

Boundary Integral Method for the Flow of Vesicles with Viscosity Contrast in Three Dimensions

Abtin Rahimian^a, Shravan K. Veerapaneni^b, Denis Zorin^a, George Biros^c

^a*Courant Institute of Mathematical Sciences, New York University, New York, NY 10003*

^b*Department of Mathematics, University of Michigan, MI 48109*

^c*Institute for Computational Engineering and Science, University of Texas at Austin, Austin 78712*

Abstract

We propose numerical algorithms for the simulation of the dynamics of three-dimensional vesicles suspended in viscous Stokesian fluid. Our method is an extension of our previous work [S. K. Veerapaneni, A. Rahimian, G. Biros, and D. Zorin. A Fast Algorithm for Simulating Vesicle Flows in Three Dimensions. *Journal of Computational Physics*, 230(14):5610–5634, 2011] to flows with viscosity contrast. This generalization requires a change in the boundary integral formulation of the solution, in which a double-layer Stokes integral is introduced, and leads to changes in the fluid dynamics due to the viscosity contrast of the vesicles, which can no longer be efficiently resolved with existing algorithms.

In this paper we describe the algorithms needed to handle flows with viscosity contrast accurately and efficiently. We show that a globally semi-implicit method does not have any time-step stability constraint for flows with single and multiple vesicles with moderate viscosity contrast and the computational cost per simulation unit time is comparable to or less than that of an explicit scheme. Automatic oversampling adaptation enables us to achieve high accuracy with very low spectral resolution. We conduct numerical experiments to investigate the stability, accuracy, and the computational cost of the algorithms. Overall, our method achieves several orders of magnitude speed-up compared to the standard explicit schemes.

Keywords: Stokes flow, Boundary integral, Galerkin method, Viscosity contrast, Implicit time stepping

1. Introduction

Vesicles 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 of biomembrane mechanics. Vesicle-inspired mechanical models can be used to approximate red blood cell mechanics. For example, at equilibrium, vesicles and 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 blood cell dynamics, such as tank-treading and tumbling motions, are shared with vesicles [4, 13, 15].

The vesicle evolution dynamics is characterized by an interplay between membrane's 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.

Email addresses: arahimian@acm.org (Abtin Rahimian), shravan@umich.edu (Shravan K. Veerapaneni), dzorin@cims.nyu.edu (Denis Zorin), gbiros@acm.org (George Biros)

In this paper, we present algorithms for the simulation of three-dimensional vesicle flows with contrast between the viscosity of the suspending fluid and that of the fluid enclosed inside each vesicle. This work is an extension of our previous work presented in [27].

1.1. Overview and Contributions

Our method is based on Galerkin formulation corresponding to Lagrangian tracking of spectral collocation points placed on the membrane of the vesicle. We represent the vesicles in the spherical harmonic basis. For weakly-singular integrals, we use the high-order scheme proposed by Graham and Sloan [11] that enables accurate simulations with a small number of degrees of freedom per vesicle, compared to low-order schemes. For the position update in time, we present variants of semi-implicit marching scheme first derived for advection-diffusion equations [1] and then applied to integral-equation based fluid-structure interaction problems in [26].

The time-marching scheme requires the solution to a linear system of equations at each time step, which we perform using a Krylov iterative method, GMRES [24]. The problem of poor conditioning is addressed by a preconditioner based on the analytically-obtained spectrum of the operators on a unit sphere [27].

The main contributions of this article are:

- *Efficient and accurate treatment of vesicles with viscosity contrast.* The flow of vesicles with viscosity contrast requires introduction of a double-layer Stokes integral with the velocity as the density (Eq. (3a)). The resulting system behaves differently from the equal-viscosity case. As the viscosity contrast of vesicles increases, they behave more like rigid bodies, which causes the vesicles to get very close to each other under certain flow conditions. Due to this proximity, explicit or block implicit methods become inaccurate and unstable (Table 4). We propose and analyze a global implicit scheme, in which the interaction of vesicles is treated implicitly (Section 3.4). We show that the stable time step for this scheme is orders of magnitude larger than the explicit scheme and its computational cost (per unit time) is superior to that of the explicit or block implicit schemes (Table 5).
- *Galerkin formulation.* We use a Galerkin formulation for the boundary integral solution. In [27], we used pseudo-spectral method, which has the same accuracy as the Galerkin method, but requires twice as many variables (the grid points) compared to spherical harmonic coefficients used in the Galerkin method and, unlike our Galerkin method, leads to overdetermined systems.
- *Characterization and reduction of aliasing in differentiation.* As our simulations involve differentiation and sampling of various functions (such as the force distribution on the surface) that may not be band-limited, aliasing may occur. Specifically, under-sampled high-frequency components may contribute to the low-frequency content of various fields we compute. In order to facilitate simulations with a low spherical harmonics' truncation order, we introduce an algorithm (Algorithm 1) which automatically adjusts the upsampling rate of functions for differentiation. In addition to this, we use reparametrization of the surface (Algorithm 3) from [27].

1.2. Limitations

We restrict our attention to suspensions of vesicles in unbounded domains. We have ignored inertial terms, so the overall method is restricted to low Reynolds numbers. Only vesicles with spherical topology are considered and topological changes are not allowed. We extend our method to vesicles with viscosity contrast, but our method does not extend to the limiting cases of bubbles ($\lambda = 0$) and solid particles ($\lambda = \infty$). Another important limitation of our scheme is the lack of adaptivity (both in space and time). This lack of adaptivity manifests itself in the evaluation of nearly-singular integrals and can cause vesicle-vesicle collisions when the viscosity contrast is high. Indeed, one can easily construct simulations with high viscosity contrast where our algorithms fail to resolve inter-vesicle interactions accurately. Quaipe and Biros [23] addressed this issue in 2D and we will extend their method to 3D in our future work.

1.3. Related Work

This work is an extension of [27] and we refer the reader to [27] for a review of the related work on three-dimensional simulation of vesicles. The work of Graham and Sloan [11] on singular quadratures for the scalar Helmholtz operator, the work of Zinchenko and Davis [36] on surface reparametrization schemes for drops and deformable particles, and the work of Zhao et al. [33] on simulations of red blood cells¹ have influenced the work presented in this paper.

In spite of the large body of literature devoted to the investigation of rheological properties of red blood cell and vesicles suspensions, to the best of our knowledge, the work on numerical methods for vesicle flows with viscosity contrast is rather limited. Pozrikidis [20] reviews the work on boundary integral formulations for particulate flows prior to the year 2000. Freund [9] reviews numerical simulation methods for the flow of red blood cells and challenges therein. Bagchi and Kalluri [2, 3] consider the flow of capsules with viscosity contrast in shear flow based on a front tracking method. The capsules lack any resistance to bending. Biben and Misbah [5] outline an advected-field method for the vesicle, but they use a radial based representation, which limits the applicability of their method to vesicles with very high reduced volume. Pozrikidis [22] present a boundary integral formulation and explicit time stepping scheme, but the method suffers from large grid distortion for low and moderate viscosity contrasts. Zhao and Shaqfeh [34] look at the dynamics of a single vesicle. They represent the vesicle’s radius in spherical harmonic basis, hence restricting the shape to radially single-valued cases. This representation, although limiting for the general case, enables them to perform linear stability analysis for the dynamics of a single vesicle. Moreover, Zhao and Shaqfeh [34] treat the bending force explicitly and solve for tension such that the surface velocity is divergence free. Zhao and Shaqfeh [35] look at the dynamics of vesicle suspension in shear flow. In this work, vesicle’s surface is represented using piece-wise linear triangular mesh coupled with prediction/correction time stepping to update the position and tension and followed by surface reparametrization (first order spatial and temporal accuracy). Farutin et al. [8] present boundary integral formulation with explicit time stepping and surface triangulation (second order in space and first order time stepping).

To our knowledge, the majority of particulate flow solvers in three dimensions use explicit schemes with the exception of the work of Dimitrakopoulos [7], in which a Jacobian-free, finite-difference based Newton method was used. That paper considers the flow of a single droplet (no surface inextensibility constraint) in which the bending force is proportional to the mean curvature H . To our knowledge, there is no prior work on implicit schemes for locally inextensible vesicles with viscosity contrast in three dimensions.

1.4. Nomenclature

Throughout this paper, lowercase letters refer to (infinite dimensional) scalar or vector fields. Finite dimensional values are denoted by sans-serif boldface letters and, when necessary, the spherical harmonics’ truncation order is used as a superscript. Convolution operators are referred to by uppercase calligraphic script, e.g. \mathcal{S} , and the discrete version of a convolution operator is referred to by the same letter in uppercase sans-serif bold, e.g. \mathbf{S} . As a general rule, uppercase sans-serif bold letters denote finite dimensional linear operators. The truncation order of the spherical harmonics expansion is generally denoted by p , and variables in spectral space are distinguished by a hat symbol “ $\hat{\cdot}$ ”. We let N denote the number of vesicles. The interface between the i^{th} vesicle and the surrounding fluid is denoted by γ_i ($i = 1, \dots, N$) and the union of all these interfaces is defined as $\gamma := \cup_i \gamma_i$. The outward normal vector to the membrane of vesicles (pointing toward the suspending fluid) is denoted by n . The jump of a variable across an interface is denoted by $[[\cdot]]$. The surface Laplacian, divergence, and gradient operators are formally denoted by Δ_γ , div_γ , and grad_γ , respectively. In Table 1 we list symbols we use frequently in this paper.

¹presenting a spherical harmonic discretization of membranes with bending and shear resistance and explicit time discretization

Symbol	Definition	Symbol	Definition
$\ \cdot\ _{l^2}^p$	l^2 norm on a p -grid (Eq. (20))	K	Gaussian curvature
Δ_γ	Surface Laplacian	N	Total number of vesicles
α_i	$(1 + \lambda_i)/2$	\mathbb{S}^2	The unit sphere
γ_i	The boundary of the i^{th} vesicle	\mathcal{S}_i	The single-layer Stokes operator over i^{th} surface (Eq. (4))
γ	$\cup_i \gamma_i$	W	Area element
λ_i	The viscosity contrast, μ_i/μ	Y_{nm}	Spherical harmonic of degree n and order m (Eq. (8))
μ	Viscosity of the ambient fluid	div_γ	Surface divergence (Eq. (26))
μ_i	Viscosity of the fluid inside i^{th} vesicle	f_σ	Tensile force (Eq. (2b))
σ	Tension	f_b	Bending force (Eq. (2a))
χ	Shear rate	grad_γ	Surface gradient (Eq. (25))
ω_i	The domain enclosed by γ_i	n	Outward normal vector
ω	$\cup_i \omega_i$	p	Order of truncated spherical harmonics expansion
\mathcal{D}_i	The double-layer Stokes operator over i^{th} surface (Eq. (5))	u	Velocity
$E_n(\cdot)$	Modular l^2 norm (Eq. (18))	u^∞	The background velocity field
H	Mean curvature	x	The parametrization of the surface

Table 1: INDEX OF FREQUENTLY USED SYMBOLS AND OPERATORS.

Outline of the paper. In Section 2, we present the overall integral equation formulation. In Section 3, the spatial and temporal discretization schemes are presented respectively. In the latter part of that section we present the result of our numerical experiments. The paper is concluded with a summary of the overall algorithm and computational complexity of different steps in Section 4.

2. Mathematical Formulation

In this section we formally express the problem statement and give its boundary integral formulation. The detailed derivation of this formulation is given in [19]. The schematic of a typical domain is shown in Fig. 1.

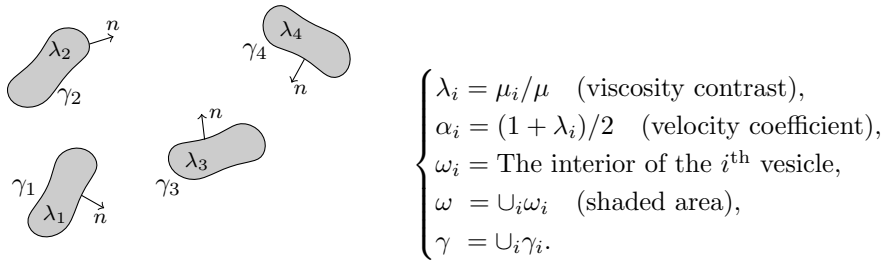


Figure 1: THE SCHEMATIC OF THE DOMAIN.

2.1. Differential formulation

In the length scale of vesicles, the Reynolds number is very small and the effect of convective terms in the Navier-Stokes equation is negligible. In the limit, the fluid dynamics are governed by the Stokes and

continuity equations

$$-\mu\Delta u(x) + \nabla p(x) = 0 \quad \text{and} \quad \text{div} u(x) = 0 \quad \text{for all } x \in \mathbb{R}^3 \setminus \omega, \quad (1a)$$

where $\mathbb{R}^3 \setminus \omega$ is the exterior region occupied by the suspending fluid, μ is the viscosity of the suspending fluid, $u(x)$ denotes the fluid velocity, and $p(x)$ denotes the pressure. Letting μ_i denote the viscosity of the fluid inside the i^{th} vesicle, Eq. (1a) holds for the inside fluid, $x \in \omega_i$, by replacing μ with μ_i . We supplement Eq. (1a) with the no-slip boundary condition on the interface of vesicles and matching far-field velocity as

$$\frac{\partial x}{\partial t} = u(x) \quad \text{for all } x \in \gamma, \quad u(x) \rightarrow u^\infty(x) \quad \text{as } \|x\| \rightarrow \infty, \quad (1b)$$

where u^∞ is the imposed far field velocity field. Moreover, since the surface of the vesicles is locally inextensible, the surface divergence of the velocity field should vanish [27]. Therefore,

$$\text{div}_\gamma u(x) = 0 \quad \text{for all } x \in \gamma. \quad (1c)$$

The balance of momentum on the membrane of vesicles implies that the jump in the surface traction is equal to the total force exerted by the interface onto the fluid, i.e.

$$\llbracket Tn \rrbracket = f(x) \quad \text{for all } x \in \gamma, \quad (1d)$$

where $T = -pI + \mu(\nabla u + \nabla u^T)$ is the Cauchy stress tensor, n denotes the normal vector to the surface at point x , $\llbracket \cdot \rrbracket$ denotes the jump across the interface, and f is the force exerted by the membrane onto the surrounding fluid. The interfacial force is composed of bending f_b and tensile f_σ forces. The source of tensile force is the local inextensibility of the surface of vesicle.

Eq. Set (1) defines a complete set of (nonlinear) equations that is solvable for the velocity field, tension, and the evolution of vesicle interfaces.

Interfacial forces. Membrane's resistance to bending and extension give rise to interfacial forces, the derivation of which can be found in [21, 28]. For any $x \in \gamma_i$ ($i = 1, \dots, N$) we define

$$f_b(x) = -\kappa_b [\Delta_{\gamma_i} H + 2H(H^2 - K)] n, \quad (2a)$$

$$f_\sigma(x, \sigma) = \sigma \Delta_{\gamma_i} x + \text{grad}_{\gamma_i} \sigma, \quad (2b)$$

where κ_b is the membrane's bending modulus, H and K are respectively mean and Gaussian curvatures at x , and σ is the tension. Note that the bending force f_b depends only on the current configuration of each vesicle and given the configuration, it can be evaluated directly. On the other hand, the tensile force f_σ depends on both the configuration and the tension σ , which is the Lagrange multiplier to enforce local inextensibility constraint, Eq. (1c). Therefore, at any given configuration and background velocity, Eq. (1c) needs to be solved to find tension σ .

2.2. Boundary integral formulation

One can follow the standard approach of potential theory [19, 20] to reformulate Eq. Set (1) as an integro-differential equation on the membrane of vesicles. It follows that for all $x \in \gamma_i$ ($i = 1, \dots, N$) we have

$$u(x) = \frac{1}{\alpha_i} \left(u^\infty(x) + \sum_{j=1}^N \mathcal{S}_j[f_b + f_\sigma](x) + \mathcal{D}_j[u](x) \right), \quad (3a)$$

$$\text{div}_{\gamma_i} u(x) = 0, \quad (3b)$$

$$\frac{\partial x}{\partial t} = u(x). \quad (3c)$$

where $\alpha_i = (1 + \lambda_i)/2$, $\mathcal{S}_j[f_b + f_\sigma](x)$ and $\mathcal{D}_j[u](x)$ denote the single-layer and double-layer convolution integrals over the j^{th} surface with the interfacial force and velocity as respective densities, evaluated at point x . The single-layer Stokes integral over the i^{th} surface, evaluated at point x is defined as

$$\mathcal{S}_i[f](x) := \int_{\gamma_i} S(x, y) f(y) d\gamma(y), \quad S(x, y) = \frac{1}{8\pi\mu} \frac{1}{\|r\|} \left(I + \frac{r \otimes r}{\|r\|^2} \right), \quad (4)$$

where $r := x - y$, I is the identity operator, \otimes denotes the tensor product, and $\|\cdot\|$ is the Euclidean norm. The free-space double-layer integral over the i^{th} surface, evaluated at point x is defined as

$$\mathcal{D}_i[u](x) := \int_{\gamma_i} D_i(x, y) u(y) d\gamma(y), \quad D_i(x, y) = -\frac{3(1 - \lambda_i)}{4\pi} \frac{(r \cdot n)(r \otimes r)}{\|r\|^5}. \quad (5)$$

The subscript i for the double-layer kernel D_i is to emphasize its dependence on the normal to surface $n(y)$ and the viscosity contrast of the i^{th} vesicle λ_i . Given the initial distribution of vesicles, the Eq. Set (3) may be used to solve for their evolution over time.

2.3. Galerkin formulation

Using the spherical harmonic functions Y_{nm} (defined in Eq. (8)) as the basis set for $L^2(\mathbb{S}^2)$, one can represent the position and tension in this basis set

$$x = \sum_{n=0}^{\infty} \sum_{m=-n}^n \hat{x}_{nm} Y_{nm} \quad \text{and} \quad \sigma = \sum_{n=0}^{\infty} \sum_{m=-n}^n \hat{\sigma}_{nm} Y_{nm}. \quad (6)$$

Letting (\cdot, \cdot) denote the inner product in this space—in which vector fields are treated element-wise—the Galerkin method seeks the solution to Eq. Set (3) by

$$\alpha_i(u, Y_{nm}) = (u^\infty, Y_{nm}) + \sum_{j=1}^N (\mathcal{S}_j[f_b + f_\sigma], Y_{nm}) + (\mathcal{D}_j[u], Y_{nm}), \quad (7a)$$

$$(\text{div}_{\gamma_i} u, Y_{nm}) = 0, \quad \text{for all } i = 1, \dots, N, \quad (7b)$$

$$\left(\frac{\partial x}{\partial t}, Y_{nm} \right) = (u, Y_{nm}) \quad (7c)$$

for all $n = 1, 2, \dots$ and $|m| \leq n$. The numerical solution of this set of equations involves:

- Accurate computation of interfacial forces f_b and f_σ , which in turn requires the accurate computation of high-order derivatives on the surfaces of vesicles.
- Fast and accurate quadrature over the sphere \mathbb{S}^2 .
- Numerical evaluation of the singular integrals for single- and double-layer Stokes such that the double integral—the inner product combined with the integral operators—in Eq. (7a) is numerically accurate.
- Fast evaluation of the the summation in Eq. (7a) to facilitate simulation of a large ensemble of vesicles.

In Section 3, we first outline our approach to perform computation over the surface of vesicles using the spherical harmonics. Afterwards, we outline a time stepping method to update the position of vesicles and then look at different schemes to solve the resulting linear system.

3. Numerical Algorithms

There are many components required to build the numerical machinery to solve the set of integro-differential equation given in the previous section. One needs a fast and accurate method to represent the surface of vesicles in space, perform differentiation on surfaces, and evaluate singular integral over the surfaces. Moreover, in order to evolve the vesicles in time, a method needs to be devised to avoid the stiffness in the evolution equation due to high order derivatives. In this section, we outline a Galerkin method in which we represent the surfaces in the spectral basis of spherical harmonics, perform singular integrals using the properties of this representation and evolve the vesicles using a first order semi-implicit time stepping scheme.

3.1. Spatial Representation

We assume that the boundary of each vesicle is parameterized by a smooth map from \mathbb{S}^2 to \mathbb{R}^3 . This assumption limits the application of this method to the smooth sphere-like (genus zero) surfaces. Nonetheless a large group of biological cells and particles, most importantly red blood cells, has genus zero. The advantage of this assumption is that it allows us to use the spherical harmonics expansions [6, 27].

3.1.1. Spherical harmonics basis. Using the spherical coordinates, a point on \mathbb{S}^2 is represented by $(\sin \theta \cos \phi, \sin \theta \sin \phi, \cos \theta)$, where $(\theta, \phi) \in U$ and $U := [0, \pi] \times [0, 2\pi)$. The spherical harmonic function of degree n ($n = 0, 1, 2, \dots$) and order m ($|m| \leq n$) evaluated at point (θ, ϕ) is defined by

$$Y_{nm}(\theta, \phi) = \frac{1}{\sqrt{2\pi}} P_{nm}(\cos \theta) e^{im\phi}, \quad (8)$$

where P_{nm} are the normalized associated Legendre polynomials of degree n and order m . General properties of the spherical harmonics can be found in [6].

3.1.2. Quadrature and discrete inner product over \mathbb{S}^2 . Given an integrable function f over the unit sphere, its integral can be written as

$$\int_{\mathbb{S}^2} f ds = \int_0^\pi \int_0^{2\pi} f(\theta, \phi) \sin \theta d\phi d\theta = \int_{-1}^1 \int_0^{2\pi} f(\cos^{-1} z, \phi) d\phi dz, \quad (9)$$

where we used the change of variable $z = \cos \theta$.

We use a tensor-product quadrature rule to numerically evaluate smooth integrals over the sphere. Given some truncation order p for the spherical harmonics expansion, one needs to numerically calculate the spherical harmonics coefficients for $n = 1, \dots, p$ and $|m| \leq n$. Requiring these coefficients to be evaluated exactly for p -bandlimited functions, our quadrature rule needs to be exact for polynomials of degree $2p$ in the latitude direction and trigonometric polynomials of degree $2p$ in the longitude direction. To this end, we use a $(p+1)$ -point Gauss-Legendre rule in the z direction, and $(2p+2)$ -point uniform trapezoidal rule in the ϕ direction. The Gauss-Legendre nodes $\{z_i\}_{i=0}^p$ are the zeros of the Legendre polynomial of degree $p+1$ and $\{\nu_i\}_{i=0}^p$ are the corresponding weights; we let $\theta_i := \cos^{-1} z_i$ ($i = 0, \dots, p$). The longitudinal nodes are uniform $\{\phi_j = \frac{\pi j}{p+1}\}_{j=0}^{2p+1}$ and the weights are $\{\mu_j = \frac{\pi}{p+1}\}_{j=0}^{2p+1}$. In the text we refer to this grid as a p -grid. We denote the discrete samples of the function f on a p -grid by $\mathbf{f}^p := \{f(\theta_i, \phi_j)\}_{i,j}$. It follows that

$$\int_{-1}^1 \int_0^{2\pi} f(\cos^{-1} z, \phi) d\phi dz \approx \sum_{i=0}^p \sum_{j=0}^{2p+1} \nu_i \mu_j f(\theta_i, \phi_j) =: \mathbf{q}^p \cdot \mathbf{f}^p, \quad (10)$$

where the dot denotes the inner product and the second equality is the definition of the quadrature operator. We let \mathbf{Q}^p denote the matrix with the entries of \mathbf{q}^p on its diagonal, i.e. $\mathbf{Q}^p = \text{diag}(\mathbf{q}^p)$.

We use the usual inner product in $L^2(\mathbb{S}^2)$ denoted by $(f, g) = \int_{\mathbb{S}^2} f \bar{g} ds$, where the overbar denotes the complex conjugate. Based on the integration scheme above, we define a discrete inner product $(\cdot, \cdot)^p$ over the sphere as

$$(f, g)^p := \bar{\mathbf{g}}^p \mathbf{Q}^p \mathbf{f}^p. \quad (11)$$

3.1.3. Spectral representations. Since the spherical harmonics are the eigenfunctions of the Laplace-Beltrami operator on the sphere, they form a basis for $L^2(\mathbb{S}^2)$ and the functions in $L^2(\mathbb{S}^2)$ can be approximated by spherical harmonics expansion, for instance, given a square-integrable function $f : \mathbb{S}^2 \rightarrow \mathbb{R}$ we can write

$$f(\theta, \phi) = \sum_{n=0}^{\infty} \sum_{m=-n}^n \hat{f}_{nm} Y_{nm}(\theta, \phi), \quad \text{with} \quad \hat{f}_{nm} = (f, Y_{nm}) = \int_{\mathbb{S}^2} f \bar{Y}_{nm} ds. \quad (12)$$

Moreover, when the function f is smooth on the sphere, $f \in C^\infty(\mathbb{S}^2)$, the sequence of partial sums converges spectrally to f [17]. In the remainder of this paper, we let p denote the degree of truncated spherical harmonic expansion, that is

$$f(\theta, \phi) \approx \sum_{n=0}^p \sum_{m=-n}^n \hat{f}_{nm} Y_{nm}(\theta, \phi). \quad (13)$$

Given a discrete p -grid and the sample of the function on this grid \mathbf{f}^p , the discrete inner product operator, defined in Eq. (11), can be used to estimate the spherical harmonic coefficients \hat{f}_{nm} , using Eq. (12). We define a discrete spherical harmonic projection $\mathbf{Y}_{\text{Proj}}^p$ such that

$$\hat{\mathbf{f}}^p = \mathbf{Y}_{\text{Proj}}^p \mathbf{f}^p := \bar{\mathbf{Y}}^p \mathbf{Q}^p \mathbf{f}^p, \quad (14)$$

where $\hat{\mathbf{f}}^p = \{\hat{f}_{nm}^p\}_{n,m}$ is the array of spherical harmonic coefficients, and \mathbf{Y}^p is the matrix with discrete value of spherical harmonics ($n \leq p, |m| \leq n$) in its columns. Now, we can simply define the inverse spherical harmonics transfer as the multiplication of the coefficient with the discrete basis set

$$\mathbf{f}^p = \mathbf{Y}^p \hat{\mathbf{f}}^p. \quad (15)$$

Due to our requirement on the quadrature rules to be exact for polynomials of degree less than or equal to $2p$ in the latitude and the longitude direction, we have

$$(Y_{nm}, Y_{n'm'})^p = \delta_{nn'} \delta_{mm'}, \quad \text{for } 0 \leq n, n' \leq p \text{ and } |m| < n, |m'| < n',$$

where δ_{ij} denotes the Kronecker delta. Note that in matrix notation, this translates to $\mathbf{Y}_{\text{Proj}}^p \mathbf{Y}^p = \mathbf{I}$. However, this discrete orthogonality does not always hold when n or n' is larger than p and there may be cases when $(Y_{nm}, Y_{n'm'})^p \neq 0$ for $n \leq p$ and $n' > p$ [6]. This phenomenon is due to the *aliasing effect*, which we will discuss in a later part of this section.

3.1.4. Representation of the surface. Using the spherical coordinates with domain $U = \{(\theta, \phi) : \theta \in [0, \pi], \phi \in [0, 2\pi)\}$, we define a parametrization of the surface $x : U \rightarrow \mathbb{R}^3$ such that $x(U) = \gamma$. Based on our general assumption that the surfaces are smooth and have genus zero, each Cartesian component of $x = (x_1, x_2, x_3)$ is a smooth function over the sphere and we can write the truncated spherical harmonic expansion of the vector field x as

$$x(\theta, \phi) \approx \sum_{n=0}^p \sum_{m=-n}^n \hat{x}_{nm} Y_{nm}(\theta, \phi), \quad (16)$$

where the spherical harmonic mapping is performed on each component of x , i.e. $\hat{x}_{nm} := ((\hat{x}_1)_{nm}, (\hat{x}_2)_{nm}, (\hat{x}_3)_{nm})$.

Integration over the surface follows from the change of variable given in Eq. (9) and the discrete integration formula, Eq. (10). Hence, for a scalar field $f(x)$ on the surface γ we have

$$\int_{\gamma} f(x) d\gamma(x) = \int_U f(\theta, \phi) W(\theta, \phi) \sin \theta d\theta d\phi \approx \mathbf{Q}^p \mathbf{W}^p \cdot \mathbf{f}^p, \quad (17)$$

where $W = \sqrt{EG - F^2}$ is the area element of the surface γ (with E , F , and G denoting the coefficients of the first fundamental form of γ) [27].

3.1.5. Norms. We define different measures of norm in the spherical harmonic domain (or equivalently on the unit sphere) as following:

$$\text{Modular } l^2: \quad E_n(f) := \left(\sum_{m=-n}^n |\hat{f}_{nm}|^2 \right)^{1/2}, \quad (18)$$

$$l^2: \quad \|f\|_{l^2} := \left(\sum_{n=0}^{\infty} \sum_{m=-n}^n |\hat{f}_{nm}|^2 \right)^{1/2} = \left(\sum_{n=0}^{\infty} E_n^2 \right)^{1/2} \quad (19)$$

$$\text{Discrete } l^2 \text{ on a } p\text{-grid:} \quad \|\mathbf{f}\|_{l^2}^p := \left(\sum_{n=0}^p \sum_{m=-n}^n |\hat{\mathbf{f}}_{nm}^p|^2 \right)^{1/2}. \quad (20)$$

Note that for the l^2 norm on the p -grid, not only the series is truncated but also the coefficients are calculated numerically on that grid using Eq. (14). For a vector field $u = (u_1, u_2, u_3)$ we define $\|u\|_{l^2} := (\|u_1\|_{l^2}^2 + \|u_2\|_{l^2}^2 + \|u_3\|_{l^2}^2)^{1/2}$.

When we compare the spectral coefficients, it is implied that one of the two series is properly truncated to match the other series. For example for p and q ($p \leq q$) we write

$$\|\mathbf{f}^p - \mathbf{f}^q\|_{l^2}^p = \left(\sum_{n=0}^p \sum_{m=-n}^n |\hat{\mathbf{f}}_{nm}^p - \hat{\mathbf{f}}_{nm}^q|^2 \right)^{1/2}. \quad (21)$$

The same convention holds when one of the series is not truncated ($q = \infty$)

$$\|\mathbf{f}^p - f\|_{l^2}^p = \left(\sum_{n=0}^p \sum_{m=-n}^n |\hat{\mathbf{f}}_{nm}^p - \hat{f}_{nm}|^2 \right)^{1/2}. \quad (22)$$

Interpolation via spherical harmonics. Given the discrete samples of the function \mathbf{f}^p on a p -grid, one can use Eq. (14) and Eq. (15) to interpolate f on the nodal points of another q -grid, where q may be larger or smaller than p . The interpolation is performed first by mapping to spherical harmonics domain, followed by zero-padding or truncation of the array of spherical harmonic coefficients $\hat{\mathbf{f}}^p$, and mapping back to the physical space:

$$\hat{\mathbf{f}}_{nm}^q = \begin{cases} \hat{\mathbf{f}}_{nm}^p & n \leq p \\ 0 & n > p \end{cases}, \quad \text{for } n = 0, \dots, q \text{ and } |m| \leq n. \quad (23a)$$

Formally, we denote this interpolation process by

$$\mathbf{f}^q = \mathbf{I}_p^q \mathbf{f}^p. \quad (23b)$$

3.1.6. Differentiation on \mathbb{S}^2 and γ . Differentiation of functions defined on the sphere is performed using their spherical harmonic expansion, for example given $f(\theta, \phi) = \sum_{n=0}^p \sum_{m=-n}^n \hat{f}_{nm} Y_{nm}(\theta, \phi)$, one can calculate

$$f_{\theta} = \sum_{n=0}^p \sum_{m=-n}^n \hat{f}_{nm} (Y_{nm})_{\theta} \quad \text{or} \quad f_{\phi} = \sum_{n=0}^p \sum_{m=-n}^n \hat{f}_{nm} (Y_{nm})_{\phi}. \quad (24)$$

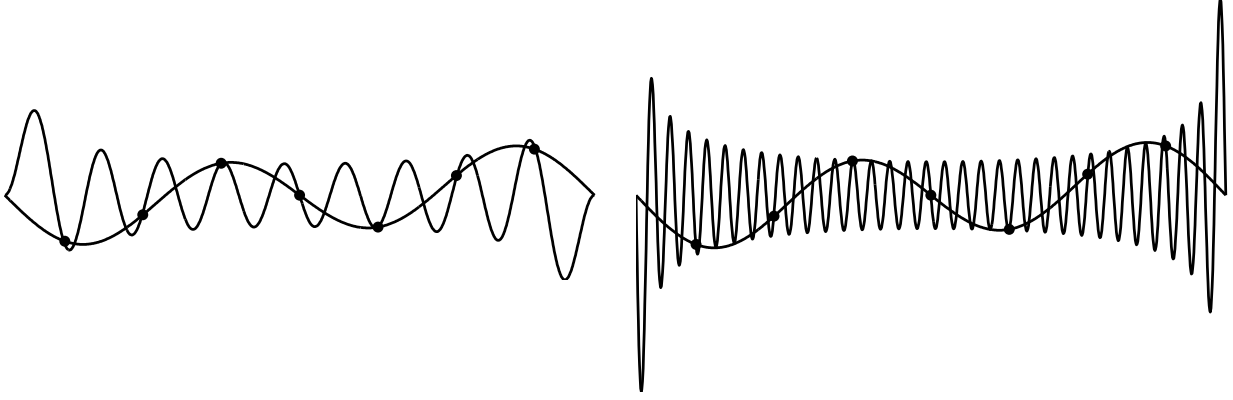


Figure 2: ALIASING FOR ASSOCIATED LEGENDRE POLYNOMIALS. *Aliasing effect for the associated Legendre polynomials on a 7-point Gauss-Legendre grid on $[-1, 1]$. On the sample points, denoted by a dot, the value of the high frequency function coincides with that of the low frequency function. In both of the plots, the less oscillatory functions is $P_{4,1}(z)$. In the left panel, $P_{19,-2}(z)$ is aliased to $P_{4,1}(z)$ and in the right panel $P_{64,1}(z)$ is aliased to $P_{4,1}(z)$.*

For a p -bandlimited function, this approach is exact and for a smooth function the only approximation is in the truncation error of the spherical harmonics expansion.

Given a scalar function $f(x)$ on the surface γ with parametrization $x : U \rightarrow \mathbb{R}^3$, one can calculate the surface gradient of f by

$$\text{grad}_\gamma f = \left(\frac{Gx_\theta - Fx_\phi}{W^2} \right) f_\theta + \left(\frac{Ex_\phi - Fx_\theta}{W^2} \right) f_\phi. \quad (25)$$

Similarly, the surface divergence of a vector field $u(x)$ is given by

$$\text{div}_\gamma u = \left(\frac{Gx_\theta - Fx_\phi}{W^2} \right) \cdot u_\theta + \left(\frac{Ex_\phi - Fx_\theta}{W^2} \right) \cdot u_\phi, \quad (26)$$

and the surface Laplacian is defined as $\Delta_\gamma f = \text{div}_\gamma \text{grad}_\gamma f$.

3.1.7. Aliasing and its remedy. Given a discrete grid on any domain, samples of certain distinct functions are indistinguishable on that grid (a phenomenon referred to as aliasing). An example of aliasing for the associated Legendre polynomials on the Gauss-Legendre grid is shown in Fig. 2.

For spherical harmonics, one high frequency basis function may alias to many different lower frequency basis on a p -grid. Suppose f is a smooth function over the sphere with the spherical harmonic coefficients \hat{f}_{nm} . Given the samples of f over a p -grid, one can use the discrete inner product $(\cdot, \cdot)^p$ to compute the spherical harmonic coefficients $\hat{\mathbf{f}}_{n'm'}^p$ ($n' = 0, \dots, p$ and $|m'| < n'$)

$$\hat{\mathbf{f}}_{n'm'}^p = (f, Y_{n'm'})^p = \sum_{n=0}^{\infty} \sum_{m=-n}^n \hat{f}_{nm} (Y_{nm}, Y_{n'm'})^p = \hat{f}_{n'm'} + \sum_{n>p} \sum_{m=-n}^n e_{n'm'}(n, m) \hat{f}_{nm}. \quad (27)$$

Note that the second sum is taken over $n > p$ and corresponds to functions not fully captured on the p -grid. The coefficient $e_{n'm'} = (Y_{nm}, Y_{n'm'})^p$ is the amount of aliasing from high frequency Y_{nm} and $|e_{n'm'}| < 1$, by Cauchy-Schwarz inequality. Eq. (27) gives us a crude upper bound on the estimation error as $|\hat{\mathbf{f}}_{n'm'}^p - \hat{f}_{n'm'}| \leq \sum_{n>p} \sum_{m=-n}^n |\hat{f}_{nm}|$ that goes to zero as $p \rightarrow \infty$ due to smoothness of f . Therefore, the smoothness of the function f (the rate of decay of \hat{f}_{nm}) and the grid resolution p control the estimation error.

In a spectral setting, aliasing is the source of pollution to lower frequencies and will cause the simulation to diverge. To control the aliasing, when computing the spectral coefficients of a function f on a p -grid, $\hat{\mathbf{f}}^p$, for a given tolerance ϵ , we require that

$$\frac{\|\mathbf{f}^p - f\|_{l_2}^p}{\|f\|_{l_2}^p} \leq \epsilon. \quad (28)$$

Since in a numerical setting, the exact value of the function is not known, we use the values that are computed on a fine-resolution q -grid as a surrogate for the exact values. Based on our experiments, having $q > 2p$ gives a good estimate for the error. The choice of ϵ depends on the desired accuracy in differentiation. In the context of time stepping methods, we chose this error to be the same order as the local truncation error of the time stepping. For instance, for the backward Euler method, we choose $\epsilon \leq \delta t^2$, where δt is the time step.

As we will see in the next section and was mentioned in [27], nonlinear manipulation of functions in the physical space is a source of aliasing. In our work, calculation of the mean and Gaussian curvatures manifest the aliasing problem the most. To control the aliasing, we upsample the functions to a finer grid q , perform differentiation and then downsample the result back to p . We use a simple algorithm outlined in Section 3.1.8 to compute the required upsampling frequency q . Letting \mathbf{D}^q denote a differentiation matrix on a q -grid, and using the interpolation operator given in Eq. (23b), we define the discrete differentiation with dealiasing as

$$\mathbf{D}_p^q \mathbf{f}^p := (\mathbf{I}_q^p \mathbf{D}^q \mathbf{I}_p^q) \mathbf{f}^p \quad (29)$$

Note that we adopt this method to keep the representation frequency p as small as possible without losing the accuracy in differentiation. As we will see, the computation complexity of Stokes singular integral is $\mathcal{O}(p^5)$ that gives us enough incentive to try to keep p as small as possible.

3.1.8. Calculating the Upsampling Rate to Control Aliasing. Nonlinear manipulations of functions in the physical space are the main sources of aliasing. Given a band-limited function over the sphere, its derivative may not be band-limited in the spherical harmonic basis and the spectral coefficients may decay much slower. The best example for this, is the mean curvature:

$$H = \frac{EN - 2FM + GL}{2W^2},$$

where E, F , and G are first fundamental form coefficients, L, M , and N are the second fundamental form coefficients, and W is the area element. Given p -grid samples of a surface x , each term in the expression for mean curvature can be calculated very accurately. Nonetheless, as it is shown in Fig. 3, the mean curvature usually has more oscillatory components compared to the position x and its spectrum decays much slower than x .

Given the discrete samples of the surface x on a p -grid, to avoid the aliasing, one may need to perform the calculation of the mean curvature (or any function that involves differentiation and nonlinear manipulation) on a finer grid q and then restrict the result to the coarse p -grid through Eq. (23b). By doing this, one controls the magnitude of the error term in Eq. (27) to satisfy Eq. (28). Due to the highly nonlinear nature of the functions involved in our calculation, the widely used “3/2 upsampling” rate proposed by Orszag [17, 18] for equations with quadratically nonlinear terms does not extend to our case. Moreover, when calculating the curvatures on a p -grid, the required resolution of the fine grid q (or the upsampling rate $v = q/p$) depends on the rate of decay of H , which is not known *a priori* and depends on the shape. As an example, in Table 2 we give the upsampling rate v required to calculate $\hat{\mathbf{H}}^8$ with different values of tolerance, according to Eq. (28). In Algorithm 1 we give the pseudo code to compute the required upsampling rate v to meet a tolerance ϵ . We calculate the required upsampling rate for the mean curvature and use this rate for all differentiations on

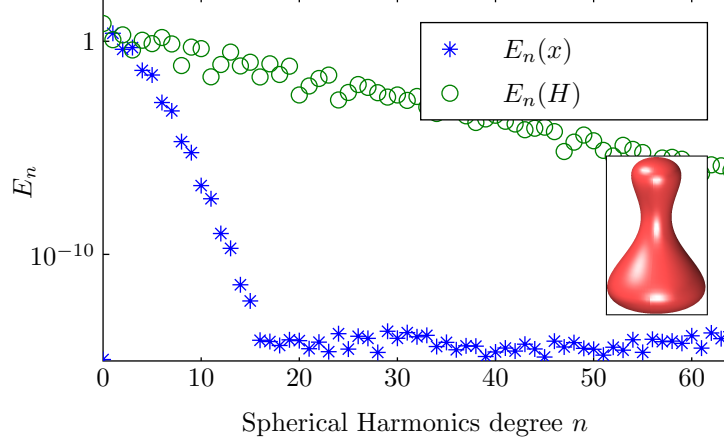


Figure 3: THE SPECTRUM OF THE MEAN CURVATURE. The modular l^2 norm of the position vector x and that of the mean curvature H for the shape plotted in the inset. The abscissa is the spherical harmonics' degree n and the ordinate is the modular l^2 energy defined in Eq. (18). The position vector is band-limited with bandwidth of 16. The bandwidth of H on the other hand, is much wider than that of the position and its spectral coefficients decay much slower. For the data in this figure, the calculation is performed on a 64-grid. On a 16-grid for instance, the shape is represented very accurately. Nonetheless, if the discrete data on this grid are used to evaluate H , through aliasing, all the high frequency components (here $n > 16$) pollute the low frequency coefficients and lead to inaccurate evaluation of \widehat{H}_{nm}^{16} .

Tolerance (ϵ)	upsampling rate				
	1	2	3	4	5
$1e-1$	2	2	1	2	2
$1e-5$	3	4	4	4	4
$1e-9$	4	6	7	7	7

Table 2: THE UPSAMPLING RATE FOR DIFFERENTIATION. The required upsampling rate v in the calculation of the spectral coefficients of the mean curvature \mathbf{H}^8 such that $\|\mathbf{H}^{vp} - H\|_{l^2}^p \leq \epsilon \|H\|_{l^2}^p$. We used a fine resolution $q = 80$ as the surrogate for the exact values. (Surfaces 1-4 are examples used in [27, Fig. 2]).

the surface. During a simulation, we update the upsampling rate periodically (adjusting upsampling every nondimensional time unit, i.e. every $1/\delta t$ steps, works well for our examples) as the shape changes over time.

In Fig. 4 we show the evolution of a single vesicle in Poiseuille flow. Without proper handling of aliasing, low resolution simulation of this type of flow is challenging because of the shape it induces on the vesicle. Nonetheless, by treating the aliasing, we can capture the deformation of a vesicle in this flow by using only $p = 8$. For this simulation, $\delta t = 5e-2$ and the final error in area and volume was $e_a = 4.24e-2$, $e_v = 5.96e-3$.

Another important method to control the aliasing and the grid quality is the reparametrization algorithm we proposed in [27]. We outline the reparametrization algorithm in Appendix A.

3.1.9. Singular integrals over the surface γ . In our previous paper [27], we extensively explained the quadrature rule to evaluate singular integrals over the sphere, which was adopted from the work of Graham and

Algorithm 1 UPSAMPLING RATE CALCULATION.

Require: x, ϵ, p, p_{\max}
 $q \leftarrow p$ and $e \leftarrow 2\epsilon$

 Calculate \mathbf{H}^p using \mathbf{D}^p as differentiation matrix

while $e > \epsilon$ and $q \leq p_{\max}$ **do**
 $q \leftarrow q + p$

 Calculate $\mathbf{H}^{p,q}$ using \mathbf{D}_p^q as differentiation matrix

// Defined in Eq. (29)

 $e \leftarrow \frac{\|\mathbf{H}^p - \mathbf{H}^{p,q}\|_{l_2}^p}{\|\mathbf{H}^{p,q}\|_{l_2}^p}$

// Defined in Eq. (28)

 $\mathbf{H}^p \leftarrow \mathbf{H}^{p,q}$
end while
return $v \leftarrow q/p$

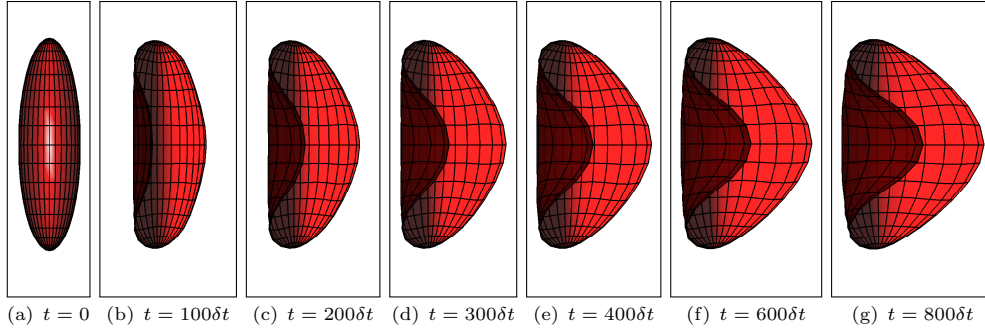


Figure 4: CROSS SECTIONS OF A BICONCAVE VESICLE IN THE POISEUILLE FLOW. *The cross sectional plots showing the evolution of a single vesicle with an initial biconcave shape in a parabolic flow. The time stepping method is implicit, the reduced volume is 0.65, and the time step δt is $5e-2$. For this simulation, $p = 8$ and when $t \leq 300\delta t$ the upsampling rate is 4 and when $t > 300\delta t$ the upsampling rate is 5.*

Sloan [11]. On a surface γ , the double-layer kernel is defined as

$$D(x, y) = -\frac{3}{4\pi} \frac{1}{\|r\|} \left(\frac{r \cdot n}{\|r\|^2} \frac{r \otimes r}{\|r\|^2} \right), \quad \text{with } r = x - y.$$


The first term in the parenthesis satisfies the smoothness requirements outlined in [11]. The second term is similar to the Stokes kernel. In [27], we argued that since $(r \otimes r)/\|r\|^2$ is reflectionally symmetric² and smooth over the parameter space of the sphere $U = \{(\theta, \phi) : \theta \in [0, \pi], \phi \in [0, 2\pi]\}$, the method of [Graham and Sloan](#) can be extended to the Stokes integral. By the same argument, the double-layer integral can be accurately evaluated using the singular quadratures given in [27]. On the surface γ with a p -grid discretization $\mathbf{x}^p = \{x_{ij}\}$ and density $\mathbf{f}^p = \{f_{ij}\}$, in which $x_{ij} = x(\theta_i, \phi_j)$ and $f_{ij} = f(\theta_i, \phi_j)$, we denote the discrete singular single- and double-layer operators (for all target point \mathbf{x}^p) by $\mathbf{S}^p \mathbf{f}^p$ and $\mathbf{D}^p \mathbf{f}^p$, such that

$$\mathcal{S}[f](x_{ij}) = \int_{\gamma} S(x_{ij}, y) f(y) d\gamma \approx (\mathbf{S}^p \mathbf{f}^p)_{ij}, \quad (30a)$$

$$\mathcal{D}[f](x_{ij}) = \int_{\gamma} D(x_{ij}, y) f(y) d\gamma \approx (\mathbf{D}^p \mathbf{f}^p)_{ij}, \quad (30b)$$

for all $0 \leq i \leq p$ and $0 \leq j < 2p + 2$. The general algorithm to evaluate the double-layer integral is the same as the one for the Stokes integral and we refer to [27] for the details. Here, we report convergence results for

²A function f is said to be reflectionally symmetric when $f(\theta, \phi) = f(-\theta, \phi + \pi)$ for all $\theta, \phi \in \mathbb{R}$ [11].



p	Relative error			
8	$6.29e-3$	$2.53e-1$	$8.11e-2$	$9.43e-2$
12	$6.39e-4$	$6.06e-2$	$2.61e-2$	$2.59e-2$
16	$6.56e-5$	$1.49e-2$	$9.91e-3$	$5.38e-3$
24	$6.96e-7$	$3.02e-3$	$9.28e-4$	$4.30e-4$
32	$7.19e-9$	$3.38e-4$	$9.83e-5$	$4.64e-5$

Table 3: CONVERGENCE OF DISCRETE DOUBLE-LAYER INTEGRAL. The relative error $\|\mathbf{D}^p \mathbf{f}^p - \mathbf{D}^q \mathbf{f}^q\|_{l_2}^p / \|\mathbf{D}^q \mathbf{f}^q\|_{l_2}^p$ in the computation of the double-layer integral over the depicted surfaces for different spherical harmonic order p . The discrete double-layer operator is defined in Eq. (30b) and the discrete norm is defined in Eq. (20). The error is calculated with respect to a fine discretization with spherical harmonics order of $q = 64$. On each surface, the density is chosen to be the same as the position vector ($f = x$).

the numerical integration of the double-layer integral over different shapes in Table 3.

In a Galerkin setup, numerical evaluation of a double integral is involved in the Eq. Set (7), namely the singular integrals followed by the projection to the basis

$$(\mathcal{S}[f], Y_{nm}) \approx (\mathbf{S}^q \mathbf{f}^q, Y_{nm})^p \quad \text{and} \quad (\mathcal{D}[u], Y_{nm}) \approx (\mathbf{D}^q \mathbf{u}^q, Y_{nm})^p. \quad (31)$$

The result presented in [11, specifically Theorem 4.1] requires that $q > p(1 + a)$ for some $a > 0$. However, it does not give a formula to estimate a . In our numerical experiments and convergence analysis, we noticed that when we solve for position and tension with $q = p$, the coefficients with highest order, i.e. $\hat{\mathbf{x}}_{pm}$ and $\hat{\boldsymbol{\sigma}}_{pm}$ for $|m| \leq p$, are calculated with large error. Nonetheless, this was fixed by choosing $q = p + 1$. Combining all these computational steps — i.e. upsampling to q ,³ performing singular integral, and mapping back to p — together, we can write

$$\widehat{\mathbf{S}}^p \widehat{\mathbf{f}}^p := (\mathbf{I}_q^p \mathbf{Y}_{\text{Proj}}^q \mathbf{S}^q \mathbf{Y}_p^q \mathbf{I}_p^q) \widehat{\mathbf{f}}^p, \quad (32)$$

for the single-layer Stokes operator in spherical harmonic domain. The same can also be written for the double-layer Stokes, which we denote by $\widehat{\mathbf{D}}^p$.

3.1.10. Nearly-singular integrals. Nearly-singular integrals arise when a kernel has a singularity on the boundary of a domain, and we need to evaluate the integral with this kernel at a point x close to the surface, so that the distance between quadrature point does not allow accurate evaluation. In the context of our work, let h denote the discretization spacing for p -grid samples of a given surface γ_i :

$$h := \max_{\mathbf{x} \in \mathbf{x}^p} \min_{\substack{\mathbf{y} \in \mathbf{x}^p \\ \mathbf{y} \neq \mathbf{x}}} |\mathbf{x} - \mathbf{y}|.$$

Now, lets consider the single-layer integral for a target point $x \in \mathbb{R}^3$ with density f on surface γ_i , $\mathcal{S}[f](x) = \frac{1}{8\pi\mu} \int_{\gamma_i} S(x, y) f(y) d\gamma(y)$. When $x \in \gamma_i$ the integral is singular and can be numerically integrated using the algorithm mentioned in Section 3.1.9; when $x \notin \gamma_i$ then $d = \text{dist}(x, \gamma_i) > 0$ and the integral is not singular anymore. Nevertheless, as $d \rightarrow 0$ the derivative of $S(x, y)$ with respect to y cannot be bounded

³In practice, the computationally efficient way to do this is to perform the computation on a p -grid and filter the highest frequency coefficients.

uniformly [31] and $S(x, y)$ becomes singular and oscillatory. In this work, nearly-singular integrals arise when computing the interaction between vesicles. When computing the interaction, we try to avoid the occurrence of nearly-singular integrals by uniformly upsampling the grid points to a finer frequency $q \approx p^{3/2}$ for integration. The choice of this upsampling frequency is based on the argument given on [31] in order to permit $d \approx h$. There are situations where the vesicles get too close and our method breaks down (an example of this is when the viscosity contrast of vesicles is large). This can be resolved by the nearly-singular algorithm proposed by Ying et al. [31], which we are currently working to incorporate into our framework.

3.1.11. Linear interfacial forces. The interfacial forces are given in Eq. Set (2). Following our approach in [27], given the configuration of the vesicle γ (defined by x), a new configuration defined by x^+ , and a new tension σ^+ we define the linear versions (with respect to x^+ or σ^+) of mean curvature, bending force, and tensile force

$$H(x^+; x) = \frac{1}{2W^2} \left(Ex_{\phi\phi}^+ - 2Fx_{\theta\phi}^+ + Gx_{\theta\theta}^+ \right) \cdot n, \quad (33a)$$

$$f_b(x^+; x) = -\kappa_b \left[\Delta_\gamma H^+ + 2H^+(H^2 - K) \right] n, \quad (33b)$$

$$f_\sigma(\sigma^+; x) = \sigma^+ \Delta_\gamma x + \text{grad}_\gamma \sigma^+, \quad (33c)$$

where the $H^+ = H(x^+; x)$ and the terms with no superscript are evaluated over x . The discrete spectral versions of these operator can be written as $\widehat{\mathbf{f}}_b = \mathbf{Y}_{\text{Proj}} \mathbf{f}_b$ and $\widehat{\mathbf{f}}_\sigma = \mathbf{Y}_{\text{Proj}} \mathbf{f}_\sigma$. Combining these expressions with the spectral stokes operator Eq. (32), we define two new linear operators $\widehat{\mathbf{B}}$ and $\widehat{\mathbf{T}}$ such that

$$\widehat{\mathbf{B}} \widehat{\mathbf{x}}^+ = \widehat{\mathbf{S}} \widehat{\mathbf{f}}_b, \quad (34)$$

$$\widehat{\mathbf{T}} \widehat{\sigma}^+ = \widehat{\mathbf{S}} \widehat{\mathbf{f}}_\sigma. \quad (35)$$

Moreover, we define the discrete, spectral surface divergence operator as

$$\widehat{\mathbf{P}} \widehat{\mathbf{u}} = \mathbf{Y}_{\text{Proj}} \text{div}_\gamma (\mathbf{Y} \widehat{\mathbf{u}}). \quad (36)$$

When we have multiple vesicles, our notation for the bending and tension operators needs to be more specific. We use the double subscript $\widehat{\mathbf{B}}_{ij}, \widehat{\mathbf{T}}_{ij}, \widehat{\mathbf{D}}_{ij}$ ($i, j = 1, \dots, N$) that implies the integral is performed on the j^{th} surface and the target points are the grid points of the i^{th} surface. For the surface divergence, we append the single subscript $\widehat{\mathbf{P}}_i$ to denote that the divergence is taken on the i^{th} surface. For the bending, tensile, and double-layer operators, when $i = j$ the integral is evaluated using the singular quadratures and when $i \neq j$ the integral is a smooth integral (as long as two vesicles do not get too close to each other, Section 3.1.10).

3.2. Galerkin Formulation

Using the machinery developed in Section 3.1, we use a Galerkin formulation to solve for the evolution of vesicles in the spectral domain. In [27], we used a pseudo-spectral method that has the same spatial accuracy as the Galerkin method. Nevertheless, because of the features of spherical harmonics, one needs twice as many variables in the pseudo-spectral method (the grid points) as in Galerkin method (spherical harmonic coefficients). For this reason, we opted for the Galerkin method.

3.2.1. System of ODEs for a suspension of N vesicles. Substituting the discrete operators defined in Section 3.1 in the Eq. Set (7), we get a system of ODEs for the discrete tension and position of the i^{th} vesicle

($i = 1, \dots, N$):

$$\alpha_i \widehat{\mathbf{u}}_i = \widehat{\mathbf{u}}^\infty(\mathbf{x}_i) + \sum_{j=1}^N \widehat{\mathbf{B}}_{ij} \widehat{\mathbf{x}}_j + \widehat{\mathbf{T}}_{ij} \widehat{\boldsymbol{\sigma}}_j + \widehat{\mathbf{D}}_{ij} \widehat{\mathbf{u}}_j, \quad (37a)$$

$$\widehat{\mathbf{P}}_i \widehat{\mathbf{u}}_i = 0, \quad (37b)$$

$$\frac{d\widehat{\mathbf{x}}_i}{dt} = \widehat{\mathbf{u}}_i. \quad (37c)$$

3.3. Time Discretization

We use the backward Euler method to approximate the time derivative of position, where we write $d\mathbf{x}/dt \approx (\mathbf{x}^+ - \mathbf{x})/\delta t$ or equivalently

$$\frac{d\widehat{\mathbf{x}}_i}{dt} \approx \frac{\widehat{\mathbf{x}}_i^+ - \widehat{\mathbf{x}}_i}{\delta t}. \quad (38)$$

Substituting this in Eq. (37c), using the result in Eq. (37a) and substituting in Eq. (37b), we get a system of equations for the new position and tension

$$\alpha_i \widehat{\mathbf{x}}_i^+ - \sum_{j=1}^N (\delta t \widehat{\mathbf{B}}_{ij} + \widehat{\mathbf{D}}_{ij}) \widehat{\mathbf{x}}_j^+ - \delta t \sum_{j=1}^N \widehat{\mathbf{T}}_{ij} \widehat{\boldsymbol{\sigma}}_j^+ = \mathbf{r}_i, \quad (39a)$$

$$- \sum_{j=1}^N (\delta t \widehat{\mathbf{P}}_i \widehat{\mathbf{B}}_{ij} + \widehat{\mathbf{P}}_i \widehat{\mathbf{D}}_{ij}) \widehat{\mathbf{x}}_j^+ - \delta t \sum_{j=1}^N \widehat{\mathbf{L}}_{ij} \widehat{\boldsymbol{\sigma}}_j^+ = \mathbf{s}_i, \quad (39b)$$

where $\mathbf{r}_i := \alpha_i \widehat{\mathbf{x}}_i + \delta t \widehat{\mathbf{u}}^\infty - \sum_{j=1}^N \widehat{\mathbf{D}}_{ij} \widehat{\mathbf{x}}_j$ and $\mathbf{s}_i := \delta t \widehat{\mathbf{P}}_i \widehat{\mathbf{u}}^\infty - \sum_{j=1}^N \widehat{\mathbf{P}}_i \widehat{\mathbf{D}}_{ij} \widehat{\mathbf{x}}_j$ for all $i = 1, \dots, N$ and $\widehat{\mathbf{L}}_{ij} := \widehat{\mathbf{P}}_i \widehat{\mathbf{T}}_{ij}$. Instead of Eq. (37b) we use Eq. (39b) — which is derived by applying $\widehat{\mathbf{P}}_i$ to Eq. (37a) — because an analytical preconditioner for $\widehat{\mathbf{L}}_{ii}$ is available from [27]. This system of equations can be written in the matrix format as

$$\sum_{j=1}^N \mathbf{A}_{ij} \begin{bmatrix} \widehat{\mathbf{x}}_j^+ \\ \widehat{\boldsymbol{\sigma}}_j^+ \end{bmatrix} = \begin{bmatrix} \mathbf{r}_i \\ \mathbf{s}_i \end{bmatrix}, \quad i = 1, \dots, N, \quad (40)$$

where

$$\mathbf{A}_{ij} := \begin{bmatrix} \alpha_i \delta_{ij} \mathbf{I} - \delta t \widehat{\mathbf{B}}_{ij} - \widehat{\mathbf{D}}_{ij} & -\delta t \widehat{\mathbf{T}}_{ij} \\ -\delta t \widehat{\mathbf{P}}_i \widehat{\mathbf{B}}_{ij} - \widehat{\mathbf{P}}_i \widehat{\mathbf{D}}_{ij} & -\delta t \widehat{\mathbf{L}}_{ij} \end{bmatrix}, \quad i, j = 1, \dots, N. \quad (41)$$

To further simplify the notation, let's write $\mathbf{A} = \mathbf{A}_d + \mathbf{A}_o$ where \mathbf{A}_d is the block diagonal matrix with \mathbf{A}_{ii} on its diagonal, and $\mathbf{A}_o = \mathbf{A} - \mathbf{A}_d$, and write $\mathbf{y}_i = [\widehat{\mathbf{x}}_i; \widehat{\boldsymbol{\sigma}}_i]$, $\mathbf{y}_i^+ = [\widehat{\mathbf{x}}_i^+; \widehat{\boldsymbol{\sigma}}_i^+]$, $\mathbf{b}_i = [\mathbf{r}_i; \mathbf{s}_i]$, and finally, $\mathbf{y} = [\mathbf{y}_1; \dots; \mathbf{y}_N]$, $\mathbf{y}^+ = [\mathbf{y}_1^+; \dots; \mathbf{y}_N^+]$, and $\mathbf{b} = [\mathbf{b}_1; \dots; \mathbf{b}_N]$. Now the system of equations that we are trying to solve can be written succinctly as

$$(\mathbf{A}_d + \mathbf{A}_o) \mathbf{y}^+ = \mathbf{b}. \quad (42)$$

The operator \mathbf{A}_d encompasses the self interaction of vesicles, which itself comprises evaluation of interfacial forces, surface differentiation, upsampling for anti-aliasing, and singular integration. The operator \mathbf{A}_o is for interaction between vesicles.

3.4. Solution Schemes

A linear system, Eq. (42), needs to be solved to compute the new position. This system does not necessarily need to be solved exactly. We need to decide on the accuracy of this solve based on a trade-off between computational costs and numerical stability. The most robust and accurate scheme is to use a Krylov iterative method like GMRES. The fastest way would be to perform a single block Jacobi iteration combined with local estimation. Below we overview the different strategies.

We present four different time-stepping schemes. The Block Jacobi/Explicit Position scheme is presented as base case for cost comparison. The Block Jacobi/Implicit method was proposed in [27] for vesicles with no viscosity contrast. Block and Globally Implicit schemes are the new schemes we introduce.

- *Block Jacobi/Explicit Position:* Given the current position and tension, we can use them as an initial guess for a single block-Jacobi iteration on the evolution matrix

$$\mathbf{A}_d \mathbf{y}^+ = \mathbf{b} - \mathbf{A}_o \mathbf{y}.$$

The solution to the diagonal blocks $\mathbf{A}_{ii} \mathbf{y}_i^+ = \mathbf{b}_i - (\mathbf{A}_o \mathbf{y})_i =: \tilde{\mathbf{b}}_i$ can be approximated individually by an explicit step for position (still solving for tension to enforce the local inextensibility constraint). First we solve for the tension $\hat{\boldsymbol{\sigma}}_i^+$:

$$-\delta t \hat{\mathbf{L}}_{ii} \hat{\boldsymbol{\sigma}}_i^+ = \hat{\mathbf{P}}_i (\delta t \hat{\mathbf{B}}_{ii} + \hat{\mathbf{D}}_{ii}) \hat{\mathbf{x}}_i + \tilde{\mathbf{s}}_i \quad (43)$$

Then, we can *explicitly* update the position

$$\hat{\mathbf{x}}_i^+ = \frac{1}{\alpha_i} \left[(\delta t \hat{\mathbf{B}}_{ii} + \hat{\mathbf{D}}_{ii}) \hat{\mathbf{x}}_i + \delta t \hat{\mathbf{T}}_{ii} \hat{\boldsymbol{\sigma}}_i^+ + \tilde{\mathbf{r}}_i \right].$$

Low accuracy and instability are major shortcomings of this approach. Because of explicit treatment of position, this method's stable time step degrades as we refine the spatial grid (Table 4).

- *Block Jacobi/Implicit Position:* Following the same approach as above, we can calculate the new tension $\hat{\boldsymbol{\sigma}}_i^+$, but instead of an explicit evaluation of the new position, we can use the first equation to solve for position, i.e

$$(\alpha_i \mathbf{I} - \delta t \hat{\mathbf{B}}_{ii} - \hat{\mathbf{D}}_{ii}) \hat{\mathbf{x}}_i^+ = \delta t \hat{\mathbf{T}}_{ii} \hat{\boldsymbol{\sigma}}_i^+ + \tilde{\mathbf{r}}_i. \quad (44)$$

This approach can be interpreted as a single block Gauss-Seidel iteration for the inner block. We used a variation of this approach for vesicles with no viscosity contrast in [27]. This approach increases the accuracy for the new position by treating the bending force implicitly and overcomes the instability limitation. However, it fails to resolve the close interactions well and therefore fails for the flow of vesicles with viscosity contrast.

- *Block Implicit:* For improved accuracy and stability, one can fully solve each of the diagonal blocks $\mathbf{A}_{ii} \mathbf{y}_i^+ = \mathbf{b}_i - (\mathbf{A}_o \mathbf{y})_i$. Letting \mathbf{y}^* denote the exact solution to Eq. (40), i.e. $(\mathbf{A}_d + \mathbf{A}_o) \mathbf{y}^* = \mathbf{b}$. The error in the new position and tension is

$$\|\mathbf{y}^+ - \mathbf{y}^*\| = \|\mathbf{A}_d^{-1} \mathbf{A}_o (\mathbf{y}^* - \mathbf{y})\| \leq \delta t \|\mathbf{A}_d^{-1} \mathbf{A}_o\| \|\mathbf{u}\|,$$

where \mathbf{u} denotes the velocity field. Hence, the magnitude of the error is controlled by $\delta t \|\mathbf{A}_d^{-1} \mathbf{A}_o\|$. Since the operator matrix \mathbf{A} depends on the configuration of vesicles, with fixed δt , one cannot guarantee a uniform error bound for the new tension and position over the course of a simulation. In Fig. 5 we plot the relative error of one Jacobi iteration with respect to the exact solution of the linear system for the given configuration of two vesicles as a function of their distance. As it is shown in the plot, when the vesicles get very close, a situation that may arise when the viscosity contrast is high, doing a single Jacobi iteration introduces error with a very large magnitude, which in turn causes the simulation to diverge.

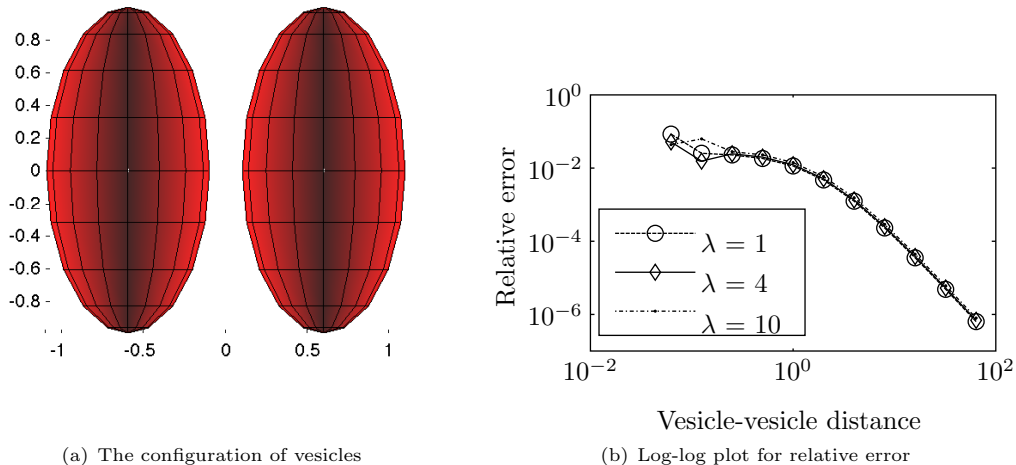


Figure 5: ACCURACY OF THE BLOCK IMPLICIT METHOD FOR TWO VESICLES. *The relative error when the solution of Eq. (40) is approximated by the result of a single Jacobi iteration. The configuration of vesicles is shown in the left subfigure. In the right subfigure, the ordinate is the relative error $\|\mathbf{y}^+ - \mathbf{y}^*\|/\|\mathbf{y}^*\|$ and the abscissa is the minimum distance between the surfaces of the two vesicles (the gap size in the left subfigure). The reference solution \mathbf{y}^* is calculated by an iterative method.*

- *Globally implicit:* Following the argument above, the most accurate way to evaluate the new position and tension is to solve Eq. (40) to a prescribed tolerance. This can be done by taking multiple Jacobi iterations — but for the same reason given above the convergence rate can be very slow — or by solving the system of equation using a Krylov subspace method such as GMRES. In this method, we solve Eq. (40) to a prescribed tolerance ϵ for the residual such that $\|\mathbf{A}\mathbf{y}^+ - \mathbf{b}\| \leq \epsilon\|\mathbf{b}\|$.

In Fig. 6 we demonstrate the stability of the fully implicit scheme. The plots in the figure show the interaction of two vesicles in shear flow. The viscosity contrast of both vesicles is 10 and the reduced volume is 0.85. Since vesicles with such a high viscosity contrast have a tendency to get very close in the shear flow, all other methods outlined above fail to capture their dynamics and diverge. With only $p = 8$ and in only 330 time steps, we can capture very rich dynamics that involve close interaction of vesicles, tank-treading, and tumbling. Tank-treading and tumbling of vesicles are distinct features of vesicle dynamics, observed extensively in experiments [12, 25].

To demonstrate the capabilities and limitations of the methods outlined above, we investigate the stable time step for these methods in case of one and two vesicles in shear flow in Table 4. The Explicit Position method shows stability constraint and as we refine the spatial resolution, the stable time step degrades. The Implicit Position and Block Implicit⁴ methods show no dependence on the spatial resolution. The last column in the table shows the stable time step for the interaction of two vesicles. An example of this type of simulation is shown in Fig. 6. We failed to find any stable time step for either of the Jacobi based methods. For the Globally Implicit case, however, the time step is the same as for one vesicle.

In the simulation of vesicles, the most expensive computation is the singular integration and with a given accuracy, the solution scheme of choice would be the one with the least number of singular integration per unit time of simulation, which therefore would be faster overall. Since the tension is the Lagrange multiplier that enforces the local inextensibility of the vesicles, even in the explicit case, at each time step we need to solve for tension. In Table 5 we report the computational cost (number of singular integral per unit time

⁴Of course, for one vesicle, Block and Globally Implicit cases are equivalent.

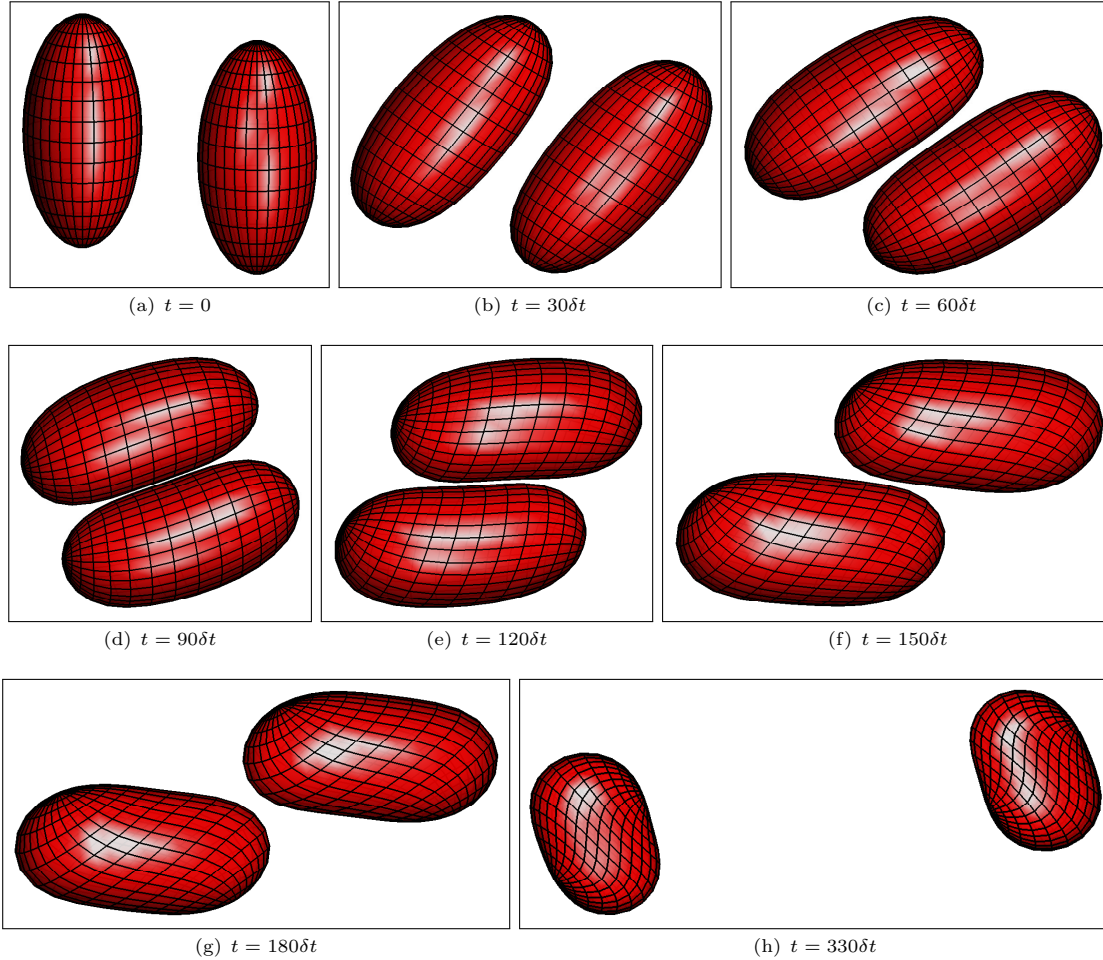


Figure 6: TWO VESICLES WITH VISCOSITY CONTRAST. *The plots show the interaction of two vesicles in the shear flow. The viscosity contrast of both vesicles is 10 and the reduced volume is 0.85. The time step δt is the same as the stable time step given in Table 4. For this simulation $p = 8$. Here we observe several characteristic vesicle dynamics, which are all observed experimentally. Namely, tank-treading (notable by following the grid lines over each surface) and tumbling of vesicles (subfigures g and h).*

of simulation) for two of the methods. It is evident from the table that in addition to higher accuracy, the implicit case is superior to the explicit case considering the cost per time unit.

3.5. Preconditioning

Each of the solution schemes that we outlined in Section 3.4 involves solving linear systems of equations of different sizes. For the Block Jacobi/Explicit Position, we solve for the tension of each vesicle, for Block Implicit we solve for position and tension of each vesicle simultaneously, and in the Globally Implicit method, we solve the whole system of equation at once. As we discussed in [27] all these linear systems are ill conditioned. The condition number of tension operator \mathbf{L} , Eq. (43), grows as $\mathcal{O}(p)$, and the condition number of the position operator, Eq. (44), grows as $\mathcal{O}(p^3)$. To improve the convergence rate of our iterative solvers, here we use the same set of preconditioners that we proposed in [27]. The tension operator is known

Time Scheme	λ	Single Vesicle					Two Vesicles
		$p = 8$	12	16	24	32	16
Block Jacobi/ Explicit Position	1	$2.7e-2$	$6.7e-3$	$1.7e-3$	$8.4e-4$	$2.1e-4$	–
Block Jacobi/ Implicit Position	1	$5.4e-2$	$5.4e-2$	$5.4e-2$	$5.4e-2$	$5.4e-2$	–
Globally Implicit	1	$2.1e-1$	$2.1e-1$	$2.1e-1$	$2.1e-1$	$2.1e-1$	$2.1e-1$
	4	$2.1e-1$	$2.1e-1$	$2.1e-1$	$2.1e-1$	$2.1e-1$	$2.1e-1$
	10	$2.1e-1$	$2.1e-1$	$2.1e-1$	$2.1e-1$	$2.1e-1$	$2.1e-1$

Table 4: STABLE TIME STEP. *The entries in the table are the non-dimensional stable time step multiplied by the non-dimensional shear rate, i.e. $\chi \delta t_{\text{stable}}$. λ is the viscosity contrast and p is the spherical harmonics order. For all schemes we calculated the stable time step for $\chi = 1, 10$, and 100. In the explicit case, these values varied slightly for different shear rates and we report the largest observed number. In other cases, no dependence on χ was observed. For all simulations, the bending modulus was set to 0.01, the time horizon was fixed to $T = 10\tau/\chi$ and was chosen such that one tank-treading or tumbling revolution was observed, the aliasing tolerance was $1e-4$, and the reparametrization tolerance was $1e-4$. In the case of two vesicles, we were not able to successfully capture the close interaction of vesicles in the Block Jacobi based schemes. In these cases, the vesicles collide and the time stepping diverges. For a single vesicle, Block and Globally Implicit schemes are equivalent and for two vesicles, the Block Implicit scheme also diverges (not reported in the table). It is only with the Globally Implicit scheme that we can capture the dynamics of vesicles with viscosity contrast.*

Time Scheme	λ	$p = 8$	12	16	24	32
Explicit Position	1	37	149	588	1190	4762
Implicit	1	203(42)	258(54)	268(56)	283(59)	290(60)
	4	133(27)	138(28)	134(28)	136(28)	137(28)
	10	88(18)	82(17)	82(17)	84(17)	84(17)

Table 5: THE COMPUTATION COST FOR A SINGLE VESICLE. *λ is the viscosity contrast and p is the spherical harmonics order. For each method, we report the average number of singular integrals per non-dimensional time unit of simulation. For the implicit case, inside the parenthesis, we report the average number of GMRES iterations per time step. We use the preconditioner given in Section 3.5. We considered the flow of a single vesicle for this table. The vesicle has reduced volume of 0.85. For each case, the time step is chosen from the Table 4. The time horizon is chosen such that tank-treading or tumbling of the vesicle is observed.*

analytically over \mathbb{S}^2 :

$$\widehat{\mathbf{L}}_{\mathbb{S}^2} = \text{diag} \left\{ -\frac{n(n+1)(2n^2+2n-1)}{(2n-1)(2n+1)(2n+3)} \right\}_{n=0,\dots,p}^{m=-n,\dots,n},$$

and we use its inverse as the preconditioner for the tension operator. For the position operator, we use a heuristic preconditioner that asymptotically matches the spectrum of bending operator, i.e. $\widehat{\mathbf{M}}_1 := \text{diag}\{(1 - \Delta t n^3)^{-1}\}$. For the Block Implicit scheme we use $\widehat{\mathbf{M}}_2 = \text{diag}\{\widehat{\mathbf{M}}_1, \widehat{\mathbf{L}}_{\mathbb{S}^2}^{-1}\}$ as the preconditioner; and for the Globally Implicit method involving N vesicles, we use a block diagonal matrix with N of $\widehat{\mathbf{M}}_2$ matrices on its diagonal.

4. Overall Algorithm and Computational Complexity

In this section we will give a summary of computational steps involved in the simulation of a suspension of vesicles. We assume that there are N vesicles in the suspension and all of these vesicles are represented with

spherical harmonics order of p . As we discussed in Section 3 the upsampling rate for differentiation depends on the shape of each vesicle. Nevertheless, for the sake of clarity, we assume that the same upsampling frequency q is shared by all vesicles. The generalization to the case in which each vesicle has a different spherical harmonics order and different upsampling rate is easy. The summary of the computational steps as well as their computational complexity is as follows:

- *Spherical harmonic transform:* There are several algorithms for the forward and inverse spherical harmonic transform [6]. The complexity of these algorithm ranges from $\mathcal{O}(p^4)$ for direct computation to $\mathcal{O}(p^2 \log^2 p)$ [16]. As it is stated in [16], the threshold where the (asymptotically) fast method becomes faster than the direct method is about $p = 128$. In the range of frequencies we use, direct Legendre transform in the latitude and FFT in the longitude direction has the complexity of $\mathcal{O}(p^3)$ and works quite well. We let $C_{\text{SH}}(p)$ denote the complexity of spherical harmonics transform for truncation order of p .
- *Differentiation:* Given the spherical harmonic coefficients of a function, differentiation is trivial in a Galerkin setup and a computational cost is only incurred when a mapping to the real space is performed. Since the surface differentials have geometric terms involved, mapping to the real space is required for them. Our dealiasing algorithm also includes a step where the surface is upsampled to some higher frequency q . Therefore the complexity of differentiation over each surface is $C_{\text{SH}}(q)$.
- *Singular integrals:* The single- and double- layer integrals over each vesicle, namely Eq. (30a) and Eq. (30b), can be efficiently computed using the algorithm given in [10, 11, 27] with complexity $C_{\text{Sing}}(p) = \mathcal{O}(p^5)$.
- *Vesicle-vesicle interaction:* The vesicle-vesicle interaction involves the evaluation of the non-singular Stokes integrals, which may involve evaluation of nearly-singular integrals. Our current approach to avoid nearly-singular integrals is to upsample the surfaces to some finer grid. Afterwards, we use Fast Multipole Method [29, 30, 32] to evaluate the Stokes integrals. The complexity of FMM is linear with respect to the number of discretization points, $C_{\text{FMM}}(N, p) = \mathcal{O}(Np^2)$. Therefore, in a Galerkin context, the complexity of vesicle interaction is $C_{\text{Inter}}(N, p) = C_{\text{FMM}}(N, p) + NC_{\text{SH}}(p)$, where the second term is due to the upsampling operation.
- *Evolution operator:* The evolution operator, which is given in Eq. (42), involves the evaluation of singular integrals over all surfaces (\mathbf{A}_d) and the interaction of vesicles (\mathbf{A}_o). The computation complexity of \mathbf{A}_d is $NC_{\text{Sing}}(p)$ and the complexity of \mathbf{A}_o is $C_{\text{Inter}}(N, p)$.
- *Reparametrization:* The reparametrization algorithm is given in Algorithm 3. Each reparametrization step includes differentiation (to compute the normal vector) and spherical harmonic transform for filtering. This implies that each step has the complexity of $C_{\text{SH}}(q)$. In all of our simulations, we never experienced a situation where more than a limited number of reparametrization steps (say more than 10) were needed. Thus, we assume $C_{\text{Reparam}}(q) = C_{\text{SH}}(q)$.
- *Upsampling rate calibration:* The algorithm to compute the upsampling rate is given in Algorithm 1. The body of the algorithm inside the while loop has $\mathcal{O}(q^3)$ complexity, and the loop is evaluated v times (where v is the final upsampling rate). Therefore, the calibration step has complexity $\mathcal{O}(p^3 v^4)$. Because the shape of each vesicle changes at the nondimensional time scale, we perform the calibration every $1/\delta t$ step. In practice this step has no extra computational cost compared to differentiation.

In Algorithm 2 we give the general simulation steps for a suspension of N vesicles.

Algorithm 2 MAIN STEPS FOR SIMULATION OF VESICLE FLOWS.

- (a) Form the right hand side for the evolution equation (Eq. (42)) using the current position of vesicles \mathbf{x} // $\mathcal{O}(NC_{\text{SH}}(q) + \mathbb{I}_{\lambda \neq 1}(NC_{\text{Sing}}(p) + C_{\text{Inter}}(p)))$, where $\mathbb{I}_{\lambda \neq 1}$ is the indicator function, signifying the fact that the double-layer terms are only present when there is viscosity contrast.
- (b) Approximate the solution to the evolution equation by either of the following methods:
- (i) Block Jacobi/Explicit Position // $\mathcal{O}(C_{\text{Inter}}(p) + NKC_{\text{Sing}}(p))$, where K is the average number of iterations for tension solver, Eq. (43).
 - (ii) Block Jacobi/Implicit Position // $\mathcal{O}(C_{\text{Inter}}(p) + N(K_p + K_t)C_{\text{Sing}}(p))$, where K_p and K_t are respectively the average number of iterations for position and tension solvers, Eq. (43) and Eq. (44).
 - (iii) Block Implicit // $\mathcal{O}(C_{\text{Inter}}(p) + KNC_{\text{Sing}}(p))$, where K is the average number of iteration for the diagonal blocks.
 - (iv) Globally Implicit // $\mathcal{O}(K(C_{\text{Inter}}(p) + NC_{\text{Sing}}(p)))$, where K is the average number of iteration for the linear solver for Eq. (42).
- (c) Reparametrize the surfaces based on Eq. (A.2). // $\mathcal{O}(NC_{\text{SH}}(q))$
- (d) Calibrate differentiation upsampling rate using Algorithm 1. // $\mathcal{O}(NC_{\text{SH}}(q))$
-

5. Conclusions

We proposed numerical schemes to simulate the motion of inextensible vesicles with viscosity contrast suspended in unbounded domain based on Galerkin method. We have demonstrated, through numerical experiments, that a globally implicit scheme does not exhibit a mesh-dependent stability constraint on the time-step size and the cost per nondimensional time unit is superior to that of the explicit scheme. Moreover, a globally implicit scheme has no proximity-dependent stability constraint for pairwise interaction of vesicles with moderate viscosity contrast. All schemes exhibit first-order accuracy in time and spectral-accuracy in space. We have presented efficient low-cost preconditioners to solve the discrete evolution equations by iterative solvers. An extension of our work would be to design an algorithm to handle nearly-singular integrals in the context of vesicle flows.

Appendix A. Reparametrization

We presented this reparametrization algorithm in [27]. A summary of the algorithm is included here for completeness. When membrane shear elasticity is present, as for example in Neo-Hookean constitutive laws (e.g., [33]), 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. To address these errors and enable long-time accurate simulations using

Algorithm 3 EXPLICIT REPARAMETRIZATION.

Require: \mathbf{x}

```
choose  $\Delta\tau, \rho_y, \rho_g, i_{\max}, i = 0$ 
 $\mathbf{y} \leftarrow$  upsample  $\mathbf{x}$  // Upsample to the anti-aliasing frequency
 $\mathbf{g}_0 = -(I - \mathbf{n}(\mathbf{y}) \otimes \mathbf{n}(\mathbf{y}))\nabla E(\mathbf{y})$ 
while  $++i < i_{\max}$  do
   $\mathbf{g} \leftarrow -(I - \mathbf{n}(\mathbf{y}) \otimes \mathbf{n}(\mathbf{y}))\nabla E(\mathbf{y})$  // projected gradient
   $\mathbf{y} \leftarrow \mathbf{y} + \Delta\tau\mathbf{g}$ 
  if  $\|\mathbf{g}\| < \max(\rho_y/\Delta\tau, \rho_g\|\mathbf{g}_0\|)$  then
    break
  end if
end while
 $\mathbf{y} \leftarrow$  downsample  $\mathbf{y}$  to  $p$ 
return  $\mathbf{y}$ 
```

a small number of spherical harmonics coefficients, we reparametrize 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 anti-aliasing is nonlinear: it is not done by a linear spectral projection but by solving a nonlinear variational problem.*

A detailed formulation of the reparametrization scheme is given in [27], here we outline the formulation. As before, we define the surface γ as the image of an embedding of the sphere $x(s) : \mathbb{S}^2 \rightarrow \mathbb{R}^3$ where s is a point on \mathbb{S}^2 and x is sufficiently smooth. We let $F : \mathbb{R}^3 \rightarrow \mathbb{R}$ denote the implicit representation of the surface such that $F(\gamma) = 0$ and ∇F does not vanish. Our goal is to choose one that minimizes the quality measure $E(y) := \sum_{n=0}^p \sum_{m=-n}^n a_{nm} |\hat{y}_{nm}|^2$:

$$\arg \min_{y \in C^\infty(\mathbb{S}^2)} E(y(s)) \quad \text{subject to} \quad F(y(s)) = 0 \quad \text{for all } s \in \mathbb{S}^2,$$

where a_{nm} is some attenuation coefficient. After introducing the Lagrangian and taking the variation of the energy E with respect to y and the Lagrange multiplier (and some algebraic manipulation) we get the first order optimality condition as

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

We solve this using pseudo-transient continuation [14] by introducing a parameter τ :

$$\frac{\partial y}{\partial \tau} + (I - n(y) \otimes n(y))\nabla E(y) = 0, \quad y(0) = x, \quad \text{and} \quad F(y) = 0. \quad (\text{A.1})$$

This evolution of y cannot increase the energy, since we are moving in the constraint manifold-projected steepest descent direction. Our parametrization optimization can be viewed as a discretization of this flow using an explicit scheme:

$$\mathbf{y}_{k+1} = \mathbf{y}_k - \Delta\tau(I - \mathbf{n}_k \otimes \mathbf{n}_k)\nabla E(\mathbf{y}_k). \quad (\text{A.2})$$

Since we want E to penalize the high frequencies in order to minimize aliasing errors, the attenuation a_{nm} should be small for low frequencies and should grow for high frequencies—in the special case of perfect high-pass filter $a_{nm} = 0$ for $n < n_{\text{cutoff}}$ and $a_{nm} = 1$ otherwise. Then, the variation of E with respect to y is $\nabla E = \sum_{n=\text{cutoff}}^p \sum_{m=-n}^n \hat{y}_{nm} Y_{nm}$. We experimentally found out that $n_{\text{cutoff}} = p/2$ gives satisfactory results and only needs very few iterations.

In Algorithm 3, we give the pseudo code for the reparametrization step. Notice that in our implementation we use upsampling (by the factor dictated from the anti-aliasing algorithm). 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 is to improve the accuracy of the map to tangent space. One additional question is how to choose ρ_y and ρ_g . Parameter ρ_g ensures that if the gradient becomes too small the algorithm terminates and ρ_y ensures that if the change in \mathbf{y} becomes small the algorithm terminates—we choose $\rho_y = 10^{-2} \|\mathbf{x} - \mathbf{x}^{\text{old}}\|$, where \mathbf{x}^{old} is the previous position during the vesicle evolution.

References

- [1] Uri M. Ascher, Steven J. Ruuth, and Brian T. R. Wetton. Implicit-explicit methods for time-dependent partial differential equations. *SIAM Journal on Numerical Analysis*, 32(3):797–823, 1995.
- [2] Prosenjit Bagchi and R. Murthy Kalluri. Dynamics of nonspherical capsules in shear flow. *Physical Review E*, 80(1), 2009.
- [3] Prosenjit Bagchi and R. Murthy Kalluri. Dynamic rheology of a dilute suspension of elastic capsules: effect of capsule tank-treading, swinging and tumbling. *Journal of Fluid Mechanics*, 669:498–526, 2011.
- [4] 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.
- [5] T Biben and C Misbah. An advected-field method for deformable entities under flow. *European Physical Journal B*, 29(2):311–316, 2002.
- [6] John P. Boyd. *Chebyshev and Fourier Spectral Methods*. Dover Publications Inc., 1999.
- [7] 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.
- [8] Alexander Farutin, Thierry Biben, and Chaouqi Misbah. 3D Numerical simulations of vesicle and inextensible capsule dynamics. Unpublished, 2013.
- [9] JB Freund. Numerical Simulation of Flowing Blood Cells. *Annual Review of Fluid Mechanics*, 46(1): 67–95, January 2014.
- [10] 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.
- [11] 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.
- [12] Vasilii Kantsler and Victor Steinberg. Orientation and dynamics of a vesicle in tank-treading motion in shear flow. *Physical Review Letters*, 95(25), 2005.
- [13] 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.
- [14] C. T. Kelley and David E. Keyes. Convergence analysis of pseudo-transient continuation. *SIAM Journal on Numerical Analysis*, 35:508–523, 1998.
- [15] M. Kraus, W. Wintz, U. Seifert, and R. Lipowsky. Fluid vesicles in shear flow. *Physical Review Letters*, 77(17):3685–3688, 1996.
- [16] M. J. Mohlenkamp. A fast transform for spherical harmonics. *Journal of Fourier analysis and applications*, 5(2):159–184, 1999.

- [17] S. A. Orszag. Fourier series on spheres. *Monthly Weather Review*, 102:56–75, 1974.
- [18] Steven A. Orszag. Numerical simulation of incompressible flows within simple boundaries: accuracy. *Journal of Fluid Mechanics*, 49:75–112, 1971.
- [19] C. Pozrikidis. *Boundary Integral and Singularity Methods for Linearized Viscous Flow*. Cambridge University Press, Cambridge, 1992.
- [20] C. Pozrikidis. Interfacial dynamics for Stokes flow. *Journal of Computational Physics*, 169:250–301, 2001.
- [21] C. Pozrikidis. Effect of membrane bending stiffness on the deformation of capsules in simple shear flow. *Journal of Fluid Mechanics*, 440:269–291, 2001.
- [22] C. Pozrikidis. A spectral collocation method with triangular boundary elements. *Engineering Analysis with Boundary Elements*, 30(4):315 – 324, 2006.
- [23] Bryan Quaife and George Biros. High-volume fraction simulations of two-dimensional vesicle suspensions. arXiv:1309.1128, 2013.
- [24] Y. Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, second edition, 2003.
- [25] U. Seifert. Configurations of fluid membranes and vesicles. *Advances in Physics*, 46:13–137, 1997.
- [26] A. Tornberg and M. J. Shelley. Simulating the dynamics and interactions of flexible fibers in stokes flows. *Journal of Computational Physics*, 196(1):8–40, 2004.
- [27] Shravan K. Veerapaneni, Abtin Rahimian, George Biros, and Denis Zorin. A fast algorithm for simulating vesicle flows in three dimensions. *Journal of Computational Physics*, 230(14):5610–5634, 2011.
- [28] J. L. Weiner. On a problem of Chen, Willmore, et al. *Indiana University Mathematics Journal*, 27: 19–35, 1978.
- [29] Lexing Ying, George Biros, Denis Zorin, and Harper Langston. A new parallel kernel-independent fast multipole algorithm. In *Proceedings of SC03*, The SCxy Conference series, pages 14–14, Phoenix, Arizona, 2003. ACM/IEEE.
- [30] Lexing Ying, George Biros, and Denis Zorin. A kernel-independent adaptive fast multipole method in two and three dimensions. *Journal of Computational Physics*, 196(2):591–626, 2004.
- [31] Lexing Ying, George Biros, and Denis Zorin. A high-order 3d boundary integral equation solver for elliptic pdes in smooth domains. *Journal of Computational Physics*, 219(1):247–275, 2006. ISSN 0021-9991.
- [32] Lexing Ying, George Biros, Harper Langston, and Denis Zorin. KIFMM3D: The kernel-independent fast multipole (FMM) 3D code, 2011. GPL license.
- [33] 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*, 2010.
- [34] Hong Zhao and Eric S. G. Shaqfeh. The dynamics of a vesicle in simple shear flow. *Journal of Fluid Mechanics*, 674:578–604, March 2011. ISSN 0022-1120. doi: 10.1017/S0022112011000115.
- [35] Hong Zhao and Eric S. G. Shaqfeh. The dynamics of a non-dilute vesicle suspension in a simple shear flow. *Journal of Fluid Mechanics*, 725:709–731, May 2013.

- [36] A.Z. Zinchenko and R. H. Davis. Large-scale simulations of concentrated emulsion flows. *Philosophical Transactions Of The Royal Society Of London Series A-Mathematical Physical And Engineering Sciences*, 361(1806):813–845, 2003.