Journal of Computational Physics 228 (2009) 2334–2353

Contents lists available at ScienceDirect

Journal of Computational Physics

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

A boundary integral method for simulating the dynamics of inextensible vesicles suspended in a viscous fluid in 2D ☆

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

[a] School of Engineering and Applied Sciences, University of Pennsylvania, Philadelphia, PA 19104, United States
[b] Courant Institute of Mathematical Sciences, New York University, New York 10012, United States
[c] College of Computing, Georgia Institute of Technology, Atlanta GA 30332, United States

ARTICLE INFO

Article history:
Received 3 February 2008
Received in revised form 11 September 2008
Accepted 26 November 2008
Available online 31 December 2008

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

ABSTRACT

We present a new method for the evolution of inextensible vesicles immersed in a Stokesian fluid. We use a boundary integral formulation for the fluid that results in a set of nonlinear integro-differential equations for the vesicle dynamics. The motion of the vesicles is determined by balancing the non-local hydrodynamic forces with the elastic forces due to bending and tension. Numerical simulations of such vesicle motions are quite challenging. On one hand, explicit time-stepping schemes suffer from a severe stability constraint due to the stiffness related to high-order spatial derivatives and a milder constraint due to a transport-like stability condition. On the other hand, an implicit scheme can be expensive because it requires the solution of a set of nonlinear equations at each time step. We present two semi-implicit schemes that circumvent the severe stability constraints on the time step and whose computational cost per time step is comparable to that of an explicit scheme. We discretize the equations by using a spectral method in space, and a multistep third-order accurate scheme in time. We use the fast multipole method (FMM) to efficiently compute vesicle–vesicle interaction forces in a suspension with a large number of vesicles. We report results from numerical experiments that demonstrate the convergence and algorithmic complexity properties of our scheme.

© 2008 Elsevier Inc. All rights reserved.

1. Introduction

Vesicle flows model numerous biophysical phenomena that involve deforming particles interacting with a Stokesian fluid. The evolution dynamics are characterized by a competition between membrane elastic energy, inextensibility, and non-local hydrodynamic forces. Inextensible vesicles have received a lot of attention in the physics community as they are considered good models of biological cells. In this paper, our goal is to develop efficient numerical schemes for such flows. In the case where the fluids both inside and outside of the vesicle are the same, the equations that govern the motion of a single vesicle are

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

☆ This work was supported by the US Department of Energy under Grant DE-FG02-04ER25646, and the US National Science Foundation Grants CCF-0427985, CNS-0540372, DMS-0612578, and OCI 0749285.

* Corresponding author. Tel.: +1 215 898 3108; fax: +1 215 573 6334.
E-mail addresses: shravan@seas.upenn.edu (S.K. Veerapaneni), ddg5@cims.nyu.edu (D. Gueyffier), dzorin@cims.nyu.edu (D. Zorin), gbiros@gmail.com, biros@seas.upenn.edu (G. Biros).

0021-9991/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2008.11.036

where s is the arclength parameter, $\mathbf{x}(s,t)$ is the interfacial position, $\mathbf{f}_\sigma = (\sigma \mathbf{x}_s)_s$ (the subscript s denotes differentiation with respect to arclength) $\mathbf{f}_b = -\kappa_B \mathbf{x}_{ssss}$, σ is the tension, κ_B is the bending modulus, $\mathbf{v}_\infty$ is the far-field velocity of the bulk fluid, and $\mathcal{S}$ is the single-layer potential Stokes operator, defined in Section 2. The first equation in (1) describes the motion of the vesicle boundary; the second equation expresses the local inextensibility of the interface. As in the case of most biological membranes at mesoscopic length scales, vesicles can be modeled by smooth periodic curves [37]. In the rest of the paper, we assume that $\mathbf{x}$ is a C^∞ function of s for all times. An example of motion of multiple vesicles is given in Fig. 1.

In contrast to stencil-based formulations, like finite element and finite difference methods, integral equation formulations avoid discretization of the overall domain and instead, only discretize the vesicle boundaries. This is the main reason that integral equations have been used extensively for vesicle, and more generally, particulate and interfacial flow simulations [32]. Despite their success, several issues remain with respect to their numerical implementation. In particular, an efficient numerical scheme for (1), should address the following issues:

- *Stability*. The bending force, which involves high-order spatial derivatives, makes the evolution Eq. (1) numerically stiff. Consequently, a fully explicit scheme in time leads to stringent restrictions on the time step size.
- *Ill-conditioning*. If an implicit scheme were to be used, "inverting" the associated Jacobians would be computationally expensive due to ill-conditioning (Section 3).
- *Accuracy*. The convergence rate of the overall numerical solution is governed by the accuracy of the discretization scheme in time, the quadrature rule to compute the single-layer potentials, and the evaluation of spatial derivatives; to achieve high convergence rates all of these components must be chosen carefully.

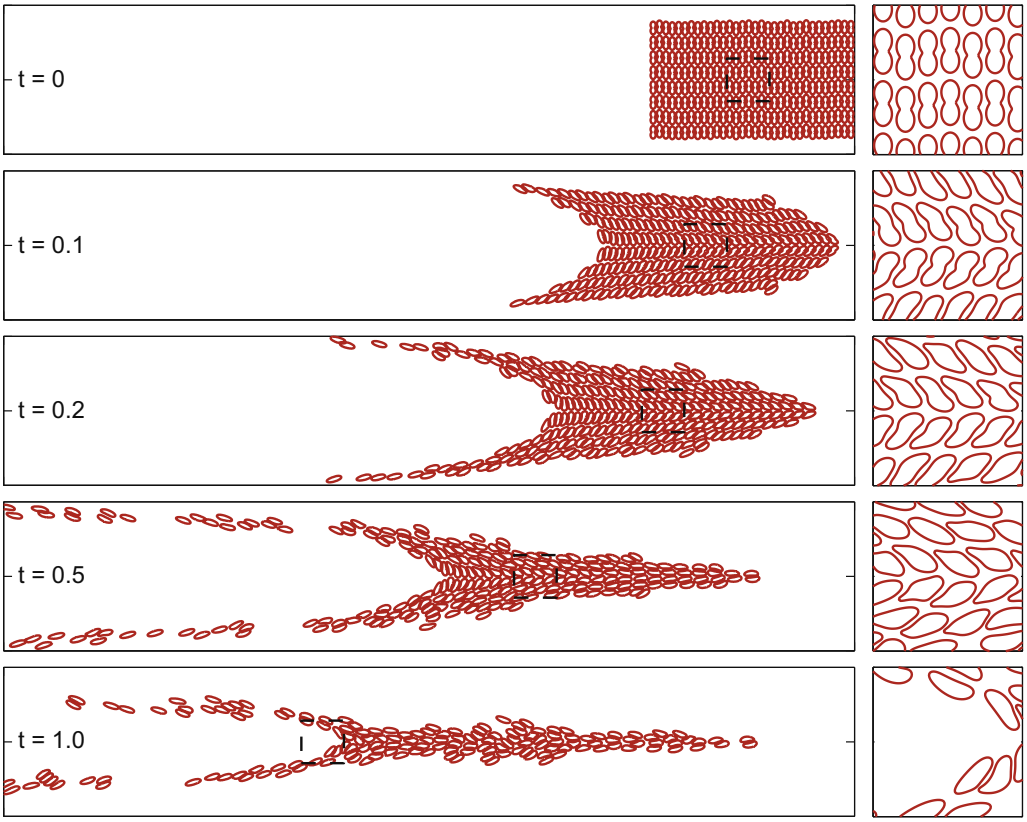

Fig. 1. In this figure, we demonstrate the capabilities of our method, in particular the running times per time step and the ability to resolve complex interactions between multiple vesicles. We simulated the motion of 256 vesicles driven by an external flow that has a parabolic profile. In this simulation, we used 64 discretization points per vesicle and we took a total of 1000 time steps. The wall-clock time per time step was 40 s (on average) on a Xeon workstation. The computations were performed using MATLAB, accelerated by external FFT, FMM, and LAPACK libraries. In the left column, we show four snapshots of the overall simulation and in the right column, we zoom in the region marked by the broken-line square to show the details on the shapes of individual vesicles. Here, t is a nondimensional time $t \in (0,1)$. The initial state is a rectangular array of vesicles in a non-equilibrium shape configuration (top row). Due to bending, the vesicle shapes are quickly smoothed. Then, the vesicles are dispersed by the shear of the background flow. We resolve high curvature regions (e.g., fourth row, second column), conserve vesicle areas and lengths, and compute the hydrodynamic interactions with sufficient accuracy to avoid collisions without employing a collision detection algorithm. Details on the accuracy and complexity of our method are presented in later Sections.

- *Algorithmic complexity*. Computing the single-layer potential at M target locations $\{\mathbf{x}_k\}_{k=1}^{M}$ from given source locations (quadrature nodes) $\{\mathbf{y}_j\}_{j=1}^{M}$ is $\mathcal{O}(M^2)$. This would severely limit the scope of computation, when one wants to simulate large number of vesicles and/or for large number of time steps.

Synopsis of our method. Inspired by the work of Hou et al. [16], Kropinski [23], and Tornberg and Shelley [41] on fast, high-accuracy solvers for problems with moving interfaces, we propose two computational schemes for simulating the motion of inextensible vesicles in two-dimensions. Our schemes address all of the issues outlined above but the accuracy in time. Both schemes are based on Lagrangian tracking of marker particles placed on the membrane of the vesicle and a semi-implicit time discretization, which, experimentally, appears to have no time-stepping stability constraints. High-order accuracy in space is ensured by using a Fourier basis discretization for all functions and computing derivatives in Fourier domain, and special high-order Gauss-trapezoidal quadrature rules, introduced by Alpert [1], designed to resolve the logarithmic singularities that appear in single-layer potentials. For the position update in time, we use two variants of a semi-implicit marching scheme first derived for advection–diffusion equations [3] and then applied on integral equation based fluid-structure interaction problems in [41] (for rod dynamics modeled by slender-body theory integral formulations – a different formulation from the one we use for vesicles); for certain flow regimes both schemes attain high-order accuracy. The time-marching schemes require the solution a linear system of equations for each time step, which is solved using a Krylov iterative method (GMRES) [35]. The problem of poor conditioning is addressed by a preconditioner based on the analytically obtained spectrum of the operators in (1) for the special case (unit circle). The complexity of matrix–vector multiplication for the dense system is reduced to linear in the number of variables using FMM. As a result, we are able to achieve high-accuracy while using a small number of unknowns per vesicle for the spatial discretization, while taking large time steps with relatively low computational cost per each time step. These improvements enable the simulation a large number of interacting vesicles, as described in Section 5 and depicted in Fig. 1.

Contributions. We would like to emphasize that the use of boundary spectral representations, semi-implicit schemes, and fast summation schemes in the context of interface problems is not novel. However, we are not aware of any previous analysis and application of implicit time-stepping schemes combined with fast solvers to vesicles suspended in Stokesian fluids. Two features distinguishing vesicles from droplets and bubbles are the bending forces related to curvature and the surface inextensibility constraint.

The contributions of this paper are: (1) the spectral analysis of the different operators related to vesicle dynamics for the unit circle and their use to derive preconditioning techniques; (2) the extension of the techniques developed in [16,23,40] to vesicle flows; (3) the numerical investigation of the stability and accuracy of the time-stepping schemes; and (4) a preliminary validation of our methodology by comparing our results to results in the literature.

Limitations. We restrict our attention to dilute suspension of vesicles in fluids with unbounded domains. The model of vesicle flows we consider here does not include forces due to gravity, electrostatics or adhesion, or inertial effects due to the mass of the fluid or the membrane. Also, we do not consider topological changes, which are often present in many biophysical phenomena involving vesicles.

Maintaining high-accuracy for vesicles closely approaching each other requires incorporation of specialized quadrature rules for nearly-singular integrals, as well as appropriate models for short-range interaction forces and efficient collision detection schemes. We do not consider such cases here, which is an important limitation. For an example of related work on this topic see [21].

In our examples, we assume that the interior and exterior of the vesicles are filled with the same liquid. The algorithm extends to the more general case, which requires evaluation of double layer potentials [32].

Our numerical experiments indicate a time step stability that is proportional to the shear rate, but it is *independent* of the spatial discretization size; experimentally, we have observed that the overall accuracy of our method is dictated by the accuracy of the time stepping scheme and is up to third-order for non-vanishing shear rates.

1.1. Related work

Vesicles (also known as fluid membranes) attracted the attention of scientists and engineers as they are present in many biological phenomena [19] and can be used experimentally to understand properties of biological membranes [12]. In addition, vesicle mechanics have been used as models for red blood cells [26,28] and drug-carrying capsules [38].

Vesicle simulations have been based both on molecular dynamics models [25] and on continuum mechanics models of the fluid and the vesicle membrane. Here, we focus on numerical schemes for continuum models of vesicle dynamics. We start by briefly reviewing the more general topic of Stokesian particulate flows. Numerical methods for Stokesian flows can be classified to unstructured finite element methods, Cartesian-grid based methods, and integral equation methods. We refer to [5] for a review of these methods for boundary value problems. Integral equation methods have been used extensively for the simulation of Stokesian particulate flows. Integral equations were introduced in Youngren and Acrivos [17] for a flow past a rigid particle of arbitrary shape. The same authors used integral equations to study the shape of a bubble in an extensional flow. Our work is based on a formulation derived by Rallison and Acrivos [33] for two fluids separated by an interface with surface forces. For a detailed presentation of the theory of integral equations for Stokes flows see [27,29].

Typical spatial discretizations of integral equation formulations are based on Galerkin or collocation projection schemes using polynomial bases – also referred as boundary element methods (BEM). Such methods have been used extensively

[24,28,30,31,44,6] to study the dynamics of a single vesicle (with compressible interfaces) suspended in Stokes flow. Most existing numerical methods use relatively low-order representation of the interface boundary, at most second order (circular arcs) or third order (cubic splines) [29,31,32], together with consistent quadrature rules. Spectral representations for two-dimensional interfacial flows with stiffness appeared in [4,16]. In the context of Stokesian flows, the closest work to ours is that in [23,22] for extensible interfaces without bending energy. Another spectral approach has been used in [44] in which the tension is calculated to enforce inextensibility but the position of the vesicle is updated explicitly. Also, a low-accuracy representation is used for the boundary, so the overall accuracy is of low-order. One advantage to our approach is its high-order of spatial accuracy and its ability to treat very large numbers of vesicles with a smaller number of marker points per vesicle.

One of the greatest challenges in numerical schemes for vesicle dynamics is the numerical stiffness. The overwhelming majority of work on particulate flows uses explicit schemes that pose severe restrictions on the time step. A powerful method in treating stiffness is related to the so-called *"small-scale decomposition"* [4,16,23] in which appropriate linearizations reveal the higher derivative terms that are responsible for the numerical stiffness; these terms are treated implicitly whereas the remaining terms are treated explicitly. In our work, we use a related but different idea, in which we analytically construct a partial-linearization Jacobian in the spectral domain for the case of a circle and then, we use it as a preconditioner in the two semi-implicit schemes we propose in which we treat only the higher order terms implicitly. Another approach that allows stable time-stepping is to use Newton's method with exact Jacobians [9] but such an approach will have quite large computational costs per time step. To our knowledge, no work on implicit schemes exists for incompressible vesicles.

The literature on numerical methods for incompressible vesicles appears to be somewhat limited; only three papers discussing numerical schemes for such problems: Kraus et al. [20], Zhou and Pozrikidis [44], and Sukumaran and Seifert [39]. All these works use the forward Euler discretization method in time, which is computationally inefficient because (1) is stiff. In problems with vesicles, the elastic and incompressibility properties of the membranes must be taken into account and the numerical schemes must be modified in order to solve the resulting set of boundary integral equations. Details of the BEM for elastic interfaces and incompressible vesicles can be found in [6,44]. In the present study, we will use the model of [20,39] for the bending energy and for the surface tension.

Other related work includes methods to compute the vesicle shape that corresponds to an elastic equilibrium (minimum L^2-norm of curvature) under area and length (volume and area in 3D) constraints without hydrodynamic interactions (this methods consider a single vesicle). Examples include a phase field method, proposed in [10,11], that can handle topological changes and [13] that uses a shell finite element based algorithm.

1.2. Contents

In Section 2, we present a summary of the derivation of the integro-differential equations (1) that govern vesicle dynamics. A qualitative understanding of stiffness can be obtained from the knowledge of the spectrum of the Jacobian of (1). Parts of the Jacobian can be analytically computed using the Fourier transform on the unit circle. We discuss the behavior of the spectrum in Section 2.1. The source of the high-order stiffness will be evident from this analysis.

In Section 3, we present two numerical schemes that overcome the high-order stiffness, are spectrally accurate in space and, for certain flow regimes, attain high-order accuracy in time; we extend these schemes to deal with multiple vesicles. Both schemes lead to systems of linear equations that need to be solved at each time step. The spectral analysis of the problem on the unit circle is used to construct preconditioners for the iterative linear solvers required in the time-stepping schemes.

In Section 5, we present numerical results for a number of problems involving single and multiple vesicles suspended in a viscous fluid. We conduct numerical experiments to investigate the stability and convergence order of different time-stepping schemes.

2. Problem formulation

Consider a single vesicle suspended in a 2D viscous fluid domain Ω and whose membrane is denoted by γ. Assume that the interior of the vesicle is filled with the same fluid. The fluid flow is modeled by the Stokes equations,

$$\nabla p - \mu \Delta \mathbf{v} = \mathbf{f}; \quad \operatorname{div} \mathbf{v} = 0 \quad \text{in} \quad \Omega \setminus \gamma, \tag{2}$$

where $p(\mathbf{x})$ and $\mathbf{v}(\mathbf{x})$ are the pressure and velocity fields, and μ is the viscosity of the fluid. The no-slip boundary condition at the vesicle boundaries and the free-space boundary condition require that

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

where $\dot{\mathbf{x}}$ is the velocity of a point on γ and $\mathbf{v}_\infty$ is the far-field fluid velocity.

The elastic energy of the membrane is given by $\varepsilon(\kappa, \sigma) = \int_{\gamma(s)} \frac{1}{2}\kappa_B \kappa^2 + \sigma\, ds$ where κ is its curvature, σ is the tension, and κ_B is the bending modulus. The forces due to bending, $\mathbf{f}_b$, and tension, $\mathbf{f}_\sigma$, are obtained by taking the L^2-gradient of ε with respect to $\mathbf{x}(s)$ (Appendix A). The total force, $\mathbf{f}_b + \mathbf{f}_\sigma$ is balanced by the jump of the fluid stress vector across the vesicle membrane, $\mathbf{f}$, across the vesicle membrane. We assume that no other forces (e.g., gravitational) are present in the system. Using

potential theory [27], the solution of (2) can be written as $\mathbf{v}(\mathbf{x}) = \mathbf{v}_\infty(\mathbf{x}) + \mathcal{S}[\mathbf{f}_b + \mathbf{f}_\sigma](\mathbf{x})$. The single-layer potential $\mathcal{S}[\mathbf{f}]$ is defined as $\mathcal{S}[\mathbf{f}](\mathbf{x}) = \int_\gamma G(\mathbf{x},\mathbf{y})\mathbf{f}(\mathbf{y})\,d\gamma(\mathbf{y})$, where the 2D Stokes free-space kernel, G, is given by

$$G(\mathbf{x},\mathbf{y}) = \frac{1}{4\pi\mu}\left(-\ln\rho\mathbf{I} + \frac{\mathbf{r}\otimes\mathbf{r}}{\rho^2}\right), \quad \mathbf{r} = \mathbf{x}-\mathbf{y}, \quad \rho = \|\mathbf{r}\|_2. \tag{4}$$

Enforcing the no-slip boundary condition, we get the first equation in (1). The tension σ can be viewed as a Lagrange multiplier that enforces the local inextensibility constraint. It is computed by requiring the 'surface' divergence, div_γ of the interfacial velocity field is zero, that is, $\mathbf{x}_s \cdot \mathbf{v}_s(\mathbf{x}) = 0$.[1] This leads to the second equation in (1).

To construct and analyze our numerical scheme, we introduce the operators $\mathcal{B}, \mathcal{T}, \mathcal{D}, \mathcal{L}, \mathcal{M}$, defined on the interface $\mathbf{x}(s)$, for any point $\mathbf{y} \in \mathbb{R}^2$; $\mathbf{f}$ is a smooth vector field on γ and σ is a smooth scalar field on γ:

$$\begin{aligned}
&\mathcal{B}(\mathbf{y},\mathbf{x})\mathbf{f} := -\mathcal{S}[\mathbf{f}_{ssss}](\mathbf{y});\\
&\mathcal{T}(\mathbf{y},\mathbf{x})\sigma := \mathcal{S}[(\sigma\mathbf{x}_s)_s](\mathbf{y});\\
&\mathcal{D}(\mathbf{x})\mathbf{f} := \mathbf{x}_s \cdot \mathbf{f}_s;\\
&\mathcal{L}(\mathbf{x}) := D(\mathbf{x})\mathcal{T}(\mathbf{x},\mathbf{x});\\
&\mathcal{M}(\mathbf{x}) := \mathcal{T}(\mathbf{x})\mathcal{L}^{-1}(\mathbf{x})\mathcal{D}(\mathbf{x}).
\end{aligned} \tag{5}$$

If $\mathbf{f} = \kappa_B\mathbf{x}$, $\mathcal{B}\mathbf{f}$ gives the single-layer potential at $\mathbf{y}$ due to the bending force on the interface. Similarly, $\mathcal{T}\sigma$ gives the single-layer potential due to the tension force. We define $\mathcal{B}(\mathbf{x}) = \mathcal{B}(\mathbf{x},\mathbf{x})$ and $\mathcal{T}(\mathbf{x}) = \mathcal{T}(\mathbf{x},\mathbf{x})$. Following this notation, we can rewrite the governing equations for a single vesicle as

$$\dot{\mathbf{x}} = \mathbf{v}_\infty(\mathbf{x}) + \kappa_B\mathcal{B}(\mathbf{x})\mathbf{x} + \mathcal{T}(\mathbf{x})\sigma, \quad \mathcal{L}(\mathbf{x})\sigma = -\mathcal{D}(\mathbf{x})[\mathbf{v}_\infty(\mathbf{x}) + \kappa_B\mathcal{B}(\mathbf{x})\mathbf{x}]. \tag{6}$$

Alternatively, we can eliminate the equation for the surface tension.

$$\dot{\mathbf{x}} = \mathbf{v}_\infty(\mathbf{x}) + \kappa_B\mathcal{B}(\mathbf{x})\mathbf{x} - \mathcal{T}(\mathbf{x})\mathcal{L}^{-1}(\mathbf{x})\mathcal{D}(\mathbf{x})[\mathbf{v}_\infty(\mathbf{x}) + \kappa_B\mathcal{B}(\mathbf{x})\mathbf{x}], = (1-\mathcal{M}(\mathbf{x}))[\mathbf{v}_\infty(\mathbf{x}) + \kappa_B\mathcal{B}(\mathbf{x})\mathbf{x}], \tag{7}$$

where the $\mathcal{M}$ operator, which we call *stretching operator*, modifies the interface velocity field $\dot{\mathbf{x}}$ to enforce the inextensibility constraint.

Scaling. In most of our numerical experiments we focus in the case in which $\mathbf{v}_\infty$ is a simple shear flow. Following the analysis of [20], we define different scales as follows. The velocity is given by $\mathbf{v}_\infty = \chi(x_2, 0)$, where χ is the shear rate. The length scale R_0 is determined by the perimeter L of the boundary, given by $R_0 = L/2\pi$, which is the radius of circle having the same perimeter. Then, the time scale τ is defined by $\tau = \mu R_0^3/\kappa_B$. The dimensionless shear rate is defined by $\tilde{\chi} = \dot{\gamma}\mu R_0^3/\kappa_B$. The governing equation in the nondimensional form, for a vesicle suspended in simple shear flow, becomes

$$\dot{\tilde{\mathbf{x}}} = \tilde{\mathbf{v}}_\infty + \mathcal{B}(\tilde{\mathbf{x}})\tilde{\mathbf{x}} + \mathcal{T}(\tilde{\mathbf{x}})\tilde{\sigma}, \tag{8}$$

where $\tilde{\mathbf{x}} = \frac{\mathbf{x}}{R_0}$, $\tilde{\sigma} = \frac{R_0^2\sigma}{\kappa_B}$ and $\tilde{\mathbf{v}}_\infty = \tilde{\chi}(\tilde{x}_2, 0)$. Hence, $\tilde{\chi}$ and τ characterize the vesicle dynamics. From now on, unless stated otherwise, all the equations are written in nondimensional form and for simplicity of notation and we suppress '$\sim$' in the notation. Another parameter that we use is the so-called *reduced area* denoted by ν and defined as $\nu = \frac{A}{\pi R_0^2} = \frac{4\,A\pi}{L^2}$, where A is the area of the vesicle. It is the ratio of the vesicle area over the area of circle of the same perimeter. It is used extensively in the literature to classify vesicle shapes.

Multiple vesicles. If K vesicles are suspended in the shear flow, equations (1) can be expanded into the following equations for the evolution of the jth vesicle:

$$\dot{\mathbf{x}}_j = \mathbf{v}_\infty(\mathbf{x}_j) + \mathcal{B}(\mathbf{x}_j)\mathbf{x}_j + \mathcal{T}(\mathbf{x}_j)\sigma_j + \sum_{\substack{k=1\\k\neq j}}^{K} \mathcal{B}(\mathbf{x}_j,\mathbf{x}_k)\mathbf{x}_k + \mathcal{T}(\mathbf{x}_j,\mathbf{x}_k)\sigma_k, \tag{9}$$

$$\mathcal{L}(\mathbf{x}_j)\sigma_j = -D(\mathbf{x}_j)[\mathbf{v}_\infty(\mathbf{x}_j) + \mathcal{B}(\mathbf{x}_j)\mathbf{x}_j] - D(\mathbf{x}_j)\sum_{\substack{k=1\\k\neq j}}^{K} \mathcal{B}(\mathbf{x}_j,\mathbf{x}_k)\mathbf{x}_k + \mathcal{T}(\mathbf{x}_j,\mathbf{x}_k)\sigma_k. \tag{10}$$

The fluid velocity at a point $\mathbf{x}$ away from the vesicle boundaries is computed by

$$\mathbf{v}(\mathbf{x}) = \mathbf{v}_\infty(\mathbf{x}) + \sum_{k=1}^{K} \mathcal{B}(\mathbf{x},\mathbf{x}_k)\mathbf{x}_k + \mathcal{T}(\mathbf{x},\mathbf{x}_k)\sigma_k. \tag{11}$$

[1] By definition, $\mathrm{div}_\gamma = \mathrm{Trace}[\tau\otimes\tau\nabla\mathbf{v}] = (\nabla\mathbf{v})\tau\cdot\tau = \mathbf{v}_s\cdot\mathbf{x}_s$, with $\tau = \mathbf{x}_s$ being the unit tangent vector.

Table 1
The maximum eigenvalue of $\mathcal{J}$ for different spatial discretizations. Asymptotically, we observe that $\lambda_{\min}$ behaves as $\mathcal{O}(-M^3)$.

M	32	64	128	256
$\lambda_{\min}$	−7.25e+01	−7.55e+02	−6.56e+03	−5.40e+04

2.1. Spectral properties

The choice of a computationally-efficient time-stepping scheme depends on whether (7) is stiff or not ([2] p. 50). In this section, we present an approximate stiffness analysis for (7) by constructing the spectrum of its Jacobian.[2] We discuss the case of $\mathbf{v}_\infty = \mathbf{0}$ and $\kappa_B = 1$, in which (7) becomes

$$\dot{\mathbf{x}} = \mathcal{B}(\mathbf{x})\mathbf{x} - \mathcal{M}(\mathbf{x})\mathcal{B}(\mathbf{x})\mathbf{x} \quad \text{or} \quad \dot{\mathbf{x}} = \mathcal{Q}(\mathbf{x})\mathbf{x}.$$

The stiffness of this dynamical system can be characterized by $\min_\lambda Re(\lambda(\mathcal{J}))$ [15], where $\mathcal{J}(\mathbf{x}) = \frac{\partial \mathcal{Q}(\mathbf{x})\mathbf{x}}{\partial \mathbf{x}}$ and λ is an eigenvalue of $\mathcal{Q}$. In particular,

$$\mathcal{J}(\mathbf{x}_0) = \frac{\partial \mathcal{Q}(\mathbf{x})}{\partial \mathbf{x}}\bigg|_{\mathbf{x}=\mathbf{x}_0}[\mathbf{x}_0] + (1 - \mathcal{M}(\mathbf{x}_0))\mathcal{B}(\mathbf{x}_0).$$

We construct $\mathcal{M}$ and $\mathcal{B}$ analytically in the spectral domain and we evaluate $\frac{\partial \mathcal{Q}}{\partial \mathbf{x}}$ numerically. Using the analytic expressions, the symbol of the operators $\mathcal{S}$, $\mathcal{B}$, $\mathcal{L}$, and $\mathcal{M}$ behaves as

for $\mathcal{S}$ (*single-layer potential operator*), $\mathcal{O}(|k|^{-1})$;
for $\mathcal{L}$ (inextensibility operator), $\mathcal{O}(-|k|)$;
for $\mathcal{B}$ (bending force potential operator), $\mathcal{O}(-|k|^3)$;
for $\mathcal{M}$ (stretching operator), $\mathcal{O}(1)$

for large k, where k is the Fourier-mode index. A derivation of these results is presented in Appendix B. While $\mathcal{S}$ is a smoothing operator, $\mathcal{B}$ and $\mathcal{L}$ are ill-conditioned operators, with $\mathcal{B}$ having the higher stiffness.

To account for the additional $\frac{\partial \mathcal{Q}}{\partial \mathbf{x}}$ term, we compute the eigenvalues of $\mathcal{J}$ numerically, by constructing $\mathcal{J}$ using finite differences. For a small parameter ϵ, the i_{th} column of $\mathcal{J}_i$ is given by

$$\mathcal{J}_i = \frac{(\mathcal{Q}(\mathbf{x})\mathbf{x})|_{\mathbf{x}_0+\epsilon\mathbf{e}_i} - (\mathcal{Q}(\mathbf{x})\mathbf{x})|_{\mathbf{x}_0-\epsilon\mathbf{e}_i}}{2\epsilon},$$

where $\mathbf{e}_i$ is the i_{th} coordinate unit vector. The results are given in Table 1. We can see that the overall stiffness is dominated by the stiffness of $\mathcal{B}$. *We conclude that, an implicit time-stepping method is essential for computational efficiency.*

Solving (7) using an implicit scheme requires using a nonlinear solver and calculating rather complicated operator derivatives if exact Jacobians are to be used. An alternative is to use a linearly-implicit scheme [15], with $(1 - \mathcal{M})\mathcal{B}$ being used as an inexact Jacobian in place of $\mathcal{J}$. Stiffness implies a rapid growth of the condition number of $\mathcal{J}$. To efficiently solve linear systems involving $(1 - \mathcal{M})\mathcal{B}$ we use an iterative Krylov method, which requires preconditioning. It turns out that the inverses of the analytically obtained operators on the unit circle yield good results when used as preconditioners for the operators defined on general geometries. The details on the time-stepping and the linear solvers are given in the following section.

3. Numerical scheme

In this section, we present numerical schemes for (1) and discuss extensions to multiple vesicles. First, we discuss the discretization in time and then, the discretization in space along with preconditioning. We conclude with a discussion for the case of multiple vesicles and pseudo-code for the overall algorithm.

3.1. Discretization in time

The existing literature on vesicle simulations is based on explicit-time stepping schemes for (1). Such schemes are expected to suffer from severe stability constraints the size of time step. Here, we discuss two semi-implicit schemes that avoid such stringent constraints.

[2] We have computed the Jacobian numerically for different mesh sizes and we discovered that it is a *non-normal* operator. Thus, the eigenvalues give an incomplete picture of the stability properties. For a more accurate analysis of the numerical stability of the linearized case the pseudospectrum of the Jacobian should be considered [34].

Table 2
The semi-implicit BDF coefficients stated in equation (18) for the second, third, and fourth-order accurate schemes.

Order	β	$\mathbf{x}^o$	$\mathbf{x}^e$
2	$\frac{3}{2}$	$2\mathbf{x}^n - \frac{1}{2}\mathbf{x}^{n-1}$	$2\mathbf{x}^n - \mathbf{x}^{n-1}$
3	$\frac{11}{6}$	$3\mathbf{x}^n - \frac{3}{2}\mathbf{x}^{n-1} + \frac{1}{3}\mathbf{x}^{n-2}$	$3\mathbf{x}^n - 3\mathbf{x}^{n-1} + \mathbf{x}^{n-2}$
4	$\frac{25}{12}$	$4\mathbf{x}^n - 3\mathbf{x}^{n-1} + \frac{4}{3}\mathbf{x}^{n-2} - \frac{1}{4}\mathbf{x}^{n-3}$	$4\mathbf{x}^n - 6\mathbf{x}^{n-1} + 4\mathbf{x}^{n-2} - \mathbf{x}^{n-3}$

There are several ways of tracking the one-dimensional interface. (1) One can track points uniformly-distributed in the arclength [41], which requires resampling at each time step. (2) The shape of the interface is altered only by the normal component of the velocity field. Hence, an arbitrary tangential velocity can be imposed without altering the shape. This is often used to maintain good sampling on the interface [16]. (3) A Lagrangian formulation, in which, we always track the motion of the initial set of material points on the interface.

We have adopted a Lagrangian formulation for several reasons. First, due to the local inextensibility of the interface, point clustering does not happen. If the parametrization at time $t = 0$ is uniform in the arclength, it remains so for all times – up to discretization errors. Second, it simplifies implementation of high-order multistep schemes since it does not require interpolation. Third and most important, methods requiring resampling are far more difficult to extend to surfaces. Next, we describe two variants of a semi-implicit first-order scheme. The first variant has more work per time step but better stability properties.

Scheme I. Let $\triangle t$ be the time step size and let $\mathbf{x}^n(\alpha)$ be a point on the vesicle interface at $n\triangle t$. Then, a first order semi-implicit scheme to compute its position at $(n+1)\triangle t$ is given by

$$\frac{1}{\triangle t}(\mathbf{x}^{n+1} - \mathbf{x}^n) = \mathbf{v}_\infty + \mathcal{B}(\mathbf{x}^n)\mathbf{x}^{n+1} + \mathcal{T}(\mathbf{x}^n)\sigma^{n+1}, \tag{12}$$

$$\mathcal{L}(\mathbf{x}^n)\sigma^{n+1} = -D(\mathbf{x}^n)[\mathbf{v}_\infty + \mathcal{B}(\mathbf{x}^n)\mathbf{x}^{n+1}], \quad \text{where} \tag{13}$$

$$\mathcal{B}(\mathbf{x}^n)\mathbf{x}^{n+1}(\alpha') = -\int_0^{2\pi} G(\mathbf{x}^n(\alpha'), \mathbf{x}^n(\alpha))\left(\frac{1}{|\mathbf{x}^n_\alpha(\alpha)|}\left(\frac{1}{|\mathbf{x}^n_\alpha(\alpha)|}\left(\frac{\mathbf{x}^{n+1}_\alpha(\alpha)}{|\mathbf{x}^n_\alpha(\alpha)|}\right)_\alpha\right)_\alpha\right)_\alpha d\alpha, \quad \text{and} \tag{14}$$

$$\mathcal{T}(\mathbf{x}^n)\sigma^{n+1}(\alpha') = \int_0^{2\pi} G(\mathbf{x}^n(\alpha'), \mathbf{x}^n(\alpha))\left(\sigma^{n+1}(\alpha)\frac{\mathbf{x}^n_\alpha(\alpha)}{|\mathbf{x}^n_\alpha(\alpha)|}\right)_\alpha d\alpha. \tag{15}$$

Since the vesicle is locally-inextensible, the Jacobian $s_\alpha = |\mathbf{x}_\alpha|$ is time-independent; this fact motivates the explicit treatment of $|\mathbf{x}_\alpha|$. Equations (12) and (13) are solved simultaneously for σ^{n+1} and $\mathbf{x}^{n+1}$. We do this by eliminating the equation for σ or working on its Schur complement: if we consider a background velocity field $\mathbf{v}_\infty(\mathbf{x}) = \mathcal{A}\mathbf{x}$, where $\mathcal{A}$ is a constant operator, then each update for $\mathbf{x}^{n+1}$ requires inverting $1 - \Delta t(1 - \mathcal{M})(\mathcal{A} + \mathcal{B})$.

Scheme II. This scheme is inspired by the scheme proposed in [41] for the simulation of flexible fibers in viscous flows. The difference with the first scheme is that we treat the tension differently. That is, we first compute σ^{n+1} by

$$\mathcal{L}(\mathbf{x}^n)\sigma^{n+1} = -D(\mathbf{x}^n)[\mathbf{v}_\infty + \mathcal{B}(\mathbf{x}^n)\mathbf{x}^n]. \tag{16}$$

and then we update $\mathbf{x}^{n+1}$ by

$$\frac{1}{\triangle t}(\mathbf{x}^{n+1} - \mathbf{x}^n) = \mathbf{v}_\infty + \mathcal{B}(\mathbf{x}^n)\mathbf{x}^{n+1} + \mathcal{T}(\mathbf{x}^n)\sigma^{n+1}. \tag{17}$$

At each time step, this scheme requires solving linear systems with the operators $\mathcal{L}$ and $1 - \Delta t\mathcal{B}$. Its main advantage over Scheme I is that the discrete equation for the tension is decoupled from the evolution equation. In our numerical experiments, we have observed that the stability properties of Scheme II are somewhat inferior to that of scheme I for certain flow regimes. Despite this, however, we found Scheme II quite attractive in the case of high-shear flows in which both schemes require time steps of similar sizes for stability. Finally, notice that in both schemes, the dependence of all linear operators on $\mathbf{x}$ is treated explicitly.

High-order schemes. We use the high-order, semi-implicit, backward difference formula (BDF) introduced in [3] and used in [41] for the motion of inextensible filaments in a Stokes flow. A high-order equivalent scheme for (12) can be written as

$$\beta\mathbf{x}^{n+1} - \mathbf{x}^o = \triangle t[\mathbf{v}_\infty + \mathcal{B}(\mathbf{x}^e)\mathbf{x}^{n+1} + \mathcal{T}(\mathbf{x}^e)\sigma^{n+1}], \tag{18}$$

where $\mathbf{x}^e$ is the interfacial position obtained by extrapolation from previous time steps. In Table 2, we list $\beta, \mathbf{x}^o$ and $\mathbf{x}^e$ for second through fourth-order schemes.

We do not have a theoretical analysis for the accuracy or stability of the first- and higher order semi-implicit schemes. Numerical experiments demonstrating the properties of our schemes are presented in Section 5.

3.2. Spatial discretization

We use a Fourier basis to represent the interface. Assuming that the point positions $\mathbf{x}(\alpha)$ are given at M uniformly-distributed points $\{\alpha_k = 2\pi(k-1)/M\}_{k=1}^M$ in the parametric domain, we write

Table 3
The condition number of the discrete operator $\mathcal{L}_M$ defined on the five-armed starfish vesicle shown in Fig. 2. Asymptotically, we observe that the condition number increases as $\mathcal{O}(M)$. Hence, the matrix $\mathcal{L}_M$ is ill-conditioned.

M	64	128	256	512	1024
cond($\mathcal{L}_M$)	3.13e+01	9.19e+01	2.47e+02	5.87e+02	1.31e+03

$$\mathbf{x}(\alpha) = \sum_{k=-M/2}^{M/2-1} \hat{\mathbf{x}}(k)e^{-ik\alpha} \quad \text{and} \quad \mathbf{x}_\alpha = \sum_{k=-M/2}^{M/2-1} (-ik)\hat{\mathbf{x}}(k)e^{-ik\alpha}, \quad \alpha \in [0, 2\pi]. \tag{19}$$

FFTs are used to switch between $\mathbf{x}$ and $\hat{\mathbf{x}}$. The arclength s is given by $s(\alpha) = \int_0^\alpha |\mathbf{x}_\alpha| d\alpha$. Derivatives with respect to the arclength are computed as $\mathbf{x}_s = \frac{\mathbf{x}_\alpha}{s_\alpha}$. Given σ at uniform locations $\{\alpha_k\}_{k=1}^M$, $\mathbf{f}_\sigma$ is computed similarly. Since the vesicle boundary is assumed to be smooth, computing derivatives in this manner yields spectral accuracy.

Quadrature rule. The single-layer potential $\mathcal{S}[\mathbf{f}]$ is computed by the hybrid Gauss–trapezoidal quadrature rules of [1], which were designed to handle logarithmic singularity[3] (Table 8 in [1]). Let $\mathbf{x} \in \gamma$, then we write

$$\mathcal{S}[\mathbf{f}](\mathbf{x}) \approx \sum_{k=1}^{M+m} w_k G(\mathbf{x}, \mathbf{y}(\alpha_k))\mathbf{f}(\mathbf{y}(\alpha_k))|\mathbf{y}_\alpha(\alpha_k)|. \tag{20}$$

Here, M is the number of nodes used in the trapezoidal rule and m is the fixed number of quadrature nodes (determined by the convergence order). These m nodes correct the trapezoidal rule to yield high-order convergence. The convergence order of these quadrature rules is up to 32. Since derivatives are computed spectrally and quadrature rule (20) is governed by the singularity correction. In our implementation, we use the 16th-order correction rule from Table 8 in [1] to calculate the right-hand side of (1).

Fast summation. Direct evaluation (20) at M points on the boundary requires $\mathcal{O}(M^2 + Mm)$ work. This cost can be reduced to $\mathcal{O}(M + Mm)$ by using the FMM. In [43,40], fast summation schemes for summing Stokeslets were proposed based on fast multipole methods for the Poisson problem. Here, we adopt a similar algorithm for (20), which we describe in Appendix D.

3.3. Preconditioners

As discussed, the time-stepping Schemes I and II require solution of systems with the operators $\mathcal{L}, 1 - \Delta t(\mathcal{A} + \mathcal{B})$, and $1 - \Delta t(1 - \mathcal{M})(\mathcal{A} + \mathcal{B})$. Based on our spectral analysis, the corresponding condition numbers should be behave as $\mathcal{O}(M)$, $\mathcal{O}(M^3/N)$, and $\mathcal{O}(M^3/N)$, where M is the number of modes in space, and N is the number of time steps.[4]

We propose low-cost preconditioners for these operators: the inverses of the corresponding operators for the unit circle. For example, the discretized inextensibility constraint has the form

$$\mathcal{L}_M \sigma_M = f_M. \tag{21}$$

where M is the number of discretization points. We have shown in Section 2.1 that the eigenvalues of $\mathcal{L}$ are given by $\lambda_k = \frac{|k|}{4}$, $k \in \mathbb{Z}$. Hence, on a circle, the condition number of $\mathcal{L}_M$ is $\mathcal{O}(M)$, assuming the null space is removed. We conducted numerical experiments to test how good an approximation the $\mathcal{L}$ spectrum on the unit circle is to the spectrum of $\mathcal{L}$ on a general boundary (Table 3). Fig. 2 shows the spectrum of $\mathcal{L}_M$ for different boundary configurations that have the same perimeter. We observe that the spectrum agrees quite well with the spectrum of a circle of same perimeter. This motivates the following preconditioner for (21):

$$P = \mathcal{F}^{-1}\Lambda_c^{-1}\mathcal{F}, \quad \Lambda_c = \text{diag}\left\{\lambda_{-\frac{M}{2}}, \lambda_{-\frac{M}{2}+1}, \ldots, \lambda_{\frac{M}{2}-1}\right\}, \tag{22}$$

where λ_k is the kth-eigenvalue of $\mathcal{L}_M$ defined on unit circle (we set $\lambda_0 = 1$) and $\mathcal{F}$ is the Fourier transform operator i.e., $\mathcal{F}f = \hat{f}$. Since the entries of Λ_c are known (Section 2.1) and FFT can be used for accelerating $\mathcal{F}f$, the cost of applying P is $\mathcal{O}(M \log M)$ per GMRES iteration. Tables 4 and 7 show the effectiveness of P. Without any preconditioner, the number of GMRES iterations increases with M (roughly as $\mathcal{O}(\sqrt{M})$). When using the preconditioner, the number of GMRES iterations for a fixed tolerance remains approximately the same.

Next, we present a preconditioner to solve (12). Let us write (12) as

$$[\mathbf{I} - \triangle t(1 - \mathcal{M})(\mathcal{A} + \mathcal{B})]\mathbf{x}^{n+1} = \mathbf{x}^n, \tag{23}$$

where the operator $\mathcal{A}$ gives the far-field velocity,[5] $\mathbf{v}_\infty = \mathcal{A}\mathbf{x}$.

[3] The quadrature rules of [1] are designed to compute integrals of the form $I(x) = \int_0^1 \phi_1(x) \log x + \phi_2(x)\, dx$. The integral operators that we compute are of the form $I(x) = \int_0^1 \phi_1(x) \log(\psi(x)) + \phi_2(x)\, dx$, where $\psi(x)$ is a smooth function with $\psi(0) = 0$. We can show that we still get the expected order of convergence. For details see Appendix C.

[4] Note that if we had bending only, we would expect an M^4/N behavior; the convolution with the Stokes single-layer potential has a smoothing effect.

[5] For general $\mathbf{v}_\infty$, $\mathcal{A}$ can be defined as $\frac{\partial \mathbf{v}_\infty}{\partial \mathbf{x}}\big|_{\mathbf{x}=\mathbf{x}_n}$.

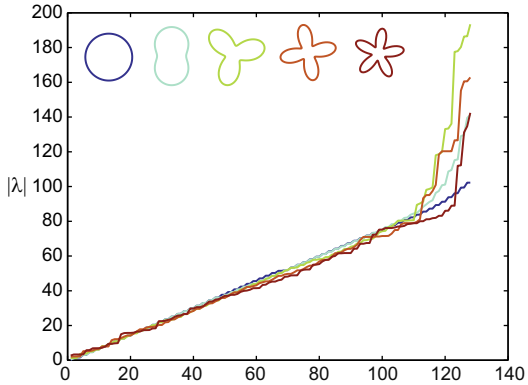

Fig. 2. The spectrum of $\mathcal{L}_M$ with $M = 128$ for different shapes of unit perimeter. The boundaries are shown (not to scale) with corresponding color of their spectrum plot.

Table 4
The number of GMRES iterations required to solve the incompressibility Eq. (21) with and without preconditioning. Here, ϵ is the relative GMRES tolerance and we solve (21). This case corresponds to a vesicle having a five-armed starfish shape (shown in 2). The shear rate is zero.

Preconditioner	None		P	
M	$\epsilon = 10^{-6}$	$\epsilon = 10^{-12}$	$\epsilon = 10^{-6}$	$\epsilon = 10^{-12}$
64	21	35	12	22
128	30	55	13	25
256	41	74	12	28
512	59	102	11	30
1024	91	123	10	28

We construct the operators analytically on the unit circle. The inverses of $\mathcal{B}$ and $\mathcal{L}$ are known analytically since they are diagonal; the unit circle representations of $\mathcal{A}$ and $\mathcal{M}$ are sparse with small bandwidth (ten for $\mathcal{M}$) and their factorizations can be computed and applied in $\mathcal{O}(M)$ time. Again, the cost of applying the preconditioner on (23) is $\mathcal{O}(M \log M)$ per GMRES iteration. In our numerical experiments, we have found that using $\mathcal{M}$ does not result in significant improvements and we have not included it in our implementation. We present results on the performance of this preconditioner in Tables 7 and 8.

3.4. Multiple vesicles

The semi-implicit schemes can be extended in a straightforward manner to deal with multiple vesicles. We restrict our discussion to the extension of Scheme II since empirically it has similar stability properties with Scheme I but is less expensive computationally (see Section 5). We discretize each of the K vesicles with M points. The spatial derivatives and convolutions are computed as described in Section 3.2. We discretize (9), (10) by first solving for $\{\sigma_j^n\}_{j=1}^K$

$$\mathcal{L}(\mathbf{x}_j^{n-1})\sigma_j^n + \mathcal{D}(\mathbf{x}_j^{n-1}) \sum_{\substack{k=1 \\ k\neq j}}^{K} \mathcal{T}(\mathbf{x}_j^{n-1}, \mathbf{x}_k^{n-1})\sigma_k^n = -\mathcal{D}(\mathbf{x}_j^{n-1})\mathbf{v}_\infty - \mathcal{D}(\mathbf{x}_j^{n-1}) \sum_{k=1}^{K} \mathcal{B}(\mathbf{x}_j^{n-1}, \mathbf{x}_k^{n-1})\mathbf{x}_k^{n-1}, \tag{24}$$

and then updating the positions $\{\mathbf{x}^{n+1}\}_{j=1}^K$

$$\frac{1}{\triangle t}(\mathbf{x}_j^{n+1} - \mathbf{x}_j^n) = \mathbf{v}_\infty + \mathcal{B}(\mathbf{x}_j^n)\mathbf{x}_j^{n+1} + \mathcal{T}(\mathbf{x}_j^n)\sigma_j^n + \sum_{\substack{k=1 \\ k\neq j}}^{K} \mathcal{B}(\mathbf{x}_j^n, \mathbf{x}_k^n)\mathbf{x}_k^n + \mathcal{T}(\mathbf{x}_j^n, \mathbf{x}_k^n)\sigma_k^n. \tag{25}$$

Note that the inter-vesicle coupling is treated implicitly in the tension calculation and explicitly in the force calculation.[6] We construct a block diagonal preconditioner (P^b) for (24). Each of its block diagonal entry is set to P, defined in (22). We can solve (25) on each vesicle separately because of the explicit treatment of the interactions and can use the single vesicle preconditioner. When the suspension is dilute (i.e., the distance between vesicles is typically significantly more than the vesicle size), we found that (25), (24) would still have the same stability constraint as the one for single vesicle. We use the trapezoidal rule to compute the interaction component $\mathcal{S}_k[\mathbf{f}_\sigma + \mathbf{f}_b](\mathbf{x}_j)$ in (25) and (24).

[6] We conducted numerical experiments in which inter-vesicle tension forces were treated explicitly in (24), but we discovered that such an approach introduces significant violations of the inextensibility constraint.

4. Summary

In this section we summarize the algorithm for multiple vesicles. The input includes the positions of M points per vesicle for K vesicles, the material parameters κ_B and μ, the background velocity $\mathbf{v}_\infty$ (for shear flows parametrized by the shear rate), the time horizon T, and the number of time steps N. The output is the interfacial tension and position of all the vesicles at $\{k\triangle t\}_{k=1}^{N}$.

Algorithm 1. Time-marching scheme II for multiple vesicles in shear flow

```
for n = 1 : N − 1 do
  σ = ComputeTension(x)                                   givenpositions x, computetension
  for k = 1 to K do
    f_k^b = −κ_B ComputeDerivative(x_k, x_k, 4)           traction jump due to bending
    t_k = ComputeDerivative(x_k, x_k, 1)                  tangentvector
    f_k^σ = ComputeDerivative(σ_k t_k, x_k, 1)            traction jump due to tension
  end for
  F = v_∞(x) + ComputeInteraction(f^σ + f^b, x)
  for k = 1 to K do
    F_k = F_k − ComputeInteraction(f_k^b, x_k)            subtract the self-interaction due to bending
    Solve: (I − △t B(x_k)) y_k = x_k + △t F_k             using preconditioned GMRES
    x_k := y_k
  end for
end for
```

Algorithm 2. ComputeTension $(\mathbf{x})$

```
Given positions, computes the
tensions
f_k^b = −κ_B ComputeDerivative(x_k, x_k, 4),   k = 1, ..., K     traction jump due to bending
F = ComputeInteraction(f^b, x)                                   velocity field due to bending
t_k = ComputeDerivative(x_k, x_k, 1),   k = 1, ... K             tangentvector
F_k = −t_k · [F_k + v_∞(x_k)],   k = 1, ..., K                   surfacedivergence
Solve for σ : TensionMatVec(σ, x) = F                            using preconditioned GMRES
```

Algorithm 3. TensionMatVec$(\sigma, \mathbf{x})$

```
Given tensions and positions,
computes the left hand side of
(24)
for k = 1 to K do
  t_k = ComputeDerivative(x_k, x_k, 1)                           tangentvector
  f_k^σ = ComputeDerivative(σ_k t_k, x_k, 1)                     traction jump due to tension
end for
F = ComputeInteraction(f^σ, x)
return F_k = t_k · F_k,   k = 1, ... K
```

Algorithm 4. ComputeDerivative $(\mathbf{f}, \mathbf{x}, m)$

```
Givenjumps f across the vesicle boundaries x, computes Σ_{k=1}^K S_k[f_k](x)
φ_1 = Σ_{k=1}^K ∫_{γ_k} log|x − y| f_1(y) ds(y)                  using trapezoidal rule and FMM
φ_2 = Σ_{k=1}^K ∫_{γ_k} log|x − y| f_2(y) ds(y)                  using trapezoidal rule and FMM
φ_3 = Σ_{k=1}^K ∫_{γ_k} log|x − y| [f_1(y) y_1 + f_2(y) y_2] ds(y)   using trapezoidal rule
and FMM
F = {φ_1, φ_2} + x_1 ∇_x φ_1 + x_2 ∇_x φ_2 − ∇_x φ_3
Correct the trapezoidal rule to compute self-interactions
Get the nodes and weights {α_i^c, w_i}_{i=1}^m corresponding to order q, that correct the trapezoidal rule [1].
for k = 1 to K do
  for j = 1 to M do
    for i = 1 to m do
      Compute f_k, x_k at α_j + α_i^c                            using spline interpolation/nonuniform FFT
      F_kj = F_kj + w_i G(x_kj, x_k(α_j + α_i^c)) f_k(α_j + α_i^c)   add the correction
    end for
  end for
end for
```

Algorithm 5. ComputeDerivative $(\mathbf{f}, \mathbf{x}, m)$

Computes the mth derivative of a vector field $\mathbf{f}$ *with respect to the arclength* s. $\mathbf{x}$ *is the position of the boundary.*

$c = \left[-\frac{M}{2}, -\frac{M}{2}+1, \ldots, \frac{M}{2}-1\right]$ *coefficientvector*

Set $\mathbf{F} = \mathbf{f}$ *initialization*

$|\mathbf{x}_\alpha| = \sqrt{[\mathtt{IFFT}(ic\mathtt{FFT}(x_1))]^2 + [\mathtt{IFFT}(ic\mathtt{FFT}(x_2))]^2}$ *Jacobian*

for l to m do

$\mathbf{F} = \left\{\frac{\mathtt{IFFT}(ic\mathtt{FFT}(F_1))}{|\mathbf{x}_\alpha|}, \frac{\mathtt{IFFT}(ic\mathtt{FFT}(F_2))}{|\mathbf{x}_\alpha|}\right\}$ *differentiateonce*

end for

return F

Complexity analysis. The main steps of the algorithm are the solution of coupled system of equations for the tensions (24) and the position update using (25). Since we use FFTs to compute the derivatives, the complexity of computing $\mathbf{f}^b$ and $\mathbf{f}^\sigma$ is $\mathcal{O}(M \log M)$ per vesicle. Using FMM, the complexity of evaluating the single-layer potentials at the MK discrete points is $\mathcal{O}(MK)$. Therefore, each GMRES iteration to solve (24) and similarly to solve (25) over all the vesicles, requires $\mathcal{O}(MK \log M)$ work. In Section 5, we demonstrate numerically that the number of iterations are nearly independent of the problem size (Tables 12, 8). Hence, the cost of solving the linear system of equations is $\mathcal{O}(MK \log M)$ per time step.

4.0.1. Selection of discretization parameters and their effect on complexity

We have not discussed the selection of the spatial discretization M, and the temporal discretization N and q. Currently, we do not have an adaptive scheme for the selection of the spatial and temporal accuracy. For dilute suspensions, asymptotically, the errors are dominated by the temporal discretization since we use spectral discretization (combined with a very high-order scheme for the singularities) in space. In our experiments, 64–128 spatial modes are typically sufficient to fully resolve the shapes of the vesicles in the shear rate regimes we have examined. For concentrated suspensions, adaptive schemes combined with a posteriori estimates are necessary; the complexity analysis for such methods is an open problem.

5. Results and discussion

In this section, we present results on the convergence, stability, and algorithmic complexity of the proposed methods.

5.1. Single vesicle

We present two test cases. In the first case, we consider a vesicle suspended in a simple shear flow. If the interior and exterior of the vesicle are filled with the same fluid, the vesicle undergoes a tank-treading motion at its equilibrium configuration. This was established by several authors through numerical simulations [44,20] and experiment [8,18]. In Fig. 3, we simulate the motion of an arbitrary shaped vesicle, suspended in a simple shear flow and in Fig. 4 we show the streamlines around the vesicle. The orientation angle and the tank-treading frequency of the vesicle were shown to be independent of the shear rate in [20]. We verify this result in Fig. 5.

We study the stability and convergence properties of the proposed numerical time-stepping schemes I and II, using the single vesicle shear flow as a test case.

Stability. In Table 5, we list the maximum allowable time-step, Δt, for different time-stepping schemes. To determine Δt, we start from an arbitrarily large time step Δt_0 and if the numerical simulation is stable, we set $\Delta t = \Delta t_0$. Otherwise, we reduce Δt_0 by half and repeat the experiment. All calculations are run until the vesicle shape reaches steady state. In the case of an explicit scheme, we can observe that asymptotically $\Delta t \propto M^{-p}$, where p is greater than three. Hence, irrespective of the accuracy requirements, we have to take a smaller time step to satisfy the stability criterion. The time step required for stability is asymptotically mesh-independent for both semi-implicit schemes. The maximum time step required for stability depends on the shear rate. In low-shear regimes, Scheme I allows for larger time steps and, despite its higher cost per iter-

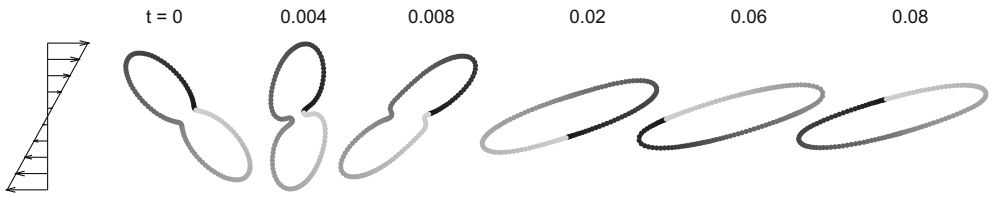

Fig. 3. Snapshots of a vesicle suspended in a shear flow. In this experiment, the reduced area of the vesicle is 0.5, the shear rate $\chi = 250$ and t is the nondimensional time. Lagrangian particles on the vesicle membrane are grayscale colored. We can observe that once the vesicle reaches an equilibrium shape, its interface undergoes a tangential motion, commonly known as the tank-treading motion.

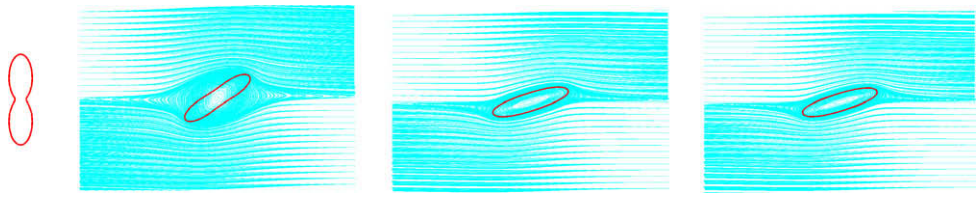

Fig. 4. Snapshots of streamlines of the velocity field around the vesicle. The streamline pattern is in agreement with [44]. Notice that at equilibrium a vortex is formed in the interior of the vesicle and the fluid surrounding the interface undergoes a tangential motion.

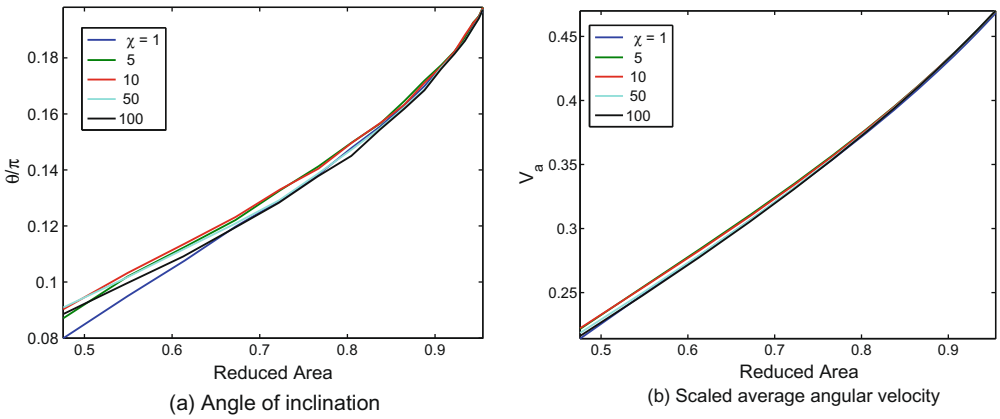

Fig. 5. The angle of inclinations and the angular velocities of vesicles with different reduced areas suspended in shear flow for different shear rates. We set the scaled velocity to $V_a = \frac{\overline{V}}{R_0\gamma}$, where $\overline{V}$ is velocity averaged over all of the marker particles at the equilibrium configuration. We observe that both θ and V_a are nearly independent of the shear rate. This result is in agreement with [20].

Table 5
Stable time step sizes for the first-order explicit and semi-implicit schemes. The initial vesicle configuration is shown in Fig. 3. We observe that the explicit scheme has severe stability constraints on the time step size. The semi-implicit schemes, on the other hand, do not suffer from such high-order constraints. Their stable time step size is inversely proportional to the shear rate. For lower shear rates, Scheme I allows for larger time steps relative to Scheme II and is preferable. However, the lower computational cost per time step of Scheme II makes it the method of choice for higher-shear rates.

M	Explicit scheme			Semi-implicit Scheme I			Semi-implicit Scheme II		
	$\chi = 0$	10	100	0	10	100	0	10	100
32	3.90e−03	7.81e−03	9.76e−04	∞	1.56e−02	9.76e−04	3.12e−02	1.56e−02	9.76e−04
64	9.76e−04	9.76e−04	4.88e−04	∞	1.56e−02	9.76e−04	1.56e−02	7.81e−03	9.76e−04
128	6.10e−05	3.05e−05	6.10e−05	∞	1.56e−02	9.76e−04	7.81e−03	7.81e−03	9.76e−04
256	3.81e−06	3.81e−06	3.81e−06	∞	1.56e−02	9.76e−04	7.81e−03	7.81e−03	9.76e−04
512	2.38e−07	2.38e−07	2.38e−07	∞	1.56e−02	9.76e−04	7.81e−03	7.81e−03	9.76e−04

ation, it is preferable to Scheme II. For higher-shear regimes, the stable time steps are roughly the same so, since its cost per iteration is smaller, Scheme II is preferable.

Convergence and complexity. The fluid enclosed in the vesicle is incompressible and the interface is locally-inextensible. Therefore, the enclosed area and the perimeter of the vesicle should remain constant throughout the simulation. We report the errors in preserving the perimeter and area of the vesicle (Fig. 3) in Table 6.

For the same setup and different shear rates, we report the errors in the positions of marker points on the vesicle, measured at the end of the simulation, in Fig. 6. While we observe high-order convergence in the case of high-shear rate flows, we found that it is not the case for low-shear rate flows. Such erratic convergence behavior was also observed for the same semi-implicit BDF schemes in Tornberg and Shelley [41] (p. 30), and in Kropinski [23] (p. 498).

The performance of the spectral preconditioners in accelerating the solution of the linear systems that appear in the time-stepping algorithm is presented in Tables 7 and 8. The number of iterations are almost mesh-independent in the case of Scheme II (Table 7). In Scheme I, the preconditioner significantly reduces the number of iterations (Table 8) but does not eliminate the mesh-dependence of the total number of GMRES iterations entirely.

For the second test case, we consider a vesicle suspended freely in a stationary fluid. In the absence of inextensibility constraint, the equilibrium shape is a circle. However, since the interface is locally inextensible, equilibrium shapes can be

Table 6
Errors in the length and area of the vesicle measured at end of the simulation shown in Fig. 3. q is the convergence order of the semi-implicit time scheme, M is the total number of discretization points on the vesicle, and $\triangle t = \frac{0.004}{M}$.

M	$\lvert L - L_f \rvert / L$				$\lvert A - A_f \rvert / A$			
	$q = 1$	2	3	4	$q = 1$	2	3	4
32	8.19e−02	1.39e−04	9.27e−04	5.43e−04	3.70e−02	8.48e−04	3.75e−04	4.34e−04
64	4.28e−02	1.22e−04	7.88e−05	1.32e−05	1.96e−02	9.12e−05	2.29e−05	1.11e−05
128	2.13e−02	2.66e−05	8.96e−06	5.27e−07	1.04e−02	2.44e−05	1.59e−06	2.36e−08
256	1.06e−02	5.67e−06	1.17e−06	1.49e−07	5.36e−03	5.95e−06	2.05e−07	5.10e−09

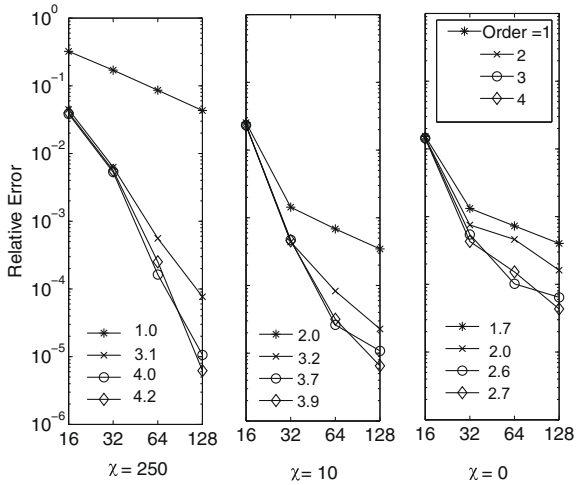

Fig. 6. Max-norm errors in computing the position $\mathbf{x}(t_{\text{eq}})$ of the vesicle at $t = t_{\text{eq}}$, the time to reach an equilibrium shape. In the experiments with different shear rates, the initial shape of the vesicle is set to the shape corresponding to $t = 0$ in Fig. 3. The exact solution is computed by a finer discretization in space and time ($M = 256, \triangle t = \frac{t_{\text{eq}}}{10M}$). For lower shear rates, we do not observe high-order convergence.

Table 7
Performance of the preconditioners described in Section 3.3 for solving the position update Eq. (12) (outer) and the inextensibility constraint (13) (inner), in the simulation shown in Fig. 3. We report the maximum number of GMRES iterations over all the time steps. The outer GMRES tolerance is set to 10^{-8} and the inner is set to 10^{-12}.

Preconditioner	None				Spectral			
M	$\triangle t$ = 1.90e−06		9.76e−04		1.90e−06		9.76e−04	
	Outer	Inner	Outer	Inner	Outer	Inner	Outer	Inner
32	3	17	9	17	3	11	10	14
64	3	31	21	35	3	15	23	19
128	5	48	53	61	5	17	40	21
256	10	79	131	93	9	19	53	24
512	28	119	323	149	14	22	70	24

Table 8
The number of GMRES iterations required to solve the discrete evolution equation of semi-implicit scheme II (17). For each value of χ, $\triangle t$ is chosen to be the maximum allowable listed in Table 5. The GMRES tolerance is set to 10^{-9}.

Preconditioner	None			Spectral		
M	$\chi = 0$	10	100	0	10	100
32	16	20	11	11	14	10
64	43	48	25	14	16	16
128	111	121	65	14	16	21
256	282	300	163	14	16	23
512	701	731	442	14	16	23
1024	1656	1699	1095	14	16	23

different from circles. In [36], it was shown that these equilibrium shapes depend only on the reduced area of the vesicle and are independent of the material properties of the interface and surrounding fluid. In Fig. 7, we plot the relaxation to equilibrium of an arbitrary shaped vesicle of reduced area 0.33. In this experiment, we could take four orders of magnitude bigger time step than a fully explicit scheme.

5.2. Multiple vesicles

Fig. 8 shows the streamline patterns for the fluid surrounding two and three vesicles. We let the vesicles relax to equilibrium, and capture the velocity profile of the fluid at the nondimensional time $t = 0.1$. Fig. 9 shows the streamlines when the vesicles are suspended in a simple shear flow. We report the convergence results for the four-vesicle simulation (Fig. 9) in Table 9. For the same simulation, we list the maximum number of GMRES iterations to solve the inextensibility constraint

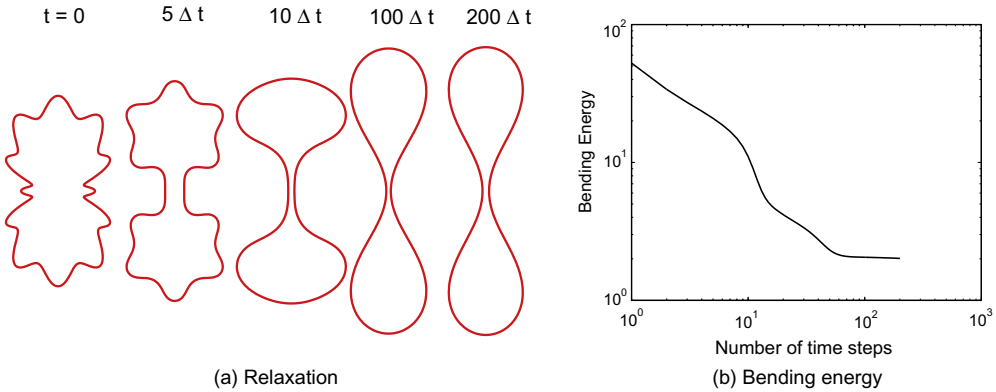

Fig. 7. Snapshots of a freely suspended vesicle relaxing to equilibrium shape. In Figure (b), we plot the bending energy as a function of the number of time steps. We have used $M = 256$ points on the boundary. For this simulation, the explicit scheme requires approximately 10^6 time steps to reach equilibrium.

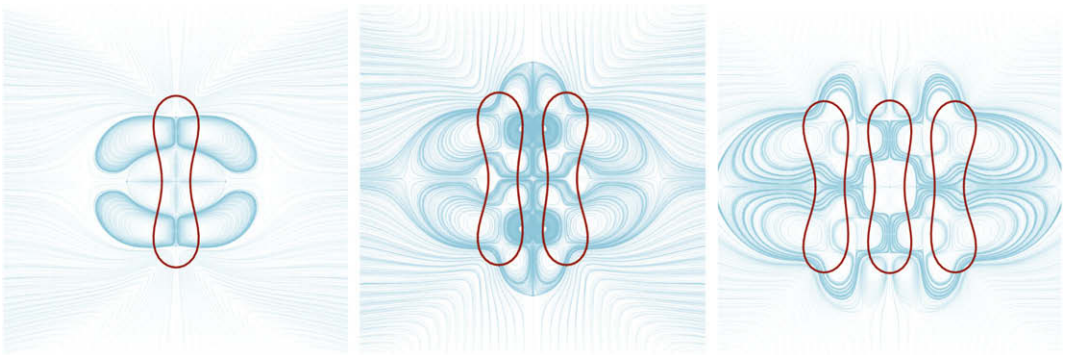

Fig. 8. Streamline patterns in the bulk and enclosed fluid when the vesicles are freely suspended ($\chi = 0$) at $t = 0.1$. The background fluid velocity is computed on a 128^2 Cartesian-grid. The trapezoidal rule is used to discretize the equation for the velocity (11).

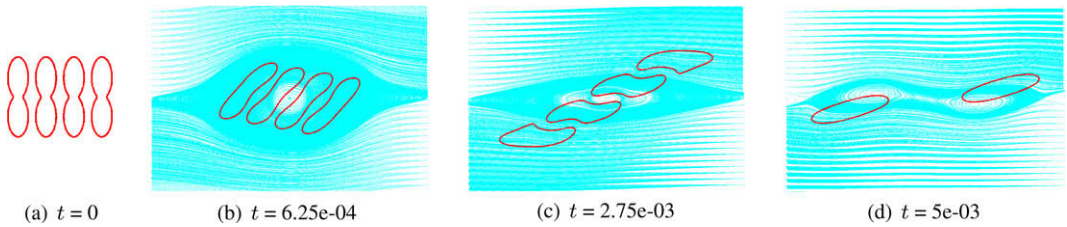

Fig. 9. Streamlines in the bulk and enclosed fluid multiple vesicles are suspended in simple shear flow ($\chi = 250$). Initially, the collection of vesicles behaves like a single vesicle; for instance, compare the streamlines at $t = 6.25e-04$ with Fig. 4). Once they are separated, each vesicle undergoes tangential motion by itself.

Table 9
Relative errors measured at t = 6.25e−04, for the simulation shown in Fig. 9. M is the number of discretization points per each of the four-vesicles and q is the convergence order of the scheme II.

M	$\lvert L-L_f\rvert/L$			$\lvert A-A_f\rvert/A$		
	$q=1$	2	3	1	2	3
32	5.36e−004	4.98e−004	4.98e−004	1.69e−004	3.50e−004	3.50e−004
64	4.28e−005	2.27e−005	2.28e−005	1.18e−004	8.67e−006	6.48e−006
128	1.01e−005	6.27e−008	2.73e−008	6.82e−005	2.20e−006	9.73e−007
256	5.02e−006	1.76e−008	4.72e−009	3.42e−005	6.60e−007	2.25e−007

Table 10
The number of GMRES iterations to solve the inextensibility constraint (24) in the multiple vesicle case shown in Fig. 9(a). ϵ is the GMRES tolerance and M is the number of discretization points per vesicle.

Preconditioner	None		Spectral	
M	$\epsilon=10^{-6}$	$\epsilon=10^{-12}$	$\epsilon=10^{-6}$	$\epsilon=10^{-12}$
64	47	83	12	27
128	66	123	10	26
256	97	183	10	26
512	142	270	10	24
1024	209	402	10	24

Table 11
CPU timings (in seconds) per time step as the number of vesicles is increased. The number of spatial discretization points per vesicle is fixed at $M=64$ and the number of vesicles is increased from $K=16$ to $K=4096$. The computational times scale linearly with the problem size. All calculations were done in MATLAB (linked to external libraries for the FFT and FMM).

$64K$	1024	4096	16384	65536	262144
With inextensibility	2.3	9.3	35.5	134	537
Without	0.9	2.2	8.9	36.9	164

Table 12
The number of GMRES iterations required to to solve the inextensibility constraint on multiple vesicles (24). In this experiment, $K=\frac{M}{4}$ and the relative GMRES tolerance is 10^{-6}. In this experiment, the K vesicles are aligned along the x_1-axis (similar to Fig. 9(a)). Clearly, the number of iterations do not grow with the problem size.

MK	256	1024	4096	16384	65536
Iterations	14	11	11	10	10

(24), over all the time steps, in Table 10. Notice that preconditioning significantly reduces the number of iterations and provides mesh-independent convergence.

In integral equation based methods, computing the interaction forces between the vesicles tends to be the dominant part of the computational cost at every time step. If there are K vesicles, the interactions grow as $\mathcal{O}(K^2)$. Using the FMM, the computational cost of our scheme scales linearly with K. We report CPU timings per time step to solve the discretized governing equations in Table 11. Another expensive step of our scheme is computing the tensions, given the interfacial positions of the

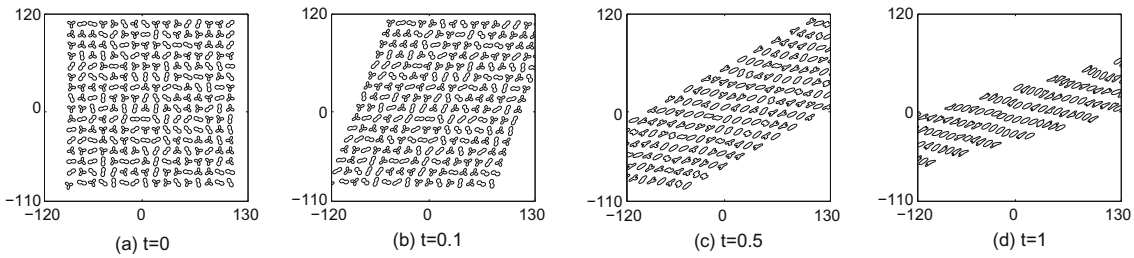

Fig. 10. Simulation of multiple vesicles ($K=256$) suspended in simple shear flow ($\chi=150$).

K vesicles, by solving (24) through iterative methods. Using FFTs and FMM, the cost per iteration is $\mathcal{O}(MK\log M)$, which is work-optimal up to a logarithmic factor. In our numerical experiments, we found that the spectral preconditioner performs very well and yields mesh-independent convergence, see Table 12. Hence, the overall cost to solve for the tensions would be $\mathcal{O}(MK\log M)$.

As a result of incorporating these fast algorithms, we are able to simulate the dynamics of large number of interacting vesicles. We show simulations of interacting vesicles suspended in linear shear flow in Fig. 10 and quadratic flow in Fig. 1.

6. Conclusions and future work

We have presented two semi-implicit schemes to simulate the motion of inextensible vesicles suspended in a viscous fluid. Unlike a fully explicit scheme, these schemes do not exhibit a mesh-dependent high-order stability constraint on the time step size. The only stability restriction on is that the time step must be inversely proportional to the shear rate. For several important cases, we have numerically demonstrated that four orders of magnitude bigger time step can be taken when compared to a fully explicit scheme, even for a modest (32) number of spatial discretization points. Our schemes exhibit high-order accuracy in space and time. We have presented efficient low-cost preconditioners to solve the discrete evolution equations by iterative solvers. We incorporated FMM to compute the interaction forces in a suspension of large number of vesicles. We included analytical results for the unit circle case, and we conducted numerical studies that confirm our convergence and complexity estimates.

The vesicle–vesicle interaction forces are computed using the trapezoidal rule, which yields spectral accuracy for dilute suspension of vesicles. However, when two vesicles come closer, we need to modify the quadrature rules to compute the interaction forces because of the logarithmic singularity of the Stokes kernel. To resolve this issue, we plan to explore correction methods like the one suggested in [7]. One additional extension is to circumvent the time stability dependence on the shear rate. Such an algorithm however, would require nonlinear solvers and contact detection algorithms that fully couple the vesicle position updates.

Our long-term goal is to conduct simulations of deformable incompressible vesicles in three-dimensions. We plan to built on our past work on fast Stokes solvers in complex geometries [42] that includes accurate quadratures, surface parameterization, and FMM. Nevertheless, computing high-order derivatives in high-accuracy will be challenging; and unlike the 2D case, the incompressibility constraint does not prevent mesh distortion: accurate tracking the moving interface will require additional work. Furthermore, the bending and stretching operators have more complicated form and coming up with effective linearizations and preconditioners will be more challenging.

Acknowledgments

We would like to thank Leslie Greengard for providing a MATLAB-compatible fast multipole method implementation.

Appendix A. Variational formulation

For completeness, we present the derivations for the expressions for $\mathbf{f}_b$ and $\mathbf{f}_\sigma$. Without loss of generality, let $\kappa_B = 1$ and let the interface perimeter be equal to one. The bending energy is defined by $\varepsilon(\mathbf{x}) = \frac{1}{2}\int_0^1 \kappa^2 ds$. We introduce a perturbation $\delta\mathbf{x}$ and we define $\bar{\mathbf{x}}$ to be the perturbed interface. We use a Lagrangian parameterization in which the perturbed boundary is a function of the arclength 's' defined on the original configuration. We introduce the following notation: $\bar{\mathbf{x}}(s) = \mathbf{x}(s) + \epsilon\mathbf{y}(s)$ and $\mathbf{y} = u\mathbf{t} + v\mathbf{n}$, that is, u and v are the tangential and normal perturbations. Assuming the normal to the boundary is pointing outwards, we can derive the following relations:

$$\bar{\mathbf{x}}_s = \mathbf{x}_s + \epsilon\mathbf{y}_s, \quad \bar{\mathbf{x}}_{ss} = -\kappa\mathbf{n} + \epsilon\mathbf{y}_{ss}, \tag{26}$$

$$\mathbf{y}_s = (u_s + v\kappa)\mathbf{t} + (v_s - u\kappa)\mathbf{n}, \quad \mathbf{y}_{ss} = (u_{ss} + 2\kappa v_s + \kappa_s v - \kappa^2 u)\mathbf{t} + (v_{ss} - 2\kappa u_s \kappa^2 v - \kappa_s u)\mathbf{n}. \tag{27}$$

Let $\bar{s}$ be the arclength parameter in the deformed configuration. Then, we have

$$\bar{s}_s = |\bar{\mathbf{x}}_s| = \sqrt{1 + 2\epsilon(\mathbf{t}\cdot\mathbf{y}_s)} \approx 1 + \epsilon(u_s + v\kappa). \tag{28}$$

The bending energy in the deformed state is given by $\bar{\varepsilon} = \frac{1}{2}\int_0^1 \bar{\kappa}^2|\bar{\mathbf{x}}_s|\,ds$. The curvature $(\bar{\kappa})$ of the deformed boundary is given by $\bar{\kappa} = \frac{\sqrt{\bar{\mathbf{x}}_{ss}^2\bar{\mathbf{x}}_s^2 - (\bar{\mathbf{x}}_s\cdot\bar{\mathbf{x}}_{ss})^2}}{|\bar{\mathbf{x}}_s|^3}$. Using (26), (27) and neglecting the higher order terms in ϵ, we get

$$\bar{\kappa}^2 = \kappa^2 - 2\epsilon(v_{ss} + \kappa^2 v - \kappa_u). \tag{29}$$

The variation in the bending energy is computed as,

$$\begin{aligned}\delta\varepsilon &= \frac{1}{2}\int_0^1 (\bar{\kappa}^2|\bar{\mathbf{x}}_s| - \kappa^2)\,ds = \frac{\epsilon}{2}\int_0^1 (\kappa^2 u_s + 2\kappa\kappa_s u - \kappa^3 v - 2\kappa v_s s)ds \\ &= \frac{\epsilon}{2}\int_0^1 (-2\kappa\kappa_s u + 2\kappa\kappa_s u - \kappa^3 v - 2k_{ss} v)\,ds \text{ and integrating by parts } \delta\varepsilon = -\int_0^1 \left(\kappa_{ss} + \frac{\kappa^3}{2}\right)\mathbf{n}\cdot\delta\mathbf{x}\,ds.\end{aligned} \tag{30}$$

Hence, $\mathbf{f}_b = -\frac{\delta\varepsilon}{\delta\mathbf{x}} = (\kappa_{ss} + \frac{\kappa^2}{2})\mathbf{n}$. However, if the vesicle is locally-inextensible, other equivalent forms of $\mathbf{f}_b$ exist. The local inextensibility constraint can be enforced by requiring that $|\bar{\mathbf{x}}_s| = 1$. From Eq. (28), we can infer that $u_s + \kappa v = 0$. Consider an interfacial force of the form $\mathbf{f} = (h(s)\mathbf{x}_s)_s$, where $h(s)$ is a scalar function. The energy due to this force is given by $\delta\varepsilon = \int_0^1 (h\mathbf{x}_s)_s \cdot \delta\mathbf{x}\, ds$. Integrating by parts and substituting for $\delta\mathbf{x}$, we get

$$\delta\varepsilon = \frac{1}{2}\int_0^1 (h\mathbf{x}_s)_s \cdot \delta\mathbf{x}\, ds = -\frac{1}{2}\int_0^1 h\mathbf{x}_s \cdot (\delta\mathbf{x})_s\, ds. \tag{31}$$

From Eq. (27), the tangential component of $(\delta\mathbf{x})_s$ is $u_s + \kappa v$, which vanishes for a locally-inextensible vesicle. Therefore, interfacial forces of the form $(h\mathbf{x}_s)_s$ will not do any work and hence such forces can be added to $\mathbf{f}_b$. We have set the bending force as $-\mathbf{x}_{ssss}$, which is obtained by adding $\frac{(3\kappa^2\mathbf{x}_s)_s}{2}$ to $\mathbf{f}_b$. In [6], the bending force $\kappa_{ss}\mathbf{n} + \kappa\kappa_s\mathbf{t}$ is obtained by adding $-\frac{(\kappa^2\mathbf{x}_s)_s}{2}$ to $\mathbf{f}_b$.

Appendix B. Analytic construction on the unit circle

Here we construct analytic expressions for the operators $\mathcal{S}, \mathcal{L}, \mathcal{B}$, and $\mathcal{M}$ defined in in Section 2.

Stokes single-layer potential operator ($\mathcal{S}$). On the unit circle, $\mathbf{x}$ and $\mathbf{y}$ used in Eq. (4) can be written as $[\cos s, \sin s]$ and $[\cos t, \sin t]$ respectively. Then, the components of the Stokes kernel can be expanded as follows:

$$\log\rho = \frac{1}{2}\log(2 - 2\cos(s-t)) = -\sum_{|k|>0}\frac{1}{2|k|}e^{ik(t-s)}, \quad \text{and} \tag{32}$$

$$\frac{\mathbf{r}\otimes\mathbf{r}}{\rho^2} = -\frac{1}{2}\begin{bmatrix}\cos(s+t)-1 & \sin(s+t)\\ \sin(s+t) & -1-\cos(s+t)\end{bmatrix}. \tag{33}$$

Let $\mathcal{S}_1[\mathbf{f}] = \int_0^{2\pi}\log\rho\mathbf{I}\mathbf{f}\,dt$ and $\mathcal{S}_2[\mathbf{f}] = \int_0^{2\pi}\frac{\mathbf{r}\otimes\mathbf{r}}{\rho^2}\mathbf{f}\,dt$, so that $\mathcal{S}[\mathbf{f}] = \mathcal{S}_1[\mathbf{f}] + \mathcal{S}_2[\mathbf{f}]$. The operator $\mathcal{S}_2$ acts only on the first three low-frequency components and zeros all the other components, therefore, we restrict our attention to $\mathcal{S}_1[\mathbf{f}]$.

$$\mathcal{S}_1[\mathbf{f}] = \sum_{|k|>0}\frac{1}{8\pi|k|}e^{iks}\int_0^{2\pi}\begin{bmatrix}e^{-ikt} & 0\\ 0 & e^{-ikt}\end{bmatrix}\sum_{m\in\mathbb{Z}}\begin{bmatrix}\hat{v}_{1m}\\ \hat{v}_{2m}\end{bmatrix}e^{imt}\,dt, \tag{34}$$

$$= \sum_{|k|>0}\frac{1}{4|k|}e^{iks}\begin{bmatrix}\hat{v}_{1k}\\ \hat{v}_{2k}\end{bmatrix}. \tag{35}$$

where $\hat{\mathbf{v}}_m = (\hat{v}_{1m}, \hat{v}_{2m})$, $m \in \mathbb{Z}$ are the Fourier components of $\mathbf{f}$.

Hence, $\mathcal{S}_1[\mathbf{f}]$ is diagonalizable with the Fourier basis and the eigenvalues are given by $\left\{\frac{1}{4|k|}\right\}_{k\in\mathbb{Z}}$.

Bending operator ($\mathcal{B}$). We write $\mathcal{B} = \mathcal{B}_1 + \mathcal{B}_2$, where $\mathcal{B}_1 = \mathcal{S}_1[\mathbf{f}_b]$ and $\mathcal{B}_2 = \mathcal{S}_2[\mathbf{f}_b]$. By substituting $\mathbf{f} = -\mathbf{x}_{ssss}$ in (35), we obtain the spectrum of $\mathcal{B}_1$ as $\Lambda[\mathcal{B}_1] = -\left\{\frac{k^4}{4|k|}\right\}_{k\in\mathbb{Z}}$.

Inextensibility operator ($\mathcal{L}$). Expanding σ with Fourierbasis, we get

$$(\sigma\mathbf{x}_s)_s = \sum_{k=-\infty}^{\infty}\hat{\sigma}_k e^{iks}\begin{bmatrix}-ik\sin s - \cos s\\ ik\cos s - \sin s\end{bmatrix} = -\hat{\sigma}_0\begin{bmatrix}\cos s\\ \sin s\end{bmatrix} + \sum_{|k|>0}\frac{k}{2}e^{iks}\begin{bmatrix}\hat{\sigma}_{k+1} - \hat{\sigma}_{k-1}\\ i(\hat{\sigma}_{k+1} + \hat{\sigma}_{k-1})\end{bmatrix}. \tag{36}$$

We write $\mathcal{L} = \mathcal{L}_1 + \mathcal{L}_2$, where $\mathcal{L}_1\sigma = D\mathcal{S}_1[(\sigma\mathbf{x}_s)_s]$ and $\mathcal{L}_2 = D\mathcal{S}_2[(\sigma\mathbf{x}_s)_s]$. Using Eqs. (35) and (36), we get

$$\mathcal{L}_1[\sigma] = D\left(-\frac{\hat{\sigma}_0}{4}\begin{bmatrix}\cos s\\ \sin s\end{bmatrix} + \sum_{|k|>0}\frac{k}{8|k|}e^{iks}\begin{bmatrix}\hat{\sigma}_{k+1} - \hat{\sigma}_{k-1}\\ i(\hat{\sigma}_{k+1} + \hat{\sigma}_{k-1})\end{bmatrix}\right) \tag{37}$$

$$= -\frac{\hat{\sigma}_0}{4} - \frac{1}{8}\sum_{|k|>0}\frac{k^2}{|k|}\left[\hat{\sigma}_{k-1}e^{i(k-1)s} + \hat{\sigma}_{k+1}e^{i(k+1)s}\right], \tag{38}$$

$$= -\frac{\hat{\sigma}_0}{4} - \frac{1}{4}\sum_{|k|>0}|k|\hat{\sigma}_k e^{iks}. \tag{39}$$

By the change of variable $\tau = s + t$, we compute $\mathcal{L}_2\sigma$ as

$$\mathcal{L}_2[\sigma] = D\sum_{k\in\mathbb{Z}}\frac{\hat{\sigma}_k}{8\pi}e^{-iks}\int_0^{2\pi}\begin{bmatrix}-1+\cos\tau & \sin\tau\\ \sin\tau & -1-\cos\tau\end{bmatrix}\begin{bmatrix}-ik\sin(\tau-s) - \cos(\tau-s)\\ ik\cos(\tau-s) - \sin(\tau-s)\end{bmatrix}e^{ik\tau}\,d\tau, \tag{40}$$

$$= -\frac{1}{8}D\begin{bmatrix}-\hat{\sigma}_{-1} - \hat{\sigma}_1 - 2\hat{\sigma}_0\cos s\\ i(-\hat{\sigma}_1 + \hat{\sigma}_{-1}) - 2\hat{\sigma}_0\sin s\end{bmatrix} = \frac{\hat{\sigma}_0}{4}. \tag{41}$$

Therefore, the eigenvectors of $\mathcal{L}_2$ are $\{e^{ikt}\}_{k\in\mathbb{Z}}$ and the only non–zero eigenvalue is $\lambda_0 = 1/4$, corresponding to zero frequency. From (39) and (41), we conclude that the eigenvalues of $\mathcal{L}$ are $\left\{-\frac{|k|}{4}\right\}_{k\in\mathbb{Z}}$ and hence $\mathcal{L}$ has a null space of dimension one (constant vectors).

Stretching operator ($\mathcal{M}$). We state the result without the details, which follow the previous derivations. First, we analyze the action of the operators $D, \mathcal{T}$. We define $\mathcal{M} = \mathcal{T}\mathcal{L}^{-1}D$ and rewrite the operators as $D = [D^1 \quad D^2]$, $\mathcal{T} = \begin{bmatrix} \mathcal{T}^1 \\ \mathcal{T}^2 \end{bmatrix}$ and $\mathcal{M} = \begin{bmatrix} \mathcal{M}^{11} & \mathcal{M}^{12} \\ \mathcal{M}^{21} & \mathcal{M}^{22} \end{bmatrix}$. Then, the nonzero components of $\mathcal{M}$ are given by

- $D^1_{k,k-1:k+1} = \left[-\frac{k-1}{2} \quad 0 \quad \frac{k+1}{2}\right]$; $D^2_{k,k-1:k+1} = \left[\frac{i(k-1)}{2} \quad 0 \quad \frac{i(k+1)}{2}\right]$.
- $\mathcal{T}^1_{k,k-1:k+1} = \frac{ks}{8|k|}[-1 \quad 0 \quad 1]$; $\mathcal{T}^2_{k,k-1:k+1} = i\frac{ks}{8|k|}[1 \quad 0 \quad 1]$.
- $\mathcal{M}^{11}_{k,k-2:k+2} = \frac{ks}{8|k|}\left[\frac{k-2}{|k-1|} \quad 0 \quad -k\left(\frac{1}{|k-1|} + \frac{1}{|k+1|}\right) \quad 0 \quad \frac{k+2}{|k+1|}\right]$.
- $\mathcal{M}^{12}_{k,k-2:k+2} = i\frac{ks}{8|k|}\left[-\frac{k-2}{|k-1|} \quad 0 \quad k\left(\frac{1}{|k-1|} - \frac{1}{|k+1|}\right) \quad 0 \quad \frac{k+2}{|k+1|}\right]$; $\mathcal{M}^{21}_{k,k-2:k+2} = \mathcal{M}^{12}_{k,k-2:k+2}$.
- $\mathcal{M}^{22}_{k,k-2:k+2} = \frac{k}{8|k|}\left[-\frac{k-2}{|k-1|} \quad 0 \quad -k\left(\frac{1}{|k-1|} - \frac{1}{|k+1|}\right) \quad 0 \quad -\frac{k+2}{|k+1|}\right]$ where we use $l, i:j$ to denote the elements of the l-th row of the (infinite) matrix $\mathcal{M}$ with indices from i to j.

We constructed the matrix resulting from the finite-dimensional approximation of $\mathcal{M}$ and computed the eigenvalues. In Table 13, we list the highest magnitude of the eigenvalues for different discretizations.

Appendix C. Quadrature

Here, we provide more details on the computation of the logarithmic potential

$$\phi[f](\mathbf{x}) = \int_0^{2\pi} \log \rho f(\alpha) s_\alpha \, d\alpha, \quad \rho(\alpha) = \|\mathbf{x} - \mathbf{y}(\alpha)\|_2, \tag{42}$$

with $\mathbf{y}(\alpha) : [0, 2\pi] \to \gamma$. When $\mathbf{x} \in \gamma$, the integrand has a logarithmic singularity. Let $\mathbf{x} = \mathbf{y}(0)$ and assume that γ is a simple closed curve with no self-intersections. Then, the integrand in (42) becomes singular as α approaches 0 or 2π. We split the potential as $\phi = \phi_1 + \phi_2$, where $\phi_1[f] = \int_0^\pi \log \rho f s_\alpha \, d\alpha$ and $\phi_2[f] = \int_\pi^{2\pi} \log \rho f s_\alpha \, d\alpha$. We write ϕ_1 as

$$\phi_1[f](\mathbf{x}) = \int_0^\pi \left(\log \alpha + \log \frac{\rho}{\alpha}\right) f(\alpha) s_\alpha \, d\alpha. \tag{43}$$

If we prove that $\log(\rho/\alpha)$ is a smooth function, then we can use the high-order quadrature rules of [1] (specifically, Table 8 of [1]). By a Taylor's expansion of ρ around zero, we have

$$\frac{\rho(\alpha)}{\alpha} = \rho_\alpha(0) + \frac{\rho_{\alpha\alpha}(0)}{2}\alpha + \cdots \tag{44}$$

From (42) and by substituting $\mathbf{x} = \mathbf{y}(0)$, we get

$$\rho_\alpha = \frac{(\mathbf{y}(\alpha) - \mathbf{y}(0)) \cdot \mathbf{y}_s}{\|\mathbf{y}(\alpha) - \mathbf{y}(0)\|_2} s_\alpha. \tag{45}$$

By definition, on a smooth planar curve, $\lim_{\alpha \to 0} \frac{\mathbf{y}(\alpha) - \mathbf{y}(0)}{\|\mathbf{y}(\alpha) - \mathbf{y}(0)\|_2} = \mathbf{y}_s(0)$. Therefore, $\lim_{\alpha \to 0} \frac{\rho}{\alpha} = s_\alpha(0)$. Since, s_α is a nonzero function, we conclude that $\log(\rho/\alpha)$ is smooth.

By a transformation of variables $\alpha = 2\pi - \beta$, we get $\phi_2[f](\mathbf{x}) = -\int_0^\pi \log \rho f s_\beta \, d\beta$. This is in the same form as (43) and hence can be computed with high-accuracy using the quadrature rules of [1]. The convergence of the quatrature rule as a function of q and N is reported in Table 14 for the case of constant layer density.

Table 13
The maximum magnitude of the eigenvalues of a N-dimensional approximation of $\mathcal{M}$ on the unit circle.

N	32	128	1024
$\lvert\lambda\rvert_{\max}$	0.5	0.5	0.5

Table 14
Relative errors in computing $\mathcal{S}[\mathbf{f}]$ of $\mathbf{f} = 1$, defined on the boundary of the starfish vesicle shown in the adjacent figure.

N	32	64	128	256	512
$q = 4$	9.23e−003	4.93e−004	1.38e−005	8.03e−007	2.61e−008
$q = 8$	9.73e−003	5.13e−004	6.68e−006	1.52e−008	2.52e−011
$q = 16$	5.91e−001	4.98e−004	6.57e−006	8.70e−009	2.43e−013

Appendix D. Fast summation for stokes operator

Let G_l denote the fundamental solution for the Laplacian kernel, that is, $G_l(\mathbf{x},\mathbf{y}) = -\frac{1}{4\pi}\log\|\mathbf{x}-\mathbf{y}\|$. The FMM [14] accelerates the computation of the single layer potential $\int_\gamma G_l(\mathbf{x},\mathbf{y})u(\mathbf{y})\,d\mathbf{y}$. Here, we show that the fast evaluation of the Stokes single-layer can be accomplished using the FMM. We shall need the following identity:

$$\frac{\mathbf{r}\otimes\mathbf{r}}{\rho^2}\mathbf{v} = \frac{\mathbf{r}}{\rho^2}(\mathbf{r}\cdot\mathbf{v}) = (\mathbf{r}\cdot\mathbf{v})\nabla_{\mathbf{x}}\log\rho. \tag{46}$$

The convolution of any vector field $\mathbf{v}(\mathbf{y})$ with the Stokes kernel can be written as,

$$\begin{aligned}\mathcal{S}[\mathbf{v}](\mathbf{x}) &= \int_\gamma G_l(\mathbf{x},\mathbf{y})\mathbf{v} + \frac{1}{4\pi}\int_\gamma \frac{\mathbf{r}\otimes\mathbf{r}}{\rho^2}\mathbf{v} = \int_\gamma G_l(\mathbf{x},\mathbf{y})\mathbf{v} + \int_\gamma \nabla_{\mathbf{x}}G_l(\mathbf{x},\mathbf{y})(\mathbf{v}\cdot\mathbf{x}-\mathbf{v}\cdot\mathbf{y})\\ &= \int_\gamma G_l(\mathbf{x},\mathbf{y})\mathbf{v} + x_1\nabla_{\mathbf{x}}\int_\gamma G_l(\mathbf{x},\mathbf{y})v_1 + x_2\nabla_{\mathbf{x}}\int_\gamma G_l(\mathbf{x},\mathbf{y})v_2 - \nabla_{\mathbf{x}}\int_\gamma G_l(\mathbf{x},\mathbf{y})(\mathbf{v}\cdot\mathbf{y})\end{aligned}$$

Therefore, we only need fast evaluation of the following single-layer Laplace potentials,

1. $\int_\gamma G_l(\mathbf{x},\mathbf{y})v_1$,
2. $\int_\gamma G_l(\mathbf{x},\mathbf{y})v_2$,
3. $\int_\gamma G_l(\mathbf{x},\mathbf{y})(\mathbf{v}\cdot\mathbf{y})$.

References

[1] B.K. Alpert, Hybrid Gauss-trapezoidal quadrature rules, SIAM Journal on Scientific Computing 20 (1999) 1551–1584.
[2] U.M. Ascher, L.R. Petzold, Computer Methods for Ordinary Differential Equations and Differential–Algebraic Equations, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1998.
[3] U.M. Ascher, S.J. Ruuth, B.T.R. Wetton, Implicit-explicit methods for time-dependent partial differential equations, SIAM Journal on Numerical Analysis 32 (1995) 797–823.
[4] J.T. Beale, T.Y. Hou, J. Lowengrub, Convergence of a boundary integral method for water waves, Siam Journal On Numerical Analysis 33 (1996) 1797–1843.
[5] G. Biros, L. Ying, D. Zorin, A fast solver for the stokes equations with distributed forces in complex geometries, Journal of Computational Physics 194 (2004) 317–348.
[6] C. Pozrikidis, Effect of membrane bending stiffness on the deformation of capsules in simple shear flow, Journal of Fluid Mechanics 440 (2001) 269–291.
[7] H. Cheng, L. Greengard, On the numerical evaluation of electrostatic fields in dense random dispersions of cylinders, Journal of Computational Physics 136 (1997) 629–639.
[8] K.H. de Haas, C. Blom, D. van den Ende, M.H.G. Duits, J. Mellema, Deformation of giant lipid bilayer vesicles in shear flow, Physical Review E 56 (1997) 7132–7137.
[9] P. Dimitrakopoulos, Interfacial dynamics in Stokes flow via a three-dimensional fully-implicit interfacial spectral boundary element algorithm, Journal Of Computational Physics 225 (2007) 408–426.
[10] Q. Du, C. Liu, X. Wang, A phase field approach in the numerical study of the elastic bending energy for vesicle membranes, Journal of Computational Physics 198 (2004) 450–468.
[11] Q. Du, C. Liu, X. Wang, Simulating the deformation of vesicle membranes under elastic bending energy in three-dimensions, Journal of Computational Physics 212 (2006) 757–777.
[12] E. Sackmann, Supported membranes: scientific and practical applications, Science 271 (1996) 43–48.
[13] F. Feng, W.S. Klug, Finite element modeling of lipid bilayer membranes, Journal of Computational Physics 220 (2006) 394–408.
[14] L. Greengard, The Rapid Evaluation of Potential Fields in Particle Systems, MIT Press, Cambridge, Mass, 1988.
[15] E. Hairer, G. Wanner, Solving Ordinary Differential Equations II, Springer-Verlag, 1980.
[16] T.Y. Hou, J.S. Lowengrub, M.J. Shelley, Removing the stiffness from interfacial flows with surface tension, Journal of Computational Physics 114 (1994) 312–338.
[17] A. Acrivos, G.K. Youngren, Stokes flow past a particle of arbitrary shape: a numerical method of solution, Journal of Fluid Mechanics 69 (1975) 377–403.
[18] V. Kantsler, V. Steinberg, Orientation and dynamics of a vesicle in tank-treading motion in shear flow, Physical Review Letters 95 (2005).
[19] M. Kraus, W. Wintz, U. Seifert, R. Lipowsky, Fluid vesicles in shear flow, Physical Review Letters 77 (1996).
[20] M. Kraus, W. Wintz, U. Seifert, R. Lipowsky, Fluid vesicles in shear flow, Physical Review Letters 77 (1996) 3685–3688.
[21] M.C.A. Kropinski, Integral equation methods for particle simulations in creeping flows, Computers and Mathematics with Applications 38 (1999) 67–87.
[22] M.C.A. Kropinski, Numerical methods for multiple inviscid interfaces in creeping flows, Journal of Computational Physics 180 (2002) 1–24.
[23] M.C.A. Kropinski, An efficient numerical method for studying interfacial motion in two-dimensional creeping flows, Journal of Computational Physics 171 (2001) 479–508.
[24] X. Li, D. Barthes-Biesel, A. Helmy, Large deformations and burst of a capsule freely suspended in an elongational flow, Journal of Fluid Mechanics 187 (1988) 179–196.
[25] A.J. Markvoort, R.A. Van Santen, P.A.J. Hilbers, Vesicle shapes from molecular dynamics simulations, Journal of Physical Chemistry B 110 (2006) 22780–22785.
[26] H. Noguchi, G. Gompper, Shape transitions of fluid vesicles and red blood cells in capillary flows, in: Proceedings of The National Academy of Sciences of The United States of America, vol. 102, 2005, pp. 14159–14164.
[27] H. Power, L. Wrobel, Boundary integral methods in fluid mechanics, Computational Mechanics Publications, 1995.
[28] C. Pozrikidis, The axisymmetric deformation of a red blood cell in uniaxial straining Stokes flow, Journal of Fluid Mechanics 216 (1990) 231–254.
[29] C. Pozrikidis, Boundary Integral and Singularity Methods for Linearized Viscous Flow, Cambridge University Press, Cambridge, 1992.
[30] C. Pozrikidis, On the transient motion of ordered suspensions of liquid drops, Journal of Fluid Mechanics 246 (1993) 301–320.
[31] C. Pozrikidis, Finite deformation of liquid capsules enclosed by elastic membranes in simple shear flow, Journal of Fluid Mechanics 297 (1995) 123–152.

[32] C. Pozrikidis, Interfacial dynamics for stokes flow, Journal of Computational Physics 169 (2001) 250–301.
[33] J. Rallison, A. Acrivos, A numerical study of the deformation and burst of a viscous drop in an extensional flow, Journal of Fluid Mechanics 89 (1978) 191–200.
[34] S.C. Reddy, L.N. Trefethen, Stability of the method of lines, Numerische Mathematik 62 (1992) 235–267.
[35] Y. Saad, Iterative Methods for Sparse Linear Systems, second ed., SIAM, Philadelpha, PA, 2003.
[36] U. Seifert, Adhesion of vesicles in two-dimensions, Physical Review A 43 (1991) 6803–6814.
[37] U. Seifert, Configurations of fluid membranes and vesicles, Advances in Physics 46 (1997) 13–137.
[38] M.J. Stevens, Coarse–grained simulations of lipid bilayers, Journal of Chemical Physics 121 (2004).
[39] S. Sukumaran, U. Seifert, Influence of shear flow on vesicles near a wall: a numerical study, Physical Review E 64 (2001).
[40] A.-K. Tornberg, L. Greengard, A fast multipole method for the three-dimensional Stokes equations, Journal of Computational Physics 227 (2008) 1613–1619.
[41] A.-K. Tornberg, M. Shelley, Simulating the dynamics and interactions of flexible fibers in stokes flows, Journal of Computational Physics 196 (2004) 8–40.
[42] L. Ying, G. Biros, D. Zorin, A high-order 3D boundary integral equation solver for elliptic pdes in smooth domains, Journal of Computational Physics 219 (2006) 247–275.
[43] G.J.R. Yuhong Fu, Fast solution method for three-dimensional stokesian many-particle problems, Communications in Numerical Methods in Engineering 16 (2000) 145–149.
[44] H. Zhou, C. Pozrikidis, Deformation of liquid capsules with incompressible interfaces in simple shear flow, Journal of Fluid Mechanics 283 (1995) 175–200.