The Tethered Infinitesimal Tori and Spheres Algorithm: A Versatile Calculator for Axisymmetric Problems in Equilibrium Membrane Mechanics

Gerald H. W. Lim[†‡] and Greg Huber[†§*]

[†]Richard Berlin Center for Cell Analysis & Modeling and Department of Cell Biology, University of Connecticut Health Center, Farmington, Connecticut; [‡]Department of Biochemistry and Molecular Biology, Baylor College of Medicine, Houston, Texas; and [§]Department of Mathematics, University of Connecticut, Storrs, Connecticut

ABSTRACT Constrained minimization of energy functionals is a central part, and usually the difficult part, of solving problems in the equilibrium mechanics of biological and biomimetic membranes. The inherent difficulties of the conventional variational-calculus approach prevents the numerical calculation involved from being made routine in the analyses of experimental results. We have developed a simulated annealing-based computational technique for routinizing the task of constrained minimization of energy functionals governing whole, or small patches of whole, fluid membranes with axisymmetry, spherical topology, and no domains of inhomogeneity. In this article, we describe the essential principles of the technique and apply it to five examples to demonstrate its versatility. It gives membrane shapes that are automatically stable to axisymmetric perturbations. Presently, it can account for constraints on 1), the membrane area or the effective membrane tension; 2), the enclosed volume or the effective pressure difference across the membrane thickness; and 3), the axial end-to-end distance or the applied axial point force.

INTRODUCTION

A central and generally difficult task in solving problems in equilibrium mechanics of biological and biomimetic membranes is the constrained minimization of the energy functionals governing the equilibrium mechanical properties of these membranes.

In the conventional variational-calculus approach to solving such problems, one parametrizes the mathematical surface S representing a membrane, then writes down, and solves the Euler-Lagrange equations governing the equilibrium mechanical behavior of the membrane. This method has several intrinsic difficulties. First, it may be impossible in practice, particularly for general three-dimensional situations, to write down the Euler-Lagrange equations due to the complexity of the energy functional, constraints, and/or boundary conditions. Second, even if the Euler-Lagrange equations can be written down analytically in simpler, usually axisymmetric, problems, their nonlinearity means that they generally have to be solved numerically, itself a nontrivial exercise in numerical computation. Third, it is inflexible, in that minor changes in the energetics, constraints, or boundary conditions often require one to expend considerable effort to reparameterize S, write a new set of Euler-Lagrange equations, and write a new computer program instead of making minor modifications to an existing one. Fourth, the solutions are strictly stationary and additional calculations are required to determine their stability.

We have developed a purely-numerical constrained-minimization method for the purpose of overcoming these difficulties and reducing to a routine the task of energy minimization in axisymmetric problems involving fluid membranes whose bending elasticity is a functional of the area-difference elasticity (ADE) type or one of its limiting forms (1–3). This tethered infinitesimal tori and spheres algorithm, so-called because the membrane is represented as the infinitesimal limit of rotated circles and circle arcs, is based on simulated annealing. In analogy with the physical process of annealing, the algorithm at every step replaces the current membrane shape with a stochastically determined shape that, in a certain sense, is nearby in the space of shapes. The determination depends on the difference in the values of the energy functional and on a global computational parameter (the computational temperature) that is gradually lowered during the procedure. We give below a brief overview of the ADE model, followed by a detailed description of the algorithm and then a demonstration of its versatility by applying it to five examples.

Bending energy and constraints

The simplest and best-studied class of membranes is fluid lipid bilayers with uniform monolayer composition. They resist bending deformation but not in-plane deformation. Their bending elasticity gives rise to a bending energy F_{ADE} described by the ADE model (1,2) or one of its specialized forms (see (3) for a comprehensive review). The general form of the ADE model is

$$F_{\mathrm{ADE}}[S] = \frac{B_1}{2}\int_S [2H(\mathbf{r}) - C_0]^2 \mathrm{d}A + \frac{\pi B_2}{2A_0D^2} \times (\Delta A[S] - \Delta A_0)^2 + B_3\int_S K(\mathbf{r})dA, \quad (1)$$

where S is the mathematical surface representing an entire closed membrane or a small patch of a closed membrane

Submitted May 22, 2008, and accepted for publication October 30, 2008.

*Correspondence: huber@uchc.edu

Editor: Arup Chakraborty.

0006-3495/09/03/2064/18 $2.00

doi: 10.1016/j.bpj.2008.10.074

(4); B_1 is the local bending modulus; H is the mean curvature at a point $\mathbf{r}$ on S; C_0 is the spontaneous curvature; B_2 is the nonlocal bending modulus; A_0 is the fixed area of the entire membrane (even if S is a membrane patch); D is the distance between the neutral surfaces of the outer and inner membrane leaflets; ΔA, given by

$$\Delta A[S] \equiv 2D \int_S H(\mathbf{r}) dA, \tag{2}$$

is the actual area difference between the neutral surfaces of the outer and inner membrane leaflets; ΔA_0 is the preferred leaflet area difference of S; B_3 is the Gaussian bending modulus; and K is the Gaussian curvature at a point $\mathbf{r}$ on S.

Equation 1 involves four integrals over surface S,

$$I_{\mathrm{HH}}[S] \equiv \int_S H^2(\mathbf{r}) dA, \tag{3}$$

$$I_{\mathrm{H}}[S] \equiv \int_S H(\mathbf{r}) \mathrm{d}A, \tag{4}$$

$$I_{\mathrm{K}}[S] \equiv \int_S K(\mathbf{r}) \mathrm{d}A, \tag{5}$$

and

$$A[S] \equiv \int_S \mathrm{d}A, \tag{6}$$

which are the respective integrated mean-curvature squared, integrated mean curvature, integrated Gaussian curvature, and area of S. A rescaling of S ($\mathbf{r} \rightarrow \lambda \mathbf{r}$) gives $I_{\mathrm{HH}} \rightarrow I_{\mathrm{HH}}$, $I_{\mathrm{H}} \rightarrow \lambda I_{\mathrm{H}}$, $I_{\mathrm{K}} \rightarrow I_{\mathrm{K}}$, and $A \rightarrow \lambda^2 A$, i.e., I_{HH} and I_{K} are scale-invariant, whereas I_{H} and A are not.

Rewriting Eq. 1 in terms of Eqs. 3–6 gives

$$\frac{F_{\mathrm{ADE}}}{B_1} = \frac{F_{\mathrm{b}}}{B_1} + \frac{C_0^2 A}{2} + \frac{B_3 I_{\mathrm{K}}}{B_1}, \tag{7}$$

where

$$\frac{F_{\mathrm{b}}}{B_1} = 2I_{\mathrm{HH}} + \frac{\overline{B}_2}{2}\left[\left(\frac{I_{\mathrm{H}}}{R_0}\right)^2 - 2\overline{m}_0\left(\frac{I_{\mathrm{H}}}{R_0}\right)\right], \tag{8}$$

$$\overline{B}_2 \equiv \frac{B_2}{B_1}, \tag{9}$$

$$R_0 \equiv \sqrt{\frac{A_0}{4\pi}}, \tag{10}$$

$$\overline{m}_0 \equiv \frac{2\pi R_0}{D}\left(\frac{\Delta A_0}{A_0} + \frac{B_1 D C_0}{B_2 \pi}\right), \tag{11}$$

and we have omitted a shape-independent constant $\pi \overline{B}_2 (\Delta A_0)^2/(2A_0 D^2)$ on the right-hand side of Eq. 7. The dimensionless parameter $\overline{m}_0$ is referred to as the reduced effective area difference between the neutral surfaces of the outer and inner membrane leaflets. We denote the length scale of S to be R and define the dimensionless quantities $\overline{R}_0 \equiv R_0/R$, $\overline{H} \equiv HR$, $\overline{A} \equiv A/(4\pi R^2)$, and $\overline{C}_0 \equiv C_0 R$. Under a scale transformation, both R and R_0 are scaled by the same factor λ, H is scaled by λ^{-1}, and A is scaled by λ^2, so that $\overline{R}_0$, $\overline{H}$, $\overline{A}$, and the ratio

$$\frac{I_{\mathrm{H}}}{R_0} = \frac{4\pi}{\overline{R}_0} \int_S \overline{H} \mathrm{d}\overline{A} \tag{12}$$

in Eq. 8 are scale-invariant. It follows that F_{ADE} and F_{b} of Eqs. 7 and 8 are scale-invariant at fixed $\overline{m}_0$ and $\overline{C}_0$.

In the case where S represents an entire closed membrane, the last two terms on the right-hand side of Eq. 7 can be omitted because they are shape-independent constants. The penultimate term is a shape-independent constant due to the whole membrane area $A = A_0$ being fixed. The last term is a shape-independent constant because of the fixed topology of the closed membrane, which, according to the Gauss-Bonnet theorem (5), leads to a constant I_{K}. Therefore, Eq. 7 reduces to Eq. 8.

In the case where S represents a small patch of a closed membrane ($A << A_0$), the penultimate term of Eq. 7 needs to be retained because A depends on the shape of S and, thus, is not constant. The last term of Eq. 7 can be omitted: By the Gauss-Bonnet theorem,

$$I_{\mathrm{K}}[S] = 2\pi\chi[S] - \oint_{\partial S} K_{\mathrm{g}}(s) \mathrm{d}s, \tag{13}$$

where χ is the Euler characteristic of S, ∂S is the boundary contour of S, and K_{g} is the geodesic curvature at a point s on ∂S. The first term on the right-hand side of Eq. 13 is a constant and the second term, the patch-boundary contribution to I_{K}, is always cancelled by an equal and opposite boundary contribution from the rest of the larger membrane (since I_{K} for the overall closed membrane must be constant). Thus, in this case, Eq. 7 reduces to

$$\frac{F_{\mathrm{b}}'}{B_1} = \frac{F_{\mathrm{b}}}{B_1} + \frac{C_0^2 A}{2}, \tag{14}$$

where F_{b}/B_1 is given by Eq. 8.

Henceforth, we focus on an axisymmetric S without topological handles, arising from rotating a contour C about an axis.

We define V to be the volume enclosed by S, and Z to be the projection of the end-to-end distance of C onto the cylindrical axis.

Constraints: closed membranes

A closed membrane S, in a solution with fixed osmolarity, is always subject to constraints on its area and volume (3): $A = A_0$ and $V = V_0$, where V_0 is the fixed volume set by the solution osmolarity. Thus, the functional governing the equilibrium mechanics of a freely-suspended closed membrane (vesicle) is

$$W[S] = F_{\mathrm{b}}[S] + \sigma A[S] - pV[S], \tag{15}$$

where F_b is given by Eq. 8, and σ and p are Lagrange multipliers enforcing the constraints $A = A_0$ and $V = V_0$, respectively.

Deformation of a closed membrane by various means will introduce additional terms into Eq. 15. A situation that we consider is axial deformation of the membrane by applying an axial force with magnitude f at each pole of S, the governing functional for which is (6):

$$W[S] = F_b[S] + \sigma A[S] - pV[S] - fZ[S]. \quad (16)$$

Note that either f or Z can be arranged to be the constrained quantity (7). In the fixed-Z situation, f is the Lagrange multiplier enforcing the constraint $Z = Z_0$, where Z_0 is the value at which Z is to be fixed. In the fixed-f situation, f is set to a constant value f_0 and the resulting force balance gives the final Z. Equation 15 is a special case of Eq. 16 with f fixed at $f_0 = 0$. Thus, in this context, problems involving closed membranes are divided into the fixed-Z or the fixed-f type.

We define the dimensionless quantities $\overline{A}_0 \equiv A_0/(4\pi R^2)$, $\overline{\sigma} \equiv 4\pi R^2\sigma/B_1$, $\overline{V} \equiv 3V/(4\pi R^3)$, $\overline{V}_0 \equiv 3V_0/(4\pi R^3)$, $\overline{p} \equiv 4\pi R^3 p/(3B_1)$, $\overline{Z} \equiv Z/R$, $\overline{Z}_0 \equiv Z_0/R$, $\overline{f} \equiv Rf/B_1$, and $\overline{f}_0 \equiv Rf_0/B_1$. We choose $R = R_0$, so that $\overline{R}_0 = \overline{A}_0 = 1$. We formulate the above two types of problems in term of these and previously defined dimensionless quantities, the result of which is given in Table 1 (Types I and II).

Constraints: membrane patches

In the case of a membrane patch S whose single pole is subject to an axial force of magnitude f (4), the full energy functional has the same form as Eq. 16,

$$F[S] = F'_b[S] + \sigma^0_{\text{eff}} A[S] - p_{\text{eff}} V[S] - fZ[S], \quad (17)$$

where F'_b is given by Eq. 14. The difference here is that A and V are not fixed. The larger membrane in contact with the membrane patch, acts as an area reservoir and a volume reservoir that give rise to a fixed effective membrane tension σ^0_{eff} (not to be confused with surface tension) and a fixed effective pressure difference p_{eff} across the membrane thickness, respectively. The membrane-tension term of Eq. 17 can be combined with the last term of Eq. 14 to give

$$F = F_b + \sigma_{\text{eff}} A - p_{\text{eff}} V - fZ, \quad (18)$$

where

$$\sigma_{\text{eff}} \equiv \sigma^0_{\text{eff}} + \frac{B_1 C_0^2}{2} \quad (19)$$

is the modified effective membrane tension that must be positive. The parameter C_0 is now absorbed into σ_{eff} and need not be specified. Equation 19 allows for a negative σ^0_{eff}, with the bound $\sigma^0_{\text{eff}} > -B_1 C_0^2/2$. The pressure term in

TABLE 1 Types of physical problems for which the algorithm of Fig. 4 can account, and the corresponding algorithm settings

Type I	Closed membrane with fixed $\overline{f}$.
Functional minimized	$W/B_1 = 8\pi\int_S \overline{H}^2 d\overline{A} + 4\pi\overline{B}_2[2\pi(\int_S \overline{H} d\overline{A})^2 - \overline{m}_0\int_S \overline{H} d\overline{A}] + \overline{\sigma}\overline{A} - \overline{p}\overline{V} - \overline{f}\ \overline{Z}$.
Physical parameters	$R = R_0, \overline{B}_2, \overline{m}_0, \overline{V}_0, \overline{f}_0$.
Physical constraints	$\overline{A} = 1, \overline{V} = \overline{V}_0, \overline{f} = \overline{f}_0$.
Settings	In line 1, set B_1 to some arbitrary positive value (simplest is 1), B_2 to $\overline{B}_2 B_1$, $\overline{m}_0$ to the given value, A_0 to some arbitrary positive value (simplest is 4π, so that $R_0 = 1$), V_0 to $4\pi R_0^3 \overline{V}_0/3$, and Z_0, R^0_p, and K_H to some arbitrary values (simplest is 0). In line 2, set σ and p to some finite initial values, and f to $f_0 = \overline{f}_0 B_1/R_0$. Turn on parts 2 and 3 to adjust σ and p, by setting *switch*2 and *switch*3 in lines 5 and 6 to 1. Turn off part 4 to keep f fixed, by setting *switch*4 in line 7 to 1.
Type II	Closed membrane with fixed $\overline{Z}$.
Functional minimized	Has same form as that of Type I.
Physical parameters	$R = R_0, \overline{B}_2, \overline{m}_0, \overline{V}_0, \overline{Z}_0$.
Physical constraints	$\overline{A} = 1, \overline{V} = \overline{V}_0, \overline{Z} = \overline{Z}_0$.
Settings	In line 1, set B_1, B_2, $\overline{m}_0$, A_0, V_0, R^0_p, and K_H as in Type I above, and set Z_0 to $\overline{Z}_0 R_0$. In line 2, set σ, p, and f to some finite initial values. Turn on parts 2–4 to adjust σ, p, and f, by setting *switch*2, *switch*3, and *switch*4 in lines 5–7 to 1.
Type III	Membrane patch with fixed $\overline{f}$.
Functional minimized	$W/B_1 = 8\pi\int_S \overline{H}^2 d\overline{A} + 4\pi\overline{B}_2[2\pi(\int_S \overline{H} d\overline{A}/\overline{R}_0)^2 - \overline{m}_0\int_S \overline{H} d\overline{A}/\overline{R}_0] + 2\pi\overline{A} - \overline{p}_{\text{eff}}\overline{V} - 2\pi\overline{f}'\overline{Z} + \overline{K}_H\overline{H}^2_p/2$.
Physical parameters	$R = R_t, \overline{B}_2, \overline{m}_0, \overline{R}_0, \overline{p}_0, \overline{f}'_0, \overline{R}^0_p$.
Computational parameter	$\overline{K}_H$.
Physical constraints	$\overline{p}_{\text{eff}} = \overline{p}_0, \overline{f}' = \overline{f}'_0, \overline{R}_p = \overline{R}^0_p, \overline{H}_p = 0$.
Settings	In line 1, set B_1 to some arbitrary positive value (simplest is 1), B_2 to $\overline{B}_2 B_1$, $\overline{m}_0$ to the given value, A_0 to $4\pi(\overline{R}_0 R_t)^2$, V_0 and Z_0 to some arbitrary values (simplest is 0), R^0_p to $\overline{R}^0_p R_t$, and K_H to $\overline{K}_H B_1 R_t^2$. In line 2, set the initial value of σ_{eff} to some arbitrary positive value (simplest is $0.5B_1$, so that $R_t = 1$), the initial value of p_{eff} to $3B_1\overline{p}_0/(4\pi R_t^3)$, and the initial value of f to $f_0 = 2\pi B_1\overline{f}'_0/R_t$. Turn off parts 2, 3, and 4 to keep σ_{eff}, p_{eff}, and f fixed, by setting *switch*2, *switch*3, and *switch*4 in lines 5–7 to 0.
Type IV	Membrane patch with fixed $\overline{Z}$.
Functional minimized	Has same form as that of Type III.
Physical parameters	$R = R_t, \overline{B}_2, \overline{m}_0, \overline{R}_0, \overline{p}_0, \overline{Z}_0$, and $\overline{R}^0_p$.
Computational parameter	$\overline{K}_H$.
Physical constraints	$\overline{p}_{\text{eff}} = \overline{p}_0, \overline{Z} = \overline{Z}_0, \overline{R}_p = \overline{R}^0_p, \overline{R}^0_p = 0$.
Settings	In line 1, set B_1, B_2, $\overline{m}_0$, A_0, V_0, R^0_p, and K_H as in Type III above, and set Z_0 to $\overline{Z}_0 R_t$. In line 2, set the initial value of σ_{eff} to some arbitrary positive value (simplest is $0.5B_1$, so that $R_t = 1$), the initial value of p_{eff} to $3B_1\overline{p}_0/(4\pi R_t^3)$, and f to some finite initial value. Turn off parts 2 and 3 to keep σ_{eff} and p_{eff} fixed, by setting *switch*2 and *switch*3 in lines 5 and 6 to 0. Turn on part 4 to adjust f, by setting *switch*4 in line 7 to 1.

Eqs. 17 and 18 has a negligible effect on F and can be omitted (4,8,9), equivalent to keeping p_{eff} fixed at zero. As in the case where S represents a closed membrane, either f or Z can be fixed. Therefore, we can also divide problems involving membrane patches into the fixed-f or the fixed-Z type.

Consideration of the patch boundary introduces two constraints. First, we keep A finite by constraining the cylindrical radius of the patch boundary, R_{p}, to some positive value R_{p}^0. Second, we ensure that there is no bending moment at the patch boundary by constraining H at the patch boundary, H_{p}, to zero (8,9).

In our numerical method, R_{p} is fixed at R_{p}^0 when we define the initial geometry of the membrane patch and we minimize the functional

$$\begin{aligned} W &= F + F_{\text{H}} \\ &= F_{\text{b}} + \sigma_{\text{eff}} A - p_{\text{eff}} V - fZ + \frac{K_{\text{H}}}{2} H_{\text{p}}^2 , \end{aligned} \tag{20}$$

where F is given by Eq. 18, F_{b} is given by Eq. 8, and

$$F_{\text{H}} \equiv \frac{K_{\text{H}}}{2} H_{\text{p}}^2 \tag{21}$$

is a fictitious energy term with the adjustable computational parameter K_{H}. We constrain H_{p} to 0 by setting the dimensionless $\overline{K}_{\text{H}} \equiv K_{\text{H}}/(B_1 R^2)$ to a large value.

It has been shown (4) that, in the situation where axial deformation of a membrane patch induces a stable cylindrical tube, the tube length L_{t}, the tube radius R_{t}, and the applied force f_{t} are related by

$$\sigma_{\text{eff}} = R_{\text{t}} p_{\text{eff}} + \frac{B_1}{2R_{\text{t}}^2} \tag{22}$$

and

$$f_{\text{t}} = \pi R_{\text{t}}^2 p_{\text{eff}} + 2\pi B_1 \left(\frac{1}{R_{\text{t}}} - C_0 \right) + \frac{2\pi^2 B_2}{A_0} \left(2\pi L_{\text{t}} - \frac{\Delta A_0}{D} \right). \tag{23}$$

Omitting the negligible effects of p_{eff} in Eq. 22 leads to the well-known law expressing the tube radius R_{t} in terms of membrane mechanical parameters B_1 and σ_{eff}:

$$R_{\text{t}} = \sqrt{\frac{B_1}{2\sigma_{\text{eff}}}}. \tag{24}$$

Moreover, omitting the negligible effects of p_{eff} in Eq. 23 gives

$$f_{\text{t}} = 2\pi B_1 \left(\frac{1}{R_{\text{t}}} - C_0 \right) + \frac{2\pi^2 B_2}{A_0} \left(2\pi L_{\text{t}} - \frac{\Delta A_0}{D} \right) \tag{25}$$

$$= f_{\text{t}}^0 + \frac{\pi B_2}{R_0} \left(\frac{\pi L_{\text{t}}}{R_0} - \overline{m}_0 \right) , \tag{26}$$

where the characteristic force

$$f_{\text{t}}^0 \equiv \frac{2\pi B_1}{R_{\text{t}}} = 2\pi \sqrt{2 B_1 \sigma_{\text{eff}}} \tag{27}$$

is notable for being independent of the tube length L_{t}, and being constant for given B_1 and σ_{eff}. The effect of C_0 on f_{t} becomes important when $C_0 R_{\text{t}}$ is ~1 or larger. The effect of L_{t} on f_{t} becomes important when

$$\frac{2\pi^2 B_2 L_{\text{t}} R_{\text{t}}}{A_0 B_1} = \left(\frac{\pi B_2 R_{\text{t}}^2}{2 B_1 R_0^2} \right) \frac{L_{\text{t}}}{R_{\text{t}}} \tag{28}$$

is ~1 or larger. The last term in the last bracket of Eq. 25 has a negligible effect on f_{t} and can be omitted. Thus, when $L_{\text{t}}/R_{\text{t}}$ is much smaller than $2B_1 {R_0}^2/(\pi B_2 {R_{\text{t}}}^2)$ (typically 10^3 to 10^4), Eq. 25 reduces to

$$f_{\text{t}} \approx 2\pi B_1 \left(\frac{1}{R_{\text{t}}} - C_0 \right). \tag{29}$$

However, in general, f_{t} is linearly dependent on the tube length L_{t}. Thus, in the case of a very long cylindrical tube extracted from a membrane with determinable A_0, the nonlocal bending modulus B_2 of the membrane can be calculated from the slope $4\pi^3 B_2/A_0$ of the plot of f_{t} against L_{t}.

We define p_0 to be the value at which p_{eff} is to be fixed and the dimensionless quantities

$$\overline{\sigma}_{\text{eff}} \equiv 4\pi R^2 \sigma_{\text{eff}}/B_1,$$

$$\overline{p}_{\text{eff}} \equiv 4\pi R^3 p_{\text{eff}}/(3B_1),$$

$$\overline{p}_0 \equiv 4\pi R^3 p_0/(3B_1),$$

$$\overline{f} \equiv Rf/(2\pi B_1),$$

$$\overline{f}_0 \equiv Rf_0/(2\pi B_1),$$

$$\overline{R}_{\text{p}} \equiv R_{\text{p}}/R,\ \overline{R}_{\text{p}}^0 \equiv R_{\text{p}}^0/R,$$

$$\overline{H}_{\text{p}} \equiv H_{\text{p}} R.$$

We set $R = R_{\text{t}}$, and choose f_{t}^0 as our unit of force, so that $\overline{\sigma}_{\text{eff}} = 2\pi$, $\overline{f} = f/f_{\text{t}}^0$, and $\overline{f}_0 = f_0/f_{\text{t}}^0$. We formulate membrane-patch problems of the fixed-f or the fixed-Z type in term of these and previously defined dimensionless quantities, the result of which is given in Table 1 (Types III and IV).

We present below our numerical method for computing the local minima of W and the corresponding locally stable contours $\{C_{\text{min}}\}$.

Discretization of the contour

There are two possible normals to contour C and we choose the one on the left of C when C is traversed clockwise. We approximate C by a two-dimensional chain $\tilde{C}$ made up of line elements $\{\tilde{C}_i |\ i = 1, 2, \ldots\}$ sequentially linked at their endpoints. We henceforth refer to the line elements and their endpoints as edges and vertices, respectively. There are N_{v} vertices and $N_{\text{v}} - 1$ edges. Each edge $\tilde{C}_i$ has a length $\tilde{l}_i$ and a constant unit normal $\tilde{\mathbf{n}}_i$. To account for boundary

conditions, we define a unit normal at each end of $\tilde{C}$. The one at the first vertex is denoted $\tilde{\mathbf{n}}_0$ and the one at the last vertex, $\tilde{\mathbf{n}}_{N_v}$. Thus, there are $N_v + 1$ unit normals in total. The angle $\tilde{\theta}_i$ between two neighboring unit normals $\tilde{\mathbf{n}}_{i-1}$ and $\tilde{W}$ is given by

$$\cos\tilde{\theta}_i = \tilde{\mathbf{n}}_{i-1} \cdot \tilde{\mathbf{n}}_i. \tag{30}$$

Note that $\tilde{\theta}_i$ is also the exterior angle at a vertex and it is computationally quicker (since one skips the calculation of unit normals) to calculate $\tilde{\theta}_i$ from the relation

$$\cos\tilde{\theta}_i = \tilde{\mathbf{l}}_{i-1} \cdot \tilde{\mathbf{l}}_i, \tag{31}$$

where $\tilde{\mathbf{l}}_i$ is the unit edge vector 90° clockwise from $\tilde{\mathbf{n}}_i$. The contour length L of C is approximated by the contour length of $\tilde{C}$,

$$\tilde{L} = \sum_{i=1}^{N_v-1} \tilde{l}_i \ . \tag{32}$$

The curvature $\kappa(\mathbf{r})$ along C is described in our discretization by the set of N_v angles $\{\tilde{\theta}_i\}$. The intensive quantity $|\tilde{\theta}_i|^{-1}$ is the scale-invariant vertex density (number per radian) along $\tilde{C}$. In our piecewise-linear approximation, $\tilde{C}$ is completely described by $\tilde{L}$ and $\{\tilde{\theta}_i\}$.

Boundary conditions

Axisymmetry dictates that the cylindrical radii of all vertices cannot be negative. C (and hence $\tilde{C}$) cannot physically self-intersect. Although obvious to us, we must account for these restrictions in the computer code.

If C represents a closed membrane with spherical topology, then the first and last vertices are both polar vertices. Axisymmetry dictates that they can only move along the cylindrical axis and that the polar normals $\tilde{\mathbf{n}}_0$ and $\tilde{\mathbf{n}}_{N_v}$ must lie on the cylindrical axis and point away from each other.

If C represents a patch of a closed membrane with spherical topology, then the first vertex is a polar vertex and the last vertex represents the patch boundary. Axisymmetry dictates that the first vertex can only move along the cylindrical axis and that the polar normal $\tilde{\mathbf{n}}_0$ must lie on the cylindrical axis. The patch boundary is fixed, so the last vertex has a fixed position. The slope near the patch boundary is such that $H_p = 0$ (8,9). In practice, we set the boundary normal $\tilde{\mathbf{n}}_{N_v}$ to some arbitrary initial direction and then use an energy-penalty method to force $\tilde{\mathbf{n}}_{N_v}$ to point in the direction that gives $H = 0$.

Discretization of the bending energy

Although $\tilde{C}$ is piecewise linear, it is possible to formulate $\tilde{F}_b$, the discretized form of F_b. As illustrated in Fig. 1, we define a circle with radius δ about each vertex $\tilde{\mathbf{r}}_i = (\tilde{z}_i, \tilde{x}_i)$ and we take the z axis to be the cylindrical axis. The circles are tangentially tethered to each other by line segments that become the edges $\{\tilde{C}_i\}$ in the limit $\delta \to 0$. Under rotation about the z axis, the circles give tori and one or two spheres, and the line segments give truncated cones. The number of spheres is two if C represents a closed membrane with spherical topology, or one if C represents a patch of a closed membrane with spherical topology. The tori and sphere(s) are partially hidden by the cones and these hidden sections are irrelevant to our formulation. The truncated cones and the exposed sections of the tori and sphere(s) contribute to $\tilde{F}_b$, as described below.

For a polar vertex, the associated spherical exposed section has integrated mean curvature $2\pi\delta(1 - \cos\tilde{\theta}_i)$ that reduces to

$$M_i^{(v)} = 0 \tag{33}$$

in the limit $\delta \to 0$. For any other vertex, the associated toroidal exposed section has integrated mean curvature $\pi(\delta\sin\tilde{\theta}_i' + \tilde{x}_i)\tilde{\theta}_i$, which reduces to

$$M_i^{(v)} = \pi\tilde{x}_i\tilde{\theta}_i, \tag{34}$$

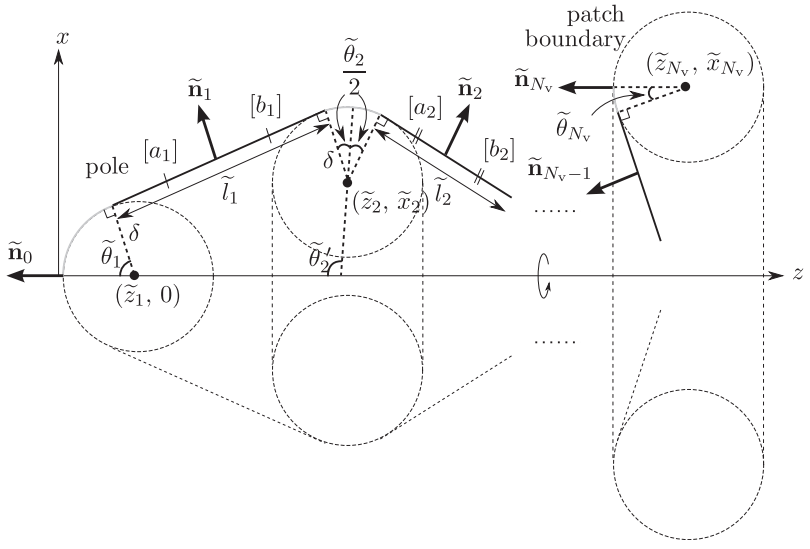

FIGURE 1 Illustration of the geometrical idea behind our formulation of $\tilde{F}_b$. The shaded arcs are outlines of the exposed sections of the tori and sphere(s). The solid lines are outlines of the truncated cones.

in the limit $\delta \to 0$, where $\tilde{\theta}_i$ is as defined in Fig. 1. Lastly, the truncated cone corresponding to an edge $\tilde{C}_i$ has integrated mean curvature

$$M_i^{(e)} = \pi\left(\tilde{z}_{i+1} - \tilde{z}_i\right). \tag{35}$$

Thus, I_H of Eq. 4 is approximated by

$$\tilde{I}_H \equiv \sum_{i=1}^{N_v} M_i^{(v)} + \sum_{j=1}^{N_v-1} M_j^{(e)}. \tag{36}$$

The midpoint of an edge gives a circle under cylindrical rotation. This circle divides the area

$$A_i = \pi \tilde{l}_i\left(\tilde{x}_i + \tilde{x}_{i+1}\right) \tag{37}$$

of the truncated cone on which the circle resides into two parts that are generally unequal. The part on the left has area fraction

$$a_i = \frac{\tilde{x}_i}{2(\tilde{x}_i + \tilde{x}_{i+1})} + \frac{1}{4}, \tag{38}$$

whereas that on the right has area fraction

$$b_i = 1 - a_i. \tag{39}$$

We associate each vertex with a local area A_i^{loc}, where

$$A_1^{loc} \equiv a_1 A_1, \tag{40}$$

$$A_i^{loc} \equiv b_{i-1}A_{i-1} + a_i A_i \tag{41}$$

for $i = 2, \ldots, N_v - 1$, and

$$A_{N_v}^{loc} \equiv b_{N_v-1}A_{N_v-1}, \tag{42}$$

and a local integrated mean curvature M_i, where

$$M_1 \equiv M_1^{(v)} + \frac{1}{2}M_1^{(e)}, \tag{43}$$

$$M_i \equiv M_i^{(v)} + \frac{1}{2}\left(M_{i-1}^{(e)} + M_i^{(e)}\right), \tag{44}$$

for $i = 2, \ldots, N_v - 1$, and

$$M_{N_v} \equiv M_{N_v}^{(v)} + \frac{1}{2}M_{N_v-1}^{(e)}. \tag{45}$$

Note that, if the first or last vertex is a polar vertex, then $M_1^{(v)}$ or $M_{N_v}^{(v)}$ vanishes according to Eq. 33. In analogy with the general situation in three dimensions (10–12), we define the approximate local integrated mean-curvature squared,

$$\tilde{I}_{HH}^{(i)} \equiv \frac{M_i^2}{A_i^{loc}}, \tag{46}$$

and approximate I_{HH} of Eq. 3 by

$$\tilde{I}_{HH} \equiv \sum_{i=1}^{N_v} \tilde{I}_{HH}^{(i)}. \tag{47}$$

It is computationally convenient to treat $\tilde{F}_b$ as a sum of contributions from individual vertices, $\tilde{F}_b \equiv \sum_{i=1}^{N_v} \tilde{F}_b^{(i)}$, where

$$\tilde{F}_b^{(i)} \equiv 2\tilde{I}_{HH}^{(i)} + \frac{\overline{B}_2}{2}\left(\frac{\tilde{I}_H}{R_0} - 2\overline{m}_0\right)\frac{M_i}{R_0}. \tag{48}$$

Note that $\tilde{F}_b^{(i)}$ is nonlocal when $\overline{B}_2 \neq 0$ and $\tilde{I}_H/R_0 \neq 2\overline{m}_0$, because it contains the product $\tilde{I}_H M_i$.

Discretization errors

The dependence of F_b on I_H and I_{HH} gives rise to two sources of error in the discretization of F_b to $\tilde{F}_b$. We define the discretization errors for I_H, I_{HH}, and F_b to be

$$\delta I_H \equiv \tilde{I}_H - I_H, \tag{49}$$

$$\delta I_{HH} \equiv \tilde{I}_{HH} - I_{HH}, \tag{50}$$

and

$$\delta F_b \equiv \tilde{F}_b - F_b, \tag{51}$$

respectively. Substituting these three equations into Eq. 8 yields δF_b as a function of δI_H and δI_{HH}:

$$\frac{\delta F_b}{B_1} = 2\delta I_{HH} + \overline{B}_2\left[\left(\frac{\tilde{I}_H}{R_0} - \overline{m}_0\right)\frac{\delta I_H}{R_0} - \frac{1}{2}\left(\frac{\delta I_H}{R_0}\right)^2\right]. \tag{52}$$

In the usual situation where $N_v >> 1$, the last term in the square brackets in Eq. 52 is negligible and δF_b is effectively a linear function of δI_H and δI_{HH}.

If F_b, I_H, and I_{HH} are nonzero, then we can define the fractional discretization errors $\delta I_H/I_H$, $\delta I_{HH}/I_{HH}$, and

$$\frac{\delta F_b}{F_b} = \frac{\delta I_{HH} + \frac{\overline{B}_2}{2}\left(\frac{\tilde{I}_H}{R_0} - \overline{m}_0\right)\frac{\delta I_H}{R_0}}{I_{HH} + \frac{\overline{B}_2}{4}\left(\frac{I_H}{R_0} - 2\overline{m}_0\right)\frac{I_H}{R_0}}, \tag{53}$$

where $I_{HH} \geq 0$ and $I_H^2 \geq 0$. Equation 53 has four limits to note:

1. In the simple case of $\overline{B}_2 = 0$, $\delta F_b/F_b = \delta I_{HH}/I_{HH}$.
2. If $R_0 \to \infty$, then $\delta F_b/F_b \approx \delta I_{HH}/I_{HH}$.
3. If the term proportional to $(I_H/R_0)^2$ in the denominator is dominant ($I_H/R_0 > 2\overline{m}_0$), then $\delta F_b/F_b \approx 2\delta I_H/I_H$.
4. If the term proportional to $\overline{m}_0$ in the denominator is dominant, then $\delta F_b/F_b \approx \delta I_H/I_H$.

We write the discretization errors in terms of the contributions from individual vertices,

$$\delta I_H = \sum_{i=1}^{N_v} \delta I_H^{(i)},$$

$$\delta I_{\mathrm{HH}} = \sum_{\mathrm{i}=1}^{N_\mathrm{v}} \delta I_{\mathrm{HH}}^{(\mathrm{i})},$$

and

$$\delta F_\mathrm{b} = \sum_{\mathrm{i}=1}^{N_\mathrm{v}} \delta F_\mathrm{b}^{(\mathrm{i})} = \sum_{\mathrm{i}=1}^{N_\mathrm{v}} \frac{\delta F_\mathrm{b}^{(\mathrm{i})}}{\tilde{F}_\mathrm{b}^{(\mathrm{i})}} \tilde{F}_\mathrm{b}^{(\mathrm{i})}, \tag{54}$$

where $\tilde{F}_\mathrm{b}^{(\mathrm{i})}$ is given by Eq. 48 and $\delta F_\mathrm{b}{}^{(\mathrm{i})}$ is related to $\delta I_\mathrm{H}{}^{(\mathrm{i})}$ and $\delta I_{\mathrm{HH}}^{(\mathrm{i})}$ by

$$\frac{\delta F_\mathrm{b}^{(\mathrm{i})}}{B_1} = 2\delta I_{\mathrm{HH}}^{(\mathrm{i})} + \overline{B}_2 \left(\frac{\tilde{I}_\mathrm{H}}{R_0} - \overline{m}_0 \right) \frac{\delta I_\mathrm{H}^{(\mathrm{i})}}{R_0}. \tag{55}$$

The ratio $\delta F_\mathrm{b}^{(\mathrm{i})}/\tilde{F}_\mathrm{b}^{(\mathrm{i})}$ in Eq. 54 is a local fractional error. According to Eq. 54, a vertex can give a relatively-large local error (relatively-large $\delta F_\mathrm{b}^{(\mathrm{i})}/\tilde{F}_\mathrm{b}^{(\mathrm{i})}$) and still have an insignificant overall effect on δF_b if $\tilde{F}_\mathrm{b}^{(\mathrm{i})}$ itself is small. This applies for any region of $\tilde{C}$ with small cylindrical radii (such as the neighborhood of a pole and a kissing neck), the small area of which leads to a small $\tilde{F}_\mathrm{b}^{(\mathrm{i})}$.

We consider the discretization of a spherical S with $\overline{B}_2 = \overline{m}_0 = 1$. Insight gained from this case is later applied to the dynamic regridding of a general S. Since $I_{\mathrm{HH}} = I_\mathrm{H}/R_0 = 4\pi > 2\overline{m}_0 = 2$, we expect $\delta F_\mathrm{b}/F_\mathrm{b} \approx 2\delta I_\mathrm{H}/I_\mathrm{H}$. We approximate the corresponding half-circle C by equilateral contours with $N_\mathrm{v} - 1 = 20, 30, 40, \ldots, 100, 120, 140, \ldots, 320$ edges. The angle subtended at the center of C by the equilateral edges of a contour is

$$\tilde{\theta} = \frac{\pi}{N_\mathrm{v} - 1}. \tag{56}$$

The vertex density for each contour is $|\tilde{\theta}_\mathrm{i}|^{-1} = \alpha |\tilde{\theta}|^{-1}$, where $\alpha = 2$ for $i = 1$ and N_v, and $\alpha = 1$ otherwise. We constrain the areas of the surfaces of revolution for these contours to the area of S. Therefore, the vertices of each contour lie on a half-circle slightly larger in radius than C. In Fig. 2 *a*, we plot the fractional errors, $\delta I_\mathrm{H}/I_\mathrm{H}$, $\delta I_{\mathrm{HH}}/I_{\mathrm{HH}}$, and $\delta F_\mathrm{b}/F_\mathrm{b}$, against $\tilde{\theta}^2 = \pi^2/(N_\mathrm{v} - 1)^2$. In Fig. 2 *b*, we plot the slopes of the fractional errors, $(\delta I_\mathrm{H}/I_\mathrm{H})'$, $(\delta I_{\mathrm{HH}}/I_{\mathrm{HH}})'$, and $(\delta F_\mathrm{b}/F_\mathrm{b})'$, as functions of $\tilde{\theta}^2$. These plots show clearly that the fractional errors are effectively proportional to $1/(N_\mathrm{v} - 1)^2$. Both $\delta I_\mathrm{H}/I_\mathrm{H}$ and $\delta I_{\mathrm{HH}}/I_{\mathrm{HH}}$ are positive, indicating that $\tilde{I}_H$ overestimates I_H, and $\tilde{I}_{\mathrm{HH}}$ overestimates I_{HH}. Fig. 2 *a* confirms that $\delta F_\mathrm{b}/F_\mathrm{b} \approx 2\delta I_\mathrm{H}/I_\mathrm{H}$, and since $\delta I_\mathrm{H}/I_\mathrm{H} > 0$, therefore $\delta F_\mathrm{b}/F_\mathrm{b} > 0$. To reduce the fractional errors by an order of magnitude, we must decrease the angle per vertex $|\tilde{\theta}_\mathrm{i}|$ by a factor of $\sqrt{10}$. The fractional errors are $\sim 10^{-3}$ for the roughest discretization $|\tilde{\theta}_\mathrm{i}| = \pi/(20\alpha) = 9^\circ/\alpha$. Geometrically decreasing them to $10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}, \ldots$ requires $|\tilde{\theta}_\mathrm{i}|$ to decrease to $2.85^\circ/\alpha$, $0.9^\circ/\alpha$, $0.285^\circ/\alpha$, $0.09^\circ/\alpha$, etc.

In the case of a general $\tilde{C}$ with different edge lengths and different vertex angles, and subject to a finite computational temperature T (not the physical temperature, since temperature does not enter into an equilibrium mechanical problem),

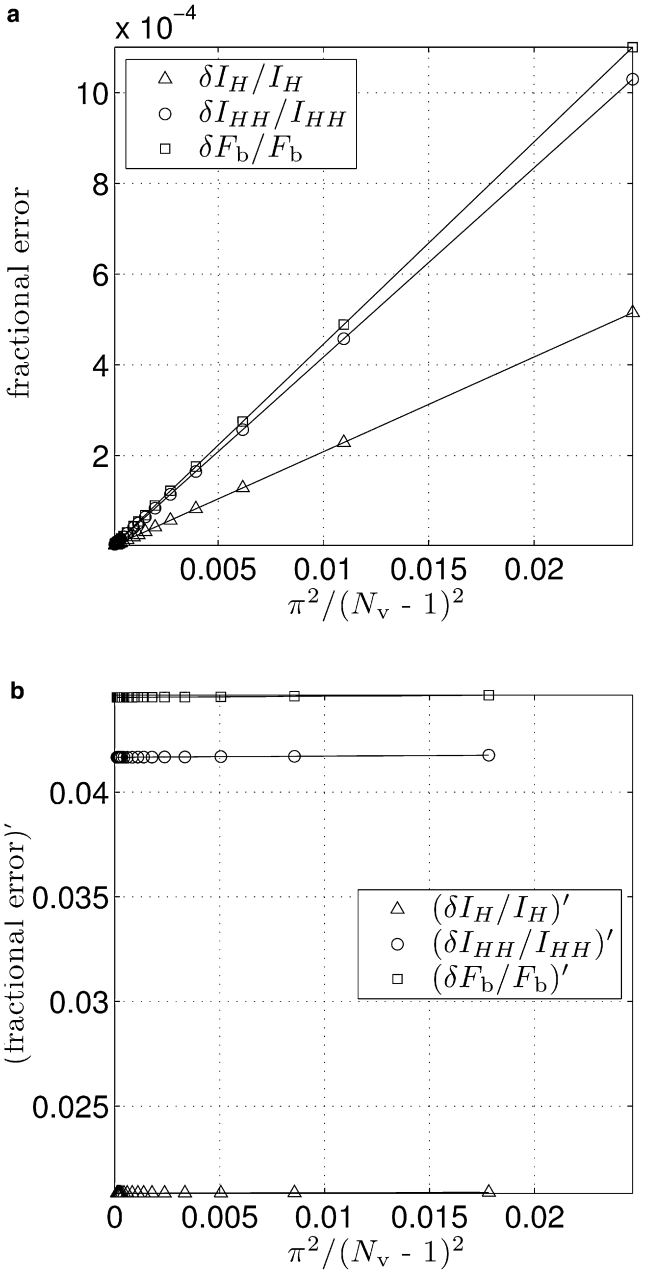

FIGURE 2 (*a*) Plots of $\delta I_\mathrm{H}/I_\mathrm{H}$, $\delta I_{\mathrm{HH}}/I_{\mathrm{HH}}$, and $\delta F_\mathrm{b}/F_\mathrm{b}$, against $\pi^2/(N_\mathrm{v} - 1)^2$. (*b*) Plots of the slopes of the curves in panel *a*, $(\delta I_\mathrm{H}/I_\mathrm{H})'$, $(\delta I_{\mathrm{HH}}/I_{\mathrm{HH}})'$, and $(\delta F_\mathrm{b}/F_\mathrm{b})'$, against $\pi^2/(N_\mathrm{v} - 1)^2$.

we regulate $\delta F_\mathrm{b}^{(\mathrm{i})}/\tilde{F}_\mathrm{b}^{(\mathrm{i})}$ of Eq. 54 based on the above relationship between $\delta F_\mathrm{b}/F_\mathrm{b}$ and $|\tilde{\theta}_\mathrm{i}|$. Specifically, we implement dynamic regridding of $\tilde{C}$ and set its two angular tolerances $\theta_{\max}$ and $\theta_{\min}$ (to be discussed later) so that $|\delta F_\mathrm{b}^{(\mathrm{i})}/\tilde{F}_\mathrm{b}^{(\mathrm{i})}| \lesssim 10^{-4}$ as $B_1/T \to \infty$.

Discretization of the fictitious energy

We approximate H_p by

$$\tilde{H}_\mathrm{p} \equiv \frac{M_{N_\mathrm{v}}}{A_{N_\mathrm{v}}^{\mathrm{loc}}}, \tag{57}$$

where M_{N_v} and $A^{loc}_{N_v}$ are given by Eqs. 45 and 42, respectively. Thus, we approximate F_H of Eq. 21 by

$$\tilde{F}_H = \frac{K_H}{2}\tilde{H}_p^2, \tag{58}$$

$$= \frac{K_H}{2}\frac{\tilde{I}^{(N_v)}_{HH}}{A^{loc}_{N_v}}, \tag{59}$$

where we have made use of Eq. 46.

Discretization of geometric quantities

We approximate A, V, and Z by

$$\tilde{A} \equiv \sum_{i=1}^{N_v-1} \pi \tilde{l}_i (\tilde{x}_i + \tilde{x}_{i+1}), \tag{60}$$

$$\tilde{V} \equiv \sum_{i=1}^{N_v-1} \frac{\pi}{3}(\tilde{z}_{i+1} - \tilde{z}_i)(\tilde{x}_i^2 + \tilde{x}_{i+1}^2 + \tilde{x}_i\tilde{x}_{i+1}), \tag{61}$$

and

$$\tilde{Z} \equiv \tilde{z}_{N_v} - \tilde{z}_1, \tag{62}$$

respectively. Note that $\tilde{Z}$ depends only on the first and last vertices.

Variation of the contour

The contour $\tilde{C}$ is described by $\tilde{L}$ and $\{\tilde{\theta}_i\}$. We use simulated annealing to determine $\tilde{L}$ and $\{\tilde{\theta}_i\}$ for $\tilde{C}_{min}$, the approximation to C_{min}. We take advantage of the property of simulated annealing that, at very low computational temperatures, the system automatically seeks out the right $\tilde{L}$ and $\{\tilde{\theta}_i\}$ as it evolves toward an energy minimum. We need to use a probabilistic (Monte Carlo) method to vary $\tilde{L}$ and $\{\tilde{\theta}_i\}$, and we develop a method based on the Metropolis algorithm (13).

The salient structure of our computer program is outlined graphically in Fig. 3 and in pseudocode form in Fig. 4. The eight main parts are lines 14 (part 1); 15–17 (part 2); 18–20 (part 3); 21–23 (part 4); 24 (part 5); 25 (part 6); 26 (part 7); and 28–31 (part 8) of Fig. 4. Parts 1 and 5–8 are always on. We choose which of parts 2–4 to run, based on the constraints under consideration, as described in Table 1. The main computational parameters and their values are listed in Table 2. (Please also see Fig. 5.)

The standard Metropolis algorithm involves the basic step of performing a vertex trial move, where a vertex is displaced in a random direction. In part 1 of our program, we implement Monte Carlo sweeps (a sweep is a pass through every vertex once), with modifications to the trial moves out of the necessity of accounting for boundary conditions and numerical accuracy. Our routine for a vertex trial move is outlined in Figs. 6 and 7. First, we preserve axisymmetry by constraining a polar vertex to move along the cylindrical axis (line 15 in Fig. 7) and forbidding all interior vertices from crossing the cylindrical axis (preventing their cylindrical radii from becoming negative in line 2 in Fig. 7). Second, in the case of a membrane patch, we randomly rotate the boundary normal $\tilde{\mathbf{n}}_{N_v}$ about the boundary vertex representing the patch boundary (line 23 in Fig. 7), in lieu of randomly displacing the boundary vertex. Third, we implement strict local self-avoidance and partial global self-avoidance of $\tilde{C}$ by imposing the restriction $|\tilde{\theta}_i| < \theta_{lim}$ (lines 6–8, 19–21,

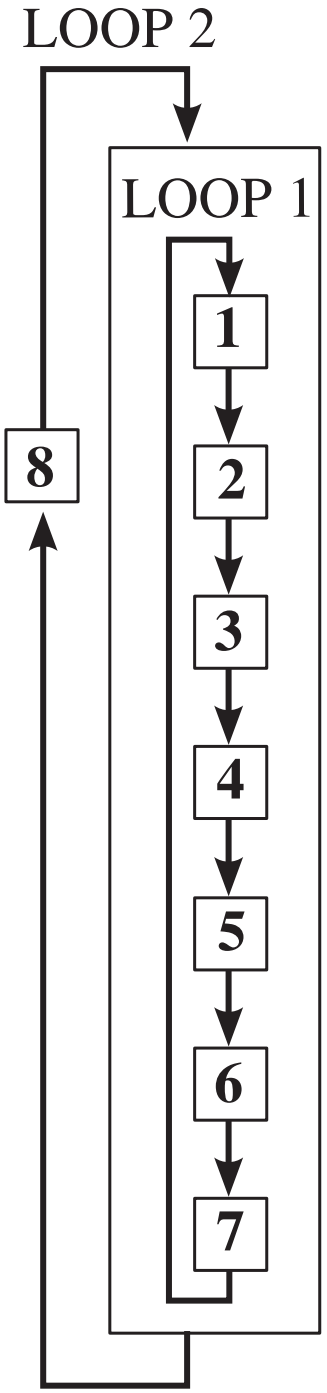

FIGURE 3 Graphical outline of the flow through the eight main parts of the algorithm described in Fig. 4.

```
1  set B_1, B_2, m̄_0, A_0, V_0, Z_0, R_p^0, and K_H;
2  set initial values of σ (or σ_eff), p (or p_eff), and f;
3  σ^AL ← σ or σ_eff; p^AL ← −p or −p_eff; f^AL ← −f;
4  ⟨Ã⟩_prev ← Ã; ⟨Ṽ⟩_prev ← Ṽ; ⟨Z̃⟩_prev ← Z̃;
5  set Switch2 to 1 or 0; /* turn part 2 on (1) or off (0) */
6  set Switch3 to 1 or 0; /* turn part 3 on (1) or off (0) */
7  set Switch4 to 1 or 0; /* turn part 4 on (1) or off (0) */
8  set the initial value of T;
9  while the end of the cooling schedule has not been reached do
10     for index2 ← 1 to n_2 do /* LOOP 2 */
11         report that L̃^ref is not changed;
12         ⟨θ̃_i⟩ ← 0, i = 1, ..., N_v;
13         for index1 ← 1 to n_1 do /* LOOP 1 */
14             do n_loc = 100 Monte Carlo sweeps to evolve C̃, generate
               local averages ⟨Ã⟩, ⟨Ṽ⟩, and ⟨Z̃⟩, and obtain contribution
               to ⟨θ̃_i⟩; /* PART 1, see Fig. 7 */
15             if Switch2 = 1 then /* PART 2 */
16                 adjust σ or σ_eff based on ⟨Ã⟩ and ⟨Ã⟩_prev; /* see
                   Fig. 5 */
17             end
18             if Switch3 = 1 then /* PART 3 */
19                 adjust p or p_eff based on ⟨Ṽ⟩ and ⟨Ṽ⟩_prev; /* see
                   Fig. 5 */
20             end
21             if Switch4 = 1 then /* PART 4 */
22                 adjust f based on ⟨Z̃⟩ and ⟨Z̃⟩_prev; /* see Fig. 5 */
23             end
24             ⟨Ã⟩_prev ← ⟨Ã⟩; ⟨Ṽ⟩_prev ← ⟨Ṽ⟩; ⟨Z̃⟩_prev ← ⟨Z̃⟩; /* PART 5
               */
25             update the maximum step size for each vertex; /* PART 6
               */
26             modify L̃^ref if necessary; /* PART 7, see Fig. 8 */
27         end
28         if L̃^ref is not changed then /* PART 8 */
29             determine ⟨θ̃_i⟩, i = 1, ..., N_v;
30             attempt to add or remove vertices; /* see Fig. 9 */
31         end
32     end
33     lower T;
34 end
```

FIGURE 4 Pseudocode for the algorithm, with eight main parts that are discussed in the text. Part 1: line 14. Part 2: lines 15–17. Part 3: lines 18–20. Part 4: lines 21–23. Part 5: line 24. Part 6: line 25. Part 7: line 26. Part 8: lines 28–31.

and 24–25 in Fig. 7). To save computation time, we have not implemented strict self-avoidance of $\tilde{C}$.

There are four considerations in setting the value of θ_{lim}. First, $|\tilde{\theta}_i|$ must be $<180°$. The physical reason is that the two associated edges intersect when $|\tilde{\theta}_i|$ is a multiple of 180°. The technical reason is that $\tilde{C}$ develops numerically ill-behaved spikes when $|\tilde{\theta}_i| > 180°$. (Computer implementation of the arc cosine function only gives angles within the range [0°, 180°], leading to a modulo-180° problem in the calculation of $\tilde{\theta}_i$, that often occurs when $\tilde{C}$ has to evolve toward a very different final configuration.) Second, lowering θ_{lim} reduces the chances of $\tilde{C}$ evolving to a nonphysical final configuration, where $\tilde{C}$ loops in on itself one or more times to self-intersect nonlocally (this typically occurs for a $\tilde{C}$ that has to relax to a very different, physical final configuration). Third, θ_{lim} must be as large as possible to allow $\tilde{C}$ to relax quickly. Fourth, θ_{lim} must be substantially greater than the upper threshold θ_{max} for the absolute average

TABLE 2 Main computational parameters, their values, and their roles

Parameter	Value	Role
n_{loc}	100	Looping.
n_1	Generally set to 50.	
n_2	~400 or more, depending upon relaxation rate.	
Vertex trial move success rate	$(40 \pm 2)\%$.	Adjusting the maximum step size for each vertex.
θ_{lim}	80°.	Imposition of local self-avoidance, ensuring orderly evolution of nonrelaxed $\tilde{C}$.
ε_1	0.1.	Allowing small variations of the $N_{\mathrm{v}} - 1$ edge lengths $\tilde{l}_{\mathrm{i}}$.
T	Lowered until the mean minimized $\tilde{W}$ converges to a constant at $B_1/T >> 1$.	Adjusting the extensiveness of exploration of parameter space.
K_{H}	Must be $>> B_1 R_{\mathrm{t}}^2$.	Enforcing the membrane-patch boundary condition $H_{\mathrm{p}} = 0$.
ε_2	0.095, must be slightly $<\varepsilon_1$.	Increase or decrease $\tilde{L}$ until its permitted range includes the minimum-energy $\tilde{L}$.
ε_3	If the starting and final $\tilde{C}$ are far apart in parameter space: 0.05 and then 0; 0 otherwise.	Adding or removing vertices.
θ_{max}	If the starting and final $\tilde{C}$ are far apart in parameter space: 45° and then 5°; 5° otherwise.	
ε_4	0.025.	
θ_{min}	0.25°, must be $<<\theta_{\mathrm{max}}$.	

$|\langle\tilde{\theta}_{\mathrm{i}}\rangle|$ of $\tilde{\theta}_{\mathrm{i}}$. As described in detail later, the algorithm adds vertices to high-curvature regions of $\tilde{C}$ until $|\langle\tilde{\theta}_{\mathrm{i}}\rangle| \lesssim \theta_{\mathrm{max}}$. The requirement that θ_{lim} be substantially larger than θ_{max} eventually leads to $|\tilde{\theta}_{\mathrm{i}}| \lesssim \theta_{\mathrm{max}} \ll \theta_{\mathrm{lim}}$, at which point the effect of the somewhat artificial limit θ_{lim} is removed, since every $\tilde{\theta}_{\mathrm{i}}$ in a minimization run is then so small that no trial moves fail on account of $|\tilde{\theta}_{\mathrm{i}}|$ exceeding θ_{lim}. After some experimentation, we set θ_{lim} to 80°.

Bending-energy models of the ADE type are parameterization-invariant. This property enables vertices to drift freely in the tangential direction, since such movements constitute a reparameterization of the contour $\tilde{C}$ and, thus, do not cost energy. Unrestricted tangential motions of the vertices will lead to extreme differences in edge lengths, with some edges becoming so long that they no longer approximate C locally, an undesirable effect that introduces numerical inaccuracies. We ensure that the edges are short and, therefore, accurate local approximations of C by

1. Defining a small reference length $\tilde{l}_{\mathrm{i}}^{\mathrm{ref}}$ for each edge, which satisfies the condition $\tilde{l}_{\mathrm{i}}^{\mathrm{ref}} \ll \tilde{L}^{\mathrm{ref}} \equiv \sum_{\mathrm{j}=1}^{N_{\mathrm{v}}-1} \tilde{l}_{\mathrm{j}}^{\mathrm{ref}}$.
2. Allowing $\tilde{l}_{\mathrm{i}}$ to increase or decrease in a vertex trial move as long as both its old and new values are in the narrow range $[(1-\varepsilon_1)\tilde{l}_{\mathrm{i}}^{\mathrm{ref}}, (1+\varepsilon_1)\tilde{l}_{\mathrm{i}}^{\mathrm{ref}}]$ (lines 3–5 and 16–18 in Fig. 7), where the maximum fractional length change, ε_1, must satisfy $0 < \varepsilon_1 < 1$ and is set to 0.1.

The edge reference lengths $\{\tilde{l}_{\mathrm{i}}^{\mathrm{ref}}\}$ are used exclusively as a mathematical device to maintain numerical accuracy and have no physical significance.

Lowering or raising ε_1 too much will prolong the computation. Too small an ε_1 leads to a very frustrated $\tilde{C}$ that takes a much longer time to relax at a given T. Too large an ε_1 gives rise to large-scale fluctuations of $\tilde{C}$ and forces us to extend the annealing procedure to much lower values of T. We experimented with different values of ε_1 and find that $\varepsilon_1 = 0.1$ allows $\tilde{C}$ just enough fluctuations to relax at a relatively quick rate.

In general, a vertex trial move leads to the following changes:

$$\tilde{F}_{\mathrm{b}} \rightarrow \tilde{F}_{\mathrm{b}} + \Delta\tilde{F}_{\mathrm{b}}, \tag{63}$$

$$\tilde{A} \rightarrow \tilde{A} + \Delta\tilde{A}\,, \tag{64}$$

$$\tilde{V} \rightarrow \tilde{V} + \Delta\tilde{V}, \tag{65}$$

$$\tilde{Z} \rightarrow \tilde{Z} + \Delta\tilde{Z}, \tag{66}$$

$$\tilde{F}_{\mathrm{H}} \rightarrow \tilde{F}_{\mathrm{H}} + \Delta\tilde{F}_{\mathrm{H}}, \tag{67}$$

and

$$\tilde{W} \rightarrow \tilde{W} + \Delta\tilde{W}, \tag{68}$$

The Tethered Infinitesimal Tori and Spheres Algorithm

1 $ConvergingFromLeft = \langle G\rangle_{\mathrm{prev}} < \langle G\rangle < G_0$;
2 $ConvergingFromRight = \langle G\rangle_{\mathrm{prev}} > \langle G\rangle > G_0$;
3 $NoAdjustment = ConvergingFromLeft$ or $ConvergingFromRight$;
4 **if** *not NoAdjustment* **then** $\gamma \leftarrow \gamma + K_G^{\mathrm{AL}}\left[\langle G\rangle - G_0\right]$;

FIGURE 5 Probabilistic augmented Lagrangian method for adjusting γ, which represents σ^{AL}, p^{AL}, or f^{AL}. If $\gamma = \sigma^{\mathrm{AL}}$, $K_{\mathrm{G}}^{\mathrm{AL}} = K_{\mathrm{A}}^{\mathrm{AL}}$, $\langle G\rangle = \langle\tilde{A}\rangle$, $\langle G\rangle_{\mathrm{prev}} = \langle\tilde{A}\rangle_{\mathrm{prev}}$, and, $G_0 = A_0$. If $\gamma = p^{\mathrm{AL}}$, $K_{\mathrm{G}}^{\mathrm{AL}} = K_{\mathrm{V}}^{\mathrm{AL}}$, $\langle G\rangle = \langle\tilde{V}\rangle$, $\langle G\rangle_{\mathrm{prev}} = \langle\tilde{V}\rangle_{\mathrm{prev}}$, and, $G_0 = V_0$. If $\gamma = f^{\mathrm{AL}}$, $K_{\mathrm{G}}^{\mathrm{AL}} = K_{\mathrm{Z}}^{\mathrm{AL}}$, $\langle G\rangle = \langle\tilde{Z}\rangle$, $\langle G\rangle_{\mathrm{prev}} = \langle\tilde{Z}\rangle_{\mathrm{prev}}$, and, $G_0 = Z_0$.

```
1  switch vertex position do
2      case first vertex /* first, permanent pole of C̃ */
3          do lines 15–22 of Fig. 7(b), with relevant variables
           l̃_i^new = l̃_1^new, θ̃_i^new = θ̃_1^new, and θ̃_{i+1}^new = θ̃_2^new, and ΔW̃ = ΔF̃_b
           + σΔÃ − pΔṼ − fΔZ̃;
4      case second vertex /* first interior vertex, next to
       first pole */
5          do lines 1–14 of Fig. 7(a), with the polar angle θ̃_{i−1}^new = θ̃_1^new
           calculated differently from θ̃_2^new and θ̃_3^new, and ΔW̃ = ΔF̃_b +
           σΔÃ − pΔṼ;
6      case last vertex
7          switch boundary represented by last vertex do
8              case pole /* second pole of closed membrane */
9                  do lines 15–22 of Fig. 7(b), with relevant variables
                   l̃_{i−1}^new = l̃_{N_v−1}^new, θ̃_i^new = θ̃_{N_v}^new, and θ̃_{i−1}^new = θ̃_{N_v−1}^new, and
                   ΔW̃ = ΔF̃_b + σΔÃ − pΔṼ − fΔZ̃;
10             otherwise /* boundary vertex of membrane patch */
11                 do lines 23–25 of Fig. 7(c), with ΔW̃ = ΔF̃_b + ΔF̃_H;
12             end
13         end
14     case second-last vertex /* last interior vertex */
15         switch boundary represented by last vertex do
16             case pole /* next to second pole */
17                 do lines 1–14 of Fig. 7(a), with the polar angle
                   θ̃_{i+1}^new = θ̃_{N_v}^new calculated differently from θ̃_{N_v−1}^new and
                   θ̃_{N_v−2}^new, and ΔW̃ = ΔF̃_b + σΔÃ − pΔṼ;
18             otherwise /* next to patch-boundary vertex */
19                 do lines 1–14 of Fig. 7(a), with the patch-boundary
                   angle θ̃_{i+1}^new = θ̃_{N_v}^new calculated differently from θ̃_{N_v−1}^new
                   and θ̃_{N_v−2}^new, and ΔW̃ = ΔF̃_b + σΔÃ − pΔṼ + ΔF̃_H;
20             end
21         end
22     otherwise /* all other, interior vertices */
23         do lines 1–14 of Fig. 7(a), with ΔW̃ = ΔF̃_b + σΔÃ − pΔṼ;
24     end
25 end
```

FIGURE 6 Outer structure of the pseudocode for a vertex trial move. We use σ and σ_{eff}, and p and p_{eff}, interchangeably in $\Delta\tilde{W}$.

where $\tilde{W}$ is the discretized form of W. The change in $\tilde{W}$ between the new and old configurations of $\tilde{C}$, $\Delta\tilde{W}$, is given by

$$\Delta\tilde{W} = \Delta\tilde{F}_{\text{b}} + \sigma\Delta\tilde{A} - p\Delta\tilde{V} - f\Delta\tilde{Z} + \Delta\tilde{F}_{\text{H}}\,, \tag{69}$$

where we use σ and σ_{eff}, and p and p_{eff}, interchangeably, and

$$\sigma \text{ or } \sigma_{\text{eff}} = \sigma^{\text{AL}} + K_A^{\text{AL}}\big(\tilde{A} - A_0\big), \tag{70}$$

$$-p \text{ or } -p_{\text{eff}} = p^{\text{AL}} + K_{\text{V}}^{\text{AL}}\big(\tilde{V} - V_0\big), \tag{71}$$

and

$$-f = f^{\text{AL}} + K_{\text{Z}}^{\text{AL}}\big(\tilde{Z} - Z_0\big)\,. \tag{72}$$

(The superscript *AL* denotes ''augmented Lagrangian'' (14).) The three computational parameters σ^{AL}, p^{AL}, and f^{AL} are automatically set initially by the algorithm to the initial values of σ or σ_{eff}, $-p$ or $-p_{\text{eff}}$, and $-f$, respectively (line 3 in Fig. 4), and are to be adjusted iteratively in parts 2–4 of the algorithm. The other three computational parameters, K_{A}^{AL}, K_{V}^{AL}, and K_{Z}^{AL}, are set automatically by the algorithm according to the constraints under consideration:

$$K_{\text{A}}^{\text{AL}} = \mathit{Switch2} \times \frac{\tilde{L}^{\text{ref}} B_1}{\min\big(\tilde{l}_1^{\text{ref}},\, \tilde{l}_{N_{\text{v}}-1}^{\text{ref}}\big)} \quad \text{per unit area squared,} \tag{73}$$

The Tethered Infinitesimal Tori and Spheres Algorithm

```
   /* Part (a):  interior vertex */
 1 randomly displace the vertex (z̃_i, x̃_i) to (z̃_i^new, x̃_i^new);
 2 if x̃_i^new > 0 then
 3 |   calculate the two new edge lengths l̃_{i-1}^new and l̃_i^new;
 4 |   EdgeOK = (1-ε1) l̃_{i-1}^ref ≤ l̃_{i-1}^new ≤ (1+ε1) l̃_{i-1}^ref and
   |   (1-ε1) l̃_i^ref ≤ l̃_i^new ≤ (1+ε1) l̃_i^ref;
 5 |   if EdgeOK then
 6 |   |   calculate the three new angles θ̃_{i-1}^new, θ̃_i^new, and θ̃_{i+1}^new;
 7 |   |   AngleOK = |θ̃_{i-1}^new| < θ_lim and |θ̃_i^new| < θ_lim and
   |   |   |θ̃_{i+1}^new| < θ_lim;
 8 |   |   if AngleOK then
 9 |   |   |   calculate ΔW̃;
10 |   |   |   MoveOK = exp(-ΔW̃/T) > a random number between 0
   |   |   |   and 1;
11 |   |   |   if MoveOK then update computational variables;
12 |   |   end
13 |   end
14 end

   /* Part (b):  polar vertex */
15 randomly displace the vertex (z̃_i, 0) to (z̃_i^new, 0);
16 calculate the new edge length l̃_{i-1}^new or l̃_i^new;
17 EdgeOK = (1-ε1) l̃_{i-1}^ref ≤ l̃_{i-1}^new ≤ (1+ε1) l̃_{i-1}^ref or
   (1-ε1) l̃_i^ref ≤ l̃_i^new ≤ (1+ε1) l̃_i^ref;
18 if EdgeOK then
19 |   calculate the two new angles θ̃_i^new and either θ̃_{i-1}^new or θ̃_{i+1}^new;
20 |   AngleOK = |θ̃_i^new| < θ_lim and either |θ̃_{i-1}^new| < θ_lim or
   |   |θ̃_{i+1}^new| < θ_lim;
21 |   repeat lines 8–12 above;
22 end

   /* Part (c):  patch-boundary vertex */
23 randomly alter the patch boundary angle θ̃_{N_v} to θ̃_{N_v}^new;
24 AngleOK = |θ̃_{N_v}^new| < θ_lim;
25 repeat lines 8–12 above;
```

FIGURE 7 Inner structures of the pseudocode for a vertex trial move. Parts *a–c* are for an interior vertex, a polar vertex, and a patch-boundary vertex, respectively.

$$K_{\mathrm{V}}^{\mathrm{AL}} = \mathit{Switch}3 \times \frac{\tilde{L}^{\mathrm{ref}} B_1}{\min\left(\tilde{l}_1^{\mathrm{ref}}, \tilde{l}_{N_{\mathrm{v}}-1}^{\mathrm{ref}}\right)} \quad \text{per unit volume squared,} \tag{74}$$

and

$$K_{\mathrm{Z}}^{\mathrm{AL}} = \mathit{Switch}4 \times \frac{\tilde{L}^{\mathrm{ref}} B_1}{\min\left(\tilde{l}_1^{\mathrm{ref}}, \tilde{l}_{N_{\mathrm{v}}-1}^{\mathrm{ref}}\right)} \quad \text{per unit length squared.} \tag{75}$$

Thus, $K_{\mathrm{A}}^{\mathrm{AL}} = 0$ if we keep σ or σ_{eff} constant by setting *Switch*2 to 0 in line 5 in Fig. 4, $K_{\mathrm{V}}^{\mathrm{AL}} = 0$ if we keep p or p_{eff} constant by setting *Switch*3 to 0 in line 6 in Fig. 4, and $K_{\mathrm{Z}}^{\mathrm{AL}} = 0$ if we keep f constant by setting *Switch*4 to 0 in line 7 in Fig. 4.

In situations where $\tilde{A}$, $\tilde{V}$, and $\tilde{Z}$ are being constrained, grid refinement of $\tilde{C}$ has very little effect on the fluctuation dynamics of σ (or σ_{eff}) and p (or p_{eff}). In contrast, the dynamics of f is highly dependent on the level of refinement of the polar region(s). At a given T, decreasing the length(s) of the polar edge(s) decreases the rate of oscillation of the fluctuating pole(s) about $\tilde{Z} = Z_0$, but does not reduce the range of the oscillation. This leads to a significantly slower rate of change of $\langle \tilde{Z} \rangle$ and, in turn, a significantly slower rate of change of f as f becomes very insensitive to minute variations of $\langle \tilde{Z} \rangle$. We correct for this effect, thus shortening computation time, by introducing the adaptive magnification factor $\tilde{L}^{\mathrm{ref}}/\min(\tilde{l}_1^{\mathrm{ref}}, \tilde{l}_{N_{\mathrm{v}}-1}^{\mathrm{ref}})$ in Eqs. 73–75. It increases as a polar edge becomes smaller, which increases the

responsiveness of f to minute variations in Z and, in turn, reduces the range of deviation of $\tilde{Z}$ from Z_0. Although not essential for the adjustment of σ and p, the magnification factor has the benefit of constraining $\tilde{A}$ to A_0, and $\tilde{V}$ to V_0, very precisely.

We accept or reject a vertex trial move according to the Metropolis criterion (13). The Boltzmann factor we use here is $\exp(-\Delta\tilde{W}/T)$. Fig. 6 shows the different vertex-dependent simplifications of $\Delta\tilde{W}$. Determination of $\Delta\tilde{F}_{\mathrm{H}}$ is necessary for the last two vertices when $\tilde{C}$ represents a membrane patch. Determination of $\Delta\tilde{Z}$ is necessary for the first and last vertices in the case of a closed membrane, and for the first vertex only in the case of a membrane patch. Determination of $\Delta\tilde{A}$ and $\Delta\tilde{V}$ is not required for the last vertex in the case of a membrane patch. Determination of $\Delta\tilde{A}$, $\Delta\tilde{V}$, and $\Delta\tilde{Z}$ requires knowledge of the nearest-neighbor edge(s). Determination of $\Delta\tilde{F}_{\mathrm{H}}$ requires knowledge of the nearest-neighbor edge and the patch-boundary normal $\tilde{\mathbf{n}}_{N_{\mathrm{v}}}$. Determination of $\Delta\tilde{F}_{\mathrm{b}}$ requires knowledge of both the nearest-neighbor edges and the next-nearest-neighbor edge(s).

Part 1 of the algorithm (line 14 in Fig. 4) performs $n_{\mathrm{loc}} = 100$ Monte Carlo sweeps to

1. Evolve $\tilde{C}$ as described above.
2. Generate $\langle\tilde{A}\rangle$, $\langle\tilde{V}\rangle$, $\langle\tilde{Z}\rangle$, the average per n_{loc} sweeps of $\tilde{A}$, $\tilde{V}$, and $\tilde{Z}$, respectively, for use in adjusting σ or σ_{eff} (part 2), p or p_{eff} (part 3), or f (part 4) later on.
3. Obtain the contribution of each part to $\langle\tilde{\theta}_{\mathrm{i}}\rangle$, the average of θ in a pass through loop 1 (n_1 iterations of parts 1–7). The set of N_{v} average angles, $\{\langle\tilde{\theta}_{\mathrm{i}}\rangle\}$, is to be used for adding or removing vertices in part 8.

Parts 2, 3, and 4 are turned on or off according to the states of the binary variables *Switch*2, *Switch*3, and *Switch*4, respectively (lines 5–7 in Fig. 4). When turned on, part 2 adjusts σ^{AL}, hence σ or σ_{eff} (lines 15–17 in Fig. 4); part 3 adjusts p^{AL}, hence p or p_{eff} (lines 18–20 in Fig. 4); and part 4 adjusts f^{AL}, hence f (lines 21–23 in Fig. 4). Their computational rules, a probabilistic version of the deterministic augmented Lagrangian method (14), are identical and given in Fig. 5, where γ denotes σ^{AL}, p^{AL}, or f^{AL}; G denotes the corresponding geometric quantity to be constrained ($\tilde{A}$, $\tilde{V}$, or $\tilde{Z}$); G_0 denotes the value at which G is to be fixed (A_0, V_0, or Z_0); $\langle G\rangle$ denotes $\langle\tilde{A}\rangle$, $\langle\tilde{V}\rangle$, or $\langle\tilde{Z}\rangle$; $\langle G\rangle_{\mathrm{prev}}$ denotes the previous value of $\langle G\rangle$; and $K_{\mathrm{G}}^{\mathrm{AL}}$ denotes $K_{\mathrm{A}}^{\mathrm{AL}}$, $K_{\mathrm{V}}^{\mathrm{AL}}$, or $K_{\mathrm{Z}}^{\mathrm{AL}}$.

Upon exiting the Lagrange-multiplier adjustment process of parts 2–4, part 5 (line 24 in Fig. 4) retains $\langle\tilde{A}\rangle$, $\langle\tilde{V}\rangle$, and $\langle\tilde{Z}\rangle$ for use as $\langle\tilde{A}\rangle_{\mathrm{prev}}$, $\langle\tilde{V}\rangle_{\mathrm{prev}}$, and $\langle\tilde{Z}\rangle_{\mathrm{prev}}$, respectively, in the next iteration.

Part 6 of the algorithm (line 25 in Fig. 4) adjusts the trial-move maximum step size for each vertex. The maximum step sizes for all the vertices are different and are individually adjusted in the usual Monte Carlo way (13) to keep the success rate of a vertex trial move at $40 \pm 2\%$. In the case of a closed membrane, all the step sizes are distances. In the case of a membrane patch, all step sizes are distances except for the last one, which is an angle by necessity (the position of the last vertex is fixed).

The contour length $\tilde{L}_{\mathrm{min}}$ for a locally stable $\tilde{C}$ is a priori unknown and may be less than the permitted lower limit, $(1-\varepsilon_1)\tilde{L}^{\mathrm{ref}}$, or more than the permitted upper limit, $(1+\varepsilon_1)\tilde{L}^{\mathrm{ref}}$. When $(1-\varepsilon_1)\tilde{L}^{\mathrm{ref}} < \tilde{L}_{\mathrm{min}} < (1+\varepsilon_1)\tilde{L}^{\mathrm{ref}}$, the fractional edge-length changes, $\varepsilon_1^{(\mathrm{i})} \equiv \tilde{l}_{\mathrm{i}}/\tilde{l}_{\mathrm{i}}^{\mathrm{ref}} - 1$, are randomly distributed in the range $[-\varepsilon_1, \varepsilon_1]$. When $\tilde{L}_{\mathrm{min}} < (1-\varepsilon_1)\tilde{L}^{\mathrm{ref}}$, the edges collectively shorten as much as possible ($\varepsilon_1^{(\mathrm{i})} \to -\varepsilon_1$), leading to $\tilde{L}$ decreasing to near $(1-\varepsilon_1)\tilde{L}^{\mathrm{ref}}$. Conversely, when $\tilde{L}_{\mathrm{min}} > (1+\varepsilon_1)\tilde{L}^{\mathrm{ref}}$, all edges lengthen as much as possible ($\varepsilon_1^{(\mathrm{i})} \to \varepsilon_1$), resulting in $\tilde{L}$ increasing to near $(1+\varepsilon_1)\tilde{L}^{\mathrm{ref}}$. The above behavior of $\tilde{L}$ in simulated annealing is accounted for in part 7 of the algorithm (line 26 in Fig. 4), where we implement the adaptive rules of Fig. 8 to enable $\tilde{L}$ to increase or decrease further. The computational parameter ε_2 must be $<\varepsilon_1$, since $\tilde{L}$ is always slightly above its lower limit or slightly below its

The Tethered Infinitesimal Tori and Spheres Algorithm

```
1  if L̃ < (1 − ε2)L̃^ref then
2  |  /* decrease all edge reference lengths by a factor of
   |  1 − ε1 */
3  |  l̃_i^ref ← (1 − ε1)l̃_i^ref, i = 1, ..., N_v − 1;
4  |  L̃^ref ← (1 − ε1)L̃^ref;
5  |  report that L̃^ref has changed;
6  else if L̃ > (1 + ε2)L̃^ref then
7  |  /* increase all edge reference lengths by a factor of
   |  1 + ε1 */
8  |  l̃_i^ref ← (1 + ε1)l̃_i^ref, i = 1, ..., N_v − 1;
9  |  L̃^ref ← (1 + ε1)L̃^ref;
10 |  report that L̃^ref has changed;
11 end
```

FIGURE 8 Pseudocode for adjusting $\tilde{L}^{\mathrm{ref}}$ through uniform dilation or contraction of $\tilde{l}_{\mathrm{i}}^{\mathrm{ref}}$.

upper limit. On the other hand, it must be as close to ε_1 as possible to ensure that the edges are clearly contracting or dilating collectively. We therefore set ε_2 to 0.095. The rules of Fig. 8 ensure that each actual edge length is within both its old allowed range and its new allowed range during the dilation or contraction of $\tilde{L}^{\text{ref}}$.

The angular freedom of $\tilde{C}$ depends on the distribution of vertices along its length. To ensure $\tilde{C}$ has sufficient angular flexibility to capture the curvature of C accurately, we implement, in line 30 in part 8 in Fig. 4, the adaptive rules of Fig. 9 to vary the vertex distribution. Our routine attempts to add vertices first (lines 2–11 in Fig. 9). If no vertices are added, it then attempts to delete vertices (lines 12–20 in Fig. 9). The addition scheme finds all vertices that are too pointy (i.e., the absolute values of their associated local angular averages, $|\langle\tilde{\theta}_{\text{i}}\rangle|$, exceed the angular tolerance $\theta_{\max}$) and evenly divides the actual and reference lengths of their corresponding nearest-neighbor edges, provided that the ratios of the resultant edge reference lengths to the maximum of the original edge reference lengths ($\tilde{l}^{\text{ref}}_{\max}$) exceed the computational parameter ε_3. The deletion scheme removes all interior vertices that are too blunt (i.e., the absolute values of their associated local angular averages, $|\langle\tilde{\theta}_{\text{i}}\rangle|$, are lower than some threshold $\theta_{\min}$, and those of their nearest neighbors, $|\langle\tilde{\theta}_{\text{i}-1}\rangle|$ and $|\langle\tilde{\theta}_{\text{i}+1}\rangle|$, are lower than $\theta_{\max}$) and do not share common edges (they must be at least two edges removed from each other), and inserts new edges with identical actual and reference lengths in the resulting gaps in $\tilde{C}$, provided that the new edges remain short and accurate local approximations of C (we require the ratios of the new actual edge lengths to the original $\tilde{L}$ to be lower than a small parameter $\varepsilon_4 << 1$). The main purpose of the parameter ε_3 in line 3 is to moderate the addition of vertices when the starting and final $\tilde{C}$ are far apart in parameter space. The setting

The Tethered Infinitesimal Tori and Spheres Algorithm

1 /* vertex addition begins */
2 find $\widetilde{l}^{\text{ref}}_{\max}$, the maximum of the $N_{\text{v}} - 1$ edge reference lengths;
3 find all edges with edge reference lengths exceeding $2\epsilon_3\widetilde{l}^{\text{ref}}_{\max}$;
4 find all edges with associated local angular averages $\langle\widetilde{\theta}_i\rangle$ satisfying $\left|\langle\widetilde{\theta}_i\rangle\right| > \theta_{\max}$;
5 find all edges with associated local angular averages $\langle\widetilde{\theta}_{i+1}\rangle$ satisfying $\left|\langle\widetilde{\theta}_{i+1}\rangle\right| > \theta_{\max}$;
6 find all edges satisfying line 3 and either line 4 or line 5 above;
7 **if** *there is at least one edge satisfying line 6* **then**
8 /* evenly divide all edges satisfying line 6 */
9 let the midpoints of all edges satisfying line 6 be the new vertices;
10 let the edge reference lengths for the new edges be half of those for their respective parents;
11 update computational variables; /* vertex addition ends */
12 **else** /* vertex removal begins */
13 find all interior vertices whose associated local angular averages $\langle\widetilde{\theta}_i\rangle$ satisfy $\left|\langle\widetilde{\theta}_i\rangle\right| < \theta_{\min}$;
14 find vertices from line 13 that are at least two edges removed from each other;
15 find vertices from line 14 with nearest neighbors whose associated local angular averages $\langle\widetilde{\theta}_{i-1}\rangle$ and $\langle\widetilde{\theta}_{i+1}\rangle$ satisfy $\left|\langle\widetilde{\theta}_{i\pm1}\rangle\right| < \theta_{\max}$;
16 find vertices from line 15 whose nearest neighbors are less than $\epsilon_4\widetilde{L}$ in separation;
17 **if** *there is at least one vertex satisfying line 16* **then**
18 remove from $\widetilde{C}$ all vertices satisfying line 16;
19 insert new edges with identical actual and reference lengths in the gaps in $\widetilde{C}$ resulting from line 18;
20 update computational variables; /* vertex removal ends */
21 **end**
22 **end**

FIGURE 9 Pseudocode for attempts to add or remove vertices.

$\varepsilon_3 \geq 0.5$ completely stops vertices from being added. The setting $\varepsilon_3 \leq 0$ enables vertices to be added freely. The setting $0 < \varepsilon_3 < 0.5$ leads to moderated addition of vertices, with the degree of moderation increasing with increasing ε_3. If computation time is not a concern, one may set $\varepsilon_4 \leq 0$ to prevent vertices from being deleted.

The angular tolerance $\theta_{\max}$ in lines 4, 5, and 15 is set according to circumstances. When the starting and final $\tilde{C}$ are far apart in parameter space, we save computation time by performing two minimization runs: a first run with $\theta_{\max} = 45^\circ$ and $\varepsilon_3 = 0.05$ to get quickly to a rough final $\tilde{C}$, without creating an excessive number of vertices, and then a second run with $\theta_{\max} = 5^\circ$ and $\varepsilon_3 = 0$ to get the accurate final $\tilde{C}$. We skip the first run when the starting and final $\tilde{C}$ are close in parameter space. The angular tolerance $\theta_{\min}$ in line 13 must be significantly smaller than $\theta_{\max}$ and is generally set to 0.25°. The computational parameter ε_4 in line 16 is generally set to 0.025. Our choice of final values for the four parameters above is meant to ensure $|\delta F_{\mathrm{b}}^{(\mathrm{i})}/\tilde{F}_{\mathrm{b}}^{(\mathrm{i})}| \lesssim 10^{-4}$ as $B_1/T \to \infty$.

APPLICATIONS

We give five examples to demonstrate the capability and versatility of the tethered infinitesimal tori and spheres algorithm.

The first example is determination of a local minimum of $\tilde{F}_{\mathrm{b}}$, $\tilde{F}_{\mathrm{b}}^{\min}$, and its corresponding $\tilde{C}_{\min}$ for a freely suspended closed membrane (Type I of Table 1). The governing parameters are set to values that allow a comparison of results with Jarić et al. (15): $\overline{B}_2 = 1.4$, $\overline{m}_0/(4\pi) = 1.13$, $\overline{V}_0 = 0.7$, and $\overline{f}_0 = 0$. We choose $A_0 = 8\pi\ \mu\mathrm{m}^2$. The algorithm evolves an initially spherical $\tilde{C}$ with $N_{\mathrm{v}} = 21$, to the prolate with $N_{\mathrm{v}} = 76$, plotted in Fig. 10. The mean of $\tilde{F}_{\mathrm{b}}$ decreases with decreasing T and eventually becomes constant at very low T ($B_1/T >> 1$). At the lowest T, which we set to $B_1/T = 2^{20}$, the mean and standard deviation of the relaxed $\tilde{F}_{\mathrm{b}}/(8\pi B_1)$ are 1.696412 ± 0.000002 and 0.00003, respectively. These results are consistent with those given by the strict energy minimization of Jarić et al. (15), wherein $\tilde{F}_{\mathrm{b}}^{\min}/(8\pi B_1) = 1.6975$ for a prolate-shaped $\tilde{C}_{\min}$.

The second example is determination of $\overline{f}$ for an axially deformed closed membrane with $\overline{Z}$ fixed (Type II of Table 1). To allow comparison with the results of Heinrich et al. (7), the values of the governing parameters are set to $\overline{B}_2 = 4/\pi$, $\overline{m}_0/(4\pi) = 1.02209$, $\overline{V}_0 = 0.95$, and $\overline{Z}_0 = 4.0$. We choose $A_0 = 4\pi\ \mu\mathrm{m}^2$. The algorithm evolves an initially spherical $\tilde{C}$ with $N_{\mathrm{v}} = 41$, to the shape with $N_{\mathrm{v}} = 133$, plotted in Fig. 11. At the lowest computational temperature, which we set to $B_1/T = 2^{26}$, the mean and standard deviation of $\overline{Z}$ are 4.00000001 ± 0.00000001 and 0.0000007, respectively, and the mean and standard deviation of $\overline{f}/(8\pi)$ are 8.0700 ± 0.0001 and 0.004, respectively. We estimate from Fig. 8 of Heinrich et al. (7) that $\overline{f}/(8\pi) \approx 8.02$, which is consistent with our value above.

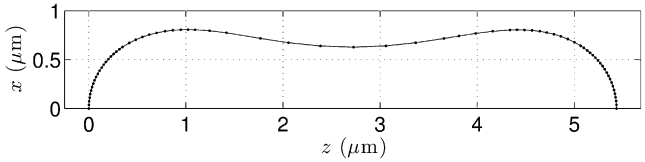

FIGURE 10 Locally stable prolate with $\overline{B}_2 = 1.4$, $\overline{m}_0/(4\pi) = 1.13$, $\overline{V}_0 = 0.7$, and $\overline{f}_0 = 0$, given by the algorithm. The dots show the 76 vertices.

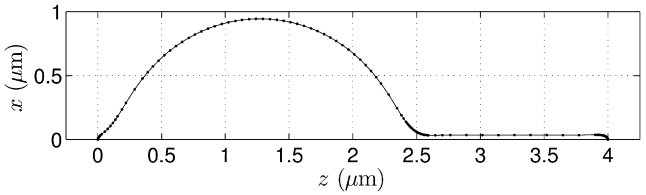

FIGURE 11 $\tilde{C}_{\min}$ with $\overline{B}_2 = 4/\pi$, $\overline{m}_0/(4\pi) = 1.02209$, $\overline{V}_0 = 0.95$, and $\overline{Z}_0 = 4.0$, given by the algorithm. The dots show the 133 vertices.

The third example is determination of $\overline{f}$ for an axially deformed membrane patch with $\overline{Z}$ fixed (Type IV of Table 1). We set the governing parameters to values that allow comparison of results with Dérenyi et al. (9): $\overline{B}_2 = 0$, $\overline{p}_0 = 0$, $\overline{Z}_0 = 5$ and 37.5, and $\overline{R}_{\mathrm{p}}^0 = 20$. The algorithm evolves an initially flat $\tilde{C}$ to the two $\tilde{C}_{\min}$ configurations plotted in Fig. 12, with an increase in N_{v} from 41 to 49 for $\overline{Z}_0 = 5$ and an increase in N_{v} from 41 to 98 for $\overline{Z}_0 = 37.5$. At the lowest computational temperature, which we set to $B_1/T = 2^{20}$, the means and standard deviations of $\overline{Z}$ and the corresponding $\overline{f}$ for the two situations we consider are 1), 5.000001 ± 0.000004 and 0.00006, and 0.8849 ± 0.0002 and 0.003, respectively; and 2), 37.499999 ± 0.000001 and 0.00002, and 1.0030 ± 0.0008 and 0.01,

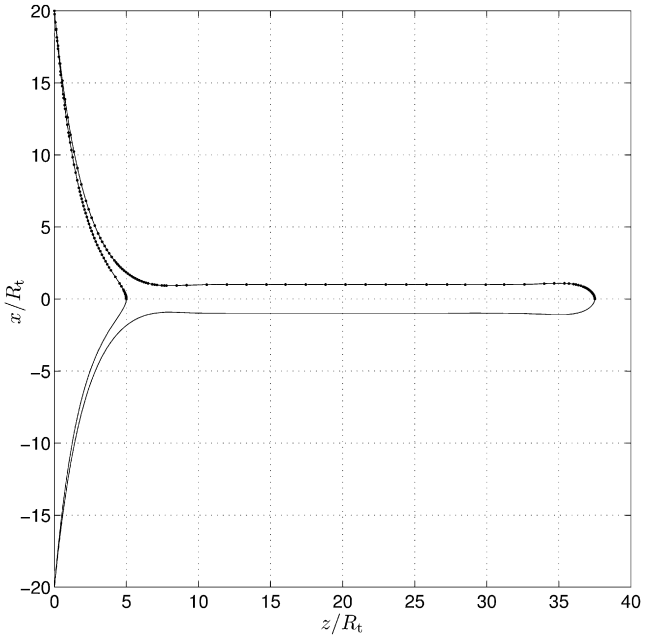

FIGURE 12 $\tilde{C}_{\min}$ for $\overline{Z}_0 = 5$ and $\tilde{C}_{\min}$ for $\overline{Z}_0 = 37.5$, with $\overline{B}_2 = 0$, $\overline{p}_0 = 0$ and $\overline{R}_{\mathrm{p}}^0 = 20$, given by the algorithm. The dots show the 49 vertices in the case of $\overline{Z}_0 = 5$ and the 98 vertices in the case of $\overline{Z} = 37.5$.

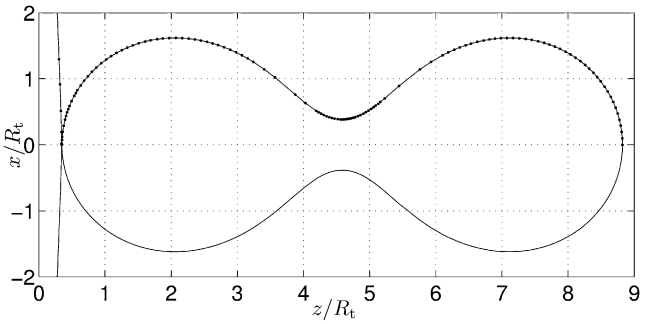

FIGURE 13 Magnified view of the peanutlike protrusion from an otherwise almost flat $\tilde{C}_{\min}$, with $\overline{B}_2 = 2/\pi$, $\overline{m}_0 = 90\pi$, $\overline{R}_0 = 100$, $\overline{p}_0 = 0$, $\overline{f}'_0 = 0$, and $\overline{R}^0_p = 20$, given by the algorithm. The dots denote vertices.

respectively. We estimate from Fig. 1 of Dérenyi et al. (9) that $\overline{f}' \approx 0.886$ for $\overline{Z}_0 = 5$ and $\overline{f}' \approx 1.0$ for $\overline{Z}_0 = 37.5$, which agree with our values above.

The fourth example is determination of the shape for an axially deformed membrane patch with $\overline{f}'$ fixed at $\overline{f}'_0 = 0$. We set the governing parameters to values that allow comparison of results with Dérenyi et al. (16): $\overline{B}_2 = 2/\pi, \overline{m}_0 = 90\pi$, $\overline{R}_0 = 100$, $\overline{p}_0 = 0$, and $\overline{R}^0_p = 20$. We first set $\overline{f}'$ to a nonzero initial value and use the fixed-$\overline{Z}$ setting of the algorithm (Type IV of Table 1) to pull an initially flat $\tilde{C}$ with $N_v = 21$, to $\overline{Z} = 11$ by varying $\overline{f}'$. Then, we release the strained $\tilde{C}$ by fixing $\overline{f}'$ at 0 (Type III of Table 1), and let it evolve into the relaxed shape with $N_v = 170$, plotted in Fig. 13. At the lowest computational temperature, which we set to $B_1/T = 2^{20}$, the mean and standard deviation of $\overline{Z}$ are 8.82147 ± 0.00003 and 0.0007, respectively. This mean value falls within the range for which $\overline{f}'_0 = 0$, estimated from Fig. 7.4 of Dérenyi et al. (16) to be $7.64 \lesssim \overline{Z} \lesssim 8.93$.

Finally, we have used a slightly modified version of the algorithm for numerical energy minimization in a recent study to elucidate the mechanics governing the geometry of the fission-yeast nucleus (17). Our theoretical consideration leads to the minimal model illustrated graphically in Fig. 14, suitable for describing interphase nuclei and abnormal nuclei in which a nuclear microtubule bundle is induced. The free energy functional governing the equilibrium mechanics of the nuclear envelope (NE) has the form

$$F_{NE}[S] = 4B_1 \int_S H^2(\mathbf{r})\mathrm{d}A + 2\sigma_{NE}A[S], \qquad (76)$$

where S is the neutral surface of the NE, H is the mean curvature at a point $\mathbf{r}$ on S, A is the area of S, B_1 is the bending modulus (or stiffness) of the inner and outer bilayers, and σ_{NE} is the membrane tension of the inner and outer bilayers. The area reservoir regulates σ_{NE} and A increases during the fission-yeast cell cycle to twice its initial value at the end of mitosis. The mechanics in the simplest case (interphase nuclei) is described by Eq. 76 and the following two constraints. First, the part of the NE thickness enclosed within S, the shell with thickness t_i, has a small but nonnegligible volume. We account for this finite thickness by introducing a constraint on t_i. Second, we account for the finite volume of the nucleus by introducing a constraint on the volume enclosed by the inner bilayer, V_{net}. When a microtubule bundle is induced inside the nucleus, we must add another constraint to the mechanical description above: the NE cannot physically penetrate the microtubule bundle and, therefore, must be excluded from the space occupied by the bundle. We approximate the bundle as a rigid, impermeable, cylindrical rod with length L_{MT}, radius r_{MT}, and hemispherical ends. We define the volume of this rod to be V_{MT}. As polymerization elongates the bundle sufficiently, it exerts a force $f = \partial F_{NE}/\partial L_{MT}$ on each pole of S, with L_{MT} playing an analogous role to Z in the point-force case.

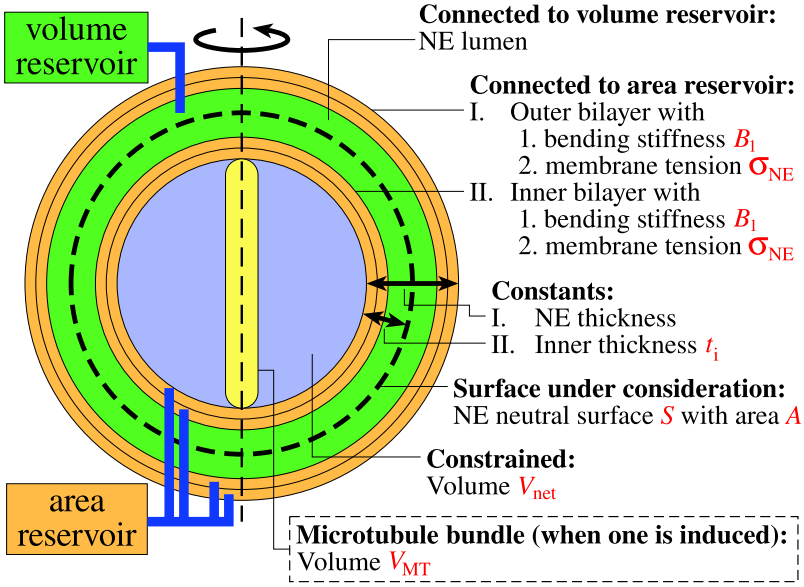

FIGURE 14 Minimal model describing the mechanics governing the geometries of interphase fission yeast nuclei and abnormal nuclei in which a nuclear microtubule bundle is induced.

The general problem is thus to calculate a locally stable S that minimizes the functional

$$W = 4B_1 \int_S H^2 \mathrm{d}A + 2\sigma_{\mathrm{NE}} A - pV_{\mathrm{net}} \qquad (77)$$

for fixed σ_{NE} and V_{net}, subject to two computational conditions added into the modified algorithm: 1), the separation between S and the surface of the cylindrical rod must be $>t_{\mathrm{i}}$; and 2), the total volume of S is $V = V_{\mathrm{net}} + V_{\mathrm{MT}} + At_{\mathrm{i}}$.

A handy feature of the modified algorithm is the placement of hard-ellipsoid potentials with adjustable hardness, ellipticity, and spatial extent on the cylindrical axis. Putting one at each end of the cylindrical rod enables us to push the two poles of S apart and increase L_{MT} easily. In general, they can be used to represent spatially extended objects that push on the membrane from the inside or outside.

We used the modified algorithm to study the change in geometry of S with increasing L_{MT}. We find the deformation of the globally stable S by an elongating microtubule bundle to be a two-stage process. In the first stage, an initially spherical S transforms into a mirror-symmetric lemon. In the second stage, a tube (NE tether) forms at one end of the lemon and elongates with elongation of the microtubule bundle. A notable theoretical result is that the geometry of the lemon stabilizes during NE tether formation and then becomes essentially constant during NE tether elongation. In other words, this constant geometry is effectively set by the two parameters σ_{NE}/B_1 and V_{net}. The geometry of the lemon, in the form of its cross-sectional height and width in the z,x plane, can be determined directly from experiment. Therefore, we can obtain an experimental estimate of σ_{NE}/B_1 by adjusting the values of σ_{NE}/B_1 and V_{net} until the height and width of the theoretical lemon match those of actual nuclei captured in experimental images. We performed this matching procedure, again relying on the modified algorithm for calculating S, on an actual abnormal, one-tether nucleus, the height and width of the lemon-shaped main body of which are measured from electron micrographs (see Fig. 15). This gives an estimate of $\sigma_{\mathrm{NE}}/B_1 = 65\ \mu\mathrm{m}^{-2}$ for this particular nucleus. Assuming $B_1 = 2 \times 10^{-19}$ J (18), this estimate corresponds to a membrane tension of 0.013 mN/m, comparable to the estimated membrane tensions of other biological bilayers (19–21).

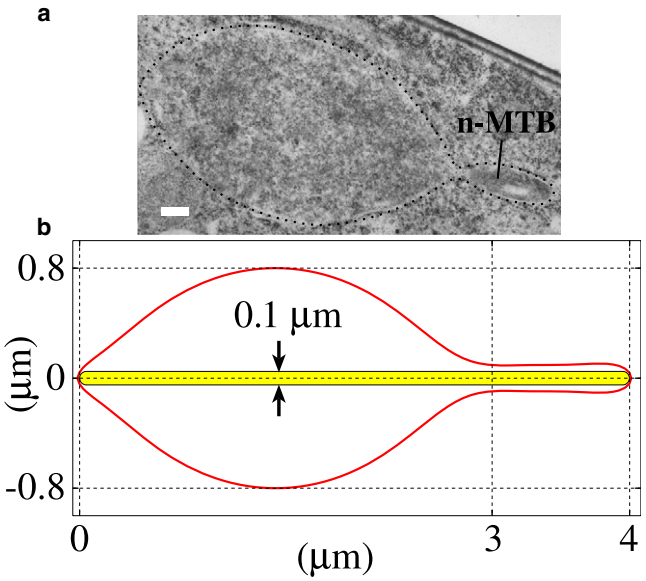

FIGURE 15 (*a*) Electron micrograph of thin section of an abnormally shaped nucleus (*dotted outline*) with a nuclear microtubule bundle (n-MTB) due to overexpression of the *ned1* gene (34). The n-MTB extends across the nucleus but is only partially visible in this section. Based on this and the other sections (not shown), we estimate that the bulge dimensions are ~1.6 μm × 3.0 μm and the n-MTB radius is ~0.05 μm. Bar: 0.2 μm. (*b*) Predicted stable axisymmetric surface S with the same scale as above, obtained by adjusting σ_{NE}/B_1 and V_{net} so that its bulge dimensions match the actual ones, with the constraints $t_{\mathrm{i}} = 0.016$ μm, $L_{\mathrm{MT}} = 4$ μm, and $r_{\mathrm{MT}} = 0.05$ μm. Its size measures are $A = 11.43$ $\mu\mathrm{m}^2$ and $V_{\mathrm{net}} = 2.90$ $\mu\mathrm{m}^3$, corresponding to $\sigma_{\mathrm{NE}}/B_1 = 65$ $\mu\mathrm{m}^{-2}$ and $p/B_1 = 134.28$ $\mu\mathrm{m}^3$, respectively. Figure reproduced from Lim et al. (17).

SUMMARY AND DISCUSSION

To summarize, we have developed a simulated annealing-based numerical algorithm that combines analytical simplicity, numerical robustness, broad applicability, and easy modifiability, for the purpose of making routine the task of constrained minimization of energy functionals governing the equilibrium mechanics of whole, and small patches of whole, ADE-type (1–3) fluid membranes that are closed, spherical in topology, and axisymmetric. The mathematical surface representing a membrane is approximated as the surface of revolution of a one-dimensional chain of vertices and edges. The vertices and edges are treated as the infinitesimal limits of circles and their tangentially linking line segments, respectively. When considered as a surface of revolution, the resulting stack of infinitesimal tori and spheres, linked by truncated cones, allows for the formulation of a discrete form of the continuum elastic energy. The surface is deformed by allowing random displacements of vertices (subject to the boundary conditions). Reparameterization invariance enables vertices to drift freely in the tangential direction, and this is exploited to maintain numerical accuracy. Vertex displacements are accepted or rejected according to the Metropolis criterion, and the corresponding computational temperature (unrelated to physical temperature) is governed by a heuristic annealing schedule. The algorithm as presently implemented can account for constraints on

1. The membrane area or the effective membrane tension.
2. The enclosed volume or the effective pressure difference across the membrane thickness.
3. The axial distance or the applied axial force.

The algorithm can easily be extended for use in the analyses of

1. Deformation of membranes with in-plane order (22) or elasticities, particularly the membranes of red blood cells (10,11,23,24).
2. The effect of the size of extended objects used as handles to pull or push on membranes (17,25–27).
3. The mechanics of membrane engulfment of spherical objects (28,29).
4. The mechanics of membrane pulling or indentation using the tip of an atomic force microscope (30–33).

G.L. gratefully acknowledges the substantial support and computational resources provided by the laboratory of Professor Jonathan Miller at Baylor College of Medicine, where this work was begun. G.H. thanks the Richard Berlin Center for Cell Analysis & Modeling for support, and thanks Professor Miller and Professor Shelley Sazer for their discussions and hospitality during visits to Baylor. We thank an anonymous referee for noticing the connection between our problem and the augmented Lagrangian method for adjusting Lagrange multipliers.

This work was also supported in part by grant No. U54RR022232 from the National Institutes of Health to G.H.

REFERENCES

1. Heinrich, V., S. Svetina, and B. Žekš. 1993. Nonaxisymmetric vesicle shapes in a generalized bilayer-couple model and the transition between oblate and prolate axisymmetric shapes. *Phys. Rev. E Stat. Phys. Plasmas Fluids Relat. Interdiscip. Topics.* 48:3112–3123.
2. Miao, L., U. Seifert, M. Wortis, and H. -G. Döbereiner. 1994. Budding transitions of fluid-bilayer vesicles: the effect of area-difference elasticity. *Phys. Rev. E Stat. Phys. Plasmas Fluids Relat. Interdiscip. Topics.* 49:5389–5407.
3. Seifert, U. 1997. Configurations of fluid membranes and vesicles. *Adv. Phys.* 46:13–137.
4. Bukman, D. J., J. H. Yao, and M. Wortis. 1996. Stability of cylindrical vesicles under axial tension. *Phys. Rev. E Stat. Phys. Plasmas Fluids Relat. Interdiscip. Topics.* 54:5463–5468.
5. Millman, R. S., and G. D. Parker. 1977. Elements of Differential Geometry. Prentice-Hall, Englewood Cliffs, NJ.
6. Božič, B., S. Svetina, and B. Žekš. 1997. Theoretical analysis of the formation of membrane microtubes on axially strained vesicles. *Phys. Rev. E Stat. Phys. Plasmas Fluids Relat. Interdiscip. Topics.* 55:5834–5842.
7. Heinrich, V., B. Božič, S. Svetina, and B. Žekš. 1999. Vesicle deformation by an axial load: from elongated shapes to tethered vesicles. *Biophys. J.* 76:2056–2071.
8. Powers, T. R., G. Huber, and R. E. Goldstein. 2002. Fluid-membrane tethers: minimal surfaces and elastic boundary layers. *Phys. Rev. E Stat. Nonlin. Soft Matter Phys.* 65:041901.
9. Derényi, I., F. Jülicher, and J. Prost. 2002. Formation and interaction of membrane tubes. *Phys. Rev. Lett.* 88:238101.
10. Lim H. W., G. 2003. A numerical study of morphologies and morphological transformations of human erythrocyte based on membrane mechanics. PhD thesis, Simon Fraser University, Burnaby, Canada.
11. Lim H. W., G., M. Wortis, and R. Mukhopadhyay. 2008. Red blood cell shapes and shape transformations: Newtonian mechanics of a composite membrane. *In* Lipid Bilayers and Red Blood Cells Soft Matter., *Vol. 4* G. Gompper and M. Schick, editors. Wiley-VCH, Weinheim, Germany.
12. Jülicher, F. 1996. The morphology of vesicles of higher topological genus: conformal degeneracy and conformal modes. *J. Phys. II.* 6:1797–1824.
13. Allen, M. P., and D. J. Tildesley. 1989. Computer Simulation of Liquids. Oxford, New York.
14. Nocedal, J., and S. J. Wright. 1999. Numerical optimization. *In* Springer Series in Operations Research. P. Glynn and S. M. Robinson, editors. Springer-Verlag, New York.
15. Jarić, M., U. Seifert, W. Wintz, and M. Wortis. 1995. Vesicular instabilities: the prolate-to-oblate transition and other shape instabilities of fluid bilayer membranes. *Phys. Rev. E Stat. Phys. Plasmas Fluids Relat. Interdiscip. Topics.* 52:6623–6634.
16. Derényi, I., G. Koster, M. M. van Duijn, A. Czövek, M. Dogterom, et al. 2007. Membrane nanotubes. *In* Controlled Nanoscale Motion, Lecture Notes in Physics., *Vol. 711* H. Linke and A. Månsson, editors. Springer-Verlag, Berlin, Germany.
17. Lim H. W., G., G. Huber, Y. Torii, A. Hirata, J. Miller, et al. 2007. Vesicle-like biomechanics governs important aspects of nuclear geometry in fission yeast. *PLoS ONE.* 2:e948.
18. Boal, D. H. 2002. Mechanics of the Cell. Cambridge University Press, Cambridge, UK.
19. Dai, J., H. P. Ting-Beall, and M. P. Sheetz. 1997. The secretion-coupled endocytosis correlates with membrane tension changes in RBL 2H3 cells. *J. Gen. Physiol.* 110:1–10.
20. Dai, J., M. P. Sheetz, X. Wan, and C. E. Morris. 1998. Membrane tension in swelling and shrinking molluscan neurons. *J. Neurosci.* 18:6681–6692.
21. Dai, J., and M. P. Sheetz. 1999. Membrane tether formation from blebbing cells. *Biophys. J.* 77:3363–3370.
22. Jiang, H., G. Huber, R. A. Pelcovits, and T. R. Powers. 2007. Vesicle shape, molecular tilt, and the suppression of necks. *Phys. Rev. E Stat. Nonlin. Soft Matter Phys.* 76:031908.
23. Mukhopadhyay, R., G. Lim H. W., and M. Wortis. 2002. Echinocyte shapes: bending, stretching and shear determine spicule shape and spacing. *Biophys. J.* 82:1756–1772.
24. Lim H. W., G., R. Mukhopadhyay, and M. Wortis. 2002. Stomatocyte-discocyte-echinocyte sequence of the human red blood cell: evidence for the bilayer-couple hypothesis from membrane mechanics. *Proc. Natl. Acad. Sci. USA.* 99:16766–16769.
25. Inaba, T., A. Ishijima, M. Honda, F. Nomura, K. Takiguchi, et al. 2005. Formation and maintenance of tubular membrane projections require mechanical force, but their elongation and shortening do not require additional force. *J. Mol. Biol.* 348:325–333.
26. Koster, G., A. Cacciuto, I. Derényi, D. Frenkel, and M. Dogterom. 2005. Force barriers for membrane tube formation. *Phys. Rev. Lett.* 94:068101.
27. Góźdź, W. T. 2007. Deformations of lipid vesicles induced by attached spherical particles. *Langmuir.* 23:5665–5669.
28. Deserno, M. 2004. Elastic deformation of a fluid membrane upon colloid binding. *Phys. Rev. E Stat. Nonlin. Soft Matter Phys.* 69:031903.
29. Sun, S. X., and D. Wirtz. 2006. Mechanics of enveloped virus entry into host cells. *Biophys. J.* 90:L10–L12.
30. Scheffer, L., A. Bitler, E. Ben-Jacob, and R. Korenstein. 2001. Atomic force pulling: probing the local elasticity of the cell membrane. *Eur. Biophys. J.* 30:83–90.
31. Hategan, A., R. Law, S. Kahn, and D. E. Discher. 2003. Adhesively-tensed cell membranes: lysis kinetics and atomic force microscopy probing. *Biophys. J.* 85:2746–2759.
32. Steltenkamp, S., M. M. Müller, M. Deserno, C. Hennesthal, C. Steinem, et al. 2006. Mechanical properties of pore-spanning lipid bilayers probed by atomic force microscopy. *Biophys. J.* 91:217–226.
33. Norouzi, D., M. M. Müller, and M. Deserno. 2006. How to determine local elastic properties of lipid bilayer membranes from atomic-force-microscope measurements: a theoretical analysis. *Phys. Rev. E Stat. Nonlin. Soft Matter Phys.* 74:061914.
34. Tange, Y., A. Hirata, and O. Niwa. 2002. An evolutionarily conserved fission yeast protein, Ned1, implicated in normal nuclear morphology and chromosome stability, interacts with Dis3, Pim1/RCC1 and an essential nucleoporin. *J. Cell Sci.* 115:4375–4385.