

Meshering of flexible membranes under the control of free energy

Xinxin Wang and Gaudenz Danuser*

Lyda Hill Department of Bioinformatics,

Department of Cell Biology,

University of Texas Southwestern Medical Center, Dallas, United States

gaudenz.danuser@utsouthwestern.edu

May 12, 2022

Abstract

Cell membranes are flexible and often undergo extraordinary shape changes during processes like mitosis, formation of protrusions, and vesicle fusion. Mathematical modeling of cell membranes depends on a representation of the free-form surface by discrete meshes. During shape changes, these meshes must be adjusted under the minimization of the total free energy. Current methodology for meshing is limited in one of two ways: 1) Free energy-dependent methods have no restriction on the mesh geometry. The resulting irregular meshes cause artifacts in follow-up models of morphodynamics. 2) Geometry-dependent methods maintain mesh quality but violate the physics of free energy. To fill this gap, we introduce a double-barrier potential between neighboring vertices in the meshes that governs morphodynamics under free energy principles while maintaining regular mesh geometries due to the constraints imposed by barrier crossings. We demonstrate several models of salient morphological changes that are represented by fully intact meshes. An implementation of the algorithm in Matlab is distributed at (<https://github.com/DanuserLab/membraneModels>).

Keywords: remeshing, barrier crossing, morphodynamics

Introduction

Studying the mechanics of membranes is a primary task in biophysical analyses of processes, like morphogenesis [1], cell migration [2], and cytokinesis [3]. During these processes, the membrane adopts a wide range of shapes in response to mechanical and/or biochemical inputs. Recent advances in experimental approaches have enabled the measurement of variables that govern the relation between membrane mechanics and shape. For example, membrane tension can be measured by optical-tweezer- or fluorescence-based approaches [4]; and membrane shape can be measured by 3D light-sheet microscopy combined with computer vision [5].

Mathematical modeling is a critical complement to these experiments. Such models can overcome limitations in the spatiotemporal resolution of the experiment, infer unmeasured variables, and support mechanistic interpretation. Most of these models are grounded in the formalism of the Helfrich free energy [6], which describes the resistance of the membrane to bending. In addition, the total free energy of the membrane should reflect the influence of membrane-internal tension, osmotic pressure and external forces. The minimization of the sum of these energy terms defines the dynamics and final shape of the membrane [7], namely the morphodynamics.

Developing membrane models to study cell morphodynamics requires a sampling of the total free energy in a discretized representation of the free form surface. Using algorithms in computer graphics, such as Delaunay triangulation [8], enables meshing a given membrane into piecewise linear elements, providing the discretization of the initial morphology. However, when minimizing the total free energy, the vertices in the initial mesh must be moved. Suppose the connectivity of the meshes is fixed; such motion can cause geometric distortions that limit the accuracy of the succeeding computation of curvature and free energy (Figure 1) and hence the ensuing balance of mechanical forces that dictates the morphodynamics. Therefore, the mesh connectivity has to be flexible.

Limitations of current approaches

Two types of approaches for updating mesh connectivity have been developed; neither provides the rigor and flexibility to model complex morphodynamic processes:

I) Free energy-determined approaches lack appropriate control over the mesh geometry, including flipping-based (Figure 2A) [9] and splitting- and merging-based (Figure 2B-C) [10] remeshing. These remeshing manipulations act on randomly

chosen edges following the Metropolis–Hastings scheme [9]. Any manipulation that lowers the designated energy function, such as the Helfrich free energy, is considered valid. Acceptance of other manipulations is penalized as a function of the associated energy increase. Usually, the acceptance rate of such random attempts is low. This renders these approaches computationally inefficient. More importantly, these manipulations do not always preserve mesh quality because no geometric restriction is considered. The modeled membrane is globally flexible, but maybe locally distorted. For example, as shown in Figure 2A, flipping without geometric restriction may not improve the mesh quality and often even worsen it. As a result, the computation of the free energy and the ensuing morphodynamics can suffer from major distortion.

Works related to this approach. D. H. Boal and U. Seifert use flexible meshes that allow flipping to represent the membrane of red blood cells [11]. Random attempts of flipping are performed to identify operations that lower the free energy of the membrane. Thermal annealing is allowed based on the Metropolis scheme: new meshes based on the flipping attempts are 100% accepted if the energy is lowered; and conditionally accepted at less and less probability if the free energy increases. The mesh quality is not precisely controlled after the flipping attempts. M. Sadeghi *et. al.* apply the Metropolis flipping method to simulate the dynamic process of a membrane wrapping around a spherical particle [12]. M. Giani *et. al.* apply the Metropolis flipping method to simulate membrane deformation during endocytosis [13]. For additional variants of flipping, we refer to the review [9]. Other than flipping, H. Hoppe *et. al.* implemented splitting and merging as an energy lowering process while maintaining the topological features of the original meshes [10]. L. Ma and W. S. Klug [14] introduce an artificial viscosity to achieve r-adaptive remeshing [15] that keeps the connectivity of the edges and the equilateral shape of the triangles but allows area variation among individual triangles. Despite the effectiveness in describing global membrane deformations, these approaches suffer from local mesh distortions that deteriorate curvature dependent computation of free energy.

II) Geometry-based manipulations restore triangles with an unfavorable edge or angle to nearly equilateral [16]. For instance, long edges are split and vertices delimiting too short edges are merged iteratively until all edges are within the desired range to achieve sufficient mesh quality [17]. Although effective at mesh-quality control, the underlying rules are arbitrary and lack a connection to the free energy. Thus, they are incompatible with physically realistic simulations of morphodynamics.

Works related to this approach. M. Botsch and L. Kobbelt use the maximal and minimal length of edges to trigger splitting and merging [17]. Flipping is performed to reduce the variation in the number of neighboring vertices. M. Dunyach *et. al.* apply the rules in [17] to adaptively remesh the triangles [18], so that they stay nearly equilateral while allowing them to vary in size. As a result, regions with high curvature variation are meshed more finely compared to regions with low curvature variation. I.P. Ivrissimtzis *et. al.* use a neural network based algorithm to achieve high-quality meshing [19]. Y. Guo and Y. Hai use tightly packed spheres and adjust their radius, so that connecting the centers of every three nearby spheres gives high-quality triangles [20]. We refer to [16, 21, 22] for more comprehensive reviews. Despite the effectiveness in maintaining regular meshes, these approaches rely on empirical rules that are unrelated to the physics of membrane deformation.

Contribution

The main contribution of this work is to fit geometry-based mesh-quality control into the rules of free energy minimization, providing a cohesive model of flexible membranes. Our model prescribes that pairs of connected vertices are mechanically coupled based on a double-barrier (or triple-valley) potential V_{in} as a function of the edge length l between the pairs of vertices (Figure 2B). The dynamics of the vertices can activate two remeshing operations (Figure 2B): i) An increase of the distance between two connected vertices beyond the far barrier leads to splitting of the connection and introduction of a third vertex; and ii) a shortening of the distance below the near barrier leads to removal of the connection and merging of the two vertices into one. Based on the spatially restrictive design of V_{in} , iteratively remeshing preserves high mesh quality despite erratic morphodynamics. We demonstrate the remeshing performance in several scenarios of strong local changes of cell shapes without compromising the quality of individual triangles (Figure 2C). Overall, we fill a significant gap in the current methodology for modeling free-form membranes by introducing a remeshing that 1) follows the physical rules of free-energy minimization while 2) following geometrical rules that preserve high mesh quality. We implemented the remeshing method in Matlab and provided scripts for every example discussed as part of the following Results section. The computation of the Helfrich energy is written in C++ and integrated into Matlab via the MEX function. The entire computation is supported by Matlab’s parallel computing toolbox for deployment on an HPC infrastructure.

Results

We propose a method to realize a flexible meshing of a cell surface in a free energy-dependent model of the membrane. In an initial step, the membrane is meshed with triangles, as indicated in Figure 2D. The sequence of the vertices in each triangle follows the right-hand rule that indicates the direction pointing cell-outwards. The dynamics of the vertices follow the minimization of the newly designed internal potential, the Helfrich energy and other free energy terms that describe the membrane tension, osmotic pressure and external mechanics. If the dynamics cause any barrier crossing in V_{in} , the connectivity

between the vertices is adjusted by either a splitting- or merging-based remeshing manipulation. Here, we introduce the free energy terms, describe the remeshing and implement the method in five examples of complex morphodynamics.

Internal potential V_{in}

The internal potential is a scalar function of the edge length l

$$\begin{aligned} V_{in}(l)/V_0 &= b \left\{ \left[1 + e^{-k_{11}(l-r_{11})} \right] + \left[1 + e^{-k_{12}(r_{12}-l)} \right] \right\} : \text{Near barrier} \\ &+ b \left\{ \left[1 + e^{-k_{21}(l-r_{21})} \right] + \left[1 + e^{-k_{22}(r_{22}-l)} \right] \right\} : \text{Far barrier} \\ &+ \tanh[k_w(l-l_{min})] + \tanh[k_w(l-l_{max})] : 2 \text{ walls} \end{aligned}$$

All parameter values and the scales used in this paper are indicated in Table 1. The potential depicts two local barriers (exponential terms) and two walls (hyperbolic tangent terms) that regulate the edge length (Figure 2B). The walls forbid the edge length to go outside the bounds $[l_{min}, l_{max}]$. The two barriers further confine the edges within three ranges, which will play a decisive role in the remeshing operations.

In the middle valley between the two barriers (Figure 2B), edges are retained even under moderate perturbation, keeping $l \approx l_0$. Accordingly, the triangles remain nearly equilateral. Strong expansive perturbations can drive the edges into the valley across the far barrier, where $l \approx 2l_0$ is favored. An edge entering this far valley will split into two $\sim l_0$ -long edges to restore the geometry. Strong compressive perturbations can drive the edges into the near valley, where $l \approx 0$ is favored. An edge entering this valley will merge into a vertex to restore the geometry. The formulated potential thus imposes geometrical constraints to the edges to neutralize the deteriorations caused by external perturbations. Here, we write the free energy term resulting from V_{in} at a given vertex i

$$E_{in}(i) = \sum_{j \in nb(i)} \frac{1}{2} V_{in}(|\vec{r}_i - \vec{r}_j|) \quad (1)$$

where each V_{in} is evenly shared between neighboring vertices; and V_{in} values based on the distances between the position of vertex i and the positions of its 1-ring neighbors Figure 2E are summed. From this term, we derive the resulting internal force

$$\vec{f}_{in}(i) = -\vec{\nabla}_i E_{in} = - \left[\frac{\partial E_{in}(i)}{\partial r_\alpha(i)} + \sum_{j \in nb(i)} \frac{\partial E_{in}(j)}{\partial r_\alpha(i)} \right] \hat{\alpha}, \quad (2)$$

where Einstein's summation rule is implied for the three Cartesian components indicated by α . Of note, $E_{in}(i)$ is governed by the position of vertex i relative to its neighboring vertices $nb(i)$. Shifts in the position of vertex i lead to changes in the local $E_{in}(i)$ of all neighboring vertices $nb(i)$. Therefore, there are two partial-derivative terms in \vec{f}_{in} , addressing how spatially varying i and $nb(i)$ will change $E_{in}(i)$ and the internal force.

Other free energy terms

First, the Helfrich energy H is the free energy term to describe how membranes resist bending. For a closed membrane surface, H depends on the difference between the mean curvature C and the intrinsic mean curvature of the membrane C_0 [23]

$$H = \frac{\kappa}{2} \int_S dS (2C - C_0)^2, \quad (3)$$

where S denotes a closed-2D surface embedded in 3D space, and κ is the bending modulus of the membrane. In this paper, we exclusively consider $C_0 = 0$. The continuous Helfrich free energy is replaced by its discretized version

$$\mathbf{H} = \sum_{i=1}^{N_v} \mathbf{H}(i) = \kappa/2 \sum_{i=1}^{N_v} A_v(i) [2|C(i)|]^2, \quad (4)$$

where N_v is the total number of vertices; and at the i th vertex, the local Helfrich energy $\mathbf{H}(i)$ is governed by the absolute value of the discrete mean curvature and the Voronoi area $A_v(i)$ (Figure 2E). On a mesh surface, the absolute value of the mean curvature can be approximated by $|C(i)| = 1/2|\vec{K}(i)|$. The curvature operator $\vec{K}(i)$ is defined by the cotangent weight method [24]

$$\vec{K}(i) = \frac{1}{2A_v(i)} \sum_{j \in nb(i)} (\cot \theta_{ij} + \cot \varphi_{ij}) [\vec{r}(i) - \vec{r}(j)]. \quad (5)$$

This method guarantees complete coverage of the meshed surface and is therefore ideal for computing the Helfrich free energy. From this term, we derive the resulting bending force

$$\vec{f}_b(i) = -\vec{\nabla}_i H = - \left[\frac{\partial H(i)}{\partial r_\alpha(i)} + \sum_{j \in nb(i)} \frac{\partial H(j)}{\partial r_\alpha(i)} \right] \hat{\alpha}, \quad (6)$$

where Einstein's summation rule is implied and the two partial-derivative terms result from the same reasoning as in the internal force \vec{f}_{in} .

Second, following [25, 26], two additional terms,

$$E_v = k_v \frac{(V - V_0)^2}{V_0} \text{ and } E_s = k_s \frac{(S - S_0)^2}{S_0} \quad (7)$$

are implemented, which penalize the total volume V and surface area S for diverting from the targeted values V_0 and S_0 respectively. k_v and k_s indicate the strength of such penalties. These two terms reflect the influence of osmotic pressure and global membrane tension. Here, we applied the method in [27] to compute the three-dimensional volume enclosed by the membrane. The total free energy of the membrane E is thus the sum of E_{in} , H , E_v and E_s .

Equations of motion

From the gradient of the free energy terms with respect to the positions of the mesh vertices \vec{r} , we obtain the forces necessary to change shape. In addition, a random force $\vec{\xi}$ adds fluctuation to the morphodynamics. $\vec{\xi}$ and the forces resulting from H , E_v and E_s change moderately with the changes of \vec{r} and thus are referred to as regular forces, \vec{f}_{reg} . Additionally, the forces corresponding to E_v and E_s are averaged at each vertex. The averaged forces include the forces at the vertex and the vertex's neighbors. At the i th vertex,

$$\vec{f}_{reg}(i) = \vec{f}_{in}(i) + \vec{f}_b(i) + \vec{\xi}(i) - \frac{\partial(E_v + E_s)}{\partial \vec{r}(i)} \quad (8)$$

\vec{f}_{reg} is updated regularly after every time step $\Delta\tau$. We set $\vec{\xi} = 0$ in the following examples to eliminate the disruption of the random force to the morphodynamics.

However, the potential V_{in} is spatially complex and the corresponding force \vec{f}_{in} is updated adaptively. The time step is shortened whenever the edges reach the potential's complex regions. Otherwise, the time step remains large when all edges are in regular (flat) regions of the potential. As a result, we refine the morphodynamics based on V_{in} to avoid the overshoot problem. This adaptive time step Δt_δ^{in} is recalculated at every simulation step δ . See Methods for details. By combining the two types of forces, we obtain the membrane's equation of motion (EOM). First, the constant time step $\Delta\tau$ and the adaptive time step Δt_δ^{in} are superimposed as an effective time step Δt_δ^{eff} illustrated in Figure 2F. The schedule of Δt_δ^{eff} is defined as follows: when the simulation lands on the ticks of the Δt_δ^{in} schedule, we update \vec{f}_{in} only. When the simulation lands on the ticks of the $\Delta\tau$ schedule, we update both \vec{f}_{in} and \vec{f}_{reg} . Next, we introduce \underline{t} as the time when \vec{f}_{reg} is last updated. Last, at the i th vertex, the EOM is written as

$$\vec{r}(i, t_\delta) = \vec{r}(i, t_{\delta-1}) + \mu \left[\vec{f}_{in}(i, t_{\delta-1}) + \vec{f}_{reg}(i, \underline{t}) \right] \Delta t_\delta^{eff}, \quad (9)$$

where μ is the mobility. See Methods for the computation of Δt_δ^{eff} and more details of the adaptive dynamics.

Remeshing: splitting and merging

Given the EOM of the vertices, it is necessary to also dynamically remesh the membrane representation to prevent degeneration of the mesh geometry. Per simulation step, one edge in the mesh may cross a barrier in the potential V_{in} , referred to as a barrier crossing event. Such an event can expand an edge by a factor of 2, requiring a split, or compress an edge by a factor of 4, requiring a merge of the two connected vertices (Figure 2B). After a splitting or merging event, all edges are restored approximately to the target length l_0 . In addition, the connectivities near the added or discarded edges are updated to maintain the triangular mesh topology; a local relaxation is applied to the added or discarded edges to maintain global stability. See the detailed steps of splitting and merging in Methods. Iteratively applying the remeshing manipulations resolves all barrier crossing events and thus preserves high mesh quality.

Spontaneous morphodynamics

We applied the EOM/remeshing to simulate the morphodynamics of a biconcave red blood cell (RBC), see Figure 3A. Starting from a sphere, V_0 was set to 60% of the volume of the initial sphere, and S_0 was set to the surface area of the initial sphere.

This constraints imposed a force field that let the morphology spontaneously converge towards a biconcave equilibrium shape (Video 1). Locally, high mesh quality was preserved despite the global change of the morphology. Obtaining this well-known morphology validates the new method. See [26] for a comparison, where the remeshing is based on geometrical rules.

Next, we applied the EOM/remeshing to simulate the morphodynamics of two fusing vesicles, see Figure 3B. In this example, V_0 was equal to the total volume of the two vesicles; and S_0 was equal to the total surface area of the two vesicles. Under these assumptions, starting from two vesicles connected with a single triangular mesh, the morphology of the simulated membrane gradually became one elliptical vesicle (Video 2). High mesh quality was preserved throughout the entire fusion process.

Morphodynamics under external constraints

Most shape changes are driven by intra/extracellular structures mechanically coupled to the membrane. Adding an external term in the total free energy is required to reflect how these structures affect the morphodynamics. In our simulations, we represent these structures by external points that are connected to select target vertices and their 1-ring neighbors via Hookean springs with a spring constant k_{ex} (Figure 4A). As an approximation of the high rigidity of the mechanical structures relative to the membrane, we fix the position of the external points and only use the external forces to move the membrane.

Using these external forces, we mimic several cell morphogenic processes entailing substantial shape changes. Our investigation of the method's performance focused on the relationship between morphological changes and the mesh quality.

Filopodium. We simulated the formation of this most salient form of cell protrusion, often encountered in neurons and other types of motile cells [28]. To trigger the extension of a narrow tube from a globular cell, we placed one external point away from the sphere while maintaining the overall shape with external points on the sphere (Figure 4B). The equilibrium shape correctly reflected the very local expansion of the membrane without reducing mesh quality throughout the simulation, neither on the tube or tip nor at the neck (Video 3).

Lamellipodium/ruffle. We simulated the formation of this sheet-like form of cell protrusion, often encountered at the leading edge of polarized, motile cells and around non-polar but stimulated cells [29]. To trigger the extension of a flat membrane-fold from a globular cell, we placed a rim of external points away from the sphere while maintaining the overall shape with external points on the sphere (Figure 4C). The equilibrium shape correctly reflected the expansion of the fold without reducing the mesh quality near the rim or the neck. The method accomplishes this performance by inserting a significant number of vertices on the protruding portion of the membrane, avoiding the build-up of internal tension along the neck (Video 4).

Invagination. We simulated the formation of this intracellular invagination of cell membrane, often encountered at the surface of cells during entry processes such as endocytosis and macropinocytosis [30]. To trigger the retraction of a membrane dome from a globular cell, we placed a hemisphere of external points inside of the sphere while maintaining the overall shape with external points on the sphere (Figure 4D). The equilibrium shape correctly reflected the inward curving of the membrane without reducing mesh quality throughout the simulation, neither on the concave dome nor along the convex surrounding area (Video 5).

Discussion

We present a new approach for remodeling triangular-meshed membranes that follows the rules of free energy minimization. We introduce a new potential that controls the quality of the meshes via splitting- and merging-based remeshing in the context of membrane potential. Of particular importance, this approach supports accurate computing of the bending force, which highly relies on mesh quality (Figure 1).

Parameter interpretation

The parameters in the internal potential V_{in} are responsible for the geometrical control over the mesh quality. For this purpose, we determine these parameters' baseline values to keep the two barriers and the two walls of V_{in} steep. In addition, we determine V_0 to set the height of the barriers properly. For V_0 s smaller than the baseline value, or lower barriers, the edges are more likely to experience the barrier-crossing followed by splitting and merging. Too frequent remeshing could lead to a rapid increase and decrease of the valence numbers, causing many unsplittable and unmergeable edges in large-force regions. As a result, these abnormal edges halt further remeshing and reduce the flexibility of the membrane. For V_0 s larger than the baseline value, or higher barriers, the edges are less likely to experience the barrier-crossing. This could lead to highly inert mesh topologies under the baseline external force characterized by k_{ex} and the bending force characterized by κ . Increasing the external force and decreasing the bending force may remedy stagnation of the membrane under large V_0 s. We provide a section in the Matlab script for users to adjust the parameters.

Model limitations

Our new method is limited by spatial scales. Obtaining molecular details of the interaction between membrane components and the external environment requires finer treatments such as molecular dynamics or coarse-grain models. A recent publication [31] discusses a framework to convert meshed membranes to the coarse-grain structures of the basic components of the membrane. When finer details are needed this method could be applied to adaptively convert the mesh results into membrane representations with discrete elements.

The heterogeneity of membrane components is not reflected by the potential proposed in this paper. For example, a membrane raft is a heterogeneous region distinctive from the regular regions of the membrane. Different treatments to the meshes in such regions are needed. Currently, all of the membrane properties in the model are globally homogeneous, including the bending modulus, osmotic pressure, and intrinsic curvature ($C_0 = 0$). However, the variety of spatial arrangement of the external points provides flexibility to indirectly introduce homogeneity and trigger the formation of salient shape features.

Changes in membrane area are physically associated with rearrangements in the lipid bilayer. Thus, membrane patches cannot be added or removed arbitrarily, even under large external forces. In this paper, we assume instantaneously equilibrating lipids, and no feedback from lipid homeostasis on the morphodynamics. Adding non-equilibrium distribution of the lipids to the current approach can lead to a more realistic representation of the membrane.

External force point-membrane interaction

In this paper, we assumed that the membrane is externally sculpted by a set of constant point forces. It is straightforward to add dynamics to either external point location or the force magnitude an external point exerts in order to reflect mechanical processes outside or inside the cell and even to capture feedback between membrane shape and external forces. For example, cortical actin network growth or the dynamics of actin filaments in filopodia and lamellipodia are in direct feedback interaction with the morphodynamics of the membrane. Force feedbacks also arise from curvature sensitive signals [32]. The proposed algorithm can capture such relations by spatiotemporal adjustment of external force magnitudes. This will permit integration of the proposed membrane model with additional biochemical and mechanical systems. This possibility highlights the advantage of a meshing framework that couples a physical model to optimized representation of the membrane geometry.

Methods

We introduce an adaptive Langevin equation (LE) to implement the dynamics of the membrane subject to a remeshing, in which both the mechanically driven motion and the geometric regulation of mesh quality are considered.

Conventional Langevin equation

The dynamics of the meshed membrane can be described by an over-damped LE that defines the motion of any vertex i by the mechanical coupling to its 1-ring neighbors (Figure 2E), accounting for the bending resistance and attraction from external points, as well as for stochastic forces

$$\frac{d\vec{r}(i)}{dt} = \mu\vec{f}(i) + \sqrt{2\mu k_B T}\vec{R} \quad (10)$$

Here, \vec{f} and \vec{R} denote deterministic and stochastic forces, respectively, and k_B is the Boltzmann constant and T the temperature. The distribution of the random force follows Gaussian white noise. Using natural unit and component expression (using α as Cartesian coordinates, see Table 1), integration of the LE over a finite time step Δt yields

$$r_\alpha(i, t + \Delta t) = r_\alpha(i, t) + \mu f_\alpha(i, t) \Delta t + \varepsilon_\alpha(i, t) \sqrt{2\mu k_B T \Delta t} \quad (11)$$

where $\varepsilon_\alpha(i, t)$ is a normal distributed random number with $N(0, 1)$, and μ is the mobility. The integration of the deterministic force is approximated by multiplication of the time step and the force calculated at the beginning of the time step. However, this simple form of the LE is unsuitable for simulating the membrane dynamics. Unless Δt is extremely small, replacing the time integral by a finite step is inaccurate for regions with salient variation in the triple-valley potential (Figure 2B). Thus, we propose an alternative LE, which is based on adaptive time steps to better capture the effect of variation on membrane dynamics.

To be formally consistent with the deterministic forces, we introduce the time-averaged stochastic force

$$\xi_\alpha(i, t) = \frac{\varepsilon_\alpha(i, t) \sqrt{2\mu k_B T \Delta t}}{\mu \Delta t}, \quad (12)$$

so that the discrete Langevin equation (eq. 11) is rewritten as

$$r_\alpha(i, t + \Delta t) = r_\alpha(i, t) + \mu f_\alpha(i, t) \Delta t + \mu \xi_\alpha(i, t) \Delta t. \quad (13)$$

For one dimensional systems,

$$l(i, t + \Delta t) = l(i, t) + \mu f(i, t) \Delta t + \mu \xi(i, t) \Delta t. \quad (14)$$

Adaptive dynamics

To implement a dynamics that is adaptive to the complexity of the potential, we finely segment regions where V_{in} is complex and coarsen regular regions. Then, we reversely calculate the time steps according to these inhomogeneous segments to refine the simulation. The principle is that every edge, regardless of its location and the complexity of the associated potential, extends or shrinks no more than one segment during each time step. Thus, the edges scan through all the segments one by one without skipping. Because the complex regions have more segments than the regular regions, the dynamics in the complex regions are treated more delicately.

We first evenly sample the potential values separated by a small distance dl in the edge length- l space (Figure S1),

$$V_{in}(l) \text{ for } l := 0, dl, \dots, mdl, \dots, Mdl, \quad (15)$$

The total number of dl bins is given by $M = l_{\max}/dl - 1$. Note that by removing one dl , we avoid the singularity in V_{in} at $l = l_{\max}$.

Next, we divide the l space into segments according to l . Each segment consists of one or multiple dl in length

$$\mathbf{s} := \mathbf{s}_1 \cup \mathbf{s}_2 \cup \dots, \mathbf{s}_{\omega} \cup \dots, \cup \mathbf{s}_{\Omega}, \quad (16)$$

where Ω is the total number of segments. A given segment \mathbf{s}_{ω} reads

$$\mathbf{s}_{\omega} := (n_{\omega-1}dl, n_{\omega}dl], \quad (17)$$

where the coefficient n_{ω} , defines the inclusive upper-bound of the segment, and $n_{\omega-1}$ defines the exclusive lower-bound of the segment (Figure S1).

The sequential coefficients $n_0 < n_1 < \dots < n_{\omega} < \dots n_{\Omega}$ make l space inhomogeneously segmented. The complex regions possess many short segments, and the regular regions possess fewer but longer segments (Figure S1). To begin the segmentation, we represent the discrete space l with seven critical points (CP), including the two boundaries at $l = 0$ and $l_{\max} - dl$, three local minima (in the three valleys) and two local maxima (on the two barriers) of $V_{in}(l)$. Therefore, there are six initial segments $\mathbf{s} = \mathbf{s}_1 \cup \dots \cup \mathbf{s}_6$ separated by the CPs. Then, if any segment \mathbf{s}_{ω} in \mathbf{s} violates any of the two conditions,

1. The linear regression fitting of $V_{in}(l)$ within \mathbf{s}_{ω} gives $R^2 > 0.95$
2. The standard deviation of $V_{in}(l)$ within \mathbf{s}_{ω} gives $\sigma < 0.05V_0$

the segment \mathbf{s}_{ω} is divided into two segments from its middle. By repeating this operation until all the segments satisfy the two conditions, the final segmentation is obtained. Within every segment, the sampled $V_{in}(l)$ (dots in Figure S1) is highly linear and uniform as controlled by the two conditions. For example, in Figure S1, the red segment is long, where $V_{in}(l)$ is highly linear and moderately varying; the black segment is short, where $V_{in}(l)$ is highly nonlinear; and the blue segment is short, where $V_{in}(l)$ is strongly varying.

After the segmentation, we obtain a piece-wise linear substitute for the original potential (Figure S1):

$$V_{in}(l) = V_{in}(n_{\omega-1}dl) + \frac{V_{in}(n_{\omega}dl) - V_{in}(n_{\omega-1}dl)}{(n_{\omega} - n_{\omega-1})dl} (l - n_{\omega-1}dl), \text{ for } l \in \mathbf{s}_{\omega} \quad (18)$$

Next, we write the total time of the simulation as consecutive time steps δ ,

$$t_{\delta} = \sum_{\delta'=0}^{\delta} \Delta t_{\delta'}, \text{ for } \delta = 0, 1, \dots, N_{\delta} \quad (19)$$

with $t_0 = 0$ and $\Delta t_0 = 0$. During a given step δ , the time begins at $t_{\delta-1}$ and ends at t_{δ} (blue-shaded area in Figure 3A). The choice of the time step Δt_{δ} is based on an adaptive Δt_{δ}^{in} indicating when to update the internal force \vec{f}_{in} , and a constant time step $\Delta\tau$ indicating when to update the regular forces, including the bending \vec{f}_b , external \vec{f}_{ex} , time-averaged stochastic forces $\vec{\xi}$ and the forces resulting from the volume and area penalty. In contrast to \vec{f}_{in} , these forces change slowly with the changes of the vertex positions and thus are referred to as the regular forces, \vec{f}_{reg} . To avoid unnecessary computation, we separate the time scales of updating the regular forces \vec{f}_{reg} from the time scale of updating the internal forces \vec{f}_{in} .

The constant $\Delta\tau$ and the adaptive Δt_{δ}^{in} are superimposed as illustrated in Figure 3A. To obtain the effective time step, we compute the adaptive time step Δt_{δ}^{in} based on the internal force. Considering an edge ϵ within the segment \mathbf{s}_{ω} at the beginning, i.e. $l_{\epsilon}(t_{\delta-1}) \in \mathbf{s}_{\omega}$, ϵ can be extended or compressed at most to the mid-points of \mathbf{s}_{ω} 's neighbors— $\mathbf{s}_{\omega+1}$ or $\mathbf{s}_{\omega-1}$ at the end of δ . At t_{δ} the edge length is thus either

$$l_{\epsilon}^{+}(t_{\delta}) = \frac{n_{\omega+1}dl + n_{\omega}dl}{2} \text{ or } l_{\epsilon}^{-}(t_{\delta}) = \frac{n_{\omega-1}dl + n_{\omega-2}dl}{2} \quad (20)$$

and then reversely calculate the two time steps for reaching these lengths.

Considering a given edge ϵ connecting vertex i and j , the deterministic part of its dynamics in the component-wise Langevin equation reads

$$\begin{aligned} r_\alpha(i, t_\delta) &= r_\alpha(i, t_{\delta-1}) + \mu f_\alpha^{tot}(i, t_{\delta-1}) \Delta t_\delta^{in} \\ r_\alpha(j, t_\delta) &= r_\alpha(j, t_{\delta-1}) + \mu f_\alpha^{tot}(j, t_{\delta-1}) \Delta t_\delta^{in}, \end{aligned} \quad (21)$$

where the total force is $f_\alpha^{tot} = f_{in,\alpha} + f_{reg,\alpha} + \xi_\alpha$ as in eq. 9. Then, we calculated the difference in squared-length after Δt_δ^{in}

$$|l_\epsilon(t_\delta)|^2 - |l_\epsilon(t_{\delta-1})|^2 = \sum_\alpha \left\{ [r_\alpha(j, t_\delta) - r_\alpha(i, t_\delta)]^2 - [r_\alpha(j, t_{\delta-1}) - r_\alpha(i, t_{\delta-1})]^2 \right\}, \quad (22)$$

which is simplified as

$$\sum_\alpha \left\{ 2[r_\alpha(j, t_{\delta-1}) - r_\alpha(i, t_{\delta-1})] [\Delta r_\alpha(j, t_{\delta-1}) - \Delta r_\alpha(i, t_{\delta-1})] + [\Delta r_\alpha(j, t_{\delta-1}) - \Delta r_\alpha(i, t_{\delta-1})]^2 \right\}, \quad (23)$$

where the difference terms read

$$\begin{aligned} \Delta r_\alpha(i, t_{\delta-1}) &= r_\alpha(i, t_\delta) - r_\alpha(i, t_{\delta-1}) \\ \Delta r_\alpha(j, t_{\delta-1}) &= r_\alpha(j, t_\delta) - r_\alpha(j, t_{\delta-1}) \end{aligned} \quad (24)$$

Then, by substituting eq. 24 in eq. 22 and ignoring the second order term we have

$$|l_\epsilon(t_\delta)|^2 - |l_\epsilon(t_{\delta-1})|^2 \approx 2 \sum_\alpha [r_\alpha(j, t_{\delta-1}) - r_\alpha(i, t_{\delta-1})] [\Delta r_\alpha(j, t_{\delta-1}) - \Delta r_\alpha(i, t_{\delta-1})], \quad (25)$$

Substituting eq. 21, to replace the difference terms in eq. 25

$$|l_\epsilon(t_\delta)|^2 - |l_\epsilon(t_{\delta-1})|^2 \approx 2\mu \Delta t_\delta^d \sum_\alpha [r_\alpha(j, t_{\delta-1}) - r_\alpha(i, t_{\delta-1})] [f_\alpha^{tot}(j, t_{\delta-1}) - f_\alpha^{tot}(i, t_{\delta-1})],$$

Finally, the expression of the adaptive time step for forward (extensive) and backward (compressive) jumping is obtained as

$$\begin{aligned} \Delta t_\delta^{in+}(\epsilon) &\approx \frac{|(n_{\omega+1}dl + n_\omega dl)/2|^2 - |l_\epsilon(t_{\delta-1})|^2}{2\mu \sum_\alpha [r_\alpha(j, t_{\delta-1}) - r_\alpha(i, t_{\delta-1})] [f_\alpha^{tot}(j, t_{\delta-1}) - f_\alpha^{tot}(i, t_{\delta-1})]} \\ \Delta t_\delta^{in-}(\epsilon) &\approx \frac{|(n_{\omega-1}dl + n_{\omega-2}dl)/2|^2 - |l_\epsilon(t_{\delta-1})|^2}{2\mu \sum_\alpha [r_\alpha(j, t_{\delta-1}) - r_\alpha(i, t_{\delta-1})] [f_\alpha^{tot}(j, t_{\delta-1}) - f_\alpha^{tot}(i, t_{\delta-1})]} \end{aligned} \quad (26)$$

where ‘+’ and ‘-’ indicates extension and compression respectively. Because of the different signs of the numerators but the same denominators, only one deformation gives a positive time step, which is the physically plausible choice:

$$\Delta t_\delta^{in}(\epsilon) = \begin{cases} \Delta t_\delta^{in+}(\epsilon) & \text{if } \Delta t_\delta^{in+}(\epsilon) > 0 \\ \Delta t_\delta^{in-}(\epsilon) & \text{else.} \end{cases} \quad (27)$$

Each edge triggers a different time step. We choose the shortest step

$$\Delta t_\delta^{in} = \min [\Delta t_\delta(\epsilon)], \forall \epsilon = 1, \dots, \Omega. \quad (28)$$

Δt_δ^{in} ensures that the edge under maximal extension or compression crosses only one segment. Deformations of all edges are limited to one segment or less allowing rapid computation of the updated potential via the piece-wise linear approximation. Therefore, the adaptive LE (eq. 9) based on the adaptive Δt_δ^{in} and the constant $\Delta\tau$ optimally captures the spatial complexity of the potential V_{in} .

Steps of splitting and merging

Following a barrier crossing event (BCE) in V_{in} , we outline the steps S1 – S5 of a split, and steps M1 – M5 of a merge (Figure S2), starting from the initial mesh configuration shown in (S0) and (M0) respectively.

- *Splitting:*

(S1) Given an edge ϵ connecting vertices i_1 and i_2 with a length $l_\epsilon > l_+$, the global dynamics of the mesh is paused to allow relocation of only i_1 , i_2 and their 1-ring neighbors (green circles). These vertices respond only to the internal forces and noise. The bending and the external forces are halted to avoid inaccuracy in computing these forces caused by the extended ϵ . Additionally, we include the positions of ϵ 's 2-ring neighbors as a fixed boundary (blue dots). In

this way, we can complete the BCE while limiting the effects of changes in local mesh topology to the other vertices, hereon referred as the local relaxation.

In order to accelerate the local relaxation, we change the Gaussian noise to a biased Levy noise [33] for efficiently extending ϵ and keeping the other edges near l_0 . The Levy noise is biased to extend ϵ and nearby edges that get overly short ($l < l_-$) or overly long ($l > l_+$) are stabilized. Two possible results are expected. First, if $l_\epsilon > l_{++}$ and the other edges satisfy $l_- < l < l_+$, the local relaxation is complete and the BCE is successful. The splitting proceeds to the next step (S2). Second, if $l_\epsilon < l_+$ and the other edges satisfy $l_- < l < l_+$, the local relaxation is also complete but the BCE is unsuccessful. The splitting is canceled and the mesh is restored to the initial configuration (S0). Note that the local relaxation merely provides an intermediate mesh configuration to facilitate the remeshing. The final shape of the membrane is determined by the global dynamics with all the forces restored after finishing the remeshing.

Before describing the following steps, we clarify the notation: the vertices connecting to both i_1 and i_2 are numbered as j_1 and j_2 . Vertex j_1 fulfills the right-hand rule in triangle $j_1 \rightarrow i_1 \rightarrow i_2$ (Figure 3A), vertex j_2 fulfills the right-hand rule in triangle $j_2 \rightarrow i_2 \rightarrow i_1$. Vertex k_1 forms a triangle with the edge $i_1 j_1$, and analogously, vertices k_2 , k_3 , and k_4 form triangles with the edges $i_2 j_1$, $i_2 j_2$ and $i_1 j_2$, respectively.

(S2) To split ϵ we insert vertex i_n at the mid-point between i_1 and i_2 , yielding two $\sim l_0$ -long edges. Then, we define a new mesh structure around i_n by disconnecting $i_1 i_2$, $i_1 j_1$, $i_2 j_2$, $i_1 j_2$ and $i_2 j_2$ and connecting i_n to its closest neighbors, i.e. $i_n i_1$, $i_n i_2$, $i_n j_1$ and $i_n j_2$. The outcome of this step is four quadrilaterals, which require proper triangulation.

(S3) Each quadrilateral has two options (magenta dashed lines) for triangulation, yielding $2^4 = 16$ options of triangular meshes. We only keep options that preserve the valence of every vertex between 5 and 8. The valence of a given vertex is equal to the number of its 1-ring neighbors. The choice between 5 and 8 avoids unreasonable geometries. If no option can satisfy the valence requirement, the edge is considered unsplittable, and the configuration restored to (S0). Additionally, the extended ϵ is restored to $l_\epsilon < l_+$ via a similar local relaxation as in (S1), except that the Levy noise in this case compresses ϵ . Otherwise, if one or more options can satisfy the valence requirement, the option minimizing the variance of the valence is chosen.

(S4) The new edges (black dashed lines) in (S3) undergo a second local relaxation, similar to (S1). All of the moveable vertices in the new edges experience the internal force and Levy noise. The neighbors at the local border remain fixed.

(S5) The local relaxation is complete when every edge satisfies $l_- < l < l_+$.

- *Merging:*

(M1) Given an edge ϵ connecting vertices i_1 and i_2 and a length of l_ϵ , when $l_\epsilon < l_+$, the global dynamics of the mesh is paused to allow relocation of only i_1 , i_2 and their 1-ring neighbors (green circles). These vertices respond only to the internal forces and noise. The bending and the external forces are halted to avoid inaccuracy in computing these forces caused by the compressed ϵ . Additionally, we include the positions of ϵ 's 2-ring neighbors as a fixed boundary (blue dots).

In order to accelerate the local relaxation, we change the Gaussian noise to a biased Levy noise [33] for efficiently compressing ϵ and keeping the other edges near l_0 . The Levy noise is biased to compress ϵ and nearby edges that get overly short ($l < l_-$) or overly long ($l > l_+$) are stabilized. Two possible results are expected. First, if $l_\epsilon < l_-$ and the other edges satisfy $l_- < l < l_+$, the local relaxation is complete and the BCE is successful; then the merging proceeds to the next step (M2). Second, if $l_\epsilon > l_-$ and the other edges satisfy $l_- < l < l_+$, the local relaxation is also complete but the BCE is unsuccessful; then the merging is canceled and the mesh is restored to the initial configuration (M0).

Before describing the following steps, we clarify the notation: the vertices connecting to both i_1 and i_2 are numbered as j_1 and j_2 . Vertex j_1 fulfills the right-hand rule in triangle $j_1 \rightarrow i_1 \rightarrow i_2$ (Figure 3A), vertex j_2 fulfills the right-hand rule in triangle $j_2 \rightarrow i_2 \rightarrow i_1$. Vertex k_1 forms a triangle with the edge $i_1 j_1$, and analogously, vertices k_2 , k_3 , and k_4 form triangles with the edges $i_2 j_1$, $i_2 j_2$ and $i_1 j_2$, respectively.

(M2) To merge ϵ we remove i_2 and the original edges $i_1 i_2$, $i_1 j_1$, $i_1 j_2$, $i_2 j_1$ and $i_2 j_2$. The outcome of this step is two quadrilaterals, which require proper triangulation.

(M3) Each quadrilateral has two options (magenta dashed lines) for triangulation, yielding $2^2 = 4$ options of triangular meshes. We only keep options that preserve the valence of every vertex between 5 and 8. If no option can satisfy the valence requirement, the edge is considered unmergeable, and the configuration restored to (M0). Additionally, the compressed ϵ is restored to $l_\epsilon > l_-$ via a similar local relaxation as in (M1), except that the Levy noise in this case extends ϵ . Otherwise, if one or more options can satisfy the valence requirement, the option minimizing the variance of the valence is chosen.

(M4) The new edges (black dashed lines) in (M3) undergo a second local relaxation, similar to (M1). All of the moveable vertices in the new edges experience the internal force and Levy noise. The neighbors at the local border remain fixed.

(M5) The local relaxation is complete when every edge satisfies $l_- < l < l_+$.

Pseudocode

1. Generate an initial mesh.
2. Compute the effective time step Δt_{δ}^{eff} for the current simulation step δ .
3. Advance the dynamics by Δt_{δ}^{eff} following the adaptive Langevin Equation.
4. Pause the dynamics if an edge reaches $l < l_-$ or $l > l_+$, apply splitting and/or merging
5. Resume the global dynamics
6. Return to ‘2’ to repeat 2-5 until reaching equilibrium.

Acknowledgment

We thank Sandra Schmid for helpful discussion and comments on our manuscript. We thank Jenny (Qiongjing) Zou for software management. This work was supported by NIH grants R35GM135428 to Gaudenz Danuser and R01GM73165 to Marcel Mettlen.

References

- [1] Aurélien Villedieu, Floris Bosveld, and Yohanns Bellaïche. Mechanical induction and competence in epithelial morphogenesis. *Current opinion in genetics & development*, 63:36–44, 2020.
- [2] Dani L Bodor, Wolfram Pönisch, Robert G Endres, and Ewa K Paluch. Of cell shapes and motion: the physical basis of animal cell migration. *Developmental cell*, 52(5):550–562, 2020.
- [3] Jeremy G Carlton, Hannah Jones, and Ulrike S Eggert. Membrane and organelle dynamics during cell division. *Nature Reviews Molecular Cell Biology*, 21(3):151–166, 2020.
- [4] Ewa Sitarska and Alba Diz-Muñoz. Pay attention to membrane tension: mechanobiology of the cell surface. *Current opinion in cell biology*, 66:11–18, 2020.
- [5] Meghan K Driscoll, Erik S Welf, Andrew R Jamieson, Kevin M Dean, Tadamoto Isogai, Reto Fiolka, and Gaudenz Danuser. Robust and automated detection of subcellular morphological motifs in 3d microscopy images. *Nature methods*, 16(10):1037–1044, 2019.
- [6] Wolfgang Helfrich. Elastic properties of lipid bilayers: theory and possible experiments. *Zeitschrift für Naturforschung C*, 28(11-12):693–703, 1973.
- [7] Udo Seifert and Reinhard Lipowsky. Morphology of vesicles. *Handbook of biological physics*, 1:403–464, 1995.
- [8] Jonathan Richard Shewchuk. Delaunay refinement algorithms for triangular mesh generation. *Computational Geometry*, 22(1):21–74, 2002.
- [9] N Ramakrishnan, PB Sunil Kumar, and Ravi Radhakrishnan. Mesoscale computational studies of membrane bilayer remodeling by curvature-inducing proteins. *Physics reports*, 543(1):1–60, 2014.
- [10] Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. Mesh optimization. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 19–26, 1993.
- [11] David H Boal, Udo Seifert, and Andreas Zilker. Dual network model for red blood cell membranes. *Physical review letters*, 69(23):3405, 1992.
- [12] Mohsen Sadeghi, Thomas R Weikl, and Frank Noé. Particle-based membrane model for mesoscopic simulation of cellular dynamics. *The Journal of chemical physics*, 148(4):044901, 2018.
- [13] M Giani, WK Den Otter, and WJ Briels. Early stages of clathrin aggregation at a membrane in coarse-grained simulations. *The Journal of chemical physics*, 146(15):155102, 2017.
- [14] Lin Ma and William S Klug. Viscous regularization and r-adaptive remeshing for finite element analysis of lipid membrane mechanics. *Journal of Computational Physics*, 227(11):5816–5835, 2008.
- [15] H Askes, LJ Sluys, and BBC De Jong. Remeshing techniques for r-adaptive and combined h/r-adaptive analysis with application to 2d/3d crack propagation. In *European Congress on Computational Methods in Applied Sciences and Engineering*. ECCOMAS, 2000.
- [16] Mario Botsch, Mark Pauly, Christian Rossli, Stephan Bischoff, and Leif Kobbelt. Geometric modeling based on triangle meshes. In *ACM SIGGRAPH 2006 Courses*, pages 1–es. 2006.
- [17] Mario Botsch and Leif Kobbelt. A remeshing approach to multiresolution modeling. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 185–192, 2004.
- [18] Marion Dunyach, David Vanderhaeghe, Loïc Barthe, and Mario Botsch. Adaptive remeshing for real-time mesh deformation. In *Eurographics 2013*. The Eurographics Association, 2013.
- [19] IV Ivriessimtzis, W-K Jeong, and H-P Seidel. Using growing cell structures for surface reconstruction. In *2003 Shape Modeling International.*, pages 78–86. IEEE, 2003.
- [20] Yufei Guo and Yongqing Hai. Adaptive surface mesh remeshing based on a sphere packing method and a node insertion/deletion method. *Applied Mathematical Modelling*, 98:1–13, 2021.
- [21] Darren Engwirda. Locally optimal delaunay-refinement and optimisation-based mesh generation. 2014.
- [22] Dakshata Panchal and Deepak Jayaswal. Computational paradigms for direct triangular surface remeshing. *Computers & Graphics*, 94:87–110, 2021.

- [23] Udo Seifert, Karin Berndl, and Reinhard Lipowsky. Shape transformations of vesicles: Phase diagram for spontaneous-curvature and bilayer-coupling models. *Physical review A*, 44(2):1182, 1991.
- [24] Mark Meyer, Mathieu Desbrun, Peter Schröder, and Alan H Barr. Discrete differential-geometry operators for triangulated 2-manifolds. In *Visualization and mathematics III*, pages 35–57. Springer, 2003.
- [25] Achim Guckenberger and Stephan Gekle. Theory and algorithms to compute helfrich bending forces: A review. *Journal of Physics: Condensed Matter*, 29(20):203001, 2017.
- [26] Xin Bian, Sergey Litvinov, and Petros Koumoutsakos. Bending models of lipid bilayer membranes: Spontaneous curvature and area-difference elasticity. *Computer Methods in Applied Mechanics and Engineering*, 359:112758, 2020.
- [27] Cha Zhang and Tsuhan Chen. Efficient feature extraction for 2d/3d objects in mesh representation. In *Proceedings 2001 International Conference on Image Processing (Cat. No. 01CH37205)*, volume 3, pages 935–938. IEEE, 2001.
- [28] Jan Faix and Klemens Rottner. The making of filopodia. *Current opinion in cell biology*, 18(1):18–25, 2006.
- [29] Laurent Blanchoin, Rajaa Boujema-Paterski, Cécile Sykes, and Julie Plastino. Actin dynamics, architecture, and mechanics in cell motility. *Physiological reviews*, 94(1):235–263, 2014.
- [30] Sean D Conner and Sandra L Schmid. Regulated portals of entry into the cell. *Nature*, 422(6927):37–44, 2003.
- [31] Weria Pezeshkian, Melanie König, Tsjerk A Wassenaar, and Siewert J Marrink. Backmapping triangulated surfaces to coarse-grained membrane models. *Nature communications*, 11(1):1–9, 2020.
- [32] Hunter Elliott, Robert S Fischer, Kenneth A Myers, Ravi A Desai, Lin Gao, Christopher S Chen, Robert S Adelstein, Clare M Waterman, and Gaudenz Danuser. Myosin ii controls cellular branching morphogenesis and migration in three dimensions by minimizing cell-surface curvature. *Nature cell biology*, 17(2):137–147, 2015.
- [33] Bartłomiej Dybiec, Ewa Gudowska-Nowak, and Peter Hänggi. Lévy-brownian motion on finite intervals: Mean first passage time analysis. *Physical Review E*, 73(4):046104, 2006.
- [34] wil. icosphere. <https://www.mathworks.com/matlabcentral/fileexchange/50105-icosphere>, 2021.
- [35] Anand Banerjee, Alexander Berezhkovskii, and Ralph Nossal. Stochastic model of clathrin-coated pit assembly. *Bioophysical journal*, 102(12):2725–2730, 2012.
- [36] Khaled Khairy, JiJinn Foo, and Jonathon Howard. Shapes of red blood cells: comparison of 3d confocal images with the bilayer-couple model. *Cellular and molecular bioengineering*, 1(2):173–181, 2008.

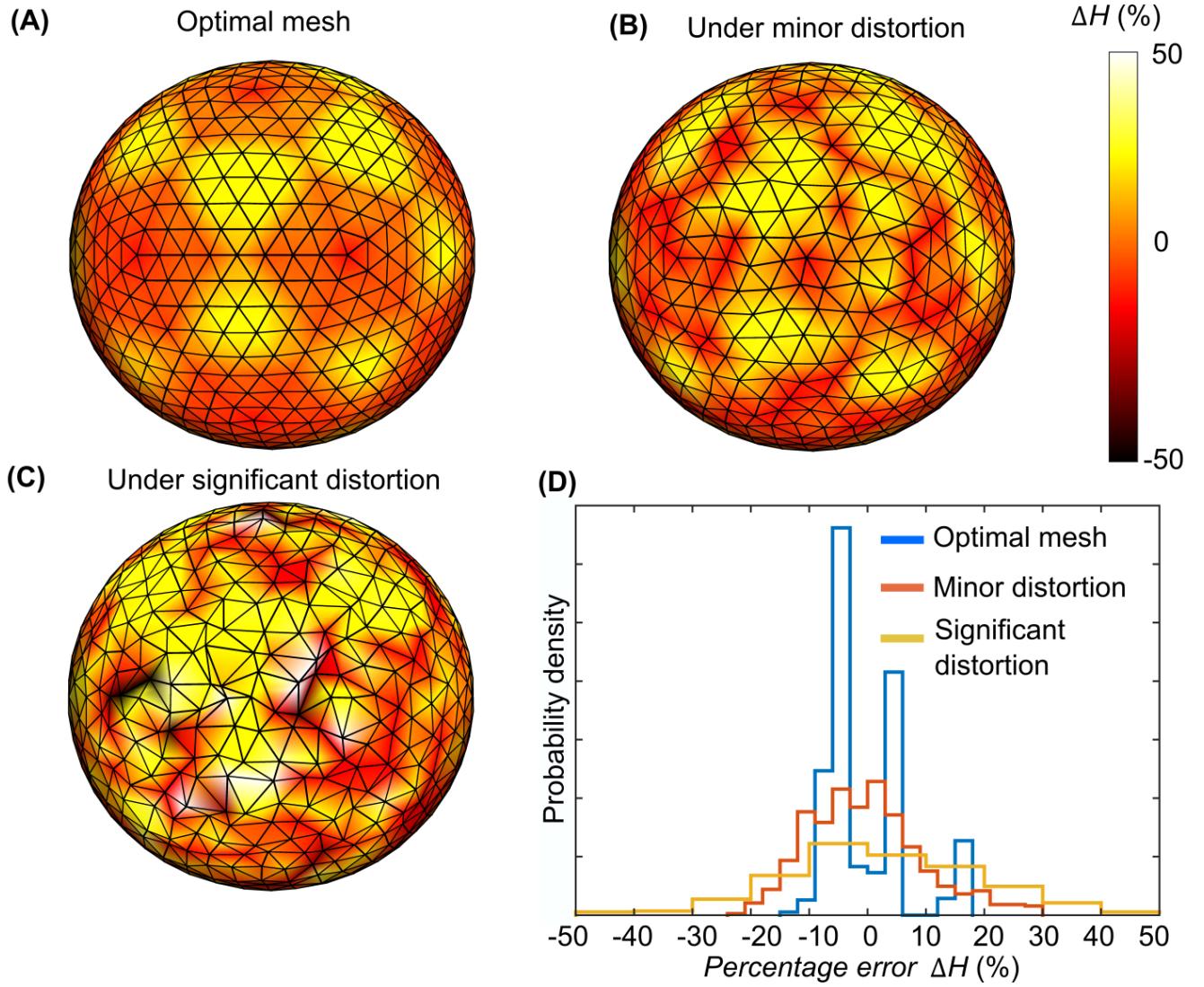


Figure 1: Error in Helfrich energy computation on meshes with variable quality.

(A) Percentage difference ΔH between mesh calculated Helfrich energy and theoretical Helfrich energy on a sphere with optimal mesh quality (mesh generated by using [34]). (B) ΔH under minor distortion to the mesh. (C) ΔH under significant distortion. (D) The probability distribution function of ΔH for (A-C). Percentage error $\Delta H = [\mathbf{H}(i) - H(i)]/H(i) \times 100\%$ at each vertex comparing discrete value of Helfrich energy $\mathbf{H}(i)$ at each vertex i (see Methods) compared to the exact value $H(i) = 8\pi\kappa/N_v$, where $8\pi\kappa$ is the total Helfrich energy of an ideal sphere and N_v is the total number of vertices.

Scale	Value	Note
<i>Energy</i> : \bar{E}	$10k_B T$	$T = 300K$
<i>Length</i> : \bar{l}	$1000nm$	
<i>Time</i> : \bar{s}	$1second$	
Parameter	Value	Reference
κ , bending modulus	\bar{E}	same order [35, 36]
C_0 , spontaneous curvature	0	This paper
μ , vertex mobility	$100\bar{l}^2 \bar{E}^{-1} \bar{s}^{-1}$	This paper
l_0 , default edge length	\bar{l}	This paper
V_0 , normalization factor of V_{in}	$0.2\bar{E}$	This paper
b , barrier height of V_{in}	0.5	This paper
r_{11} , near slope location of near barrier	$0.05\bar{l}$	This paper
r_{12} , far slope location of near barrier	$0.75\bar{l}$	This paper
r_{21} , near location of far barrier	$1.25\bar{l}$	This paper
r_{22} , far location of far barrier	$1.75\bar{l}$	This paper
k_{11} , steepness of near slope of near barrier	$50/\bar{l}$	This paper
k_{12} , steepness of far slope of near barrier	$25/\bar{l}$	This paper
k_{21} , steepness of near slope of far barrier	$25/\bar{l}$	This paper
k_{22} , steepness of far slope of far barrier	$50/\bar{l}$	This paper
k_w , steepness of wall	$100/\bar{l}$	This paper
l_{min} , location of near wall	$-0.05\bar{l}$	This paper
l_{max} , location of near wall	$2.2\bar{l}$	This paper
dl	$0.0001\bar{l}$	This paper
l_{--} , compressive BCE finish	$0.1\bar{l}$	This paper
l_- , compressive BCE trigger	$0.7\bar{l}$	This paper
l_+ extensive BCE trigger	$1.3\bar{l}$	This paper
l_{++} extensive BCE finish	$1.7\bar{l}$	This paper

Table 1: Parameter values.

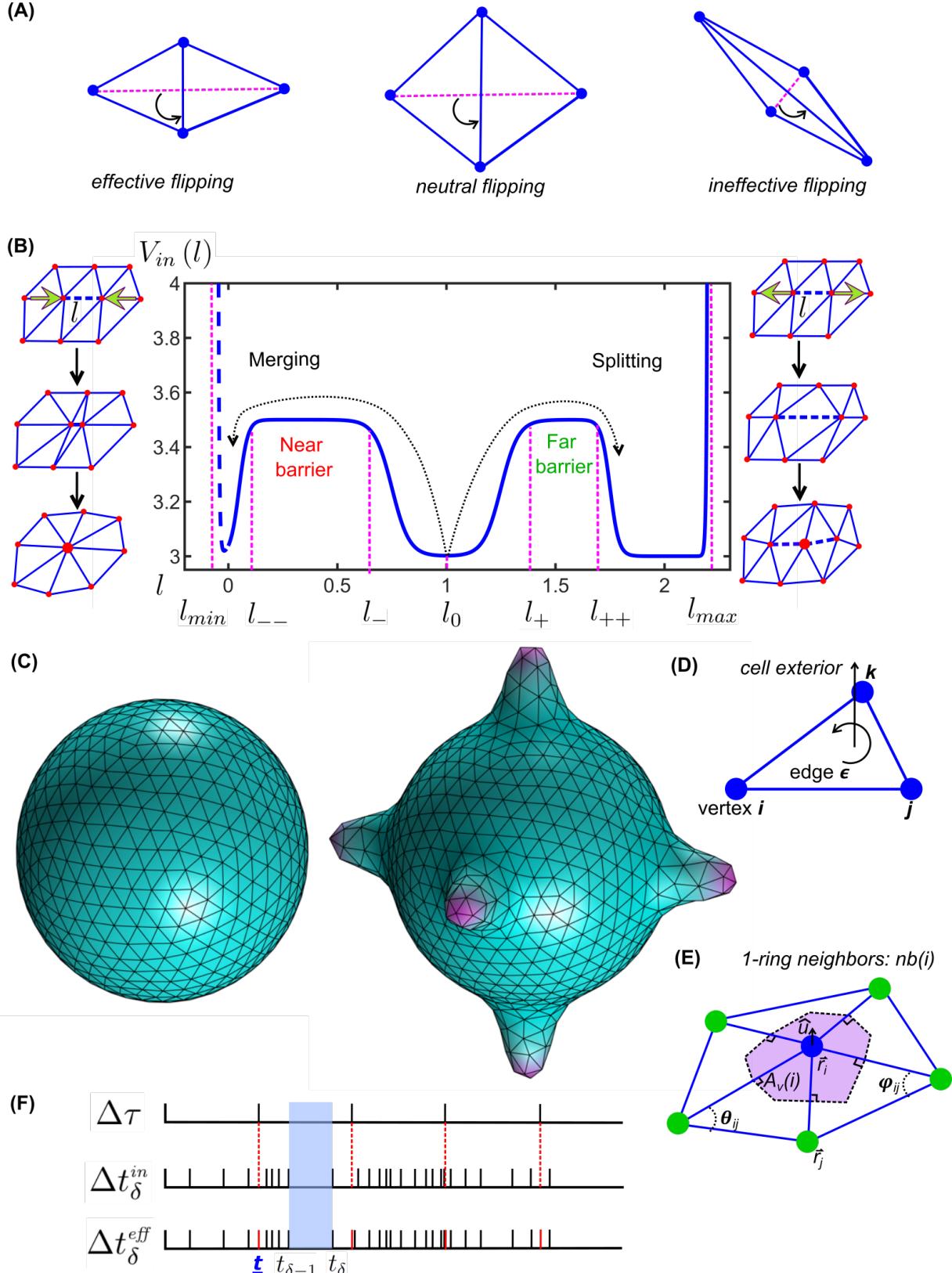


Figure 2: Effective remeshing guided by barrier crossings in a double-barrier potential.

(A) Remeshing via flipping procedures. (B) Remeshing via splitting and merging procedures guided as the barrier crossings by a potential V_{in} at indicated lengths. (C) Initial sphere of triangular meshes and high-quality triangular mesh after protrusion. (D) Triangular mesh element: right-hand rule. (E) Definition of 1-ring neighbors $nb(i)$ and Voronoi area A_v in purple [24]. \hat{u} is the unit vector in the normal direction following (D), r_i and r_j are the vectors defined by vertex i and j , and θ_{ij} and ϕ_{ij} are the two angles for computing the discrete Helfrich energy at i eq(5). (F) Integration of a constant time step $\Delta\tau$ and an adaptive time step Δt_δ^{in} in a simulation time schedule Δt_δ . Blue box indicates a time steps from $t_{\delta-1}$ to t_δ , subject to constant regular forces \vec{f}_{reg} defined at \underline{t} .

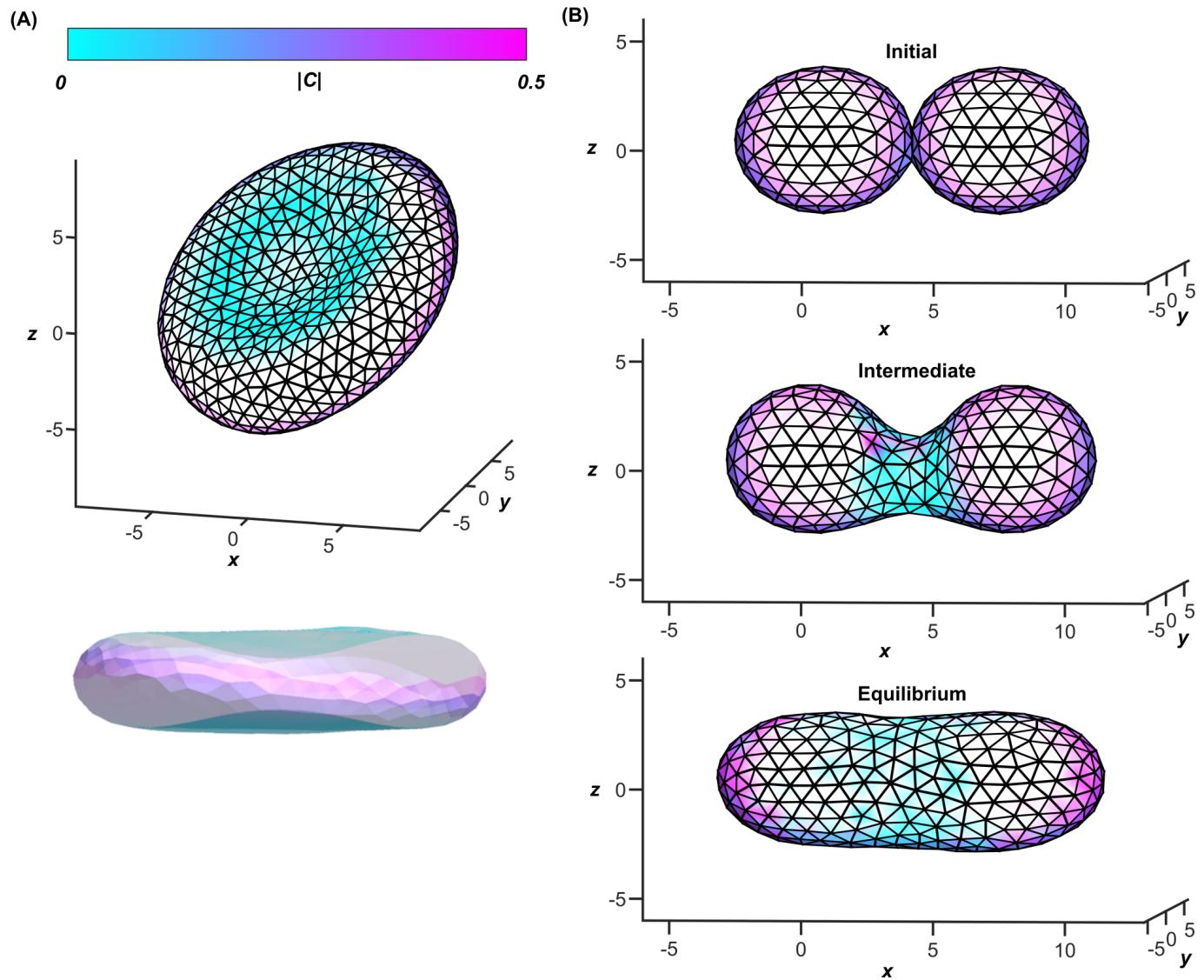


Figure 3: Morphodynamics of a biconcave red blood cell and a vesicle fusion yields high mesh quality. (A) Biconcave morphology at equilibrium. Bottom: down-sized side view. Color indicates the absolute value of the mean curvature. The size of the simulated cell is comparable to the typical size of a red blood cell at $6 - 8 \mu\text{m}$ in width. A matching video is provided in Video 1. (B) Vesicle fusion at three stages. Color scheme same as in (A). A matching video is provided in Video 2.

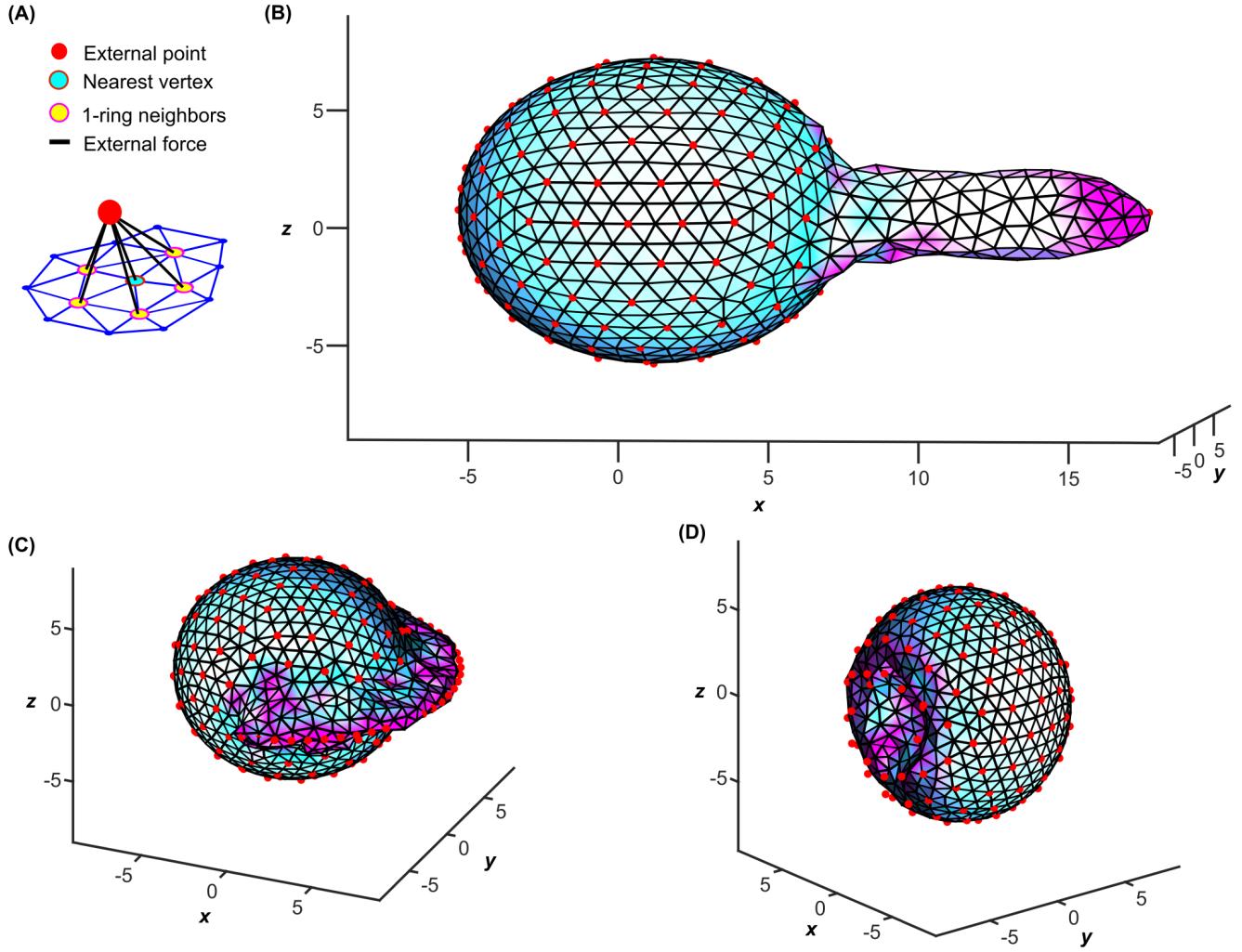


Figure 4: Applications of the proposed method to generate high-quality meshes of salient geometries under external forces. (A) Definition of external control points attracting the nearest mesh vertex i and i 's 1-ring neighbors $nb(i)$. (B) Formation of a filopodium by placement of a control point at the tip and control points holding the remainder of the globular cell in place. (C) Formation of a lamellipodium by placement of control points in a circular rim and control points holding the remainder of the globular cell in place. (D) Formation of an invagination by placement of a control point $1/3$ into the globular cell and control points holding the remainder of the cell in place. Color scheme same as in Figure 3A. Matching videos are provided in Videos 3-5.

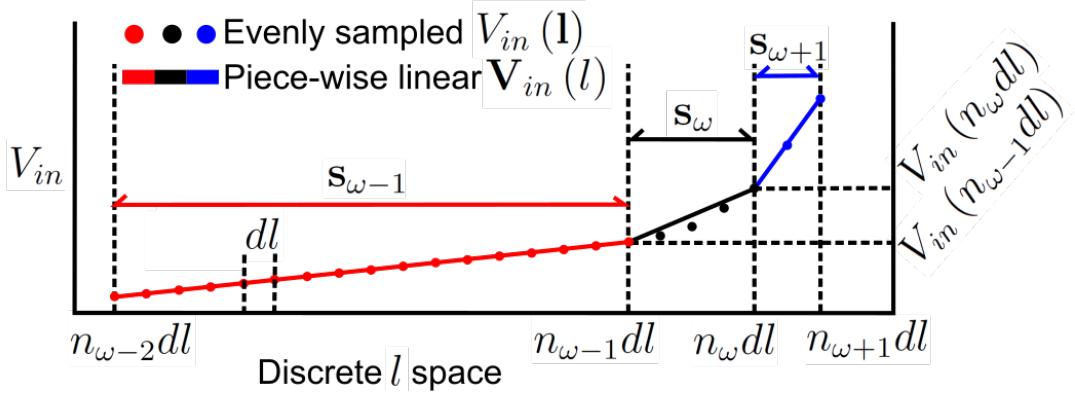


Figure S1: Linear segmentation of V_{in} .

l -space discretization and segmentation according to V_{in} : examples of complex regions of V_{in} in black (non-linear) and blue (highly varying) segments and regular regions of V_{in} in red segments.

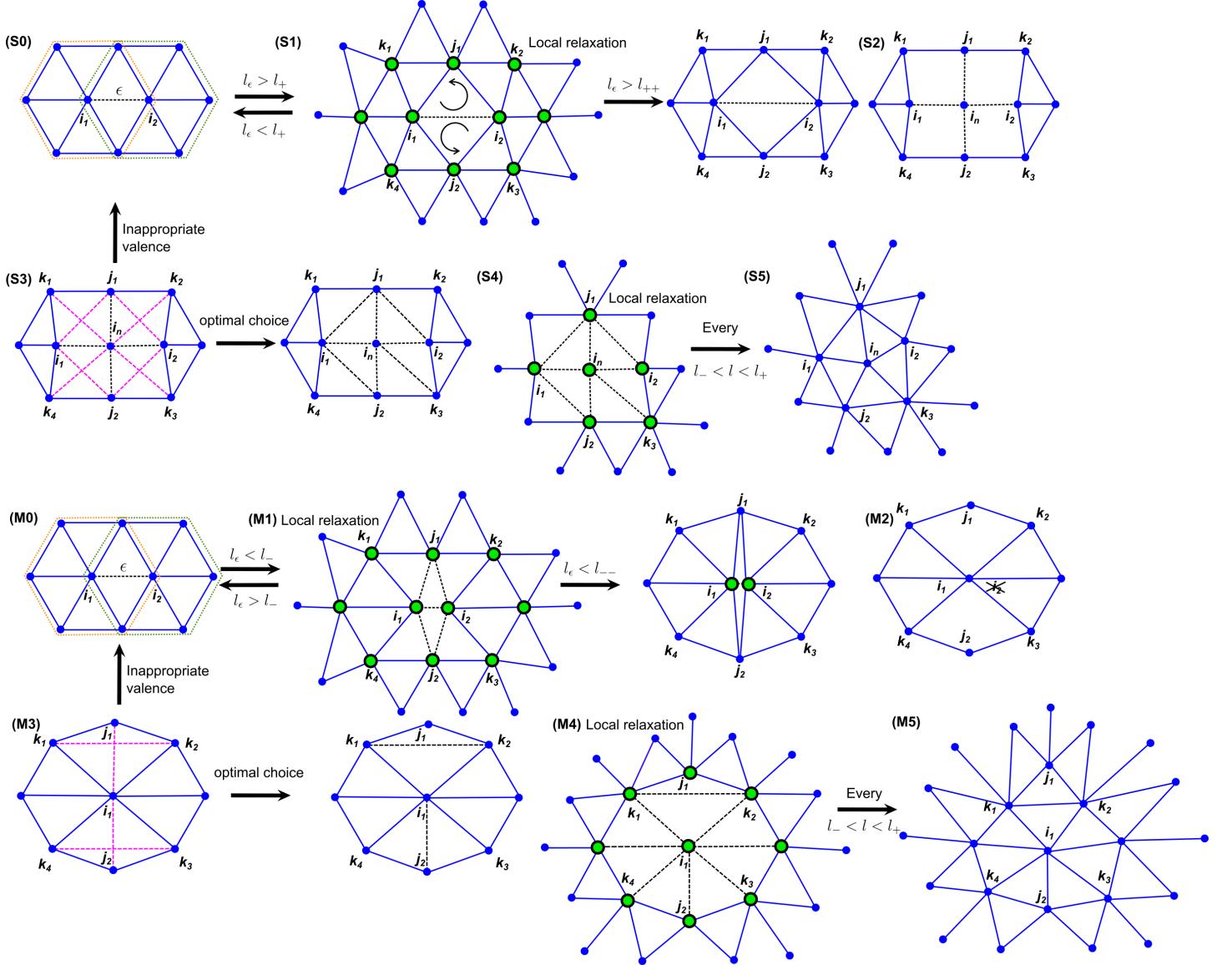


Figure S2: Edge splitting and merging steps.

Initial configuration (S0) and steps of splitting (S1-S5); and initial configuration (M0) and steps of merging (M1-M5). See text for detailed procedures.