steps for the constrained minimization problem (33). One could use a line search approach for nonlinear programming (e.g., with an  $\ell_2$  merit function [46]) but this requires access to  $F$  and is more complex to implement. Instead, in our approach, the parameter  $\Delta\tau$  (which corresponds to the line search step-length) can be chosen using curvature information. That is given  $\mathbf{x}$  so that  $F(\mathbf{x}) = 0$  and a perturbation in the steepest descent direction  $\Delta\tau \mathbf{g}$ , then

$$F(\mathbf{x} + \Delta\tau \mathbf{g}) \approx F(\mathbf{x}) + \Delta\tau \nabla F(\mathbf{x}) \cdot \mathbf{g} + \frac{\Delta\tau^2}{2} \mathbf{z} \cdot \nabla \nabla F(\mathbf{x}) \mathbf{z} = \frac{\Delta\tau^2}{2} \mathbf{g} \cdot \nabla \nabla F(\mathbf{x}) \mathbf{g}.$$

The Hessian  $\nabla \nabla F(\mathbf{x})$  can be computed using the parametric form of the surface. Then we can choose  $\Delta\tau$  so that  $F(\mathbf{x} + \Delta\tau \mathbf{g}) < \rho_F \|\mathbf{x}\|$ . (Notice that the trace of the Hessian is equal to the mean curvature, so roughly-speaking the pseudo-time step is inversely proportional to mean curvature.)

One additional question is how to choose  $\rho_y$  and  $\rho_g$ . Parameter  $\rho_y$  ensures that if the change in  $\mathbf{y}$  becomes small the algorithm terminates. (We choose  $\rho_y = 10^{-2} \|\mathbf{x}^{n+1} - \mathbf{x}^n\|$ , where  $\mathbf{x}^{n+1}$  and  $\mathbf{x}^n$  are the new and old positions of the vesicle evolution.)  $\rho_g$  ensures that if the gradient becomes too small the algorithm terminates. We demonstrate the effect of reparametrization on the stability of the time marching scheme in the the next section, along with numerical experiments that examine different aspects of our method.

## 7 Results

In this section, we study the accuracy, stability and complexity of our computational scheme through numerical experiments. In particular, we test (1) the accuracy of the high-order derivatives

calculation (curvature) (§4), (2) the accuracy of the smooth, the weakly-singular, and the nearly-singular quadratures (§4.1), (3) the need for and effects of reparametrization (§6), and (4) the time-marching stability properties of the explicit and semi-implicit schemes (§5). In addition to verifying our method, we also present results on the relaxation shapes of dilute suspension under shear flow, examine a two-vesicle interaction problem, study the sedimentation of a vesicle, and provide an example of a simulation with multiple vesicles.

First, we define the length and time scales for the different flow regimes we consider in our experiments. For a vesicle of area  $A$  and volume  $V$  suspended in a linear shear flow  $\mathbf{v}_\infty = \dot{\gamma}(x_3, 0, 0)$ , the non-dimensional parameters are given by

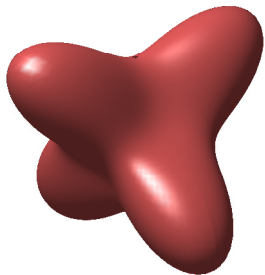
$$\begin{aligned} \text{length scale:} & R_0 = \sqrt{\frac{A}{4\pi}}, \\ \text{time scale:} & \tau = \frac{\mu R_0^3}{\kappa_B}, \\ \text{reduced volume:} & \nu = \frac{6\sqrt{\pi}V}{A^{3/2}}, \\ \text{shear rate:} & \chi = \dot{\gamma}\tau. \end{aligned}$$

Quiescent flows are characterized by the nondimensional parameter  $\tau$ , and simple shear flows are characterized by  $\tau$  and  $\chi$ .

**Derivative accuracy.** Consider a vesicle’s surface defined by

$$\mathbf{x}(u, v) = \begin{bmatrix} \rho(u, v) \sin u \cos v \\ \rho(u, v) \sin u \sin v \\ \rho(u, v) \cos u \end{bmatrix}, \quad \rho(u, v) = 1 + e^{-3Re(Y_3^2(u, v))}, \quad u \in [0, \pi], v \in [0, 2\pi]. \quad (40)$$

As we have discussed, we use equation (13) for differentiation; in Figure 3, we report the relative errors in computing the Gaussian curvature  $K$ , and the mean curvature  $H$  of this surface.



$M(p)$	$H$	$K$
162(8)	$2.44e-1$	$2.21e-1$
578(16)	$3.09e-3$	$1.68e-3$
1250(24)	$1.78e-6$	$1.36e-6$
2178(32)	$4.25e-10$	$2.94e-10$
3362(40)	$2.05e-11$	$8.40e-11$
4802(48)	$3.29e-11$	$1.27e-10$

Figure 3: *Relative errors in computing the principal curvatures  $H$  and  $K$  numerically on the shape shown. Here  $M$  is the number of spatial discretization points and  $p$  is the order of corresponding spherical harmonic approximation. The exact values of  $H$  and  $K$  are computed analytically.*

The errors decay super-algebraically with  $p$ . However, for higher values of  $p$ , the round-off errors dominate and the relative errors start to grow as  $\mathcal{O}(p^2\epsilon)$  where  $\epsilon$  is the machine precision.



$p$	Smooth integrals		Singular integrals	
	Area	Volume	Harmonic	Stokes
8	$7.19e-3$	$1.67e-3$	$1.15e-2$	$9.90e-3$
16	$1.42e-6$	$7.53e-8$	$2.50e-4$	$2.96e-4$
24	$6.79e-7$	$2.65e-13$	$2.01e-5$	$2.00e-5$
32	$2.33e-8$	$3.21e-15$	$3.25e-7$	$2.42e-7$

Table 3: *Relative errors in computing (i) smooth integrals: the area,  $A = \int_{\gamma} d\gamma = \int_U W du dv$ , and the volume,  $V = \frac{1}{3} \int_U (\mathbf{x} \cdot \mathbf{n}) W du dv$ , (ii) singular integrals: harmonic and Stokes potentials of unit function defined on the shape shown in Figure 3. The reference values are computed using a finer discretization.*

This behavior is typical for spectral methods<sup>11</sup> [21]. In the axisymmetric case [61], we advocated differentiating non-bandlimited functions as frugally as possible. This approach abates the round-off error growth to some extent. Another promising approach is proposed recently in [11] based on expressing high-order derivatives as Cauchy integrals over circular contours. We will investigate the effectiveness of these schemes in the present context and report the results at a later date.

**Accuracy of the numerical integration schemes.** Our singular integral evaluation scheme outlined in Section 4 is spectrally accurate. Other popular spectral method is the *floating partition of unity* scheme [13]. Because of the steep gradients in the partition of unity functions, this scheme loses a few digits of accuracy. Our scheme, on the other hand, computes singular integrals as accurately as smooth integrals. See [28] for a comparison of those two schemes.

We report the convergence results for the smooth and singular integral computation schemes in Table 3. We can observe that the singular integrals are computed with nearly the same order of accuracy as the smooth integrals (compare, for instance, the harmonic potential and area computations). Consequently, using very few spatial discretization points, we are able capture the essential vesicle dynamics. We show one such example in Figure 10.

**Reparametrization.** To verify the effectiveness of the reparametrization, let us consider a single vesicle in a shear flow. Let the bending modulus of the vesicle  $\kappa = 1e-2$ , the shear rate  $\chi = 15$ , and the time horizon  $T = 0.6$ . In Figure 4(a) we show the configuration of the vesicle at different time steps for two test cases. The top vesicle snapshots correspond to the case when we perform reparametrization, and the bottom set of snapshots is without reparametrization. As we can see in 4(b), the energy of the surface remains bounded when we use reparametrization. In its absence, as soon as the vesicle starts tank-treading, the energy blows up. Note that the need for maintaining the grid quality is not limited to the case when the vesicle undergoes tank-treading motion. As reported in [76], even in the case of axisymmetric gravity-induced motions of deformable drops, the Lagrangian points cluster at the tail of a drop. Moreover, a simple elimination of the tangential component of the velocity, that is, updating the positions using the normal velocity  $(\mathbf{u} \cdot \mathbf{n})\mathbf{n}$ , suffers from similar mesh degradations [76].

<sup>11</sup>In the general case, the error grows as  $\mathcal{O}(p^k \epsilon)$  when computing a  $k$ th order derivative.

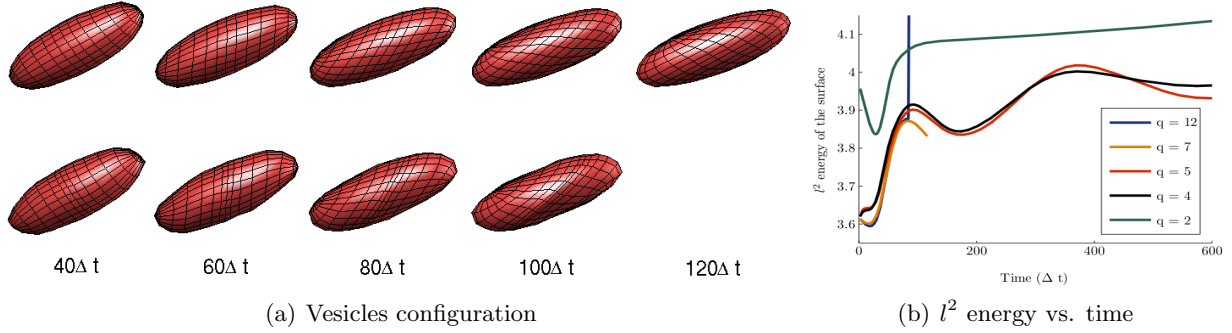


Figure 4: (a) Here we demonstrate the significance of using reparametrization for the representation of the surface. In the simulation depicted on the top row of subfigure (a), we perform surface reparametrization; in the bottom simulation we do not reparametrize. In both cases, we have taken  $p = 12$ , shear rate  $\chi = 15$ ,  $\Delta t = 3e-3$ , and we have used the semi-implicit time stepping scheme described in Section 5.3. We observe that without reparametrization, at  $t \approx 60\Delta t$ , the surface becomes highly distorted and the method diverges. Experimentally, the blow-up starts as soon as the vesicle begins to tank-tread. When we do use reparametrization, at the end of simulation, the relative error in area and volume of the vesicle are respectively  $3.49e-2$  and  $3.86e-2$ . (b) Here we plot the  $l^2$  norm of the spherical harmonics coefficients of the surface (for the case of  $p = 12$ ) vs. time for different filtering frequencies, which are denoted by  $q$ .

**Stability of the time-marching scheme.** We compare the stability constraints of the explicit and the semi-implicit schemes in simulating the motion of a vesicle suspended in simple shear flow (Figure 5). The initial shape of the vesicle is given by equation (40) in which we take  $\rho(u, v) =$

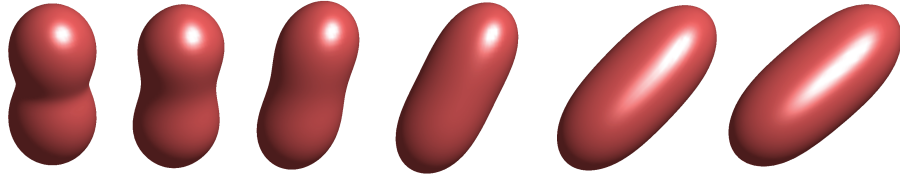


Figure 5: Relaxation shape of a vesicle suspended in simple shear flow. Due to the bending term the vesicle quickly relaxes to a shape that has small curvature, but then it gets stretched due to the effect of the shear flow. At the final equilibrium shape the vesicle tank treads. We have used this experiment to test the time-stepping schemes. The results are reported in Table 4.

$1 + Y_2^0(u, v)$ . We computed the stable time-step size  $\Delta t$  using the bisection method: starting from an arbitrary large time step we reduce it by half until the simulation is stable within a predefined time horizon. For this experiment, we have chosen the time horizon long enough so that the vesicle reaches its equilibrium shape in shear flow. The stable time steps are summarized in Table 4 as a function of the shear rate and the spatial resolution  $p$ . The explicit scheme has two shortcomings: (i) for a fixed  $p$ ,  $\Delta t$  decreases as  $\chi$  is increased, (ii) for a fixed  $\chi$  and low shear rates,  $\Delta t$  decreases dramatically as  $p$  is increased. We can explain these constraints as follows. Embedding the tension force and the constraint equation in a projection operator  $\mathcal{P}$ , we can write the non-dimensionalised

$p$	Explicit			Semi-implicit		
	$\chi = 0$	15	150	0	15	150
12	$1.04e-2$	$7.81e-3$	$9.76e-4$	$5.00e-1$	$1.02e-2$	$9.77e-4$
16	$4.67e-3$	$1.59e-3$	$3.98e-4$	$3.53e-1$	$1.02e-2$	$4.51e-4$
24	$1.27e-3$	$1.17e-3$	$1.99e-4$	$2.04e-1$	$1.02e-2$	$9.56e-4$
32	$2.05e-4$	$1.70e-4$	$1.25e-4$	$1.49e-1$	$8.15e-3$	$6.79e-4$
48	$1.00e-4$	$1.00e-4$	$5.00e-5$	$1.49e-1$	$8.15e-3$	$6.79e-4$

Table 4: Here we report stable time-step sizes for the explicit and semi-implicit schemes for a vesicle in simple shear flow. We observe that the implicit-scheme requires a time step whose size is almost independent of the spatial resolution.

evolution equation as

$$\dot{\mathbf{x}} = \mathcal{P} \left( \chi \begin{bmatrix} x_3 \\ 0 \\ 0 \end{bmatrix} + \mathcal{S}[-(\Delta_S H + 2H(H^2 - K))\mathbf{n}] \right), \quad (41)$$

where  $\mathcal{P}$  acts on a surface velocity field and eliminates the extensible component. We can easily verify that  $\mathcal{P}$  is a well-conditioned operator using the spectral analysis of Section 3. It does not contribute to the stiffness. Only the second term within brackets in (41) induces high-order stiffness. Now consider two extreme cases. When  $\chi = 0$ , the evolution equation is stiff and for this reason we observe a stringent restriction on the time-step. On the other hand, when  $\chi$  is high, the first term dominates and it attenuates the stiffness arising from the second term. For this reason, the CFL appears to be milder for  $\chi = 150$ .

The semi-implicit scheme suffers only from the first constraint and its stable time-step size is inversely proportional to  $\chi$ . But it does not suffer from CFL constraint. Hence, it allows much larger time steps, particularly for low shear rates, compared to the explicit scheme. We have performed additional simulations, reported in Figures 10, 9 and 1 in which we have observed the spatial-resolution independence of the time step. In Figure 6 we summarize the equilibrium shapes of vesicles in shear flow for different reduced volumes and shear rates.

**Simulations in presence of gravity.** When there is a difference in the enclosed fluid density ( $\rho^{\text{in}}$ ) and the ambient fluid density ( $\rho^{\text{out}}$ ), the governing equations in the presence of gravity are given by

$$\dot{\mathbf{x}} = \mathcal{S}[\mathbf{f}_b + \mathbf{f}_\sigma + \mathbf{f}_g], \quad \text{div}_\gamma(\mathcal{S}[\mathbf{f}_\sigma]) = -\text{div}_\gamma(\mathcal{S}[\mathbf{f}_b + \mathbf{f}_g]), \quad (42)$$

$$\text{where } \mathbf{f}_g = (\rho^{\text{in}} - \rho^{\text{out}})(\mathbf{g} \cdot \mathbf{x})\mathbf{n} \quad (43)$$

and  $\mathbf{g}$  is the gravitational acceleration. The non-dimensional parameter  $\hat{g} = \frac{(\rho^{\text{in}} - \rho^{\text{out}})\|\mathbf{g}\|R_0^4}{\kappa_B}$  characterizes gravitational flows. When  $\hat{g}$  is less than a threshold value  $\hat{g}^*$ , the vesicle reaches its equilibrium shape and translates as a rigid body with constant velocity. When  $\hat{g} > \hat{g}^*$ , the vesicle reaches a constant average velocity after an initial transient period but the terminal shapes are not unique: they depend in the initial shape. We show three examples in Figure 7. As a further validation of our code, we compare the sedimentation shape that we computed using this code, with the sedimentation shapes computed using our axisymmetric code [61]. We report area, volume, and position errors in Table 5.

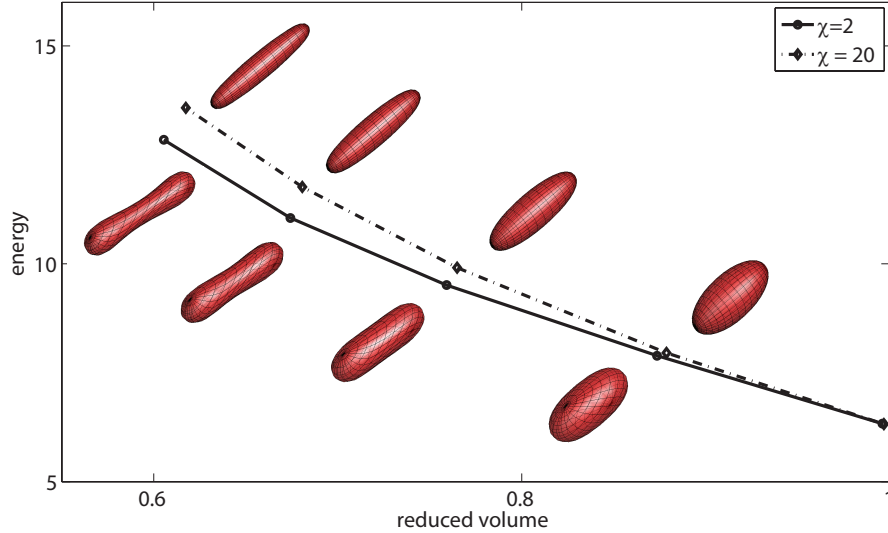


Figure 6: Terminal shapes of vesicles in a simple shear flow for various shear rates. Contrary to the general belief [35], we find that the terminal shapes depend on the shear rate, especially, for vesicles of low reduced volumes. For low shear rates (equivalently, stiffer membranes), they resemble equilibrium shapes in quiescent flows and consequently possesses lower surface energy compared to the ones at high shear rates.

$M(p)$	$\frac{ A-A_T }{A}$	$\frac{ V-V_T }{V}$	$\frac{\max\ \mathbf{x}(T)-\mathbf{x}^{\text{axi}}(T)\ }{\max\ \mathbf{x}(T)\ }$
162(8)	$3.1e-2$	$1.1e-1$	$1.5e-1$
338(12)	$4.8e-4$	$2.4e-4$	$3.2e-2$
578(16)	$6.1e-5$	$1.0e-4$	$1.8e-3$

Table 5: Relative errors in the surface area ( $A$ ) and volume ( $V$ ) at the end of the simulation shown in the first column of Figure 7. The first-order semi-implicit time marching presented in Section 5 is used for this simulation. We also report the max-norm errors in positions  $\mathbf{x}(T)$  using the solution  $\mathbf{x}^{\text{axi}}(T)$  of the axisymmetric solver [62] as reference.  $M$  is the number of spatial discretization points,  $p$  is the order of spherical harmonic approximation, and the time-step size  $\Delta t$  is set to  $\mathcal{O}\left(\frac{T}{p}\right)$ . The axisymmetric solution is obtained using 64 spatial discretization points and a smaller time-step size.

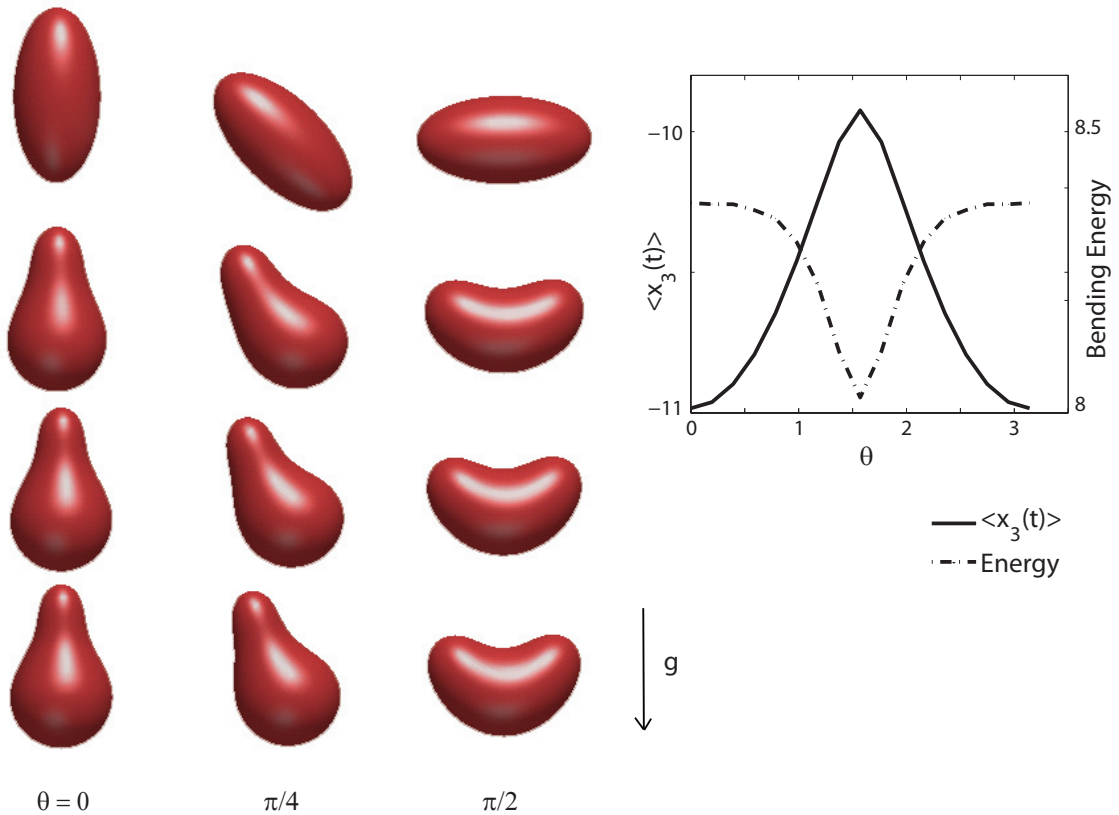


Figure 7: Snapshots of the sedimentation of ellipsoidal vesicles ( $\mathbf{x}(u, v) = (\sin u \cos v, \sin u \sin v, 2 \cos u)$ ). In all of the three cases, the initial vesicle shape is the same but their orientations ( $\theta$ ) with respect to the axis of gravitational force vary. Unlike the equilibrium shapes in the absence of gravity, the terminal shapes depend both on the reduced volume of the vesicle and on  $\theta$ . In the right image, we plot the average height of the vesicle at a specific instant  $t > 0$ , denoted by  $\langle x_3(t) \rangle$ , and also the bending energy at that instant with respect to initial orientation  $\theta$ . We can conclude from this plot that the vesicle with  $\theta = 0$  (gourd shape) is the most efficient in sedimentation and the one with  $\theta = \pi/2$  (ellipsoidal-cap shape) is least efficient. The bending energy, on the other hand, is least for the latter.

**Multiple vesicles.** The main additional component required for simulating vesicle-vesicle interactions is the near singular evaluation scheme presented in Section 4. Because we have not incorporated collision detection, near singular integral evaluation not only effects the overall accuracy but also the stability of the time-stepping scheme. Although more sophisticated approaches for nearly singular evaluations exist, we have used a simpler scheme based on upsampling. We estimate the accuracy of using upsampling for nearly-singular integration (Figure 8). Let us emphasize that

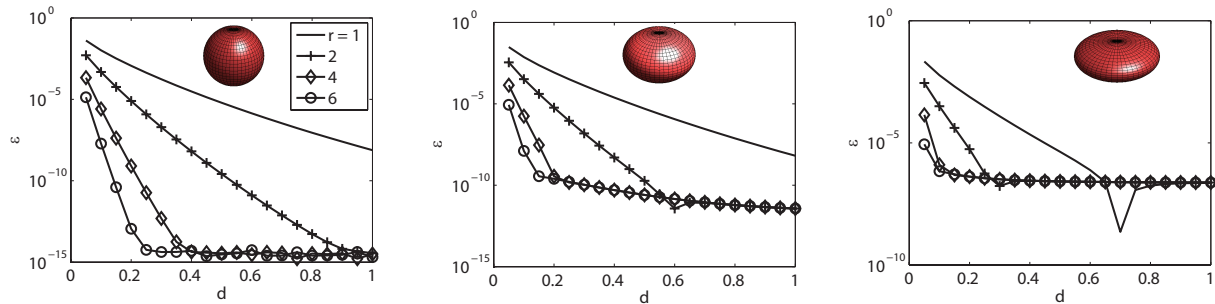


Figure 8: *Relative errors  $\epsilon$  in computing nearly-singular integrals at an evaluation point whose distance from the vesicle is  $d$  using the scheme discussed in Section 4 with an oversampling ratio  $r$ . For distant points, oversampling improves the accuracy by only a few digits and eventually as we move away from the vesicle (eg.,  $d > 0.6$  in the rightmost figure) there is no advantage of oversampling. On the other hand, for points closer to the vesicle, oversampling improves the accuracy significantly. Improving accuracy for close interactions is important because it has a direct effect on the overall stability of the numerical simulation. For instance, without any oversampling, the simulation in Figure 9 breaks down when  $t > 70\Delta t$  because of numerical instabilities.*

for a provably accurate scheme for nearly-singular integrals one needs to use the method discussed in [67].

Next, we present two simulations with multiple vesicles. In the first (Figure 9), we consider two vesicles suspended in a simple shear flow. Their initial vertical separation  $\delta$  gets magnified to  $\Delta$ . In dilute suspensions, this sort of pairwise interaction is a commonplace and exhibits similar behavior. A detailed study of the influence of various parameters on the pairwise interaction will be presented in a separate article. In Figure 10, we show two simulations that were performed using very few discretization points per vesicle.

## A Proofs

*Proof of Theorem 5.1.* The single layer potential satisfies the homogeneous Stokes PDE

$$-\mu\Delta\mathbf{u} + \nabla p = 0, \quad \nabla \cdot \mathbf{u} = 0. \quad (44)$$

inside and outside of  $S^2$ , combined with velocity continuity and a jump in tractions across the interface. We prove the theorem by first finding a solution to (44). We write the fluid velocity  $\mathbf{u}$

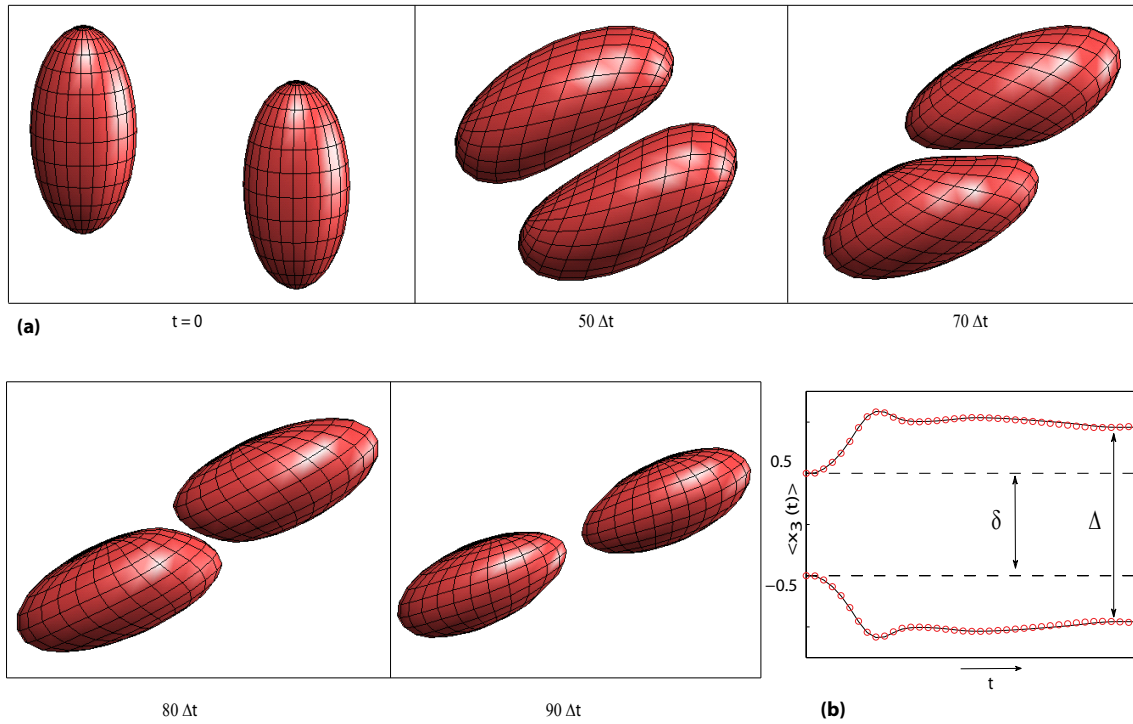


Figure 9: (a) Snapshots of two vesicles suspended in a simple shear flow. (b) Evolution of the average height  $\langle x_3(t) \rangle$ , or equivalently the height of center of mass, of the two vesicles. To show convergence of the numerical scheme, we plot the result obtained using two different discretizations: a finer discretization ( $p = 20$ ) marked by the solid line and a coarser discretization ( $p = 12$ ) marked by “o”. The dashed lines represent the path each vesicle would have followed in the absence of the other. A consequence of this pairwise interaction is that the initial vertical separation  $\delta$  of the center of masses gets magnified to  $\Delta$ . This phenomenon is well studied for suspension of drops and elastic capsules [27, 37].



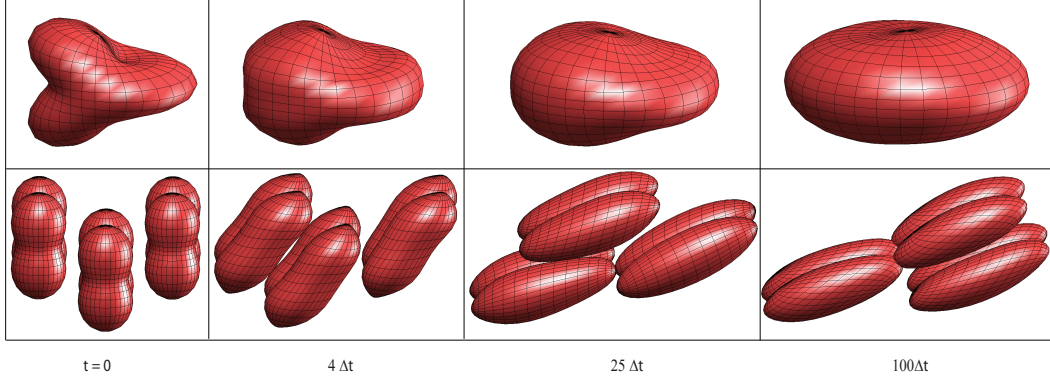


Figure 10: *Top row: snapshots of a freely suspended vesicle relaxing to equilibrium. Bottom row: Multiple vesicle suspension in shear flow. In this example, we used a modest 98 ( $p = 6$ ) spatial discretization points per vesicle. The change in volume and surface area due to numerical errors was less than 1%. Evidently, while finer details are missed (see for example the simulation in Figure 1), our method captures the essential dynamics with very few discretization points.*

and the pressure  $p$  as

$$\mathbf{u} = f(r)\mathbf{V}_n^m + g(r)\mathbf{W}_n^m \quad (45)$$

$$p = \frac{h(r)}{r}Y_n^m \quad (46)$$

and substitute in (44). We get the following three ordinary differential equations for  $f, g$  and  $h$ :

$$r^2 f_{rr} + 2r f_r - \frac{r}{2n+1} h_r - (n+1)(n+2)f = 0 \quad (47)$$

$$r^2 g_{rr} + 2r g_r + \frac{r}{2n+1} h_r - n(n-1)g + \frac{n}{2n+1} h = 0 \quad (48)$$

$$(n+1)r f_r - n r g_r + (n+1)(n+2)f + n(n-1)g = 0 \quad (49)$$

Solving these equations analytically, we obtain four sets of basic solutions,

	(i)	(ii)	(iii)	(iv)
$f$	$\frac{1}{r^{n+2}}$	$-\frac{n}{(n+1)r^n}$	0	$\frac{2nr^{n+1}}{(2n+1)(2n+3)(n+1)}$
$g$	0	$\frac{2}{(2n-1)r^n}$	$r^{n-1}$	$\frac{r^{n+1}}{2n+1}$
$h$	0	$-\frac{2n(2n+1)}{(n+1)r^n}$	0	$-2r^{n+1}$

In the exterior region ( $r > 1$ ), only (i) and (ii) are admissible as the other two are unbounded when  $r \rightarrow \infty$ . In the interior region, only (iii) and (iv) are admissible as the other two are singular at  $r = 0$ . Therefore, the velocity field  $\mathbf{u}$  can be expressed as a linear combination of two solutions in the exterior and two in the interior, leading to four unknown constants. Two of these are determined by enforcing the velocity continuity across  $S^2$  and the remaining two are determined by the jump

in traction. Let us first consider  $\mathcal{S}[\mathbf{V}_n^m]$ , for which,

$$\llbracket [-\mu(\nabla \mathbf{u} + \nabla \mathbf{u}^T) \mathbf{n} + p \mathbf{n}] \rrbracket_{S^2} = \mathbf{V}_n^m. \quad (50)$$

We get two equations by taking inner products with  $\mathbf{V}_n^m$  and  $\mathbf{W}_n^m$ . Solving these equations, we get a closed form expression for the velocity  $\mathbf{u}$  which is same as the single-layer potential and is given by

$$\mathcal{S}[\mathbf{V}_n^m](r, u, v) = \begin{cases} \frac{n}{(2n+1)(2n+3)r^{n+2}} \mathbf{V}_n^m & \text{for } r \geq 1 \\ \frac{nr^{n+1}}{(2n+1)(2n+3)} \mathbf{V}_n^m + \frac{(n+1)(r^{n-1}-r^{n+1})}{2(2n+1)} \mathbf{W}_n^m & \text{for } r \leq 1. \end{cases} \quad (51)$$

The result (24a) is obtained by substituting  $r = 1$  in the above equation. Similarly, equating the traction jump to  $\mathbf{W}_n^m$ , we get the following solution,

$$\mathcal{S}[\mathbf{W}_n^m](r, u, v) = \begin{cases} \frac{n}{2(2n+1)} \left( \frac{1}{r^{n+2}} - \frac{1}{r^n} \right) \mathbf{V}_n^m + \frac{n+1}{(2n-1)(2n+1)r^n} \mathbf{W}_n^m & \text{for } r \geq 1 \\ \frac{(n+1)r^{n-1}}{(2n-1)(2n+1)} \mathbf{W}_n^m & \text{for } r \leq 1. \end{cases} \quad (52)$$

and (24b) is obtained by substituting  $r = 1$ . ■

*Proof of Theorem 5.2.* The proof follows directly from (24). First we write the integrand in terms of the vector harmonics:

$$Y_n^m \Delta_\gamma \mathbf{x} + \nabla_\gamma Y_n^m = \frac{(n+2)\mathbf{V}_n^m + (n-1)\mathbf{W}_n^m}{2n+1}. \quad (53)$$

Then using Theorem 5.1, we have

$$\mathcal{S}[Y_n^m \Delta_\gamma \mathbf{x} + \nabla_\gamma Y_n^m] = \frac{n(n+2)}{(2n+1)^2(2n+3)} \mathbf{V}_n^m + \frac{(n-1)(n+1)}{(2n-1)(2n+1)^2} \mathbf{W}_n^m \quad (54)$$

The result follows by substituting the identities

$$\nabla_\gamma \cdot \mathbf{V}_n^m = -(n+1)(n+2)Y_n^m, \quad \text{and} \quad \nabla_\gamma \cdot \mathbf{W}_n^m = -n(n+1)Y_n^m \quad (55)$$

in equation (54). ■

*Proof of Theorem 5.3.* The proof follows from Theorem 5.1 and the identities  $\Delta_\gamma \mathbf{V}_n^m = -n(n-1)\mathbf{V}_n^m$  and  $\Delta_\gamma \mathbf{W}_n^m = (n+1)(n+2)\mathbf{W}_n^m$ . ■

## B Analytical Solutions

If the initial vesicle shape is a sphere, the bending force is zero and the tension and the velocity can be computed analytically using Theorems 5.1 and 5.2. The evolution equation reduces to

$$\dot{\mathbf{x}} = \mathcal{P} \mathbf{v}_\infty, \quad \sigma = \llbracket p \rrbracket + \mathcal{L}^{-1} D \mathbf{v}_\infty. \quad (56)$$

Because of the local inextensibility and the constant enclosed volume, the vesicle behaves like a rigid body in any type of external flow. In the absence of any external flow, the tension is isotropic and is simply the difference in external and internal pressure ( $\llbracket p \rrbracket$ ). Now let us consider three

different types of external flows. Assume the radius of the sphere is  $R_0$  and its center is located at origin. The tension and the rotational velocity  $\omega_0$  in each case are given by,

	$\mathbf{v}_\infty$	$\sigma$	$\omega_0$
linear	$\dot{\gamma}(x_3, 0, 0)$	$\mu\dot{\gamma}R_0 \frac{35}{22} \sqrt{\frac{2\pi}{15}} (Y_2^1 + Y_2^{-1})$	$\dot{\gamma}/2$
parabolic	$\dot{\gamma}(x_3^2, 0, 0)$	$\mu\dot{\gamma}R_0^2 \left( \sqrt{\frac{2\pi}{3}} (Y_1^1 + Y_1^{-1}) + \frac{2\sqrt{21\pi}}{23} (Y_3^1 + Y_3^{-1}) \right)$	0
extensional	$\dot{\gamma}(x_1, -x_2, 0)$	$\mu\dot{\gamma}R_0 \frac{35}{11} \sqrt{\frac{2\pi}{15}} (Y_2^2 + Y_2^{-2})$	0

The translational velocity  $U_0$  is zero for the linear and extensional flows. In the case of parabolic flow, the translational velocity attains a non-zero value that can be derived using classical Faxen's law [38]. Additionally, we can compute the interior and exterior fluid velocity using (51) and (52). These results are easily obtained using Theorems 5.1 - 5.3 and they can be used as a reference for numerical simulations.

## C Pseudocodes

---

**Algorithm 3** Forward spherical harmonics transform

---

```

for  $j = 0$  to  $p$  do
   $\hat{\mathbf{x}}^m(u_j) = \int_0^{2\pi} \mathbf{x}(u_j, v) e^{-imv} W(u_j, v) dv, \quad \text{for } |m| \leq p$ 
   $\approx \frac{\pi}{p} \sum_{k=1}^{2p} \mathbf{x}(u_j, v_k) e^{-imv_k} W(u_j, v_k)$  // trapezoidal rule and FFT
end for // Can be done in  $\mathcal{O}(p^2 \log p)$ 
for  $m = -p$  to  $p$  do
  for  $n = |m|$  to  $p$  do
     $\hat{\mathbf{x}}_n^m = \int_0^\pi \bar{P}_n^m(u) \hat{\mathbf{x}}^m(u) du$ 
     $\approx \sum_{j=0}^p (\lambda_j / \sin u_j) \bar{P}_n^m(u_j) \hat{\mathbf{x}}^m(u_j)$  // Gaussian quadrature and FLT
  end for
end for // Can be done in  $\mathcal{O}(p^2 (\log p)^2)$ 

```

---

## D Formulas

Here, we summarize the formulas for the first fundamental form coefficients  $E, F$  and  $G$ , the second fundamental form coefficients  $L, M$  and  $N$ , the unit normal to the surface  $\mathbf{n}$ , the Gaussian curvature  $H$ , mean curvature  $K$ , the Laplace-Beltrami operator  $\Delta_\gamma$ , the surface gradient  $\nabla_\gamma$  of a scalar  $\phi$ , the surface divergence  $\nabla_\gamma \cdot$  of a vector  $\mathbf{f}$ , and the surface energy  $\mathcal{E}$ :

Symbol	Definition	Symbol	Definition
$E$	$\mathbf{x}_u \cdot \mathbf{x}_u$	$N$	$\mathbf{x}_{vv} \cdot \mathbf{n}$
$F$	$\mathbf{x}_u \cdot \mathbf{x}_v$	$H$	$\frac{1}{2} \frac{EN - 2FM + GL}{W^2}$
$G$	$\mathbf{x}_v \cdot \mathbf{x}_v$	$K$	$\frac{LN - M^2}{W^2}$
$W$	$\sqrt{EG - F^2}$	$\Delta_\gamma \phi$	$\frac{1}{W} \left( \frac{E\phi_v - F\phi_u}{W} \right)_v + \frac{1}{W} \left( \frac{G\phi_u - F\phi_v}{W} \right)_u$
$\mathbf{n}$	$\frac{\mathbf{x}_u \times \mathbf{x}_v}{W}$	$\Delta_\gamma \mathbf{x}$	$(\Delta_\gamma x_1, \Delta_\gamma x_2, \Delta_\gamma x_3)$
$L$	$\mathbf{x}_{uu} \cdot \mathbf{n}$	$\nabla_\gamma \cdot \mathbf{f}$	$\frac{G\mathbf{f}_u - F\mathbf{f}_v}{W^2} \mathbf{x}_u + \frac{E\mathbf{f}_v - F\mathbf{f}_u}{W^2} \mathbf{x}_v$
$M$	$\mathbf{x}_{uv} \cdot \mathbf{n}$	$\nabla_\gamma \phi$	$\frac{G\mathbf{x}_u - F\mathbf{x}_v}{W^2} \phi_u + \frac{E\mathbf{x}_v - F\mathbf{x}_u}{W^2} \phi_v$

The bending and tension forces (4) are obtained by taking the  $L^2$ -gradient of the surface energy, that is, the total force  $\mathbf{f}_b + \mathbf{f}_\sigma = -\frac{\delta \mathcal{E}}{\delta \mathbf{x}}$ . See [65] for a detailed derivation.

## References

- [1] U. M. Ascher and L. R. Petzold. *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1998.
- [2] U. M. Ascher, S. J. Ruuth, and B. T. R. Wetton. Implicit-explicit methods for time-dependent partial differential equations. *SIAM Journal on Numerical Analysis*, 32(3):797–823, 1995.
- [3] K. E. Atkinson. The numerical solution of laplace’s equation in three dimensions. *SIAM Journal on Numerical Analysis*, 19(2):263–274, 1982.
- [4] K. E. Atkinson. *The Numerical Solution of Integral Equations of the Second Kind*. Cambridge University Press, 1997.
- [5] J. T. Beale, T. Y. Hou, and J. Lowengrub. Convergence of a boundary integral method for water waves. *SIAM Journal on Numerical Analysis*, 33(5):1797–1843, 1996.
- [6] J. Beaucourt, F. Rioual, T. Séon, T. Biben, and C. Misbah. Steady to unsteady dynamics of a vesicle in a flow. *Physical Review E*, 69(1):011906, Jan 2004.
- [7] T. Biben, A. Farutin, and C. Misbah. Numerical study of 3d vesicles under flow: discovery of new peculiar behaviors. *arXiv:0912.4702v1*.
- [8] T. Biben, K. Kassner, and C. Misbah. Phase-field approach to three-dimensional vesicle dynamics. *Physical Review E*, 72(4):41921, 2005.
- [9] G. Biros, L. Ying, and D. Zorin. A fast solver for the Stokes equations with distributed forces in complex geometries. *Journal of Computational Physics*, 193(1):317–348, 2003.

- [10] M. I. G. Bloor and M. J. Wilson. Method for efficient shape parametrization of fluid membranes and vesicles. *Physical Review E*, 61(4):4218–4229, Apr 2000.
- [11] F. Bornemann. Accuracy and stability of computing high-order derivatives of analytic functions by cauchy integrals. *arXiv:0910.1841v3*.
- [12] J. P. Boyd. Chebyshev and fourier spectral methods, 1999.
- [13] O. P. Bruno and L. A. Kunyansky. A fast, high-order algorithm for the solution of surface scattering problems: basic implementation, tests, and applications. *Journal of Computational Physics*, 169(1):80–110, 2001.
- [14] 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.
- [15] 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.
- [16] I. Cantat and C. Misbah. Dynamics and similarity laws for adhering vesicles in haptotaxis. *Physical Review Letters*, 83(1):235–238, Jul 1999.
- [17] I. Cantat, K. Kassner, and C. Misbah. Vesicles in haptotaxis with hydrodynamical dissipation. *The European Physical Journal E: Soft Matter and Biological Physics*, 10:175–189, 2003.
- [18] B. Cichocki, R. B. Jones, R. Kutteh, and E. Wajnryb. Friction and mobility for colloidal spheres in stokes flow near a boundary: The multipole method and applications. *The Journal of Chemical Physics*, 112(5):2548–2561, 2000.
- [19] V. Cristini, J. Blawdziewicz, and M. Loewenberg. An adaptive mesh algorithm for evolving surfaces: simulations of drop breakup and coalescence. *Journal of Computational Physics*, 168(2):445–463, 2001.
- [20] P. Dimitrakopoulos. Interfacial dynamics in Stokes flow via a three-dimensional fully-implicit interfacial spectral boundary element algorithm. *Journal of Computational Physics*, 225(1):408–426, 2007.
- [21] W. S. Don and A. Solomonoff. Accuracy enhancement for higher derivatives using Chebyshev collocation and a mapping technique. *SIAM Journal on Scientific Computing*, 18(4):1040–1055, 1997.
- [22] 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.
- [23] Q. Du, C. Liu, and X. 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.
- [24] G. Dumas and A. Leonard. A divergence-free spectral expansions method for three-dimensional flows in spherical-gap geometries. *Journal of Computational Physics*, 111(2):205–219, 1994.

- [25] F. Feng and W. S. Klug. Finite element modeling of lipid bilayer membranes. *Journal of Computational Physics*, 220(1):394–408, 2006.
- [26] F. Feng and W.S. Klug. Finite element modeling of lipid bilayer membranes. *Journal of Computational Physics*, 220(1):394–408, 2006.
- [27] C. Pozrikidis G. Breyiannis. Simple shear flow of suspensions of elastic capsules. *Theoretical and Computational Fluid Dynamics*, 13:327–347, 2000.
- [28] M. Ganesh and I. G. Graham. A high-order algorithm for obstacle scattering in three dimensions. *Journal of Computational Physics*, 198(1):211–242, 2004.
- [29] Z. Gimbutas and L. Greengard. Short note: A fast and stable method for rotating spherical harmonic expansions. *Journal of Computational Physics*, 228(16):5621–5627, 2009.
- [30] I. G. Graham and I. H. Sloan. Fully discrete spectral boundary integral methods for helmholtz problems on smooth closed surfaces in  $\mathbb{R}^3$ . *Numerische Mathematik*, 92, 2002.
- [31] E. L. Hill. The theory of vector spherical harmonics. *American Journal of Physics*, pages 211–214, 1954.
- [32] X. Jiao, A. Colombi, X. Ni, and J.C. Hart. Anisotropic Mesh Adaptation for Evolving Triangulated Surfaces. In *Proceedings of the 15th International Meshing Roundtable*, volume 51, page 173. Springer Verlag, 2006.
- [33] S. R. Keller and R. Skalak. Motion of a tank-treading ellipsoidal particle in a shear flow. *journal of fluid mechanics*, 120:27–47, 1982.
- [34] C. T. Kelley and David E. Keyes. Convergence analysis of pseudo-transient continuation. *SIAM Journal on Numerical Analysis*, 35:508–523, 1998.
- [35] M. Kraus, W. Wintz, U. Seifert, and R. Lipowsky. Fluid vesicles in shear flow. *Physical Review Letters*, 77(17), 1996.
- [36] S. Kwak and C. Pozrikidis. Adaptive triangulation of evolving, closed, or open surfaces by the advancing-front method. *Journal of Computational Physics*, 145(1):61–88, 1998.
- [37] E. Lac, A. Morel, and D. Barthes-Biesel. Hydrodynamic interaction between two identical capsules in simple shear flow. *Journal of Fluid Mechanics*, 573(-1):149–169, 2007.
- [38] L. G. Leal. *Advanced Transport Phenomena: Fluid Mechanics and Convective Transport Processes*. Cambridge University Press. 2007.
- [39] S. H. Lo. A new mesh generation scheme for arbitrary planar domains. *International Journal for Numerical Methods in Engineering*, 21(8):1403–1426, 1985.
- [40] M. Loewenberg and E. J. Hinch. Collision of two deformable drops in shear flow. *Journal of Fluid Mechanics*, 338:299–315, 1997.
- [41] M. Loewenberg and EJ Hinch. Numerical simulation of a concentrated emulsion in shear flow. *Journal of Fluid Mechanics*, 321(8):395–419, 1996.

- [42] R. Lohner. Regridding surface triangulations. *Journal of Computational Physics*, 126(1):1–10, 1996.
- [43] 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.
- [44] M. J. Mohlenkamp. A fast transform for spherical harmonics. *Journal of Fourier analysis and applications*, 5(2):159–184, 1999.
- [45] J.-C. Nedelec. *Acoustic and Electromagnetic Equations*. Springer–Verlag, New York, 2000.
- [46] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, 1999.
- [47] S.A. Orszag. Fourier series on spheres. *Monthly Weather Review*, 102:56–75, 1974.
- [48] C. Pozrikidis. Interfacial dynamics for Stokes flow. *Journal of Computational Physics*, 169: 250–301, 2001.
- [49] C. Pozrikidis. Numerical simulation of the flow-induced deformation of red blood cells. *Annals of Biomedical Engineering*, 31(10):1194–1205, 2003.
- [50] C. Pozrikidis. Axisymmetric motion of a file of red blood cells through capillaries. *Physics of Fluids*, 17(3):14, 2005.
- [51] A. Rahimian, I. Lashuk, S.K. Veerapaneni, A. Chandramowlishwaran, D. Malhotra, L. Moon, R. Sampath, A. Shringarpure, J. Vetter, R. Vuduc, D. Zorin, and G. Biros. Petascale direct numerical simulation of blood flow on 200k cores and heterogeneous architectures. In *SC '10: Proceedings of the 2010 ACM/IEEE conference on Supercomputing*, pages 1–12, Piscataway, NJ, USA, 2010. IEEE Press.
- [52] A. Rahimian, S. K. Veerapaneni, and G. Biros. Dynamic simulation of locally inextensible vesicles suspended in an arbitrary two-dimensional domain, a boundary integral method. *Journal of Computational Physics*, 229(18):6466–6484, 2010. ISSN 0021-9991.
- [53] S. Ramanujan and C. Pozrikidis. Deformation of liquid capsules enclosed by elastic membranes in simple shear flow: large deformations and the effect of fluid viscosities. *Journal of Fluid Mechanics*, 361:117–143, 1998.
- [54] Y. Saad. *Iterative Methods for Sparse Linear Systems*. PWS Publishing Company, 1996.
- [55] R. Schmitz and B. U. Felderhof. Creeping flow about a spherical particle. *Physica A: Statistical and Theoretical Physics*, 113(1-2):90 – 102, 1982.
- [56] U. Seifert, K. Berndl, and R. Lipowsky. Shape transformations of vesicles: Phase diagram for spontaneous- curvature and bilayer-coupling models. *Physical Review A*, 44(2):1182–1202, 1991.
- [57] S. Sukumaran and U. Seifert. Influence of shear flow on vesicles near a wall: A numerical study. *Physical Review E*, 64, 2001.



- [58] A.-K. Tornberg and M.J. Shelley. Simulating the dynamics and interactions of flexible fibers in Stokes flows. *Journal of Computational Physics*, 196:8–40, May 2004.
- [59] L. N. Trefethen. *Spectral methods in MATLAB*. Society for Industrial Mathematics, 2000.
- [60] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, and YJ Jan. A front-tracking method for the computations of multiphase flow. *Journal of Computational Physics*, 169(2):708–759, 2001.
- [61] S. K. Veerapaneni, D. Gueyffier, G. Biros, and D. Zorin. A numerical method for simulating the dynamics of 3D axisymmetric vesicles suspended in viscous flows. *Journal of Computational Physics*, 228(19):7233–7249, 2009.
- [62] 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*, 228(7):2334–2353, 2009.
- [63] P. M. Vlahovska, M. Loewenberg, and J. Blawdziewicz. Deformation of a surfactant-covered drop in a linear flow. *Physics of Fluids*, 17(10), 2005.
- [64] E. J. Weinberg. Monopole vector spherical harmonics. *Physical Review D*, 49:1086–1092, 1994.
- [65] J. L. Weiner. On a problem of Chen, Willmore, et al. *Indiana University Mathematics Journal*, 27:19–35, 1978.
- [66] L. Ying, G. Biros, and D. Zorin. A kernel-independent adaptive fast multipole method in two and three dimensions. *Journal of Computational Physics*, 196(2):591–626, 2004.
- [67] L. Ying, G. Biros, and D. Zorin. A high-order 3D boundary integral equation solver for elliptic pdes in smooth domains. *Journal of Computational Physics*, 219(1):247–275, 2006.
- [68] H. Zhao, A. H. G. Isfahani, L. N. Olson, and J. B. Freund. A Spectral Boundary Integral Method for Flowing Blood Cells. *Journal of Computational Physics*, 229(10):3726–3744, 2010.
- [69] H. Zhou and C. Pozrikidis. Deformation of liquid capsules with incompressible interfaces in simple shear flow. *Journal of Fluid Mechanics*, 283:175–200, 1995.
- [70] A. Z. Zinchenko and R. H. Davis. An efficient algorithm for hydrodynamical interaction of many deformable drops. *Journal of Computational Physics*, 157(2):539–587, 2000.
- [71] A. Z. Zinchenko and R. H. Davis. Shear flow of highly concentrated emulsions of deformable drops by numerical simulations. *Journal of Fluid Mechanics*, 455:21–61, 2002.
- [72] A. Z. Zinchenko and R. H. Davis. Large-scale simulations of concentrated emulsion flows. *Philosophical Transactions of the Royal Society London A*, 361:813–845, 2003.
- [73] A. Z. Zinchenko and R. H. Davis. A boundary-integral study of a drop squeezing through interparticle constrictions. *Journal of Fluid Mechanics*, 564:227–266, 2006.
- [74] A. Z. Zinchenko and R. H. Davis. Algorithm for direct numerical simulation of emulsion flow through a granular material. *Journal of Computational Physics*, 227(16):7841–7888, 2008.

- [75] A. Z. Zinchenko and R. H. Davis. Squeezing of a periodic emulsion through a cubic lattice of spheres. *Physics of Fluids*, 20:040803, 2008.
- [76] A. Z. Zinchenko, M. A. Rother, and R. H. Davis. A novel boundary-integral algorithm for viscous interaction of deformable drops. *Physics of Fluids*, 9:1493, 1997.
- [77] A. Z. Zinchenko, M. A. Rother, and R. H. Davis. Cusping, capture, and breakup of interacting drops by a curvatureless boundary-integral algorithm. *Journal of Fluid Mechanics*, 391:249–292, 1999.