

Accepted Manuscript

A continuous energy-based immersed boundary method for elastic shells

Ondrej Maxian, Andrew T. Kassen, Wanda Strychalski

PII: S0021-9991(18)30354-1
DOI: <https://doi.org/10.1016/j.jcp.2018.05.045>
Reference: YJCPH 8045

To appear in: *Journal of Computational Physics*

Received date: 21 November 2017
Revised date: 28 April 2018
Accepted date: 27 May 2018

Please cite this article in press as: O. Maxian et al., A continuous energy-based immersed boundary method for elastic shells, *J. Comput. Phys.* (2018), <https://doi.org/10.1016/j.jcp.2018.05.045>

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.



Highlights

- New variational method for computing forces on thin elastic shells within the IB method is presented.
- Method gives a continuous force function on the entire surface of a hyperelastic shell.
- Comparison to a previous formulation where the surface and energy functional are first discretized is provided.
- Evidence of improved accuracy of elastic forces is presented.
- Method is applied to 3D models of a red blood cell in capillary flow and cellular blebbing.

A continuous energy-based immersed boundary method for elastic shells

Ondrej Maxian^{a,b}, Andrew T. Kassen^c, Wanda Strychalski^{b,*}

^aDepartment of Chemical and Biomolecular Engineering, Case Western Reserve University, Cleveland, OH 44106

^bDepartment of Mathematics, Applied Mathematics, and Statistics, Case Western Reserve University, Cleveland, OH 44106

^cDepartment of Mathematics, University of Utah, UT 84112

Abstract

The immersed boundary method is a mathematical formulation and numerical method for solving fluid-structure interaction problems. For many biological problems, such as models that include the cell membrane, the immersed structure is a two-dimensional infinitely thin elastic shell immersed in an incompressible viscous fluid. When the shell is modeled as a hyperelastic material, forces can be computed by taking the variational derivative of an energy density functional. A new method for computing a continuous force function on the entire surface of the shell is presented here. The new method is compared to a previous formulation where the surface and energy functional are discretized before forces are computed. For the case of Stokes flow, a method for computing quadrature weights is provided to ensure the integral of the elastic spread force density remains zero throughout a dynamic simulation. Tests on the method are conducted and show that it yields more accurate force computations than previous formulations as well as more accurate geometric information such as mean curvature. The method is then applied to a model of a red blood cell in capillary flow and a 3D model of cellular blebbing.

Keywords: fluid-structure interaction, stokes flow, hyperelasticity, red blood cells, blebbing

1. Introduction

The immersed boundary (IB) method was first introduced by Charles Peskin in 1972 to model blood flow around heart valves [1]. Since then, it has been applied to a wide variety of biological models, including insect flight [2, 3], animal swimming [4], and cellular mechanics [5]. In three-dimensional models, computing the elasticity of an immersed 2D elastic membrane or 3D elastic solid as it deforms is more challenging than in 2D because it involves stress tensors. The first studies in this area used lattice-spring models of elasticity (e.g. [6]), but these methods are disconnected from a constitutive law. The finite element method has also been used to compute the elasticity of immersed structures [7–10], but has not been extended to infinitely thin shells. The IB method has been applied to hyperelastic solids or shells whose energies are governed by a strain energy density function [11, 12]. However, the surface or solid representation has been assumed to be piecewise linear, and the accuracy of force computations with such a method has not been rigorously tested.

The authors showed in [11] that for a hyperelastic solid, forces can be computed without using stress tensors. In this work, the derivation was limited to Cartesian coordinates and applied to solids and thick elastic shells. The method was later applied to infinitely thin elastic shells represented in two curvilinear coordinates and was subsequently used to model red blood cells [12], osmotic swelling [13], and two-phase gels [14]. However, these models have used a piecewise linear boundary representation (surface triangulation) [12] or marker and cell discretization of the surface [14]. Both approaches are based on the assumption that curved surfaces are locally planar, which introduces surface discretization error and fails to capture the inherent curvature of the surface.

*Corresponding Author

Email addresses: oxm60@case.edu (Ondrej Maxian), kassen@math.utah.edu (Andrew T. Kassen), wis6@case.edu (Wanda Strychalski)

Studies on surface representations have generally been conducted separately from those on force computations. In [15, 16], the authors focus on representing thin surfaces continuously with spherical harmonic or radial basis function (RBF) interpolants. In both cases, force computations are treated through simple explicit expressions for surface tension [15] or fiber elasticity [16]. The authors do not consider finding forces through variational derivatives.

Boundary integral methods (BIMs) have been progressing concurrently with the IB method and have been used by others to model fluid-structure interactions in zero Reynolds number flow [17–20]. In BI methods, the “hydrodynamic traction jump” across the membrane [17], which is equivalent to the Lagrangian elastic force density per unit current area [20], must be computed independently prior to integration. [21] and [22] both demonstrate the use of a spectrally convergent spherical harmonic representation for surfaces that allows for the calculation of force in BIMs. However, the computations in [21] and [22] both follow the traditional formulation of BI methods in computing the traction jump from a system of equations involving the Cauchy stress tensor, which itself comes from a series of tensor expansions. Additionally, BI methods require numerical schemes to resolve integrals that are singular at the immersed surfaces.

The aim of this work is to provide a “bridge” between the fields of surface representation and energy-based force functions within the IB method. Our goal is to use the continuous surface representations from [15] to formulate a continuous function that represents the force density on the membrane as a function of its curvilinear coordinates. In this manner, we avoid the discretization error from surface representation. The advantages of the approach presented here are that the force is computed directly in a single computation, bypassing the need to compute stress tensors at each time step, the interpolation of structure velocities in the IB method occurs over a coarser mesh, and the use of the IB method avoids the need for numerical methods to compute singular integrals as in BI methods.

Since many biological phenomena occur on the cellular level, the small length scales (microns) lead to very small Reynolds numbers ($\text{Re} \sim 10^{-2} - 10^{-5}$). In order to simulate Stokes equations with periodic boundary conditions, the integral of the force density over the fluid domain must be zero [23]. In [24], the authors proposed a method for ensuring this condition is met when tether forces are used in the IB method. For translation invariant hyperelastic materials represented by continuous functions, the integral condition is satisfied in the continuous formulation [25]. However, extra care must be taken to ensure that the discrete integral of the forces in the IB method is zero. Another feature of this work is that we provide a method for computing the correct quadrature weights for closed surfaces so that the IB method can be used with 3D Stokes flow.

The rest of the paper is organized as follows. In Section 2, we present the mathematical formulation of the immersed membrane problem, including our method of computing force densities and correct area quadrature weights to ensure that the forces on the fluid integrate to zero in Stokes’ flow. We continue by describing an alternative method based on piecewise linear surface discretization, which we use for comparison. In Section 3, we discuss the numerical formulation and discretization of the problem and provide implementation details. We conclude by presenting our results in Section 4, where we perform force computations and IB simulations on two test objects. In Section 5, we apply our model to two different biological processes: a red blood cell flowing through a capillary and a 3D model of cellular blebbing.

2. Mathematical Formulation

In this section, we begin by presenting an overview of the model equations and mathematical formulation. We then describe how a continuous elastic force density function can be computed on shells using the energy-based formulation. We provide a method for computing quadrature weights so that the forces applied to the fluid integrate to zero. We compare this to the discretized surface formulation from [12, 26] and conclude by presenting the energy functionals used in our numerical tests.

2.1. Immersed boundary method formulation

Our model system is an infinitely thin elastic shell immersed in a viscous fluid. For our applications, the shell represents the cell membrane. Due to the small length scales on the cellular level, inertial forces

can be neglected and the fluid obeys Stokes' equations,

$$\mu \Delta \mathbf{u} - \nabla p + \mathbf{f} = \mathbf{0}, \quad (1)$$

$$\nabla \cdot \mathbf{u} = 0, \quad (2)$$

where \mathbf{u} represents the fluid velocity, p is the pressure, \mathbf{f} is an external force, and μ is the fluid viscosity. The model is formulated using the IB method so that the thin elastic shell is represented on a moving Lagrangian coordinate system while the velocity and pressure are represented on a fixed Eulerian grid. Elastic forces are computed on the Lagrangian grid and spread onto the Eulerian grid to construct the external force density. The spreading operator S is defined as

$$\mathbf{f} = S\mathbf{F} = \int_{\Gamma} \mathbf{F}(\mathbf{q}, t) \delta(\mathbf{x} - \mathbf{X}(\mathbf{q}, t)) d\mathbf{q}, \quad (3)$$

where Γ represents the Lagrangian structure (shell), $\mathbf{q} = (q_1, q_2)$ represents material surface coordinates, $\mathbf{X}(\mathbf{q}, t)$ represents the position of the structure at time t , and $\delta(\mathbf{x})$ represents the Dirac delta function. We use the convention that lower case letters indicate quantities on the Eulerian grid and capitalized letters denote quantities on the Lagrangian structure. The structure is updated with the local fluid velocity, satisfying the no-slip boundary condition,

$$\frac{d\mathbf{X}}{dt} = \mathbf{U}, \quad (4)$$

where \mathbf{U} is the interpolated fluid velocity. The interpolation operator is the adjoint of the spreading operator in Eq. (3) and is given by

$$\mathbf{U} = S^* \mathbf{u} = \int_{\Omega} \mathbf{u}(\mathbf{x}, t) \delta(\mathbf{x} - \mathbf{X}(\mathbf{q}, t)) d\mathbf{x}, \quad (5)$$

where Ω denotes the fluid domain.

2.2. Computing the elastic forces

We outline two different methods to compute Lagrangian forces analytically based on the assumption that the structure is a hyperelastic material characterized by an energy density functional

$$\mathcal{E} = \int_{\Gamma_0} W d\mathbf{q}, \quad (6)$$

where W is a strain energy density specified by a constitutive law, such as the neo-Hookean model. Exact forms of W are provided in Section 2.6. The energy-based model for computing Lagrangian forces on an immersed surface was derived in [11]. Let the surface Γ with undeformed configuration Γ_0 be defined by Lagrangian coordinates q_1 and q_2 (q_i 's are the parameters). Then the force density per unit reference configuration is related to the variational derivative of energy $\frac{\delta \mathcal{E}}{\delta \mathbf{X}}$ by

$$\mathbf{F} = -\frac{\delta \mathcal{E}}{\delta \mathbf{X}}(q_1, q_2, t), \quad (7)$$

where \mathcal{E} is the surface energy and t is time.

The variational derivative is defined as follows. Let $\phi_{\delta \mathbf{X}}(\alpha) = \mathcal{E}(\mathbf{X} + \alpha \delta \mathbf{X})$. Then the variational derivative is the function $\frac{\delta \mathcal{E}}{\delta \mathbf{X}}$ that obeys

$$\left. \frac{d\phi_{\delta \mathbf{X}}(\alpha)}{d\alpha} \right|_{\alpha=0} = \int_{\Gamma_0} \frac{\delta \mathcal{E}}{\delta \mathbf{X}}(q_1, q_2, t) \cdot \delta \mathbf{X}(q_1, q_2) dA_0. \quad (8)$$

In [11], this theory was applied to discrete solids in Cartesian coordinates. We extend the theory further to curvilinear coordinates on *continuous closed* surfaces.

In the subsequent sections, we describe two different approaches for computing the force density in Eq. (7). We first outline a procedure to find the variational derivative of the continuous energy density functional and follow with a discussion on computing the proper area weights to get force from force density. We next discuss the second method, the linear discrete surface method (LDSM) from [12]. The difference between the two methods lies primarily in the fact that our formulation relies on a *continuous* force function whose integral is discretized after forces are computed, whereas in LDSM, the surface and energy functional are first discretized. Forces are then computed after the discretized surface energy is exactly integrated.

2.3. Computing forces from the variational derivative of the energy functional

Since the surfaces in our applications are topological spheres, we parameterize our surface in terms of spherical coordinates. However, our derivations can be generalized to any coordinate system. We begin by defining a two dimensional surface in terms of spatial coordinates $q_1 = \lambda$ and $q_2 = \theta$. Let

$$\mathbf{Z}(\lambda, \theta) = \begin{pmatrix} \hat{X}(\lambda, \theta) \\ \hat{Y}(\lambda, \theta) \\ \hat{Z}(\lambda, \theta) \end{pmatrix} \text{ and } \mathbf{X}(\lambda, \theta) = \begin{pmatrix} X(\lambda, \theta) \\ Y(\lambda, \theta) \\ Z(\lambda, \theta) \end{pmatrix} \quad (9)$$

be the reference and current configurations of the surface, respectively, where $-\pi < \lambda \leq \pi$ and $-\pi/2 < \theta \leq \pi/2$. Here $\mathbf{X}, \mathbf{Z} : (\lambda, \theta) \mapsto \mathbb{R}^3$, and we have intentionally expressed Eq. (9) in a general form to emphasize that our derivation can be applied to any particular choice of \mathbf{X} that describes a closed, smooth surface. define the matrices

$$\nabla_{\theta} \mathbf{X} = \begin{pmatrix} \frac{\partial \mathbf{X}}{\partial \lambda} & \frac{\partial \mathbf{X}}{\partial \theta} \end{pmatrix} \quad (10)$$

and

$$\nabla_{\theta} \mathbf{Z} = \begin{pmatrix} \frac{\partial \mathbf{Z}}{\partial \lambda} & \frac{\partial \mathbf{Z}}{\partial \theta} \end{pmatrix}, \quad (11)$$

where $\frac{\partial \mathbf{X}}{\partial \lambda}$ is a vector in \mathbb{R}^3 that represents the component-wise partial λ derivative of \mathbf{X} (the other components of $\nabla_{\theta} \mathbf{X}$ and $\nabla_{\theta} \mathbf{Z}$ are defined similarly). The matrix $\nabla_{\theta} \mathbf{X}$ is the deformation gradient and the rectangular analogue of the square matrix \mathbb{F} in [11]. We define the metric tensors

$$\mathcal{G} = (\nabla_{\theta} \mathbf{X})^T (\nabla_{\theta} \mathbf{X}) = \begin{pmatrix} \frac{\partial \mathbf{X}}{\partial \lambda} \cdot \frac{\partial \mathbf{X}}{\partial \lambda} & \frac{\partial \mathbf{X}}{\partial \lambda} \cdot \frac{\partial \mathbf{X}}{\partial \theta} \\ \frac{\partial \mathbf{X}}{\partial \theta} \cdot \frac{\partial \mathbf{X}}{\partial \lambda} & \frac{\partial \mathbf{X}}{\partial \theta} \cdot \frac{\partial \mathbf{X}}{\partial \theta} \end{pmatrix} \quad (12)$$

and

$$\mathcal{G}_0 = (\nabla_{\theta} \mathbf{Z})^T (\nabla_{\theta} \mathbf{Z}) = \begin{pmatrix} \frac{\partial \mathbf{Z}}{\partial \lambda} \cdot \frac{\partial \mathbf{Z}}{\partial \lambda} & \frac{\partial \mathbf{Z}}{\partial \lambda} \cdot \frac{\partial \mathbf{Z}}{\partial \theta} \\ \frac{\partial \mathbf{Z}}{\partial \theta} \cdot \frac{\partial \mathbf{Z}}{\partial \lambda} & \frac{\partial \mathbf{Z}}{\partial \theta} \cdot \frac{\partial \mathbf{Z}}{\partial \theta} \end{pmatrix}. \quad (13)$$

The Green-Lagrange strain energy tensor is defined as

$$\gamma = \frac{1}{2}(\mathcal{G} - \mathcal{G}_0). \quad (14)$$

In Cartesian coordinates, the Cauchy-Green deformation tensor is given by $\mathcal{C} = \mathcal{A}^T \mathcal{A}$, where \mathcal{A} is the surface displacement gradient, $\mathcal{A} = \frac{d\mathbf{X}}{d\mathbf{Z}}$ [18]. As in [12] we consider a related tensor,

$$\mathcal{C} = \mathcal{G} \mathcal{G}_0^{-1}. \quad (15)$$

In hyperelastic materials, the energy density functional W is defined in terms of two deformation invariants, I_1 and I_2 , where $I_1 = \text{tr}(\mathcal{C}) - 2$ and $I_2 = \det(\mathcal{C}) - 1$. As noted in [11, 18], the neo-Hookean energy density can be represented as a function of the invariants, i.e. $W(I_1, I_2)$, or as a function of the deformation gradient, $\nabla_\theta \mathbf{X}$. The total energy of the surface is given by an integral of the energy density

$$\mathcal{E}(\mathbf{X}) = \int_{\Gamma_0} W(I_1, I_2) dA_0 = \int_{\Gamma_0} W(\nabla_\theta \mathbf{X}) dA_0 = \int_{\Gamma_0} W(\nabla_\theta \mathbf{X}) \sqrt{\det \mathcal{G}_0} d\lambda d\theta, \quad (16)$$

where Γ_0 is the undeformed shell surface, and the area differential dA_0 denotes integration over the *reference* configuration. The final equality comes from the fact that $\sqrt{\det \mathcal{G}_0} = \|\mathbf{n}_0\|$, where $\mathbf{n}_0 = \frac{\partial \mathbf{Z}}{\partial \lambda} \times \frac{\partial \mathbf{Z}}{\partial \theta}$ is the normal (not necessarily a unit vector) to the surface in the reference configuration, as shown in [12]. We note that the factor $\sqrt{\det \mathcal{G}_0}$ is a function of the coordinates λ and θ , but *not* of the current configuration \mathbf{X} or its deformation gradient $\nabla_\theta \mathbf{X}$.

In order to compute the Lagrangian force density Eq. (7), consider a deformation by $\delta \mathbf{X}$. Evaluating Eq. (8), we have

$$\phi_{\delta \mathbf{X}}(\alpha) = \mathcal{E}(\mathbf{X} + \alpha \delta \mathbf{X}) = \int_{\Gamma_0} W(\nabla_\theta (\mathbf{X} + \alpha \delta \mathbf{X})) \sqrt{\det \mathcal{G}_0} d\lambda d\theta. \quad (17)$$

To compute the variational derivative, we follow the derivation from [11] and first apply the chain rule for $\nabla_\theta \mathbf{X}$ (note that the area weight $\sqrt{\det \mathcal{G}_0}$ is unaffected),

$$\left. \frac{d\phi_{\delta \mathbf{X}}(\alpha)}{d\alpha} \right|_{\alpha=0} = \sum_{i=1}^3 \sum_{j=1}^2 \int_{\Gamma_0} \frac{\partial W}{\partial (\nabla_\theta \mathbf{X})_{ij}} \frac{\partial \delta X_i}{\partial q_j} \sqrt{\det \mathcal{G}_0} d\lambda d\theta. \quad (18)$$

The index i describes the Cartesian coordinates (indexed from 1 to 3), and the index j describes the curvilinear coordinates λ and θ (indexed from 1 to 2). We note that the first Piola-Kirchhoff stress tensor is

$$\mathcal{P} = \frac{\partial W}{\partial (\nabla_\theta \mathbf{X})}. \quad (19)$$

Applying integration by parts on each of the coordinates λ and θ , we have

$$\left. \frac{d\phi_{\delta \mathbf{X}}(\alpha)}{d\alpha} \right|_{\alpha=0} = - \sum_{i=1}^3 \sum_{j=1}^2 \int_{\Gamma_0} \frac{\partial (\mathcal{P}_{ij} \sqrt{\det \mathcal{G}_0})}{\partial q_j} \delta X_i d\lambda d\theta. \quad (20)$$

The boundary terms evaluate to zero under the assumption that the surface is closed. We obtain an area integral by multiplying and dividing by $\sqrt{\det \mathcal{G}_0}$,

$$\left. \frac{d\phi_{\delta \mathbf{X}}(\alpha)}{d\alpha} \right|_{\alpha=0} = - \sum_{i=1}^3 \sum_{j=1}^2 \int_{\Gamma_0} \frac{1}{\sqrt{\det \mathcal{G}_0}} \frac{\partial (\mathcal{P}_{ij} \sqrt{\det \mathcal{G}_0})}{\partial q_j} \delta X_i dA_0. \quad (21)$$

Using the surface divergence operator in the undeformed state

$$\nabla_\theta \cdot = \frac{1}{\sqrt{\det \mathcal{G}_0}} \left(\frac{\partial}{\partial \lambda} \sqrt{\det \mathcal{G}_0} \mathbf{e}_1 + \frac{\partial}{\partial \theta} \sqrt{\det \mathcal{G}_0} \mathbf{e}_2 \right) \cdot, \quad (22)$$

we can write the variational derivative from Eq. (7) as

$$\frac{\delta \mathcal{E}}{\delta \mathbf{X}} = -\nabla_\theta \cdot \mathcal{P}^T. \quad (23)$$

The above notation differs from [11] and instead follows [27]; it relies on a traditional tensor expansion in which \mathcal{P} is written in terms of base dyads as $\sum_{i,j} \mathcal{P}_{ij} \mathbf{e}_i \mathbf{e}_j$. As shown in Appendix A, using ∇_θ from Eq.

(22) and carrying out the dot product via tensor expansion in Eq. (23) yields the product in Eq. (21). The force density per unit undeformed configuration in Eq. (7) is then

$$\mathbf{F} = -\frac{\delta \mathcal{E}}{\delta \mathbf{X}} = \nabla_{\theta} \cdot \mathcal{P}^T. \quad (24)$$

We point out the analogy with [11], where the surface divergence operator in Cartesian coordinates has been replaced by the curvilinear surface divergence. Although we have obtained a representation for the force, it is in practice quite cumbersome to compute $\mathcal{P} = \frac{\partial W}{\partial(\nabla_{\theta} \mathbf{X})}$ by directly differentiating the energy density (however, it is possible for some simple forms of W). We therefore expand the matrix \mathcal{P}^T using the chain rule with the Green-Lagrange strain tensor, γ , as an intermediate:

$$\mathcal{P}^T = \frac{\partial W}{\partial(\nabla_{\theta} \mathbf{X})^T} = \frac{\partial W}{\partial \gamma} : \frac{\partial \gamma}{\partial(\nabla_{\theta} \mathbf{X})^T}, \quad (25)$$

where the double dot product $:$ arises from the product of a fourth order tensor $\frac{\partial \gamma}{\partial(\nabla_{\theta} \mathbf{X})^T}$ and second order tensor $\frac{\partial W}{\partial \gamma}$ to form the second order tensor \mathcal{P}^T (see Appendix A for a detailed expansion). We denote the symmetric second order tensor $\frac{\partial W}{\partial \gamma} = \mathcal{S}$, where \mathcal{S} is the second Piola-Kirchhoff stress tensor [20]. We show in Appendix A that

$$\mathcal{P}^T = \frac{\partial W}{\partial(\nabla_{\theta} \mathbf{X})^T} = \frac{\partial W}{\partial \gamma} : \frac{\partial \gamma}{\partial(\nabla_{\theta} \mathbf{X})^T} = \mathcal{S} : \frac{\partial \gamma}{\partial(\nabla_{\theta} \mathbf{X})^T} = \mathcal{S} \cdot (\nabla_{\theta} \mathbf{X})^T. \quad (26)$$

By the symmetry of γ (and \mathcal{S}), the transpose of both sides in Eq. (26) yields

$$\mathcal{P} = (\nabla_{\theta} \mathbf{X}) \cdot \mathcal{S}^T = (\nabla_{\theta} \mathbf{X}) \cdot \mathcal{S}. \quad (27)$$

In Eq. (27), we recover the correspondence between the first and second Piola-Kirchhoff stress tensors [28].

To compute the tensor \mathcal{S} in terms of the coordinates λ and θ , we begin by observing that in Eq. (15), $\mathcal{C} = \mathcal{I}_2 + 2\gamma \mathcal{G}_0^{-1}$, where \mathcal{I}_2 is the rank 2 identity tensor. Applying the chain rule and differentiating \mathcal{C} directly with respect to γ yields

$$\mathcal{S} = \frac{\partial W}{\partial \gamma} = \frac{\partial W}{\partial \mathcal{C}} : \frac{\partial \mathcal{C}}{\partial \gamma} = 2 \frac{\partial W}{\partial \mathcal{C}} \cdot \mathcal{G}_0^{-1}. \quad (28)$$

Computing $\frac{\partial W}{\partial \mathcal{C}}$ by the chain rule, we have

$$\frac{\partial W}{\partial \mathcal{C}} = \frac{\partial W}{\partial I_1} \frac{\partial I_1}{\partial \mathcal{C}} + \frac{\partial W}{\partial I_2} \frac{\partial I_2}{\partial \mathcal{C}}, \quad (29)$$

where each $\frac{\partial I_j}{\partial \mathcal{C}}$ is a second order tensor obtained from taking the derivative of each invariant I_j with respect to each element of \mathcal{C} . We show in Appendix A that

$$\frac{\partial I_1}{\partial \mathcal{C}} = \mathcal{I}_2 \quad (30)$$

and

$$\frac{\partial I_2}{\partial \mathcal{C}} = (\det \mathcal{C}) (\mathcal{G}^{-1} \mathcal{G}_0). \quad (31)$$

Substituting Eqs. (30) and (31) into Eqs. (28) and (29),

$$\mathcal{S} = 2 \frac{\partial W}{\partial \mathcal{C}} \cdot \mathcal{G}_0^{-1} = 2 \left(\frac{\partial W}{\partial I_1} \frac{\partial I_1}{\partial \mathcal{C}} + \frac{\partial W}{\partial I_2} \frac{\partial I_2}{\partial \mathcal{C}} \right) \cdot \mathcal{G}_0^{-1}, \quad (32)$$

$$\mathcal{S} = 2 \left(\frac{\partial W}{\partial I_1} \mathcal{I}_2 + \frac{\partial W}{\partial I_2} (\det \mathcal{C}) (\mathcal{G}^{-1} \mathcal{G}_0) \right) \cdot \mathcal{G}_0^{-1}, \quad (33)$$

$$\mathcal{S} = 2 \frac{\partial W}{\partial I_1} \mathcal{G}_0^{-1} + 2 \frac{\partial W}{\partial I_2} (\det \mathcal{C}) \mathcal{G}^{-1}. \quad (34)$$

Since the constitutive law is often expressed in terms of the invariants I_1 and I_2 , this formulation allows us to find \mathbf{F} with respect to λ and θ using Eq. (27).

Returning to the computation of force density as $\mathbf{F} = \nabla_{\theta} \cdot \mathcal{P}^T$ from Eq. (24), we apply the surface divergence operator in the undeformed state in Eq. (22). Then the force density can be expressed as

$$\mathbf{F} = \frac{1}{\sqrt{\det \mathcal{G}_0}} \left(\frac{\partial}{\partial \lambda} \sqrt{\det \mathcal{G}_0} \left(\mathcal{S}_{11} \frac{\partial \mathbf{X}}{\partial \lambda} + \mathcal{S}_{12} \frac{\partial \mathbf{X}}{\partial \theta} \right) + \frac{\partial}{\partial \theta} \sqrt{\det \mathcal{G}_0} \left(\mathcal{S}_{21} \frac{\partial \mathbf{X}}{\partial \lambda} + \mathcal{S}_{22} \frac{\partial \mathbf{X}}{\partial \theta} \right) \right). \quad (35)$$

Further details are located in Appendix A. It is also possible to arrive at Eq. (35) by expanding the tensor \mathcal{P} in Eq. (18) prior to integrating by parts.

2.4. From force density to force

The method developed in Section 2.3 allows us to find the force *density* per unit reference configuration at any point \mathbf{X} . However, because the IB method requires the *force* to be spread onto the fluid, it remains to compute the area weights associated with each point on the surface. While some applications do not require precise area weights, discretized structures in Stokes flow must have the correct area weights so that the discretized integral of force over the surface is zero. Because Eq. (35) is a translation invariant force density function, it can be shown [25] that its integral is zero in the continuous sense. The discretization of the force integral must preserve this property for structures immersed in Stokes flow. In our formulation, this condition becomes one on the surface quadrature weights. If accurate surface quadrature weights can be computed, the force density will integrate to zero discretely (within some quadrature error). This is because the integral of the force generated by a translation invariant hyperelastic energy over a closed surface is known to be zero [25].

Let n be the number of points at which the force is evaluated and consider the discretized integral of the force over the reference configuration

$$\int_{\Gamma_0} \mathbf{F} dA_0 \approx \sum_{i=1}^n \mathbf{F}(\mathbf{X}^i) (\Delta A_0)_i. \quad (36)$$

The factors $(\Delta A_0)_i$ are the appropriate area weights. Our goal is to determine the vector of area weights $\Delta \mathbf{A}_0 = \Delta \mathbf{A}_0(\mathbf{Z})$ (of length n) such that the integral in Eq. (36) is as close to exact as possible; that is, we are computing quadrature weights for integration over the undeformed configuration. Since the integral is over the reference configuration, the weights $\Delta \mathbf{A}_0$ only need to be computed once for the undeformed shape. In our applications, the undeformed shape is usually a sphere of radius 1, but the method we outline here can be used to compute the area weights in any reference configuration.

2.4.1. Weights on a sphere of radius 1

Beginning with the unit sphere \mathbb{S}^2 , we use a method from [29] to compute nonnegative weights $\Delta \mathbf{A}_0$. Note that the nonnegativity constraint is necessary due to the physics of the problem; the force and force densities should point in the same directions, so the area weights should be nonnegative.

Let n be the number of points at which the force function from Section 2.3 is evaluated (we refer to these as *evaluation points*.) Following [29], we compute nonnegative quadrature weights $(\Delta A_0)_i$ such that

$$\int_{\mathbb{S}^2} f(\mathbf{X}) dA_0 = \sum_{i=1}^n f(\mathbf{X}^i) (\Delta A_0)_i \quad (37)$$

is exact for all functions, $f \in \Pi_N = \text{span} \{Y_\ell^k : \ell = 0, 1, \dots, N, k = -\ell, \dots, 0, \dots, \ell\}$. Here Π_N is the $(N+1)^2$ dimensional space of spherical harmonics up to degree N , defined as

$$Y_\ell^k(\lambda, \theta) = \begin{cases} \sqrt{\frac{(2\ell+1)(\ell-k)!}{4\pi(\ell+k)!}} P_\ell^k(\sin \theta) \cos(k\lambda) & k \geq 0 \\ \sqrt{\frac{(2\ell+1)(\ell+k)!}{4\pi(\ell-k)!}} P_\ell^{-k}(\sin \theta) \sin(-k\lambda) & k < 0, \end{cases} \quad (38)$$

where P_ℓ^k is an associated Legendre function of degree ℓ and order k . Let $f(\mathbf{X}) = f(\lambda, \theta) = Y_0^0(\lambda, \theta)Y_\ell^k(\lambda, \theta)$. Then to satisfy Eq. (37),

$$\int_{\mathbb{S}^2} f(\mathbf{X}) d\mathbf{x} = \iint_{\mathbb{S}^2} Y_0^0(\lambda, \theta) Y_\ell^k(\lambda, \theta) dA_0 \quad (39)$$

$$= \sum_{i=1}^n Y_0^0(\lambda_i, \theta_i) Y_\ell^k(\lambda_i, \theta_i) (\Delta A_0)_i \quad (40)$$

$$= \frac{1}{\sqrt{4\pi}} \sum_{i=1}^n Y_\ell^k(\lambda_i, \theta_i) (\Delta A_0)_i. \quad (41)$$

The last equality is due to $Y_0^0 = \frac{1}{\sqrt{4\pi}}$. Since spherical harmonics form an orthonormal set,

$$\int_{\mathbb{S}^2} f(\mathbf{X}) d\mathbf{x} = \iint_{\mathbb{S}^2} Y_0^0(\lambda, \theta) Y_\ell^k(\lambda, \theta) dA_0 = \begin{cases} 1 & \ell = 0 \\ 0 & \ell \neq 0. \end{cases} \quad (42)$$

Equating Eqs. (41) and (42), we have that

$$\sum_{i=1}^n Y_\ell^k(\lambda_i, \theta_i) (\Delta A_0)_i = \begin{cases} \sqrt{4\pi} & \ell = 0 \\ 0 & \ell \neq 0. \end{cases} \quad (43)$$

Let the matrix

$$\mathbf{Y} = \begin{pmatrix} Y_0^0(\lambda_1, \theta_1) & Y_1^0(\lambda_1, \theta_1) & Y_1^{-1}(\lambda_1, \theta_1) & Y_1^1(\lambda_1, \theta_1) & \dots \\ Y_0^0(\lambda_2, \theta_2) & Y_1^0(\lambda_2, \theta_2) & Y_1^{-1}(\lambda_2, \theta_2) & Y_1^1(\lambda_2, \theta_2) & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ Y_0^0(\lambda_n, \theta_n) & Y_1^0(\lambda_n, \theta_n) & Y_1^{-1}(\lambda_n, \theta_n) & Y_1^1(\lambda_n, \theta_n) & \dots \end{pmatrix} \in \mathbb{C}^{n \times (N+1)^2}. \quad (44)$$

Then in Eq. (43) we seek the nonnegative least squares solution to

$$\mathbf{Y}^* \Delta \mathbf{A}_0 = \sqrt{4\pi} \mathbf{e}_1, \quad (45)$$

where $\mathbf{e}_1 = (1 \ 0 \ \dots \ 0)^T$ is an $(N+1)^2$ dimensional vector. The nonnegative least squares solution for $\Delta \mathbf{A}_0$ gives the area weights for Eq. (37).

To solve Eq. (45) for $\Delta \mathbf{A}_0$, we use the available implementation from [30] to obtain the nonnegative area weights $\Delta \mathbf{A}_0$ for a unit sphere. Although any point set can be used in Eq. (45), we let $n = (N+1)^2$ and choose evaluation points that maximize the determinant of the Gram matrix $\mathbf{Y}\mathbf{Y}^*$. For these “maximal determinant” (MD) points, which are discussed in [31, 32] and can be downloaded from [33], it has been conjectured that the quadrature weights satisfying the equality in Eq. (45) exist and are positive for all N [32].

Choosing weights that satisfy Eq. (45) ensures that the integral in Eq. (37) is exact for functions in Π_N , the space of spherical harmonics up to degree N . However, the calculated weights can still be used to

integrate any function on the sphere. In fact, the force function in Eq. (35) is not necessarily in Π_N since it involves derivatives of arbitrary position functions. A quadrature error is therefore introduced. In [32], it is shown that the quadrature error, ϵ , in a certain Hilbert space is bounded by

$$\epsilon \leq \frac{\sqrt{4\pi}}{(N+1)^{1/2}} = \frac{\sqrt{4\pi}}{n^{1/4}}, \quad (46)$$

where the last equality holds only in the case that $n = (N+1)^2$. For MD points, the measured worst case error is much better; it is $\mathcal{O}(n^{-3/4})$ (see [32], Table 2). Our observations in Section 4.1 confirm this, as the actual quadrature error in integrating the force function was found to be several orders of magnitude lower than the theoretical bound, even for a highly irregular shape.

2.4.2. Weights on an arbitrary reference configuration

We use the formulation from [34] to determine the quadrature weights on an arbitrary configuration. Let \mathbb{L} denote a space diffeomorphic to \mathbb{S}^2 , and let $\mathbf{Z}_{\mathbb{S}^2}(\lambda, \theta)$ and $\mathbf{Z}_{\mathbb{L}}(\lambda, \theta)$ be the mappings from Eq. (9) that describe $(\hat{X}, \hat{Y}, \hat{Z})$ from (λ, θ) on each space. Then, as shown in [34], the surface elements for each configuration are given by

$$dA_{\mathbb{L}} = \sqrt{\det(\mathcal{G}_0(\mathbf{Z}_{\mathbb{L}}))} d\lambda d\theta \quad (47)$$

and

$$dA_{\mathbb{S}^2} = \sqrt{\det(\mathcal{G}_0(\mathbf{Z}_{\mathbb{S}^2}))} d\lambda d\theta, \quad (48)$$

where \mathcal{G}_0 is evaluated for each surface according to Eq. (13). From Eqs. (47) and (48), it is clear that

$$dA_{\mathbb{L}} = \sqrt{\frac{\det(\mathcal{G}_0(\mathbf{Z}_{\mathbb{L}}))}{\det(\mathcal{G}_0(\mathbf{Z}_{\mathbb{S}^2}))}} dA_{\mathbb{S}^2}. \quad (49)$$

Thus the analogous integral to Eq. (36) is

$$\int_{\mathbb{L}} f(\mathbf{X}) dA_{\mathbb{L}} = \int_{\mathbb{S}^2} f(\mathbf{X}) \sqrt{\frac{\det(\mathcal{G}_0(\mathbf{Z}_{\mathbb{L}}))}{\det(\mathcal{G}_0(\mathbf{Z}_{\mathbb{S}^2}))}} dA_{\mathbb{S}^2} = \sum_{i=1}^n f(\mathbf{X}^i) \sqrt{\frac{\det(\mathcal{G}_0(\mathbf{Z}_{\mathbb{L}}))}{\det(\mathcal{G}_0(\mathbf{Z}_{\mathbb{S}^2}))}} (\Delta A_0)_i. \quad (50)$$

According to Eq. (50), the weights for the arbitrary reference configuration are given by

$$\Delta \mathbf{A}_{\mathbb{L}} = \sqrt{\frac{\det(\mathcal{G}_0(\mathbf{Z}_{\mathbb{L}}))}{\det(\mathcal{G}_0(\mathbf{Z}_{\mathbb{S}^2}))}} \Delta \mathbf{A}_{\mathbb{S}^2}. \quad (51)$$

For example, if the reference configuration is a sphere of radius r , then $\det(\mathcal{G}_0(\mathbf{Z}_{\mathbb{L}})) = r^4 \det(\mathcal{G}_0(\mathbf{Z}_{\mathbb{S}^2}))$, so $\Delta \mathbf{A}_{\mathbb{L}} = r^2 \Delta \mathbf{A}_{\mathbb{S}^2}$, as expected since the surface area of a sphere scales by r^2 .

With the area weights from Eq. (51), we have finally found the *force*, $\hat{\mathbf{F}}$ at point i from the force density \mathbf{F} as a function of position in (λ, θ) .

$$\hat{\mathbf{F}}_i(\lambda, \theta) = \mathbf{F}_i(\lambda, \theta)(\Delta A_0)_i. \quad (52)$$

2.5. The linear discrete surface method (LDSM)

In LDSM (first introduced in [26]), the surface of the shell is triangulated and the integral for energy is discretized as such. Consider an energy functional from Eq. (6) of the form

$$\mathcal{E} = \int_{\Gamma_0} W(I_1, I_2) dA_0. \quad (53)$$

The above integral can be discretized into a sum over the surface triangles as follows,

$$\mathcal{E} = \sum_{\text{tri}} \int_{\text{tri}} W(I_1, I_2) dA_0. \quad (54)$$

LDSM uses barycentric coordinates $(\gamma_1, \gamma_2, \gamma_3)$ to represent a deformed triangle t with vertices $\mathbf{X}^1, \mathbf{X}^2, \mathbf{X}^3$ in the deformed configuration that corresponds to a reference triangle s with vertices $\mathbf{Z}^1, \mathbf{Z}^2, \mathbf{Z}^3$ in the undeformed configuration. Any point within the triangle can be written as

$$\mathbf{X} = \gamma_1 \mathbf{X}^1 + \gamma_2 \mathbf{X}^2 + \gamma_3 \mathbf{X}^3 \quad (55)$$

and

$$\mathbf{Z} = \gamma_1 \mathbf{Z}^1 + \gamma_2 \mathbf{Z}^2 + \gamma_3 \mathbf{Z}^3, \quad (56)$$

where $\gamma_1 + \gamma_2 + \gamma_3 = 1$ and $\gamma_1, \gamma_2, \gamma_3 \geq 0$. Using the relation $\gamma_3 = 1 - \gamma_1 - \gamma_2$, the surface can be represented in terms of two material coordinates (note that, for $\gamma_3 \geq 0$ to be satisfied, $\gamma_1 \leq 1 - \gamma_2$). As in the continuous case, we define the matrices

$$\nabla_{\mathbf{q}} \mathbf{X} = \begin{pmatrix} \frac{\partial \mathbf{X}}{\partial \gamma_1} & \frac{\partial \mathbf{X}}{\partial \gamma_2} \end{pmatrix} = (\mathbf{X}^1 - \mathbf{X}^3 \quad \mathbf{X}^2 - \mathbf{X}^3) \quad (57)$$

and

$$\nabla_{\mathbf{q}} \mathbf{Z} = \begin{pmatrix} \frac{\partial \mathbf{Z}}{\partial \gamma_1} & \frac{\partial \mathbf{Z}}{\partial \gamma_2} \end{pmatrix} = (\mathbf{Z}^1 - \mathbf{Z}^3 \quad \mathbf{Z}^2 - \mathbf{Z}^3). \quad (58)$$

Denote $\mathbf{X}^i - \mathbf{X}^j = \mathbf{X}^{ij}$ (and likewise for \mathbf{Z}). Then we can again define the metric tensors

$$\mathcal{G} = (\nabla_{\mathbf{q}} \mathbf{X})^T (\nabla_{\mathbf{q}} \mathbf{X}) = \begin{pmatrix} \mathbf{X}^{13} \cdot \mathbf{X}^{13} & \mathbf{X}^{13} \cdot \mathbf{X}^{23} \\ \mathbf{X}^{23} \cdot \mathbf{X}^{13} & \mathbf{X}^{23} \cdot \mathbf{X}^{23} \end{pmatrix} \quad (59)$$

and

$$\mathcal{G}_0 = (\nabla_{\mathbf{q}} \mathbf{Z})^T (\nabla_{\mathbf{q}} \mathbf{Z}) = \begin{pmatrix} \mathbf{Z}^{13} \cdot \mathbf{Z}^{13} & \mathbf{Z}^{13} \cdot \mathbf{Z}^{23} \\ \mathbf{Z}^{23} \cdot \mathbf{Z}^{13} & \mathbf{Z}^{23} \cdot \mathbf{Z}^{23} \end{pmatrix}. \quad (60)$$

The tensors \mathcal{G} and \mathcal{G}_0 are *constant* on any triangle t or s . Therefore, the tensor $\mathcal{C} = \mathcal{G} \mathcal{G}_0^{-1}$ and its invariants I_1 and I_2 are constant on t and s . Thus any energy density functional defined in terms of the invariants is also constant on t and s and can be defined in terms of the vertices $\mathbf{X}^1, \mathbf{X}^2, \mathbf{X}^3$, $\mathbf{Z}^1, \mathbf{Z}^2$, and \mathbf{Z}^3 . Denote the energy density of a triangle as $W(t, s)$ (it is a function of the deformed triangle t and undeformed triangle s). The discretized energy is then

$$\mathcal{E} = \sum_{\text{tri}} \int_s W(I_1, I_2) dA_0 = \sum_{\text{tri}} \int_s W(t, s) (\det \mathcal{G}_0)^{1/2} d\gamma_1 d\gamma_2. \quad (61)$$

Note that the integration is carried out over the *reference* triangle s and that $dA_0 = (\det \mathcal{G}_0)^{1/2} d\gamma_1 d\gamma_2$ since $(\det \mathcal{G}_0)^{1/2}$ is the magnitude of the normal to the reference triangle [12]. Evaluating the integral, we have

$$\mathcal{E} = \sum_{\text{tri}} W(t, s) (\det \mathcal{G}_0)^{1/2} \int_s d\gamma_2 d\gamma_1, \quad (62)$$

$$\mathcal{E} = \sum_{\text{tri}} W(t, s) (\det \mathcal{G}_0)^{1/2} \int_0^1 \int_0^{1-\gamma_1} d\gamma_2 d\gamma_1, \quad (63)$$

$$\mathcal{E} = \frac{1}{2} \sum_{\text{tri}} W(t, s) (\det \mathcal{G}_0)^{1/2}. \quad (64)$$

From the above equation, note that $\Delta A_0(s) = \frac{1}{2} (\det \mathcal{G}_0)^{1/2}$, which is expected *a priori* since it is simply the formula for the area of the reference triangle s . Since the energy functional has already been integrated,

the area weights at each point have already been accounted for, and the *force* (not force density), $\hat{\mathbf{F}}$, at a point \mathbf{X}^i due to triangle t can be computed by taking the derivative of the energy with respect to that point,

$$\hat{\mathbf{F}}_i(t) = -\frac{\partial \mathcal{E}}{\partial \mathbf{X}^i}(t) = -\frac{\partial}{\partial \mathbf{X}^i} \left(\frac{1}{2} W(t, s) (\det \mathcal{G}_0)^{1/2} \right). \quad (65)$$

The force at each point is then the sum of the force contributions from each triangle that includes the point, i.e.

$$\hat{\mathbf{F}}_i(\mathbf{X}^i) = \sum_{t \ni \mathbf{X}^i} -\frac{\partial \mathcal{E}}{\partial \mathbf{X}^i}(t) = \sum_{t \ni \mathbf{X}^i} -\frac{\partial}{\partial \mathbf{X}^i} \left(\frac{1}{2} W(t, s) (\det \mathcal{G}_0)^{1/2} \right). \quad (66)$$

It can be shown using vector calculus that the sum of the forces over the points in LDSM is zero. The force in Eq. (66) therefore satisfies the discrete integral constraint for Stokes flow. The force density per unit undeformed area is computed by dividing the force at each point by its associated area weight in the undeformed configuration. The area weight at point i in LDSM is given by 1/3 of the sum of the reference triangle areas that include point i . That is,

$$\Delta(A_{L_0})_i = \frac{1}{3} \sum_{s \ni \mathbf{Z}^i} (\det \mathcal{G}_0(s))^{1/2}. \quad (67)$$

Further details are located in [12].

2.6. Energy functionals

In this section, we describe energy functionals for neo-Hookean energy, surface tension, and bending (or curvature) energy.

2.6.1. neo-Hookean energy

We follow Evans & Skalak's formulation for the neo-Hookean energy density [35] (for other possible neo-Hookean constitutive laws, see [18]). In terms of the invariants,

$$W_{NH} = G_s \left(\frac{I_1 + 2}{2\sqrt{I_2 + 1}} - 1 + \frac{A}{2} \left(I_2 + 2 - 2\sqrt{I_2 + 1} \right) \right) \quad (68)$$

$$= G_s \left(\frac{\text{tr } \mathcal{C}}{2(\det \mathcal{C})^{1/2}} - 1 + \frac{A}{2} \left(\det \mathcal{C} + 1 - 2(\det \mathcal{C})^{1/2} \right) \right) \quad (69)$$

$$= G_s \left(\frac{\text{tr } \mathcal{C}}{2(\det \mathcal{C})^{1/2}} - 1 + \frac{A}{2} \left((\det \mathcal{C})^{1/2} - 1 \right)^2 \right) \quad (70)$$

$$= \frac{K}{2} \left((\det \mathcal{C})^{1/2} - 1 \right)^2 + \frac{G_s}{2} \left(\frac{\text{tr } \mathcal{C}}{(\det \mathcal{C})^{1/2}} - 2 \right), \quad (71)$$

where $K = AG_s$ is the area dilation modulus and G_s is the shear modulus. Eq. (71) gives the form used in [12]. We take derivatives of Eq. (68) with respect to I_1 and I_2 to find the tensors \mathcal{P} , \mathcal{S} , and force density \mathbf{F} in Eq. (35).

2.6.2. Surface energy

The most common representation of "surface energy" is $E_{ST} = \sigma \int dA$, where dA is the area element in the *current* configuration and σ is the surface tension (units force/length) of the surface. For both the variational force density method and LDSM, $dA = (\det \mathcal{G})^{1/2} d\lambda d\theta$, but this must be expressed with respect to the integral over the reference area to satisfy our method. Therefore we use Eq. (50) and [34] to write

$$\mathcal{E}_{ST} = \int_{\Gamma} \sigma dA = \int_{\Gamma_0} \sigma \left(\frac{\det \mathcal{G}}{\det \mathcal{G}_0} \right)^{1/2} dA_0 = \int_{\Gamma_0} \sigma (\det \mathcal{C})^{1/2} dA_0. \quad (72)$$

We can then express the surface energy in the form of Eq. (16) with $W_{ST} = \sigma (\det \mathcal{C})^{1/2} = \sigma (I_2 + 1)^{1/2}$. Derivatives of this density with respect to I_1 and I_2 can easily be calculated to evaluate the first and second Piola-Kirchhoff stress tensors \mathcal{P} , \mathcal{S} and force density \mathbf{F} in Eq. (35).

The variational derivative can be carried out over the current configuration as well to yield force density per unit deformed area [15, 36, 37]. When this formulation, which is related directly to the mean curvature of the surface, is used, a transformation must be made to express the force per unit reference area.

$$\mathbf{F}_{cur}^{ST} dA = \sigma(2H) \hat{\mathbf{n}} dA = \sigma(2H) \hat{\mathbf{n}} \left(\frac{\det \mathcal{G}}{\det \mathcal{G}_0} \right)^{1/2} dA_0. \quad (73)$$

The force per unit undeformed configuration then is,

$$\mathbf{F}^{ST} = \sigma(2H) \hat{\mathbf{n}} \left(\frac{\det \mathcal{G}}{\det \mathcal{G}_0} \right)^{1/2}, \quad (74)$$

where $\hat{\mathbf{n}}$ is the unit normal vector in the current configuration,

$$\hat{\mathbf{n}} = \frac{\frac{\partial \mathbf{X}}{\partial \lambda} \times \frac{\partial \mathbf{X}}{\partial \theta}}{\left\| \frac{\partial \mathbf{X}}{\partial \lambda} \times \frac{\partial \mathbf{X}}{\partial \theta} \right\|}. \quad (75)$$

H is the mean curvature (one half the total curvature, or the average of the two principle curvatures) given by the formula

$$H = \frac{eG - 2fF + gE}{2(EG - F^2)}, \quad (76)$$

where E , F , and G are coefficients of the first fundamental form

$$E = \frac{\partial \mathbf{X}}{\partial \lambda} \cdot \frac{\partial \mathbf{X}}{\partial \lambda} \quad F = \frac{\partial \mathbf{X}}{\partial \lambda} \cdot \frac{\partial \mathbf{X}}{\partial \theta} \quad G = \frac{\partial \mathbf{X}}{\partial \theta} \cdot \frac{\partial \mathbf{X}}{\partial \theta}, \quad (77)$$

and e , f , and g are coefficients of the second fundamental form,

$$e = \frac{\partial^2 \mathbf{X}}{\partial \lambda^2} \cdot \hat{\mathbf{n}} \quad f = \frac{\partial^2 \mathbf{X}}{\partial \lambda \partial \theta} \cdot \hat{\mathbf{n}} \quad g = \frac{\partial^2 \mathbf{X}}{\partial \theta^2} \cdot \hat{\mathbf{n}}. \quad (78)$$

It can be shown using elementary vector calculus that Eq. (74) is equivalent to Eq. (35) when $W_{ST} = \sigma(I_2 + 1)^{1/2}$ is used. We emphasize that for this simple case, we were able to take a surface energy given over the current configuration and transform it so we could integrate over the reference configuration and apply our method.

More complex energy functionals given over the current configuration are not as easily transformed. In these cases, differential geometry can be used to find the variational derivatives and force per unit deformed area. These expressions can then be transformed to the undeformed configuration by multiplying by the ratio of area weights as in Eq. (74). Bending energy is an example of such an energy functional and is described in the next section.

2.6.3. Bending energy

In this section, we describe the the variational formulation of forces due to bending. It is worth noting that piecewise linear representations of the surface (such as LDSM) must use a method different from that of Section 2.5, where the energy is evaluated on each triangle, to compute forces due to bending. This is because bending energies are directly related to curvature, and in a piecewise linear surface representation the triangles have zero curvature. In the case of [12], the method used to compute forces due to bending does not ensure that the bending force integrates to zero (despite the bending energy being translation invariant), and therefore cannot be used in the context of Stokes flow. Our method allows for the true bending energy functional to be used because we can properly resolve the curvature of the surface. In

addition, the use of the correct quadrature weights ensures that the bending force integrates to zero over the fluid domain.

Interestingly, variational principles have been used to derive forces due to bending rigidity for quite some time, with one early example from [38]. These previous studies addressed the problem of determining the equilibrium shape of membranes experiencing bending forces by setting the force density equal to zero. The derivation of the force density due to bending begins with the energy functional

$$\mathcal{E}_{bend} = k_{bend} \int_{\Gamma} (2H)^2 dA. \quad (79)$$

where k_{bend} is the bending rigidity and the energy is an integral of the total curvature $(2H)$ over the deformed configuration. The variational approach calls for the energy functional to be written in the form

$$\delta \mathcal{E}_{bend} = \int_{\Gamma} -\mathbf{F}_{cur}^{BR} \cdot \delta \mathbf{X} dA. \quad (80)$$

As shown in [36, 39], arguments from differential geometry can be used to write the force density per unit deformed configuration as

$$\mathbf{F}_{cur}^{BR} = -k_{bend} (4\nabla_s^2 H + 8H^3 - 4HR) \hat{\mathbf{n}}, \quad (81)$$

where H is the mean curvature defined according to Eq. (76), $\hat{\mathbf{n}}$ is the unit normal defined in Eq. (75), and ∇_s^2 is the surface Laplacian in the deformed state, defined as

$$\nabla_s^2 = \sum_{i=1}^2 \sum_{j=1}^2 \frac{1}{\sqrt{\det \mathcal{G}}} \frac{\partial}{\partial q_i} \left(\sqrt{\det \mathcal{G}} g_{ij}^{-1} \frac{\partial}{\partial q_j} \right). \quad (82)$$

Finally, R is the scalar curvature, or twice the Gaussian curvature, defined as

$$R = 2 \det \left(\mathcal{G}^{-1} \begin{pmatrix} e & f \\ f & g \end{pmatrix} \right), \quad (83)$$

where the fundamental forms e , f , and g are defined in Eq. (78). In a similar manner to that of Eq. (74), Eq. (81) can be written per unit undeformed area as

$$\mathbf{F}^{BR} = -k_{bend} \left(\frac{\det \mathcal{G}}{\det \mathcal{G}_0} \right)^{1/2} (4\nabla_s^2 H + 8H^3 - 4HR) \hat{\mathbf{n}}, \quad (84)$$

which gives the force due to bending per unit undeformed area.

3. Numerical Formulation

In this section, we describe the discretization of the surface and fluid grids. We then describe the fluid solver and temporal update for dynamic simulations.

3.1. Lagrangian surface and force discretization

For the variational force density method, some mapping must be chosen for \mathbf{X} in Eq. (9). Our method allows any mapping that defines a closed surface to be used, including interpolants composed of Lagrange, spherical harmonic, or radial basis functions. We choose to use a spherical harmonic interpolant for the maps X , Y , and Z in Eq. (9) (the reference configuration \mathbf{Z} is generally taken to be the unit sphere unless otherwise specified). Each interpolant is computed in the same manner and is also described in [15]. Suppose we have a set of $m = (M + 1)^2$ points in (x, y, z) space. Then we use a spherical harmonic interpolant of order M to approximate the function $\mathbf{X}(\lambda, \theta)$ in Eq. (9). For example,

$$X(\lambda, \theta) = \sum_{\ell=0}^M \left(\sum_{k=-\ell}^{-1} c_{\ell,k}^x Y_{\ell}^k(\lambda, \theta) + \sum_{k=0}^{\ell} c_{\ell,k}^x Y_{\ell}^k(\lambda, \theta) \right), \quad (85)$$

where the spherical harmonics are as defined in Eq. (38) and the other interpolants Y , Z , \hat{X} , \hat{Y} , and \hat{Z} are defined similarly. In [15], it is shown that Eq. (85) corresponds to a linear system

$$\mathbf{Y}\mathbf{c}^x = \begin{pmatrix} Y_0^0(\lambda_1, \theta_1) & Y_1^0(\lambda_1, \theta_1) & Y_1^{-1}(\lambda_1, \theta_1) & Y_1^1(\lambda_1, \theta_1) & \dots \\ Y_0^0(\lambda_2, \theta_2) & Y_1^0(\lambda_2, \theta_2) & Y_1^{-1}(\lambda_2, \theta_2) & Y_1^1(\lambda_2, \theta_2) & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ Y_0^0(\lambda_m, \theta_m) & Y_1^0(\lambda_m, \theta_m) & Y_1^{-1}(\lambda_m, \theta_m) & Y_1^1(\lambda_m, \theta_m) & \dots \end{pmatrix} \begin{pmatrix} c_1^x \\ c_2^x \\ \vdots \\ c_m^x \end{pmatrix} = \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_m \end{pmatrix}, \quad (86)$$

where $m = (M+1)^2$ exactly and X_1, X_2, \dots, X_m are the x coordinates of the first, second, and m th points, respectively. Since the coordinates of the interpolation points are constant throughout the algorithm, the LU factorization of the matrix \mathbf{Y} in Eq. (86) can be precomputed at the start of the algorithm. Then at each timestep, $\mathcal{O}(m^2)$ flops are required to solve Eq. (86).

For the variational force density method, we follow [15] and define the m points in (x, y, z) space that are used to construct the interpolant as *interpolation* points. Note that there are typically much fewer interpolation points than evaluation points ($m \ll n$). Because the matrix in Eq. (86) must be invertible, we use the maximal determinant points mentioned previously (downloaded from [33]) that maximize the determinant of the Gram matrix $\mathbf{Y}\mathbf{Y}^*$. See [31] for more details on choosing points on a sphere for interpolation.

The interpolant from Eq. (85) is the map $\mathbf{X}(\lambda, \theta)$ in Eq. (9), and given a reference configuration $\mathbf{Z}(\lambda, \theta)$ we proceed in computing force densities from the variational method in Eq. (35) at a different set of n maximal determinant evaluation points. We find the spherical coordinates of the evaluation points and use the function \mathbf{X} to map them to our deformed configuration, where we evaluate the forces using Eq. (35) with the area weights from Eq. (45). The calculation of the forces in Eq. (35) requires the derivatives of \mathbf{X} at each of the evaluation points. In practice, this is accomplished by multiplying matrices of spherical harmonic derivatives (e.g. $\frac{\partial \mathbf{Y}}{\partial \lambda}$, where the derivatives are taken on each element) by the vector of function weights (\mathbf{c}^j). Each derivative is thus a matrix-vector multiplication problem and requires $\mathcal{O}(mn)$ operations, although this could easily be parallelized to reduce computational time. Once the derivatives are computed, $\mathcal{O}(n)$ operations are required to compute the forces in Eq. (35) from the derivative values (this step could also easily be done in parallel). The leading order term in the flop count for the force calculation in our method is therefore $\mathcal{O}(mn)$.

In LDSM, the force is computed directly at each of the evaluation points using Eq. (66) with the triangles from the surface discretization. Since the force is computed triangle by triangle and there are approximately $2n$ triangles, the entire force calculation for LDSM is $\mathcal{O}(n)$. We perform timing tests in Section 4.2 to compare more practically the computational costs of our method with LDSM.

Each of the weights $(\Delta A_0)_i$ in Eq. (45) is computed for the n evaluation points and multiplied by its respective force density to get a total force at each of the evaluation points. Since the continuous force density representation is exact for a given position function \mathbf{X} , the only errors introduced are in representing the surface by a continuously differentiable spherical harmonic interpolant and in using a quadrature rule to integrate the continuous force density function. A detailed study of the errors associated with a spherical harmonic representation can be found in [15]. An analytic estimate for the quadrature error is located in [32]. We also analyze the error in force computations and compare it to the error in LDSM in Section 4.1.

3.2. Eulerian discretization

The fluid domain is discretized on a periodic three dimensional domain $[-L, L] \times [-L, L] \times [-L, L]$ with mesh spacing $\Delta x = \Delta y = \Delta z = 2L/\eta$, where η is the grid size. We use a spectral fluid solver to compute the pressure and velocity at each time step in $\mathcal{O}(\eta^3 \log \eta^3)$ operations [40]. The evaluation points are chosen so that approximately two Lagrangian points lie within an Eulerian cube of dimension Δx^3 .

3.3. Time update

We outline the time stepping procedure for the variational derivative method first. Given a set of m interpolation points and n evaluation points on the reference configuration, and the map $\mathbf{Z}(\lambda, \theta)$ in Eq. (9),

the area weights $\Delta \mathbf{A}_0$ in Eq. (45), and matrix \mathbf{Y} in Eq. (86) (and its LU factorization) are precomputed. Then at the k th time step,

1. The m interpolation points in the current configuration (x_i, y_i, z_i) are used to solve Eq. (86) for the map \mathbf{X} ($\mathcal{O}(m^2)$ operations).
2. The computed maps \mathbf{X} and \mathbf{Z} are used to find the force densities in Eq. (35) at each of the n evaluation points ($\mathcal{O}(mn)$ operations).
3. The forces are spread onto the Eulerian grid using the discrete delta function from [25] with the area weights from Eq. (45) that ensure the spread force integrates to zero ($\mathcal{O}(4^3n)$ operations because we are using a 4 point discrete delta function).
4. The Stokes equations are solved on the Eulerian grid ($\mathcal{O}(\eta^3 \log \eta^3)$ operations).
5. The fluid velocity is interpolated back to the structure at the m interpolation points ($\mathcal{O}(4^3m)$ operations).
6. The interpolation points are updated with the local fluid velocity,

$$\mathbf{X}_i^{k+1} = \mathbf{X}_i^k + \Delta t \mathbf{U}^{k+1}, \quad (87)$$

where \mathbf{U}^{k+1} indicates the interpolated fluid velocity from step 5 ($\mathcal{O}(m)$ operations).

In LDSM, we work with only the set of n evaluation points. We begin with a triangulated surface and use the locations of the points in the current and reference configurations to compute the forces, then proceed with steps 3-6 as above, with the exception being that the structure velocity is computed at the same n points where the force was evaluated. The interpolation of the velocity in LDSM therefore requires $\mathcal{O}(4^3n)$ operations instead of $\mathcal{O}(4^3m)$, and the forward Euler update is $\mathcal{O}(n)$ in LDSM instead of $\mathcal{O}(m)$ in our method. If $m \ll n$, our method results in fewer operations in steps 5 and 6.

4. Results

In this section we test the accuracy of LDSM and the spherical harmonic/variational derivative (SHVD) approaches in the context of both force computations and dynamic immersed boundary simulations. We first consider the accuracy of the force computation on an ellipsoid and perturbed ellipsoid and proceed with dynamic simulations on each surface.

4.1. Accuracy of force computations

We consider the accuracy of surface tension and neo-Hookean forces on two surfaces given by

$$\text{Ellipsoid: } \mathbf{X} = \begin{pmatrix} a \cos \lambda \cos \theta \\ b \sin \lambda \cos \theta \\ c \sin \theta \end{pmatrix} \quad (88)$$

and

$$\text{"Perturbed ellipsoid:" } \mathbf{X} = V_f \begin{pmatrix} a \left(1 + \frac{B}{5} \exp(-\sin \theta) \right) \cos \lambda \cos \theta \\ b (1 + B \exp(-\sin \theta)) \sin \lambda \cos \theta \\ c (1 + B \exp(-\sin \theta)) \sin \theta \end{pmatrix}. \quad (89)$$

We use parameters $a = 1.1$, and $b = c = \frac{1}{\sqrt{1.1}}$ in Eq. (88) and $a = 0.1$, $b = c = 0.2$, and $B = 0.25$ in Eq. (89). In both cases, the reference configuration is the unit sphere, $\mathbf{Z} = (\cos \lambda \cos \theta, \sin \lambda \cos \theta, \sin \theta)$, and the parameter V_f in Eq. (89) is set so that the volume of the perturbed ellipsoid is the same as that of the unit sphere (in our case, $V_f \approx 5.14$). For simplicity, we also let $\sigma = K = A = G_s = 1$ in the force and energy functionals in Eqs. (68) and (74). The shapes are shown in Fig. 1 along with the positions of the set of 529 evaluation points (sets with a higher number of evaluation points are similarly distributed).

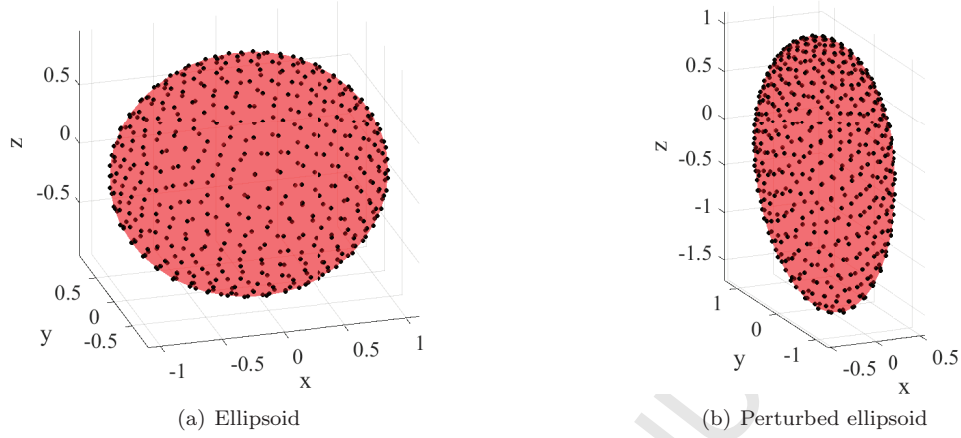


Figure 1: The shapes for study. 529 evaluation points are shown. The distribution of other sets of evaluation points is similar.

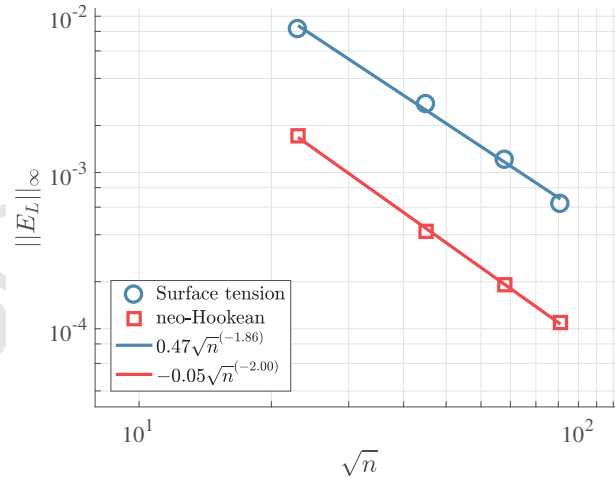


Figure 2: Errors in surface tension and neo-Hookean forces on an ellipsoid. $\|E_L\|_\infty$, the ℓ_∞ error in LDSM forces, is shown for the cases of 529, 2025, 4624, and 8281 points along with a power-law fit of the data. Because the spherical harmonic interpolant from Eq. (85) is exact for an ellipsoid, forces computed from SHVD are exact and $\|E_S\| = 0$.

We fix a number of evaluation points so that the quadrature error given by the weights in Eq. (45) is on the order 10^{-5} or less. The quadrature error is measured by the value of the discrete force integral since the exact integral of the translation invariant continuous force function evaluates to zero. We then compute the exact solution for the neo-Hookean or surface tension force densities at the n evaluation points using Eq. (35) to take the variational derivative of the *exact* mappings \mathbf{X} from Eqs. (88) or (89).

Next, we construct a spherical harmonic interpolant for the surface of the form in Eq. (85) using some number m of interpolation points. We then use this interpolant to compute the force densities in Eq. (35). We multiply both the exact force density and the force density from SHVD by the weights in Eq. (45) to get a force. In this way, we isolate the error from representing the surface with a spherical harmonic interpolant. Finally, we compute the LDSM forces by evaluating Eq. (66) with the triangle vertex locations given by Eq. (88) or (89).

We denote the exact force (not force density) as $\hat{\mathbf{F}}_E$, the force from SHVD as $\hat{\mathbf{F}}_S$ and the force from LDSM as $\hat{\mathbf{F}}_L$. We quantify the error between the models via the discrete ℓ_∞ norm. The norm of the errors from SHVD and LDSM, denoted by \mathbf{E}_S and \mathbf{E}_L , respectively, are given by

$$\|\mathbf{E}_S\|_\infty = \max_i \|\hat{\mathbf{F}}_S - \hat{\mathbf{F}}_E\|_\infty \quad (90)$$

and

$$\|\mathbf{E}_L\|_\infty = \max_i \|\hat{\mathbf{F}}_L - \hat{\mathbf{F}}_E\|_\infty. \quad (91)$$

The error in force is computed as the maximum over the n evaluation points, where i is the index that runs from 1 to n . In all convergence plots, we plot the error vs. \sqrt{m} or \sqrt{n} . Since $(dA_0)_i \approx 4\pi/n$, the number of points is proportional to the square of the length scale. We plot convergence in the length scale by taking the square root of the number of points.

Because the same area weights are used for the calculation of $\hat{\mathbf{F}}_E$ and $\hat{\mathbf{F}}_S$, the measured error in Eq. (90) is independent of the quadrature weights. Since LDSM uses different area weights, the measured error in Eq. (91) is affected by both the quadrature error in $\hat{\mathbf{F}}_E$ and the discretization error in the LDSM force calculation. We define $\hat{\mathbf{F}}_E$ to be the exact solution if the quadrature error (i.e. the distance of the force integral from zero) is negligible relative to the LDSM surface discretization error. We only consider surfaces where the quadrature error is $\mathcal{O}(10^{-5})$ or less. In our numerical tests, the LDSM error graphed in Figs. 2 and 3(a) is larger than this value so that quadrature error is negligible.

Fig. 2 shows the errors in Eqs. (90) and (91) for the forces on the ellipsoid as a function of the number of evaluation points, for $n = 529, 2025, 4624$, and 8281 . We note that for an ellipsoid in Eq. (88), the spherical harmonic interpolant with as little as 4 interpolation points gives the same mapping as Eq. (88). Therefore, the forces from SHVD are exact for an ellipsoid and are not included in Fig. 2. Consequently, we show in Fig. 2 only the errors in LDSM as a function of the number of evaluation points for both surface tension and neo-Hookean forces. A piecewise planar representation of a surface converges to a smooth surface with second order accuracy, so we expect at most second order accurate LDSM forces. We observe a second order convergence rate in the forces with respect to the length dimension (see power function fit of data in Fig. 2). We observe a similar convergence rate for computing surface tension on a sphere with LDSM (data not shown).

Our study of the perturbed ellipsoid shows that forces from SHVD converge to the exact solution with spectral accuracy in contrast to LDSM. Fig. 3 shows the errors in surface tension and neo-Hookean forces for LDSM with varying numbers of evaluation points n (Fig. 3(a)) and SHVD with varying numbers of interpolation points m (Fig. 3(b)). A convergence rate between first and second order is observed for LDSM. In contrast, the force from SHVD converges spectrally as the number of interpolation points m increases regardless of the number of evaluation points n [15]. The force estimate from SHVD is orders of magnitude smaller than LDSM when $m \geq 81$. This stems from the fact that SHVD relies on a *continuous* surface representation, and is therefore able to effectively capture the geometrical information of the surface necessary to accurately compute forces.

We comment that the error in forces from SHVD is from both interpolation error and aliasing, a well known phenomenon in which coefficients of higher order spherical harmonics are folded into lower order

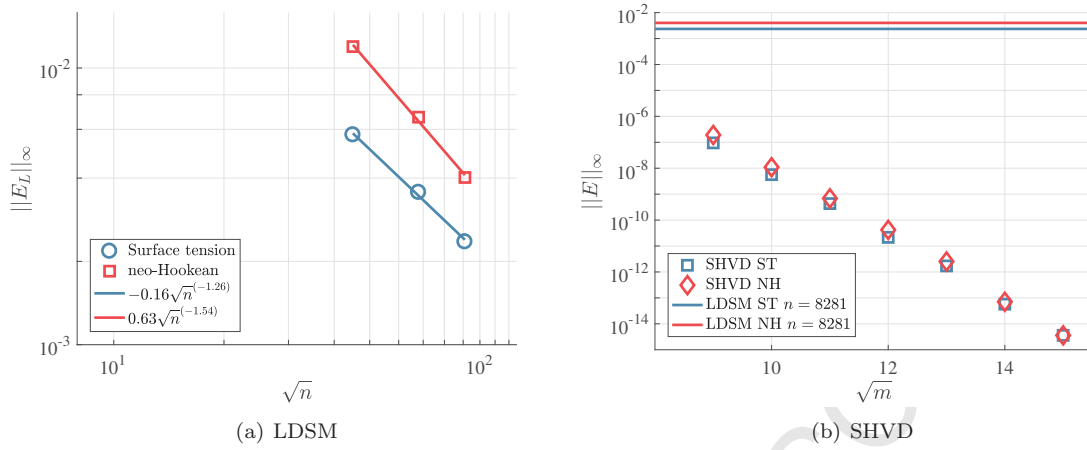


Figure 3: Errors in forces for the perturbed ellipsoid (Eq. (89)) for the two models. The discrete ℓ_∞ norms of \mathbf{E}_L and \mathbf{E}_S given by Eqs. (91) and (90) are graphed. In (a), the error $\|\mathbf{E}_L\|$ is plotted against the number of evaluation points n for both surface tension (blue circles) and neo-Hookean (red squares) forces. In (b), the error in SHVD $\|\mathbf{E}_S\|$ is plotted against the number of interpolation points m for both surface tension (blue squares) and neo-Hookean (red diamonds) forces. We observe spectral convergence in the number of interpolation points m . Forces computed with SHVD with as few as $m = 81$ interpolation points are several orders of magnitude more accurate than those computed from LDSM with $n = 8281$ evaluation points.

coefficients computed in Eq. (86) [41]. Here the analytical expressions for force are often not in the space of spherical harmonics of degree at most N , which makes them subject to aliasing error. Previous studies, among them [21, 22], have accounted for this via upsampling of points for force calculations. We take a different approach by increasing the number of interpolation points until the error becomes close to machine epsilon. Fig. 3(b) shows the error in the force calculation for the perturbed ellipse to be on the order 10^{-15} with 225 interpolation points used.

4.2. Dynamic test

We next perform full immersed boundary simulations on the test objects in Eqs. (88) and (89) with parameters $a = 1.2$, $b = c = 1/\sqrt{1.2}$, $B = 0.25$, $V_f \approx 5.14$, and $\sigma = K = A = G_s = 1$. Each object relaxes to its equilibrium configuration over time (in this case, the unit sphere). The relaxation timescale is dictated by the values of elastic moduli and the viscosity of the fluid, which we set to $\mu = 1$ for simplicity. We use 8281 evaluation points ($n = 8281$) for both objects and 225 interpolation points in the SHVD model ($m = 225$). From Fig. 2, this corresponds to an initial error in forces on the order of 10^{-3} for LDSM and 10^{-15} (machine epsilon) for SHVD. We use a total force that is the sum of surface tension and neo-Hookean forces (i.e. $\hat{\mathbf{F}}_{TOT} = \hat{\mathbf{F}}_{NH} + \hat{\mathbf{F}}_{ST}$). The Eulerian grid runs from $a = -2$ to $b = 2$ in the x, y , and z directions with a grid size of $\eta = 32$ (32 Eulerian points in each direction). For each object, we track the maximum or minimum displacement from the center of mass as a function of time. Fig. 4 shows that, despite the large difference in accuracy between LDSM and SHVD force computations, both models give similar results for displacement over time due to the first order time update and the discrete delta function in the spreading and interpolation operators.

We study the convergence of each method by examining the state of the ellipsoid in Eq. (88) and Fig. 1(a) at $t = 3$ seconds. The initial surface configuration is the ellipsoid with $a = 1.2$, and $b = c = 1/\sqrt{1.2}$. We consider a fixed Eulerian grid of size η^3 Eulerian points. We then choose a Lagrangian grid with $n \approx 2\eta^2$ points. We let $\Delta t = 1/(2\eta)$ and measure the change in distance, denoted d_{max} , of the rightmost point (the point with largest x coordinate) minus the center of mass (i.e. it is the amount the rightmost point has moved in the x direction after 3 seconds after subtracting the center of mass). The error is the difference in d_{max} with respect to a grid for the same method that is 1.5 times as refined. For example, the error value for a 32 point LDSM grid is the change in x -displacement of the rightmost point from the center of mass on the 32 point grid minus the corresponding value on a 48 point grid, both using LDSM.

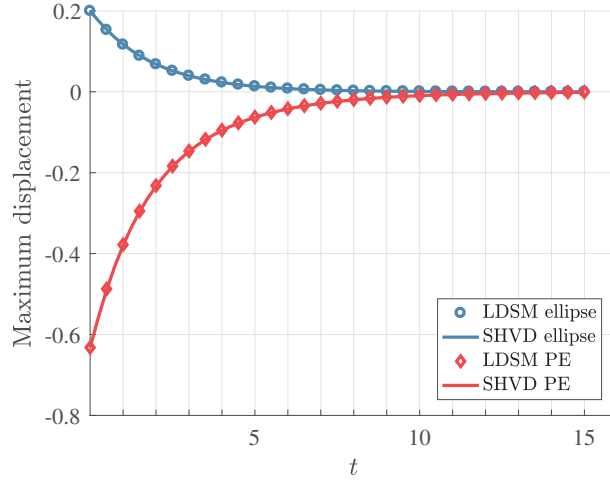


Figure 4: Maximum displacement over time for dynamic simulations of the test objects. The displacement is the maximum displacement from the center of mass minus the radius in the reference configuration. Despite the inaccuracy of LDSM in computing the forces, the full IB simulation gives similar results to one that uses SHVD for force computations.

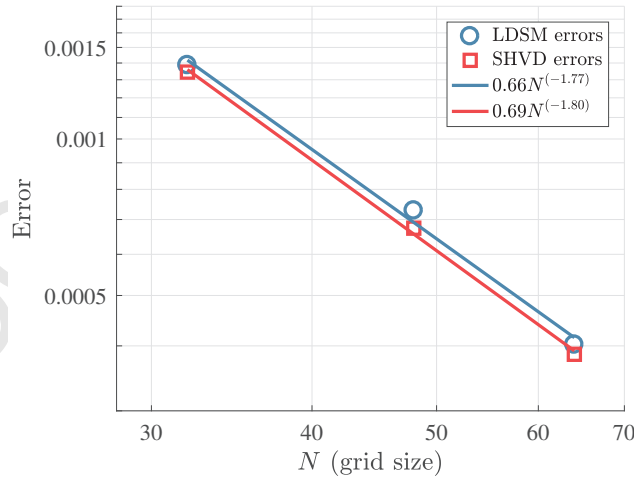


Figure 5: Convergence tests on a dynamic simulation of an ellipsoid for both LDSM and SHVD. The error is measured as the change in displacement of the point with largest x coordinate relative to the center of mass, d_{max} . The error is the difference in d_{max} with respect to a grid 1.5 times as refined. Lines show power law function fits to the data which demonstrate the approximately first order accuracy of both methods.

η	n	t -ratio for $m = 64$	t -ratio for $m = 225$
32	2025	1.15	1.73
48	4624	1.08	1.46
64	8281	0.99	1.30
72	10404	1.00	1.32

Table 1: Ratio of computational times for 15 second ellipsoidal contraction.

We consider grid sizes η , of 32, 48, and 64 corresponding to $n = 2025$, 4624, and 8281 evaluation points. For SHVD, we hold the number of interpolation points m constant at 64. Fig. 5 shows the results of the convergence test for the different grid sizes. Both methods display at least first order convergence in space and time. This is expected for the IB method, which as discussed in [42, 43], is limited to first order accuracy in the case of an infinitely thin elastic membrane. We comment on this further in Section 6.

In addition to studying the convergence of each method under these circumstances, we also compare the computational costs. In a similar procedure to the convergence tests, we track the ratio of SHVD to LDSM execution times (i.e. time for SHVD / time for LDSM) for a 15 second ellipsoidal contraction with the same parameters as the convergence study. Table 1 summarizes the results. For m two or more orders of magnitude less than n , the wall clock time of our method is within 20% of LDSM. Observe that for larger m values, the $\mathcal{O}(mn)$ operations in our method's force computation increase the computational time farther beyond that of LDSM. We will see in Sections 5.1 and 5.2 how this behavior impacts the use of the model in our applications. For $m \ll n$ (the scenario under which our method is intended for use), the algorithms have comparable execution times.

5. Applications

In this section, we consider two different applications involving cellular dynamics: red blood cells in capillary flow and a model of bleb expansion. The former application involves large and smooth deformations of the entire cell surface due to the background flow in a narrow tube. The membrane deformation in the model of cellular blebbing occurs because of a discontinuous applied force on a small region of the membrane surface. For both models, we compare simulation results from our method to LDSM. The results from the model simulations highlight the versatility of our approach.

5.1. Red blood cells in capillary flow

Red blood cells (RBCs) passing through microcapillaries is a well-studied biophysical problem that is important for understanding the rheological properties of both large and micro scale blood flow as well as transport via blood flow [44]. It has been observed experimentally that RBCs, which are nearly incompressible, deform into parachute-like shapes when flowing through capillaries, with the leading, or front, edge protruding under the influence of Poiseuille flow within the capillary [45]. For smaller tube diameters, it has been shown that RBCs align themselves in a single file configuration [45]. Our approach is therefore to model a single RBC flowing through a long capillary with SHVD. Because the velocity and lengthscales are small, the Reynolds number for flow in the microcirculation is on the order $< 10^{-2}$, and the Stokes equations, Eqs. (1)-(2), govern the flow profile. We also simplify the problem by assuming the viscosity of the internal hemoglobin solution to be equal to that of the surrounding plasma [44].

5.1.1. Model of the RBC and capillary

The reference configuration of a red blood cell that we use is taken from [46]. Let $(x_s, y_s, z_s) = (\cos \lambda \cos \theta, \sin \lambda \cos \theta, \sin \theta)$ denote the mapping from $(\lambda, \theta) \rightarrow \mathbb{R}^3$ for the unit sphere. The RBC reference configuration is given by

$$\mathbf{Z}_{RBC} = R_{RBC} \begin{pmatrix} \frac{1}{2} x_s \left(0.21 + 2 (y_s^2 + z_s^2) - 1.12 (y_s^2 + z_s^2)^2 \right) \\ y_s \\ z_s \end{pmatrix}. \quad (92)$$

The mean radius of the RBC is $R_{RBC} = 3.91 \mu\text{m}$ [44]. Elastic forces on the RBC come from resistance to stretching, shear, and bending. We use the neo-Hookean energy functional in Eq. (68) with $G_s = 2.5 \times 10^{-3} \text{ dyn/cm} = 2.5 \text{ pN}/\mu\text{m}$ and $A = 50$ to model resistance to shear and stretching. As discussed in [12, 44], A is a computational parameter that is used to enforce the incompressibility of the RBC membrane while ensuring stability at large enough timesteps. [12] reported using A in the range from 8 to 400, and we found our value of $A = 50$ to give area deformations less than 1%. The derivatives of the reference configuration required for the neo-Hookean force calculation in Eq. (35) are computed analytically from the \mathbf{Z}_{RBC} mapping. In addition, the area weights required to integrate the force over the RBC reference configuration are given by Eq. (51), where $\mathbf{Z}_L = \mathbf{Z}_{RBC}$. For bending, we use the energy functional in Eq. (79) with a typical value $k_{bend} = 1 \times 10^{-12} \text{ erg} = 0.1 \text{ pN} \cdot \mu\text{m}$.

The capillary is modeled as a thin elastic cylindrical membrane of radius $5 \mu\text{m}$ that is discretized and tethered in place. Letting \mathbf{Z}_{cap} and \mathbf{X}_{cap} be the reference and deformed configurations of the capillary, the Lagrangian force on a capillary node is given by

$$\hat{\mathbf{F}}_{teth}(\mathbf{q}) = -k_{teth} (\mathbf{X}_{cap}(\mathbf{q}) - \mathbf{Z}_{cap}(\mathbf{q})), \quad (93)$$

where \mathbf{q} refers to the Lagrangian coordinate of the node and $k_{teth} = 2.45 \times 10^{-3} \text{ dyn/cm} = 2.45 \text{ pN}/\mu\text{m}$. As discussed in [24], tether force functions such as Eq. (93) are not translation invariant, and are thus not guaranteed to integrate to zero in the discrete sense. Because the solution of the Stokes equations, Eqs. (1)-(2), on a periodic domain is non-unique, a constant velocity $\mathbf{u}_c(t)$ can be added to the structure velocities obtained from interpolation at each time step. Referring to Section 3.3, the time-stepping procedure is modified so that in the sixth step, the cylinder boundary is updated to an intermediate position

$$\hat{\mathbf{X}}_{cap}^{k+1} = \mathbf{X}_{cap}^k + \Delta t \mathbf{U}^{k+1}. \quad (94)$$

Then $\hat{\mathbf{X}}_{cap}^{k+1}$ is used to calculate a constant velocity that is added throughout the entire domain (also to the interpolated velocity of the RBC) to ensure that the tether forces sum to zero at the next time step. The constant velocity is given by

$$\mathbf{u}_c(t) = \frac{1}{A_{cap} \Delta t} \int_{cap} (\mathbf{Z}_{cap}(\mathbf{q}) - \hat{\mathbf{X}}_{cap}^{k+1}(\mathbf{q})) d\mathbf{q}, \quad (95)$$

where A_{cap} is the (lateral) surface area of the capillary. A derivation of Eq. (95) can be found in [24]. The positions of the Lagrangian structures (capillary and RBC) are then updated by $\mathbf{X}^{k+1} = \hat{\mathbf{X}}^{k+1} + \mathbf{u}_c(t) \Delta t$ to ensure the tether forces in Eq. (93) integrate to zero at the next time step.

A background force of $0.08 \text{ pN}/\mu\text{m}^3 = 8.0 \times 10^3 \text{ dyn/cm}^3$ is specified inside the capillary to establish a parabolic flow profile as in [12]. Given the viscosity of blood plasma as $1.2 \text{ cP} = 1.2 \times 10^{-3} \text{ pN} \cdot \text{s}/\mu\text{m}^2$, the resulting maximum flow velocity is $750 \mu\text{m/s}$ inside the capillary as in [12, 45]. We impose a uniform force in the opposite direction outside of the capillary to ensure the integral of the background force over the Eulerian grid is zero. We use a $24 \mu\text{m} \times 24 \mu\text{m} \times 24 \mu\text{m}$ fluid grid. The long axis of the capillary is the x -axis, and the capillary runs from $x = -8$ to $x = 8 \mu\text{m}$. We center the cell at $(-8, 0, 0)$ initially and simulate its deformation until it reaches a steady state shape (around $t = 0.05 \text{ s} = 500 \Delta t$).

5.1.2. Results

As discussed in Section 2.6.3, there is no standard way to model bending energy in LDSM because the piecewise linear triangles have zero curvature. In addition, the bending force model from [12] is not guaranteed to integrate to zero in the discrete sense, meaning it cannot be used in a zero Reynolds number regime with periodic boundary conditions. Thus in order to fairly compare our model to LDSM, we begin by examining the case of $k_{bend} = 0$. For all simulations, we use $n = 8281$ evaluation points and vary the number of interpolation points m in our method.

Fig. 6 shows the deformation of the RBC from its biconcave initial shape into the parachute-like shapes observed in experimental capillary flow [45] for LDSM (top), and SHVD with $m = 625$ (middle) and $m = 400$ (bottom). We refer to the shape at $t = 0.05$ as the steady state shape of the RBC because it does not change significantly after this time value. While the steady state shapes are similar, a closer look

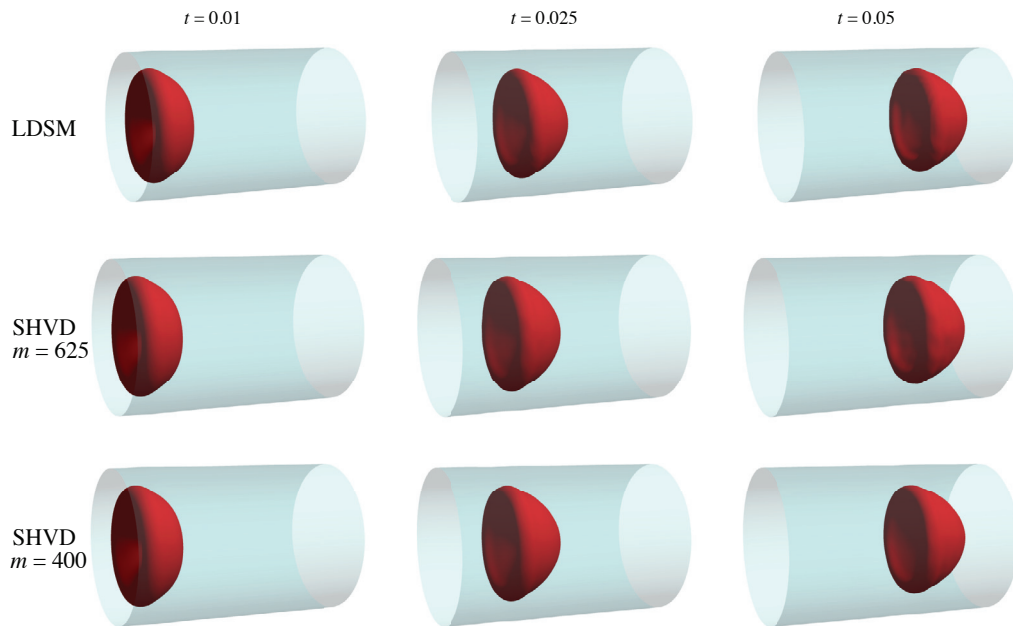


Figure 6: The cell membrane and capillary at several time values using LDSM (top) and SHVD with $m = 625$ (middle) and $m = 400$ (bottom) interpolation points. All methods used $n = 8281$ evaluation points.

reveals bumps in the RBCs modeled with LDSM and SHVD with $m = 625$. We take these asymmetric bumps to be non-physical, numerical error since the imposed flow and cylinder are symmetric. These bumps are distinct from the larger wavelength, axisymmetric, wrinkles observed in all of our simulations that exclude bending rigidity. We note that wrinkles have been observed in previous studies of shear flow at low bending rigidities [12, 47]. When the number of interpolation points is greater than 625, we observe even more bumps on the cell (data not shown), indicating instability due to higher frequency modes in the interpolant. For the remainder of this section, we take $m = 400$ to obtain smooth surfaces similar to previous computational studies of RBCs [12, 22].

Fig. 7 shows a comparison of the magnitude of the RBC elastic forces from simulations using LDSM and SHVD at $t = 0.05$. Although the overall accuracy in space and time of both methods is first order (see Section 4.2), the forces are much smoother with SHVD. Additionally, the magnitude of the forces is much larger in computations with LDSM than SHVD at the back end of the cell when its shape changes from concave to convex (and the mean curvature changes sign). The side views of the steady state shape (middle, Fig. 7) show wrinkles that occur in shapes from both simulations when the bending forces are excluded.

Surface bumps and wrinkles of the RBC are visible in Fig. 7 when $k_{bend} = 0$. Fig. 8 shows a head-on (facing the cell) view of the steady state cell shapes (at $t = 0.05$) for different values of bending rigidity. When $k_{bend} = 0$, the mean curvature H (defined in Eq. (76)) oscillates between positive and negative because of the wrinkling (Fig. 8, $k_{bend} = 0$). When $k_{bend} = 0.1 \text{ pN} \cdot \mu\text{m}$, surface wrinkles are significantly reduced, the curvature remains negative on the top face, and a smooth parachute shape is obtained. We note this is the same value of k_{bend} used in [12]. The curvature remains negative on the top face when $k_{bend} = 1$.

At steady state, the front of cells with higher bending rigidities more closely resembles a sphere. We quantify this by examining the variability in mean curvature on the front edge of the cell as a function of the bending rigidity in Fig. 8. It can be seen that as k_{bend} increases from 0 to $1 \text{ pN} \cdot \mu\text{m}$, the front of the

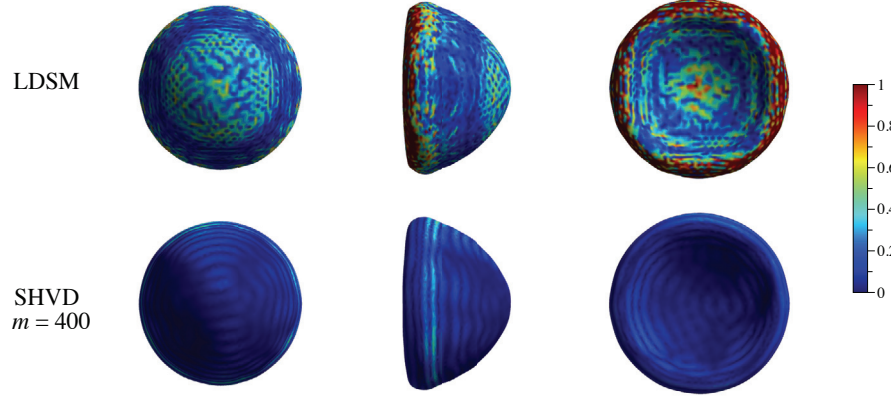


Figure 7: Magnitude of the elastic forces on the cell at the front, side, and back views of the cell at $t = 0.05$ from simulations with LDSM (top) and SHVD with $m = 400$ interpolation points (bottom). $n = 8281$ evaluations points were used for both simulations.

cell more closely resembles a sphere with uniform curvature, as is expected for large bending rigidities. We note the smoothness of the curvature values in our method, which clearly shows the trend towards a more spherical shape for larger bending rigidities.

The bending rigidity is defined as the cell's resistance to being bent from a spherical shape with uniform curvature. Fig. 9 shows a slice of the RBC and capillary through the plane $y = 0$ at several time points during a simulation. The cell's deformation decreases as the value of bending rigidity increases. In particular, we observe that a cell with no bending rigidity initially deforms to become pinched at its center, then flat, and then curved. Meanwhile, the front edge of the cell maintains more uniform mean curvature throughout the simulation when $k_{bend} = 1 \text{ pN} \cdot \mu\text{m}$ than with smaller values of k_{bend} .

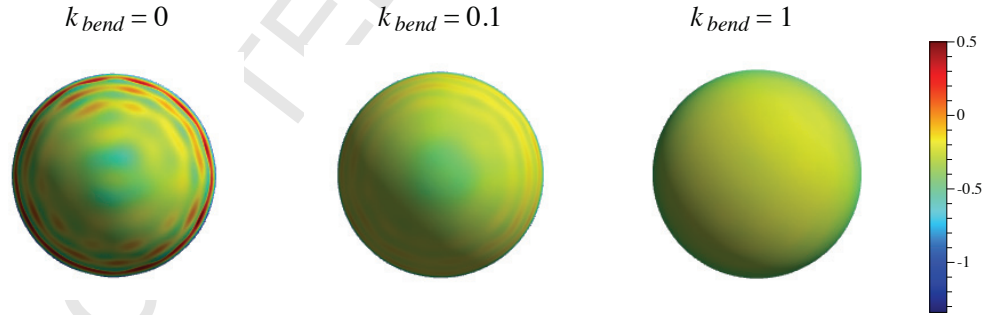


Figure 8: Mean curvature of the cell viewed from the front for increasing values of k_{bend} . The cell was simulated with SHVD ($m = 400$ interpolation and $n = 8281$ evaluation points).

5.2. Model of cellular blebbing

Blebs are round membrane protrusions that some cells use during migration [48]. Blebbing occurs when a thin layer of the cytoskeleton called the cortex delaminates from the cell membrane, where it is normally attached via linker (ERM) proteins [49]. Cells that bleb are usually under high cortical tension due to myosin molecular motors pulling actin filaments with respect to each other [50], a process referred to as actomyosin contractility. High actomyosin contractility within the cortex leads to high intracellular pressure (on the order of 10 Pa). When high intracellular pressure occurs along with a localized defect in the cortex or loss of membrane-cortex adhesion proteins, the membrane separates from the cortex, and

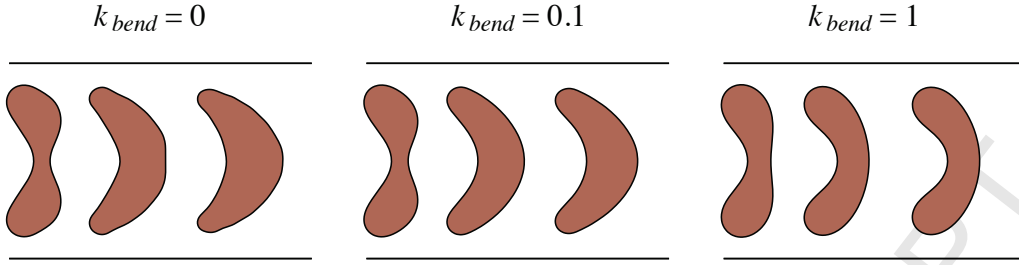


Figure 9: Cross section of the cell and capillary tube through the plane $y = 0$ at several time values ($t = 0.005, 0.025$, and 0.05 from left to right) for when $k_{bend} = 0$ (left), 0.1 (middle), and 1 (right) pN · μm. The cell was simulated with SHVD ($m = 400$ interpolation and $n = 8281$ evaluation points).

cytoplasmic flow leads to bleb expansion. Hence, blebbing is a discontinuous deformation of a localized part of the cell in contrast to the global deformation of an RBC in a microcapillary.

Several mathematical models of cellular blebbing have been developed to understand the interactions between the cytoskeleton, cell membrane, and adhesion proteins. These models typically make simplifying assumptions, such as the system being in steady state [50, 51], or are limited to two spatial dimensions [52, 53]. In this section, we present a three-dimensional extension of the model from [52] as a first step towards understanding both the dynamics and geometry of these cellular protrusions. Our goal here is to present a dynamic 3D model of cellular blebbing where membrane forces are computed using both LDSM and SHVD. We compare bleb expansion dynamics and cell shapes generated by these methods. We also describe a modification to the existing method from [12, 54] for computing mean curvature when using LDSM.

5.2.1. Mathematical model

Our mathematical model consists of the membrane, cortex, membrane-cortex adhesion, and intracellular fluid (see Fig. 10). The intracellular fluid represents the cytosol, the liquid part of the cytoplasm. The membrane is modeled as an impermeable elastic shell that experiences forces due to surface tension and stretching. The cortex is modeled as a permeable elastic structure, and membrane-cortex adhesion is modeled by elastic springs connecting the membrane to the cortex. The cytosol is modeled as a viscous incompressible fluid governed by the fluid equations

$$\begin{aligned} \mu \Delta \mathbf{u} - \nabla p + \mathbf{f}_{\text{elastic}}^{\text{mem}} + \mathbf{f}_{\text{adh}}^{\text{mem/cortex}} + \mathbf{f}_{\text{drag}}^{\text{cortex}} &= \mathbf{0} \\ \nabla \cdot \mathbf{u} &= 0, \end{aligned} \quad (96)$$

where the \mathbf{f}_i 's denote spread Lagrangian force densities due to membrane elasticity, membrane-cortex adhesion, and cortical drag. The drag force on the cortex is balanced by elastic forces within the cortex and adhesion to the membrane so that

$$\mathbf{F}_{\text{drag}}^{\text{cortex}} + \mathbf{F}_{\text{elastic}}^{\text{cortex}} + \mathbf{F}_{\text{adh}}^{\text{cortex/mem}} = \mathbf{0}. \quad (97)$$

The drag force density on the fluid is related to the cortex drag force density by

$$\mathbf{f}_{\text{drag}}^{\text{cortex}} = -S \mathbf{F}_{\text{drag}}^{\text{cortex}}, \quad (98)$$

where S denotes the spreading operator from Eq. (3).

Forces due to membrane elasticity are computed by either LDSM or SHVD with surface tension and neo-Hookean energy densities. The cortex is modeled by a different approach, similar to the lattice-spring model described in [55]. Specifically, the cortex is treated as a collection of elastic springs, where the force

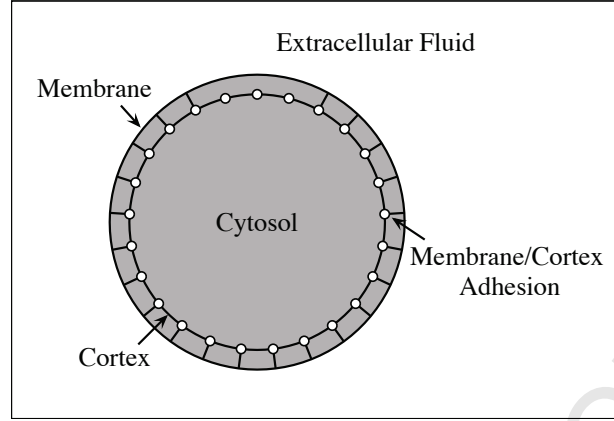


Figure 10: Bleb model schematic as a slice of the cell through the xz -plane. A bleb is initiated by removing adhesive links in a small region at the top of the cell.

at the i th point is computed by

$$\mathbf{F}_i dA_i = \sum_j k_{ij}^{\text{cortex}} \left(\frac{\|\mathbf{X}_i^{\text{cortex}} - \mathbf{X}_j^{\text{cortex}}\|}{dl_{ij}} \right) \frac{\mathbf{X}_i^{\text{cortex}} - \mathbf{X}_j^{\text{cortex}}}{\|\mathbf{X}_i^{\text{cortex}} - \mathbf{X}_j^{\text{cortex}}\|}, \quad (99)$$

where dA_i is the reference area differential, dl_{ij} is the reference length of the spring connecting the i th point to the j th point on the surface, and j is the index of all points connected to the i th point. Note the area weights are the same as those from LDSM in Eq. (67). The spring coefficient connecting the i th point to the j th point is calculated as

$$k_{ij}^{\text{cortex}} = \frac{8k_{ij}}{3dl_{ij}} \left(\frac{dA_i + dA_j}{2} \right), \quad (100)$$

as in the lattice-spring model from [55]. This particular model of the cortex was used so that cortical ablation experiments could easily be performed by setting $k_{ij}^{\text{cortex}} = 0$ in a localized region. We plan to explore this mechanism of bleb initiation in a future publication.

Membrane-cortex adhesion is modeled by elastic springs attaching the membrane to the cortex with a force density given by

$$\mathbf{F}_{\text{adh}}^{\text{mem/cortex}} = k_{\text{adh}}^{\text{mem/cortex}} \left(\|\mathbf{X}_{\text{mem}} - \mathbf{X}_{\text{cortex}}\| \right) \frac{\mathbf{X}_{\text{mem}} - \mathbf{X}_{\text{cortex}}}{\|\mathbf{X}_{\text{mem}} - \mathbf{X}_{\text{cortex}}\|}. \quad (101)$$

The membrane-cortex adhesion stiffness coefficient $k_{\text{adh}}^{\text{cortex/mem}}$ was chosen so that the velocity of the cortex is zero if no bleb is initiated. The adhesion force density on the membrane is the opposite of the corresponding force density on the cortex, with the proper scaling to ensure that the two forces balance,

$$\int_{\Omega} \mathbf{S}\mathbf{F}_{\text{adhesion}}^{\text{mem/cortex}} d\mathbf{x} + \int_{\Omega} \mathbf{S}\mathbf{F}_{\text{adhesion}}^{\text{cortex/mem}} d\mathbf{x} = 0. \quad (102)$$

Given the stiffness coefficient $k_{\text{adh}}^{\text{cortex/mem}}$, the corresponding stiffness coefficient for the cortex is obtained by $k_{\text{adh}}^{\text{mem/cortex}} = k_{\text{adh}}^{\text{cortex/mem}} dA_i^{\text{cortex}} / dA_i^{\text{mem}}$, where dA_i represents the area differential in reference coordinates. For the membrane, these dA_i values are given by Eq. (45) for SHVD and Eq. (67) for LDSM.

Given a configuration of the membrane and cortex, the Lagrangian force densities are computed, and the velocities of the fluid and cortex are obtained by solving Eqs. (96) and (97). The positions of the

Symbol	Quantity	Value	Source
r_{mem}	Cell radius	$10 \mu\text{m}$	[50]
r_{cortex}	Cortex radius	$9.99 \mu\text{m}$	[53]
γ_{mem}	Membrane surface tension	$40 \text{ pN}/\mu\text{m}$	[50]
κ_E	Membrane bulk modulus	$40 \text{ pN}/\mu\text{m}$	[53]
μ_E	Membrane shear modulus	$40 \text{ pN}/\mu\text{m}$	[53]
k_{cortex}	Cortical stiffness coefficient (average value)	$87 \text{ pN}/\mu\text{m}^2$	
$k_{\text{adh}}^{\text{mem/cortex}}$	Membrane/cortex adhesion stiffness coefficient	$3 \cdot 10^4 \text{ pN}/\mu\text{m}^3$	
μ	Cytosolic viscosity	$1 \text{ Pa}\cdot\text{s}$	[50, 56, 57]
ξ	Cortical drag coefficient	$10 \text{ pN}\cdot\text{s}/\mu\text{m}^3$	[52]

Table 2: Parameters for the blebbing model.

membrane and cortex are then updated with their respective velocities,

$$\frac{d\mathbf{X}_{\text{mem}}}{dt} = \mathbf{S}^* \mathbf{u} = \mathbf{U}, \quad (103)$$

$$\frac{d\mathbf{X}_{\text{cortex}}}{dt} = \frac{1}{\xi} \left(\mathbf{F}_{\text{elastic}}^{\text{cortex}} + \mathbf{F}_{\text{attach}}^{\text{cortex/mem}} \right) + \mathbf{U} = \mathbf{U}_{\text{cortex}}, \quad (104)$$

where ξ is the drag coefficient of the cortex. A bleb is initiated by removing adhesive links between the membrane and cortex in a small region given by the equation $\phi \leq 2\pi/25$ in spherical coordinates (see Fig. 10). Model parameters values are provided in Table 2.

The model is simulated on a box with volume $30^3 \mu\text{m}^3$ and $\Delta x = \Delta y = \Delta z = 30/32$. The membrane and cortex are initialized to be spheres centered at the point $(15, 15, 15)$ with 4761 Lagrangian points, corresponding to approximately 2 Lagrangian points per Eulerian grid cube. For the SHVD model, we compute all solutions with 4761 evaluation points and vary the number of interpolation points to be 625, 1024, and 4761. The lattice spring model on the cortex results in large spurious velocities at the beginning of the simulation. Before initiating a bleb, we allow the forces in the model to equilibrate, meaning the velocity on the boundary decreases to approximately $0.01 \mu\text{m}/\text{s}$. This process corresponds to 0.2 s of simulation time. Results are reported in time units after equilibration.

5.2.2. Model results

Pressure is initially uniform inside the cell. When adhesive links between the membrane and cortex are removed, cortical tension is no longer transmitted to the membrane locally, resulting in a pressure gradient and bleb expansion. Fig. 11 shows cross sections of the cell membrane as a bleb expands using both LSDM and SHVD with 1024 interpolation points. The cross sections show that the models have good agreement with each other.

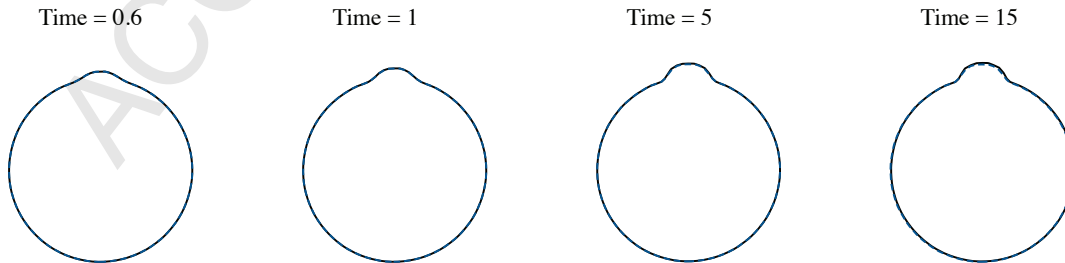


Figure 11: Slices through the cell membrane by the yz plane when $x = 15$ at several time values. The black line denotes the membrane position computed by LSDM, and the blue dashed line indicates the membrane position computed by SHVD with 1024 interpolation points and 4761 evaluation points.

In order to obtain a localized measurement of bleb size, we compute the bleb ring as the one dimensional boundary between the region where the membrane is attached to the cortex and the region where the membrane is detached from it (see Fig. 12). Bleb diameter is measured as the distance from the point closest to the z -axis to the point farthest away from it on the bleb ring. The midpoint of this line is computed, and bleb size is defined to be the vertical distance from the midpoint of the bleb diameter line to the point on the bleb with the largest z -coordinate (blue point in Fig. 12(a)) after subtracting the initial value. Note that this is a 3D extension of the method described in [53].

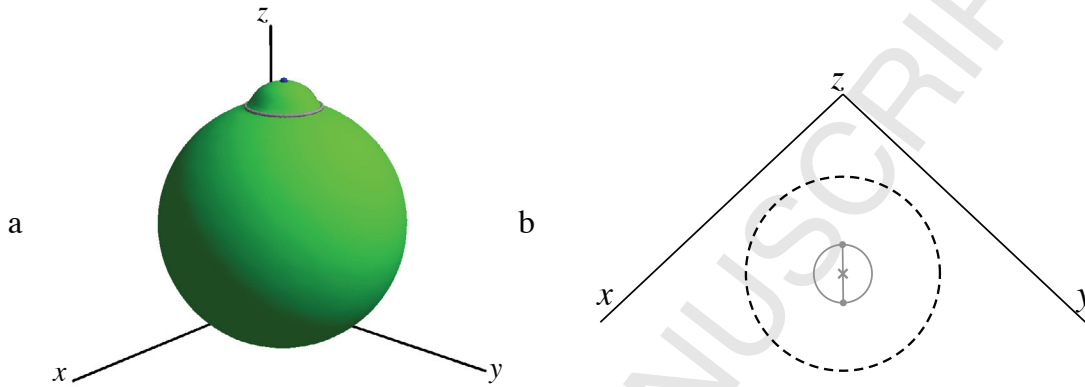


Figure 12: The bleb ring is defined by the boundary between the region where adhesive links connect the membrane to the cortex and the region where the links are removed, indicated by the gray ring in (a). The membrane shell is graphed in green in (a). Bleb diameter is found by finding the point on the bleb ring that is closest to the point $(0,0,30)$ and the point directly across from it. A top down view of the cell and bleb ring is shown in (b) with the two points that define the bleb's diameter. The dashed black line indicates the membrane and the solid gray circle indicates the bleb ring. The distance from the midpoint of the bleb diameter (indicated by \times in the top-down view) to the point with the largest z -coordinate value, as indicated by the blue point in (a), minus its initial value defines bleb size.

Bleb size over time when membrane forces are computed by LDSM and with several different values of interpolation points in SHVD is shown in Fig. 13. The data show very similar bleb expansion dynamics and steady state bleb size for all simulations. We observe that the bleb size increases as the bleb neck is better resolved with a larger number of interpolation points. Once we reach $m = n = 4761$ (red triangles in Fig. 13), each evaluation is updated explicitly with the local fluid velocity. In this case (when $m = n$), the difference between our method and LDSM is due only to the method of computing forces.

Bleb volume over time is shown in Fig. 14, where percentage relative volume difference is defined as $(V(t) - V(0))/V(0) \times 100$, where $V(t)$ is the volume of the membrane at time t . Results from LDSM and SHVD with 1024 and 4761 interpolation points show good agreement. Volume actually increases over time with the SHVD model using 625 interpolation points. Because of the sharp transition from a region with membrane-cortex adhesion to one without adhesion, a large number of interpolation points is needed to resolve the region near the bleb neck.

Notably, the number of overall interpolation points required to resolve the bleb neck, which is a highly localized region of deformation, is higher than the number of points required to resolve the deformed RBC in Section 5.1 because the points overall must be evenly spread over the cell to yield an accurate interpolant. Hence more points must be added throughout the cell, not just in the localized region, in order to increase resolution of the bleb neck. Unlike our RBC application, adding points up to the limit $m = n$ actually *stabilizes* the steady state membrane shapes. We believe this is because most of the membrane remains still while tethered to the cortex. Both of these factors (the lack of motion from a background flow and the tether forces on the membrane) prevent higher order modes from distorting the membrane shape.

Membrane mean curvature after bleb expansion (at $t = 15$ s) is shown in Fig. 15 for the same data from Figs. 13 and 14. In order to compute curvature for LDSM, we follow the method described in [12, 54], where curvature is computed on edges and vertices of surface triangles. This method results in curvatures

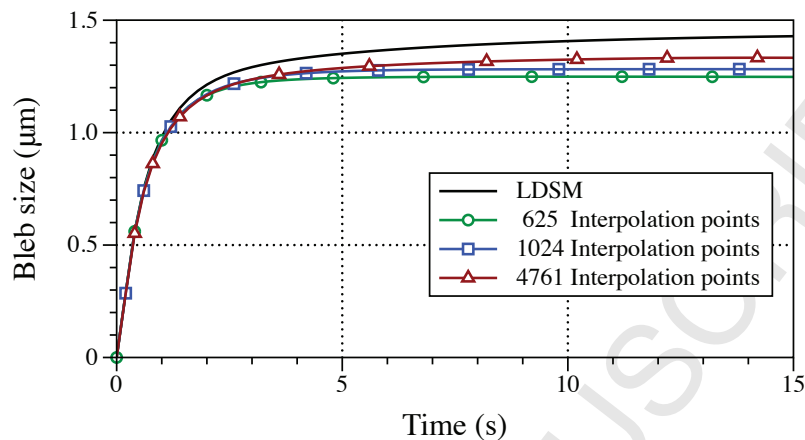


Figure 13: Bleb size over time when the membrane position is computed by LDSM (black line) and by SHVD with various numbers of interpolation points and 4761 evaluation points.

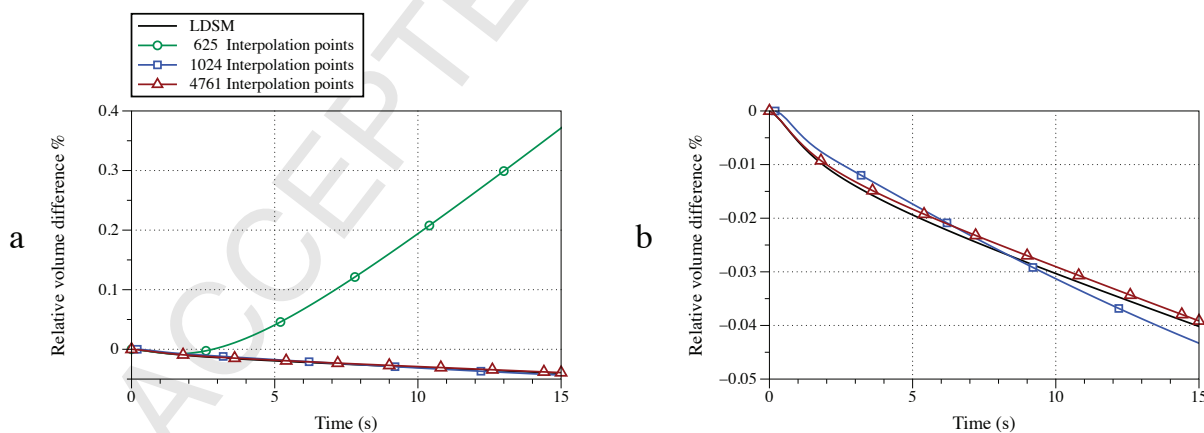


Figure 14: Membrane volume over time when SHVD is used for the membrane with 625, 1024, and 4761 interpolation points. Data from the membrane modeled with LDSM are shown in black for comparison. The graph on the right is a zoomed in view of the same data from the left.

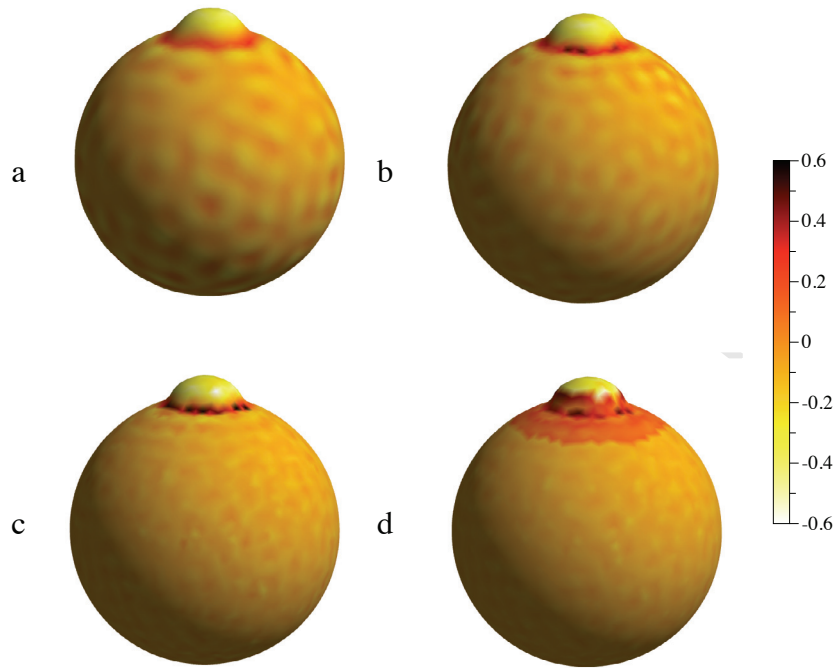


Figure 15: Membrane mean curvature after bleb expansion at $t = 10$ s when the membrane is modeled using SHVD with (a) 625, (b) 1024, and (c) 4761 interpolation points. Mean curvature from LSDM with 4761 points is shown in (d).

that are always positive. In order to obtain signed mean curvature, we post-process the surface to compute its convex hull. Curvature values at points on the convex part of the surface are assigned to be negative while the rest of the curvature values (on the concave part of the surface) maintain their positive sign. In SHVD with 625 interpolation points, the bleb appears more broad which indicates it is not well-resolved. The membrane is effectively smoothed out if not enough interpolation points are used, and volume slightly increases. The membrane appears more dimpled in SHVD with 625 interpolation points compared to data from the other simulations. This can be partially attributed to the adhesion model, where membrane evaluation points are connected to points on the cortex. Since membrane evaluation points are not explicitly updated with the fluid velocity, adhesion forces are slightly perturbed when compared to model results from Fig. 15 (c) and (d), where the interpolation and evaluation points are updated with the local fluid velocity (although some dimpling is present even in Fig. 15 (c) and (d) due to membrane-cortex adhesion). The bleb appears to be well-resolved in SHVD with 1024 interpolation points with a smooth membrane. Membrane curvature from LSDM appears discontinuous and noisy when compared to simulation data from SHVD. We emphasize that SHVD more accurately computes curvature than LSDM given the ability of the spherical harmonic interpolant to capture the membrane shape.

The results from this section show that cell shape and bleb expansion dynamics are qualitatively similar when the membrane is modeled by LSDM and SHVD. If accurate geometric information, such as membrane curvature, is desired from a simulation, SHVD should be used for computing membrane forces. In [55], the authors used a 2D dynamic model to determine the role of cortical drag and cytoplasmic viscosity on bleb expansion dynamics. We find that for obtaining dynamic information such as bleb expansion time, LSDM is sufficiently accurate. In the next section, we comment on the computational cost associated with SHVD when compared to LSDM.

6. Discussion

We show that by representing surfaces with spherical harmonic interpolants, force densities can be computed more accurately as continuous functions by taking the variational derivative of an energy density functional than by discretizing the surface first into piecewise planar elements. Furthermore, geometric information such as curvature is more accurately computed with the continuous representation. We presented tests and applications for zero Reynolds number flow, but it is straightforward to extend our method to fluid-structure interaction problems where the fluid is governed by the Navier-Stokes equations. This is an advantage of our method compared to boundary integral methods that employ variational methods for computing forces, in that ours can also be used when modeling red blood cells in intermediate Reynolds number flows, for example [12].

A limitation of our method, as with all IB methods, is its overall first order accuracy in space because the regularized delta function used in the spreading operator does not accurately capture discontinuities in the pressure and derivatives of the velocity across the infinitely thin interfaces. Immersed interface methods [58] have been proposed to improve the accuracy of the IB method. These “IM methods” depend on quantifying the jump conditions at the interface, which are functions of geometric information, Lagrangian forces, and their derivatives [59]. The order of the method depends in part on how accurately these two quantities are computed. Because our method is spectrally accurate in both, a future goal of ours is to implement it in an IM framework, thereby increasing the overall accuracy of the method.

We also provide a method that ensures the integral of the Lagrangian force density is equal to the integral of the spread force density. This allows us to apply our method to models of structures immersed in Stokes flow, a regime that is prevalent across most cellular-level applications. We do note that our formulation relies on the structure being a closed surface.

Our method is applied to models of red blood cells in capillary flow and cellular blebbing. We have intentionally chosen two applications with very different dynamics and behavior in order to showcase the versatility of our method and highlight its strengths and limitations. In the RBC model, we see a smooth change in the entire surface over time. This allows us to resolve the shape with $\mathcal{O}(100)$ interpolation points. Lower numbers of interpolation points do not resolve the shape well enough, and higher numbers lead to numerical instabilities as higher frequency modes amplify non-physical bumps and noise on the cell surface.

Meanwhile, cellular blebbing involves a discontinuous change in a small region of the cell due to a discontinuity in the membrane-cortex adhesion force density. In our present formulation, the points must be evenly distributed to build a reliable interpolant and a large number of points must be used on the entire cell to accurately resolve the bleb neck. Therefore it is necessary to choose m (number of interpolation points) and n (number of evaluation points) to be on the same order of magnitude. We found that measuring volume of the membrane over time is an indicator of surface resolution; if membrane volume increases during a simulation, the surface is likely under-resolved and more interpolation points are needed. We hypothesize that the physical linkage of the membrane to the cortex stabilizes the higher frequency modes in the spherical harmonic interpolant, and numerical instabilities from a large number of interpolation points are not observed in this model.

The restriction $m \ll n$ is for computational efficiency; our method is most efficient if the $\mathcal{O}(mn)$ cost of computing the forces is comparable to the $\mathcal{O}(n)$ cost in LDSM. This is the case for the RBC model, where the surface is well-resolved with $\mathcal{O}(100)$ interpolation points and the computational time of our method is of the same order of magnitude as LDSM (approximately 1 minute). However, to simulate the blebbing model, m must be much larger, and the computational time to compute the forces increases significantly. A simulation of bleb expansion took approximately 10 hours with SHVD compared to approximately 1 hour for LDSM. While our model is versatile enough to be applied across a wide variety of applications, this additional cost should be taken into account.

We plan to use these principles to guide our future work. For example, our method would be well-suited for simulating a model of cytokinesis, the division of the cytoplasm when one cell divides into two cells [60]. During cytokinesis, the entire membrane deforms, and our method allows the force density to be computed at any point on the surface. Models of microtubule attachment to the cortex during spindle positioning will benefit from knowing the elastic force exactly at the points where the microtubules attach to the cortex. The noisy force data from LDSM could lead to inaccurate results. Finally, some models assume that spontaneous membrane curvature initializes furrow ingression during cytokinesis [61]. Our

framework provides an accurate method for computing curvature in a dynamic 3D model of the early stages of cytokinesis.

Other future work includes extending our framework to include adaptive meshing by adding interpolation points when the surface shape undergoes large local deformations and surface features become unresolved. This capability, along with finding a more efficient way to compute the force densities (e.g. in parallel), could make our method more suitable for use with the blebbing model and allow us to investigate different bleb initiation mechanisms using our method. The formulation we have outlined can be applied to other interpolating basis functions, including radial basis functions. Because spherical harmonic basis functions were able to provide exact force calculations for very smooth (spherical) shapes, our method was computationally advantageous when it was applied to shapes resembling spheres. We anticipate that accuracy for arbitrary shapes (including blebbing cells) could be achieved by shifting the basis functions according to the application. We plan to investigate these topics in future publications.

Acknowledgements

This work was supported in part by grant #429808 from the Simons Foundation to W.S. A.K. was supported by a National Science Foundation grant DMS-1521748. The authors would like to thank Grady B. Wright for discussions about quadrature weights on the sphere, Varun Shankar for helpful discussions about surface representation, and Boyce Griffith for suggestions about computing forces due to bending. The authors also thank Thomas G. Fai and Charles S. Peskin for notes about modeling neo-Hookean materials in curvilinear coordinates.

Appendix A. Derivation of Force Density, F

Here we derive in detail the force density in Eq. (35) from the variational derivative of the energy density functional. We make use of dyad expansions of tensors ($\mathbf{e}_i \mathbf{e}_j$ denotes the i th row and j th column of the given tensor) and the definitions of the dot and double dot products ($\mathbf{e}_i \cdot \mathbf{e}_j = 1$ if $i = j$ and 0 otherwise, $\mathbf{e}_i \mathbf{e}_j : \mathbf{e}_k \mathbf{e}_p = 1$ if $j = k$ and $i = p$ and 0 otherwise).

Appendix A.1. Dyad expansion of \mathcal{P} allows us to recover Eq. (21) from Eq. (23)

We begin with showing the equivalence of Eq. (21) and Eq. (23). In Eq. (21), the variational derivative is given by

$$\frac{\delta \mathcal{E}}{\delta \mathbf{X}} = - \sum_{i=1}^3 \sum_{j=1}^2 \frac{1}{\sqrt{\det \mathcal{G}_0}} \frac{\partial (\mathcal{P}_{ij} \sqrt{\det \mathcal{G}_0})}{\partial q_j}. \quad (\text{A.1})$$

Considering the product in Eq. (23),

$$-\nabla_\theta \cdot \mathcal{P}^T = - \frac{1}{\sqrt{\det \mathcal{G}_0}} \left(\frac{\partial}{\partial \lambda} \sqrt{\det \mathcal{G}_0} \mathbf{e}_1 + \frac{\partial}{\partial \theta} \sqrt{\det \mathcal{G}_0} \mathbf{e}_2 \right) \cdot (\mathcal{P}_{11} \mathbf{e}_1 \mathbf{e}_1 + \mathcal{P}_{21} \mathbf{e}_1 \mathbf{e}_2 + \mathcal{P}_{31} \mathbf{e}_1 \mathbf{e}_3 + \mathcal{P}_{12} \mathbf{e}_2 \mathbf{e}_1 + \mathcal{P}_{22} \mathbf{e}_2 \mathbf{e}_2 + \mathcal{P}_{32} \mathbf{e}_2 \mathbf{e}_3) \quad (\text{A.2})$$

$$= - \sum_{i=1}^3 \frac{1}{\sqrt{\det \mathcal{G}_0}} \left(\frac{\partial}{\partial \lambda} (\mathcal{P}_{i1} \sqrt{\det \mathcal{G}_0}) + \frac{\partial}{\partial \theta} (\mathcal{P}_{i2} \sqrt{\det \mathcal{G}_0}) \right) \quad (\text{A.3})$$

$$= - \sum_{i=1}^3 \sum_{j=1}^2 \frac{1}{\sqrt{\det \mathcal{G}_0}} \left(\frac{\partial}{\partial q_j} (\mathcal{P}_{ij} \sqrt{\det \mathcal{G}_0}) \right). \quad (\text{A.4})$$

Eq. (A.4) is the same expression as that derived from the variational derivative definition in Eq. (21),

$$\mathbf{F} = -\nabla_\theta \cdot \mathcal{P}^T. \quad (\text{A.5})$$

Appendix A.2. Chain rule expansion of \mathcal{P}^T

We here prove Eq. (26),

$$\mathcal{P}^T = \frac{\partial W}{\partial(\nabla_\theta \mathbf{X})^T} = \frac{\partial W}{\partial \gamma} : \frac{\partial \gamma}{\partial(\nabla_\theta \mathbf{X})^T} = \mathcal{S} : \frac{\partial \gamma}{\partial(\nabla_\theta \mathbf{X})^T} = \mathcal{S} \cdot (\nabla_\theta \mathbf{X})^T. \quad (\text{A.6})$$

We begin with the double dot product $\frac{\partial W}{\partial \gamma} : \frac{\partial \gamma}{\partial(\nabla_\theta \mathbf{X})^T}$. Recall from Eq. (14) that γ is the Green-Lagrange strain energy tensor, given by $\frac{1}{2}(\mathcal{G} - \mathcal{G}_0)$. Because γ is symmetric, the tensor $\frac{\partial W}{\partial \gamma} = \mathcal{S}$ (the second Piola-Kirchhoff stress tensor) is also symmetric, and its dyad expansion is

$$\frac{\partial W}{\partial \gamma} = \mathcal{S} = \mathcal{S}_{11} \mathbf{e}_1 \mathbf{e}_1 + \mathcal{S}_{12} \mathbf{e}_1 \mathbf{e}_2 + \mathcal{S}_{21} \mathbf{e}_2 \mathbf{e}_1 + \mathcal{S}_{22} \mathbf{e}_2 \mathbf{e}_2. \quad (\text{A.7})$$

Now, consider the fourth order tensor $\frac{\partial \gamma}{\partial(\nabla_\theta \mathbf{X})^T}$. We denote

$$\left(\frac{\partial \gamma}{\partial(\nabla_\theta \mathbf{X})^T} \right)_{ijkl} = \frac{\partial \gamma_{ij}}{\partial(\nabla_\theta \mathbf{X})_{kl}^T}.$$

We write γ as

$$\gamma = \frac{1}{2} \begin{pmatrix} \frac{\partial \mathbf{X}}{\partial \lambda} \cdot \frac{\partial \mathbf{X}}{\partial \lambda} - \frac{\partial \mathbf{Z}}{\partial \lambda} \cdot \frac{\partial \mathbf{Z}}{\partial \lambda} & \frac{\partial \mathbf{X}}{\partial \lambda} \cdot \frac{\partial \mathbf{X}}{\partial \theta} - \frac{\partial \mathbf{Z}}{\partial \lambda} \cdot \frac{\partial \mathbf{Z}}{\partial \theta} \\ \frac{\partial \mathbf{X}}{\partial \theta} \cdot \frac{\partial \mathbf{X}}{\partial \lambda} - \frac{\partial \mathbf{Z}}{\partial \theta} \cdot \frac{\partial \mathbf{Z}}{\partial \lambda} & \frac{\partial \mathbf{X}}{\partial \theta} \cdot \frac{\partial \mathbf{X}}{\partial \theta} - \frac{\partial \mathbf{Z}}{\partial \theta} \cdot \frac{\partial \mathbf{Z}}{\partial \theta} \end{pmatrix}. \quad (\text{A.8})$$

Expanding entrywise,

$$\gamma = \frac{1}{2} \begin{pmatrix} \frac{\partial X_1^2}{\partial \lambda} + \frac{\partial X_2^2}{\partial \lambda} + \frac{\partial X_3^2}{\partial \lambda} - \dots & \frac{\partial X_1}{\partial \lambda} \frac{\partial X_1}{\partial \theta} + \frac{\partial X_2}{\partial \lambda} \frac{\partial X_2}{\partial \theta} + \frac{\partial X_3}{\partial \lambda} \frac{\partial X_3}{\partial \theta} - \dots \\ \frac{\partial X_1}{\partial \lambda} \frac{\partial X_1}{\partial \theta} + \frac{\partial X_2}{\partial \lambda} \frac{\partial X_2}{\partial \theta} + \frac{\partial X_3}{\partial \lambda} \frac{\partial X_3}{\partial \theta} - \dots & \frac{\partial X_1^2}{\partial \theta} + \frac{\partial X_2^2}{\partial \theta} + \frac{\partial X_3^2}{\partial \theta} - \dots \end{pmatrix}. \quad (\text{A.9})$$

Eq. (A.9) allows us to easily compute derivatives with respect to the entries of $(\nabla_\theta \mathbf{X})^T$. For example,

$$\left(\frac{\partial \gamma}{\partial(\nabla_\theta \mathbf{X})^T} \right)_{ij1\ell} = \frac{\partial \gamma}{\partial \left(\frac{\partial X_\ell}{\partial \lambda} \right)} = \begin{pmatrix} \frac{\partial X_\ell}{\partial \lambda} & \frac{1}{2} \frac{\partial X_\ell}{\partial \theta} \\ \frac{1}{2} \frac{\partial X_\ell}{\partial \lambda} & 0 \end{pmatrix} \quad (\text{A.10})$$

and

$$\left(\frac{\partial \gamma}{\partial(\nabla_\theta \mathbf{X})^T} \right)_{ij2\ell} = \frac{\partial \gamma}{\partial \left(\frac{\partial X_\ell}{\partial \theta} \right)} = \begin{pmatrix} 0 & \frac{1}{2} \frac{\partial X_\ell}{\partial \lambda} \\ \frac{1}{2} \frac{\partial X_\ell}{\partial \lambda} & \frac{\partial X_\ell}{\partial \theta} \end{pmatrix}. \quad (\text{A.11})$$

The full expansion is

$$\frac{\partial W}{\partial \gamma} : \frac{\partial \gamma}{\partial(\nabla_\theta \mathbf{X})^T} = \mathcal{S} : \frac{\partial \gamma}{\partial(\nabla_\theta \mathbf{X})^T} = \sum_{i,j} \mathcal{S}_{ij} \mathbf{e}_i \mathbf{e}_j : \sum_{m,p,k,\ell} \left(\frac{\partial \gamma}{\partial(\nabla_\theta \mathbf{X})^T} \right)_{mpk\ell} \mathbf{e}_m \mathbf{e}_p \mathbf{e}_k \mathbf{e}_\ell. \quad (\text{A.12})$$

Recall that $\mathbf{e}_i \mathbf{e}_j : \mathbf{e}_m \mathbf{e}_p = 1$ if $j = m$ and $i = p$ and is 0 otherwise. This reduces the (k, ℓ) entry of Eq. (A.12) to

$$\left(\mathcal{S} : \frac{\partial \gamma}{\partial(\nabla_\theta \mathbf{X})^T} \right)_{k\ell} = \sum_{i,j} \mathcal{S}_{ij} \left(\frac{\partial \gamma}{\partial(\nabla_\theta \mathbf{X})^T} \right)_{jik\ell}, \quad (\text{A.13})$$

or in matrix form,

$$\mathcal{S} : \frac{\partial \gamma}{\partial (\nabla_{\theta} \mathbf{X})^T} = \begin{pmatrix} \mathcal{S}_{11} \frac{\partial X_1}{\partial \lambda} + \frac{1}{2} \mathcal{S}_{12} \frac{\partial X_1}{\partial \theta} + \frac{1}{2} \mathcal{S}_{21} \frac{\partial X_1}{\partial \theta} & \mathcal{S}_{11} \frac{\partial X_2}{\partial \lambda} + \frac{1}{2} \mathcal{S}_{12} \frac{\partial X_2}{\partial \theta} + \frac{1}{2} \mathcal{S}_{21} \frac{\partial X_2}{\partial \theta} & \mathcal{S}_{11} \frac{\partial X_3}{\partial \lambda} + \frac{1}{2} \mathcal{S}_{12} \frac{\partial X_3}{\partial \theta} + \frac{1}{2} \mathcal{S}_{21} \frac{\partial X_3}{\partial \theta} \\ \frac{1}{2} \mathcal{S}_{12} \frac{\partial X_1}{\partial \lambda} + \frac{1}{2} \mathcal{S}_{21} \frac{\partial X_1}{\partial \lambda} + \mathcal{S}_{22} \frac{\partial X_1}{\partial \theta} & \frac{1}{2} \mathcal{S}_{12} \frac{\partial X_2}{\partial \lambda} + \frac{1}{2} \mathcal{S}_{21} \frac{\partial X_2}{\partial \lambda} + \mathcal{S}_{22} \frac{\partial X_2}{\partial \theta} & \frac{1}{2} \mathcal{S}_{12} \frac{\partial X_3}{\partial \lambda} + \frac{1}{2} \mathcal{S}_{21} \frac{\partial X_3}{\partial \lambda} + \mathcal{S}_{22} \frac{\partial X_3}{\partial \theta} \end{pmatrix}. \quad (\text{A.14})$$

By the symmetry of \mathcal{S} , we have

$$\mathcal{S} : \frac{\partial \gamma}{\partial (\nabla_{\theta} \mathbf{X})^T} = \begin{pmatrix} \mathcal{S}_{11} \frac{\partial X_1}{\partial \lambda} + \mathcal{S}_{12} \frac{\partial X_1}{\partial \theta} & \mathcal{S}_{11} \frac{\partial X_2}{\partial \lambda} + \mathcal{S}_{12} \frac{\partial X_2}{\partial \theta} & \mathcal{S}_{11} \frac{\partial X_3}{\partial \lambda} + \mathcal{S}_{12} \frac{\partial X_3}{\partial \theta} \\ \mathcal{S}_{21} \frac{\partial X_1}{\partial \lambda} + \mathcal{S}_{22} \frac{\partial X_1}{\partial \theta} & \mathcal{S}_{21} \frac{\partial X_2}{\partial \lambda} + \mathcal{S}_{22} \frac{\partial X_2}{\partial \theta} & \mathcal{S}_{21} \frac{\partial X_3}{\partial \lambda} + \mathcal{S}_{22} \frac{\partial X_3}{\partial \theta} \end{pmatrix}. \quad (\text{A.15})$$

From the dot product,

$$\mathcal{S} \cdot (\nabla_{\theta} \mathbf{X})^T = (\mathcal{S}_{11} \mathbf{e}_1 \mathbf{e}_1 + \mathcal{S}_{12} \mathbf{e}_1 \mathbf{e}_2 + \mathcal{S}_{21} \mathbf{e}_2 \mathbf{e}_1 + \mathcal{S}_{22} \mathbf{e}_2 \mathbf{e}_2) \cdot \left(\frac{\partial X_1}{\partial \lambda} \mathbf{e}_1 \mathbf{e}_1 + \frac{\partial X_2}{\partial \lambda} \mathbf{e}_1 \mathbf{e}_2 + \frac{\partial X_3}{\partial \lambda} \mathbf{e}_1 \mathbf{e}_3 + \frac{\partial X_1}{\partial \theta} \mathbf{e}_2 \mathbf{e}_1 + \frac{\partial X_2}{\partial \theta} \mathbf{e}_2 \mathbf{e}_2 + \frac{\partial X_3}{\partial \theta} \mathbf{e}_2 \mathbf{e}_3 \right) \quad (\text{A.16})$$

$$= \begin{pmatrix} \mathcal{S}_{11} \frac{\partial X_1}{\partial \lambda} + \mathcal{S}_{12} \frac{\partial X_1}{\partial \theta} & \mathcal{S}_{11} \frac{\partial X_2}{\partial \lambda} + \mathcal{S}_{12} \frac{\partial X_2}{\partial \theta} & \mathcal{S}_{11} \frac{\partial X_3}{\partial \lambda} + \mathcal{S}_{12} \frac{\partial X_3}{\partial \theta} \\ \mathcal{S}_{21} \frac{\partial X_1}{\partial \lambda} + \mathcal{S}_{22} \frac{\partial X_1}{\partial \theta} & \mathcal{S}_{21} \frac{\partial X_2}{\partial \lambda} + \mathcal{S}_{22} \frac{\partial X_2}{\partial \theta} & \mathcal{S}_{21} \frac{\partial X_3}{\partial \lambda} + \mathcal{S}_{22} \frac{\partial X_3}{\partial \theta} \end{pmatrix} \quad (\text{A.17})$$

$$= \mathcal{S} : \frac{\partial \gamma}{\partial (\nabla_{\theta} \mathbf{X})^T}, \quad (\text{A.18})$$

so that we recover Eq. (26).

Appendix A.3. Chain rule expansion of \mathcal{S}

As observed in Eq. (28), the second Piola-Kirchhoff stress tensor \mathcal{S} can be computed by observing that $\mathcal{C} = \mathcal{I}_2 + 2\gamma \mathcal{G}_0^{-1}$, where \mathcal{I}_2 is the rank 2 identity tensor. Applying the chain rule twice yields

$$\mathcal{S} = \frac{\partial W}{\partial \gamma} = \frac{\partial W}{\partial \mathcal{C}} : \frac{\partial \mathcal{C}}{\partial \gamma} = 2 \frac{\partial W}{\partial \mathcal{C}} \cdot \mathcal{G}_0^{-1}, \quad (\text{A.19})$$

and

$$\frac{\partial W}{\partial \mathcal{C}} = \frac{\partial W}{\partial I_1} \frac{\partial I_1}{\partial \mathcal{C}} + \frac{\partial W}{\partial I_2} \frac{\partial I_2}{\partial \mathcal{C}}. \quad (\text{A.20})$$

The rank 2 tensor $\frac{\partial I_j}{\partial \mathcal{C}}$ is obtained by taking the derivative of the invariant I_j with respect to each element of \mathcal{C} . Carrying out these calculations,

$$I_1 = \text{tr } \mathcal{C} - 2 = \mathcal{C}_{11} + \mathcal{C}_{22} - 2, \quad (\text{A.21})$$

$$\begin{aligned} \frac{\partial I_1}{\partial \mathcal{C}_{11}} &= 1, & \frac{\partial I_1}{\partial \mathcal{C}_{12}} &= 0, \\ \frac{\partial I_1}{\partial \mathcal{C}_{21}} &= 0, & \frac{\partial I_1}{\partial \mathcal{C}_{22}} &= 1, \end{aligned} \quad (\text{A.22})$$

$$\frac{\partial I_1}{\partial \mathcal{C}} = \mathcal{I}_2, \quad (\text{A.23})$$

$$I_2 = \det \mathcal{C} - 1 = \mathcal{C}_{11}\mathcal{C}_{22} - \mathcal{C}_{12}\mathcal{C}_{21} - 1, \quad (\text{A.24})$$

$$\begin{aligned} \frac{\partial I_2}{\partial \mathcal{C}_{11}} &= \mathcal{C}_{22}, & \frac{\partial I_2}{\partial \mathcal{C}_{12}} &= -\mathcal{C}_{21}, \\ \frac{\partial I_2}{\partial \mathcal{C}_{21}} &= -\mathcal{C}_{12}, & \frac{\partial I_2}{\partial \mathcal{C}_{22}} &= \mathcal{C}_{11}, \end{aligned} \quad (\text{A.25})$$

$$\frac{\partial I_2}{\partial \mathcal{C}} = (\det \mathcal{C})(\mathcal{C}^{-1})^T = (\det \mathcal{C}) ((\mathcal{G}\mathcal{G}_0^{-1})^{-1})^T = (\det \mathcal{C}) (\mathcal{G}_0\mathcal{G}^{-1})^T = (\det \mathcal{C}) (\mathcal{G}^{-1}\mathcal{G}_0). \quad (\text{A.26})$$

The last equality follows from the symmetry of \mathcal{G} and \mathcal{G}_0 . At last we have that

$$\mathcal{S} = 2 \frac{\partial W}{\partial \mathcal{C}} \cdot \mathcal{G}_0^{-1} = 2 \left(\frac{\partial W}{\partial I_1} \frac{\partial I_1}{\partial \mathcal{C}} + \frac{\partial W}{\partial I_2} \frac{\partial I_2}{\partial \mathcal{C}} \right) \cdot \mathcal{G}_0^{-1} \quad (\text{A.27})$$

$$= 2 \left(\frac{\partial W}{\partial I_1} \mathcal{I}_2 + \frac{\partial W}{\partial I_2} (\det \mathcal{C}) (\mathcal{G}^{-1}\mathcal{G}_0) \right) \cdot \mathcal{G}_0^{-1} \quad (\text{A.28})$$

$$\mathcal{S} = 2 \frac{\partial W}{\partial I_1} \mathcal{G}_0^{-1} + 2 \frac{\partial W}{\partial I_2} (\det \mathcal{C}) \mathcal{G}^{-1}. \quad (\text{A.29})$$

Appendix A.4. The force density

Computing the force density in Eq. (24), we use the chain rule expansions of \mathcal{P}^T and \mathcal{S} :

$$\nabla_\theta \cdot = \frac{1}{\sqrt{\det \mathcal{G}_0}} \left(\frac{\partial}{\partial \lambda} \sqrt{\det \mathcal{G}_0} \mathbf{e}_1 + \frac{\partial}{\partial \theta} \sqrt{\det \mathcal{G}_0} \mathbf{e}_2 \right) \cdot, \quad (\text{A.30})$$

$$\mathbf{F} = \nabla_\theta \cdot \mathcal{P}^T = \nabla_\theta \cdot (\mathcal{S} \cdot (\nabla_\theta \mathbf{X})^T), \quad (\text{A.31})$$

$$\begin{aligned} \mathbf{F} = \nabla_\theta \cdot & \left(\left(\mathcal{S}_{11} \frac{\partial X_1}{\partial \lambda} + \mathcal{S}_{12} \frac{\partial X_1}{\partial \theta} \right) \mathbf{e}_1 \mathbf{e}_1 + \left(\mathcal{S}_{11} \frac{\partial X_2}{\partial \lambda} + \mathcal{S}_{12} \frac{\partial X_2}{\partial \theta} \right) \mathbf{e}_1 \mathbf{e}_2 \right. \\ & + \left(\mathcal{S}_{11} \frac{\partial X_3}{\partial \lambda} + \mathcal{S}_{12} \frac{\partial X_3}{\partial \theta} \right) \mathbf{e}_1 \mathbf{e}_3 + \left(\mathcal{S}_{21} \frac{\partial X_1}{\partial \lambda} + \mathcal{S}_{22} \frac{\partial X_1}{\partial \theta} \right) \mathbf{e}_2 \mathbf{e}_1 \\ & \left. + \left(\mathcal{S}_{21} \frac{\partial X_2}{\partial \lambda} + \mathcal{S}_{22} \frac{\partial X_2}{\partial \theta} \right) \mathbf{e}_2 \mathbf{e}_2 + \left(\mathcal{S}_{21} \frac{\partial X_3}{\partial \lambda} + \mathcal{S}_{22} \frac{\partial X_3}{\partial \theta} \right) \mathbf{e}_2 \mathbf{e}_3 \right). \end{aligned} \quad (\text{A.32})$$

Thus

$$\mathbf{F} = \frac{1}{\sqrt{\det \mathcal{G}_0}} \left(\frac{\partial}{\partial \lambda} \sqrt{\det \mathcal{G}_0} \left(\mathcal{S}_{11} \frac{\partial X_1}{\partial \lambda} + \mathcal{S}_{12} \frac{\partial X_1}{\partial \theta} \right) + \frac{\partial}{\partial \theta} \sqrt{\det \mathcal{G}_0} \left(\mathcal{S}_{21} \frac{\partial X_1}{\partial \lambda} + \mathcal{S}_{22} \frac{\partial X_1}{\partial \theta} \right) \right. \\ \left. \frac{\partial}{\partial \lambda} \sqrt{\det \mathcal{G}_0} \left(\mathcal{S}_{11} \frac{\partial X_2}{\partial \lambda} + \mathcal{S}_{12} \frac{\partial X_2}{\partial \theta} \right) + \frac{\partial}{\partial \theta} \sqrt{\det \mathcal{G}_0} \left(\mathcal{S}_{21} \frac{\partial X_2}{\partial \lambda} + \mathcal{S}_{22} \frac{\partial X_2}{\partial \theta} \right) \right. \\ \left. \frac{\partial}{\partial \lambda} \sqrt{\det \mathcal{G}_0} \left(\mathcal{S}_{11} \frac{\partial X_3}{\partial \lambda} + \mathcal{S}_{12} \frac{\partial X_3}{\partial \theta} \right) + \frac{\partial}{\partial \theta} \sqrt{\det \mathcal{G}_0} \left(\mathcal{S}_{21} \frac{\partial X_3}{\partial \lambda} + \mathcal{S}_{22} \frac{\partial X_3}{\partial \theta} \right) \right) \quad (\text{A.33})$$

$$\mathbf{F} = \frac{1}{\sqrt{\det \mathcal{G}_0}} \left(\frac{\partial}{\partial \lambda} \sqrt{\det \mathcal{G}_0} \left(\mathcal{S}_{11} \frac{\partial \mathbf{X}}{\partial \lambda} + \mathcal{S}_{12} \frac{\partial \mathbf{X}}{\partial \theta} \right) + \frac{\partial}{\partial \theta} \sqrt{\det \mathcal{G}_0} \left(\mathcal{S}_{21} \frac{\partial \mathbf{X}}{\partial \lambda} + \mathcal{S}_{22} \frac{\partial \mathbf{X}}{\partial \theta} \right) \right). \quad (\text{A.34})$$

Eq. (A.34) recovers Eq. (35) from the main text.

References

- [1] C. S. Peskin, Flow patterns around heart valves: a numerical method, J. Comput. Phys. 10 (2) (1972) 252–271. doi:10.1016/0021-9991(72)90065-4.

- [2] L. A. Miller, C. S. Peskin, A computational fluid dynamics of ‘clap and fling’ in the smallest insects, *J. Exp. Biol.* 208 (2) (2005) 195–212. doi:10.1242/jeb.01376.
- [3] L. A. Miller, C. S. Peskin, When vortices stick: an aerodynamic transition in tiny insect flight, *J. Exp. Biol.* 207 (17) (2004) 3073–3088. doi:10.1242/jeb.01138.
- [4] L. J. Fauci, C. S. Peskin, A computational model of aquatic animal locomotion, *J. Comput. Phys.* 77 (1) (1988) 85–108. doi:10.1016/0021-9991(88)90158-1.
- [5] W. Strychalski, C. A. Copos, O. L. Lewis, R. D. Guy, A poroelastic immersed boundary method with applications to cell biology, *J. Comput. Phys.* 282 (2015) 77–97. doi:10.1016/j.jcp.2014.10.004.
- [6] R. P. Beyer, A computational model of the cochlea using the immersed boundary method, *J. Comput. Phys.* 98 (1) (1992) 145–162. doi:10.1016/0021-9991(92)90180-7.
- [7] B. Griffith, X. Luo, L. H. Charney, Hybrid finite difference/finite element version of the immersed boundary method, *Int. J. Numer. Meth. Bio.* (2017) e2888 doi:10.1002/cnm.2888.
- [8] D. Boffi, L. Gastaldi, A finite element approach for the immersed boundary method, *Comput. Struct.* 81 (8) (2003) 491–501. doi:10.1016/S0045-7949(02)00404-2.
- [9] D. Boffi, L. Gastaldi, L. Heltai, C. S. Peskin, On the hyper-elastic formulation of the immersed boundary method, *Comput. Method Appl. M.* 197 (25) (2008) 2210–2231. doi:10.1016/j.cma.2007.09.015.
- [10] D. Boffi, L. Gastaldi, L. Heltai, On the CFL condition for the finite element immersed boundary method, *Comput. Struct.* 85 (11) (2007) 775–783. doi:10.1016/j.compstruc.2007.01.009.
- [11] D. Devendran, C. S. Peskin, An immersed boundary energy-based method for incompressible viscoelasticity, *J. Comput. Phys.* 231 (14) (2012) 4613–4642. doi:10.1016/j.jcp.2012.02.020.
- [12] T. G. Fai, B. E. Griffith, Y. Mori, C. S. Peskin, Immersed boundary method for variable viscosity and variable density problems using fast constant-coefficient linear solvers I: Numerical method and results, *SIAM J. Sci. Comput.* 35 (5) (2013) B1132–B1161. doi:10.1137/120903038.
- [13] C.-H. Wu, T. G. Fai, P. J. Atzberger, C. S. Peskin, Simulation of osmotic swelling by the stochastic immersed boundary method, *SIAM J. Sci. Comput.* 37 (4) (2015) B660–B688. doi:10.1137/14098404X.
- [14] V. Camacho, A. Fogelson, J. Keener, Eulerian–lagrangian treatment of nondilute two-phase gels, *SIAM J. Appl. Math.* 76 (1) (2016) 341–367. doi:10.1137/15M1023579.
- [15] V. Shankar, G. B. Wright, A. L. Fogelson, R. M. Kirby, A study of different modeling choices for simulating platelets within the immersed boundary method, *Appl. Numer. Math.* 63 (2013) 58–77. doi:10.1016/j.apnum.2012.09.006.
- [16] V. Shankar, G. B. Wright, R. M. Kirby, A. L. Fogelson, Augmenting the immersed boundary method with radial basis functions (rbfs) for the modeling of platelets in hemodynamic flows, *Int. J. Numer. Math. Fl.* 79 (10) (2015) 536–557. doi:10.1002/flid.4061.
- [17] A. Kumar, M. D. Graham, Cell distribution and segregation phenomena during blood flow, in: *Complex Fluids in Biological Systems*, Springer, 2015, pp. 399–435.
- [18] D. Barthès-Biesel, Motion and deformation of elastic capsules and vesicles in flow, *Annu. Rev. Fluid Mech.* 48 (2016) 25–52. doi:10.1146/annurev-fluid-122414-034345.
- [19] C. Pozrikidis, *Boundary integral and singularity methods for linearized viscous flow*, Cambridge University Press, 1992.
- [20] C. Pozrikidis, Interfacial dynamics for Stokes flow, *J. Comput. Phys.* 169 (2) (2001) 250–301. doi:10.1006/jcph.2000.6582.

- [21] H. Zhao, A. H. Isfahani, L. N. Olson, J. B. Freund, A spectral boundary integral method for flowing blood cells, *J. Comput. Phys.* 229 (10) (2010) 3726–3744. doi:10.1016/j.jcp.2010.01.024.
- [22] S. K. Veerapaneni, A. Rahimian, G. Biros, D. Zorin, A fast algorithm for simulating vesicle flows in three dimensions, *J. Comput. Phys.* 230 (14) (2011) 5610–5634. doi:10.1016/j.jcp.2011.03.045.
- [23] G. Batchelor, *An Introduction to Fluid Mechanics*, Cambridge University Press, New York.
- [24] J. M. Teran, C. S. Peskin, Tether force constraints in Stokes flow by the immersed boundary method on a periodic domain, *SIAM J. Sci. Comput.* 31 (5) (2009) 3404–3416. doi:10.1137/080720217.
- [25] C. S. Peskin, The immersed boundary method, *Acta Numer.* 11 (2002) 479–517. doi:10.1017/S0962492902000077.
- [26] M. Kraus, W. Wintz, U. Seifert, R. Lipowsky, Fluid vesicles in shear flow, *Phys. Rev. Lett.* 77 (17) (1996) 3685. doi:10.1103/PhysRevLett.77.3685.
- [27] K. D. Hjelmstad, *Fundamentals of structural mechanics*, Springer Science & Business Media, 2007.
- [28] R. W. Ogden, *Non-linear elastic deformations*, Courier Corporation, 1997.
- [29] M. Gräf, S. Kunis, D. Potts, On the computation of nonnegative quadrature weights on the sphere, *Appl. Comput. Harmon. A.* 27 (1) (2009) 124–132. doi:10.1016/j.acha.2008.12.003.
- [30] D. Kim, S. Sra, I. S. Dhillon, A non-monotonic method for large-scale non-negative least squares, *Optim. Method. Softw.* 28 (5) (2013) 1012–1039. doi:10.1080/10556788.2012.656368.
- [31] R. S. Womersley, I. H. Sloan, How good can polynomial interpolation on the sphere be?, *Adv. Comput. Math.* 14 (3) (2001) 195–226. doi:10.1023/A:1016630227163.
- [32] I. H. Sloan, R. S. Womersley, Extremal systems of points and numerical integration on the sphere, *Adv. Comput. Math.* 21 (1-2) (2004) 107–125. doi:10.1023/B:ACOM.0000016428.25905.da.
- [33] R. Womersley. doi:http://web.maths.unsw.edu.au/~rsw/Sphere/Extremal/New/index.html#S:Vals.
- [34] E. Fuselier, T. Hangelsbroek, F. J. Narcowich, J. D. Ward, G. B. Wright, Kernel based quadrature on spheres and other homogeneous spaces, *Numer. Math.* 127 (1) (2014) 57–92. doi:10.1007/s00211-013-0581-1.
- [35] E. A. Evans, R. Skalak, S. Weinbaum, *Mechanics and thermodynamics of biomembranes* (1980).
- [36] R. Capovilla, Elastic bending energy: a variational approach, arXiv preprint 1709.04399.
- [37] C. Pozrikidis, *Modeling and simulation of capsules and biological cells*, CRC Press, 2003.
- [38] O.-Y. Zhong-Can, W. Helfrich, Bending energy of vesicle membranes: General expressions for the first, second, and third variation of the shape energy and applications to spheres and cylinders, *Phys. Rev. A* 39 (10) (1989) 5280. doi:10.1103/PhysRevA.39.5280.
- [39] R. Capovilla, J. Guven, Second variation of the Helfrich–Canham Hamiltonian and reparametrization invariance, *J. Phys. A- Math. Gen.* 37 (23) (2004) 5983. doi:10.1088/0305-4470/37/23/003.
- [40] R. Peyret, *Spectral Methods for Incompressible Viscous Flow*, no. 148 in *Applied Mathematical Sciences*, Springer, 2002.
- [41] C. Jekeli, Spherical harmonic analysis, aliasing, and filtering, *J. Geodesy* 70 (4) (1996) 214–223. doi:10.1007/BF00873702.
- [42] M.-C. Lai, C. S. Peskin, An immersed boundary method with formal second-order accuracy and reduced numerical viscosity, *J. Comput. Phys.* 160 (2) (2000) 705–719. doi:10.1006/jcph.2000.6483.

- [43] B. E. Griffith, C. S. Peskin, On the order of accuracy of the immersed boundary method: Higher order convergence rates for sufficiently smooth problems, *J. Comput. Phys.* 208 (1) (2005) 75–105. doi:10.1016/j.jcp.2005.02.011.
- [44] C. Pozrikidis, Axisymmetric motion of a file of red blood cells through capillaries, *Phys. Fluids* 17 (3) (2005) 031503. doi:10.1063/1.1830484.
- [45] K. Tsukada, E. Sekizuka, C. Oshio, H. Minamitani, Direct measurement of erythrocyte deformability in diabetes mellitus with a transparent microchannel capillary model and high-speed video camera system, *Microvasc. Res.* 61 (3) (2001) 231–239. doi:10.1006/mvre.2001.2307.
- [46] T. Omori, T. Ishikawa, D. Barthès-Biesel, A.-V. Salsac, Y. Imai, T. Yamaguchi, Tension of red blood cell membrane in simple shear flow, *Phys. Rev. E* 86 (5) (2012) 056321. doi:10.1103/PhysRevE.86.056321.
- [47] A. Z. Yazdani, P. Bagchi, Phase diagram and breathing dynamics of a single red blood cell and a biconcave capsule in dilute shear flow, *Phys. Rev. E* 84 (2) (2011) 026314. doi:10.1103/PhysRevE.84.026314.
- [48] G. Charras, E. Paluch, Blebs lead the way: how to migrate without lamellipodia, *Nat. Rev. Mol. Cell Biol.* 9 (9) (2008) 730–736. doi:10.1038/nrm2453.
- [49] G. T. Charras, M. Coughlin, T. J. Mitchison, L. Mahadevan, Life and Times of a Cellular Bleb, *Biophys. J.* 94 (5) (2008) 1836–53. doi:10.1529/biophysj.107.113605.
- [50] J. Y. Tinevez, U. Schulze, G. Salbreux, J. Roensch, J. F. Joanny, E. Paluch, Role of cortical tension in bleb growth, *Proc. Natl. Acad. Sci. USA* 106 (44) (2009) 18581–18586. doi:10.1073/pnas.0903353106.
- [51] T. E. Woolley, E. A. Gaffney, J. M. Oliver, S. L. Waters, R. E. Baker, A. Goriely, Global contraction or local growth, bleb shape depends on more than just cell structure, *J. Theor. Biol.* 380 (7) (2015) 83–97. doi:10.1016/j.jtbi.2015.04.023.
- [52] W. Strychalski, R. D. Guy, A computational model of bleb formation, *Math. Med. Biol.* 30 (2) (2013) 115–130. doi:10.1093/imamb/dqr030.
- [53] W. Strychalski, R. D. Guy, Intracellular Pressure Dynamics in Blebbing Cells, *Biophys. J.* 110 (5) (2016) 1168–1179. doi:10.1016/j.bpj.2016.01.012.
- [54] J. M. Sullivan, Curvatures of smooth and discrete surfaces, in: *Discrete differential geometry*, Vol. 38, Springer, 2008, pp. 175–188.
- [55] W. Strychalski, C. A. Copos, O. L. Lewis, R. D. Guy, A poroelastic immersed boundary method with applications to cell biology, *J. Comput. Phys.* 282 (2015) 77–97. doi:10.1016/j.jcp.2014.10.004.
- [56] G. T. Charras, M. Coughlin, T. J. Mitchison, L. Mahadevan, Life and Times of a Cellular Bleb, *Biophys. J.* 94 (5) (2008) 1836–1853. doi:10.1529/biophysj.107.113605.
- [57] E. Moeendarbary, L. Valon, M. Fritzsche, A. R. Harris, D. A. Moulding, A. J. Thrasher, E. Stride, L. Mahadevan, G. T. Charras, The cytoplasm of living cells behaves as a poroelastic material, *Nat. Mater.* 12 (3) (2013) 253–261. doi:10.1038/nmat3517.
- [58] Z. Li, K. Ito, The immersed interface method: numerical solutions of PDEs involving interfaces and irregular domains, Vol. 33, Siam, 2006.
- [59] S. Xu, Z. J. Wang, A 3d immersed interface method for fluid–solid interaction, *Comput. Methods in Appl. Mech. Eng.* 197 (25–28) (2008) 2068–2086. doi:10.1016/j.cma.2007.06.012.
- [60] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, P. Walter, *Molecular biology of the cell*, 4th Edition, Garland Science, New York, 2002.

- [61] R. Shlomovitz, N. S. Gov, Physical model of contractile ring initiation in dividing cells, *Biophys. J.* 94 (4) (2008) 1155–1168. doi:10.1529/biophysj.107.111351.