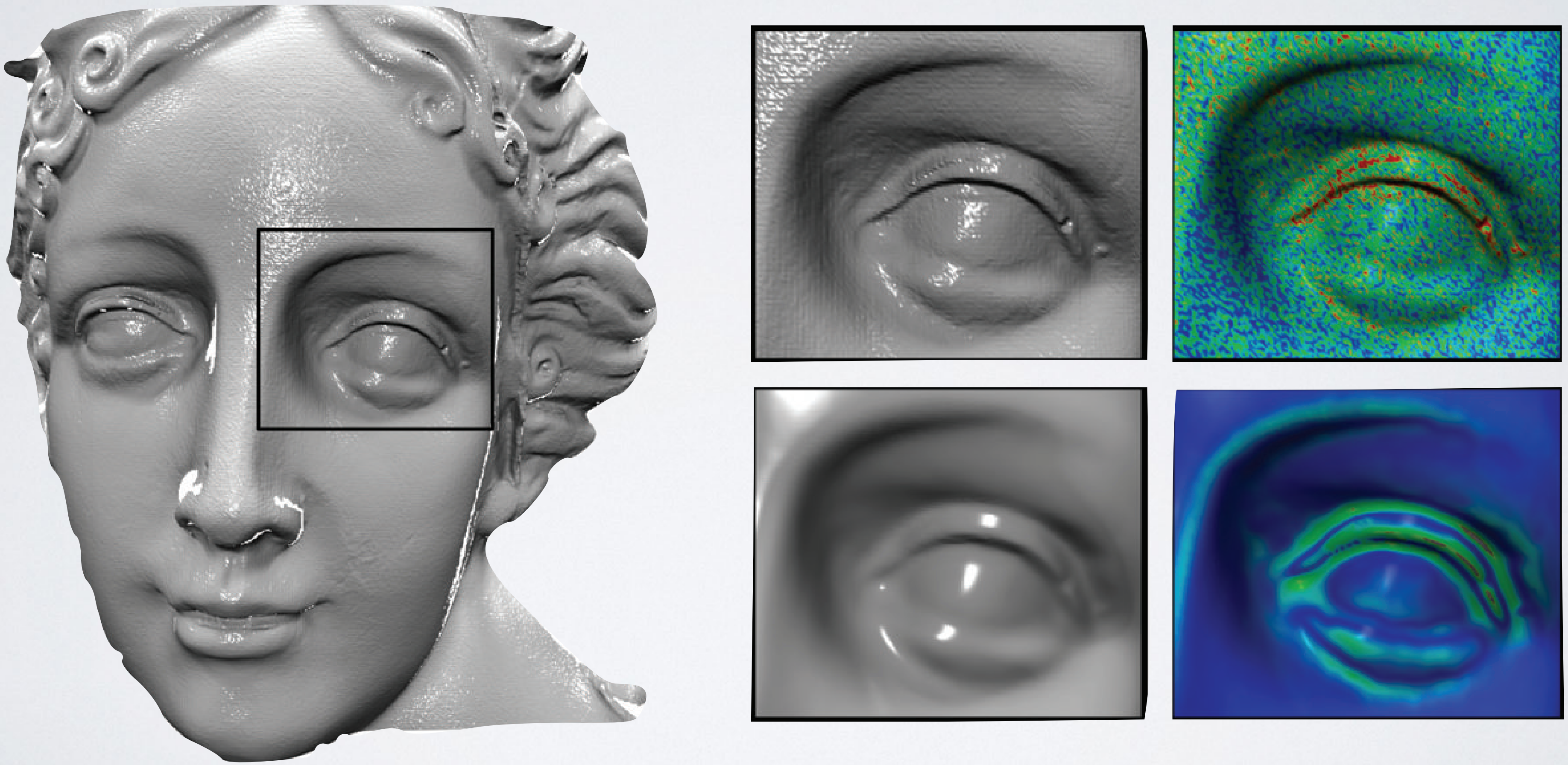
MESH SMOOTHING AND FAIRING

Lecture

NOISE AND SMOOTHING

- Polygonal meshes acquired through remote sensing techniques (e.g., range scanning) are affected by measurement noise
- Noise produces artifacts in the form of details at high frequency, making the surface somehow “rough”
- Smoothing techniques can be applied to attempt removing noise, thus making the surface smoother
- Details cannot have a scale smaller than that of noise, and details at that scale are smoothed out together with noise

NOISE AND SMOOTHING



NOISE AND SMOOTHING

- The mesh can be viewed as a two-dimensional vector signal defined over a manifold:

$$f : M \longrightarrow \mathbb{R}^3 \qquad f(v_i) = \mathbf{x}_i$$

where $\mathbf{x}_i$ denotes the position of vertex v_i in space and f is extended by linear interpolation to the rest of the mesh

- Under a signal-processing point of view, smoothing can be viewed as a filtering process of the signal f

DIFFUSION FLOW

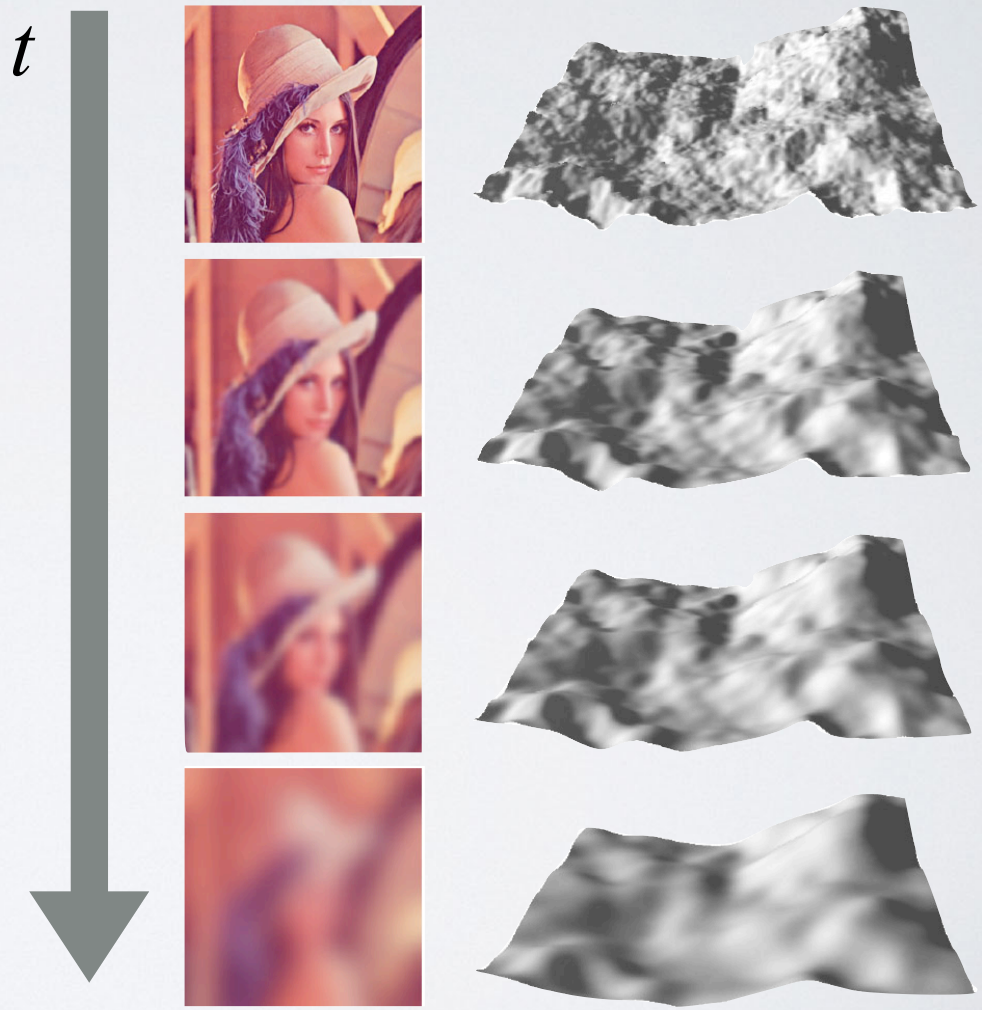
- We consider the *diffusion equation* (a.k.a. *heat equation*) describing the evolution of a signal over time

$$\frac{\partial f(\mathbf{x},t)}{\partial t} = \lambda \Delta f(\mathbf{x},t)$$

- A function f obeying to the above equation becomes smoother and smoother for increasing values of t
- $f(\mathrm{x},0)$ is the function at its initial state
- parameter λ sets the speed at which the function is smoothed

DIFFUSION FLOW

- widely used to blur images and smooth terrain surfaces
- build a ***scale space*** describing the evolution of data through time under the blurring/smoothing process



DIFFUSION FLOW

- the diffusion equation is a Partial Differential Equation
- we discretize it both in space and in time:
  - sample f at mesh vertices: $\mathbf{f}(t) = (f(v_1, t), \ldots, f(v_n, t))^T$
  - divide time in discrete steps of uniform width h

$$\frac{\partial f(v_i, t)}{\partial t} \approx \frac{f(v_i, t+h) - f(v_i, t)}{h}$$

DIFFUSION FLOW

- We obtain a system of $\boldsymbol{n}$ equations:

$$f(v_i, t+h) = f(v_i, t) + h\lambda\Delta f(v_i, t)$$

- In matrix notation:

$$\mathbf{f}(t+h) = \mathbf{f}(t) + h\lambda\mathbf{L}\mathbf{f}(t)$$

where $\mathbf{L}$ is the Laplacian matrix

DIFFUSION FLOW

- Explicit Euler integration: resolve the system by iterative substitution for small time step h

$$\mathbf{f}(t+h) = \mathbf{f}(t) + h\lambda\mathbf{L}\mathbf{f}(t)$$

- Implicit Euler integration: resolve the following linear system

$$(\mathbf{I} - h\lambda\mathbf{L})\mathbf{f}(t+h) = \mathbf{f}(t)$$

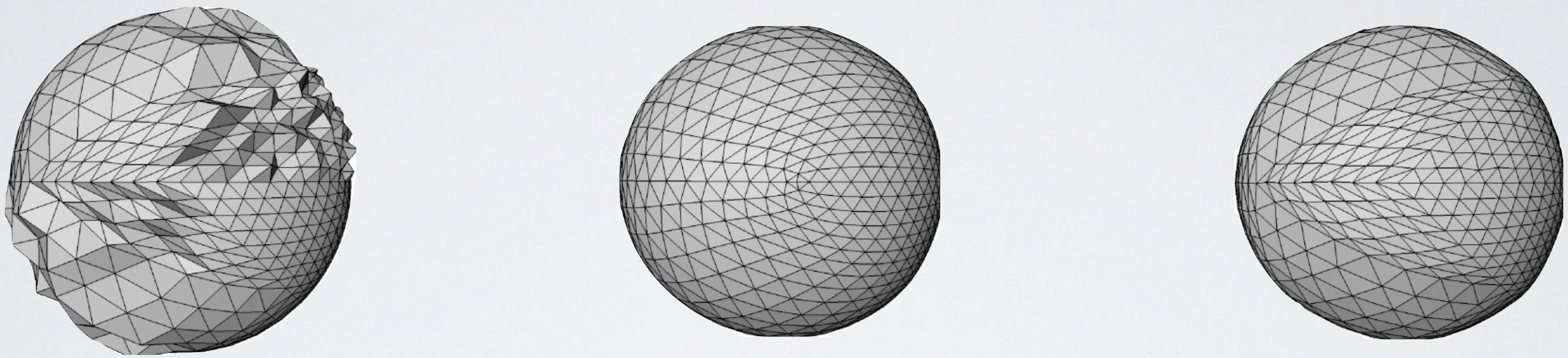
the system is very large but sparse

DIFFUSION FLOW

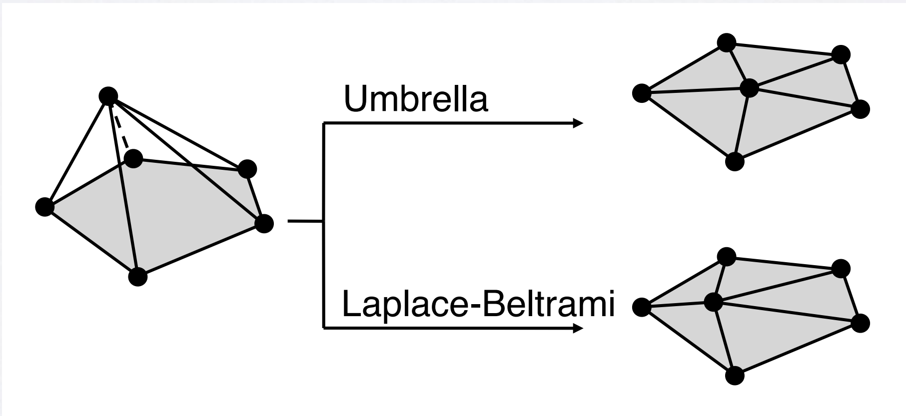
- Smoothing is performed by applying the method to the function $\mathbf{x}$ of coordinates of vertices
- With explicit integration: $\mathbf{x}_i \leftarrow \mathbf{x}_i + h\lambda\Delta\mathbf{x}_i$
- with the correct Laplace-Beltrami operator: $\Delta\mathbf{x} = -2H\mathbf{n}$

  vertices move in their normal direction by an amount proportional to the mean curvature
- With the uniform Laplacian (umbrella), vertices move towards the barycenter of their neighbors (Laplacian smoothing)

DIFFUSION FLOW



Original Umbrella Laplace-Beltrami



DIFFUSION FLOW

- Higher order diffusion flow can be used to achieve better results:

$$\frac{\partial f(\mathbf{x},t)}{\partial t} = \lambda \Delta^k f(\mathbf{x},t)$$

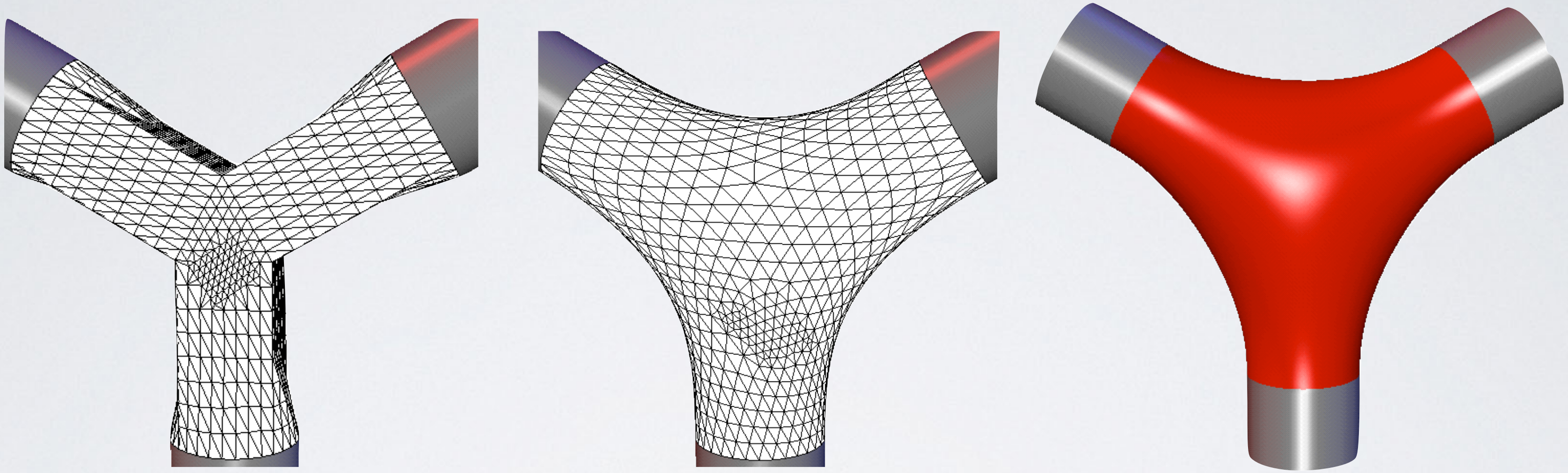
where $\Delta^k f = \Delta(\Delta^{k-1} f)$

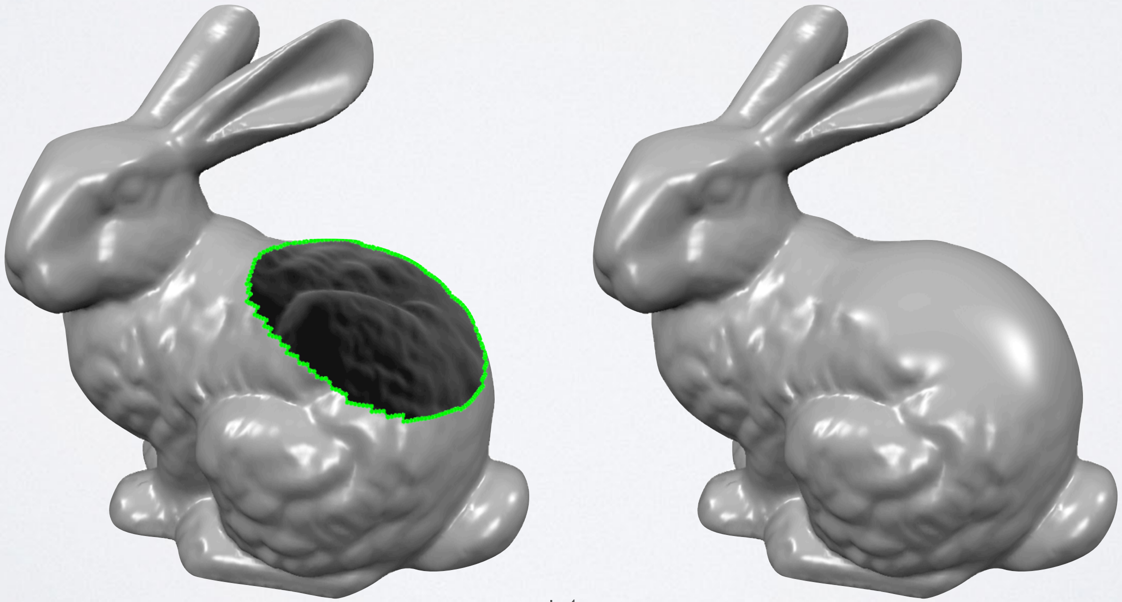
- In the discrete case: $\Delta^k f = \mathbf{L}^k f$
- The Laplacian matrix becomes less sparse at each next power, yielding higher computational cost
- Bi-Laplacian smoothing (k=2) is a good compromise between computational efficiency and smoothing quality

FAIRING

- Fairing has the purpose to compute surfaces that are *as smooth as possible*
- Actual measure of smoothness depends on application
- *Principle of simplest shape*: the surface should be free of any unnecessary details or oscillations
- General method:
  - fixed topology
  - boundary constraints (fixed position for vertices at the boundary)
  - minimize an energy depending on the position of vertices

FAIRING





FAIRING

- *Membrane energy*: measures area

$$E_M(\mathbf{x}) = \iint_\Omega \sqrt{\det(\mathbf{I})}\mathrm{d}u\mathrm{d}v$$

highly non-linear, thus difficult to minimize

- Surrogate (linearization): *Dirichlet energy*

$$\tilde{E}_M(\mathbf{x}) = \iint_\Omega ||\mathbf{x}_u||^2 + ||\mathbf{x}_v||^2\mathrm{d}u\mathrm{d}v$$

FAIRING

- Minimization of energy functional is studied with *calculus of variations*
- It can be proved that the Dirichlet energy is minimized by the function that satisfies the *Laplace equation*:

$$\mathbf{L}\mathbf{x} = \mathbf{0}$$

- Boundary conditions fix the position of some of the unknowns
- The system is sparse and, under suitable manipulations, symmetric and positive definite
- Efficient solvers can be used (e.g., *cholmod*)

FAIRING

- *Thin-plate energy*: measures curvature

$$E_{TP}(\mathbf{x}) = \iint_\Omega \kappa_1^2 + \kappa_2^2 \,\mathrm{d}u\mathrm{d}v$$

- Linearization:

$$\tilde{E}_{TP}(\mathbf{x}) = \iint_\Omega ||\mathbf{x}_{uu}||^2 + 2||\mathbf{x}_{uv}||^2 + |\mathbf{x}_{vv}||^2 \,\mathrm{d}u\mathrm{d}v$$

- Solved by the bi-Laplacian system: $\mathbf{L}^2\mathbf{x} = 0$

FAIRING

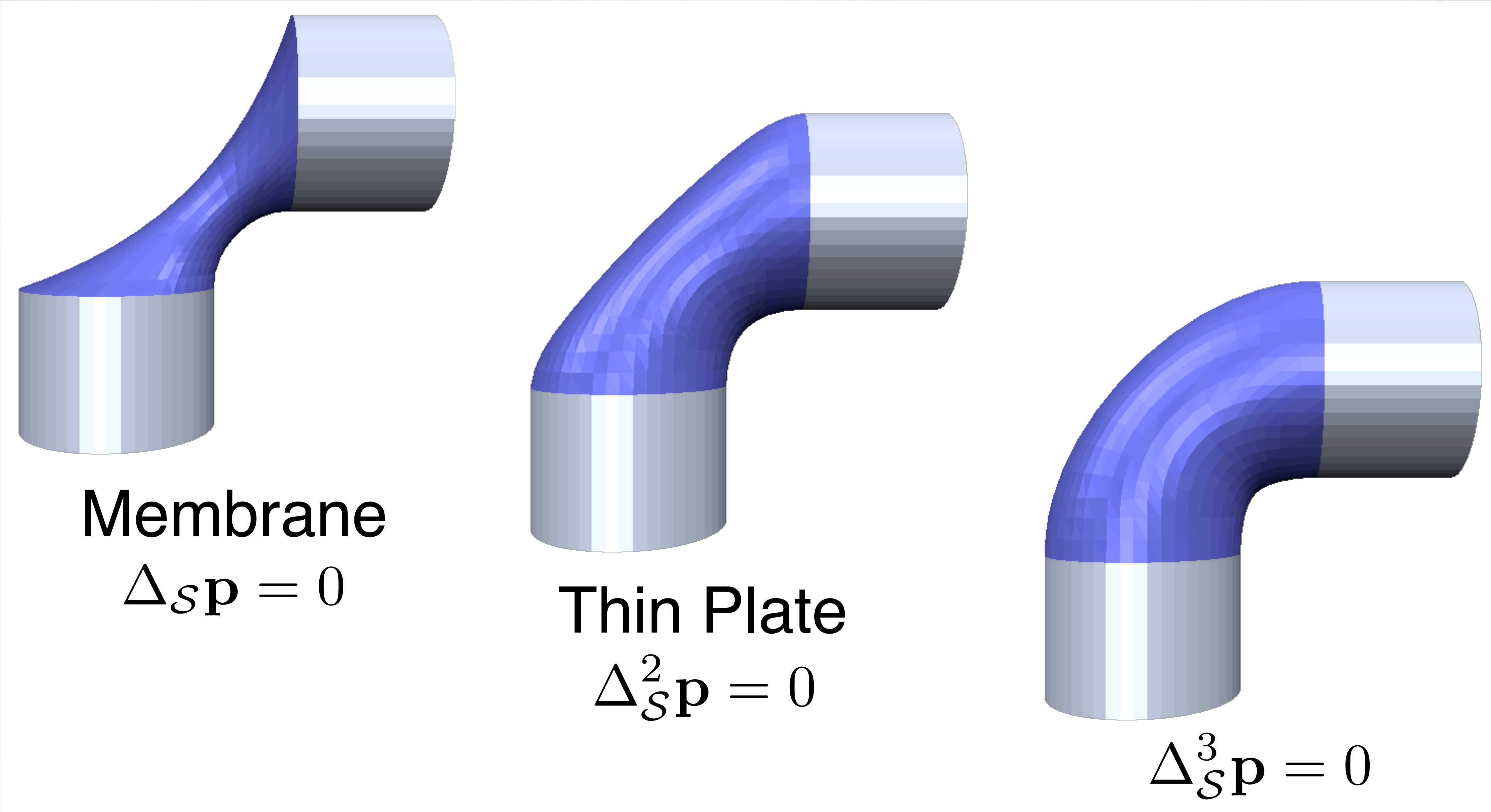
- *Higher-order energy* measuring variation of curvature

$$E_{TP}(\mathbf{x}) = \iint_{\Omega} \left(\frac{\partial \kappa_1}{\partial \mathbf{t}_1}\right)^2 + \left(\frac{\partial \kappa_2}{\partial \mathbf{t}_2}\right)^2 \, \mathrm{d}u\mathrm{d}v$$

- can be also linearized and solved by the tri-Laplacian system:

$$\mathbf{L}^3\mathbf{x} = 0$$

FAIRING


Membrane
$\Delta_{\mathcal{S}}\mathbf{p} = 0$
Thin Plate
$\Delta^2_{\mathcal{S}}\mathbf{p} = 0$
$\Delta^3_{\mathcal{S}}\mathbf{p} = 0$

RELATION BETWEEN FAIRING AND DIFFUSION FLOW

- A fair surface satisfying $\mathbf{L}^k \mathbf{x} = 0$ is a steady state for the flow

$$\frac{\partial f(\mathbf{x}, t)}{\partial t} = \lambda \Delta^k f(\mathbf{x}, t)$$

- Thus, fair surfaces are *as smooth as possible*
- Explicit integration of the Laplacian flow is equivalent to one Jacobi iteration to solve the related Laplace equation