Commun. Comput. Phys.
doi: 10.4208/cicp.OA-2015-0029

Vol. **21**, No. 1, pp. 40-64
January 2017

Fast Simulation of Lipid Vesicle Deformation Using Spherical Harmonic Approximation

Michael Mikucki[1] and Yongcheng Zhou[2,*]

[1] *Department of Applied Mathematics & Statistics, Colorado School of Mines, Golden, Colorado, 80401, USA.*
[2] *Department of Mathematics, Colorado State University, Fort Collins, Colorado, 80523, USA.*

Received 24 December 2015; Accepted (in revised version) 26 May 2016

Abstract. Lipid vesicles appear ubiquitously in biological systems. Understanding how the mechanical and intermolecular interactions deform vesicle membranes is a fundamental question in biophysics. In this article we develop a fast algorithm to compute the surface configurations of lipid vesicles by introducing surface harmonic functions to approximate the membrane surface. This parameterization allows an analytical computation of the membrane curvature energy and its gradient for the efficient minimization of the curvature energy using a nonlinear conjugate gradient method. Our approach drastically reduces the degrees of freedom for approximating the membrane surfaces compared to the previously developed finite element and finite difference methods. Vesicle deformations with a reduced volume larger than 0.65 can be well approximated by using as small as 49 surface harmonic functions. The method thus has a great potential to reduce the computational expense of tracking multiple vesicles which deform for their interaction with external fields.

AMS subject classifications: 35Q92, 92C40, 65M70

Key words: Lipid bilayer, curvature energy, surface harmonics, fast algorithm.

1 Introduction

This paper describes a fast numerical algorithm for computing the configuration of lipid bilayer vesicles. Lipid bilayers are crucial components to living systems. Being amphiphilic, lipid molecules have charged or polar hydrophilic head groups and hydrophobic tails. This allows lipids in aqueous solution to aggregate into structures that entropically favor the alignment of hydrophobic tails and the exposure of hydrophilic head

*Corresponding author. *Email addresses:* mikucki@mines.edu (M. Mikucki), yzhou@math.colostate.edu (Y. Zhou)

 ©2016 Global-Science Press

https://doi.org/10.4208/cicp.OA-2015-0029 Published online by Cambridge University Press

groups to water. A lipid bilayer is formed from the self-assembly of hydrophobic tails of the two complementary layers. The bilayer will then close so the hydrophobic core will not be exposed at the free edges, forming membranes of cells and sub-cellular organelles. These membranes are semi-permeable boundaries separating the enclosure from the surrounding environment. At the microscale, lipid bilayer membranes regulate the transportation of ions, proteins, and other molecules between separated domains, and provide a flexible platform on which molecules can aggregate to carry out vital chemical or physical reactions. At the mesoscale, for example, membranes of the red blood cells (RBCs) suspended in blood flow change their shape in response to the local flow conditions, and this change will in turn affect the RBC's ability of oxygen transport and the hydrodynamic properties of the blood flow [1,10]. Deformation of the bilayer membranes are driven by various types of force, which may dominate at different length scales. At the microscale, driven forces are mainly the results of protein-membrane or membrane-membrane interactions, such as protein binding or insertion, lipid insertion or translation, and ubiquitous electrostatic interactions, to name a few [13]. At the mesoscale, hydrodynamic forces usually dominate [5, 18, 36]. Determination of the membrane geometry in response to protein-membrane, membrane-membrane, or fluid-membrane interactions is necessary for elucidating the structure-function relation of these interacted biological systems.

The variation of lipid bilayer configurations can be characterized by its deformation energy. This energy is the handle of almost all computational methods. Classical strain energy can be defined for lipid bilayers so their deformation can be described as elastic plates [43]. A crucial difference between the plates and bilayer membranes is missing in such models: a flat membrane can be subject to some shear deformation with zero energy cost provided the deformation is so slow that the viscous effect of lipids is negligible [26]. As a result, the deformation energy of a lipid bilayer is mostly attributed to the bending energy of the monolayers. The classical energy forms proposed by Canham [3], Helfrich [17], and Evans [11], are of this type, in which the deformation energy is defined be a quadratic function of the principle curvature of the surface. Other components of the energy, such as those corresponding to the area expansion and contraction of the monolayers, and the osmotic pressure, are of several orders of magnitude larger than the bending energy, and usually serve in computational models as area and enclosed volume constraints of the bilayer membrane, respectively. With this simplification, often referred to as the spontaneous curvature model, the deformation energy of the bilayer membrane is given by the bending curvature equation with two constraints. The equilibrium configuration of lipid bilayer vesicles can be obtained by minimizing this energy. For sufficiently simplified membrane systems such as isolated vesicles with symmetric lipid composition, analytical analysis of these energies can give an excellent classification of the phases of the vesicle configurations, particularly the axisymmetric configurations, and can accurately locate the conditions under which the phase transition occurs [28, 29, 42]. Giving the complexity the realistic interacted biological system with which lipid bilayer is interacted, analytical approaches may fail in quantifying these interactions and thus

https://doi.org/10.4208/cicp.OA-2015-0029 Published online by Cambridge University Press

computational methods become indispensable.

The solutions of the energy minimization problem have been computed in various ways. Solving the Euler-Lagrange equations directly has so far been restricted by the surface parameterization to solutions which are axisymmetric [14, 28, 29]. One alternative is to minimize the energy over a smaller subspace of membrane configurations. On triangulated vesicle surface, the subspace can be that spanned by the basis functions at triangular vertices. In this subspace the deformation energy can be approximated using Rayleigh-Ritz procedure [3, 16] or finite element methods [14, 23]. In case that the triangular mesh needs to be locally refined to resolve a very large local curvature, the number of basis functions becomes large, leading to a significant increase of computational cost of numerical minimization. The other alternative is to define the surface of a lipid bilayer vesicle as the level set of a phase field function. Geometrical properties of the vesicle surface can be represented using the phase field function, and the equilibrium surface configuration can be obtained by following the gradient flow driven by the membrane curvature. Approaches of this type have been very successful in describing the nonaxisymmetric equilibrium configurations [7–9], membranes with multiple lipid species [35], membranes with surfactant sedimentation or solvation [34, 39], and membrane-flow interactions [22, 31]. Moreover, the phase field approach is the only computational method that can simulate topological changes of the surface during the merging or separation of vesicles. The spontaneous curvature model involves the integration of the Gaussian curvature, which according to the Gauss-Bonet theorem, is a topological constant [32]. As such, the classical model cannot account for topological changes, and approaches based on the Eulerian description of the configuration such as the phase field method shall be used for such phenomena. Nevertheless, there is a significant increase in the computational cost with phase field methods, because the tracking of the 2-D vesicle surface is replaced by the evolution of 3-D phase field function. Special numerical techniques, such as spectral methods [7–9], finite difference or finite element methods with local mesh refinement [37, 41] have been developed to accelerate the simulations. We note that the molecular details of bilayer membrane are not considered in this study, but similar Eulerian and Lagrangian representations of surfaces have been established if such details are necessary in the modeling [15, 38].

There are many important applications where fast algorithms are necessary for simulating vesicle deformation that does not necessarily involve topological change. These include the vesicle deformations induced by protein clusters [13, 33] and interactions between blood flow in micro-vessels and the deformable RBCs [1, 10]. For example, about 7778 nodes are used to describe the deformation of spherical vesicles in 3-D simulations of single vesicle-blood flow interactions using immersed boundary method [10]. In a 2-D simulation of multiple vesicle-blood flow interactions, about 128 to 512 nodes are used to track the deformation of a single vesicle at a satisfactory resolution [1]. A full 3-D simulation of this type can be even more expensive. An accurate and simplified representation of vesicle configuration will greatly improve the efficiency of these computational investigations.

https://doi.org/10.4208/cicp.OA-2015-0029 Published online by Cambridge University Press

In this article, we approximate the deformation energy of lipid vesicles using surface harmonic functions. These real-valued functions are the linear combination of complex-valued spherical harmonics functions. They share the same orthonormal and completeness properties as spherical harmonics, and thus provide a natural basis for vesicle configurations of star-shape. The uniform convergence of spherical harmonic functions in approximating smooth functions enables us to choose far fewer number of basis functions to approximate vesicle surface when compared to finite element methods with a triangular mesh. With the surface harmonic expansion, one can analytically compute the variations of the deformation energy with respect to the expansion coefficients. A nonlinear conjugate gradient method is employed to numerically compute the minimizer of the deformation energy, without incurring the computationally cost evaluation of Hessian matrix for the deformation energy. Our numerical experiments below show that 49 surface harmonic functions are sufficient to represent axisymmetric prolate, oblate, and stomatocyte shapes as well as many nonaxisymmetric shapes. We note that complex-valued spherical harmonic functions were used to compute the local curvature of vesicles deformed by protein clusters, where least square solution of an over-determined linear systems for spherical harmonic expansion coefficients is solved using singular value decomposition [2]. In a more relevant study, a complex-valued spherical harmonic expansion is sought for each individual Cartesian coordinate of the node on vesicle surface [20].

Curvature energy by itself can not account for the deformation and various biological functions of the membrane, the potential and kinetic energies arising in realistic intermolecular interactions shall be included in simulations [6, 24]. Although the vesicle deformation can be well represented using spherical harmonic functions as shown in this study, whether the modeling of various intermolecular interactions and the total energy can be facilitated by the spherical harmonic expansions is to be determined. For fluid flows, a simplified geometric representation of immersed vesicles will speed up the evaluation of flow field [40]. For Stokes flow in particular, the implementation of the Stokeslet can be accelerated by using the analytical expression of the vesicle surfaces. When this fast algorithm based on spherical harmonic approximations to be applied to minimize a total energy that ensembles both the deformation energy and the electrostatic potential energy arising from protein-membrane interactions, proper transformation is necessary to assign the electrostatic force computed at an arbitrary point on vesicle represented using surface harmonic functions.

The rest of the article is organized as follows. In Section 2, we present the bending energy of a lipid bilayer membrane and its variation along with the area and volume constraints. Section 3 introduces surface harmonics and establishes the variational problem in terms of the surface harmonics parameterization. Section 4 presents examples of equilibrium configurations and confirms these results from previous work in the literature. Finally, Section 5 discusses future developments of the model, including electrostatics from protein-membrane interactions.

https://doi.org/10.4208/cicp.OA-2015-0029 Published online by Cambridge University Press

2 Lipid bilayer mechanics

2.1 Total energy and variational problem

According to the classic spontaneous curvature model as developed by Canham [3], Helfrich [17], and Evans [11], the bending energy of the membrane is given by

$$E[\Gamma(\vec{x})] = \int_{\Gamma(\vec{x})} \left(\frac{1}{2}\mathscr{K}_C(2H - C_0)^2 + \mathscr{K}_G K \right) dS, \tag{2.1}$$

where $\mathscr{K}_C$ and $\mathscr{K}_G$ are the bending modulus and Gaussian saddle-splay modulus, respectively, H is the mean curvature, C_0 is the spontaneous curvature, K is the Gaussian curvature, and dS is the differential surface area element of the membrane [14]. The position of the membrane is given by $\Gamma(\vec{x})$, and so the total bending energy $E[\Gamma(\vec{x})]$ depends only upon the membrane position. In this work, we assume that the spontaneous curvature of the membrane C_0 is a constant.

In the case of protein-membrane interactions, or membranes immersed in fluid flow, the membrane will also be under an external force. The potential energy from such interactions may be added to the total potential energy of the system, denoted by $\Pi[\Gamma(\vec{x})]$:

$$\Pi[\Gamma(\vec{x})] = E[\Gamma(\vec{x})] + G[\Gamma(\vec{x})], \tag{2.2}$$

where $E[\Gamma(\vec{x})]$ is the bending energy (2.1) and $G[\Gamma(\vec{x})]$ is any external energy such as electrostatic or hydrodynamic energies. This allows for a quick expansion of this model to a broader set of applications. For this paper, we only consider the bending energy $E[\Gamma(\vec{x})]$.

Since both the energy required to expand or contract the monolayers and the energy from osmotic pressure are on higher orders of magnitude than the energy from bending, the minimization of the bending energy (2.1) is performed over membrane configurations which have continuous curvature and preserve the total surface area and volume. The space of admissible membrane configurations is therefore

$$H^2_{S_A,V}(\Gamma) = \{\vec{x} : \vec{x} \in H^2(\Gamma), S_A[\vec{x}] = \bar{S}, V[\vec{x}] = \bar{V}\}, \tag{2.3}$$

where H^2 is the standard Sobolev space, and $S_A[\vec{x}]$ and $V[\vec{x}]$ are the surface area and volume of the membrane, respectively, and $\bar{S}$ and $\bar{V}$ are the desired surface area and volume, respectively. The variational problem is

$$\min_{\vec{x} \in H^2_{S_A,V}(\Gamma)} E[\Gamma(\vec{x})]. \tag{2.4}$$

2.2 Constraints

The volume of the vesicle should remain constant due to the physical constraint of osmotic pressure. Additionally, the surface area should remain constant due to the physical constraint that the quantity of lipids are conserved. To compute the total energy and

https://doi.org/10.4208/cicp.OA-2015-0029 Published online by Cambridge University Press

account for the area expansion/contraction and osmotic pressure, the constraints for the conservation of the surface area of the membrane and the total volume enclosed are enforced as penalty terms. The new, unconstrained minimization problem (neglecting any electrostatic potential energy for now) with the penalty functions is given by

$$\min_{\vec{x}\in H^2(\Gamma)} I[\Gamma(\vec{x})], \tag{2.5}$$

$$I[\Gamma(\vec{x})] = E[\Gamma(\vec{x})] + \frac{k_S}{2}(S_A[\vec{x}]-\bar{S})^2 + \frac{k_V}{2}(V[\vec{x}]-\bar{V})^2, \tag{2.6}$$

where k_S and k_V are large constants to enforce the constraints to a chosen degree of precision. We use consistent values of these constants for tight constraints, but small enough for surface deformation to occur. (We use $\mathscr{K}_C = 1$ and $k_S = k_V = 100$.) In order for the energy to be minimized, changes to the surface must be made, which may change the surface area and volume slightly. Setting large but finite parameters for k_S and k_V allows for slight changes to the surface area and volume so that the energy can be minimized.

The minimization of (2.6) gives rise to the bending curvature equation of $\Gamma(\vec{x})$,

$$\delta_\Gamma I[\Gamma(\vec{x})] = \delta_\Gamma E[\Gamma(\vec{x})] + k_S(S_A[\vec{x}]-\bar{S})\delta S_A + k_V(V[\vec{x}]-\bar{V})\delta V. \tag{2.7}$$

The computation of the terms in (2.1), (2.6), and (2.7) depend on the choice of parameterization of the membrane surface. Since the bending energy of lipid bilayers is a function of the membrane curvature, the energy functional (2.6) is independent of the surface parameterization, [4]. Therefore, we choose to parameterize the surface not by brute force, intrinsic Cartesian coordinates $(\vec{x})$, but by a linear combination of surface harmonic functions, with $\vec{a}$ denoting the coefficients for this linear combination. Surface harmonic functions are linear combinations of real valued spherical harmonics. Since lipid vesicles are sphere-like structures, the choice of surface harmonics to represent the surface is natural, and this choice reduces the number of terms necessary for the computation.

3 Surface harmonics approximation

In this section, we simplify the problem given by (2.5) by utilizing the parameterization invariance. Rather than minimizing the energy (2.6) over membrane configurations in $H^2(\Gamma)$, we find an approximate solution by minimizing the total energy over membrane configurations determined by a linear combination of surface harmonic functions. That is, the exact membrane $\Gamma(\vec{x})$ is approximated by $\Gamma(\vec{a})$, where the real-valued coefficients $\vec{a}$ are chosen such that the bending energy $I[\Gamma]$ is minimized. The surface harmonics functions are an infinite dimensional basis; however, for the numerical implementation, we only use a finite number $(N+1)^2$ of surface harmonic functions. Stated mathematically,

https://doi.org/10.4208/cicp.OA-2015-0029 Published online by Cambridge University Press

the new minimization problem is

$$\min_{\vec{a}\in\mathbb{R}^{(N+1)^2}} I[\Gamma(\vec{a})], \tag{3.1}$$

$$I[\Gamma(\vec{a})] = E[\Gamma(\vec{a})] + \frac{k_S}{2}(S_A[\vec{a}]-\bar{S})^2 + \frac{k_V}{2}(V[\vec{a}]-\bar{V})^2. \tag{3.2}$$

Eq. (3.1) is the problem we solve in this paper. In this section, we introduce the surface harmonic parameterization. Using this parameterization, we define a mesh over which the minimization problem (3.1) is to be solved. Then, we provide formulas for the terms in (3.2) using the parameterization. Finally, a numerical method is given to solve the fully parameterized minimization problem.

3.1 Spherical harmonic functions

Spherical harmonics are solutions to Laplace's Equation in spherical coordinates. The solution can be obtained through separation of the variables θ and ϕ; however, a more convenient way to construct spherical harmonics is to use Associated Legendre Polynomials. The Associated Legendre Polynomials generalize Legendre Polynomials, provided $m\neq 0$, and are defined by

$$P_n^m(x) = (-1)^m(1-x^2)^{m/2}\frac{d^m}{dx^m}P_n(x), \qquad m\neq 0, \tag{3.3}$$

where $P_n(x)$ is the Legendre Polynomial. Using the Rodrigues representation, the Legendre Polynomials are given by

$$P_n(x) = \frac{1}{2^n n!}\frac{d^n}{dx^n}(x^2-1)^n. \tag{3.4}$$

Notice that in (3.3), if $m=0$, the Associated Legendre Polynomial is just the Legendre Polynomial. Combining Eqs. (3.3) and (3.4), we have

$$P_n^m(x) = (-1)^m(1-x^2)^{m/2}\frac{d^m}{dx^m}\left(\frac{1}{2^n n!}\frac{d^n}{dx^n}(x^2-1)^n\right), \qquad m\neq 0. \tag{3.5}$$

It is convenient to introduce the change of variables $\mu=\cos(\theta)$. In this way, the partial derivatives with respect to the polar angle $\theta\in[0,\pi]$ may be computed. Here, the physics ordering is used so that $0\leq\theta\leq\pi$ and $0\leq\phi\leq 2\pi$. Using this notation, normalized spherical harmonic functions are defined by

$$Y_n^m(\theta,\phi) = \sqrt{\left(\frac{(2n+1)(n-m)!}{4\pi(n+m)!}\right)}P_n^m(\mu)e^{im\phi}, \tag{3.6}$$

where $P_n^m(\mu)$ is the Associated Legendre Polynomial (3.5), evaluated at $\mu=\cos(\theta)$.

https://doi.org/10.4208/cicp.OA-2015-0029 Published online by Cambridge University Press

Since spherical harmonics form an orthonormal basis for $L^2(\mathbb{R}^2)$, linear combinations of them can represent smooth surfaces. The surface is parameterized in spherical coordinates (r,θ,ϕ), with $r=r(\theta,\phi)$. The radius at any angle set (θ,ϕ) is expressed in terms of spherical harmonics,

$$r(\vec{a};\theta,\phi)=\sum_{n=0}^{\infty}\sum_{m=-n}^{n}a_n^m Y_n^m(\theta,\phi). \tag{3.7}$$

The $a_n^m\in\vec{a}$ are the coefficients of the linear representation. These coefficients can be determined by the following formula:

$$a_n^m=\int_0^{2\pi}\int_0^{\pi}r(\theta,\phi)\overline{Y_n^m(\theta,\phi)}\sin(\theta)\,d\theta\,d\phi, \tag{3.8}$$

where $\overline{Y_n^m(\theta,\phi)}$ is the complex conjugate of $Y_n^m(\theta,\phi)$.

3.2 Surface harmonic functions

If the coefficients a_n^m are poorly chosen so that the complex parts of a_n^m and Y_n^m do not cancel, the radius (3.7) will be complex. In an optimization routine, the coefficients are perturbed arbitrarily, so any nonzero perturbation in the complex part will result in a complex surface. Since we seek a real-valued surface that minimizes the potential energy (2.6), we use only the real parts of the spherical harmonics to ensure that the surface under the energy minimization is real. These are surface harmonics. By Euler's formula, equation (3.6) can be rewritten as

$$Y_n^m(\theta,\phi)=f_n^m P_n^m(\mu)(\cos(m\phi)+i\sin(m\phi)), \tag{3.9}$$

where f_n^m is the normalization factor

$$f_n^m=\sqrt{\left(\frac{(2n+1)(n-m)!}{4\pi(n+m)!}\right)}. \tag{3.10}$$

Since the real and complex parts of (3.9) are also solutions to Laplace's equation, we define the surface harmonics accordingly,

$$S_n^m(\theta,\phi)=\begin{cases}f_n^m P_n^m(\mu)\cos(m\phi) & \text{if } m\geq 0,\\ f_n^{|m|}P_n^{|m|}(\mu)\sin(|m|\phi) & \text{if } m<0.\end{cases} \tag{3.11}$$

We now express the radius as a linear combination of the surface harmonics $S_n^m(\theta,\phi)$,

$$r(\vec{a};\theta,\phi)=\sum_{n=0}^{\infty}\sum_{m=-n}^{n}a_n^m S_n^m(\theta,\phi). \tag{3.12}$$

https://doi.org/10.4208/cicp.OA-2015-0029 Published online by Cambridge University Press

A more convenient representation of the radius of a surface that avoids the sign changes in m is

$$r(\vec{a};\theta,\phi)=\sum_{n=0}^{\infty}\sum_{m=0}^{n}\Big(A_n^m\cos(m\phi)+B_n^m\sin(m\phi)\Big)f_n^m P_n^m(\mu), \tag{3.13}$$

where A_n^m and B_n^m are defined via

$$a_n^m=\begin{cases}A_n^m & \text{if } m\geq 0,\\ B_n^m & \text{if } m<0.\end{cases} \tag{3.14}$$

3.3 Discretizing the surface

For the numerical implementation, we discretize the surface by a regular mesh over the angles $0\leq\theta\leq\pi$ and $0\leq\phi\leq 2\pi$. We use n_t equally spaced values of θ and n_p equally spaced values of ϕ for a total of $\mathcal{N}=n_t\cdot n_p$ grid points. Since $\mathcal{N}$ can be a very large number, performing pointwise calculations for $r(\theta,\phi)$ on the mesh is computationally costly. However, by using the global approximation method using the surface harmonics parameterization, the surface is approximated by truncating the infinite sum in (3.12) at some sufficiently large number N. This reduces the computation cost dramatically, since the number of surface harmonic functions required to accurately reconstruct biologically relevant surfaces is significantly less than $\mathcal{N}$. The coefficients $\vec{a}$ are ordered in the following way,

$$\begin{aligned}\vec{a}=[&A_0^0,A_1^0,A_1^1,B_1^1,A_2^0,A_2^1,A_2^2,B_2^1,B_2^2,\cdots,\\ &A_N^0,\cdots,A_N^N,B_N^1,\cdots,B_N^N]^T.\end{aligned} \tag{3.15}$$

We use the subscript $i=0,\cdots,(N+1)^2-1$ to denote the surface harmonic mode a_i. For each point on the mesh, θ_k, $k=1,\cdots,n_t$, and ϕ_l, $l=1,\cdots,n_p$, in the discretization, the radius r is determined by the coefficients $\vec{a}$,

$$r_{kl}(\vec{a};\theta_k,\phi_l)=\sum_{n=0}^{N}\sum_{m=0}^{n}\Big(A_n^m\cos(m\phi_l)+B_n^m\sin(m\phi_l)\Big)f_n^m P_n^m(\mu). \tag{3.16}$$

Using this parameterization, there are a total of $(N+1)^2$ coefficients to determine the surface parameterization. Our numerical results confirm that $(N+1)^2<<\mathcal{N}$.

3.4 Energy formulation in terms of surface harmonic parameterization

In this section, we complete the formulas for the terms in (3.2) using the surface harmonics approximation.

https://doi.org/10.4208/cicp.OA-2015-0029 Published online by Cambridge University Press

3.4.1 Constraints

The differential surface element is

$$dS[\vec{a}] = r[r_\phi^2 + r_\theta^2 \sin^2(\theta) + r^2 \sin^2(\theta)]^{1/2} d\theta d\phi. \tag{3.17}$$

The subscripts on r in the equation above represent partial derivatives and should not be confused with mesh positions θ_k and ϕ_l. The computations of the partial derivatives can be found in Appendix A. Each occurrence of $r = r(\vec{a},\theta,\phi)$ in the formula above is computed at each mesh point by (3.16). We note that the notation describing the dependence of r upon the surface harmonic coefficients is suppressed for readability. This will be done in the remainder of Section 3.4. The reader should understand that every time the variable r appears, it depends upon the surface harmonic coefficients, $r = r(\vec{a})$.

By (3.17), the surface area of the membrane $\Gamma(\vec{a})$ is

$$S_A[\vec{a}] = \int_0^{2\pi} \int_0^{\pi} r[r_\phi^2 + r_\theta^2 \sin^2(\theta) + r^2 \sin^2(\theta)]^{1/2} d\theta d\phi. \tag{3.18}$$

For simplicity in later calculations, we define the determinant of the covariant metric tensor to be

$$\omega[\vec{a}] = r[r_\phi^2 + r_\theta^2 \sin^2(\theta) + r^2 \sin^2(\theta)]^{1/2}, \tag{3.19}$$

so that $dS = \omega[\vec{a}]\, d\theta d\phi$, and

$$S_A[\vec{a}] = \int_0^{2\pi} \int_0^{\pi} \omega[\vec{a}]\, d\theta d\phi. \tag{3.20}$$

The volume enclosed by the membrane is

$$V[\vec{a}] = \frac{1}{3} \int_0^{2\pi} \int_0^{\pi} r^3 \sin(\theta)\, d\theta\, d\phi. \tag{3.21}$$

These two formulas (3.20) and (3.21) are used for the constraints in (3.2). The variation of (3.20) with respect to a single surface harmonic mode $a_i \in \vec{a}$ can be computed directly as

$$\delta S_A[\vec{a}] = \int_0^{2\pi} \int_0^{\pi} \frac{\delta \omega[\vec{a}]}{\delta a_i}\, d\theta\, d\phi. \tag{3.22}$$

Similarly, the variation of (3.21) can be computed directly,

$$\delta V[\vec{a}] = \int_0^{2\pi} \int_0^{\pi} r^2 \frac{\delta r}{\delta a_i} \sin(\theta)\, d\theta\, d\phi. \tag{3.23}$$

The variation of r with respect to a surface harmonic coefficient appearing in the equations above is given by

$$\frac{\delta r}{\delta a_i} = \begin{cases} \dfrac{\delta r}{\delta A_n^m} = f_n^m P_n^m(\mu)\cos(m\phi), & m \geq 0, \\ \dfrac{\delta r}{\delta B_n^m} = f_n^{|m|} P_n^{|m|}(\mu)\sin(|m|\phi), & m < 0. \end{cases} \tag{3.24}$$

https://doi.org/10.4208/cicp.OA-2015-0029 Published online by Cambridge University Press

We notice immediately that these variations match (3.11) exactly, and so the variations are just surface harmonic functions. For simplicity in the computation, for each the surface harmonic mode $a_i \in \vec{a}$, $i=0,1,2,\cdots(N+1)^2-1$, we define n and m from i to match (3.11) and (3.24) as follows:

$$n(i)=\lfloor\sqrt{i}\rfloor, \tag{3.25}$$

$$m(i)=\begin{cases} i-n^2 & \text{if } (i-n^2)\leq n, \\ n^2+n-i & \text{otherwise.} \end{cases} \tag{3.26}$$

Then, the variation in r with respect to the mode a_i is given by

$$\frac{\delta r}{\delta a_i}=S_n^m, \tag{3.27}$$

which through (3.25) and (3.26), is well defined for each i. Continuing with this simplification of subscripts,

$$\frac{\delta r_\phi}{\delta a_i}=-mS_n^{-m}, \tag{3.28}$$

$$\frac{\delta r_{\phi\phi}}{\delta a_i}=-m^2\frac{\delta S}{\delta a_i}. \tag{3.29}$$

The variations of r_θ and $r_{\theta\theta}$ are

$$\frac{\delta r_\theta}{\delta a_i}=\begin{cases} \dfrac{\delta r_\theta}{\delta A_n^m}=\cos(m\phi)f_n^m\partial_\theta P_n^m(\mu), & m\geq 0, \\ \dfrac{\delta r_\theta}{\delta B_n^m}=\sin(|m|\phi)f_n^{|m|}\partial_\theta P_n^{|m|}(\mu), & m<0, \end{cases} \tag{3.30}$$

$$\frac{\delta r_{\theta\theta}}{\delta a_i}=\begin{cases} \dfrac{\delta r_{\theta\theta}}{\delta A_n^m}=\cos(m\phi)f_n^m\partial_\theta^2 P_n^m(\mu), & m\geq 0, \\ \dfrac{\delta r_{\theta\theta}}{\delta B_n^m}=\sin(|m|\phi)f_n^{|m|}\partial_\theta^2 P_n^{|m|}(\mu), & m<0, \end{cases} \tag{3.31}$$

$$\frac{\delta r_{\theta\phi}}{\delta a_i}=\begin{cases} \dfrac{\delta r_{\theta\phi}}{\delta A_n^m}=\dfrac{\delta r_{\phi\theta}}{\delta A_n^m}=-m\sin(m\phi)f_n^m\partial_\theta P_n^m(\mu), & m\geq 0, \\ \dfrac{\delta r_{\theta\phi}}{\delta B_n^m}=\dfrac{\delta r_{\phi\theta}}{\delta B_n^m}=-m\cos(|m|\phi)f_n^{|m|}\partial_\theta P_n^{|m|}(\mu), & m<0. \end{cases} \tag{3.32}$$

https://doi.org/10.4208/cicp.OA-2015-0029 Published online by Cambridge University Press

Finally, the variation of ω is

$$\frac{\delta\omega}{\delta a_i}=\begin{cases}\dfrac{\delta\omega}{\delta A_n^m}=\dfrac{\delta r}{\delta A_n^m}[r_\phi^2+r_\theta^2\sin^2(\theta)+r^2\sin^2(\theta)]^{1/2} \\ \qquad\qquad +\dfrac{r}{2}\left(\dfrac{2r_\phi\dfrac{\delta r_\phi}{\delta A_n^m}+2r_\theta\dfrac{\delta r_\theta}{\delta A_n^m}\sin^2(\theta)+2r\dfrac{\delta r}{\delta A_n^m}\sin^2(\theta)}{\sqrt{r_\phi^2+r_\theta^2\sin^2(\theta)+r^2\sin^2(\theta)}}\right), \quad m\geq 0, \\ \dfrac{\delta\omega}{\delta B_n^m}=\dfrac{\delta r}{\delta B_n^m}[r_\phi^2+r_\theta^2\sin^2(\theta)+r^2\sin^2(\theta)]^{1/2} \\ \qquad\qquad +\dfrac{r}{2}\left(\dfrac{2r_\phi\dfrac{\delta r_\phi}{\delta B_n^m}+2r_\theta\dfrac{\delta r_\theta}{\delta B_n^m}\sin^2(\theta)+2r\dfrac{\delta r}{\delta B_n^m}\sin^2(\theta)}{\sqrt{r_\phi^2+r_\theta^2\sin^2(\theta)+r^2\sin^2(\theta)}}\right), \quad m<0.\end{cases} \tag{3.33}$$

3.4.2 Bending energy $E[\Gamma(\vec{a})]$

Now we move to the bending energy, given originally by (2.1). We need expressions for the mean and Gaussian curvatures H and K, respectively, and their variations, in terms of $\vec{a}$. We accomplish this by expressing the curvatures in terms of r, which depends upon the coefficients $\vec{a}$. It is convenient to first define the so-called "warping functions" E,F, and G, and shape operator functions L,M, and N. The warping functions are the coefficients of the first fundamental form and are given by

$$E=r_\theta^2+r^2, \tag{3.34}$$
$$F=r_\theta r_\phi, \tag{3.35}$$
$$G=r_\phi^2+r^2\sin^2(\theta). \tag{3.36}$$

Again, in each of the above formulas, the dependence upon $\vec{a}$ is via r, and the notation is suppressed for readability. The shape operator functions are the coefficients of the second fundamental form and are given by

$$L=-\vec{x}_\theta\cdot\hat{n}_\theta, \tag{3.37}$$
$$M=\frac{1}{2}\left(\vec{x}_\theta\cdot\hat{n}_\phi+\vec{x}_\phi\cdot\hat{n}_\theta\right), \tag{3.38}$$
$$N=-\vec{x}_\phi\cdot\hat{n}_\phi, \tag{3.39}$$

where $\vec{x}$ is a surface vector in spherical coordinates, $\vec{x}=[r\sin(\theta)\cos(\phi),r\sin(\theta)\sin(\phi),r\cos(\theta)]$, and $\hat{n}$ is the unit normal to the surface at $\vec{x}$,

$$\hat{n}=\frac{\vec{x}_\theta\times\vec{x}_\phi}{\left|\vec{x}_\theta\times\vec{x}_\phi\right|}. \tag{3.40}$$

https://doi.org/10.4208/cicp.OA-2015-0029 Published online by Cambridge University Press

Define

$$R=\left|\vec{x}_\theta\times\vec{x}_\phi\right| \tag{3.41}$$

for notational convenience. The shape operator functions can be expressed in terms of r and its derivatives.

$$L=R^{-1}(-2rr_\theta^2\sin(\theta)+r^2r_{\theta\theta}\sin(\theta)-r^3\sin(\theta)), \tag{3.42}$$

$$M=R^{-1}(2rr_\phi r_\theta\sin(\theta)-r^2r_{\theta\phi}\sin(\theta)+r^2r_\phi\cos(\theta)), \tag{3.43}$$

$$N=R^{-1}(-r^3\sin^3(\theta)+r^2r_{\phi\phi}\sin(\theta)+r^2r_\theta\cos(\theta)\sin^2(\theta)-2rr_\phi^2\sin(\theta)). \tag{3.44}$$

Finally, we obtain the local mean curvature and the Gaussian curvature,

$$H(\vec{a};\theta,\phi)=\frac{EN+GL-2FM}{2(EG-F^2)}, \tag{3.45}$$

$$K(\vec{a};\theta,\phi)=\frac{LN-M^2}{EG-F^2}. \tag{3.46}$$

In terms of r,

$$\begin{aligned}H(\vec{a};\theta,\phi)=\frac{1}{2}R^{-1}\Big[&3r_\theta^2r^2\sin^3(\theta)-r_\theta^2rr_{\phi\phi}\sin(\theta)-r_\theta^3r\cos(\theta)\sin^2(\theta)\\&+8r_\theta^2r_\phi^2\sin(\theta)+2r^4\sin^3(\theta)-r^3r_{\phi\phi}\sin(\theta)-r^3r_\theta\cos(\theta)\sin^2(\theta)\\&+3r^2r_\phi^2\sin(\theta)-r_\phi^2rr_{\theta\theta}\sin(\theta)-r^3r_{\theta\theta}\sin^3(\theta)-2r_\phi r_\theta rr_{\theta\phi}\sin(\theta)\\&+2r_\phi^2r_\theta r\cos(\theta)\Big]/\Big[-r^3\sin^2(\theta)-rr_\theta^2\sin^2(\theta)-rr_\phi^2\Big].\end{aligned} \tag{3.47}$$

The variation of the mean curvature with respect to a surface harmonic mode a_i is straightforward, but messy, so it is omitted.

Finally, we compute the variation of $E[\Gamma(\vec{a})]$ with respect to each coefficient $a_i\in\vec{a}$. To do this, we assume that the Gaussian modulus $\mathcal{K}_G$ is uniform over the membrane surface, and so the Gaussian curvature integrates to a constant $\int_\Gamma KdS=4\pi(1-g)$, where g is the genus of the membrane topology, according to the Gauss-Bonet Theorem, [32]. Thus, the variation of the bending energy with respect to a surface harmonic mode a_i is

$$\begin{aligned}\delta_\Gamma E[\Gamma(\vec{a});\theta,\phi]=\int_\Gamma\Big[&\mathcal{K}_C2(2H-C_0)(\delta H)\omega[\vec{a}]\\&+\mathcal{K}_C\frac{1}{2}(2H-C_0)^2(\delta\omega[\vec{a}])\Big]d\theta d\phi.\end{aligned} \tag{3.48}$$

With the surface Γ expressed in terms of the surface harmonic coefficients $\vec{a}$, the bending curvature equation is

$$\delta_\Gamma I[\Gamma(\vec{a})]=\delta_\Gamma E[\Gamma(\vec{a})]+k_S(S_A[\vec{a}]-\bar{S})(\delta S_A[\vec{a}])+k_V(V[\vec{a}]-\bar{V})(\delta V[\vec{a}]). \tag{3.49}$$

The energy (3.2) and its variation with respect to spherical harmonic coefficients (3.49) is now complete.

https://doi.org/10.4208/cicp.OA-2015-0029 Published online by Cambridge University Press

3.5 Numerical methods

We employ a Fletcher-Reeves type nonlinear conjugate gradient (NCG) method to minimize the total energy functional (3.2). For the parameter β, we chose the Hestenes-Stiefel formula. For a description of the method, please refer to [25], or, the fully detailed pseudocode is provided in the Appendix B.

In this implementation, the surface harmonic expansion (3.16) is truncated at some value N, giving a total of $(N+1)^2$ surface harmonic modes. For many vesicle structures, increasing N achieves higher accuracy; however, it also increases computation time. We seek an N large enough for accuracy, yet small enough for computational efficiency. To determine an appropriate choice for N, we reconstructed three surfaces which have analytical formulas, giving a way to compute the energy analytically. We examined the root mean square error (L^2 norm) in the surface reconstruction pointwise, and the relative error in the volume, surface area, and energy. For all structures, the accuracy is perfect as $N\to\infty$; however, the convergence is not monotone. For some vesicle structures, increasing N transiently actually gives a worse approximation to the bending energy.

The first surface we reconstructed was an energy-minimizing axisymmetric prolate from [29]. Instructions for reconstructing this surface can be found in Appendix B of [29], with choice of parameters $\bar{P}=0.1$, $\bar{\Sigma}=-1.1\bar{P}^{2/3}$, $C_0=0$, and $U(0)=0.56$. Next, we reconstructed two statistically fitted parameterizations of a red blood cell from [12]. The height of the profile of the surface is given by

$$h(x)=\frac{\pm 0.5}{R_0}\sqrt{1-x^2}(C_0+C_2x^2+C_4x^4),\qquad x\in[-1,1]. \tag{3.50}$$

Table 4 in [12] includes values for R_0,C_0,C_2, and C_4 for producing red blood cell shapes with tonicities 300 and 217 mO. The values are reproduced in Table 1. We chose two linear combinations of the parameters given for averaged shapes from the ones in [12]. The profiles for the three sample surfaces and their reconstructed surfaces with $N=4$ are included in Fig. 1.

Table 1: Shape coefficients for average RBC.

Tonicity (mO)	R_0 (μm)	C_0 (μm)	C_2 (μm)	C_4 (μm)
300	3.91	0.81	7.83	-4.39
217	3.80	2.10	7.58	-5.59

The surface harmonic parameterizations of these three surfaces are determined by

$$a_n^m=\int_0^{2\pi}\int_0^{\pi} r(\theta,\phi)S_n^m(\theta,\phi)\,d\theta\,d\phi. \tag{3.51}$$

For the reconstruction, the integration was computed numerically over 230 cubature points $\vec{\theta}_i$, $i=1,\cdots,230$. Using the coefficients from (3.51), the reconstructed radius $\tilde{r}$ was

https://doi.org/10.4208/cicp.OA-2015-0029 Published online by Cambridge University Press

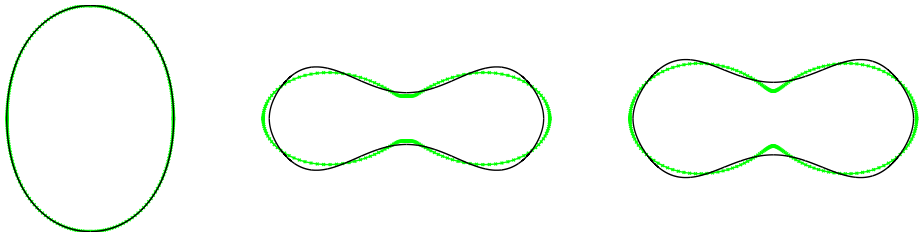


Figure 1: Profile of various test surfaces (black) and their reconstructions using surface harmonics with $N=4$ (green). The full surfaces are generated by rotation about the y-axis. Left: prolate from [29], middle and right: red blood cell from [12] with 50% and 90% weight on tonicity 217 mO coefficients, respectively.

determined by (3.16). The root mean square distance error in the reconstruction is defined over the cubature points by

$$e_{rms}=\sum_{i=1}^{230}\left(r(\vec{\theta}_i)-\tilde{r}(\vec{\theta}_i)\right)^2/230.$$

The pointwise error and the relative error in the volume, surface area, and energy are provided in Tables 2-4 for various N.

Table 2: Error from truncation in surface harmonic expansion for prolate sample surface.

N	e_{rms}	e_{Vol}	e_{SA}	e_{Eng}
1	2.06×10^{-1}	2.57×10^{-2}	2.22×10^{-2}	3.73×10^{-2}
2	8.59×10^{-4}	2.22×10^{-4}	1.08×10^{-2}	8.37×10^{-3}
3	8.59×10^{-4}	2.22×10^{-4}	1.08×10^{-2}	8.37×10^{-3}
4	5.02×10^{-7}	2.56×10^{-6}	1.11×10^{-2}	7.19×10^{-3}
5	5.06×10^{-7}	2.56×10^{-6}	1.11×10^{-2}	7.19×10^{-3}
6	1.46×10^{-7}	2.76×10^{-6}	1.11×10^{-2}	7.18×10^{-3}
7	1.53×10^{-7}	2.76×10^{-6}	1.11×10^{-2}	7.18×10^{-3}
8	3.35×10^{-8}	2.76×10^{-6}	1.11×10^{-2}	7.18×10^{-3}

As you can see in Table 2, the accuracy in the reconstruction of the prolate surface increased in all categories as N increased. The most relevant observation to this work is that the error in the energy is less than 10^{-2} (1%) using $N=2$ and greater. For a simple prolate structure, only $(2+1)^2=9$ modes are necessary.

For the red blood cell structures in Tables 3-4, initially the errors decreased as N increased, but increasing the number of modes beyond a certain threshold actually increased the error in the energy computation. For RBC 1, the best possible error in the energy is 5.3%, with $N=4$ or $N=5$ modes. For RBC 2, the best error is 7.75% with the same N. We suggest the reason for this is because higher order modes contain more bulges than the lower order modes, particularly near the two poles. In the reconstruction, the coefficients are chosen to minimize e_{rms}. While transiently increasing N does

https://doi.org/10.4208/cicp.OA-2015-0029 Published online by Cambridge University Press

Table 3: Error from truncation in surface harmonic expansion for RBC 1 (50% weight) sample surface.

N	e_{rms}	e_{Vol}	e_{SA}	e_{Eng}
1	9.80×10^{-2}	4.00×10^{-1}	4.20×10^{-1}	4.21×10^{-1}
2	6.65×10^{-3}	4.80×10^{-2}	4.99×10^{-2}	8.36×10^{-1}
3	6.65×10^{-3}	4.80×10^{-2}	4.99×10^{-2}	8.36×10^{-1}
4	2.29×10^{-3}	1.93×10^{-3}	1.72×10^{-2}	5.34×10^{-2}
5	2.29×10^{-3}	1.93×10^{-3}	1.72×10^{-2}	5.34×10^{-2}
6	1.39×10^{-3}	4.94×10^{-3}	8.35×10^{-3}	1.62×10^{0}
7	1.39×10^{-3}	4.94×10^{-3}	8.35×10^{-3}	1.62×10^{0}
8	4.13×10^{-4}	3.29×10^{-5}	1.44×10^{-2}	7.97×10^{-1}

Table 4: Error from truncation in surface harmonic expansion for RBC 2 (90% weight) sample surface.

N	e_{rms}	e_{Vol}	e_{SA}	e_{Eng}
1	7.13×10^{-2}	2.82×10^{-1}	3.12×10^{-1}	3.82×10^{-1}
2	2.06×10^{-3}	1.35×10^{-2}	9.60×10^{-3}	3.96×10^{-1}
3	2.06×10^{-3}	1.35×10^{-2}	9.60×10^{-3}	3.96×10^{-1}
4	1.26×10^{-3}	3.36×10^{-3}	5.71×10^{-3}	7.75×10^{-2}
5	1.26×10^{-3}	3.36×10^{-3}	5.71×10^{-3}	7.75×10^{-2}
6	3.07×10^{-4}	1.92×10^{-3}	5.79×10^{-3}	4.24×10^{-1}
7	3.07×10^{-4}	1.92×10^{-3}	5.79×10^{-3}	4.24×10^{-1}
8	2.15×10^{-4}	1.21×10^{-3}	7.14×10^{-3}	9.97×10^{-2}

improve the accuracy of e_{rms}, it may introduce local oscillations. These oscillations may unfavorably change the surface area, volume, and energy with increasing N. In particular, the energy is a function of the curvature, so numerous but small oscillations have a high energy cost. As an example, the left chart in Fig. 2 shows RBC 1 reconstructed with $N=4$ and $N=12$. Notice that although $N=12$ provided a better approximation to the surface pointwise, the local oscillations near the middle of the RBC unfavorably added to the energy. While these oscillations can be eliminated eventually by increasing N, as illustrated in the right chart of Fig. 2, the increase of the computational cost is significant. The modes of surface harmonic increases quadratically with N. For $N=20$ and $N=24$, the modes of surface harmonic functions amounts to 441 and 625, respectively. Furthermore, more quadrature points are necessary for the accurate quadrature of high mode surface harmonic functions, for example, a total of 3600 base points are used for the Gaussian quadrature in [20]. While the quadrature could be accelerated using various techniques [27, 30], the method becomes less competitive in the regard of efficiency for large N. The comparisons here indicate that it is possible to use a small N to achieve combined accuracy and efficiency, which will be further demonstrated in the simulations presented in Section 4.

https://doi.org/10.4208/cicp.OA-2015-0029 Published online by Cambridge University Press

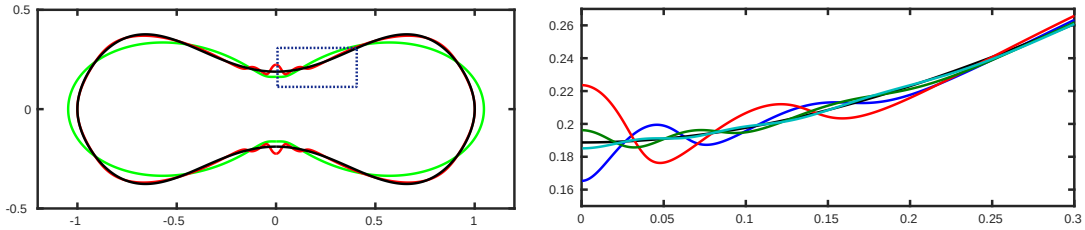


Figure 2: Effect of large N on RBC 1. Left: The analytical surface profile is in black, the reconstructed surfaces for $N=4$ and $N=12$ are in green and red, respectively. Increasing N near 4 improves the pointwise accuracy especially near the edges, but also introduces oscillations near the center with high energy cost. For $N=12$, the pointwise error in the surface is improved at 1.18×10^{-4}, but the error in the energy is 258%. Right: A close look of the dashline bounded region in left chart for $N=12$(red),16(blue), 20 (green) and 24 (cyan) on the analytical profile (black). Oscillations near the pole can be eliminated by using large N at the cost of computational complexity.

4 Examples: reduced volume

In this section, we provide numerical examples to test the robustness method. First, observe that the integration of the square local mean curvature H^2 is a dimensionless quantity. When the spontaneous curvature $C_0=0$, the mechanical bending energy (2.1) is completely governed by this dimensionless quantity and is therefore scale-invariant. Thus, for vesicle shapes, where $C_0=0$, the minimum energy is completely determined by a single dimensionless quantity called the reduced volume, denoted v. The reduced volume scales the current volume V by the volume of a sphere with surface area S_A. Since spheres maximize volume for a given surface area, the reduced volume satisfies $v\leq1$. The reduced volume is given by the formula

$$v=\frac{V}{4\pi/3R_0^3}, \tag{4.1}$$

where $R_0=\sqrt{S_A/4\pi}$. In terms of the surface area, the reduced volume is

$$v=\frac{6\sqrt{\pi}V}{(S_A)^{3/2}}. \tag{4.2}$$

Seifert et al. has compiled a library of reduced volumes and their corresponding minimum energies for axisymmetric shapes by solving the Euler-Lagrange equations using a parameterization of the vesicle shape with an axis of symmetry [29]. For verification purposes, we compare our axisymmetric results for various reduced volumes to theirs. We set the constraint volume $\bar{V}$ to be proportional to V by the (projected) reduced volume v. The volume constraint is in violation and NCG begins to change the shape to relax this configuration. If we begin with a perfectly spherical vesicle, NCG will simply scale the sphere to a sphere with a smaller volume, and the final reduced volume will be 1. Therefore, we take a slightly perturbed sphere to be our initial configuration. After NCG

https://doi.org/10.4208/cicp.OA-2015-0029 Published online by Cambridge University Press

has converged, we calculate the final reduced volume v and the final energy E scaled by the energy of a spherical vesicle $E_0 = 8\pi\mathcal{K}_C$.

From the reconstruction examples, $N = 4$ is a reasonable truncation for the surface harmonic expansion. During the iterations of NCG, 20 quadrature points are used in each dimension (for a total of 400 points). We have three criteria for convergence. NCG converges if (1) the L^2 norm of the change in gradient is less than $\epsilon_g = 10^{-10}$, (2) the L^2 norm of the change in the modes is less than $\epsilon_a = 10^{-6}$, or (3) the relative change in the energy is less than $\epsilon_E = 10^{-4}$, provided a minimum number of iterations (15) is reached. When the final configuration is achieved, the total energy is evaluated with 64 quadrature points in each dimension to provide a more accurate computation and to ensure that enough quadrature points are used. For reduced volumes above approximately $v = 0.75$, the numerical energy is within 10% of the analytical values calculated by Seifert et al. (see Table 5). However, for reduced volumes less than this, the error sometimes exceeds 10%. If the number of modes is increased to $N = 6$ (since $N = 5$ gives the same numerical results as $N = 4$ as demonstrated in Tables 2-4), the relative error is reduced. However, there is a significant difference between the energy evaluated with 20 quadrature points than with 64 quadrature points at the final iteration when using $N = 6$. This is because the added oscillation from the higher order modes is not absorbed by NCG with only 20 quadrature points. Using 30 quadrature points in each dimension when $N = 6$, the relative error is less than 1% when compared to 64 points in each dimension. For surfaces with reduced volume $0.65 \leq v \leq 0.75$, using $N = 6$ and 30 quadrature points in each dimension, the relative error in the final energy is less than 10%. For surfaces with reduced volume $0.5 \leq v < 0.65$, we determined that the use of $N = 8$ fared no better than $N = 6$, as both gave errors greater than 10%. Our method could not reconstruct surfaces with these reduced volumes in a quantitatively accurate fashion. Sample surfaces with various reduced volumes using different N are plotted in Fig. 3, overlayed by the analytic solution from Seifert [29].

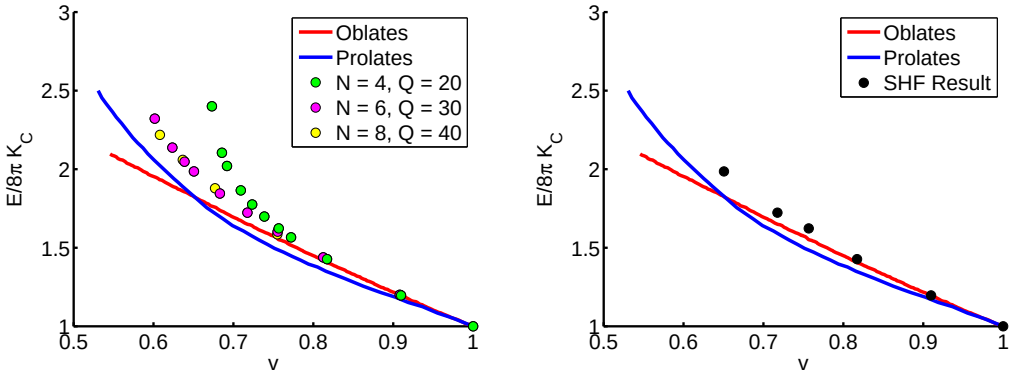


Figure 3: Vesicle energy vs. reduced volume for comparison with [29]. Left: All data points used to determine cutoff values for N given a reduced volume v. Right: Kept data according to the cutoff values. These data points are within 10% error of the appropriate analytical curve.

https://doi.org/10.4208/cicp.OA-2015-0029 Published online by Cambridge University Press

Table 5: Vesicle energy vs. reduced volume for comparison with [14].

v	1.0	0.91	0.82	0.75	0.72	0.65
E/E_0, [29]	1.0	1.19	1.43	1.62	1.72	1.98
E/E_0, SHF	1.0	1.19	1.41	1.56	1.65	1.83
Rel. error	0.00%	0.14%	1.60%	4.24%	4.41%	8.79%
Iterations	1	48	30	40	46	64

In summary, for surfaces with $0.75 \leq v \leq 1$, we used $N=4$ and 20 quadrature points in each dimension, for surfaces with $0.65 \leq v < 0.75$, we used $N=6$ and 30 quadrature points in each dimension. The results using this cutoff are overlayed by Seifert's data in Fig. 3. With these choices, NCG converges in an average of 45 steps, a number comparable to the degrees of freedom involved. The surfaces corresponding to the data in Table 5 are shown in Fig. 4.

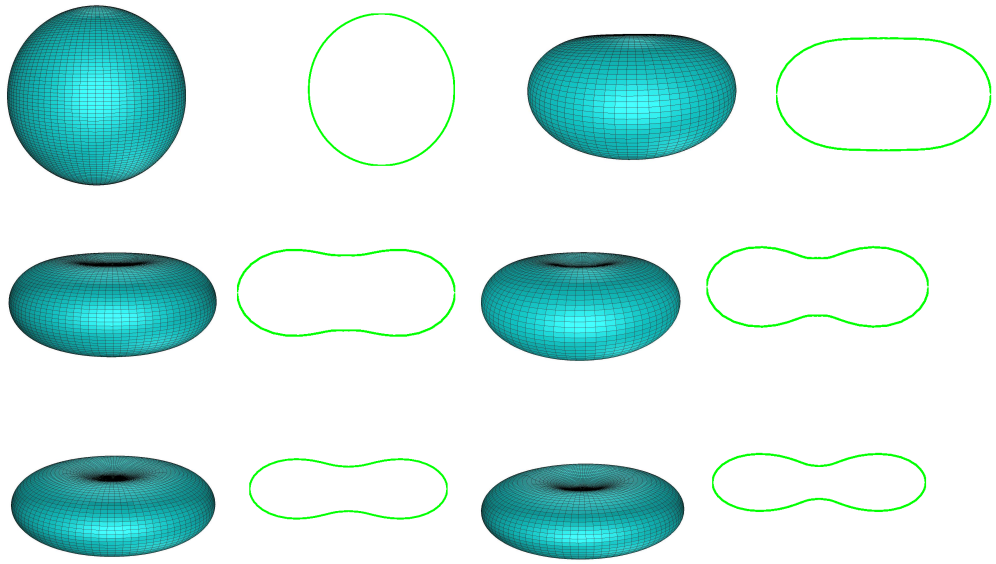


Figure 4: Axisymmetric results with various reduced volumes. Organized by row, $v =$ 1.00,0.91;0.82,0.75;0.72,0.65. The shapes here correspond to the data in Table 5. The full volumes are angled slightly down, while the profiles are not angled.

We note that the results of the numerical procedure may be only local minima and therefore only locally stable. With enough perturbation through some external force, another configuration with a lower energy may be achieved. In the range of $0.64 \leq v \leq 1$, oblate shapes are local energy minimizers, but prolates are global minimizers for axisymmetric shapes. However, it may be possible to obtain a non-axisymmetric shape with lower energy than a prolate.

https://doi.org/10.4208/cicp.OA-2015-0029 Published online by Cambridge University Press

5 Conclusion

We have presented a fast algorithm for computing axisymmetric and non-axisymmetric spherical vesicles that correspond to minimized Canham-Helfrich-Evans curvature energy subject to surface area and volume constraints. Our method is based on the real-valued surface harmonic rather than the complex-valued spherical harmonic expansion of the surface configurations. In this way, our method is advantageous to the spherical harmonic approximations of the individual coordinates of the vesicle configurations, since it excludes the complex parts unnecessary for computing real-valued surfaces [20]. Computational simulations showed that vesicles of various reduced volumes can be approximated using up to 49 surface harmonic functions, and the approximation error measured in curvature energy can be well maintained within 9%, mostly below 5% indeed. We use a nonlinear conjugate gradient method rather than the Newton's method for the numerical minimization. This makes it possible for us to avoid the computation of the Hessian matrices and the solution of linear systems to further improve the efficiency of the calculation. The nonlinear conjugate gradient algorithm converges in an average of 45 steps, which is comparable to the number of degrees of freedom used, demonstrating the efficiency of the algorithm. Further improvement of the computational efficiency is possible through the implementation of fast quadrature of spherical harmonics [19, 21]. We will explore the implementation of this fast algorithm in the external force fields such as fluid flow or electrostatic potential field, for which re-orientation of the vesicle configurations might be necessary because of the vesicle rotation caused by the non-vanishing torque applied by the external force. A fast algorithm for the vesicle deformation induced by the electrostatic force is currently under development and will be reported elsewhere.

Acknowledgments

The authors thank the reviewers for their insightful suggestions which led to an improved manuscript.

This work has been partially supported by the Simons Foundation and National Institutes of Health through the grant R01GM11759301 as part of the joint DMS/NIGMS initiative to support research at the interface of the biological and mathematical sciences.

Appendix

A Computations of the partial derivatives of r

To complete the formulas in Section 3.4, the derivatives of r with respect to θ and ϕ are computed. The derivatives with respect to ϕ are straightforward

https://doi.org/10.4208/cicp.OA-2015-0029 Published online by Cambridge University Press

$$r_\phi = \sum_{n=0}^{N}\sum_{m=0}^{n}\Big(-mA_n^m\sin(m\phi)+mB_n^m\cos(m\phi)\Big)f_n^m P_n^m(\mu)$$
$$= \sum_{n=0}^{N}\sum_{m=-n}^{n} -ma_n^m S_n^{-m}, \tag{A.1}$$
$$r_{\phi\phi} = \sum_{n=0}^{N}\sum_{m=0}^{n}\Big(-m^2A_n^m\cos(m\phi)+-m^2B_n^m\sin(m\phi)\Big)f_n^m P_n^m(\mu)$$
$$= -m^2 r. \tag{A.2}$$

The derivatives with respect to θ are

$$r_\theta = \sum_{n=0}^{N}\sum_{m=0}^{n}\Big(A_n^m\cos(m\phi)+B_n^m\sin(m\phi)\Big)f_n^m\partial_\theta P_n^m(\mu), \tag{A.3}$$
$$r_{\theta\theta} = \sum_{n=0}^{N}\sum_{m=0}^{n}\Big(A_n^m\cos(m\phi)+B_n^m\sin(m\phi)\Big)f_n^m\partial_\theta^2 P_n^m(\mu), \tag{A.4}$$
$$r_{\phi\theta} = r_{\theta\phi} = \sum_{n=0}^{N}\sum_{m=0}^{n}\Big(-mA_n^m\sin(m\phi)+mB_n^m\cos(m\phi)\Big)f_n^m\partial_\theta P_n^m(\mu)$$
$$= \sum_{n=0}^{N}\sum_{m=-n}^{n} -ma_n^m\frac{\partial S_n^{-m}}{\partial\theta}, \tag{A.5}$$

where $\partial_\theta P_n^m(\mu)$ is given by the recurrence relation for the derivative of the associated Legendre polynomial $P_n^m(\mu)$,

$$\partial_\theta P_n^m(\mu) = \frac{-1}{\sin(\theta)}\left((n+1)\cos(\theta)P_n^m(\mu)-(n-m+1)P_{n+1}^m(\mu)\right), \tag{A.6}$$

and $\partial_\theta^2 P_n^m(\mu)$ is computed directly from (A.6) as

$$\partial_\theta^2 P_n^m(\mu) = \Big((n+1+(n+1)^2\cos^2\theta)P_n^m(\mu)-2\cos\theta(n-m+1)(n+2)P_{n+1}^m(\mu)$$
$$+(n-m+1)(n-m+2)P_{n+2}^m(\mu)\Big)\frac{1}{\sin^2\theta}. \tag{A.7}$$

B Nonlinear Conjugate Gradient Method and Linear Search

The pseudocode of the nonlinear conjugate method and the related linear search method mentioned in Section 3.5 is given below to ease the implementation. The functions $I(\cdot)$ and $\delta_\Gamma I(\cdot)$ represent the curvature energy and its gradient, c.f. Eqs. (3.2) and (3.49), respectively.

https://doi.org/10.4208/cicp.OA-2015-0029 Published online by Cambridge University Press

Algorithm 1 Nonlinear conjugate gradient (NCG)

Define initial SHF modes $\vec{a}_0$
Define initial tolerances ϵ_g,ϵ_a, ϵ_E, M
Compute $E_0 \leftarrow I(\vec{a}_0)$, $\vec{g}_0 \leftarrow \delta_\Gamma I(\vec{a}_0)$
Direction $\vec{d}_0 \leftarrow -\vec{g}_0$
$k \leftarrow 0$
for k from 1 to M **do**:
    Step size $\alpha_k \leftarrow LineSearch(\vec{a}_k,\vec{d}_k)$
    Step in direction $\vec{a}_{k+1} \leftarrow \vec{a}_k + \alpha_k \vec{d}_k$
    Update energy and gradient $E_{k+1} \leftarrow I(\vec{a}_{k+1})$, $\vec{g}_{k+1} \leftarrow \delta_\Gamma I(\vec{a}_{k+1})$
    if $||\vec{g}_{k+1} - \vec{g}_k|| < \epsilon_g$ **then**:
        break
    end if
    Compute $\beta_k \leftarrow (\vec{g}_{k+1}^T(\vec{g}_{k+1} - \vec{g}_k)) / ((\vec{g}_{k+1} - \vec{g}_k)^T \vec{d}_k)$
    Update direction $\vec{d}_{k+1} \leftarrow -\vec{g}_{k+1} + \beta_k \vec{d}_k$
    if $||\vec{a}_{k+1} - \vec{a}_k|| < \epsilon_a$ **then**:
        break
    end if
    if $|E_{k+1} - E_k| / E_{k+1} < \epsilon_E$ **and** $k > 14$ **then**
        break
    end if
    $k \leftarrow k+1$
end for
return $\vec{a}_k$

References

[1] Prosenjit Bagchi. Mesoscale simulation of blood flow in small vessels. *Biophys. J.*, 92(6):1858–1877, 2007.

[2] Amir Houshang Bahrami and Mir Abbas Jalali. Vesicle deformations by clusters of transmembrane proteins. *J. Chem. Phys.*, 134:085106, 2011.

[3] P.B. Canham. The minimum energy of bending as a possible explanation of the biconcave shape of the human red blood cell. *J. Theor. Biol.*, 26(1):61–81, 1970.

[4] R. Capovilla, J. Guven, and J. A. Santiago. Deformations of the geometry of lipid vesicles. *J. Phys. A – Math. Gen.*, 36(23):6281, 2003.

[5] Fredric S. Cohen, Robert Eisenberg, and Rolf J. Ryham. A dynamic model of open vesicles in fluids. *Commun. Math. Sci.*, 10:1273–1285, 2012.

[6] Sovan Das and Qiang Du. Adhesion of vesicles to curved substrates. *Phys. Rev. E*, 77:011907, Jan 2008.

[7] Qiang Du, Chun Liu, Rolf Ryham, and Xiaoqiang Wang. A phase field formulation of the Willmore problem. *Nonlinearity*, 18:1249, 2005.

[8] Qiang Du, Chun Liu, and Xiaoqiang Wang. A phase field approach in the numerical study

https://doi.org/10.4208/cicp.OA-2015-0029 Published online by Cambridge University Press

Algorithm 2 Line Search

Define $\alpha_m \leftarrow 0, \alpha_M \leftarrow 0.15, M, \epsilon$
Compute $E_m \leftarrow I(\vec{a}+\alpha_m \cdot \vec{d})$, $E_M \leftarrow I(\vec{a}+\alpha_M \cdot \vec{d})$
if $E_m < E_M$ **then**:
    $\alpha_l \leftarrow \alpha_m$; $\alpha_u \leftarrow \alpha_M$
    $E_l \leftarrow E_m$; $E_u \leftarrow E_M$
else
    $\alpha_l \leftarrow \alpha_M$; $\alpha_u \leftarrow \alpha_m$
    $E_l \leftarrow E_M$; $E_u \leftarrow E_m$
end if
for i from 1 to M **do**:
    $\alpha_t \leftarrow (\alpha_l + \alpha_u)/2$
    $E_t \leftarrow I(\vec{a}+\alpha_t \cdot \vec{d})$
    if $E_t > E_l$ **then**:
        $\alpha_u \leftarrow \alpha_t$
    else
        $\vec{g} \leftarrow \delta_\Gamma I(\vec{a}+\alpha_t \cdot \vec{d})$
        $(D\phi) \leftarrow \vec{g}^T \vec{d}$
        if $(D\phi) \cdot (\alpha_l - \alpha_t) > 0$ **then**:
            $\alpha_l \leftarrow \alpha_t$
        else
            $\alpha_u \leftarrow \alpha_l$; $\alpha_l \leftarrow \alpha_t$
        end if
        $E_l \leftarrow E_t$
    end if
    if $|\alpha_u - \alpha_l| < \epsilon$ **then**
        break
    end if
end for
return α_t

of the elastic bending energy for vesicle membranes. *J. Comput. Phys.*, 198(2):450–468, 2004.

[9] Qiang Du, Chun Liu, and Xiaoqiang Wang. Simulating the deformation of vesicle membranes under elastic bending energy in three dimensions. *J. Comput. Phys.*, 212(2):757–777, 2006.

[10] C. D. Eggleton and A. S. Popel. Large deformation of red blood cell ghosts in a simple shear flow. *Phys. Fluids*, 10(8):1834–1845, 1998.

[11] E.A. Evans. Bending resistance and chemically induced moments in membrane bilayers. *Biophys. J.*, 14(12):923–931, 1974.

[12] Evan Evans and Yuan-Cheng Fung. Improved measurements of the erythrocyte geometry. *Microvasc. Res.*, 4(4):335–347, 1972.

[13] Khashayar Farsad and Pietro De Camilli. Mechanisms of membrane deformation. *Curr.*

https://doi.org/10.4208/cicp.OA-2015-0029 Published online by Cambridge University Press

Opin. Cell Biol., 15(4):372–381, 2003.
[14] Feng Feng and William S. Klug. Finite element modeling of lipid bilayer membranes. *J. Comput. Phys.*, 220(1):394–408, 2006.
[15] Xin Feng, Kelin Xia, Yiying Tong, and Guo-Wei Wei. Geometric modeling of subcellular structures, organelles, and multiprotein complexes. *Internat. J. Numer. Methods Eng.*, 28:1198–1223, 2012.
[16] Volkmar Heinrich, Bojan Bozic, Sasa Svetina, and Bostjan Zeks. Vesicle deformation by an axial load: From elongated shapes to tethered vesicles. *Biophys. J.*, 76:2056–2071, 1999.
[17] W. Helfrich et al. Elastic properties of lipid bilayers: theory and possible experiments. *Z. Naturforsch. C*, 28(11):693–703, 1973.
[18] Roger D. Kamm. cellular fluid mechanics. *Annu. Rev. Fluid Mech.*, 34(1):211–232, 2002.
[19] Jens Keiner and Daniel Potts. Fast evaluation of quadrature formulae on the sphere. *Math. Comp.*, 77:397–419, 2008.
[20] Khaled Khairy and Jonathon Howard. Minimum-energy vesicle and cell shapes calculated using spherical harmonics parameterization. *Soft Matter*, 7:2138–2143, 2011.
[21] Stefan Kunis and Daniel Potts. Fast spherical Fourier algorithms. *J. Comput. Appl. Math.*, 161(1):75–98, 2003.
[22] Shuwang Li, John Lowengrub, and Axel Voigt. Locomotion, wrinkling, and budding of a multicomponent vesicle in viscous fluids. *Commun. Math. Sci.*, 10:645–670, 2012.
[23] L. Ma and W. S. Klug. Viscous regularization and r-adaptive remeshing for finite element analysis of lipid membrane mechanics. *J. Comput. Phys.*, 227(11):5816–5835, 2008.
[24] M. Mikucki and Y. Zhou. Electrostatic forces on charged surfaces of bilayer lipid membranes. *SIAM J. Appl. Math.*, 74(1):1–21, 2014.
[25] Jorge Nocedal and Stephen J. Wright. *Numerical optimization*. Springer series in operations research and financial engineering. Springer, New York, NY, 2. ed. edition, 2006.
[26] Thomas R. Powers. Mechanics of lipid bilayer membranes. In Sidney Yip, editor, *Handbook of Materials Modeling*, pages 2631–2643. Springer Netherlands, 2005.
[27] Vladimir Rokhlin and Mark Tygert. Fast Algorithms for Spherical Harmonic Expansions. *SIAM J. Sci. Comput.*, 27(6):1903–28, 2006.
[28] Udo Seifert. Configurations of fluid membranes and vesicles. *Adv. Phys.*, 46(1):13–137, 1997.
[29] Udo Seifert, Karin Berndl, and Reinhard Lipowsky. Shape transformations of vesicles: Phase diagram for spontaneous-curvature and bilayer-coupling models. *Phys. Rev. A*, 44:1182–1202, Jul 1991.
[30] Avram Sidi. Application of class S_m variable transformations to numerical integration over surfaces of spheres. *J. Comput. Appl. Math.*, 184(2):475–492, 2005.
[31] Jin Sun Sohn, Yu-Hau Tseng, Shuwang Li, Axel Voigt, and John S. Lowengrub. Dynamics of multicomponent vesicles in a viscous fluid. *J. Comput. Phys.*, 229(1):119–144, 2010.
[32] I.S. Sokolnikoff. *Tensor analysis: theory and applications to geometry and mechanics of continua*. Applied mathematics series. Wiley, 1964.
[33] Jerome Solon, Olivier Gareil, Patricia Bassereau, and Yves Gaudin. Membrane deformations induced by the matrix protein of vesicular stomatitis virus in a minimal system. *J. Gen. Virol.*, 86(12):3357–3363, 2005.
[34] Knut Erik Teigen, Peng Song, John Lowengrub, and Axel Voigt. A diffuse-interface method for two-phase flows with soluble surfactants. *J. Comput. Phys.*, 230:375–393, 2011.
[35] Xiaoqiang Wang and Qiang Du. Modelling and simulations of multi-component lipid membranes and open membranes via diffuse interface approaches. *J. Math. Biol.*, 56:347–371, 2008.

https://doi.org/10.4208/cicp.OA-2015-0029 Published online by Cambridge University Press

[36] Guo-Wei Wei. Differential geometry based multiscale models. *Bulletin of Mathematical Biology*, 72(6):1562–1622, 2010.
[37] Steven Wise, Junseok Kim, and John Lowengrub. Solving the regularized, strongly anisotropic CahnHilliard equation by an adaptive nonlinear multigrid method. *J. Comput. Phys.*, 226(1):414–446, 2007.
[38] Kelin Xia, Xin Feng, Zhan Chen, Yiying Tong, and Guo-Wei Wei. Multiscale geometric modeling of macromolecules I: Cartesian representation. *J. Comput. Phys.*, 257, Part A:912–936, 2014.
[39] Jian-Jun Xu, Yin Yang, and John Lowengrub. A level-set continuum method for two-phase flows with insoluble surfactant. *J. Comput. Phys.*, 231(17):5897–5909, 2012.
[40] Shravan K. Veerapaneni, Abtin Rahimian, George Biros and Denis Zorin. A fast algorithm for simulating vesicle flows in three dimensions. *J. Comput. Phys.*, 230(14):5610–5634, 2011.
[41] Xiaofeng Yang, Ashley J. James, John Lowengrub, Xiaoming Zheng, and Vittorio Cristini. An adaptive coupled level-set/volume-of-fluid interface capturing method for unstructured triangular grids. *J. Comput. Phys.*, 217(2):364–394, 2006.
[42] Ou-Yang Zhong-can and W. Helfrich. Instability and deformation of a spherical vesicle by pressure. *Phys. Rev. Lett.*, 59:2486–2488, 1987.
[43] Y. C. Zhou, B. Lu, and A. A. Gorfe. Continuum electromechanical modeling of protein-membrane interactions. *Phys. Rev. E*, 82(4):041923, 2010.

https://doi.org/10.4208/cicp.OA-2015-0029 Published online by Cambridge University Press