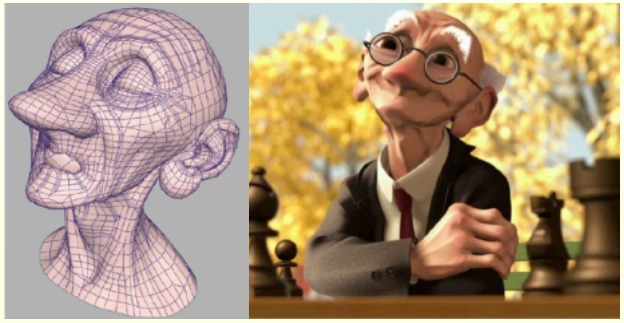
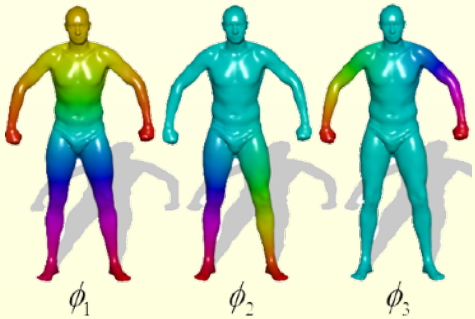
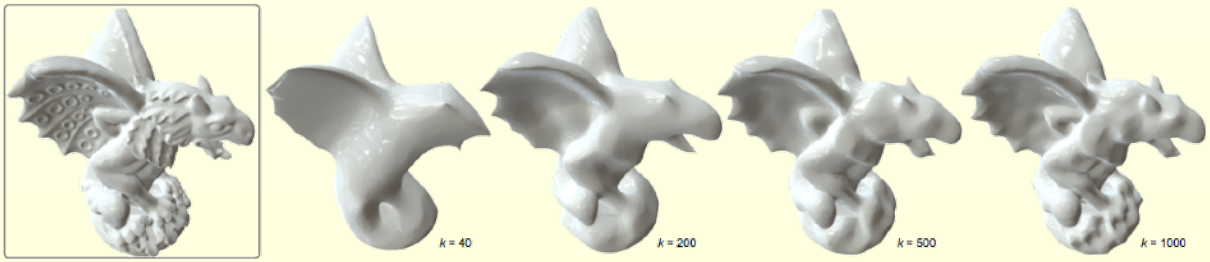
Geometry Foundations
Laplace-Beltrami Operator



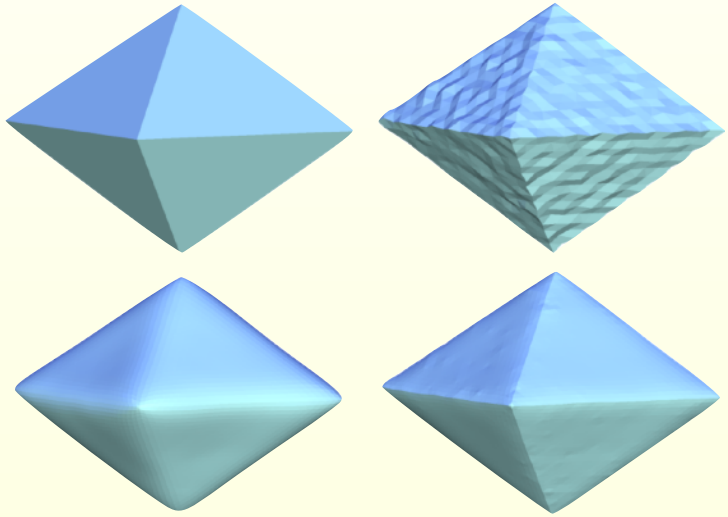
CS233
Anastasia Dubrovina
Postdoc, Guibas Lab

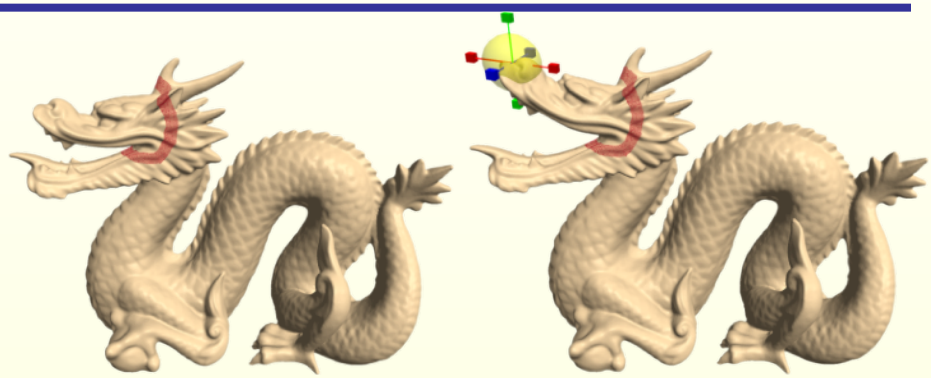


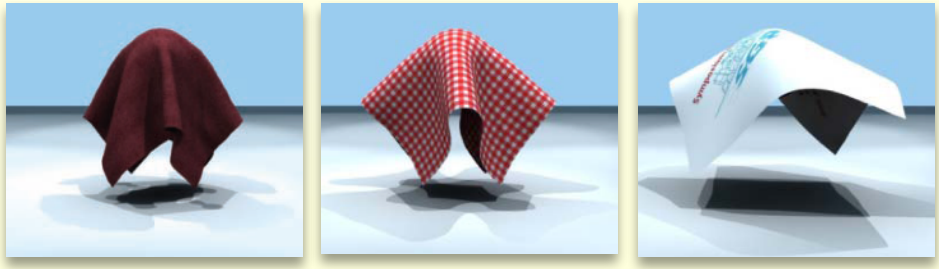


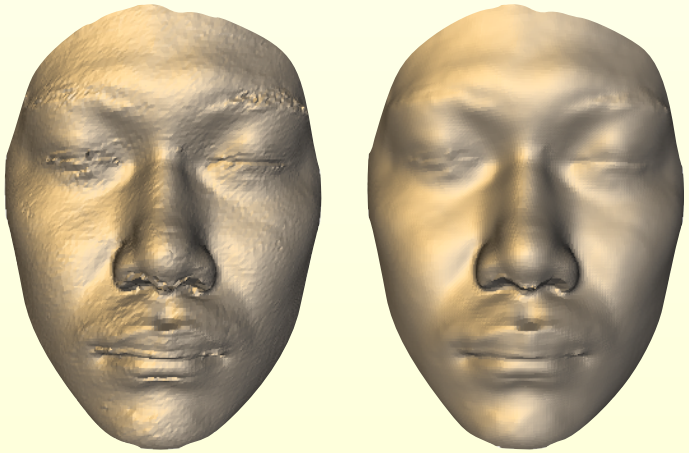
slides credits: Olga Diamanti, Olga Sorkine-Hornung, Daniele Panozzo, Chengcheng Tang, Justin Solomon

Geometry Foundations: Discrete Differential Geometry





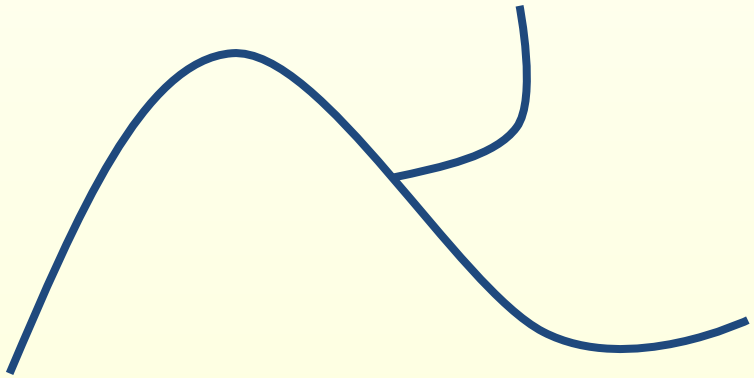




slides credits: Olga Diamanti, Daniele Panozzo

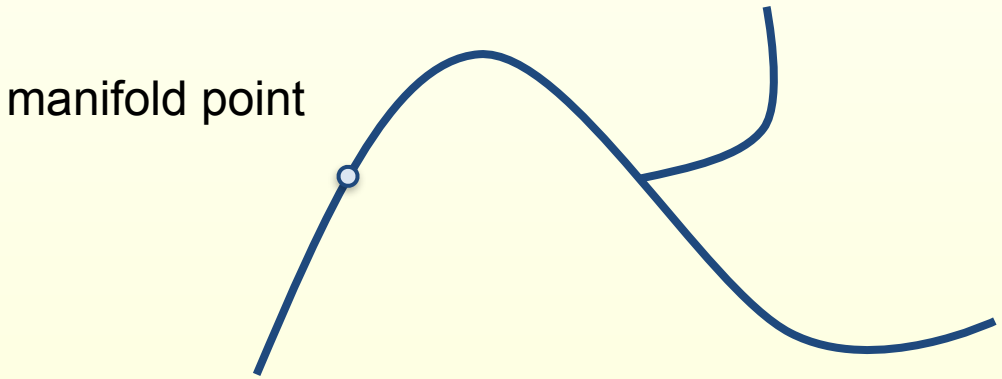
Differential Geometry Basics

- Geometry of manifolds
- Properties that can be discovered by local observation: point + neighborhood



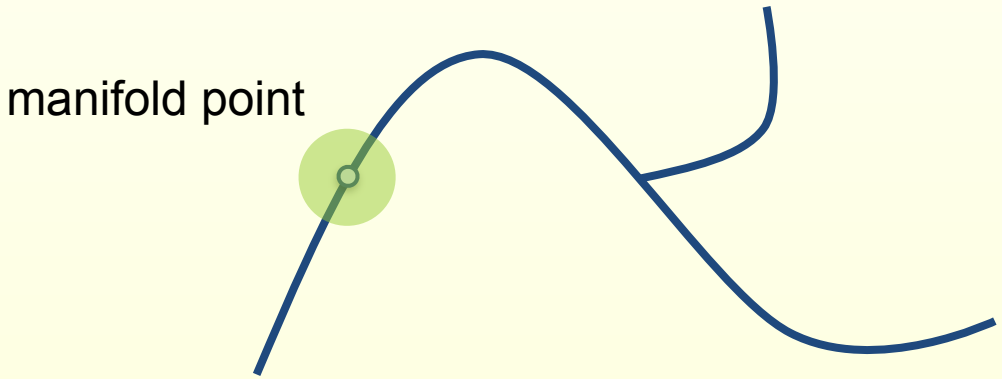
Differential Geometry Basics

- Geometry of manifolds
- Properties that can be discovered by local observation: point + neighborhood



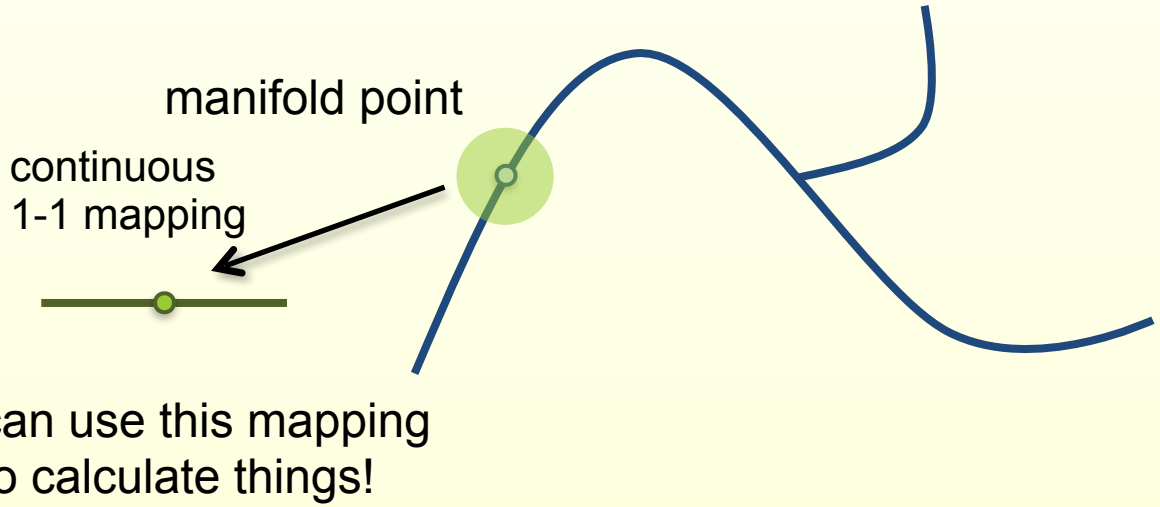
Differential Geometry Basics

- Geometry of manifolds
- Properties that can be discovered by local observation: point + neighborhood



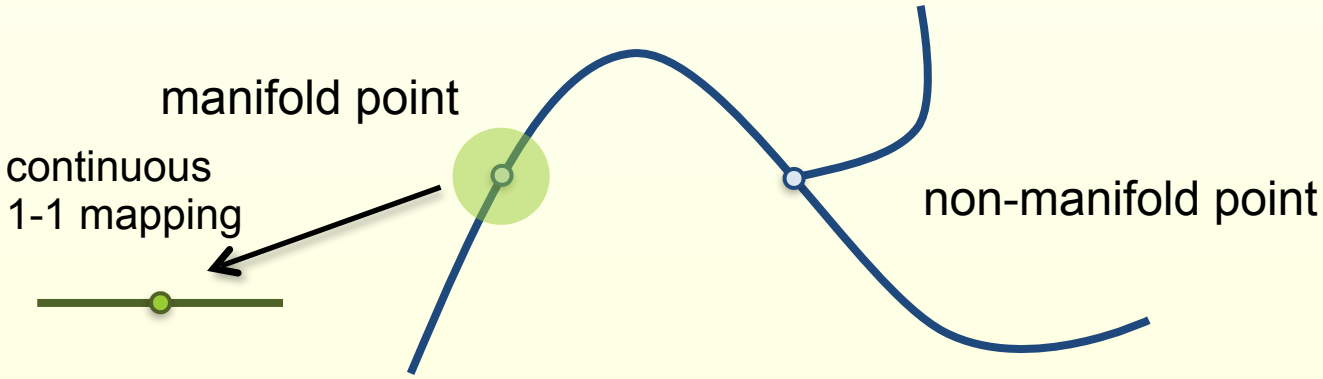
Differential Geometry Basics

- Geometry of manifolds
- Properties that can be discovered by local observation: point + neighborhood



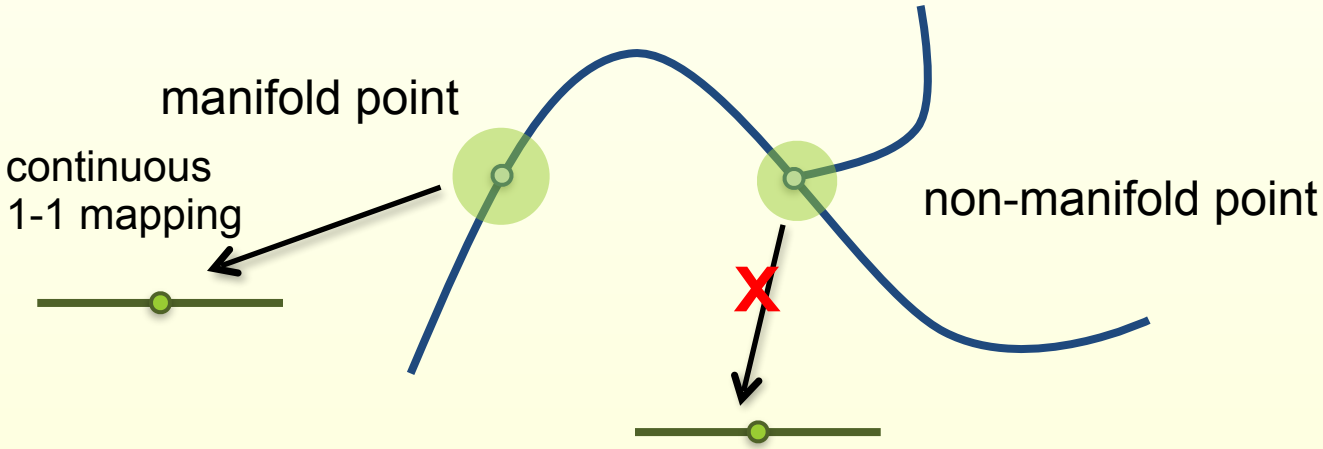
Differential Geometry Basics

- Geometry of manifolds
- Properties that can be discovered by local observation: point + neighborhood



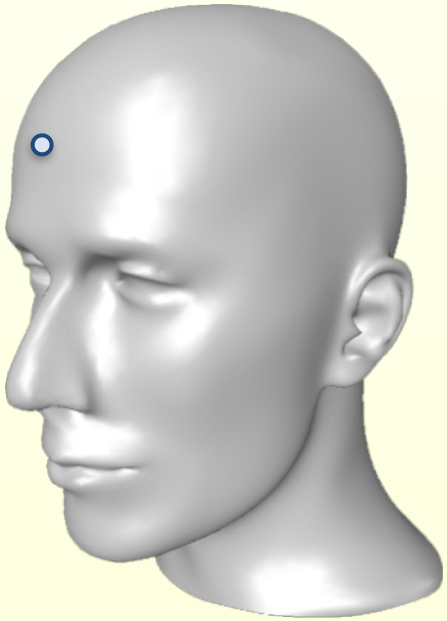
Differential Geometry Basics

- Geometry of manifolds
- Properties that can be discovered by local observation: point + neighborhood



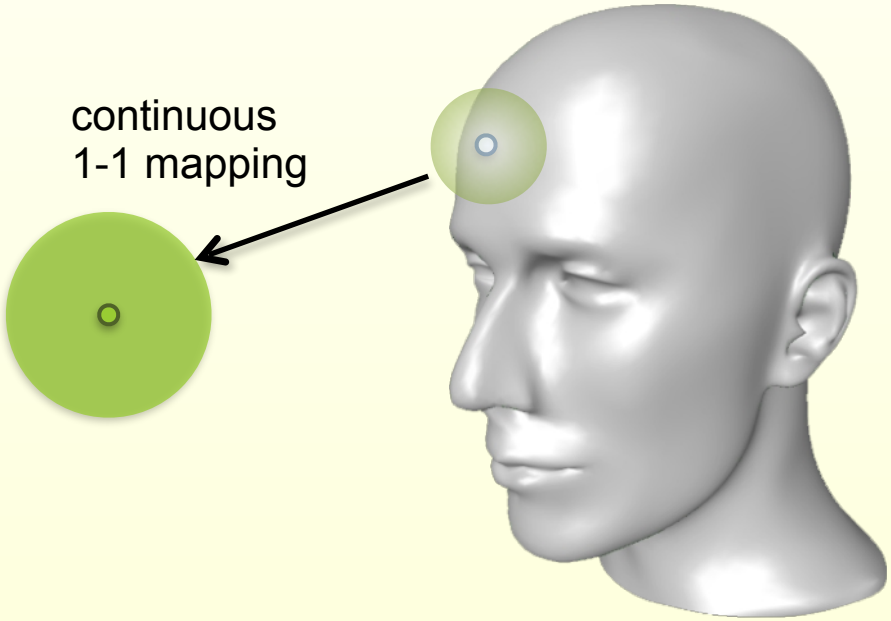
Differential Geometry Basics

- Geometry of manifolds
- Properties that can be discovered by local observation: point + neighborhood



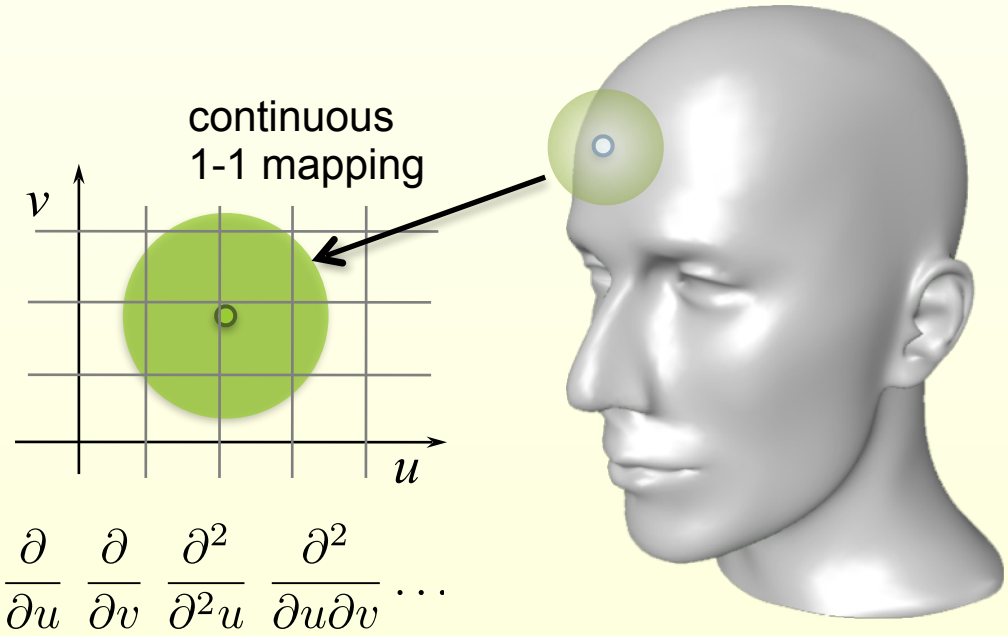
Differential Geometry Basics

- Geometry of manifolds
- Properties that can be discovered by local observation: point + neighborhood



Differential Geometry Basics

- Geometry of manifolds
- Properties that can be discovered by local observation: point + neighborhood

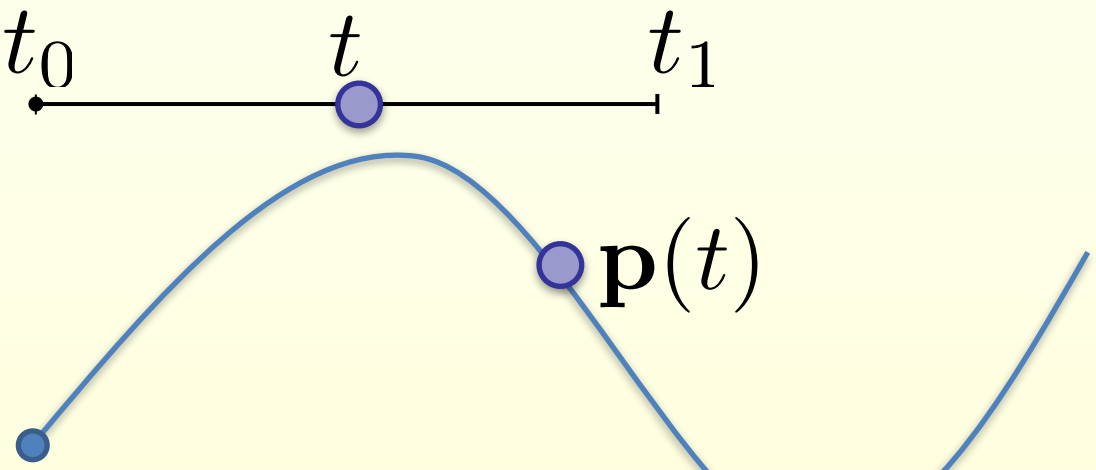


If a sufficiently smooth mapping can be constructed, we can look at its first and second derivatives

Tangents, normals, curvatures, curve angles, distances

Parametric Curves

- 2D: $\mathbf{p}(t) = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix}, \; t \in [t_0, t_1]$
- $\mathbf{p}(t)$ must be continuous



$$len\,(\mathbf{p}(t_0), \mathbf{p}(t)) = \int_{t_0}^{t} \|\mathbf{p}'(\tilde{t})\| d\tilde{t}$$

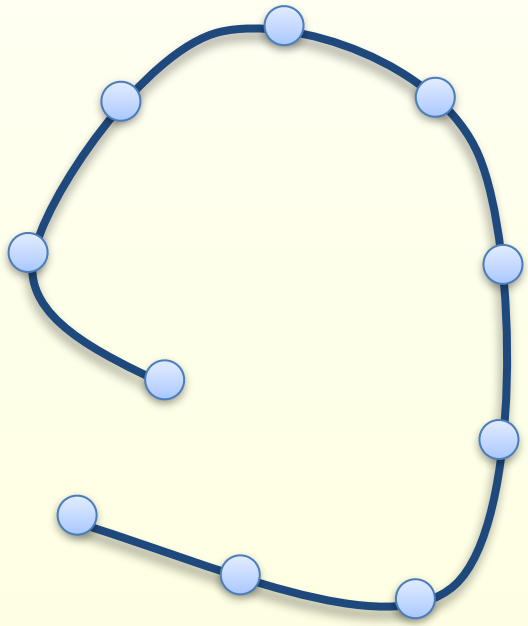
Arc Length Parameterization

- Equal pace of the parameter along the curve

- $len\ (\mathbf{p}(s_1), \mathbf{p}(s_2)) = |s_1 - s_2|$

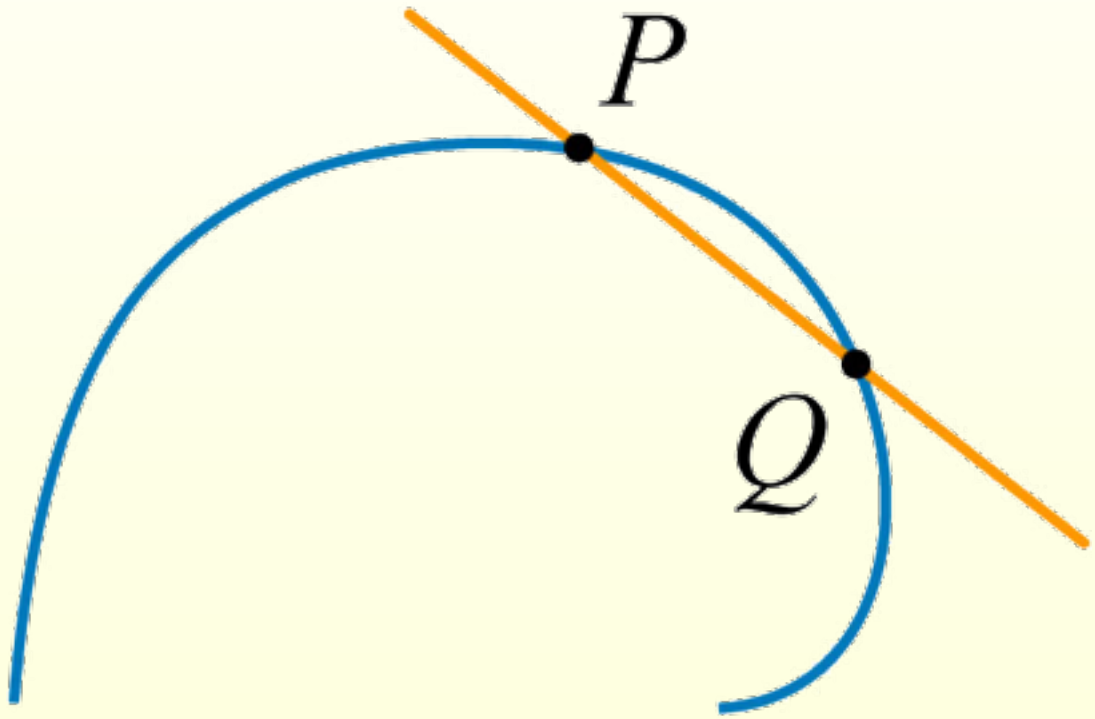
- Now parameter goes from 0 to L

$$\|\mathbf{p}'(s)\| = 1$$



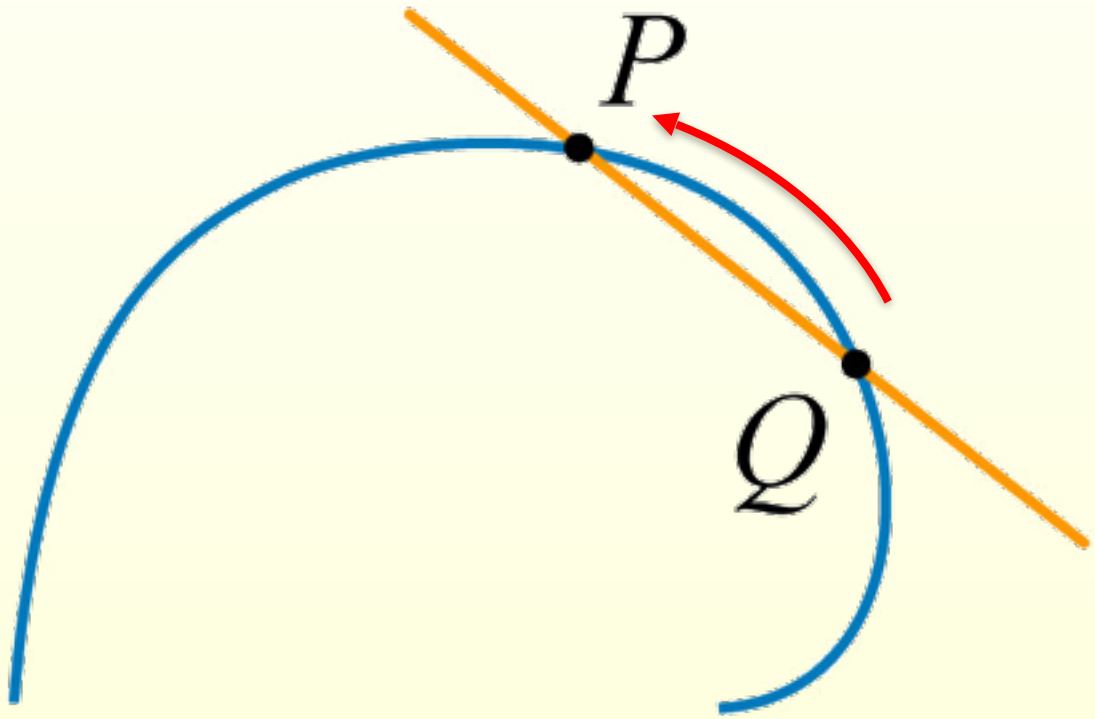
Secant

- A line through two points on the curve.



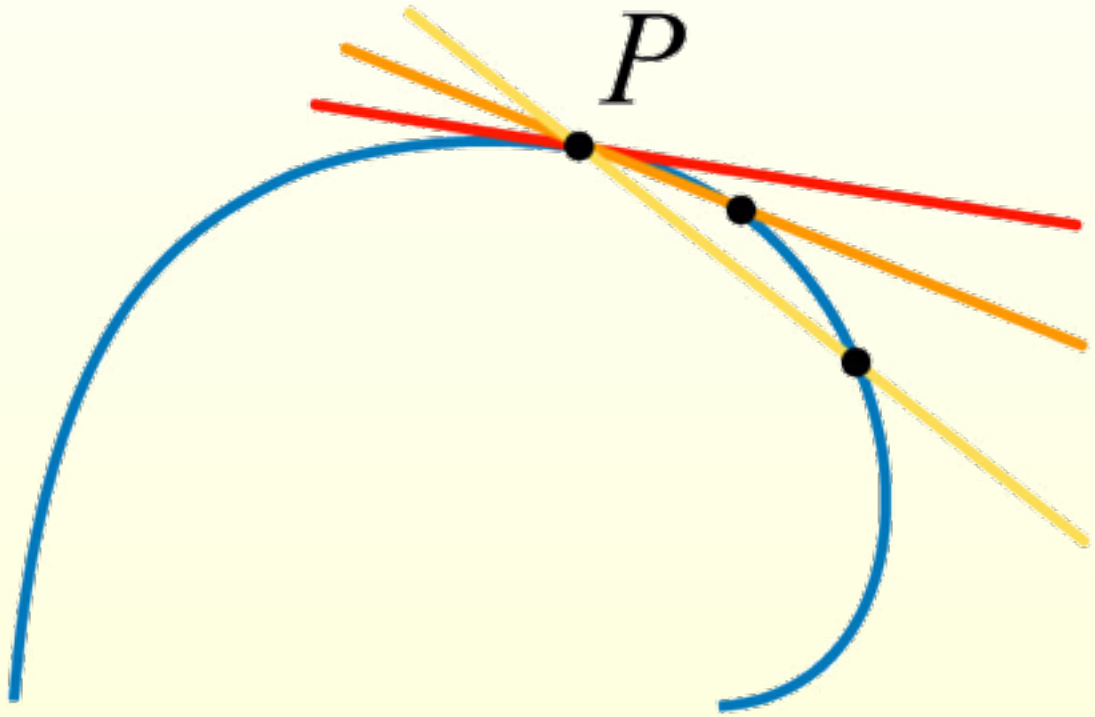
Secant

- A line through two points on the curve.



Tangent

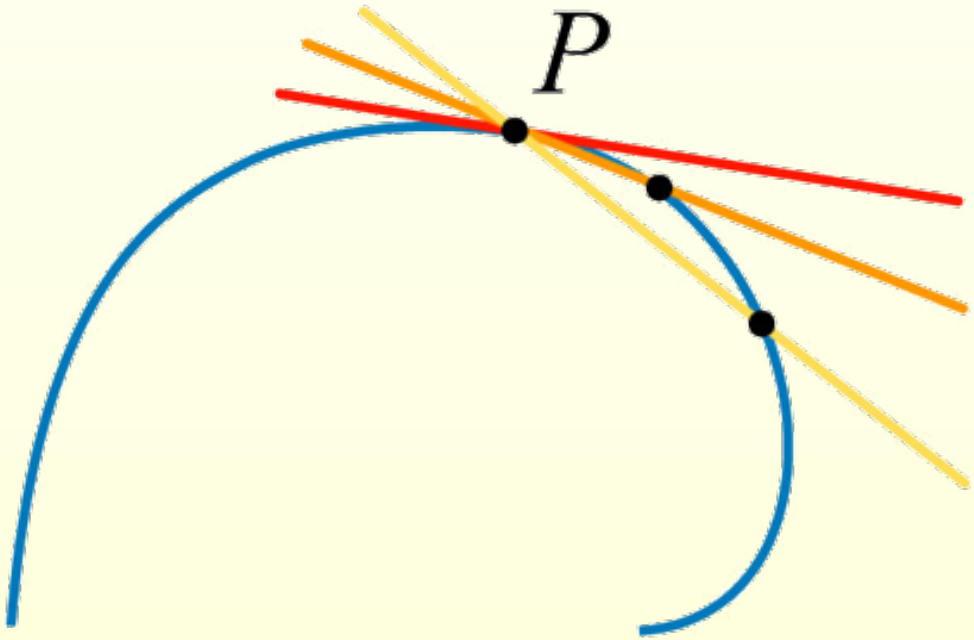
- The limiting secant as the two points come together.



Secant and Tangent – Parametric Form

- Secant: $\mathbf{p}(t) - \mathbf{p}(s)$
- Tangent: $\mathbf{p}'(t) = (x'(t), y'(t), \ldots)^{\mathrm{T}}$
- If t is arc-length:

  $\|\mathbf{p}'(t)\| = 1$

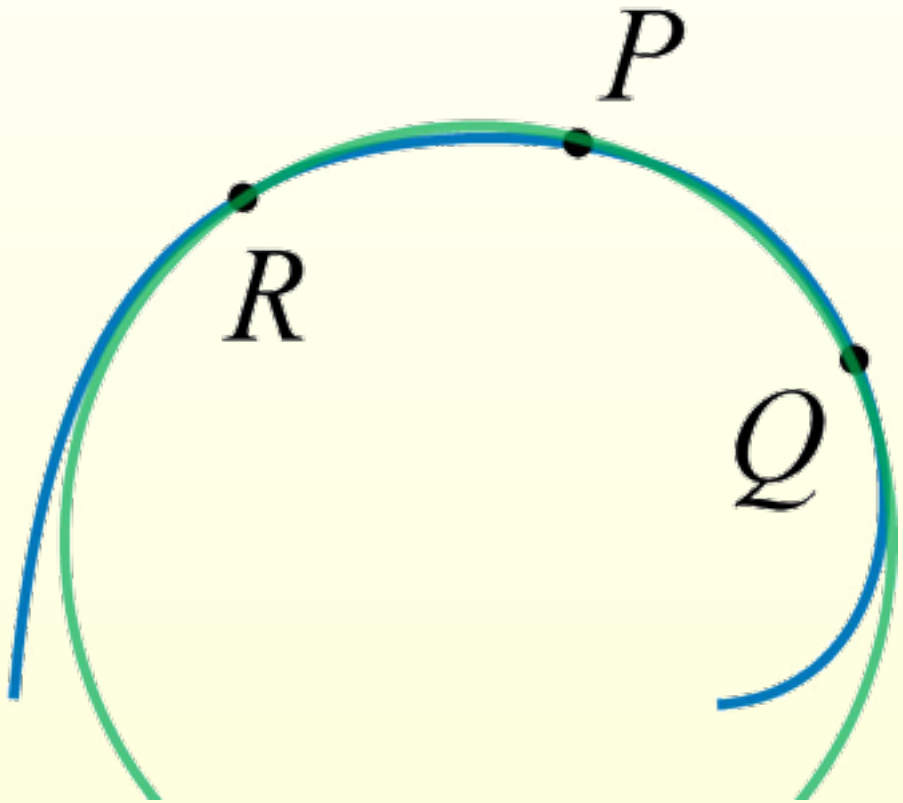


Recall

$$len(\mathbf{p}(t_0), \mathbf{p}(t)) = \int_0^t \|\mathbf{p}'(t)\| dt$$

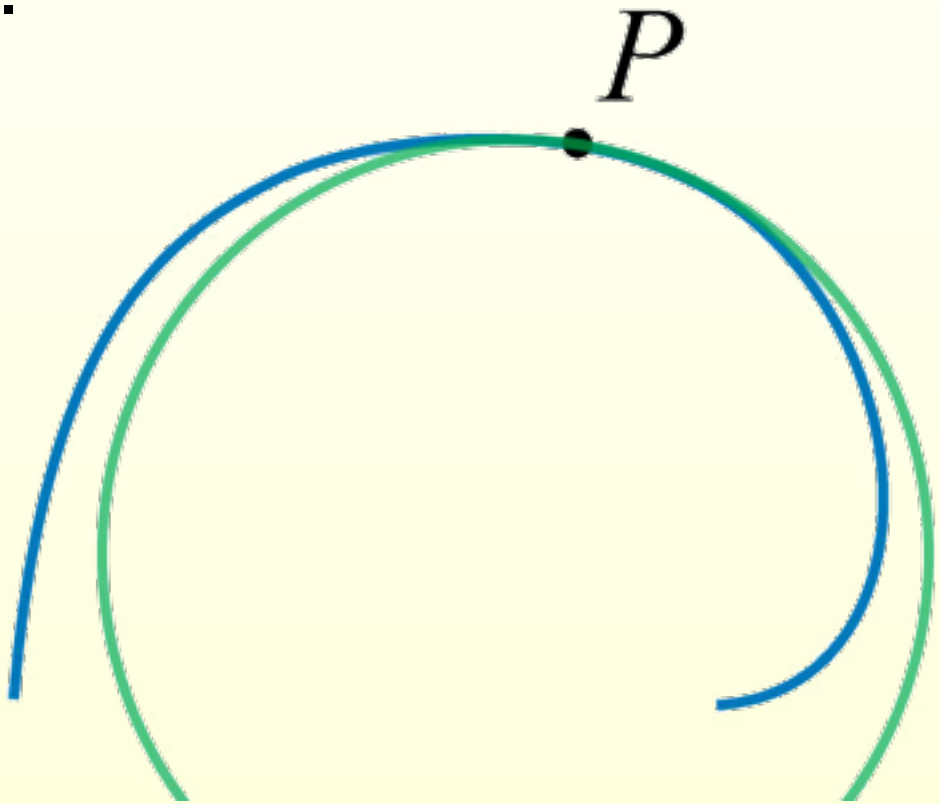
Circle of Curvature

- Consider the circle passing through three points on the curve…

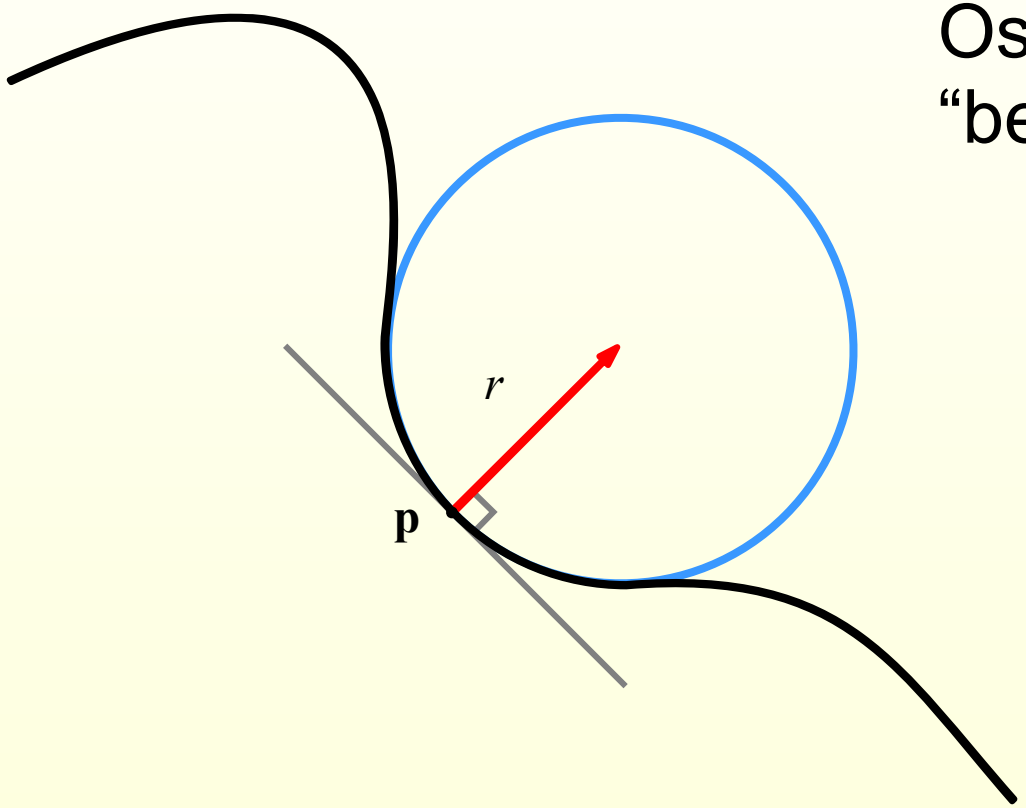


Circle of Curvature

- …the limiting circle as three points come together.



Tangent, normal, radius of curvature

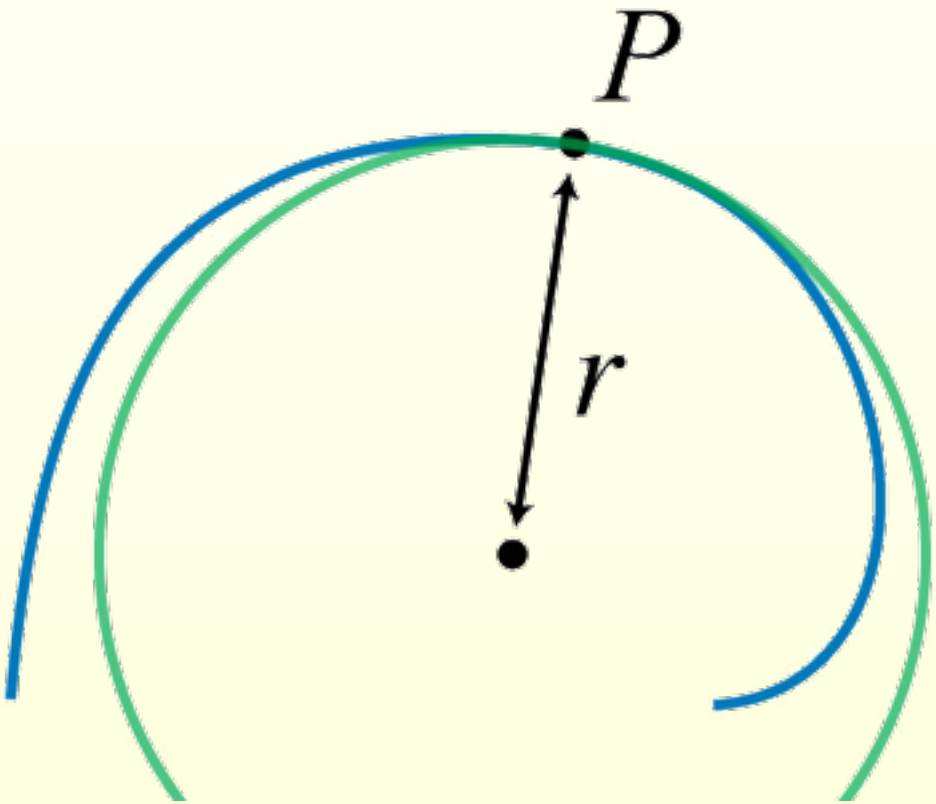


Osculating circle
“best fitting circle”

Radius of Curvature, $r = 1/\kappa$

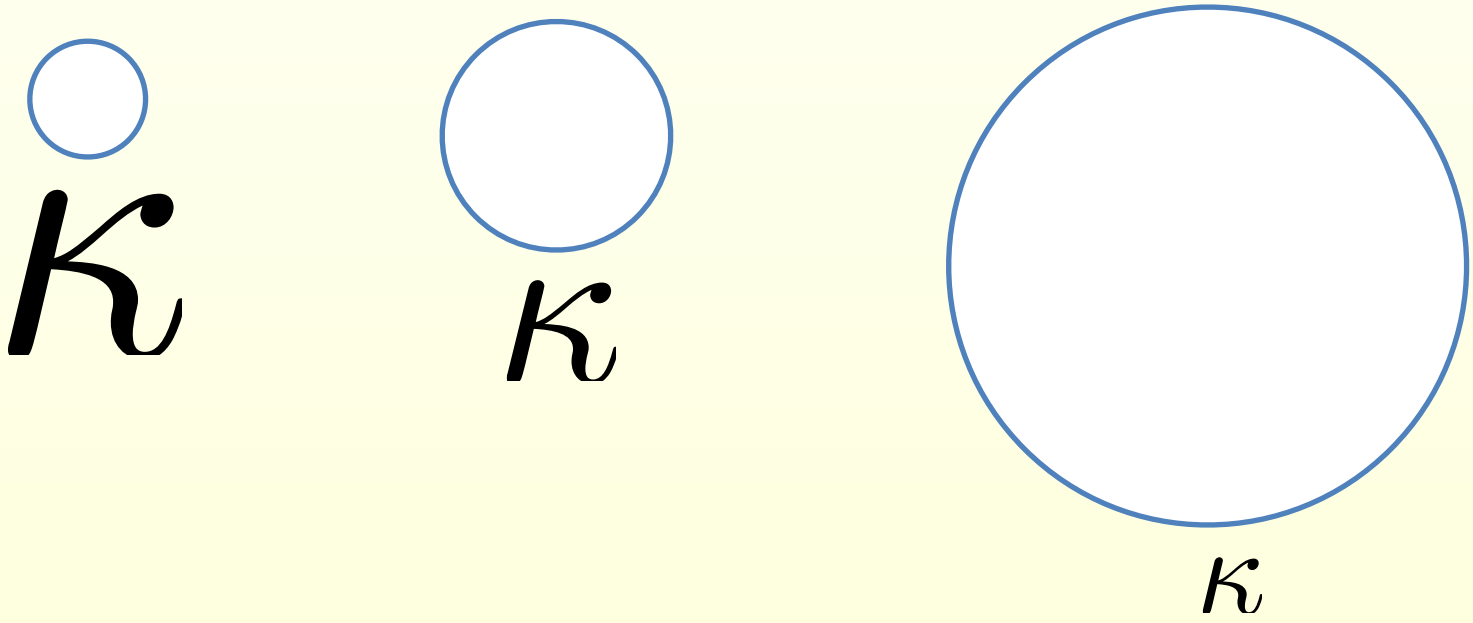
Curvature

$$\kappa = \frac{1}{r}$$



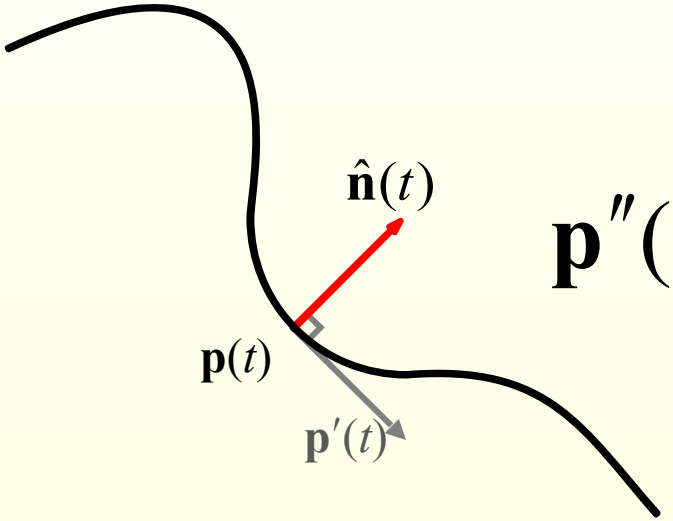
Curvature is scale dependent

$$\kappa = \frac{1}{r}$$

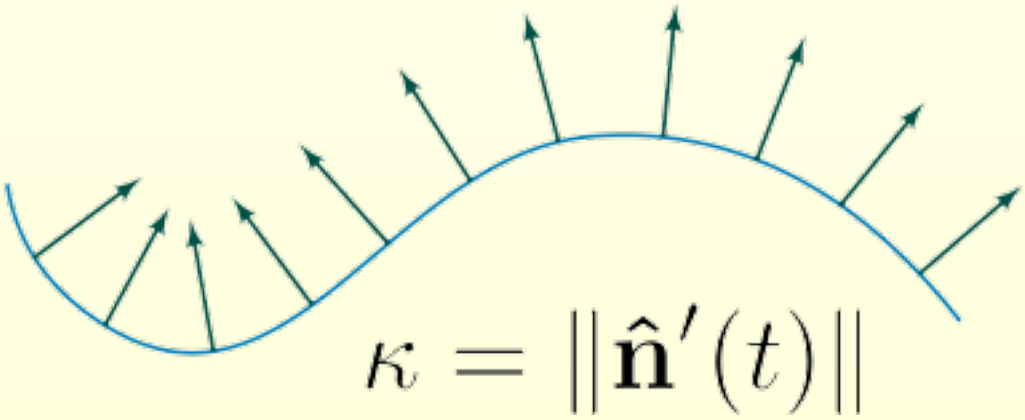


Curvature and Normal

- Assuming t is arc-length parameter:



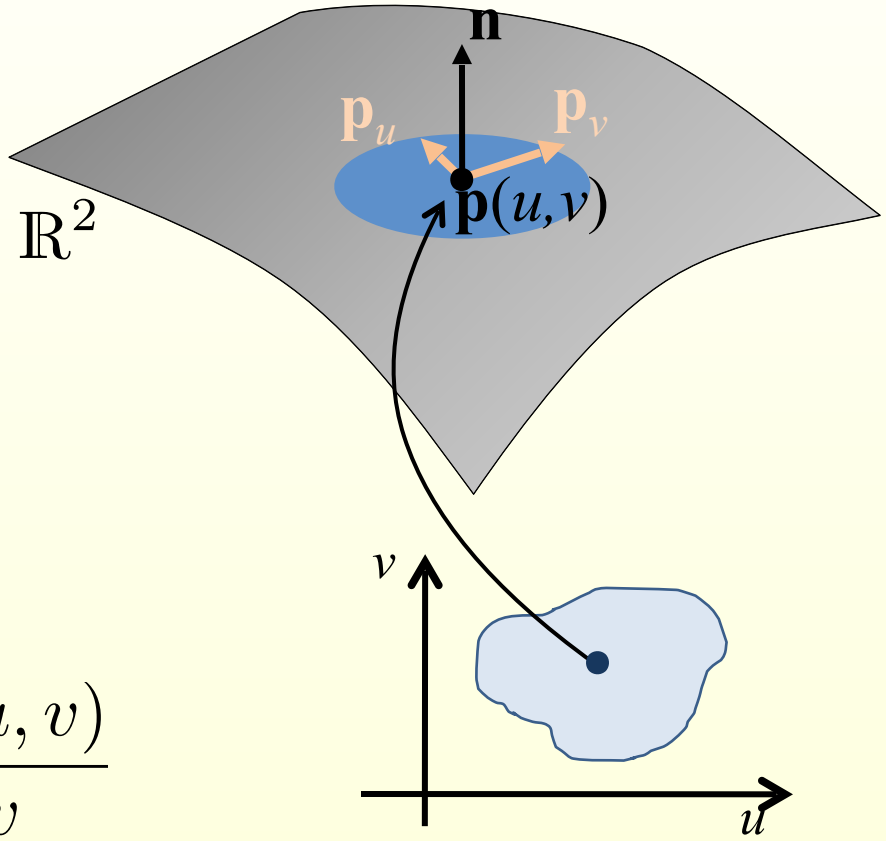
$$\mathbf{p}''(t) = \kappa\hat{\mathbf{n}}(t)$$ normal to the curve



Surfaces, Parametric Form

- Continuous surface

$$\mathbf{p}(u,v) = \begin{pmatrix} x(u,v) \\ y(u,v) \\ z(u,v) \end{pmatrix}, \quad (u,v) \in \mathbb{R}^2$$



- Tangent plane at point $\mathbf{p}(u,v)$ is spanned by

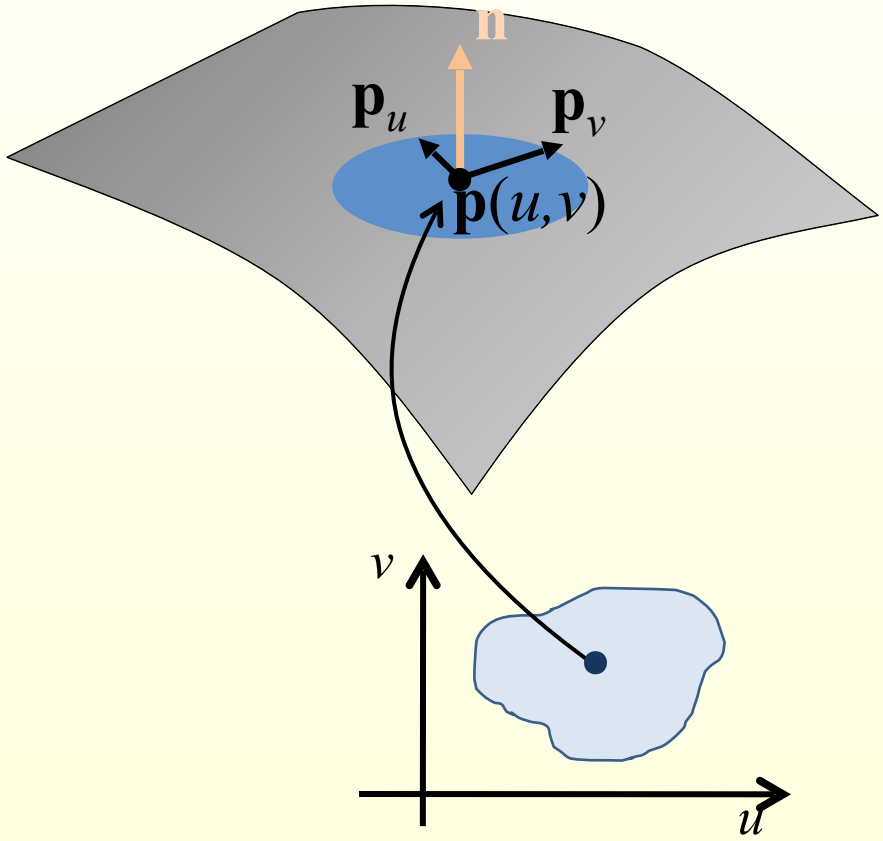
$$\mathbf{p}_u = \frac{\partial \mathbf{p}(u,v)}{\partial u}, \quad \mathbf{p}_v = \frac{\partial \mathbf{p}(u,v)}{\partial v}$$

These vectors don't have to be orthogonal

Surface Normals

- Surface normal:

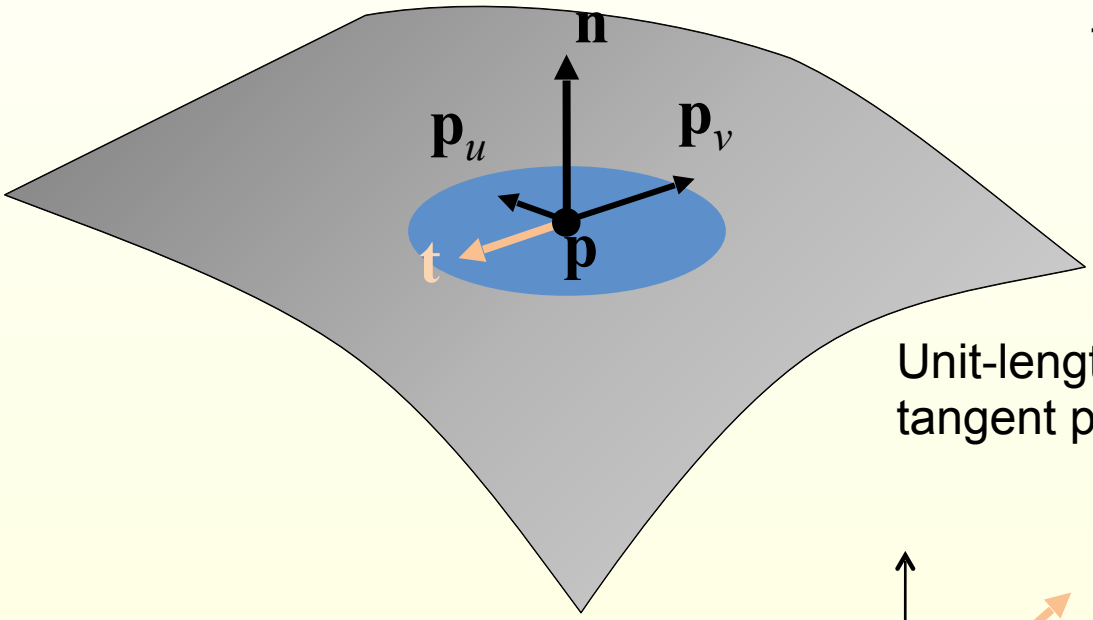
$$\mathbf{n}(u,v) = \frac{\mathbf{p}_u \times \mathbf{p}_v}{\|\mathbf{p}_u \times \mathbf{p}_v\|}$$



- Assuming *regular* parameterization, i.e.,

$$\mathbf{p}_u \times \mathbf{p}_v \neq 0$$

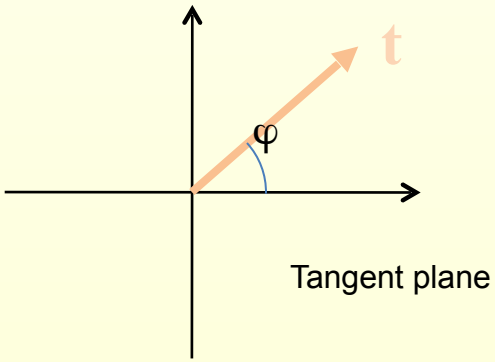
Normal Curvature



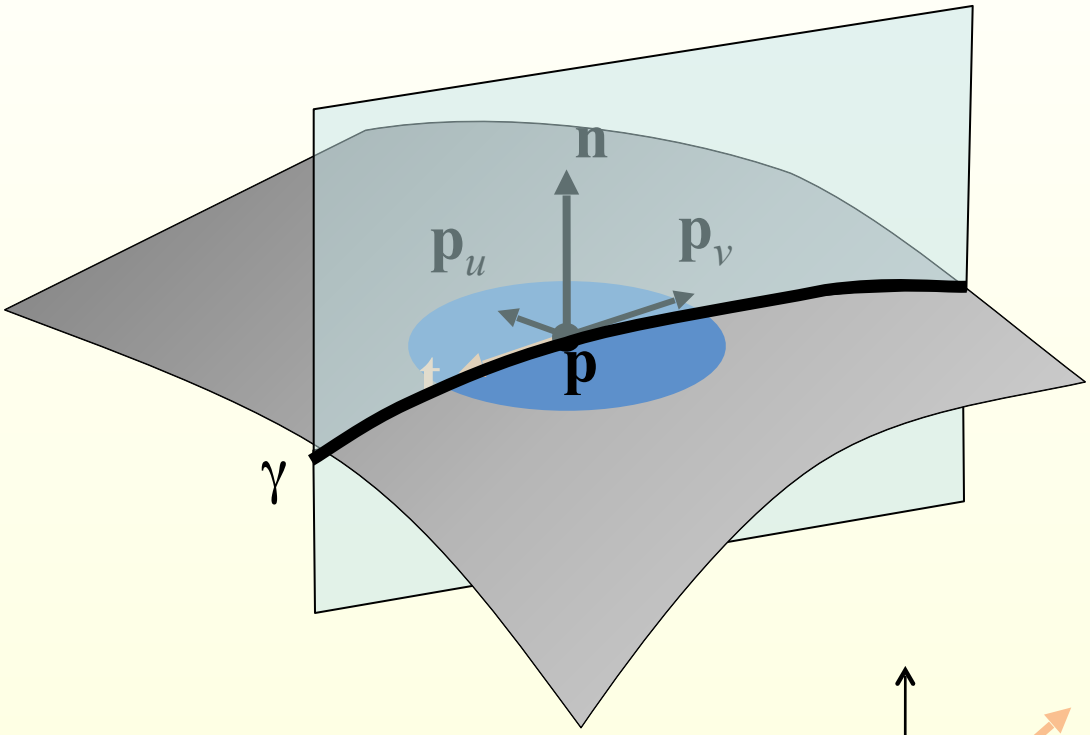
$$\mathbf{n}(u,v) = \frac{\mathbf{p}_u \times \mathbf{p}_v}{\|\mathbf{p}_u \times \mathbf{p}_v\|}$$

Unit-length direction **t** in the tangent plane (if $\mathbf{p}_u$ and $\mathbf{p}_v$ are orthogonal):

$$\mathbf{t} = \cos\varphi \frac{\mathbf{p}_u}{\|\mathbf{p}_u\|} + \sin\varphi \frac{\mathbf{p}_v}{\|\mathbf{p}_v\|}$$



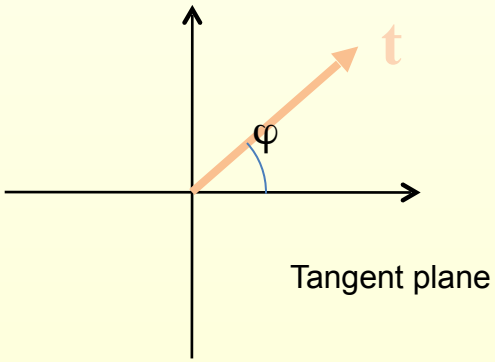
Normal Curvature



The curve γ is the intersection of the surface with the plane through $\mathbf{n}$ and $\mathbf{t}$ - a normal section.

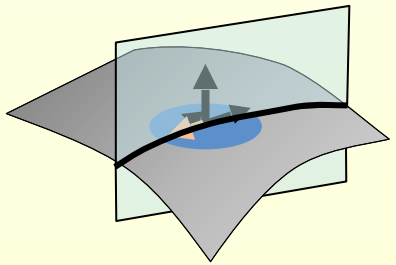
Normal curvature:

$$\kappa_n(\varphi) = \kappa(\gamma(\mathbf{p}))$$



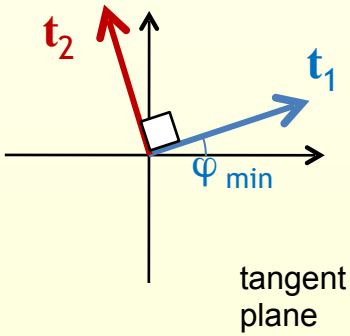
Surface Curvatures

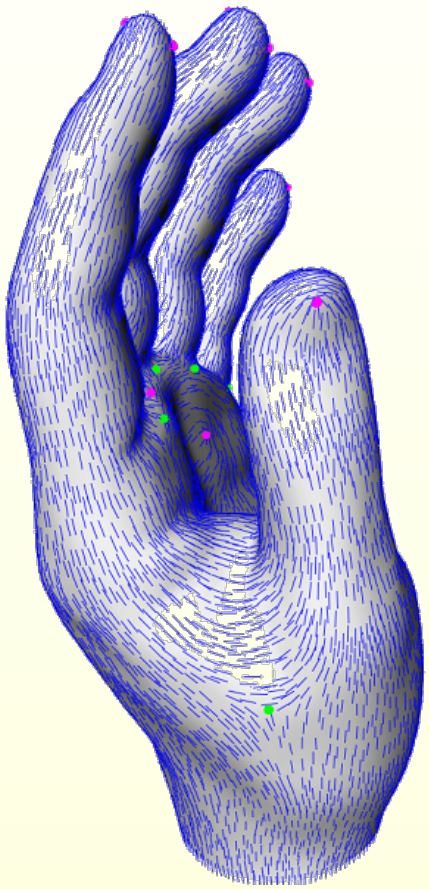
- **Principal curvatures**
  - Minimal curvature $\kappa_1 = \kappa_{\min} = \min_{\varphi} \kappa_n(\varphi)$
  - Maximal curvature $\kappa_2 = \kappa_{\max} = \max_{\varphi} \kappa_n(\varphi)$
- **Mean curvature** $H = \frac{\kappa_1 + \kappa_2}{2} = \frac{1}{2\pi} \int_0^{2\pi} \kappa_n(\varphi) d\varphi$
- **Gaussian curvature** $K = \kappa_1 \cdot \kappa_2$



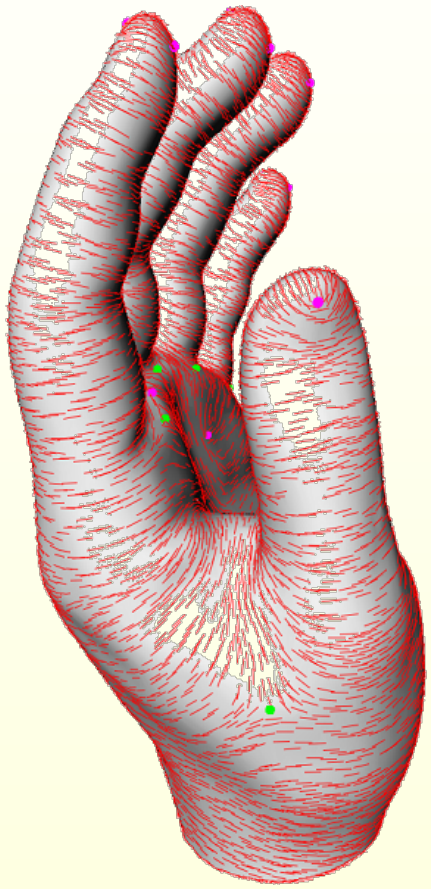
Principal Directions

- Principal directions: tangent vectors corresponding to φ_{max} and φ_{min}



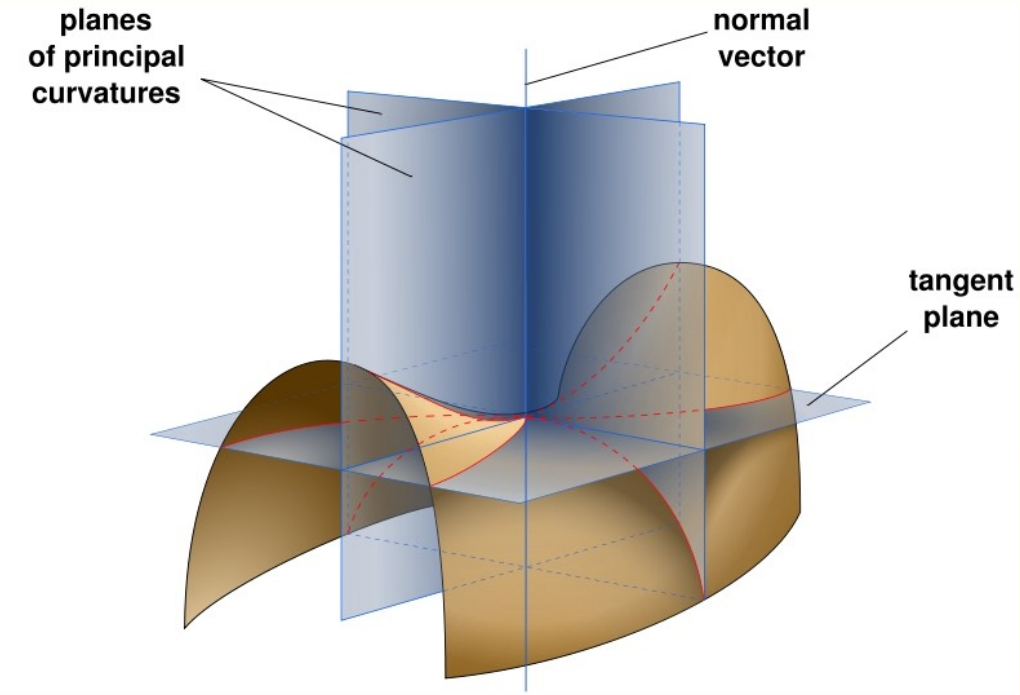


min curvature



max curvature

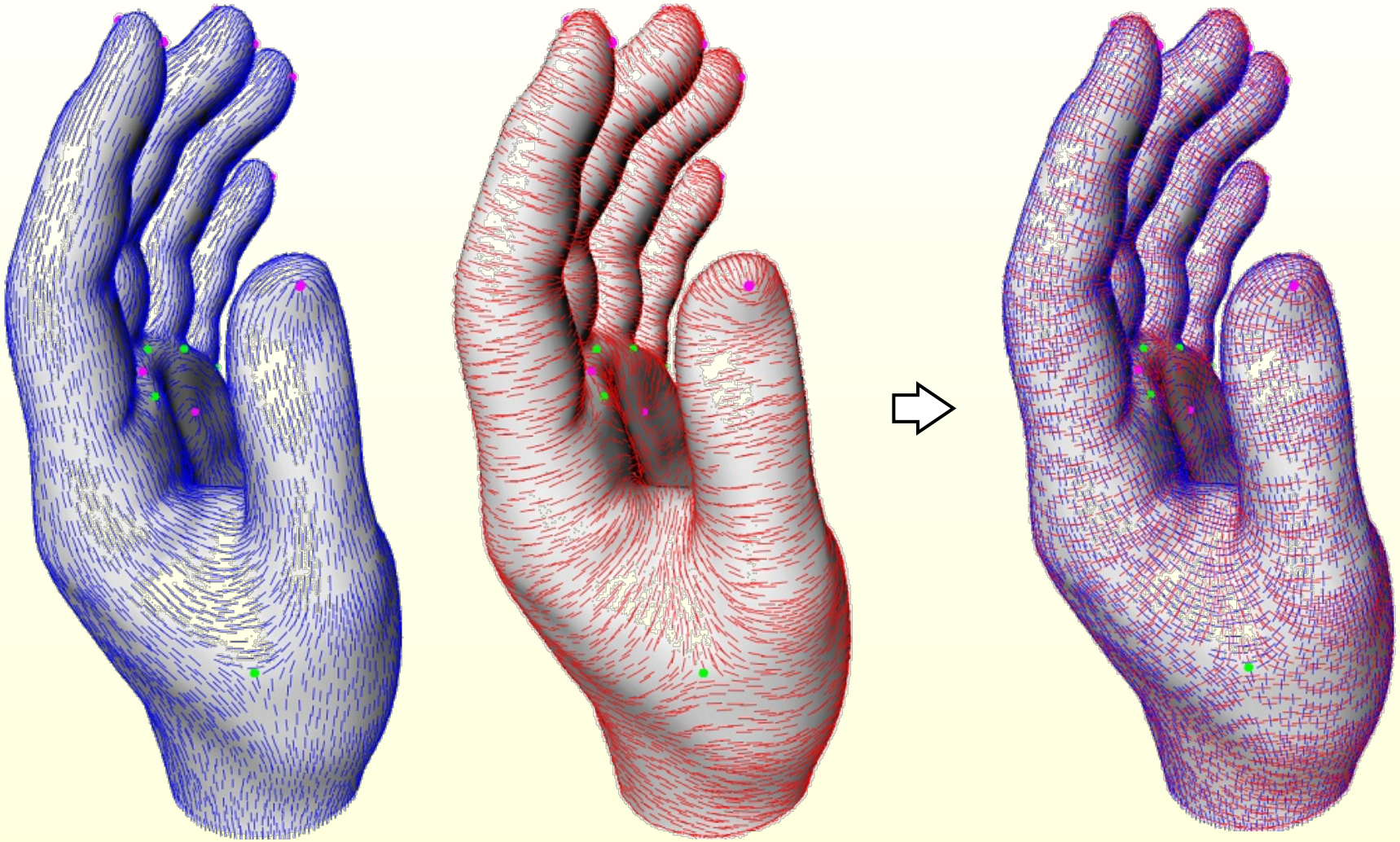
Principal Directions



Euler's Theorem: Planes of principal curvature are **orthogonal** and independent of parameterization.

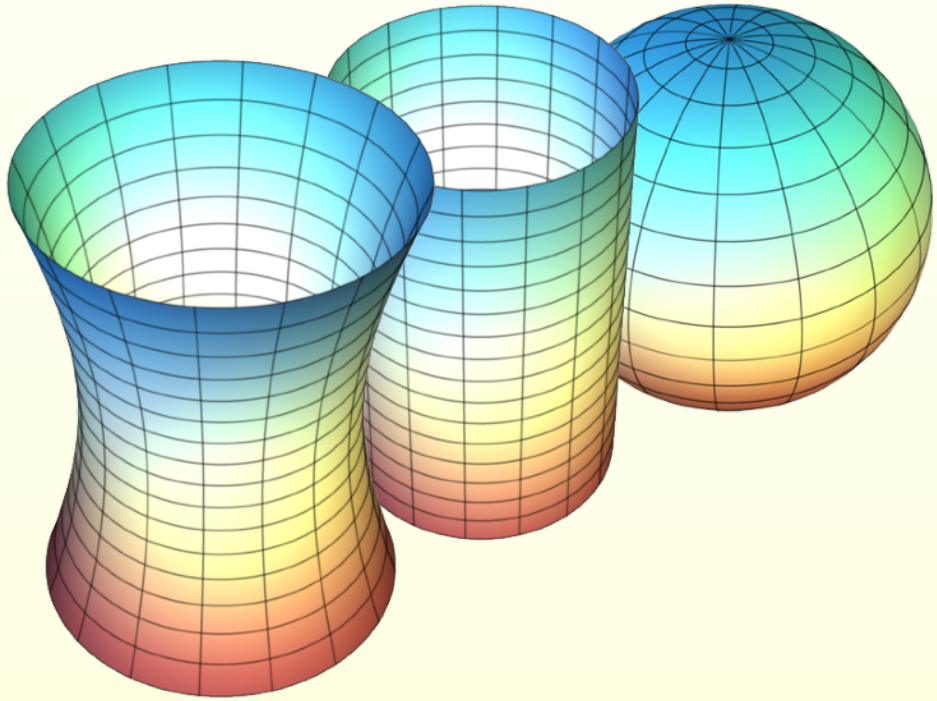
$$\kappa_n(\varphi) = \kappa_1 \cos^2 \varphi + \kappa_2 \sin^2 \varphi, \quad \varphi = \text{angle with } \mathbf{t}_1$$

Principal Directions



Classification

- A point $\mathbf{p}$ on the surface is called
  - Elliptic, if $K > 0$
  - Parabolic, if $K = 0$
  - Hyperbolic, if $K < 0$

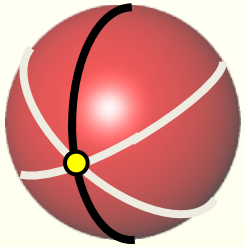


- Developable surface
  iff $K = 0$
  - can be mapped to the plane without distortion

Local Surface Shape By Curvatures

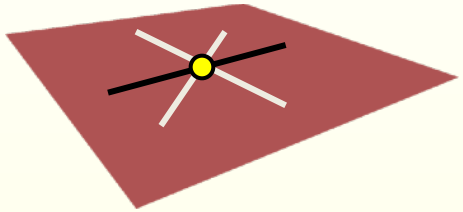
Isotropic:
all directions are principal directions

$K > 0$, $\kappa_1 = \kappa_2$



spherical (umbilical)

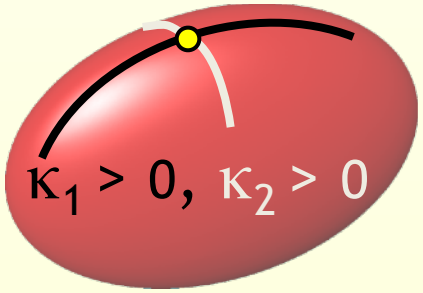
$K = 0$



planar

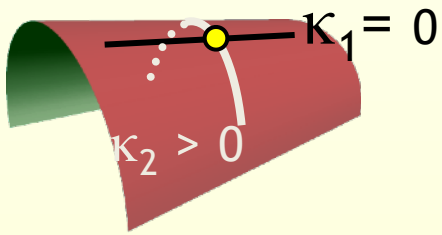
Anisotropic:
2 distinct principal directions

$K > 0$



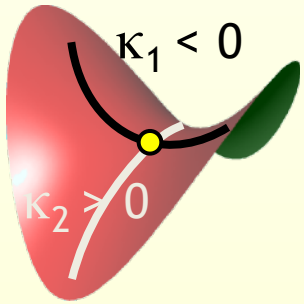
elliptic

$K = 0$



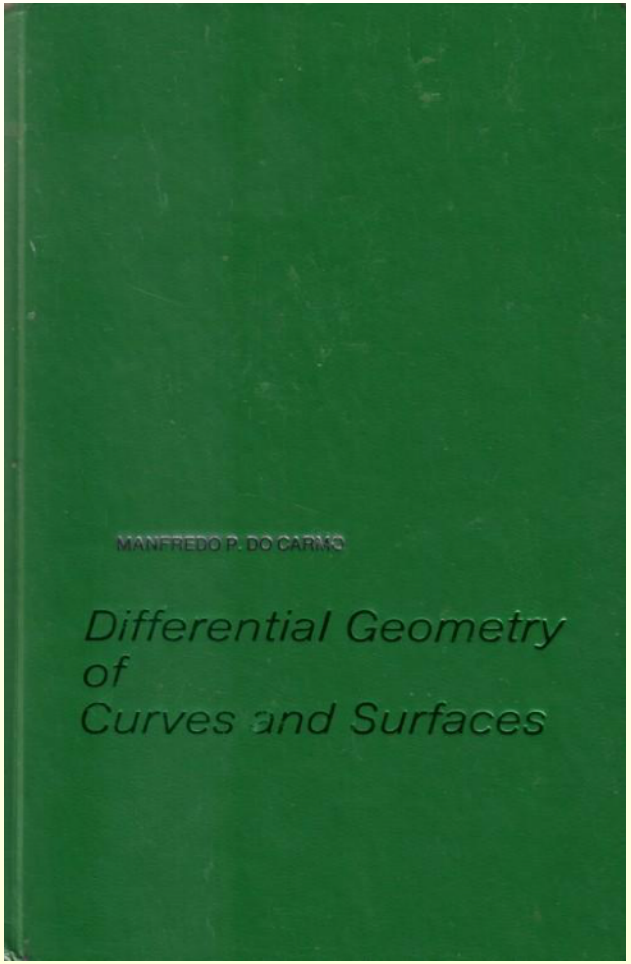
parabolic

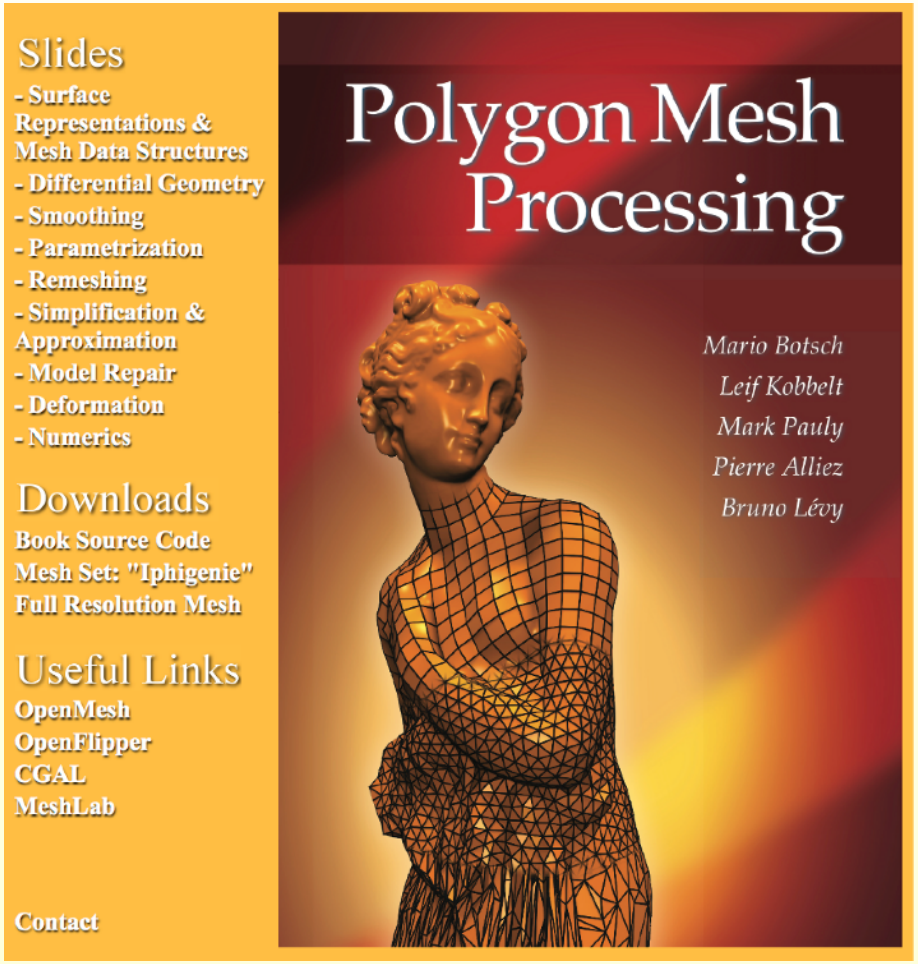
$K < 0$



hyperbolic

Additional reading





Software

- Libigl http://libigl.github.io/libigl/tutorial/tutorial.html
  - MATLAB-style (flat) C++ library, based on indexed face set structure
- OpenMesh www.openmesh.org
  - Mesh processing, based on half-edge data structure
- CGAL www.cgal.org
  - Computational geometry
- MeshLab http://www.meshlab.net/
  - Viewing and processing meshes

Software

- Alec Jacobson's GP toolbox
  - https://github.com/alecjacobson/gptoolbox
  - MATLAB, various mesh and matrix routines
- Gabriel Peyre's Fast Marching Toolbox
  - https://www.mathworks.com/matlabcentral/fileexchange/6110-toolbox-fast-marching
  - Geodesic (on-surface) distances
- OpenFlipper https://www.openflipper.org/
  - Various GP algorithms + Viewer

Spectral mesh processing

THE LAPLACE-BELTRAMI OPERATOR

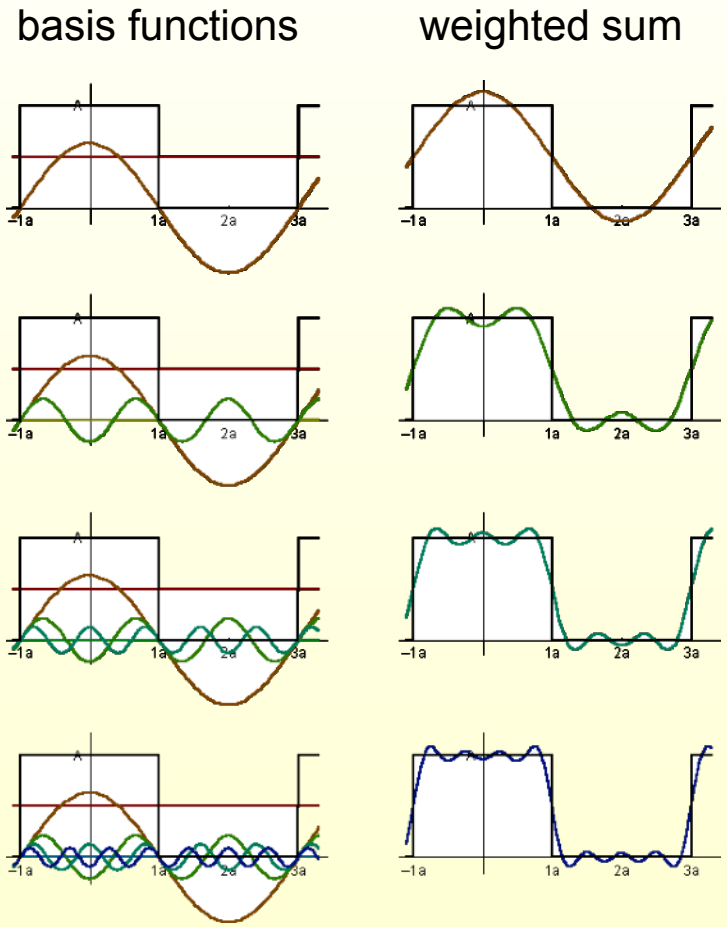
Motivation: Information Representation

- Various surface properties can be represented as functions “living” on the surface
  - Curvatures (k_1, k_2, K, H)
  - Part indicator functions
  - Vector-valued functions, like surface coordinates, normals and texture
  - …
- We need a tool to work with functions on surfaces

Reminder: Fourier analysis

- Represent a function as a weighted sum of sines and cosines (basis functions)

Joseph Fourier 1768 - 1830



$$f(x) = a_0 + a_1\cos(x) + a_2\cos(3x) + a_3\cos(5x) + a_4\cos(7x) + \ldots$$

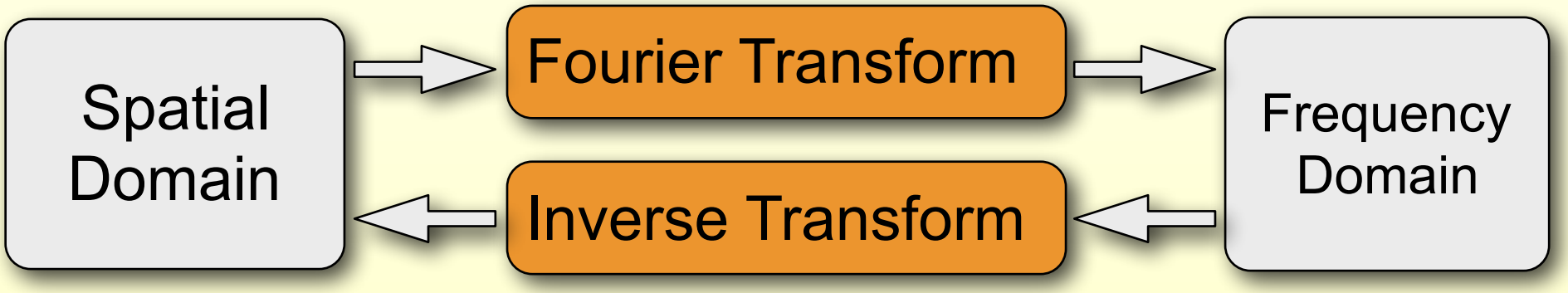
Coefficients : co-integrate function with basis

More generally - Fourier analysis

- Inner product for L_2 function space $\langle f, g \rangle := \int_{-\infty}^{\infty} f(x)\,\overline{g(x)}\,\mathrm{d}x$
- Orthonormal basis : complex "waves"

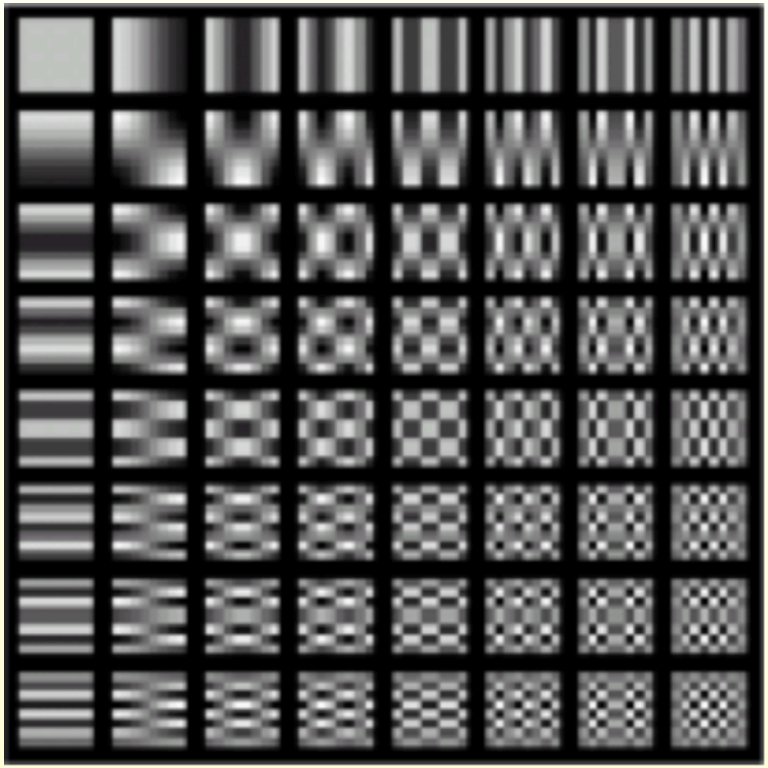
$$e_u(x) \;:=\; \mathrm{e}^{\mathrm{i}2\pi ux} \;=\; \cos(2\pi ux) - \mathrm{i}\sin(2\pi ux)$$

$$F(\omega) = \int_{-\infty}^{\infty} f(x)\mathrm{e}^{-2\pi i\omega x}\,dx$$



$$f(x) = \int_{-\infty}^{\infty} F(\omega)\mathrm{e}^{2\pi i\omega x}\,d\omega$$

We can also Fourier on rectangular 2D domains



Fourier (DCT) basis functions for 8x8 grayscale images

$$\cos(2\pi\omega_h)\cos(2\pi\omega_v)$$

Extend Fourier to meshes?

- So far, our functions have been defined on parameterized patches $f(u,v)$
  - Generalize to meshes!

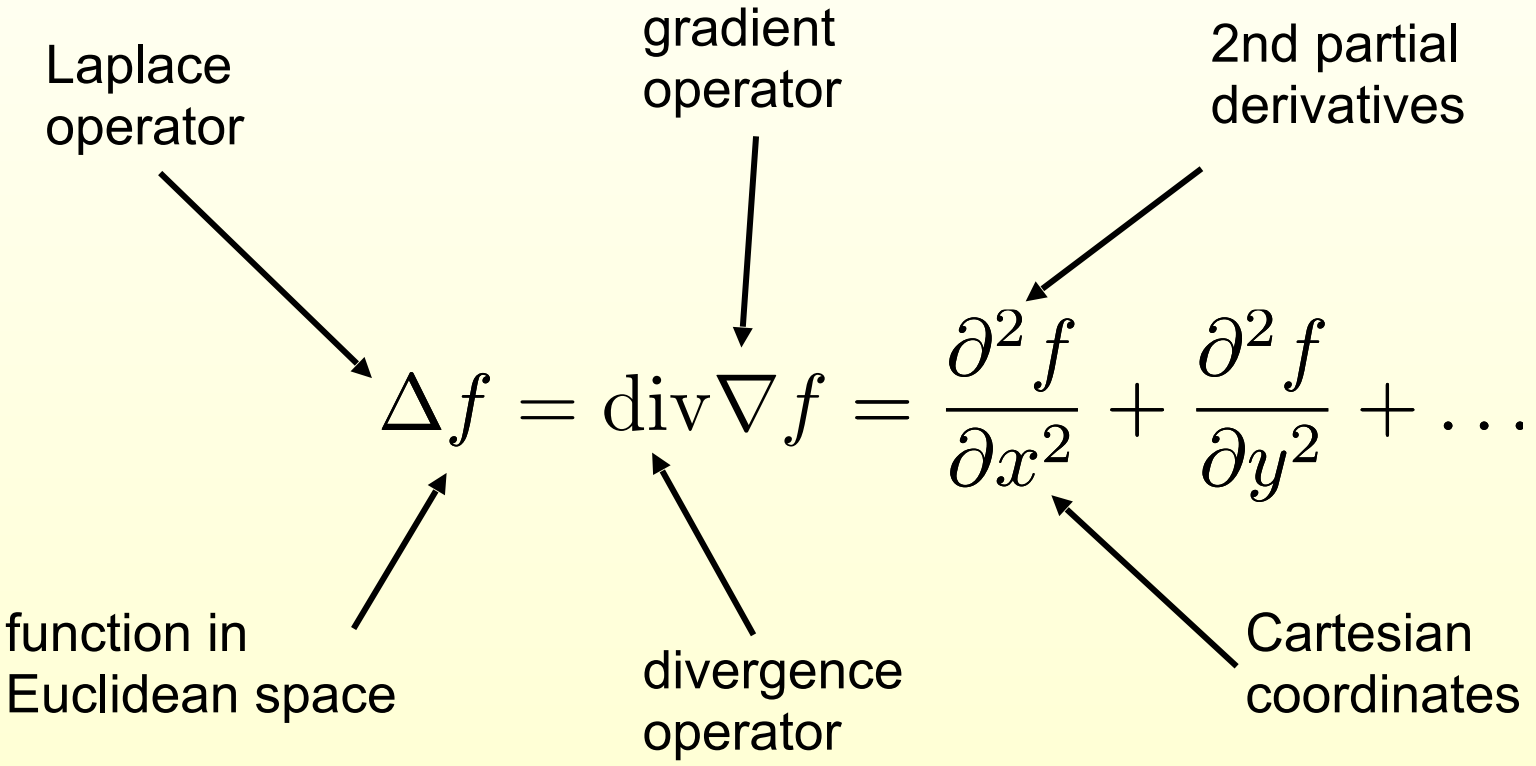
- Fourier basis functions are eigenfunctions of the (standard) Laplace operator Δ: $L^2 \rightarrow L^2$

$$\Delta\left(e^{2\pi i\omega x}\right) = \frac{\partial^2}{\partial x^2} e^{2\pi i\omega x} = -(2\pi\omega)^2 e^{2\pi i\omega x}$$

- We need
  - A version of this operator for 2D manifolds, and
  - A discrete (mesh-based) version!

Continuous Laplace Operator

$$f : \mathbb{R}^3 \to \mathbb{R} \qquad \Delta f : \mathbb{R}^3 \to \mathbb{R}$$

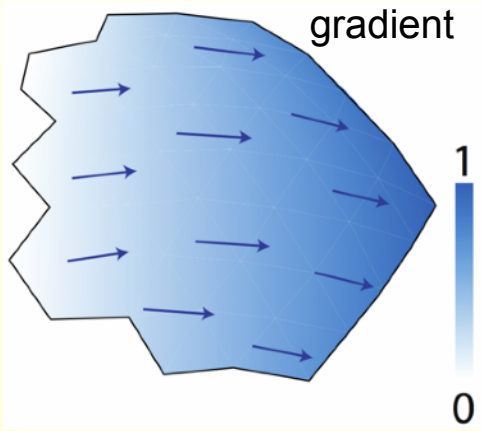


$$\mathrm{grad} f = \nabla f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z} \right) \qquad \mathrm{div} \mathbf{F} = \nabla \cdot \mathbf{F} = \frac{\partial F_x}{\partial x} + \frac{\partial F_y}{\partial y} + \frac{\partial F_z}{\partial z}$$

Continuous Laplace-Beltrami Operator

- Extension of Laplace operator to functions on manifolds

$$f : \mathcal{M} \to \mathbb{R} \qquad \Delta f : \mathcal{M} \to \mathbb{R}$$



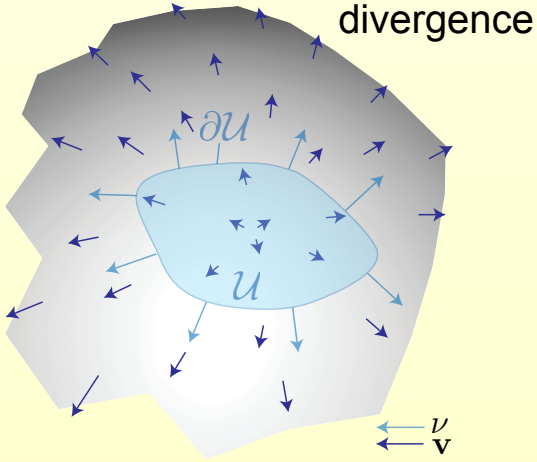
Laplace-Beltrami

gradient operator

$$\Delta_{\mathcal{M}} f = \operatorname{div}_{\mathcal{M}} \nabla_{\mathcal{M}} f$$

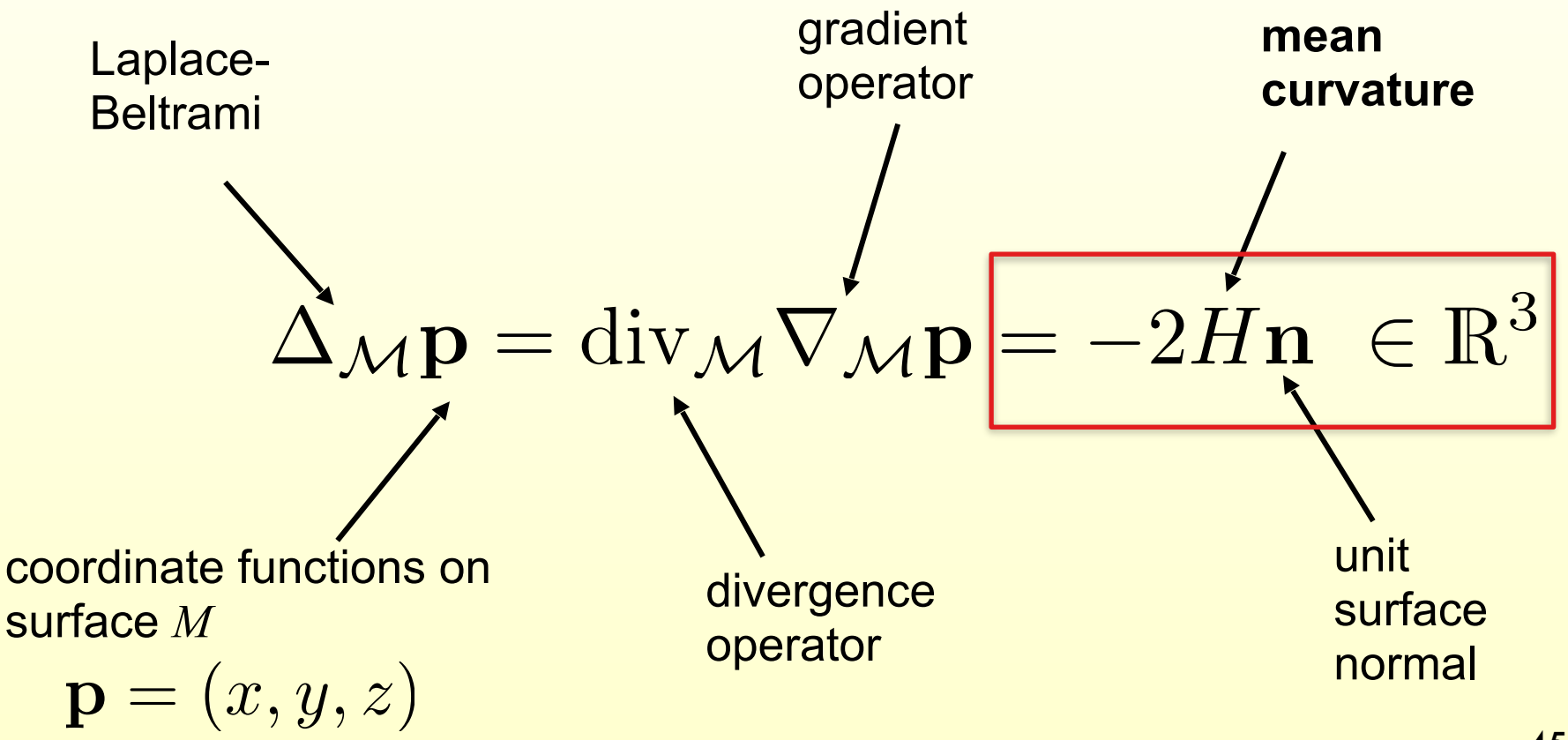
function on surface M

divergence operator



Laplace-Beltrami and Curvature

- Apply operator to coordinate functions

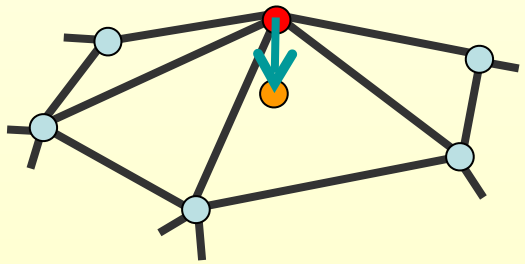


Differential Properties on Meshes

- So for Laplacian, we need differential quantities (gradient, divergence…)
- Assumption: meshes are piecewise linear approximations of smooth surfaces
- Can try fitting a smooth surface locally (say, a polynomial) and find differential quantities analytically
- But: it is often too slow for interactive setting and error prone

Discrete Differential Operators

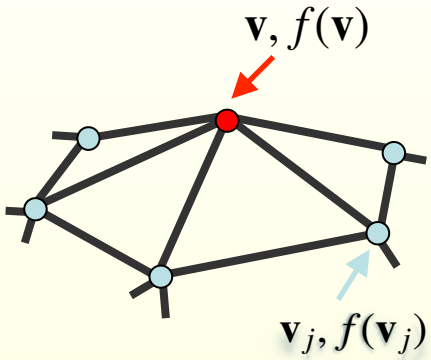
- Approach: approximate differential properties at point $\mathbf{v}$ as spatial average over local mesh neighborhood $N(\mathbf{v})$ where typically
  - $\mathbf{v}$ = mesh vertex
  - $N_k(\mathbf{v})$ = k-ring neighborhood



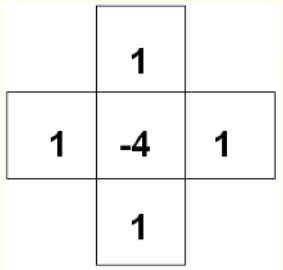
Discrete Laplace-Beltrami

- Uniform discretization: $L(f)$ or Δf

$$\begin{aligned}\Delta f(\mathbf{v}) &= \sum_{v_j \in N(v)} (f(\mathbf{v}_j) - f(\mathbf{v})) \\ &= \sum_{v_j \in N(v)} f(\mathbf{v}_j) - k f(\mathbf{v}), \; k = |N(v)|\end{aligned}$$



- Similar to 5 point stencil for images!



- Depends only on connectivity: simple and efficient

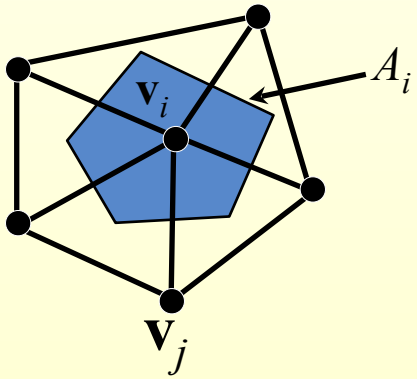
- Bad approximation for irregular triangulations

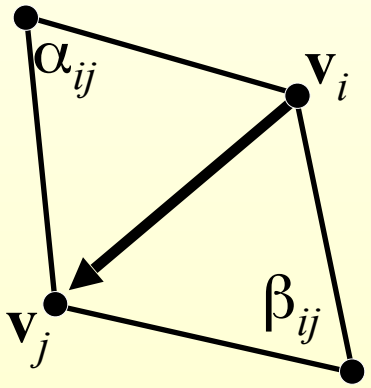
Discrete LB - cotangent formula

- Better: cotangent formula

$$\Delta_S f(v_i) := \frac{1}{2A_i} \sum_{v_j \in \mathcal{N}_1(v_i)} (\cot\alpha_{ij} + \cot\beta_{ij})\,(f(v_j) - f(v_i))$$

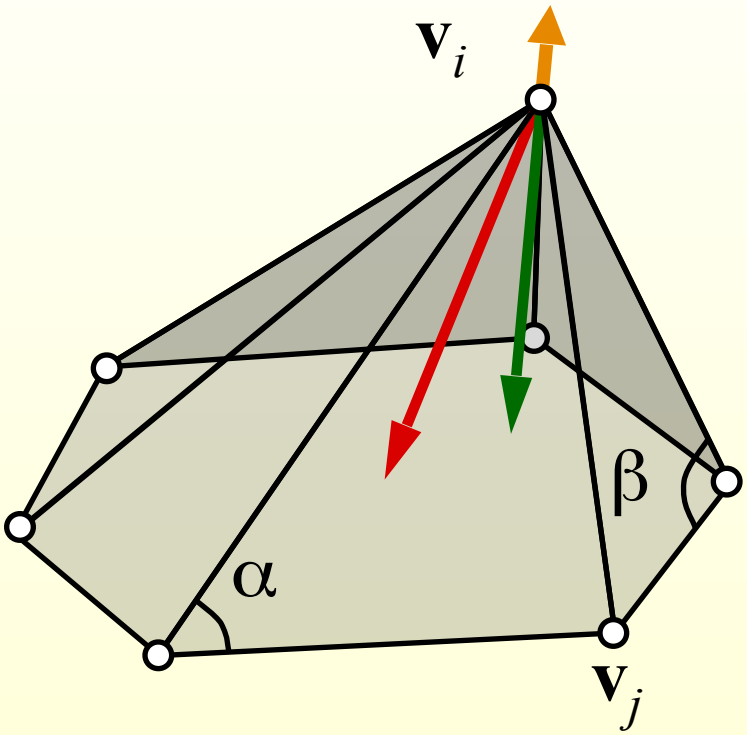
A_i : vertex area (Voronoi, barycentric..)





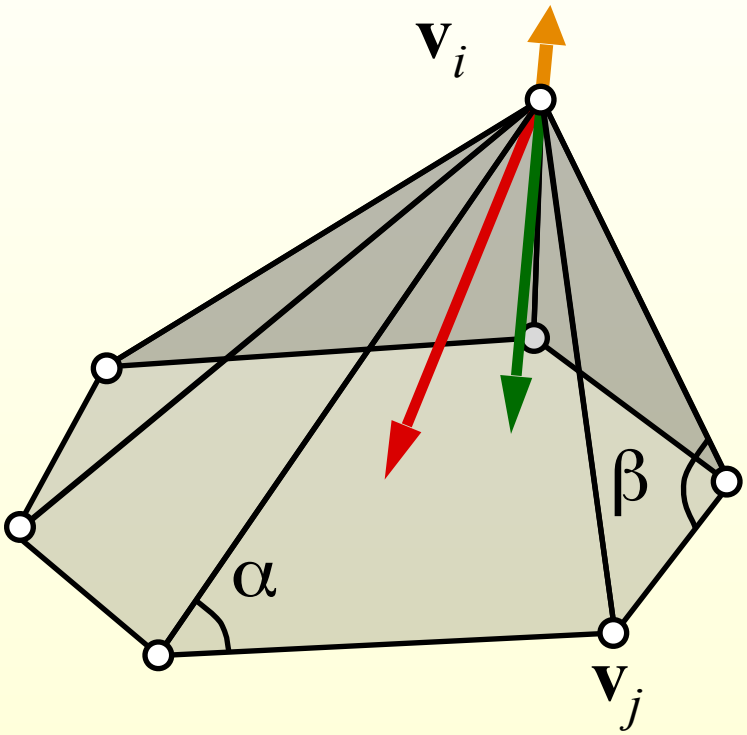
Can be derived by discretizing continuous L-B via linear Finite Elements!

Effect of the Discretization



- Uniform Laplacian $\mathbf{L}_u(\mathbf{v}_i)$
- Cotangent Laplacian $\mathbf{L}_c(\mathbf{v}_i)$
- Normal

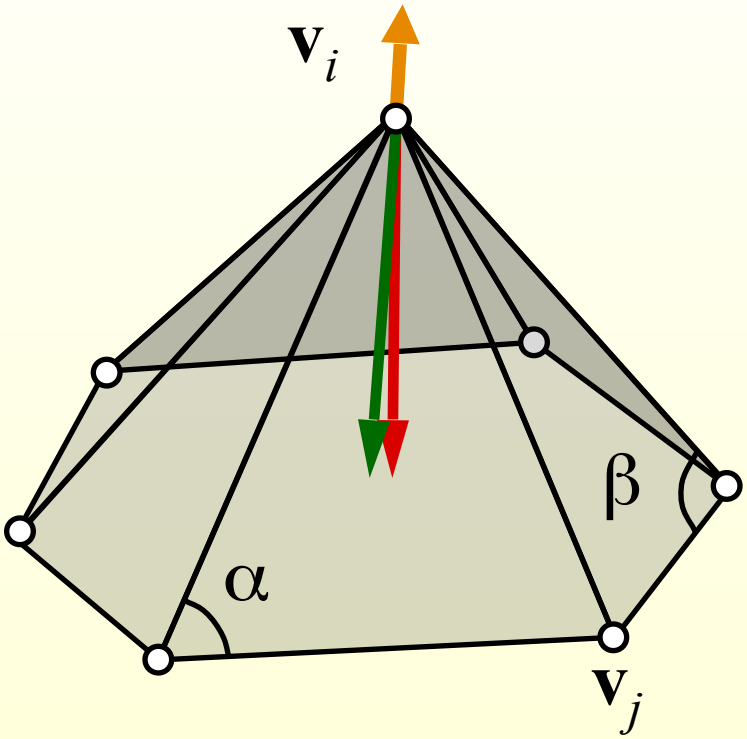
Effect of the Discretization



- Uniform Laplacian $\mathbf{L}_u(\mathbf{v}_i)$
- Cotangent Laplacian $\mathbf{L}_c(\mathbf{v}_i)$
- Normal

- For nearly equal edge lengths
  Uniform ≈ Cotangent

Effect of the Discretization



- **Uniform Laplacian $\mathbf{L}_u(\mathbf{v}_i)$**
- **Cotangent Laplacian $\mathbf{L}_c(\mathbf{v}_i)$**
- **Normal**

- **For nearly equal edge lengths**
  Uniform ≈ Cotangent

Cotangent Laplacian allows computing discrete normal

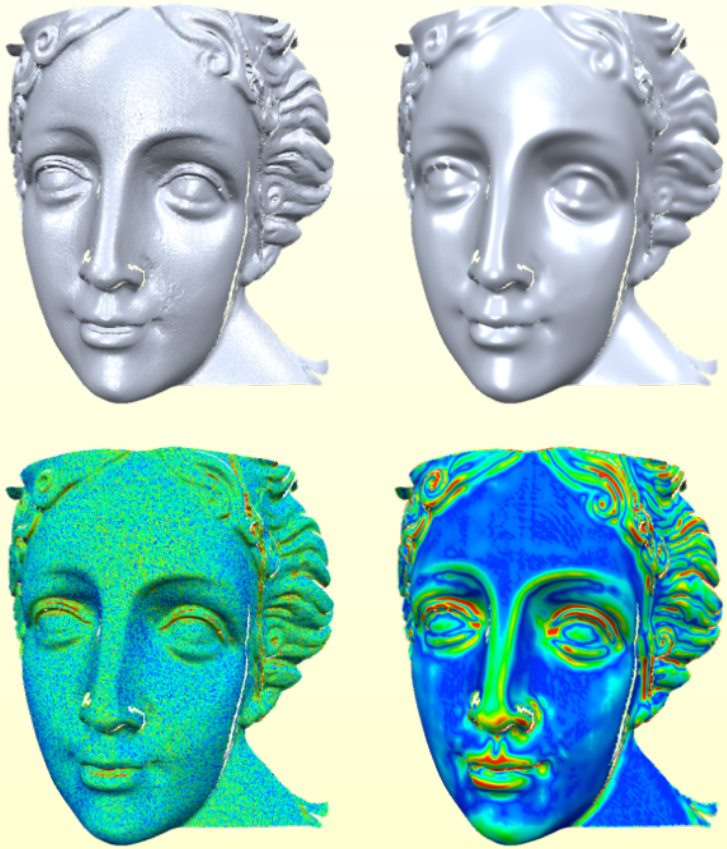
Nice property: gives zero for planar 1-rings!

How to use curvature relation for smoothing?

- Smoothing is “reducing the curvature” !

$$\Delta_{\mathcal{M}}\mathbf{p} = -2H\mathbf{n}$$

goal: $H = 0$ or $H = \text{const}$



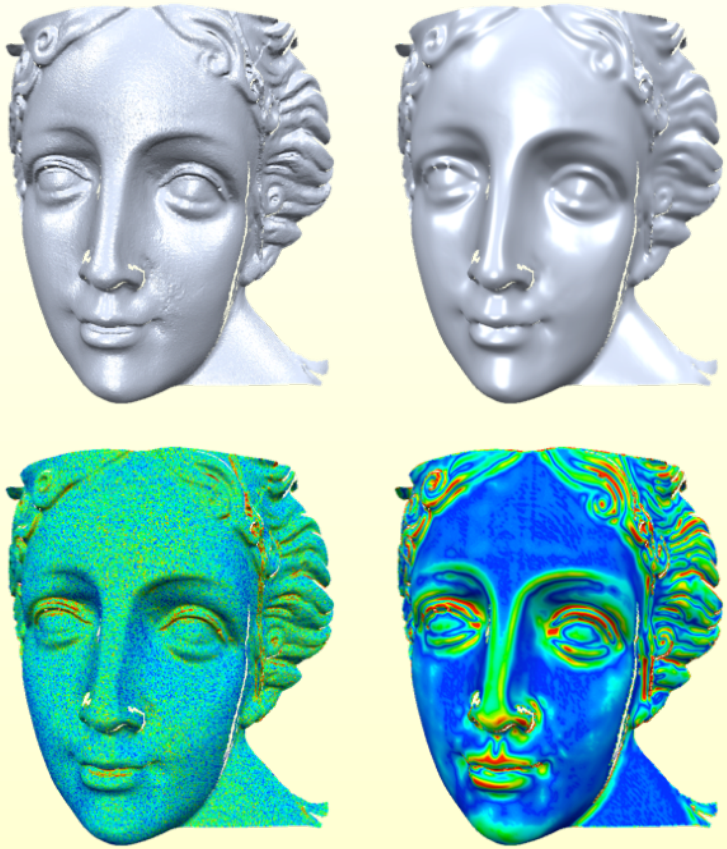
- Smooth H, obtain $\tilde{H}$
- Find a surface that has $\tilde{H}$ as mean curvature
  - H doesn’t define the surface
  - $\mathbf{n}$ nonlinear in $\mathbf{p}$

How to use curvature relation for smoothing?

- Smoothing is “reducing the curvature” !

$$\Delta_{\mathcal{M}}\mathbf{p} = -2H\mathbf{n}$$

goal: $H = 0$ or $H = \text{const}$



- Another idea:
  - Keep the old $\mathbf{n}$
  - “Flow” along $\mathbf{n}$ to decrease H

Diffusion Flow on Height Fields

- Diffusion equation

diffusion constant

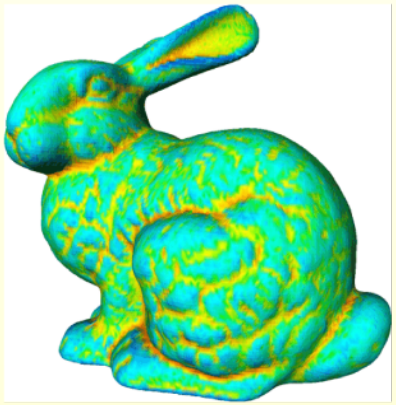
$$\frac{\partial f}{\partial t} = \lambda \Delta f$$

Laplace operator

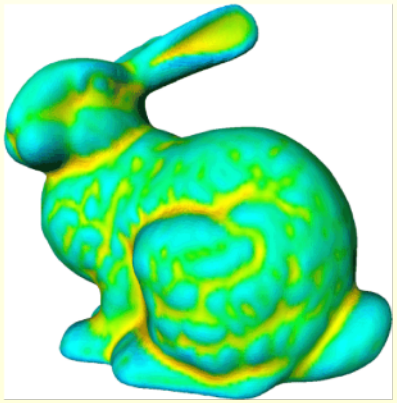
Diffusion Flow on Meshes

$$\mathbf{p}_i \leftarrow \mathbf{p}_i + \lambda \Delta \mathbf{p}_i$$

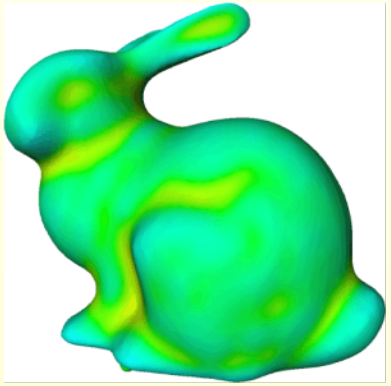
- Iterate



0 Iterations



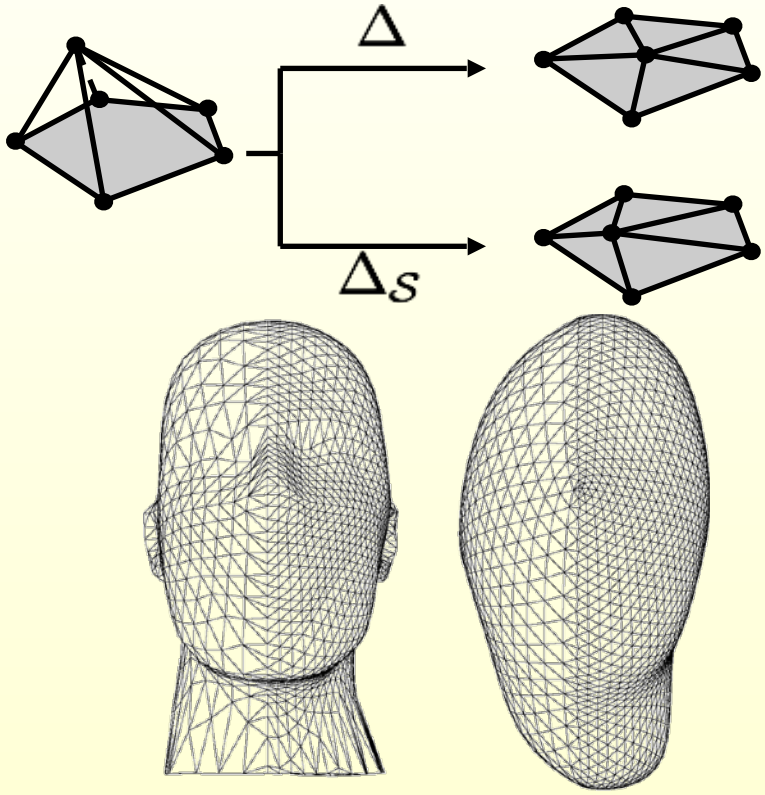
5 Iterations



20 Iterations

Effect of Laplace Discretization

- Uniform Laplace smooths geometry **and** triangulation
- Can be non-zero even for planar triangulations
- Vertex drift can lead to distortions
- Might be desired for mesh regularization

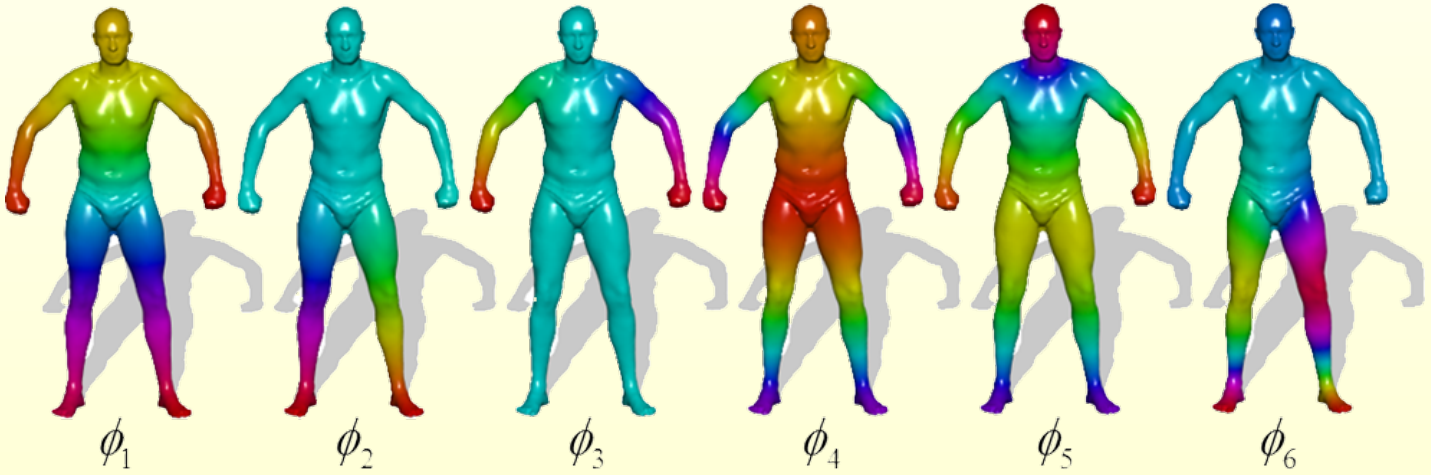


Desbrun et al., Siggraph 1999

L-B eigendecomposition

- L-B operator admits discrete decomposition

$$\Delta_M \phi = \lambda \phi \qquad 0 = \lambda_0 \leq \lambda_1 \leq \lambda_2 \leq \ldots$$



- $\{\phi_0, \phi_1, \ldots\}$ comprise basis for L_2 functions defined on M
- Function basis is adapted to the object

Eigenfunction Nodal Domains

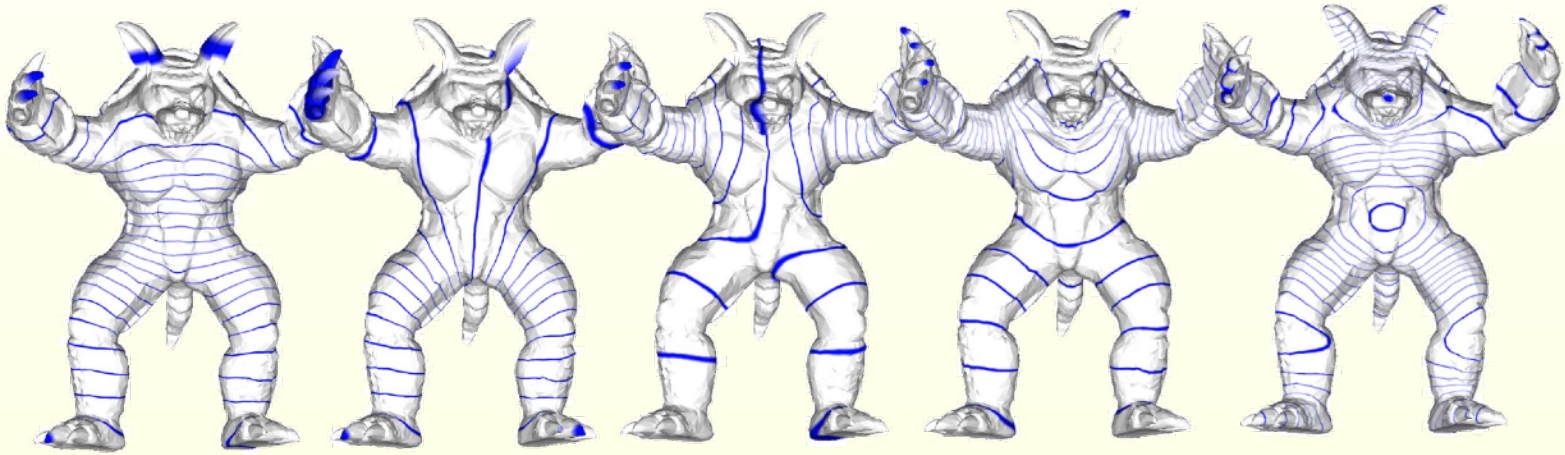


Figure 3. Contours of the first eigenfunctions. Note how the protrusions and symmetries are captured. The eigenfunctions combine the geometric and topological information contained in the shape signal.

Image credit: Bruno Lévy

A Little Bit of History: Chladni Plates

- Helmholtz's wave propagation equation $\Delta f = \lambda f$

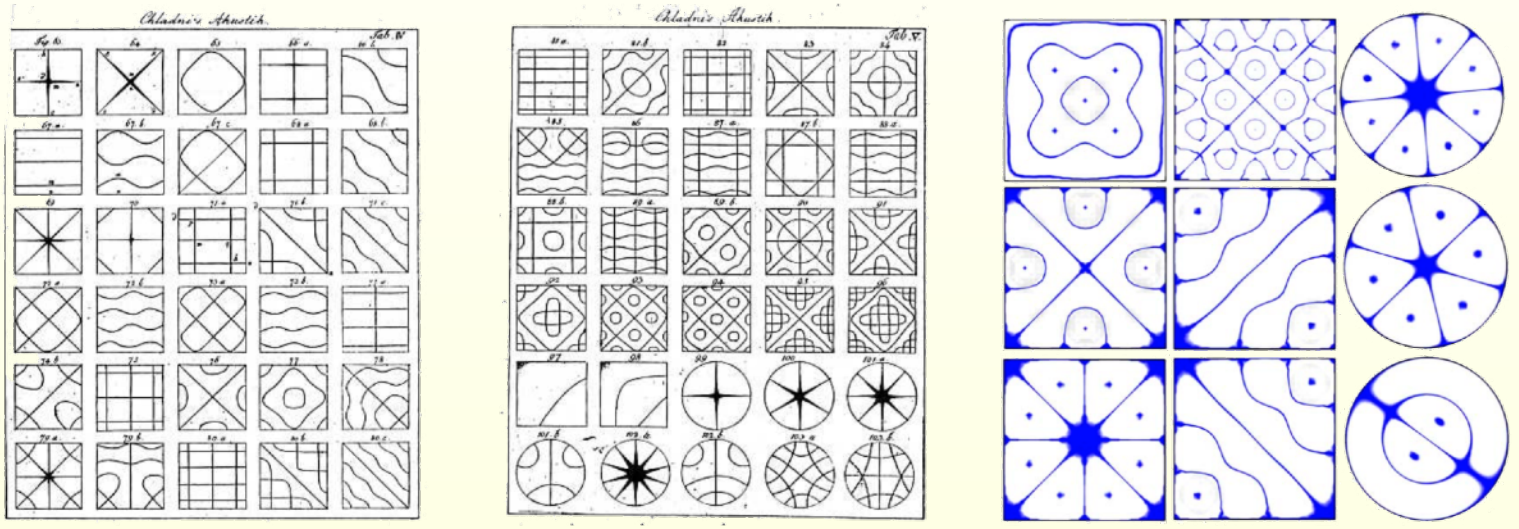


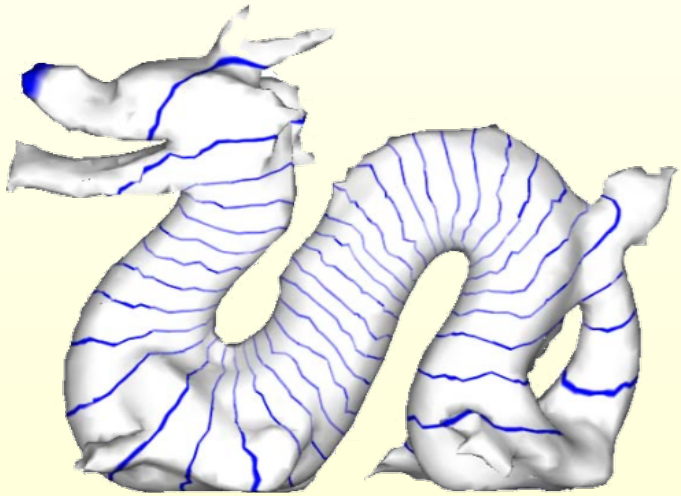
Figure 2. Left: In the late 1700's, the physicist Ernst Chaldni was amazed by the patterns formed by sand on vibrating metal plates. Right: numerical simulations obtained with our approach.

- https://www.youtube.com/watch?v=lRFysSAxWxI

Image credit: Bruno Lévy

Fiedler vector

- The second eigenvector



- "The Fiedler vector gives a natural ordering of the nodes of a graph" (Lévy 2006)

Dragon image credit: Bruno Lévy

Uniform vs. Cotangent Weights

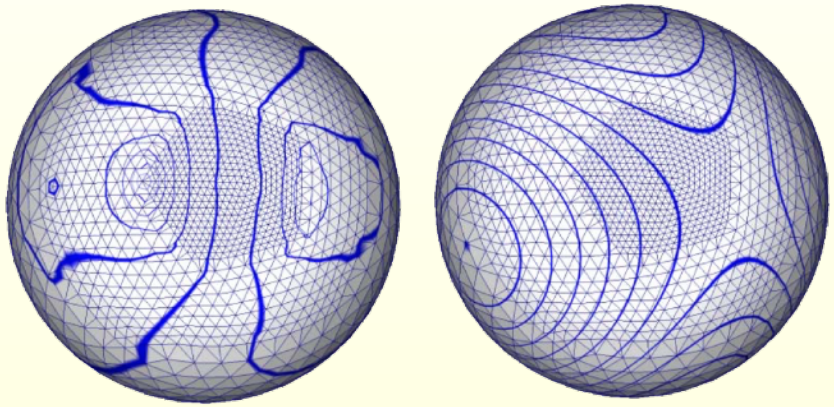


Figure 4. Contours of the 4th eigenfunction, computed from the Graph Laplacian (left) and cotangent weights (right) on an irregular mesh.

- Using only the connectivity of the graph may lead to highly distorted mappings

Image credit: Bruno Lévy

Spectral analysis on meshes

- Take your favorite L-B matrix L
- Compute eigenvectors $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_k$ with the k smallest eigenvalues
- Reconstruct mesh geometry (= coordinate functions, e.g. $f(x,y,z) = x$) from the eigenvectors:

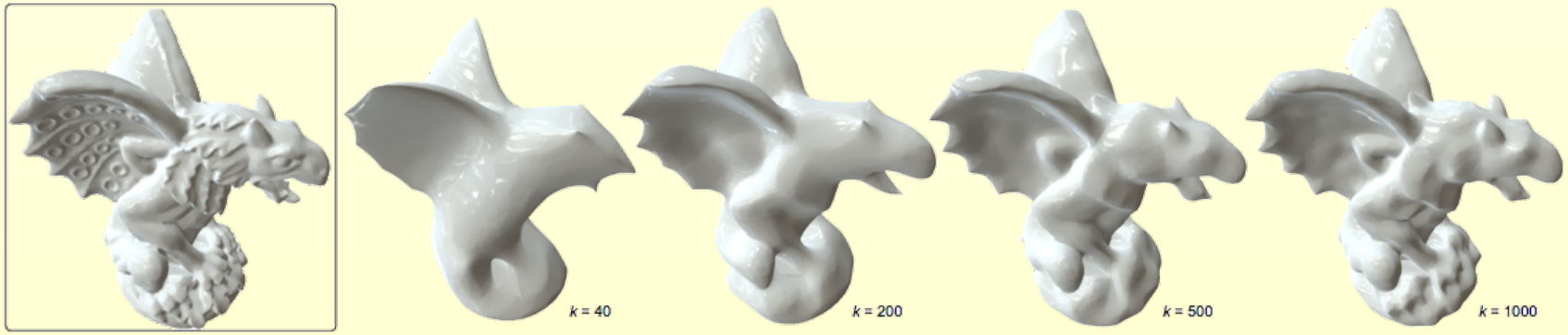
$$\mathbf{x} = [x_1, \ldots, x_n]^T \qquad \mathbf{y} = [y_1, \ldots, y_n]^T \qquad \mathbf{z} = [z_1, \ldots, z_n]^T$$

$$\tilde{\mathbf{x}} = \sum_{i=1}^{k} (\mathbf{x}^T \mathbf{e}_i)\, \mathbf{e}_i \qquad \tilde{\mathbf{y}} = \sum_{i=1}^{k} (\mathbf{y}^T \mathbf{e}_i)\, \mathbf{e}_i \qquad \tilde{\mathbf{z}} = \sum_{i=1}^{k} (\mathbf{z}^T \mathbf{e}_i)\, \mathbf{e}_i$$

$$\tilde{\mathbf{p}} = [\tilde{\mathbf{x}}\ \tilde{\mathbf{y}}\ \tilde{\mathbf{z}}] \in \mathbb{R}^{n \times 3}$$

Spectral analysis on meshes

- Take your favorite L-B matrix L
- Compute eigenvectors $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_k$ with the k smallest eigenvalues
- Reconstruct mesh geometry (= coordinate functions, e.g. $f(x, y, z) = x$) from the eigenvectors:



Information Transfer

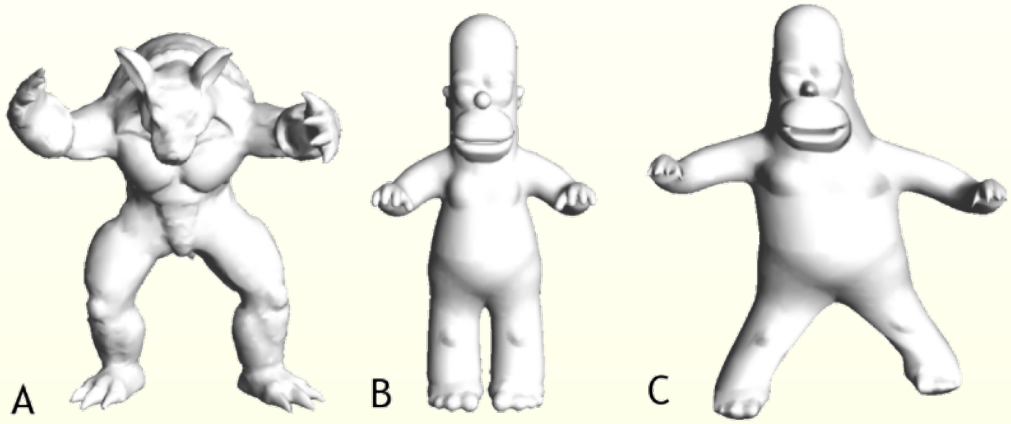


Figure 7. Pose transfer: the 5 first coefficients of the Armadillo (A) were copied to Homer (B) and adapted its pose to the Armadillo (C)

- More - in the following lectures

Image credit: Bruno Lévy

Additional reading

- “Laplace-Beltrami Eigenfunctions: Towards an algorithm that “understands” geometry” by Bruno Lévy, 2006
- “Discrete Differential-Geometry Operators for Triangulated 2-Manifolds” by Meyer et al., 2003
- “Laplace-Beltrami spectra as ”shape-DNA” of surfaces and solids” by Reuter et al., 2005
- “Discrete Laplace operators: No free lunch” by Wardetzky et al., 2007
- “Laplace-Beltrami Eigenfunctions for Deformation Invariant Shape Representation” by Rustamov, 2007