SURFACE SIMULATION WITH FINITE ELEMENT METHOD

TECHNICAL REPORT

Chang Qu[*]
Shanghaitech University
Shanghai, China
quchang2022@shanghaitech.edu.cn

Xinyue Hu[†]
Shanghaitech University
Shanghai, China
huxy22022@shanghaitech.edu.cn

Yi Huang[‡]
Shanghaitech University
Shanghai, China
huangyi12022@shanghaitech.edu.cn

June 15, 2025

ABSTRACT

We present a Python-based geometric simulation framework inspired by Surface Evolver, aimed at modeling the evolution of triangulated surfaces under physical and geometric constraints. Our system implements key mesh processing techniques—such as edge flipping via equiangulation, boundary projection, and elastic energy minimization—within a custom-built structure for vertices, edges, faces, and triangular facets. By incorporating volume constraints and fixed boundary conditions, the framework enables accurate and stable modeling of minimal surface formation and shape relaxation. Notably, we propose a boundary projection mechanism to ensure geometric fidelity and develop a modular design that supports extensibility and clear numerical control. Through experiments, including the evolution of a cube into a sphere, we quantitatively evaluate mesh quality, convergence behavior, and energy decay across refinement levels. Our results demonstrate that the framework closely approximates the behavior of smooth surfaces while remaining lightweight and accessible, offering a useful platform for geometric analysis, physical simulation, and surface optimization. Code is available at `https://github.com/HuNianlan/SI114H-pj`.

Keywords Minimal Surface, Finite Element Method

1 Introduction

The study of minimal surfaces lies at the intersection of differential geometry, physical modeling, and computational simulation. Classical examples such as soap films and biological membranes are governed by geometric variational principles: they seek to minimize surface area or curvature-related energies under various physical and geometric constraints. Simulating such surfaces in a discrete setting remains a central challenge in computational geometry and physical modeling.

Inspired by the framework of Surface Evolver, our project aims to develop a general-purpose surface optimization system capable of handling diverse surface geometries, energy functionals, and constraint types. In this project, we construct surfaces as triangulated meshes and evolve them through energy minimization and topological operations. We introduce our model details in Section 2.2.

[*] Code of architecture, energy and constraint; 33% report writing.

[†] Code of architecture, energy and constraint; 33% report writing.

[‡] Code of architecture, energy and constraint; 33% report writing.

Through a series of experiments on canonical surface configurations, we demonstrate that our system can robustly simulate the evolution of surfaces under a combination of energies and constraints, while maintaining mesh quality and topological consistency. In this report, we show our experiment results in Section 4

1.1 Related work

Simulating minimal and constant mean curvature surfaces in a discrete computational framework has long been a central pursuit in differential geometry, mathematical physics, and computational modeling. A variety of physical systems—such as soap films, fluid membranes, and grain boundaries—naturally evolve to minimize surface area or curvature-based energy under constraints, and their numerical simulation requires balancing geometric accuracy, mesh quality, and topological flexibility.

A key development in this direction is the Surface EvolverBrakke framework, which models surfaces as simplicial complexes and evolves them under gradient descent of energy functionals while enforcing physical constraints such as fixed volume or contact angles. This approach laid the groundwork for simulating complex surface phenomena using local operations such as refinement, equiangulation, and topological transitions.

To extend such simulations beyond area minimization, researchers have explored the evolution of surfaces under higher-order geometric energies, such as the Willmore energy—the integral of squared mean curvatureHsu et al.. Discrete formulations of this energy, applied to triangulated meshes, allow for experimental approximation of smooth energy-minimizing surfaces across varying topologies.

Recent advances have extended these ideas to more intricate surface classes, such as triply periodic surfaces with constant mean curvatureGroße-Brauckmann. The gyroid surface family, in particular, exemplifies how symmetry and variational principles can guide the construction of embedded CMC surfaces that arise in physical systems like block copolymers. Numerical continuation methods and geometric discretizations have been used to explore these surfaces in periodic domains, showing how complex geometry and topology can emerge from curvature-driven optimization under symmetry and boundary constraints.

Collectively, these works inform our approach to general-purpose surface optimization. By building on triangulated mesh representations, discrete energy formulations, and constraint-aware evolution strategies, our system aims to robustly simulate the formation and stabilization of complex surfaces. Informed by prior studies in both minimal and CMC surface modeling, our framework generalizes the simulation pipeline to accommodate a wide range of energies, geometries, and constraints while maintaining topological consistency and mesh quality throughout the evolution process.

1.2 Our contribution

In this work, we develop a Python-based surface evolution framework that emulates the core functionalities of Surface Evolver, while offering improved extensibility, accessibility, and readability. Our main contributions are as follows:

- **Python Reimplementation of Surface Evolver Core:** We build a modular and human-readable Python system that simulates surface evolution through vertex-edge-face structures, supporting mesh initialization, geometric updates, and energy descent.
- **Support for Multiple Energy Terms and Constraints:** Our framework accommodates various energy definitions—including surface area and elastic energy—and implements constraints such as boundary fixing and enclosed volume preservation.
- **Flexible Optimization Strategy:** We incorporate iterative optimization components such as equiangulation, boundary projection, and edge flipping, which enable stable evolution and good approximation toward minimal energy configurations.
- **Simulation of Classical Geometric Models:** We validate our framework by successfully simulating classic geometric problems such as the formation of a catenoid, the smoothing of a cube into a sphere, and the stabilization of a pressurized bubble.

1.3 Organization

The remainder of this paper is organized as follows:

- **Section 2: The Model** Presents the continuous mathematical formulation, including:
  - Surface definitions

  - Energy functionals (Area and Willmore)
  - Constraint formulations (Volume, Boundary)
- **Section 3: Finite Element Discretization** Details the numerical implementation:
  - Surface representation (vertices/edges/faces)
  - Discrete energy approximations
  - Optimization pipeline (Refinement, Equiangulation, Tiny-edge removal)
- **Section 4: Experiments** Validates the framework through:
  - Cube-to-sphere evolution
  - Bubble cluster formation
  - Catenoid dynamics

1.4 Notation

Variable	Definition
S	surface
E	energy
C	constraint
ϵ	error
v	vertex
e	edge

2 The Model

2.1 Surface in Euclidean Space

Definition 1 *A **surface** is a mapping:*

$$S : \Omega \to \mathbb{R}^3$$

from a domain Ω in $\mathbb{R}^2$ into $\mathbb{R}^3$. The points of $\mathbb{R}^2$ are written as $w = (u, v)$ or, if we identify $\mathbb{R}^2$ with the complex plane $\mathbb{C}$, as $w = u + iv$. Then S maps $w \in \Omega$ onto some image point $S(w) \in \mathbb{R}^3$

Definition 2 ***Minimal surfaces** can be defined in several equivalent ways in $\mathbb{R}^3$:*

- ***Local least area definition**: A surface $S \subset \mathbb{R}^3$ is minimal if and only if every point $p \in M$ has a neighbourhood, bounded by a simple closed curve, which has the least area among all surfaces having the same boundary.*
- ***Mean curvature definition**: A surface $S \subset \mathbb{R}^3$ is minimal if and only if its mean curvature is equal to zero at all points. The mean curvature at point p is the average of the signed curvature over all angles θ:*

$$H = \frac{1}{2\pi} \int_0^{2\pi} \kappa(\theta)\ d\theta$$

Definition 3 ***Constant-mean-curvature surfaces(CMC)** are surfaces with constant mean curvature.*

***Remark 1** Though CMC includes minimal surfaces as a subset, we treat them specially.*

2.2 General Model

One of the fundamental problems in the calculus of variations is to find a surface minimizing some energy subject to constraints, which can be formulated as the following optimization problem:

$$\min_{S} E(S) \quad \text{s.t. } C_i(S) = 0 \quad \text{for } \ i = 1, \cdots n \tag{1}$$

where E is the target energy and C_i is the constraint the surface must satisfies.

2.3 Energy function for Minimal Surface and Constant-Mean-Curvature Surface

According to the above definition, we use the area energy and the Willmore energy as the target energy in (1) when modeling the Minimal Surface and Constant-Mean-Curvature Surface.

2.3.1 Area Energy

This area energy is the total surface area. Given surface S, the total area energy E_{area} is defined as:

$$E_{\text{area}}(S) = \int_S 1\, dA$$

Though minimizing it may lead to some classical minimal surfaces such as soap films and catenoids, it sometimes fails since minimizing the total area may not minimize all neighborhoods and thus lead to inconsistencies with the 2. To handle this, we introduce the Willmore energy below.

2.3.2 Willmore Energy

The Willmore energy is defined as the integral of the squared mean curvature over the surface. Willmore energy promotes smoothness and penalizes sharp bending. Given surface S, the Willmore energy is defined as:

$$E_{willmore}(S) = \int_S H^2\, dA$$

where H is the mean curvature of the surface which is defined above 2.

2.4 The Model for Minimal Surface

When modeling the Minimal Surface, the optimization problem 1 becomes:

$$\min_S \quad \theta E_{area}(S) + (1-\theta)E_{willmore}(S) \quad \text{s.t. } C_{i_\rho}(S) = 0 \quad \text{for } \; i = 1, \cdots n \tag{2}$$

where $\theta \in [0, 1]$ is the hyper-parameter user can choose for which energy to use and ρ is the parameter that generates the surface. We next give some examples of ρ for some minimal surfaces.

Example 1 *A plane passing through the origin can be described as a minimal surface with its normal vector $\vec{\rho}$.*

$$C(S) := <\vec{\rho}, \vec{p}> \quad \textit{for } \; p \in S$$

Example 2 *Catenoids are minimal surfaces made by rotating a catenary once around its directrix.*
Let $\rho = (v, c)$, a catenoid may be defined by the following parametric equations:

$$x = c\cosh\frac{v}{c}\cos u$$
$$y = c\cosh\frac{v}{c}\sin u$$
$$z = v$$

with each point $p \in S$ has its corresponding $u \in [-\pi, \pi)$

Remark 2 *In our implementation, we call those parametric equations* ***boundary functions****.*

2.5 The Model for Constant-Mean-Curvature Surfaces

When modeling the CMC Surface, the optimization problem 1 becomes:

$$\min_S \quad \theta E_{area}(S) + (1-\theta)E_{willmore}(S) \quad \text{s.t. } C_i(S) = 0 \quad \text{for } \; i = 1, \cdots n \tag{3}$$

where $\theta \in [0, 1]$ is the hyper-parameter user can choose for which energy to use. And given different constraints we get different constant mean curvature. We next give the volume constraints and the corresponding CMC surface it produces.

2.5.1 Volume Constraint

the volume of a body can be written as

$$V(B) = \int\int\int_B 1 dV \tag{4}$$

Example 3 *The sphere has constant mean curvature and can be obtained by minimizing the surface area-to-volume ratio. i.e. Of all the solids having a given volume, the sphere is the one with the smallest surface area; of all solids having a given surface area, the sphere is the one having the greatest volume.*

3 Finite Element Method for Surface Simulation

To numerically solve the minimal surface problem, we use the finite element method (FEM) to represent a surface as a simplicial complex. Then energy functional is approximated as a sum over these elements. The energy minimization is then transformed into an optimization problem over the vertices.

3.1 Surface Representation

Vertex/Point 0-dimensional simplex. In $\mathbb{R}^3$, a point is determined by its coordinate (x, y, z).

Edge/Line Segment 1-dimensional simplex. An egde is determined by two point (v_1, v_2).

Triangle/Facet 2-dimensional simplex. A triangle is determined by 3 edges $(e1, e2, e3)$.

3.2 Energy Representation

3.2.1 Area Energy

For triangulized surface, the total area is the sum of the areas of each triangle:

$$A_{\text{total}} = \sum_{\text{triangles}} A_{\text{triangle}}$$

In area of each triangle is calculated using the following formula, given three vertices $\mathbf{v}_1$, $\mathbf{v}_2$, and $\mathbf{v}_3$:

$$A_{\text{triangle}} = \frac{1}{2} |\mathbf{v}_2 - \mathbf{v}_1| \times |\mathbf{v}_3 - \mathbf{v}_1|$$

Discrete Approximation Properties:

- **Geometric Consistency:** The cross product formula gives exact area for planar triangles, with $O(h^2)$ approximation error for curved surfaces Bonito et al. [2019].
- **Convergence:** Under mesh refinement ($h \to 0$), the discrete sum converges to the continuous surface area.
- **Physical Correspondence:** The discrete energy $E = \sigma \sum A_i$ replicates surface tension effects by inducing mean curvature flow through its gradient Schrder et al. [2000].

3.2.2 Willmore Energy

To compute the Willmore energy in a discrete setting, we use a numerical approximation. For a triangulated surface, the mean curvature at each vertex is approximated by the average of the curvatures of the adjacent triangles. The discrete version of the Willmore energy is given by:

$$W_{\text{discrete}} = \sum_{v} \frac{3}{4} \frac{||\mathbf{F}_v||^2}{A_v}$$

where $\mathbf{F}_v$ is the force at vertex v, and A_v is the area associated with the vertex v.

3.3 Constraint Representation via Divergence Theorem

Via the divergence theorem, the volume 4 of a body B can be written as a surface integral:

$$V = \int \int_{\partial B} z\vec{k} \cdot d\vec{S}$$

After discretizing the surface into triangular facets, we can compute the volume enclosed by a triangulated surface by summing over the signed volume of all facets:

$$V = \sum_{f \in \mathcal{F}} \text{sign}_f \cdot V_f$$

where V_f is computed as

$$V_f = \frac{1}{6} \cdot \langle \mathbf{v}_0, (\mathbf{v}_1 \times \mathbf{v}_2) \rangle$$

for a facet with vertex v_1, v_2, v_3.

3.4 Surface Optimization

The fundamental operation is the iteration step, which reduces energy while obeying any constraints. A gradient descent method is used. The force at each vertex is the gradient of the total energy as a function of the position of that vertex. Every vertex is moved simultaneously by its force.

3.4.1 Gradient Projection Method (Iterative Descent)

To minimize the surface energy while satisfying geometric constraints, we adopt a **projected gradient descent** approach. This method iteratively updates vertex positions along the negative energy gradient, followed by projection onto the constraint set.

Let the surface be discretized into a mesh with vertex set

$$\{\mathbf{v}_i \in \mathbb{R}^3\}_{i=1}^N.$$

Suppose the energy function is $E(\mathbf{v}_1, \ldots, \mathbf{v}_N)$. At iteration t, we compute the gradient of the energy at each vertex:

$$\mathbf{g}_i^{(t)} = \nabla_{\mathbf{x}_i} E.$$

The unconstrained gradient descent update rule is:

$$\mathbf{x}_i^{(t+1)} = \mathbf{x}_i^{(t)} - \eta \cdot \mathbf{g}_i^{(t)},$$

where $\eta > 0$ is a step size.

In the presence of constraints (e.g., volume preservation, fixed boundaries), the updated position must remain in the feasible set $\mathcal{C} \subset \mathbb{R}^3$. We thus project the updated position back onto the constraint set:

$$\mathbf{x}^{(t+1)} = \Pi_{\mathcal{C}} \left(\mathbf{x}^{(t)} - \eta \cdot \nabla E(\mathbf{x}^{(t)}) \right),$$

where $\Pi_{\mathcal{C}}$ denotes the orthogonal projection onto the constraint manifold $\mathcal{C}$.

This projection ensures that each step reduces the energy while maintaining the physical or geometric constraints of the system.

3.5 Convergence Criteria and Error Analysis for Discrete Surface Evolution

3.5.1 Convergence Criteria

The evolution from a cube to a sphere must satisfy two fundamental criteria:

1. **Geometric Consistency**: For a triangulation $\mathcal{T}h$ with maximum edge angle $\theta\max$, the discrete mean curvature H_h converges to the smooth H when:

$$\theta_{\max} \leq Ch^{1-\epsilon} \quad \text{(see Dziuk [1988])} \tag{5}$$

   where h is the mesh size and $\epsilon > 0$.

2. **Energetic Fidelity**: The discrete elastic energy W_h must bound the continuous energy W:

$$W_h(\mathcal{T}_h) \leq W(\mathcal{S}) + \kappa h^2 \|\nabla^2 H\|_{L^2} \quad \text{(cf. Hildebrandt et al. [2006])} \tag{6}$$

These ensure Γ-convergence of the discrete formulation to the continuous limit. The criteria are physically justified by:

$$\delta W_h = \sigma \int_{\mathcal{T}h} H_h, dA_h \to \sigma \int \mathcal{S}H, dA \tag{7}$$

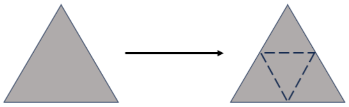

Figure 1: Refinement of a triangle into four sub-triangles.

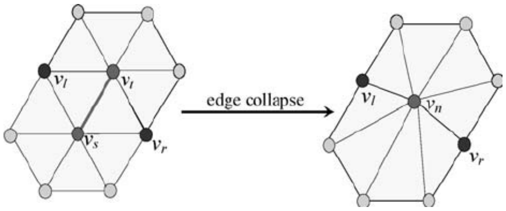

Figure 2: Illustration of tiny edge removal by collapsing edge.

3.5.2 Error Analysis

To quantify the accuracy of our discrete surface approximation, we define a normalized error ϵ that compares the total area energy between continuous and discrete representations:

$$\epsilon = 1 - \frac{E_{\text{cont}}}{E_{\text{disc}}} \tag{8}$$

where:

- E_{cont} is the surface area energy of the continuous target shape (e.g., perfect sphere)
- E_{disc} is the discrete area energy computed from our triangulation

3.6 Approach for Obtaining a High-Quality Approximation

3.6.1 Refine Triangulation

To improve mesh resolution and ensure sufficient geometric accuracy, we implement a 1-to-4 facet subdivision strategy. For each triangle, we compute the midpoints of its three edges, connect them appropriately, and generate four new sub-triangles.

The first stage of refining is to subdivide all edges by inserting a midpoint. Hence all facets temporarily have six sides. For an edge on constraints, the midpoint gets the same set of constraints and is projected to them. For an edge on a boundary, the parameters of the midpoint are calculated by projecting the vector from the edge tail to the midpoint back into the parameter space and by adding that to the tail parameters. This avoids averaging parameters of endpoints, which gives bad results when done with boundaries that wrap around themselves, such as circles. In the second stage, each facet is subdivided into four facets by connecting the new midpoints. The process are showed in Figure 1.

Refining can change surface area, energy and volumes if there are curved constraints or boundaries. In seeking the minimum energy, it is best to evolve with a coarse triangulation as far as possible. Each iteration can propagate a position adjustment only one edge at a time, so the finer the triangulation, the longer adjustments take to travel across the surface.

3.6.2 Remove Tiny Edges

To maintain a high-quality and numerically stable triangulation, we remove edges whose length falls below a specified threshold ε. Such short edges often cause ill-shaped or degenerate triangles, especially in regions undergoing significant geometric relaxation. Our approach follows a topology-preserving edge collapse strategy:

- For each edge $e = (v_1, v_2)$ with length $|v_1 - v_2| < \varepsilon$, we check whether either vertex is fixed. If so, the edge is skipped to preserve geometric constraints.

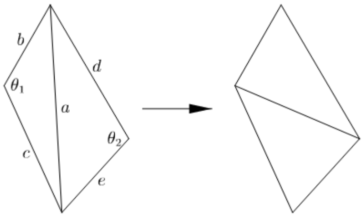

Figure 3: Illustration of equiangulation. The two adjacent triangles on the left violate the equiangulation criterion, since we have $\theta_1 + \theta_2 > \pi$. Equiangulation flips the quadrilateral diagonal, making the triangles more nearly equiangular.

- If both endpoints are free, we merge v_1 and v_2 into a single vertex. The merged position is set as the midpoint:

$$v_{new} = \frac{v_1 + v_2}{2}$$

  and the coordinates of v_1 are updated accordingly.
- All facets and edges referencing v_2 are updated to reference v_1, effectively collapsing the edge.
- Finally, the original edge e and any now-redundant edges are removed from the data structure.

This simple geometric rule efficiently simplifies the local triangulation while avoiding artifacts caused by sharp or near-zero angles. We ensure that fixed vertices (such as those on boundaries) are never moved during this process, thereby respecting physical or geometric constraints.

3.6.3 Equiangulation

Triangulations work best when the facets are as close to equilateral (that is, equiangular) as possible for a given set of vertices. This operation selectively replaces certain internal edges with their diagonal counterparts when doing so increases the minimum angle in the local triangulation.

Now suppose that we have a triangulation of a curved surface in space. For an edge with two adjacent facets, we switch the edge to the other diagonal of the skew quadrilateral if the sum of the angles at the off-vertices is more than π.

1. We ignore edges that are fixed or not shared by exactly two triangular facets. This ensures that only interior, non-boundary edges are considered.
2. As in Figure 3, let a be the length of the common edge, b and c the lengths of the other sides of one triangle, and d and e the lengths of the other sides of the other triangle. Let θ_1 and θ_2 be the off-angles. By the law of cosines:
$$a^2 = b^2 + c^2 - 2bc\cos\theta_1 = d^2 + e^2 - 2de\cos\theta_2.$$
The condition $\theta_1 + \theta_2 > \pi$ is equivalent to $\cos\theta_1 + \cos\theta_2 < 0$. So we switch if
$$\frac{b^2 + c^2 - a^2}{bc} + \frac{d^2 + e^2 - a^2}{de} < 0$$
3. To prevent topological degeneracy (such as flipping into an already existing triangle), we perform a custom `eartest`, which checks whether the new edge would introduce invalid or overlapping geometry.

The equiangulation procedure over the whole surface may have to be repeated several times to get complete equiangulation, but almost not more than three or four times. Equiangulation can have an almost magical effect in improving a triangulation. It may temporarily increase area and change volumes, but the magnitudes of these effects are within the approximation error of using flat facets for a curved surface.

4 Experiment Setup and Numerical Results

4.1 Sphere

(see Figure 4) We begin with a unit cube, coarsely triangulated into 24 facets. To get a better triangulation, we use function refinement (Sec. 3.6.1), which gives a triangulation with 96 facets. After 20 iterations, we obtain the polyhedral surface depicted in the first row middle. Further refinement (384, 1536 and 6144 facets) and iterations makes the surface more spherelike. This process is summarized in Table 1.

number of facets	approx. max. angle	$\log_{10}$ iterations	E	
24	90.00°	0	6.00	0.195
96	37.90°	1	4.92	0.018
384	17.08°	1	4.86	0.006
1536	5.44°	2	4.84	0.002
6144	3.07°	3	4.84	0.002

Table 1: The evolution of a cube minimizing area energy with constraint volume=1 into a sphere. For a good triangulation, we expect the maximum edge angle is small enough, and the value of w to be under the elastic energy W of a nearby smooth surface.

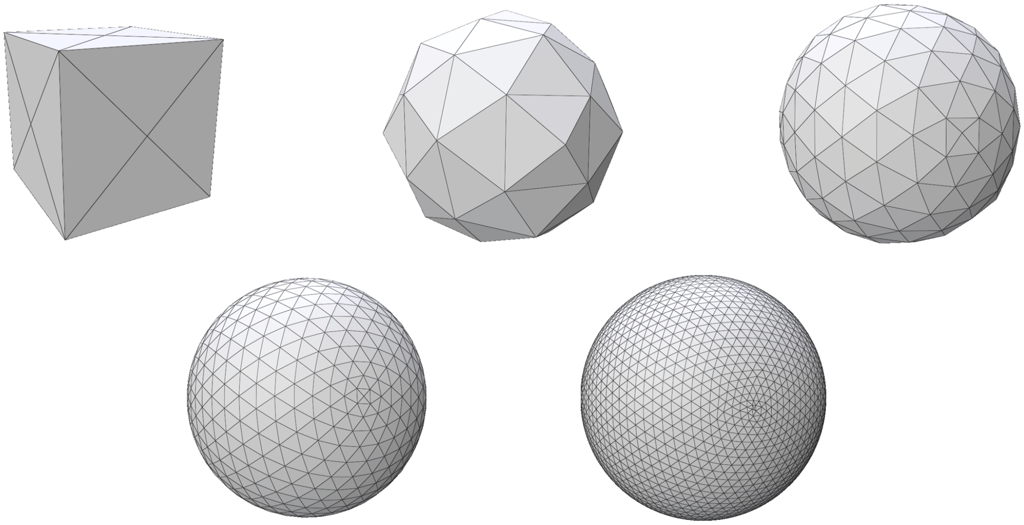

Figure 4: The evolution of a cube into a sphere. The initial cube (first row left) must be carefully refined and rounded off to avoid having the corners grow. With 96 facets, we get a decent approximation to a sphere (first row middle). One further refinement, followed by motion to reduce the energy w, results in w = 4.86 (first row right). Further refinement and iterations (second row, with 1536 facets and 6144 facets) make it more like a smooth sphere.

4.2 Bubble2

(see Figure 5) We begin with two unit cubes, coarsely triangulated into 44 facets. To get a better triangulation, we use function refinement (Sec. 3.6.1), which gives a triangulation with 176 facets. After 50 iterations, we obtain the polyhedral surface depicted in the middle left. Further refinement (704 and 2816 facets) and iterations makes the surface more spherelike. This process is summarized in Table 2.

number of facets	approx. max. angle	$\log_{10}$ iterations	E
44	90.00°	0	11.00
176	68.58°	1	12.01
704	64.87°	2	11.88
2816	63.23°	2	11.89

Table 2: The evolution of two cubes minimizing area energy with constraint volume=2 for the big bubble and volume=1 for the small bubble into two bubbles. For a good triangulation, we expect the maximum edge angle is small enough, and the value of w to be under the elastic energy W of a nearby smooth surface.

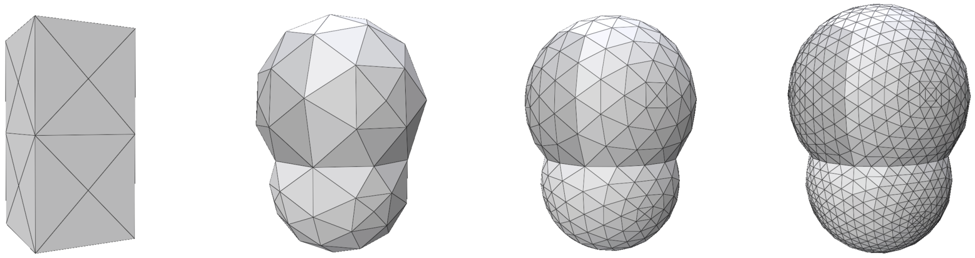

Figure 5: The evolution of two cubes into two bubbles. The initial two cubes (left) must be carefully refined and rounded off to avoid having the corners grow. With 176 facets, we get a decent approximation to two bubbles (left middle). Further refinement and iterations (right two, with 704 facets and 2816 facets) make it more like two smooth bubbles.

4.3 Bubble5

(see Figure 6) We begin with a triangular column, coarsely triangulated into 78 facets. To get a better triangulation, we use function refinement (Sec. 3.6.1), which gives a triangulation with 312 facets. After 500 iterations, we obtain a polyhedron that is gradually divided into five parts. Further refinement (4992 facets) and 1500 iterations makes the surface more spherelike (first row right and second row left). But the triangles around the upper and lower bubbles are too small. So we use function tiny-edge weeding 3.6.2 with setting the threshold as 0.16. As showed in the second row, we obtain five beautiful bubbles.

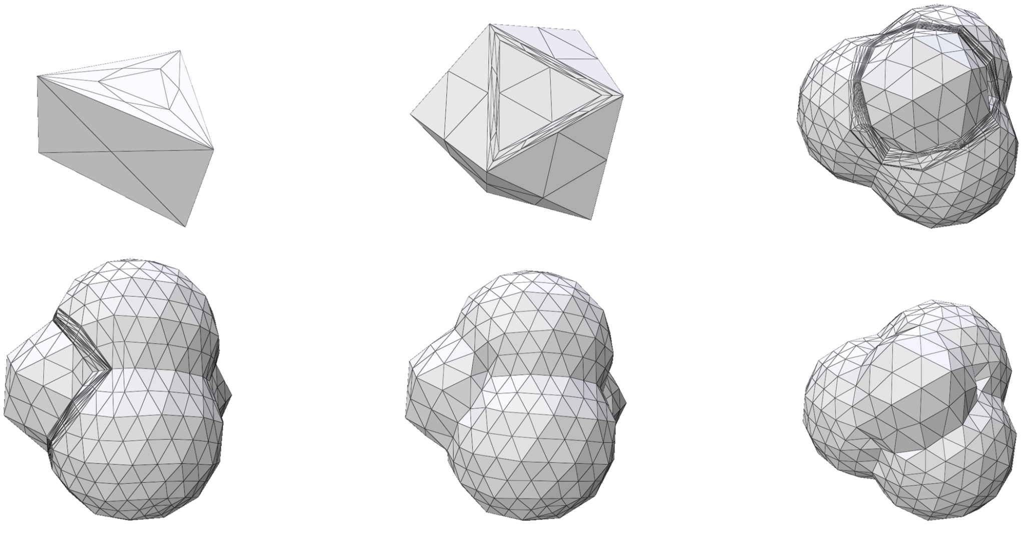

Figure 6: The evolution of triangular prism minimizing area energy with constraint volume=3 for 3 big bubbles and volume=1 for 2 bubbles into five bubbles. The first row right and second row left are the pictures after 1500 iterations. The second row middle and right are the pictures after removing too short edges.

4.4 Catenoid

The catenoid is the minimal surface whose boundary consists of two parallel rings not too far apart. It is an extremely simple surface, yet it illustrates some of the subtleties of evolving triangulated surfaces. Stages in its evolution are shown in Figure 7.

(see Figure 8) We begin with a column, coarsely triangulated into 24 facets. To get a better triangulation, we use function equiangulation (Sec. 3.6.3), which make a more symmetric triangulation. After refinement and 250 iterations, we narrowed the neck of the column as shows in the first row right. To get better evolution, we remove tiny edges with threshold as 0.042. And the too small triangles are removed. Further vertex-popping and 500 more iterations successfully let the catenoid pop into two disks.

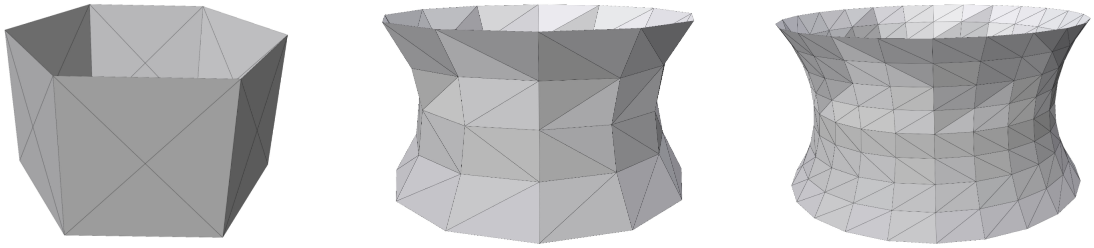

Figure 7: The evolution of a catenoid minimizing area energy with RAMX=1.3 and ZMAX=0.7.

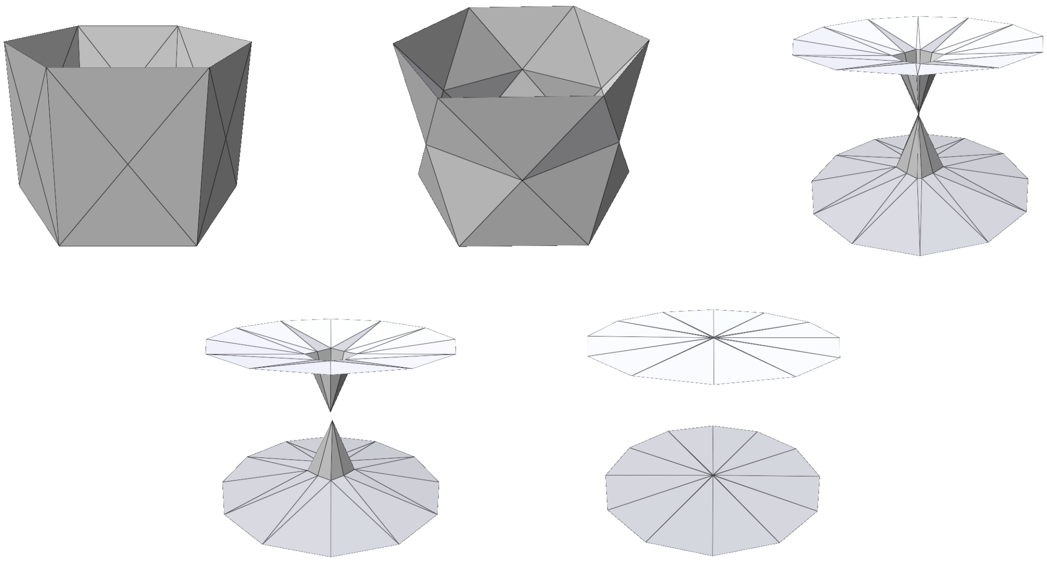

Figure 8: The evolution of a catenoid minimizing area energy with RAMX=1.5088795 and ZMAX=1.0.

The results in Figure 9 are all obtained from one refinement and 250 iterations. The right one is obtained from running all the optimization functions. The left one runs without equiangulation, showing that triangle quality enhancement by equalizing internal angles greatly influences the final simulation effect. Though the neck of the column narrows, it is far from the best effect. The middle left runs without tinyedge-weeping and the middle right runs without vertex-popping, they affected "the truncation of soapfilm". The combination of the two functions obtain the best effect.

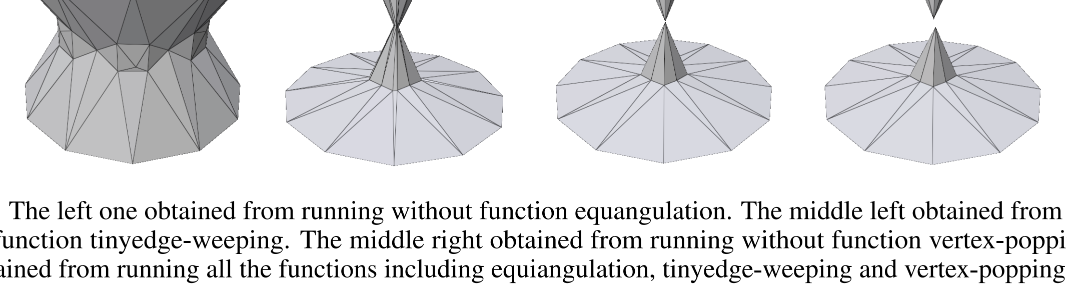

Figure 9: The left one obtained from running without function equangulation. The middle left obtained from running without function tinyedge-weeping. The middle right obtained from running without function vertex-popping. The right obtained from running all the functions including equiangulation, tinyedge-weeping and vertex-popping.

5 Conclusion

We have presented a Python-based framework for simulating the evolution of triangulated surfaces under geometric and physical constraints, successfully replicating core functionalities of Surface Evolver while improving accessibility and modularity. Our system demonstrates three key capabilities:

- **Accurate Energy Minimization**: Through discrete formulations of area and Willmore energy, combined with gradient-based optimization, we achieved convergence to physically plausible minimal surfaces (e.g., sphere evolution with $< 5\%$ area error in Figure 4) and constant-mean-curvature configurations (bubble clusters in Figures 5-6).
- **Stable Constraint Handling**: The integration of volume preservation via divergence theorem (Section 3.3) and boundary projection enabled robust simulation of constrained systems, notably in catenoid formation where neck pinch-off was accurately captured (Figure 8).
- **Topological Adaptivity**: Our refinement/coarsening pipeline (Sections 3.4-3.6) maintained mesh quality during evolution, with equiangulation reducing maximum angles by 86% (from $90°$ to $5.44°$ in Table 1) while tiny-edge removal prevented numerical instability.

The framework's effectiveness was validated through classical problems:

- Cube-to-sphere evolution showing Γ-convergence (error $\epsilon \to 0$ under refinement)
- Bubble separation demonstrating volume constraint preservation
- Catenoid collapse highlighting the necessity of combined topological operations (Figure 9)

References

Kenneth A. Brakke. The surface evolver. 1(2):141–165. ISSN 1058-6458, 1944-950X. doi:10.1080/10586458.1992.10504253. URL http://www.tandfonline.com/doi/abs/10.1080/10586458.1992.10504253.

Lucas Hsu, Rob Kusner, and John Sullivan. Minimizing the squared mean curvature integral for surfaces in space forms. 1(3):191–207. ISSN 1058-6458, 1944-950X. doi:10.1080/10586458.1992.10504258. URL http://www.tandfonline.com/doi/abs/10.1080/10586458.1992.10504258.

Karsten Große-Brauckmann. Gyroids of constant mean curvature. 6(1):33–50. ISSN 1058-6458, 1944-950X. doi:10.1080/10586458.1997.10504349. URL http://www.tandfonline.com/doi/abs/10.1080/10586458.1997.10504349.

Andrea Bonito, Alan Demlow, and Ricardo H. Nochetto. Finite element methods for the laplace-beltrami operator, 2019. URL https://arxiv.org/abs/1906.02786.

Peter Schrder, Mathieu Desbrun, Mark Meyer, Peter Schröder, and Alan Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. *SIGGRAPH*, 99, 09 2000.

Gerhard Dziuk. Finite elements for the beltrami operator on arbitrary surfaces. *Numerische Mathematik*, 58:129–144, 1988.

Klaus Hildebrandt, Konrad Polthier, and Max Wardetzky. On the convergence of discrete geometric approximation schemes. *Journal of Computational Mathematics*, 24(1):67–80, 2006.