

Approximate Formulae for Willmore Surfaces via Support Functions and Matlab

Phillip M. Galbo^a, Michael C. Barg^{*b}

^aBiology Department, Niagara University, NY

^bMathematics Department, Niagara University, NY

Student: phillipgalbo@gmail.com

Mentor: mbarg@niagara.edu*

ABSTRACT

Bio-membrane shapes are calculated numerically by minimizing a bending energy with area and volume constraints. We use a modified Helfrich's bending energy that includes buoyancy. Spheres, prolate and oblate discocytes, tori, Y-shapes, and dumbbells are some of the variety of bio-membranes that have been observed without buoyancy. Since the system of differential equations that minimizers must satisfy is extremely complex, many shapes are only known to exist through their numerical solutions. Indeed, exact analytical solutions for many shapes remain elusive. Here, we consider Y-shapes and dumbbells and give closed-form approximate formulae for these shapes using a support function method.

KEYWORDS Bio-membranes; Support Functions; Willmore Surfaces; Generating Curves; Polynomial Approximation; Numerical Solutions; Matlab; Helfrich's Bending Energy

INTRODUCTION

Bio-membrane shapes are often calculated numerically by minimizing a bending energy with area and volume constraints. In this work, we use the term bio-membrane to refer to lipid bilayers, vesicles, red blood cells, and other similar structures. Lipids can be described as hydrophobic molecules composed mostly of hydrogen and carbon molecules. A defining feature of lipids is that they are non-polar, and, therefore, insoluble in water. Lipids are an important bio-structure because they account for nearly 40% of the organic matter in the average human body and are the main components of lipid bilayers.¹ Such bilayer structures have been used as models for more complex structures like red blood cells and cell membranes. Lipid bilayers form because lipids are amphipathic molecules; meaning, they have hydrophilic heads and hydrophobic tails. This allows lipid heads to congregate in a manner that exposes them to surrounding aqueous solution while the lipid tails associate between the lipid heads, thus minimizing exposure to outer and inner cellular solution. A lipid bilayer can form various shapes and sizes depending on many factors, including the amount of lipids present in solution. While much recent work on these structures involves so-called multi-phase membranes that can arise when more than one type of lipid is present, here we concern ourselves with a single lipid type.²⁻⁴

From a mathematical perspective, a vesicle shape can be realized as a two-dimensional surface embedded in a three-dimensional space. Assuming axisymmetry, the bio-membrane is treated as a surface of revolution. The actual shape of the generating curve for the surface of revolution can be found as a solution to an optimization problem. Using Helfrich's bending energy,⁵ other authors have found such numerical solutions via a Lagrange multiplier method. The idea is to minimize a total energy functional

$$E_{Total} = E_{Bending} + Area\ Constraint + Volume\ Constraint \quad \text{Equation 1.}$$

consisting of Helfrich's bending energy, $E_{Bending}$, and two additional terms that are included with Lagrange multipliers to enforce surface area and volume constraints. The actual formula for E_{Total} appears in many references.⁶ We follow the same approach here, but include a term in the volume constraint to account for buoyant effects. In particular, instead of

introducing a Lagrange multiplier p_0 and including

$$\text{Volume Constraint} = \int_D p_0 dV$$

we use

$$\text{Volume Constraint} = \int_D (p_0 + bz) dV$$

where b is buoyancy, z is height above a fixed reference height $z = 0$, D is the interior region of the vesicle, and dV is volume measure. When $b = 0$, minimizers of E_{Total} are called Willmore surfaces. We also refer to minimizers with $b \neq 0$ as Willmore surfaces. As our goal is to give formulae for Willmore surfaces, we consider only the simplest bio-membranes, that is, homogeneous vesicles and single phase lipid bilayers. Prolate and oblate discocytes, stomatocytes, dumbbells, and Y -shapes are some of the energy minimizing shapes that have been observed. Details about the minimization process and the types of solutions are readily found in the literature.⁶⁻¹¹ Some analytical solutions are known. Spherical solutions can exist in the zero buoyancy case. Also, closed-form analytical solutions are known for tori and biconcave discoids (discocytes), in addition to other constant mean curvature surfaces.¹⁰⁻¹¹ Additionally, other authors have given formulae that approximate discocyte shapes of red blood cells.⁸ Our contribution in this work is to give closed-form approximate solutions for Y -shape and dumbbell solutions. We accomplish this task using a support function method.

If \mathbf{x} is a function representing a smooth surface and \mathbf{n} is a corresponding normal vector field on the surface, then a support function can be defined as $w = \mathbf{x} \cdot \mathbf{n}$.¹² In this way, at a given point in the surface, the support function measures the signed distance from the origin to the tangent plane. Support functions were first introduced by Minkowski in 1901, and they subsequently received attention in the study of minimal surfaces. Their use then diminished until recently.¹³ In recent work, examples of minimal surfaces are given by means of spherical parameterizations created from support functions.¹⁴ To this end, solutions of a particular partial differential equation involving w are analyzed. In another recent work, a Sturm-Liouville problem in terms of w is analyzed to help show the existence of certain minimal surfaces in the context of so-called period problems.¹³ Here, we adopt the “spherical parameterization” method common to both of these works for a curve \mathbf{x} which we interpret by means of a “circular parameterization.” The relevant details for a Y -shape vesicle are presented. We begin by giving some generating curve preliminaries, in part to establish notational conventions. After demonstrating the validity of our circular parameterization, we give the closed-form approximate formulae for Y -shape vesicles and dumbbells.

PRELIMINARIES

Assuming axisymmetry for our solutions, we seek a generating curve in the rz -plane. All the quantities described in

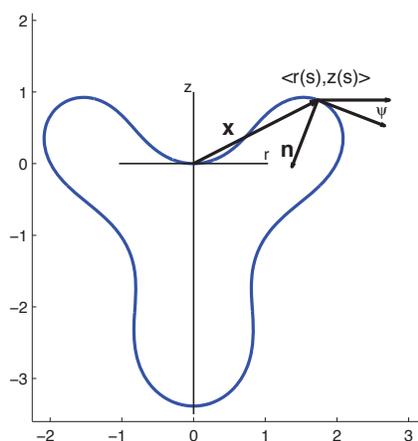


Figure 1. Generating curve for Y -shape defining r , z , ψ , and \mathbf{n}

Figure 1 are functions of s , arc length along the generating curve. $r = r(s)$ represents the radius of a section of the bio-membrane, and $z = z(s)$ is the height at which that specific radius is present on the bio-membrane. Performing the minimization of E_{Total} in Matlab, we determine r , z , and $\psi = \psi(s)$, where ψ is the angle between the horizontal

and tangent lines. The minimization is achieved by using a shooting method to compute a numerical solution to the Euler-Lagrange equations associated with Equation 1, subject to appropriate initial conditions and boundary conditions. In particular, $r(0) = z(0) = 0$ and $\psi(0) = 0$. Also, $r(L) = 0$ and $\psi(L) = -\pi$, where L is the a priori unknown length of the generating curve. It is computed as a solution in the minimization process. The details of the derivation of the appropriate system of ordinary differential equations are described elsewhere.⁶ Plotting r and z , we obtain the shape of the bio-membrane generating curve, $\mathbf{x} = \langle r, z \rangle$. As pictured in Figure 1, $\mathbf{x}(s)$ is the position vector for the point on the generating curve s units along the curve. The normal vector $\mathbf{n} = \mathbf{n}(s)$ is computed numerically as $\mathbf{n} = \langle \sin(\psi), -\cos(\psi) \rangle$.

SUPPORT FUNCTIONS AND APPROXIMATE FORMULAE

The support function is defined as the dot product of \mathbf{x} and \mathbf{n}

$$w = \mathbf{x} \cdot \mathbf{n} \tag{Equation 2}$$

In what follows, for a given w and \mathbf{n} , we demonstrate \mathbf{x} as a function of w and \mathbf{n} . The relationship is

$$\mathbf{x} = w\mathbf{n} + \frac{1}{\mathbf{n}' \cdot \mathbf{n}'} w' \mathbf{n}' \text{ if } \mathbf{n}' \cdot \mathbf{n}' \neq 0 \tag{Equation 3}$$

where \mathbf{n}' and w' are derivatives of \mathbf{n} and w with respect to the relevant independent variable, respectively. In particular, $\mathbf{n} = \mathbf{n}(\theta)$ and $w = w(\theta)$ in the following ellipse example, while $\mathbf{n} = \mathbf{n}(s)$ and $w = w(s)$ in the Willmore surface approximations. Higher dimensional analogues of Equation 3 have been used as spherical parameterizations for certain minimal surfaces.¹³⁻¹⁴ As in those works, it is convenient to interpret \mathbf{x} , defined by Equation 3, as a mapping from the unit circle, S^1 , to \mathbb{R}^2 with the generating curve as the image. For some simple geometries, this is a well-defined map. In particular, we work through an example to verify Equation 3 for an ellipse. Figure 2 is an illustration for how S^1 is deformed, via \mathbf{x} , into an ellipse. Unfortunately, for Y-shape and dumbbell generating curves, a single map $\mathbf{x} : S^1 \rightarrow \mathbb{R}^2$ is insufficient to produce the entire generating curve. However, we explain how formulae can be derived for Willmore surfaces by using Equation 3.

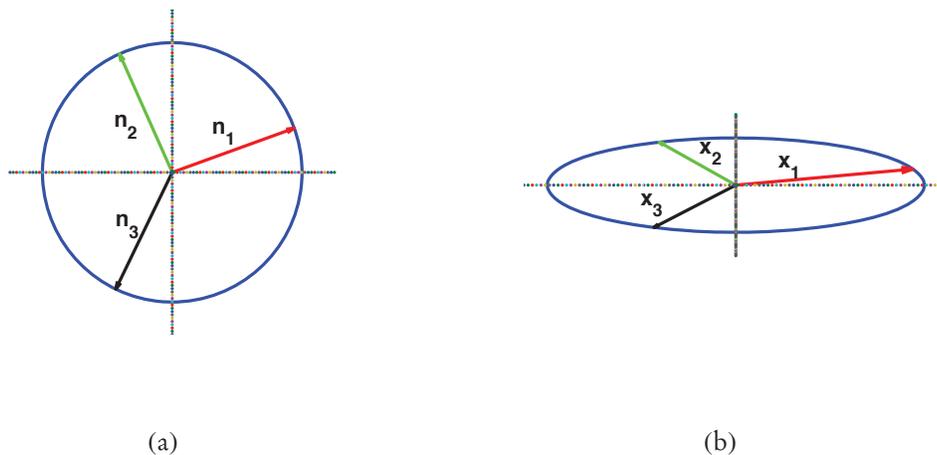


Figure 2. (a) S^1 with normal vectors $\mathbf{n}_i, i = 1, 2, 3$ (b) Ellipse with images $\mathbf{x}(\mathbf{n}_i) = \mathbf{x}_i, i = 1, 2, 3$

Ellipse Example

To show how Equation 3 can be used as a representation for the shape of a generating curve \mathbf{x} , we will consider an ellipse parameterized by $\mathbf{x} = \langle a \cos \theta, b \sin \theta \rangle$ for $\theta \in [0, 2\pi]$ and fixed $a, b \in (0, \infty)$. One can compute the unit normal vector

$$\mathbf{n} = \frac{1}{\sqrt{b^2 \cos^2 \theta + a^2 \sin^2 \theta}} \langle b \cos \theta, a \sin \theta \rangle$$

From Equation 2, the support function for the ellipse is

$$w = \frac{ab}{\sqrt{b^2 \cos^2 \theta + a^2 \sin^2 \theta}}$$

The derivative of the support function is

$$w' = \frac{dw}{d\theta} = \frac{ab(b^2 - a^2) \cos \theta \sin \theta}{(b^2 \cos^2 \theta + a^2 \sin^2 \theta)^{3/2}}$$

One can also compute the derivative of \mathbf{n} . As \mathbf{n} is a vector, the derivative is computed component-wise. The derivative of the first component is found by the quotient rule:

$$\begin{aligned} \frac{d}{d\theta} \left(\frac{b \cos \theta}{(b^2 \cos^2 \theta + a^2 \sin^2 \theta)^{1/2}} \right) &= \frac{(b^2 \cos^2 \theta + a^2 \sin^2 \theta)^{1/2}(-b \sin \theta) - \frac{1}{2}(b^2 \cos^2 \theta + a^2 \sin^2 \theta)^{-1/2}(2b^2 \cos \theta(-\sin \theta) + 2a^2 \sin \theta \cos \theta)(b \cos \theta)}{(b^2 \cos^2 \theta + a^2 \sin^2 \theta)} \\ &= \frac{(b^2 \cos^2 \theta + a^2 \sin^2 \theta)(-b \sin \theta) - (b^2 \cos \theta(-\sin \theta) + a^2 \sin \theta \cos \theta)(b \cos \theta)}{(b^2 \cos^2 \theta + a^2 \sin^2 \theta)^{3/2}} \\ &= \frac{-a^2 b \sin^2 \theta \sin \theta - a^2 b \sin \theta \cos^2 \theta}{(b^2 \cos^2 \theta + a^2 \sin^2 \theta)^{3/2}} \\ &= \frac{-a^2 b \sin \theta}{(b^2 \cos^2 \theta + a^2 \sin^2 \theta)^{3/2}} \end{aligned}$$

Likewise, the derivative of the second component can be computed as

$$\begin{aligned} \frac{d}{d\theta} \left(\frac{a \sin \theta}{(b^2 \cos^2 \theta + a^2 \sin^2 \theta)^{1/2}} \right) &= \frac{(b^2 \cos^2 \theta + a^2 \sin^2 \theta)^{1/2}(a \cos \theta) - \frac{1}{2}(b^2 \cos^2 \theta + a^2 \sin^2 \theta)^{-1/2}(2b^2 \cos \theta(-\sin \theta) + 2a^2 \sin \theta \cos \theta)(a \sin \theta)}{(b^2 \cos^2 \theta + a^2 \sin^2 \theta)} \\ &= \frac{(b^2 \cos^2 \theta + a^2 \sin^2 \theta)(a \cos \theta) - (b^2 \cos \theta(-\sin \theta) + a^2 \sin \theta \cos \theta)(a \sin \theta)}{(b^2 \cos^2 \theta + a^2 \sin^2 \theta)^{3/2}} \\ &= \frac{b^2 a \cos^2 \theta \cos \theta + b^2 a \cos \theta \sin^2 \theta}{(b^2 \cos^2 \theta + a^2 \sin^2 \theta)^{3/2}} \\ &= \frac{b^2 a \cos \theta}{(b^2 \cos^2 \theta + a^2 \sin^2 \theta)^{3/2}} \end{aligned}$$

We see that

$$\mathbf{n}' = \frac{d\mathbf{n}}{d\theta} = \frac{1}{(b^2 \cos^2 \theta + a^2 \sin^2 \theta)^{3/2}} \langle -a^2 b \sin \theta, b^2 a \cos \theta \rangle$$

For the ellipse, we note $\mathbf{n}' \cdot \mathbf{n}' \neq 0$ for any θ . Thus, we can check the validity of Equation 3 for the ellipse. First,

$$w\mathbf{n} = \frac{ab \langle b \cos \theta, a \sin \theta \rangle}{b^2 \cos^2 \theta + a^2 \sin^2 \theta} \tag{Equation 4.}$$

The second term in Equation 3 is

$$\begin{aligned} \frac{1}{\mathbf{n}' \cdot \mathbf{n}'} w' \mathbf{n}' &= \frac{(b^2 \cos^2 \theta + a^2 \sin^2 \theta)^3}{a^4 b^2 \sin^2 \theta + b^4 a^2 \cos^2 \theta} \left(\frac{ab(b^2 - a^2) \cos \theta \sin \theta}{(b^2 \cos^2 \theta + a^2 \sin^2 \theta)^{3/2}} \right) \frac{\langle -a^2 b \sin \theta, b^2 a \cos \theta \rangle}{(b^2 \cos^2 \theta + a^2 \sin^2 \theta)^{3/2}} \\ &= \frac{(b^2 - a^2) \cos \theta \sin \theta \langle -a^2 b \sin \theta, b^2 a \cos \theta \rangle}{ab(a^2 \sin^2 \theta + b^2 \cos^2 \theta)} \end{aligned} \tag{Equation 5.}$$

After distributing and adding componentwise, Equation 4 and Equation 5 yield

$$\mathbf{x} = \frac{\langle ab^2 \cos \theta - ab^2 \sin^2 \theta \cos \theta + a^3 \sin^2 \theta \cos \theta, a^2 b \sin \theta + b^3 \cos^2 \theta \sin \theta - a^2 b \cos^2 \theta \sin \theta \rangle}{a^2 \sin^2 \theta + b^2 \cos^2 \theta}$$

Upon factoring and applying the identity $\sin^2 \theta + \cos^2 \theta = 1$, we obtain

$$\begin{aligned} \mathbf{x} &= \frac{1}{a^2 \sin^2 \theta + b^2 \cos^2 \theta} \langle a \cos \theta (b^2 \cos^2 \theta + a^2 \sin^2 \theta), b \sin \theta (a^2 \sin^2 \theta + b^2 \cos^2 \theta) \rangle \\ &= \langle a \cos \theta, b \sin \theta \rangle \end{aligned}$$

As mentioned above, given w , one can interpret $\mathbf{x} = \mathbf{x}_w$ from Equation 3 as a mapping $\mathbf{x} : S^1 \rightarrow \mathbb{R}^2$ by $\mathbf{x}_w(\mathbf{n}) = w\mathbf{n} + \frac{1}{\mathbf{n}' \cdot \mathbf{n}'} w' \mathbf{n}'$. In Figure 2, we highlight this approach by showing three normal vectors $\mathbf{n}_1, \mathbf{n}_2$, and \mathbf{n}_3 and their respective images $\mathbf{x}_1, \mathbf{x}_2$, and \mathbf{x}_3 in the ellipse. We view the function \mathbf{x}_w as deforming S^1 in Figure 2(a) to the ellipse in Figure 2(b).

Approximations from Matlab

In this section, we give the closed-form approximate formulae for our Y-shape and dumbbell Willmore surface generating curves. We also check the validity of Equation 3 for the Y-shape Willmore surface. To derive the desired formulae, we use Matlab to approximate w and $\psi + (\pi/2)$ as polynomials. Our numerical solution from Matlab provides arrays for \mathbf{x} and ψ , and thus, we can form $\mathbf{n} = \langle \sin(\psi), -\cos(\psi) \rangle$. Matlab can then compute an array for the support function $w = \mathbf{x} \cdot \mathbf{n}$ (see Figure 3(a)). To obtain the desired approximate formulae for \mathbf{x} , we use Matlab to approximate w with a

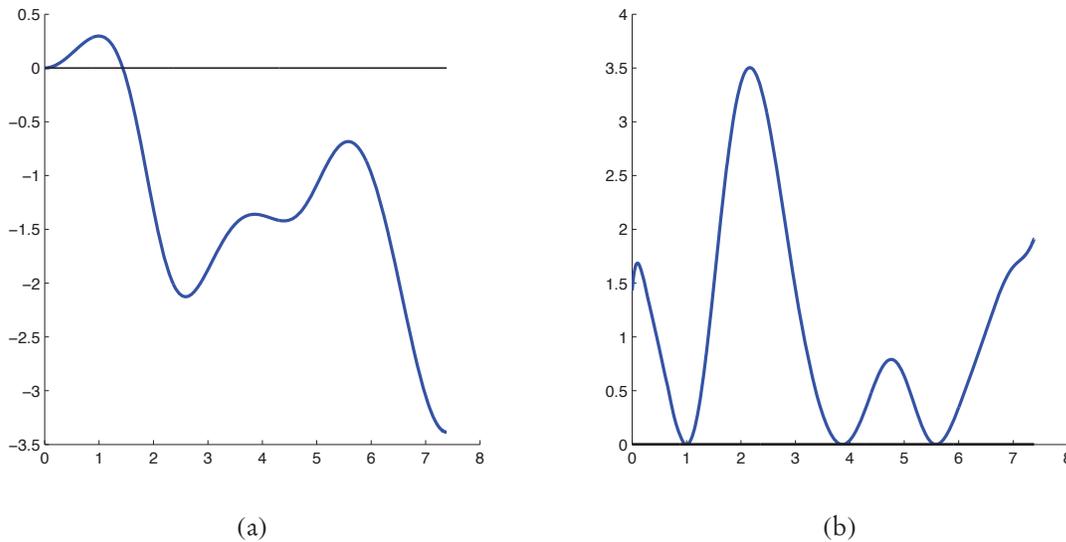


Figure 3. (a) Support function for Y-shape (b) $\mathbf{n}' \cdot \mathbf{n}'$ with zeros at $s \approx 0.9985, s \approx 3.8640, s \approx 5.5841$

polynomial. Coefficients for the polynomial approximation of w were determined by the expression ‘polyfit’, which returns the coefficients for a polynomial of any desired degree. In general, the approximation given by ‘polyfit’ gets better as the degree increases. One way to measure the accuracy of the approximation is to compute the total squared error, i.e., compute $\sum_{i=1}^{1017} (w_i - w_{p,i})^2$, where w_i is the i th component of the array w and $w_{p,i}$ is the i th component of the corresponding polynomial approximation as given by ‘polyfit’ for some chosen degree. To choose the degree, we considered the total squared error for various degrees, in addition to visually inspecting superimposed plots of w and w_p . Using the 16th degree polynomial yields a total squared error less than 4%, a tolerance that we deemed acceptable. Of course, greater accuracy could be obtained with a higher degree polynomial approximation. For example, a 19th degree polynomial can be used to achieve a total squared error less than 1%. Ultimately, we chose a 16th degree polynomial since it was the lowest degree for which the approximate polynomial appeared nearly indistinguishable from w . The approximation is $w \approx \sum_{i=0}^{16} c_i s^i$, where the coefficients c_i are given in Table 1. With the coefficients for the polynomial approximation of w , one can calculate coefficients for a polynomial approximation of w' by computing ic_i as the coefficient of s^{i-1}

$$w' = \frac{dw}{ds} \approx \sum_{i=1}^{16} ic_i s^{i-1}$$

A similar approach was taken to obtain an approximation for $\theta = \psi + (\pi/2)$, which can then be used to approximate \mathbf{n} as a function of s . In particular, θ is found from Matlab by taking the solution $\psi(s)$ (as pictured in Figure 1) and adding $\pi/2$. We add $\pi/2$ because ψ is the angle between the horizontal and tangent lines of the generating curve, while we desire an approximation for the angle between the horizontal and normal lines of the generating curve. For the approximation, we find

$$\theta(s) = \psi(s) + \pi/2 \approx \sum_{i=0}^{16} d_i s^i \tag{Equation 6.}$$

where the coefficients d_i are given in Table 1. This polynomial allows us to obtain an approximate formula for \mathbf{n} in terms

| Term | Coefficient for w | Coefficient for θ |
|----------|-------------------------------|-------------------------------|
| s^0 | $c_0 = -0.000060771569692$ | $d_0 = 1.570924108727812$ |
| s^1 | $c_1 = 0.114602595804498$ | $d_1 = 1.196236274933391$ |
| s^2 | $c_2 = -1.719844524602488$ | $d_2 = 1.241628057733661$ |
| s^3 | $c_3 = 15.603335003104629$ | $d_3 = -6.569163020942945$ |
| s^4 | $c_4 = -49.235439770081733$ | $d_4 = 15.219754968954797$ |
| s^5 | $c_5 = 85.851964954476983$ | $d_5 = -22.249839707112130$ |
| s^6 | $c_6 = -93.192274800396135$ | $d_6 = 20.931676370762279$ |
| s^7 | $c_7 = 66.572113028538610$ | $d_7 = -13.454835395052019$ |
| s^8 | $c_8 = -32.643107035876724$ | $d_8 = 6.171998900701146$ |
| s^9 | $c_9 = 11.327063427562599$ | $d_9 = -2.068316556605748$ |
| s^{10} | $c_{10} = -2.832752329974763$ | $d_{10} = 0.510835578073774$ |
| s^{11} | $c_{11} = 0.513673827418474$ | $d_{11} = -0.092743939106679$ |
| s^{12} | $c_{12} = -0.067051360050457$ | $d_{12} = 0.012205101258158$ |
| s^{13} | $c_{13} = 0.006149427856176$ | $d_{13} = -0.001130878451232$ |
| s^{14} | $c_{14} = -0.000376443328925$ | $d_{14} = 0.000069864759188$ |
| s^{15} | $c_{15} = 0.000013820393660$ | $d_{15} = -0.000002580798593$ |
| s^{16} | $c_{16} = -0.000000230250017$ | $d_{16} = 0.000000043089161$ |

Table 1. Coefficients for w and θ for a Y-shape

of the arc length on the generating curve. For convenience, we parameterize the unit circle as

$$\mathbf{n} = \langle -\cos(\theta), -\sin(\theta) \rangle \tag{Equation 7}$$

with $0 \leq \theta \leq 2\pi$. The negative value indicates that the normal vector is pointed toward the inside of the generating curve. As is evident from Equation 3, an expression for \mathbf{n}' is needed. Upon differentiating Equation 7 with respect to s , we find

$$\mathbf{n}' = \frac{d\mathbf{n}}{ds} = \langle \sin(\theta), -\cos(\theta) \rangle \theta' \tag{Equation 8}$$

where $\theta' = \frac{d\theta}{ds}$. We approximate θ' from Equation 6 as a 15th degree polynomial with id_i as the coefficient of s^{i-1} . Now, using our approximations for w , w' , \mathbf{n} , and \mathbf{n}' , the formula for the generating curve of the Y-shape Willmore surface is

$$\mathbf{x}(s) = w(s)\mathbf{n}(s) + \frac{1}{\mathbf{n}'(s) \cdot \mathbf{n}'(s)} w'(s)\mathbf{n}'(s) \tag{Equation 9}$$

with domain $\{s \in [0, L] \mid \mathbf{n}'(s) \cdot \mathbf{n}'(s) \neq 0\}$. See Figure 3(b) for a graph of $\mathbf{n}' \cdot \mathbf{n}'$ showing zeros at approximately $s = 0.998476759700923$, $s = 3.863998297698298$, and $s = 5.584116014650597$. For our Y-shape, L is found to be 7.385119679927381. While the formula for $\mathbf{x}(s)$ in Equation 9 is only valid for this one Y-shape, the procedure for creating \mathbf{x} holds regardless of the particular bio-membrane shape. Figure 4(a) shows the generating curve that we obtained from Matlab as a solution to the Euler-Lagrange equations. The figure also contains a plot of our approximate \mathbf{x} from Equation 9. The agreement is excellent. In fact, the images are indistinguishable. The points not on the curve correspond to arc lengths close to those that make $\mathbf{n}' \cdot \mathbf{n}' = 0$, thus giving erroneous results.

To further confirm the validity of our formula, we checked the approximation $\mathbf{x}(2.0554) \approx \langle r(2.0554), z(2.0554) \rangle$. To this end, $w(2.0554)$ was determined using Matlab with command ‘polyval’ which gave the numerical value of -1.44600781990. The derivative $w'(2.0554)$ was computed numerically with ‘polyval’ which resulted in the numerical value of -2.34828659727. These last two values are rounded for ease of exposition. For the normal vector, Matlab expression ‘polyval’ approximated $\theta(2.0554)$. This resulted in an approximation for \mathbf{n} :

$$\mathbf{n}(2.0554) \approx \langle -0.358125822681700, -0.933673334270908 \rangle$$

which is in good agreement with the normal vector field at the corresponding location $s \approx 2.0554$ on the curve (see Figure 1). For \mathbf{n}' , we took a similar approach with the derivative of Equation 8 while still using Matlab expression ‘polyval’ for $\theta(2.0554)$. Ultimately, the final value of \mathbf{n}' is

$$\mathbf{n}'(2.0554) \approx \langle -1.732347541734692, 0.664470501386673 \rangle$$

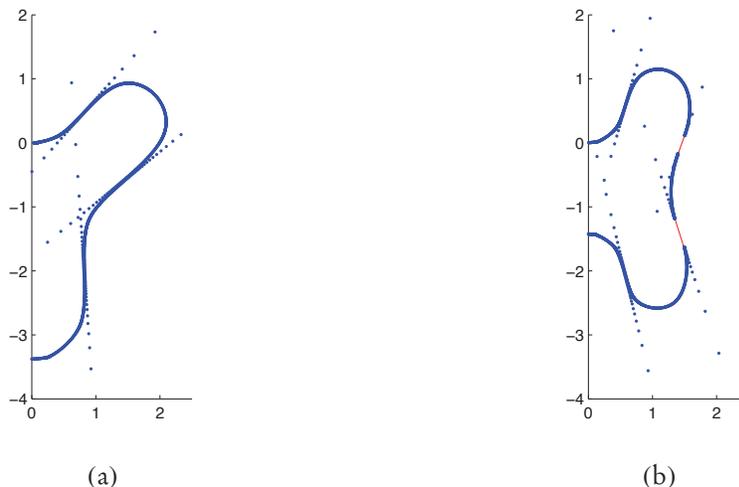


Figure 4. Generating curve from Matlab and approximate solution \mathbf{x} . (a) Y-shape (b) Dumbbell

| Variable | Value at $s \approx 2.0554$ |
|---------------|--|
| w | -1.446007819900795 |
| w' | -2.348286597274282 |
| \mathbf{n} | $\langle -0.358125822681700, -0.933673334270908 \rangle$ |
| \mathbf{n}' | $\langle -1.732347541734692, 0.664470501386673 \rangle$ |
| r | 1.732183240619059 |
| z | 0.887509330469256 |

Table 2. Values provided by Matlab for variables $w, w', \mathbf{n}, \mathbf{n}'$ at $s \approx 2.0554$.

All of these values are summarized in Table 2. When substituting the values from Table 2 into Equation 9, the numerical approximation is

$$\begin{aligned} \mathbf{x}(2.0554) &= -1.4460 \langle -0.3581, -0.9337 \rangle + \frac{(-2.3483) \langle -1.7323, 0.6645 \rangle}{\langle -1.7323, 0.6645 \rangle \cdot \langle -1.7323, 0.6645 \rangle} \\ &= \langle 1.699549341071895, 0.896839701002235 \rangle \end{aligned}$$

where we rounded to four decimal places for ease of exposition. The values of r and z at $s \approx 2.0554$ are 1.732183240619059 and 0.887509330469256, respectively. Thus, we see that \mathbf{x} is in fact a decent approximation to the numerical solution $\langle r, z \rangle$. In particular, at $s \approx 2.0554$, our approximate formula varies from the numerical solution by $\|\mathbf{x} - \langle r, z \rangle\| \approx 0.033941526393742$.

For completeness, we note that coefficients for w and θ can be obtained in order to compute an approximate formula \mathbf{x} for a dumbbell shape Willmore surface. For a dumbbell, we used a 23rd degree polynomial to approximate w . The method for obtaining \mathbf{x} is the same as above. Figure 4(b) shows the generating curve from the solution of the Euler-Lagrange equations and the approximate \mathbf{x} . Again, we see excellent agreement when $\mathbf{n}' \cdot \mathbf{n}' \neq 0$. For the dumbbell, $\mathbf{n}' \cdot \mathbf{n}' = 0$ at four values of s . While the image in Figure 4(b) has two gaps in the approximate formula \mathbf{x} , other results with different numerical integration tolerances lead to slightly different pictures with one or zero gaps. We included Figure 4(b) with two gaps in order to highlight the regions of the curve where \mathbf{n} is essentially constant and thus $\mathbf{n}' \cdot \mathbf{n}' = 0$.

CONCLUSIONS

Y-shape and dumbbell shape lipid bilayers were computed by minimizing Helfrich’s bending energy with buoyancy, subject to area and volume constraints. When analyzing the Y-shape, it is understood that the support function for this bio-membrane was approximated by a 16th degree polynomial (Table 1). Furthermore, when analyzing the dumbbell shape, the support function was approximated by a 23rd degree polynomial. Within a total squared error less than 4%, these polynomials approximated w and θ which were then used to obtain the generating curves for both bio-membrane shapes.

Ultimately, we interpret \mathbf{x} as a mapping to show how a unit circle is deformed into the generating curve. However, this expression for the generating curve in terms of w and \mathbf{n} was only partially verified, owing to the fact that **Equation 3** depended on $\mathbf{n}' \cdot \mathbf{n}' \neq 0$. With this restriction, the equation failed at certain locations along the generating curves of Y-shape and dumbbell bio-membranes. Our systematic approach can provide a way of interpreting and examining other bio-membrane shapes of interest. For example, one might use the polynomial approximation for w as a way to determine a formula for the Gaussian and mean curvatures of Willmore surfaces. Further study remains on this approximation approach before such applications are fully understood.

ACKNOWLEDGEMENTS

The authors thank the Niagara University Research Council for financial support through the 2014-2015 Research Support Grant.

REFERENCES

1. Brooker, R. et. al. (2008) *Biology: Second Edition*, The McGraw Hill Companies, New York.
2. Baginski, F., Croce, R., Gillmor, S., and Krause, R. (2014) Numerical investigations of the role of curvature in strong segregation problems on a given surface, *Appl. Math. Comput.* **227**, 399–411.
3. Baumgart, T., Hess, S. T., and Webb, W. W. (2003) Imaging coexisting fluid domains in biomembrane models coupling curvature and line tension, *Nature* **425**, 821–824.
4. Gillmor, S., Lee, J., and Ren, X. (2011) The role of Gauss curvature in a membrane phase separation problem *Phys. D* **240**, No. 24, 1913–1927
5. Helfrich, W. (1973) Elastic properties of lipid bilayers: theory and possible experiments, *Z. Naturforsch* **28 c**, 693–703.
6. Seifert, U., Berndl, K., and Lipkowsky, R. (1991) Shape transformations of vesicles: phase diagram for spontaneous curvature and bilayer-coupling models, *Phys. Rev. A* **44** No. 2, 1182–1202.
7. Blyth, M. G., and Pozrikidis, C. (2004) Solution space of axisymmetric capsules enclosed by elastic membranes, *Eur. J. Mech. A Solids* **23**, 877–892.
8. Luke, J. (1982) A method for the calculation of vesicle shapes, *SIAM J. Appl. Math.* **42**, 333–345.
9. Duelling, H., and Helfrich, W. (1976) The curvature elasticity of fluid membranes: a catalogue of vesicle shapes, *Le Journal De Physique* **37**, 1335–1345.
10. Mladenov, I. M. (2002) New solutions of the shape equation, *Eur. Phys. J. B* **29**, 327–330.
11. Tu, Z. C., and Ou-Yang, Z. C. (2004) A geometric theory on the elasticity of bio-membranes, *J. Phys. A: Math. Gen.* **37**, 11407–11429.
12. Stoker, J. J. (1969) *Differential Geometry*, John Wiley & Sons, Inc., New York.
13. Baginski, F., and Ramos Batista, V. (2011) Solving period problems for minimal surfaces with the support function, *Adv. Appl. Math. Sci.* **9**, 85–114.
14. Baginski, F. (2002) Special functions on the sphere with applications to minimal surfaces, *Adv. in Appl. Math.* **28**, 360–394.

ABOUT THE STUDENT AUTHOR

Phillip Galbo graduated from Niagara University in 2015 with a B.S. in Biology and a minor in Chemistry. The research in this manuscript began during his Sophomore year and was completed during his Senior year. In 2014, Phillip was inducted into the national biological honor society for undergraduates. Upon graduating, Phillip accepted a research position at Roswell Park Cancer Institute in the Department of Neuro-Oncology where he is currently pursuing his passion for oncological research.

PRESS SUMMARY

Bio-membrane shapes are calculated by minimizing a bending energy with area and volume constraints. Many different minimizing shapes are known to exist, including Y-shapes and dumbbells. However, exact solutions remain elusive. We describe a method to use support functions to give approximate closed-form formulae for such shapes, thus offering a way to evaluate and analyze the wide array of bio-membrane shapes.