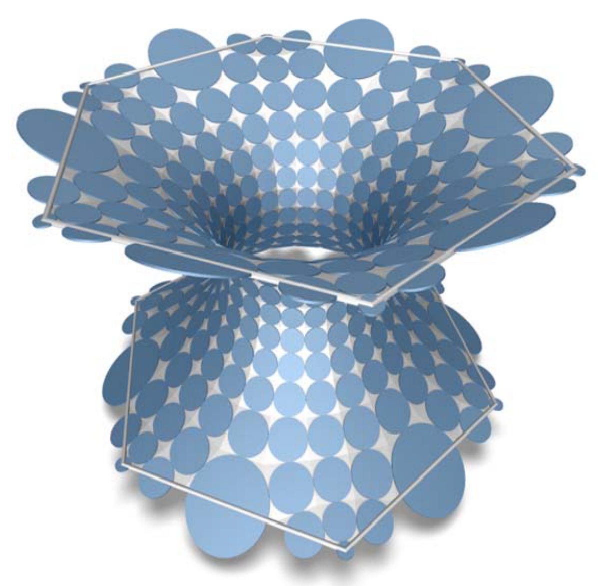
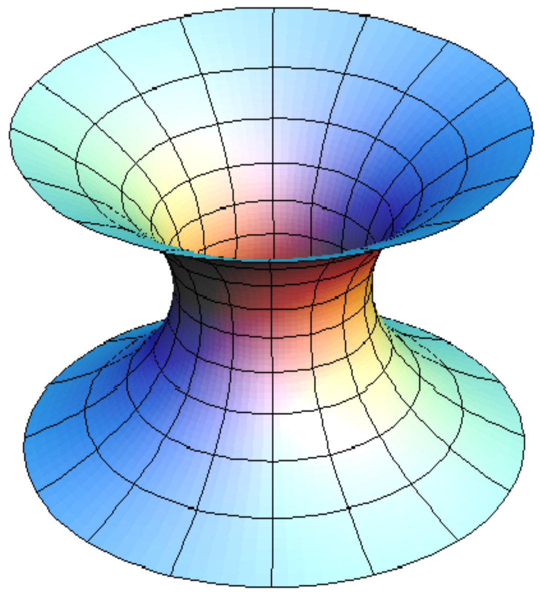
UC San Diego





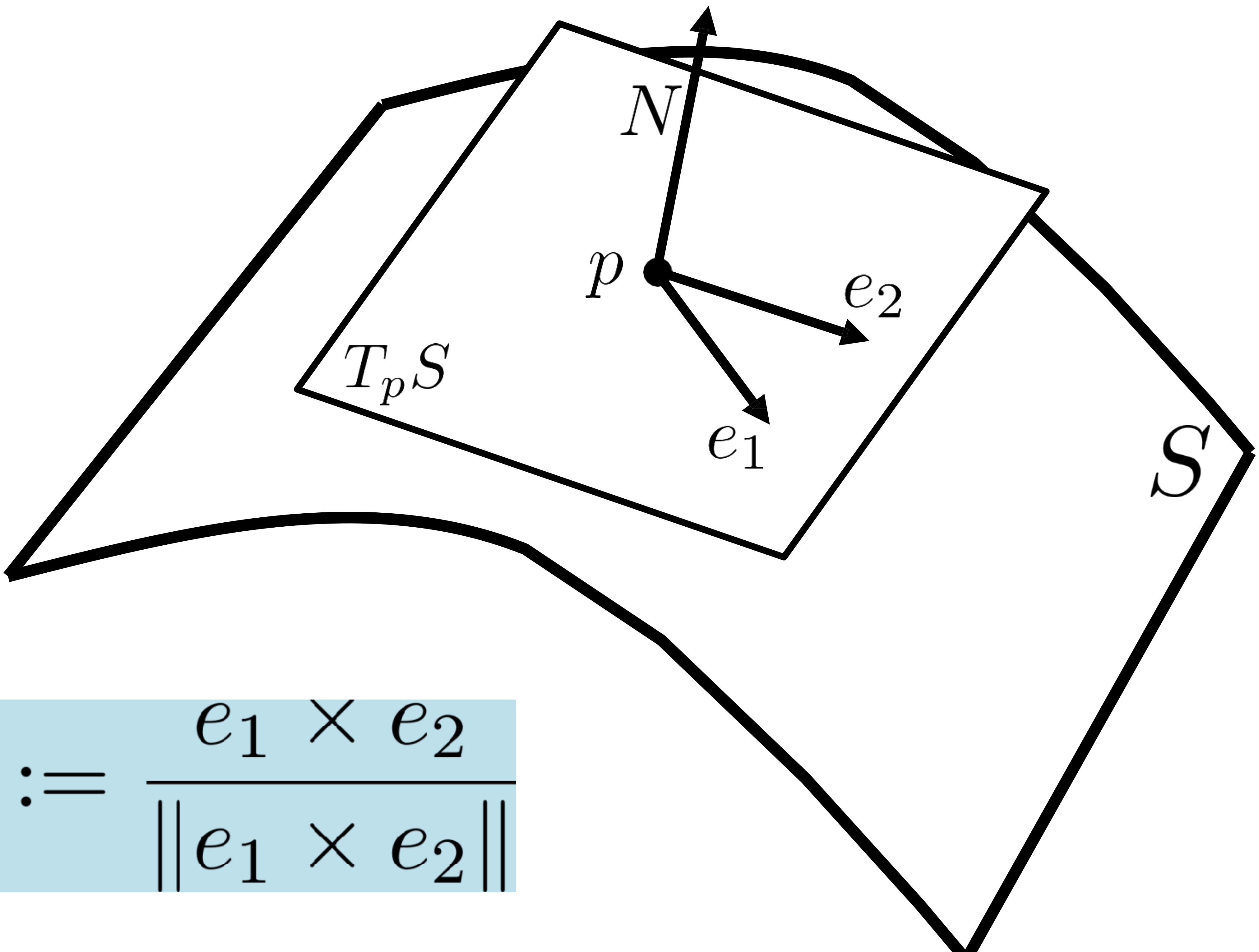
Computation for Surface Geometry

Instructor: Hao Su

Slides credits: Mira Ben-Chen (CS468 taught in 2012 at Stanford)

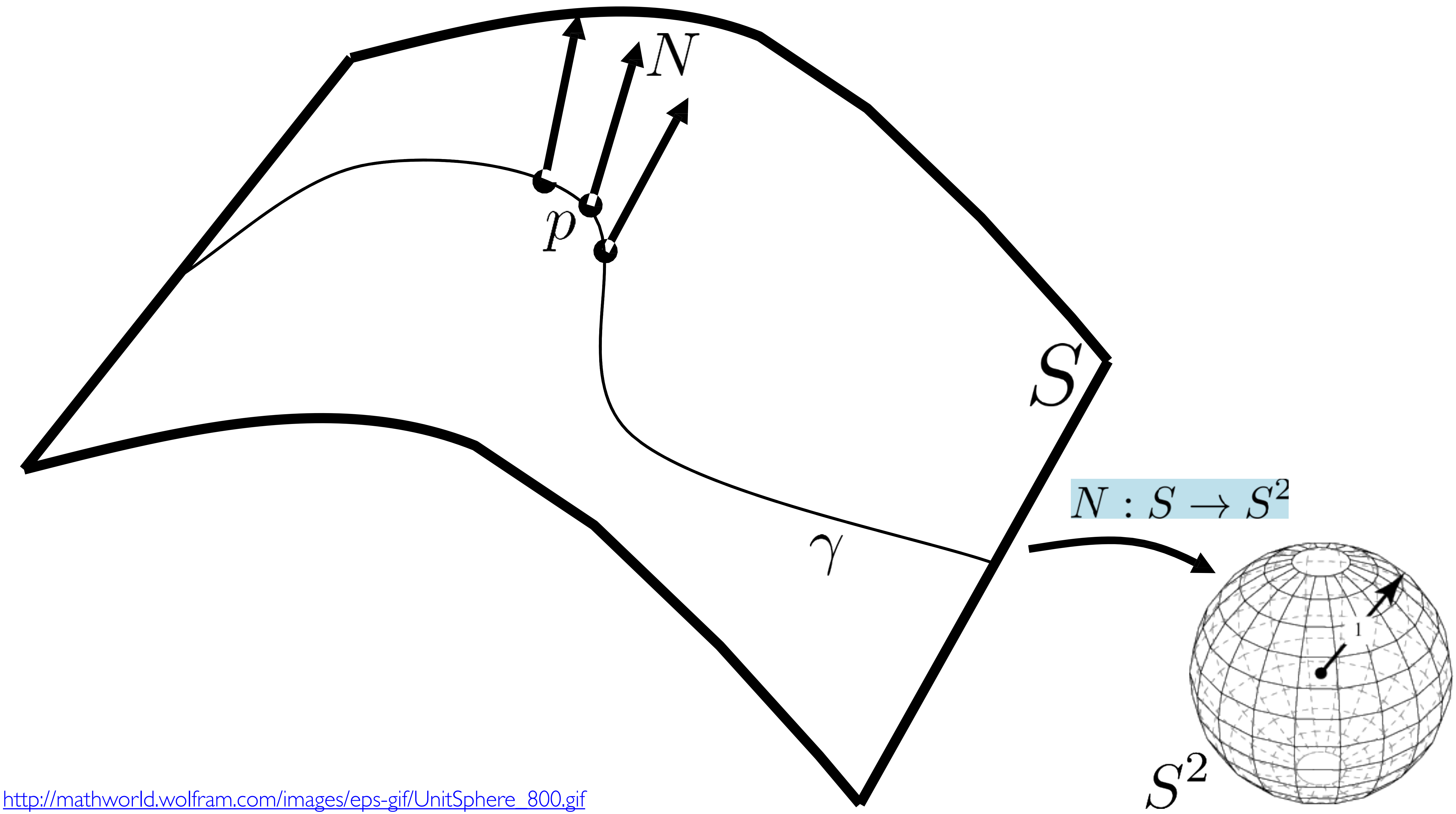
Recap

Unit Normal



$$N := \frac{e_1 \times e_2}{\|e_1 \times e_2\|}$$

Gauss Map for Surface



http://mathworld.wolfram.com/images/eps-gif/UnitSphere_800.gif

Gauss Map for Surface

$$\varphi : M \to N$$

$$\implies d\phi_p : T_pM \to T_{\varphi(p)}N$$

$$d\varphi_p(\gamma'(0)) := (\varphi \circ \gamma)'(0)$$

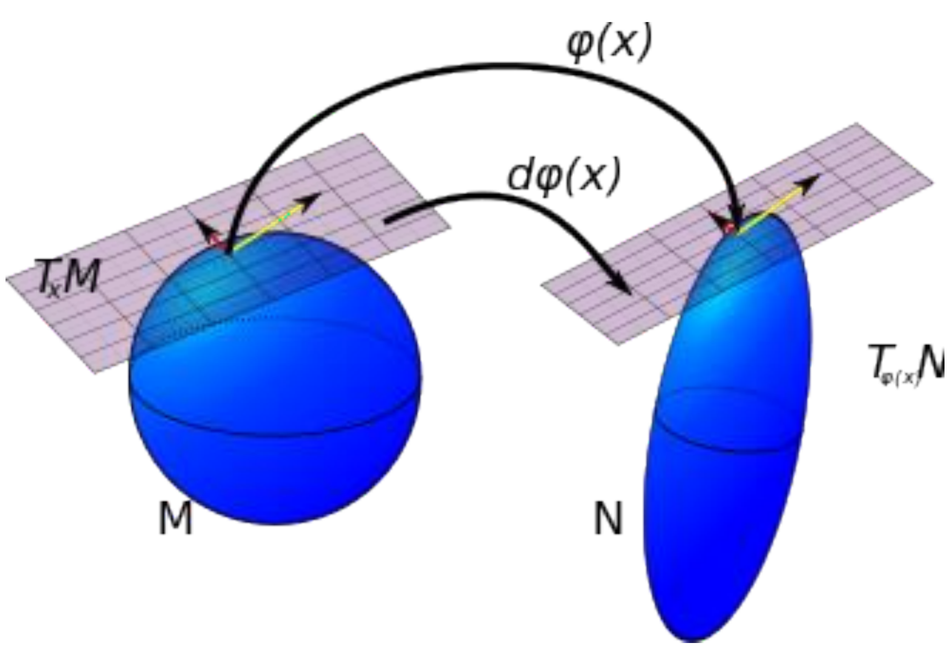


Image from Wikipedia

Shape Operator

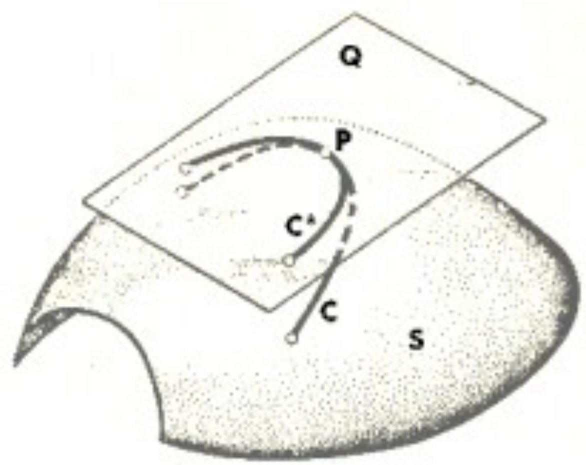
$$DN_p : T_pS \to T_pS$$

$$\downarrow$$

$$A_p(V, W) := -\langle DN_p(V), W \rangle$$

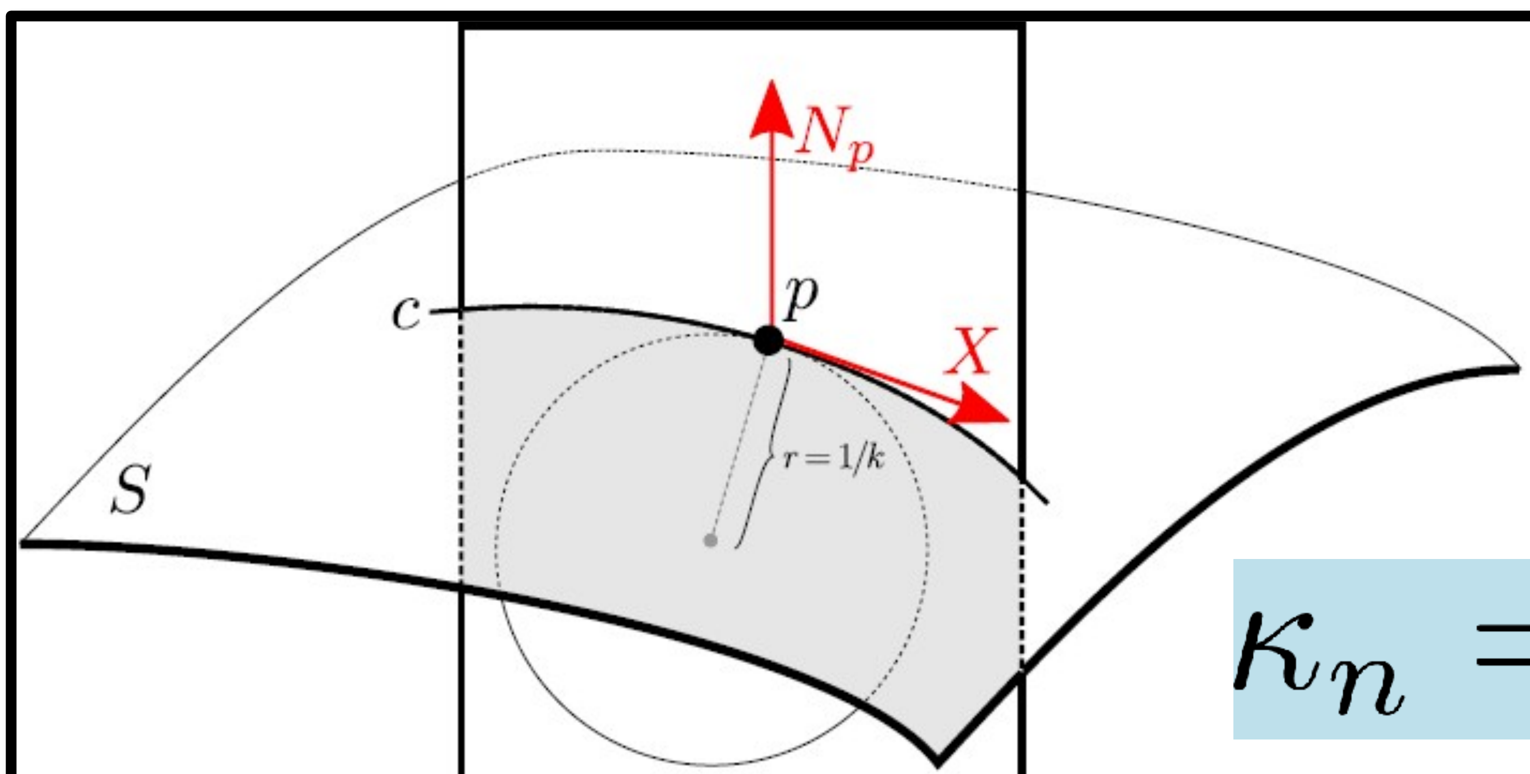
"Shape operator"

Normal Curvature



$$\kappa_g := \vec{\kappa} \cdot (\vec{N} \times \vec{T})$$

http://www.solitaryroad.com/c335.html



Drawing by Adrian Butscher

$$\kappa_n = A_p(X, X)$$

Computation
(continuous case)

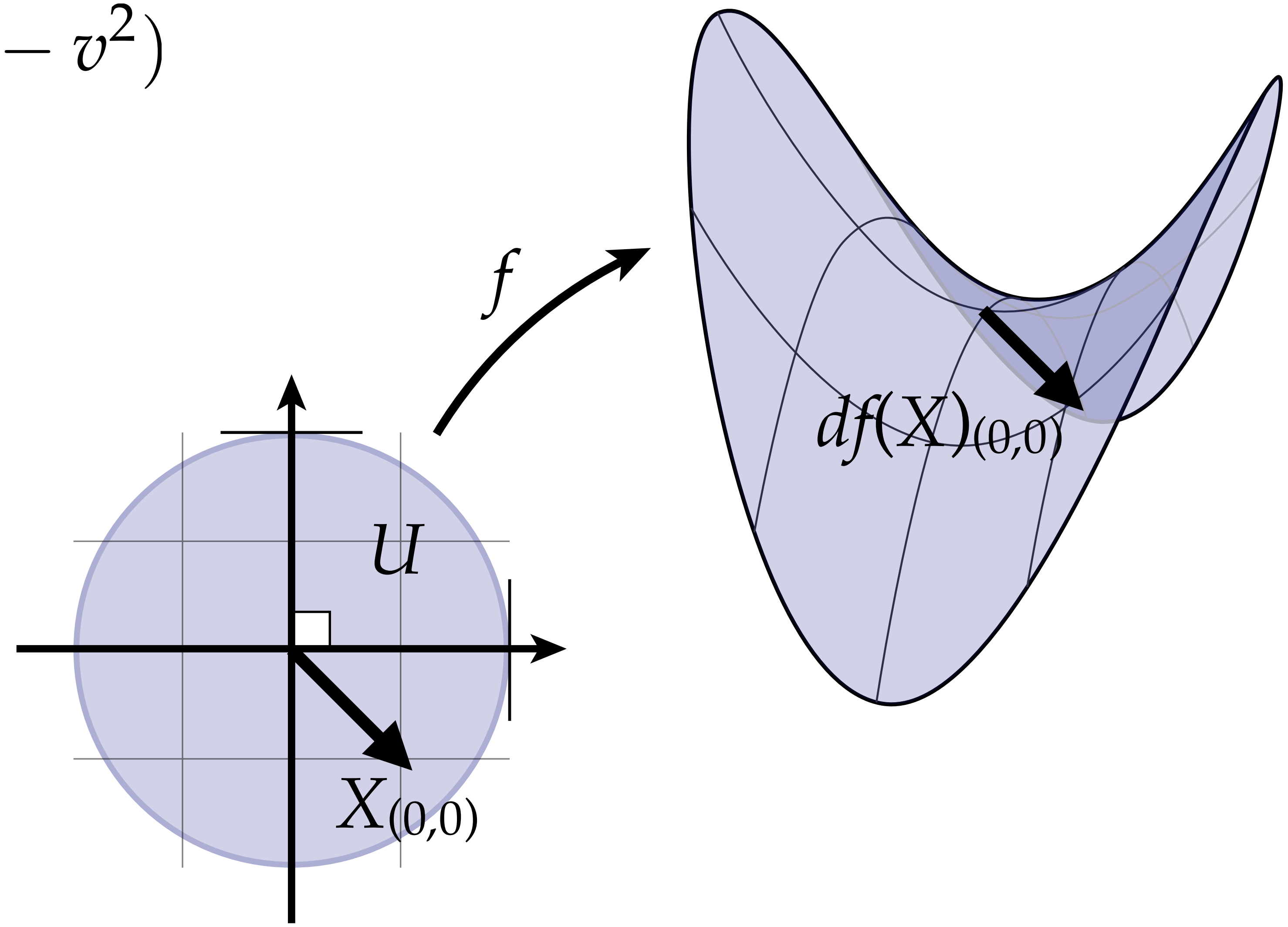
Differential in Coordinates

In coordinates, the differential is simply the exterior derivative:

$$f : U \to \mathbb{R}^3;\ (u, v) \mapsto (u, v, u^2 - v^2)$$

$$df = \frac{\partial f}{\partial u} du + \frac{\partial f}{\partial v} dv =$$

$$(1, 0, 2u)du + (0, 1, -2v)dv$$



<u>Pushforward of a vector field:</u>

$$X := \tfrac{3}{4}\left(\tfrac{\partial}{\partial u} - \tfrac{\partial}{\partial v}\right)$$

$$df(X) = \tfrac{3}{4}(1, -1, 2(u+v))$$

E.g., at $u=v=0$: $\left(\frac{3}{4}, -\frac{3}{4}, 0\right)$

Some Rules

1. Definition: $\nabla_X f = Xf$

   $Xf = (\sum_{i=1}^{2} X^i \frac{\partial}{\partial u_i})f = \sum_{i=1}^{2} X^i \frac{\partial f}{\partial u_i}$

2. Definition: $\nabla_X \overrightarrow{y} = \nabla_X(y_1, y_2, y_3) = (\nabla_X y_1, \nabla_X y_2, \nabla_X y_3) = (Xf_1, Xf_2, Xf_3)$

3. Definition: $\mathrm{d}Y(X) = \nabla_X \overrightarrow{y}$

   Weingarten map: $\mathrm{d}N(X) = \nabla_X \overrightarrow{n}$

4. Leibniz rule: $\nabla_X \langle \overrightarrow{y}, \vec{z} \rangle = \langle \nabla_X \overrightarrow{y}, \vec{z} \rangle + \langle \overrightarrow{y}, \nabla_X \vec{z} \rangle$

5. $\nabla_X \overrightarrow{y} = \nabla_Y \overrightarrow{x}$ for $\overrightarrow{x} = \mathrm{d}\phi(X)$ and $\overrightarrow{y} = \mathrm{d}\phi(Y)$

   Proof:

   $\nabla_X \overrightarrow{y} = \nabla_X \mathrm{d}\phi(Y) = \sum X^i \frac{\partial}{\partial u^i}(\sum \frac{\partial \phi}{\partial u^j} du^j \sum Y^k \frac{\partial}{\partial u^k}) = \sum_{i,j} X^i Y^j \frac{\partial^2 \phi}{\partial u^i \partial u^j}$

6. Second fundamental form: $\mathrm{II}(X, Y) = \langle dN(X), d\phi(Y) \rangle = \langle \nabla_X \overrightarrow{n}, \overrightarrow{y} \rangle$

   Note that $\nabla_X \langle \overrightarrow{n}, \overrightarrow{y} \rangle = \langle \nabla_X \overrightarrow{n}, \overrightarrow{y} \rangle + \langle \overrightarrow{n}, \nabla_X \overrightarrow{y} \rangle$ and $\langle \overrightarrow{n}, \overrightarrow{y} \rangle = 0$

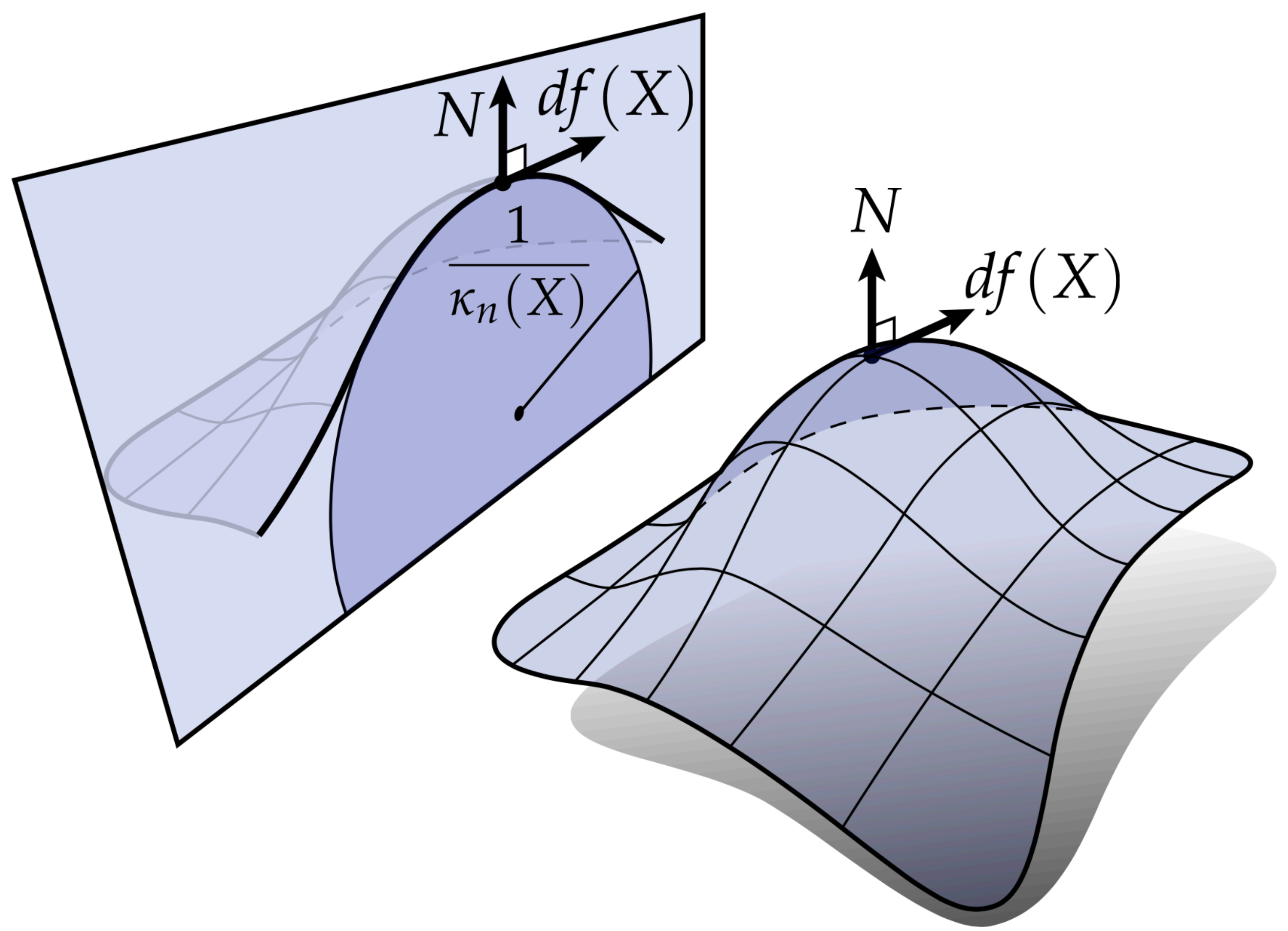
   so $\langle \nabla_X \overrightarrow{n}, \overrightarrow{y} \rangle = -\langle \overrightarrow{n}, \nabla_X \overrightarrow{y} \rangle = -\langle \overrightarrow{n}, \nabla_Y \overrightarrow{x} \rangle = \langle \nabla_Y \overrightarrow{n}, \overrightarrow{x} \rangle$

   $\mathrm{II}(X, Y) = \mathrm{II}(Y, X)$, i.e., the second fundamental form is a symmetric bilinear form

Normal Curvature

- Assume parameterization is represented as $f : U \to \mathbb{R}^3$

$$\kappa_N(X) := \frac{\langle df(X), dN(X) \rangle}{\|df(X)\|^2}$$



Normal Curvature Example

Consider a parameterized cylinder:

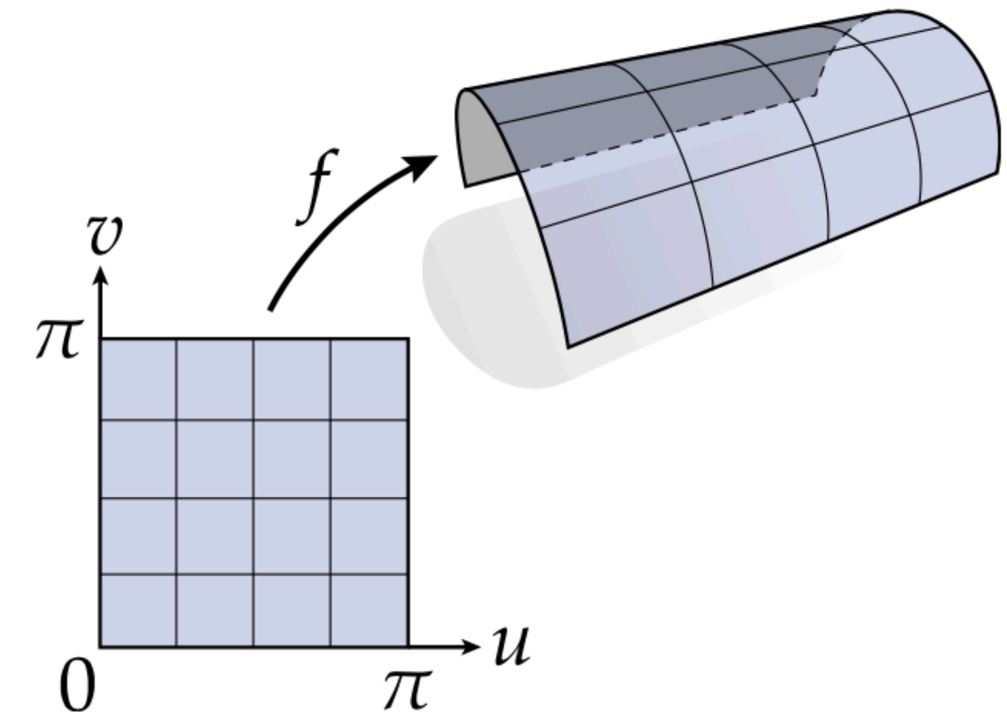
$$f(u,v) := (\cos(u), \sin(u), v)$$

$$df = (-\sin(u), \cos(u), 0)du + (0,0,1)dv$$

$$N = (-\sin(u), \cos(u), 0) \times (0,0,1)$$

$$= (\cos(u), \sin(u), 0)$$

$$dN = (-\sin(u), \cos(u), 0)du$$



$$\kappa_N(\tfrac{\partial}{\partial u}) = \frac{\langle df(\frac{\partial}{\partial u}), dN(\frac{\partial}{\partial u})\rangle}{|df(\frac{\partial}{\partial u})|^2} = \frac{(-\sin(u),\cos(u),0)\cdot(-\sin(u),\cos(u),0)}{|(-\sin(u),\cos(u),0)|^2} = 1$$

$$\kappa_N(\tfrac{\partial}{\partial v}) = \cdots = 0$$

Q: Does this result make sense geometrically?

Shape Operator Example

Consider a parameterized cylinder:

Gauss curvature: 0
Mean curvature: 0.5

$$f(u,v) := (\cos(u), \sin(u), v)$$

$$df = (-\sin(u), \cos(u), 0)du + (0,0,1)dv$$

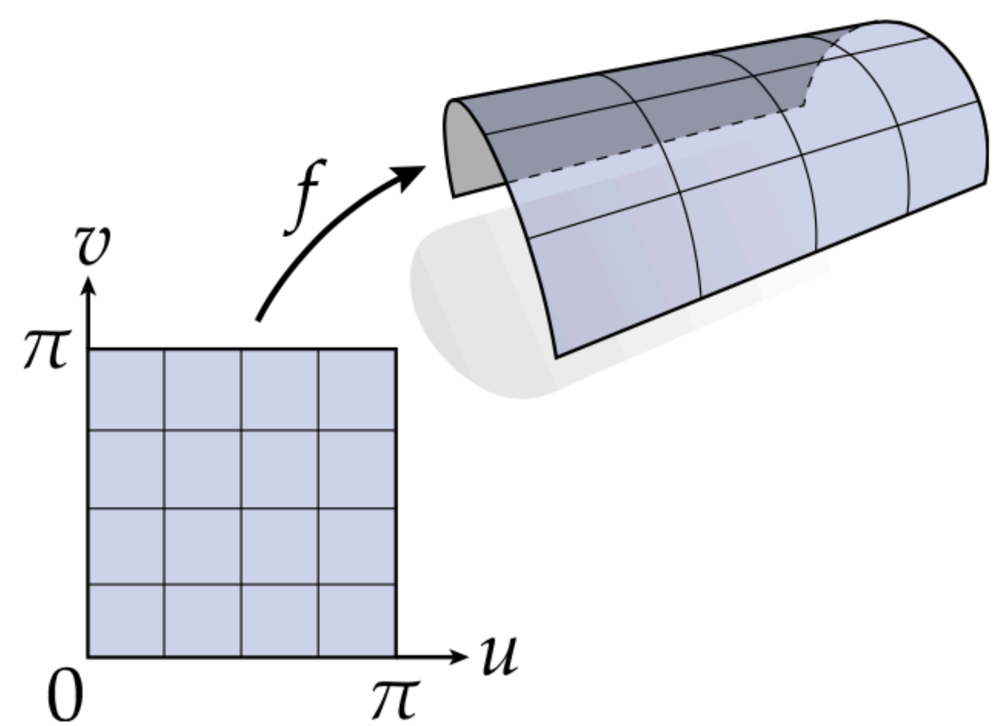
$$dN = (-\sin(u), \cos(u), 0)du$$

Consider a pair of directions $X = x_1 \frac{\partial}{\partial u} + x_2 \frac{\partial}{\partial v}$ **and** $Y = y_1 \frac{\partial}{\partial u} + y_2 \frac{\partial}{\partial v}$

$$df(Y) = (-y_1 \sin(u), y_1 \cos(u), y_2)$$

$$dN(X) = (-x_1 \sin(u), x_1 \cos(u), 0)$$

$$\begin{aligned} A(X,Y) &= \langle dN(X), df(Y) \rangle \\ &= \langle (-x_1 \sin(u), x_1 \cos(u), 0), (-y_1 \sin(u), y_1 \cos(u), y_2 \rangle \\ &= x_1 y_1 \sin^2(u) + x_1 y_1 \cos^2(u) = x_1 y_1 \\ &= [x_1, x_2] \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \end{aligned}$$

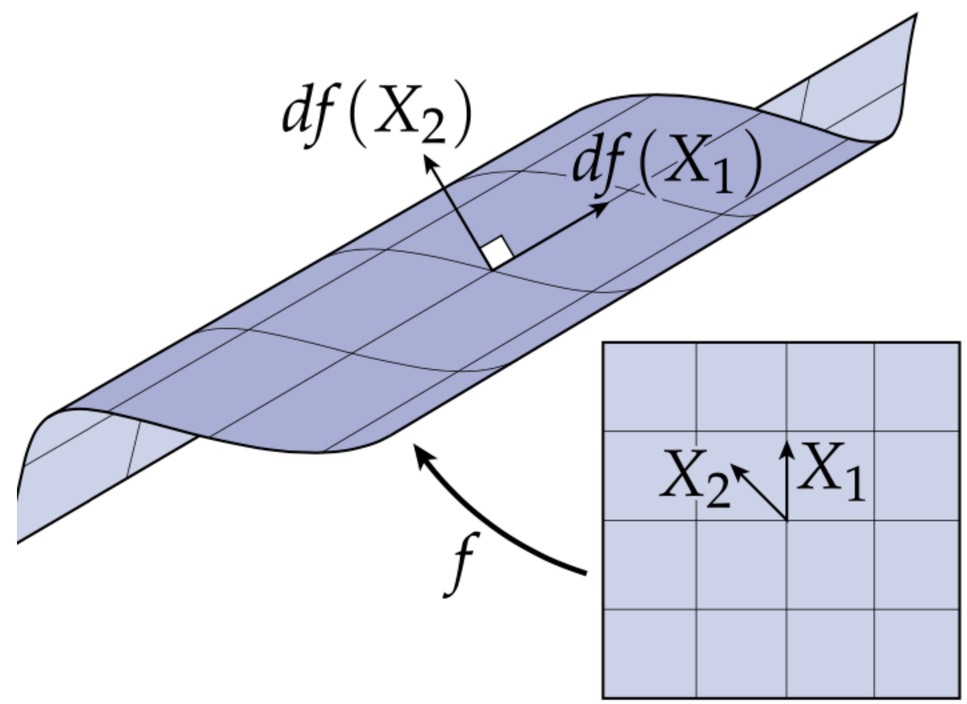


Exercise (I)

Consider a nonstandard parameterization of the cylinder (*sheared* along z):

$$f(u,v) := (\cos(u), \sin(u), u+v) \qquad df = (-\sin(u), \cos(u), 1)du + (0,0,1)dv$$

$$N = (\cos(u), \sin(u), 0) \qquad dN = (-\sin(u), \cos(u), 0)du$$



$$\kappa_1 = 0$$

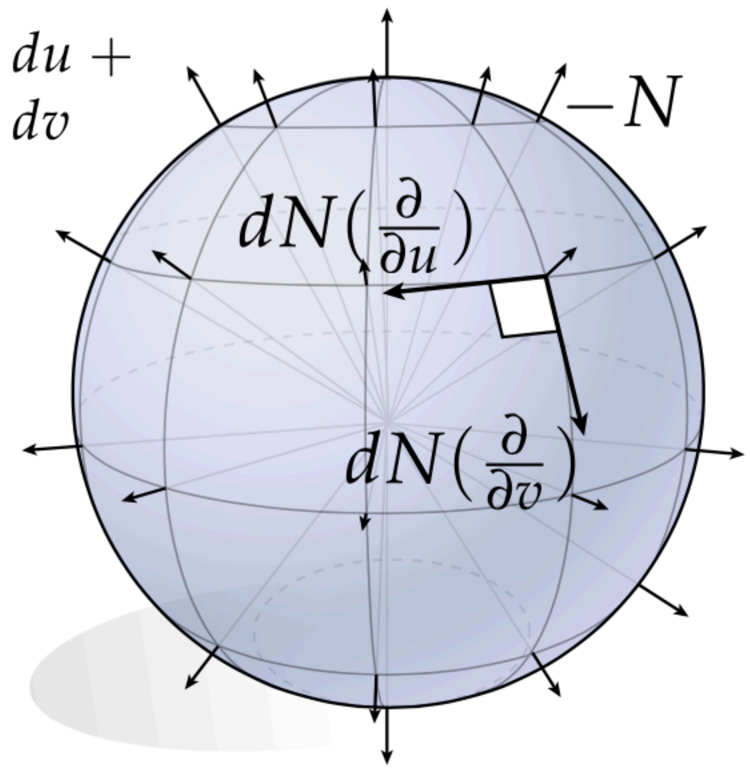
$$\kappa_2 = 1$$

Exercise (II)

$$f := (\cos(u)\sin(v), \sin(u)\sin(v), \cos(v))$$

$$df = \begin{matrix} (& -\sin(u)\sin(v), & \cos(u)\sin(v), & 0 &) & du\ + \\ (& \cos(u)\cos(v), & \cos(v)\sin(u), & -\sin(v) &) & dv \end{matrix}$$

$$dN = \begin{matrix} (& \sin(u)\sin(v), & -\cos(u)\sin(v), & 0 &) & du \\ (& -\cos(u)\cos(v), & -\cos(v)\sin(u), & \sin(v) &) & dv \end{matrix}$$



$$\kappa_1 = \kappa_2 = \frac{1}{r}$$

Theorema Egregium

- The Gaussian curvature of a surface can be expressed by the length on the surface only, i.e., it is *intrinsic*

Hessian Matrix And Gauss Curvature Example

Let us first compute the Gaussian curvature of a graph.

Let S be the surface given by $z = f(x, y)$, where $f: \mathbb{R}^2 \to \mathbb{R}$ is smooth. The first fundamental form of S is:

$$g(X, Y) = \left(1 + (\partial_x f)^2\right) X^2 + 2(\partial_x f)(\partial_y f)XY + \left(1 + (\partial_y f)^2\right) Y^2.$$

Besides, the second fundamental form of S is:

$$h(X, Y) = \frac{1}{\sqrt{1 + (\partial_x f)^2 + (\partial_y f)^2}} \left((\partial_{xx} f)X^2 + 2(\partial_{xy} f)XY + (\partial_{yy} f)Y^2\right).$$

Hence, the Gaussian curvature of S is given by:

$$K = \frac{\det(h)}{\det(g)} = \frac{(\partial_{xx} f)(\partial_{yy} f) - (\partial_{xy} f)^2}{(1 + (\partial_x f)^2 + (\partial_y f)^2)^2}.$$

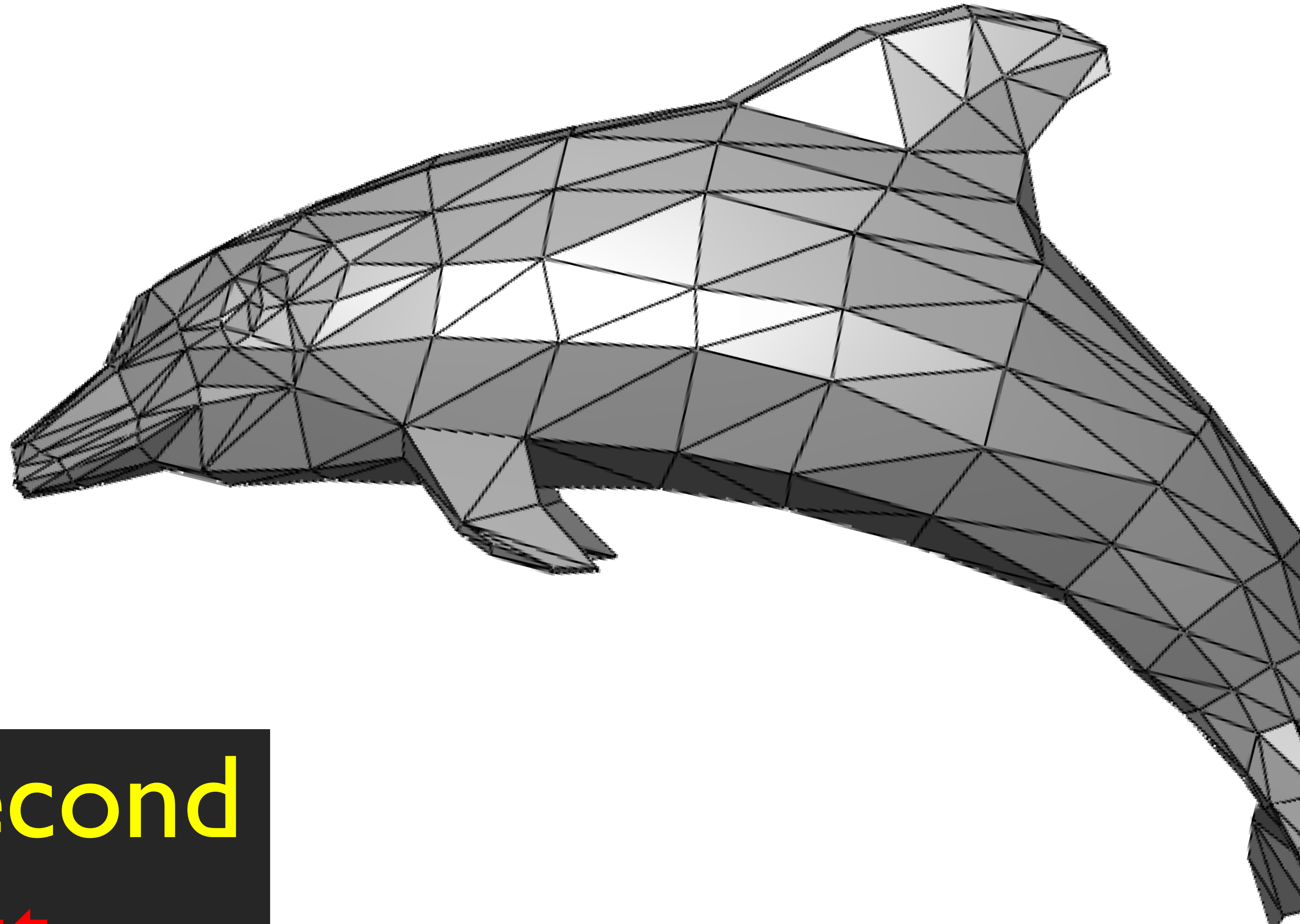
[You can compute the Gaussian curvature using the definition, but I did not want to differentiate a quotient.]

- Now, at a critical point, one has $\partial_x f = 0 = \partial_y f$, so that the Gaussian curvature boils down to:

$$K = (\partial_{xx} f)(\partial_{yy} f) - (\partial_{xy} f)^2 = \det(\text{Hess. } f).$$

Computation
(discrete case)

Challenge on Meshes



Curvature is a second derivative, but triangles are flat.

http://upload.wikimedia.org/wikipedia/commons/f/fb/Dolphin_triangle_mesh.png

Standard Citation

ESTIMATING THE TENSOR OF CURVATURE OF A SURFACE FROM A POLYHEDRAL APPROXIMATION

ICCV 1995

Gabriel Taubin

IBM T.J.Watson Research Center
P.O.Box 704, Yorktown Heights, NY 10598
taubin@watson.ibm.com

Abstract

Estimating principal curvatures and principal directions of a surface from a polyhedral approximation with a large number of small faces, such as those produced by iso-surface construction algorithms, has become a basic step in many computer vision algorithms. Particularly in those targeted at medical applications. In this paper we describe a method to estimate the tensor of curvature of a surface at the vertices of a polyhedral approximation. Principal curvatures and principal directions are obtained by computing in closed form the eigenvalues and eigenvectors of certain 3×3 symmetric matrices defined by integral formulas, and

mate principal curvatures at the vertices of a triangulated surface. Both this algorithm and ours are based on constructing a quadratic form at each vertex of the polyhedral surface and then computing eigenvalues (and eigenvectors) of the resulting form, but the quadratic forms are different. In our algorithm the quadratic form associated with a vertex is expressed as an integral, and is constructed in time proportional to the number of neighboring vertices. In the algorithm of Chen and Schmitt, it is the least-squares solution of an overdetermined linear system, and the complexity of constructing it is quadratic in the number of neighbors.

2 The Tensor of Curvature

Taubin Matrix

$$M := \frac{1}{2\pi} \int_{-\pi}^{\pi} \kappa_\theta T_\theta T_\theta^\top \, d\theta$$

$$\kappa_\theta := \kappa_1 \cos^2\theta + \kappa_2 \sin^2\theta$$

$$T_\theta := T_1 \cos\theta + T_2 \sin\theta$$

Taubin Matrix

$$M := \frac{1}{2\pi}\int_{-\pi}^{\pi} \kappa_\theta T_\theta T_\theta^\top \, d\theta$$

- Eigenvectors are N, T_1, and T_2
- Eigenvalues are $\frac{3}{8}\kappa_1 + \frac{1}{8}\kappa_2$ and $\frac{1}{8}\kappa_1 + \frac{3}{8}\kappa_2$

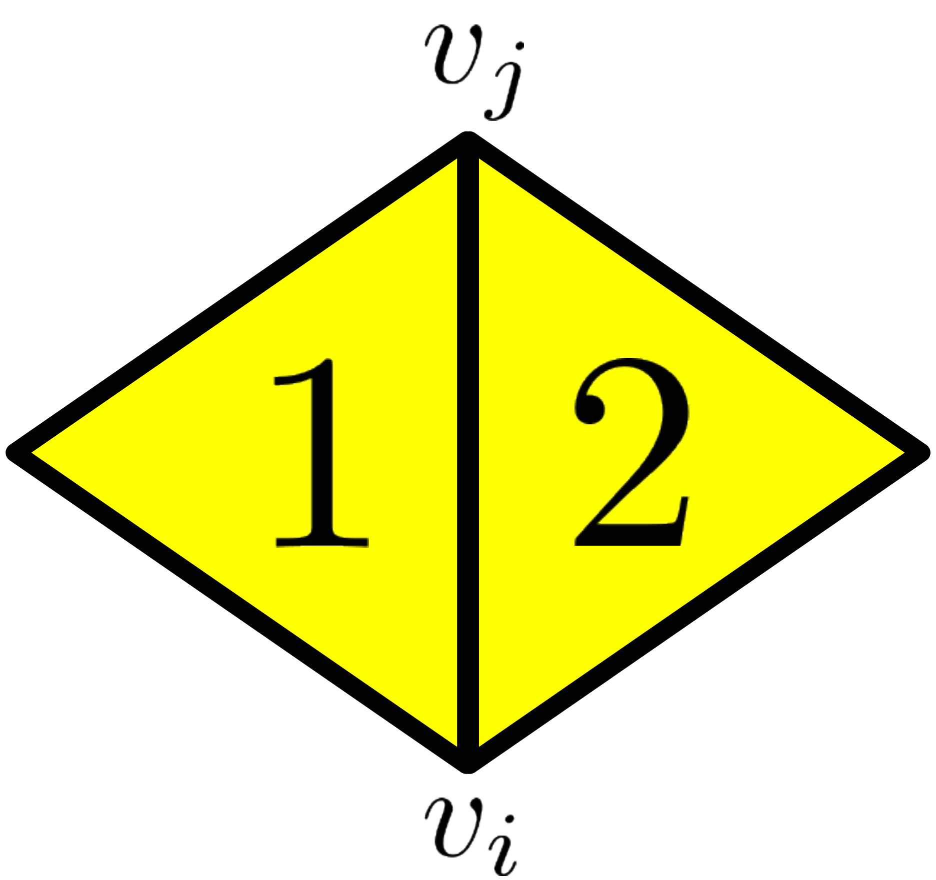
Prove at home!

Taubin's Approximation

$$M := \frac{1}{2\pi} \int_{-\pi}^{\pi} \kappa_\theta T_\theta T_\theta^\top \, d\theta$$

$$\tilde{M}_{v_i} := \sum_{v_j \sim v_i} w_{ij} \kappa_{ij} T_{ij} T_{ij}^\top$$

Divided difference approximation



$$\tilde{M}_{v_i} := \sum_{v_j \sim v_i} w_{ij} \kappa_{ij} T_{ij} T_{ij}^\top$$

The volume and surface area descriptor

3D counterparts of the invariants $A_r(\mathbf{p})$, $CA_r(\mathbf{p})$ are the *volume descriptor* $V_r(\mathbf{p})$ and the *surface area descriptor* $SA_r(\mathbf{p})$ of a point $\mathbf{p}$ on the boundary surface of a domain D:

$$V_r(\mathbf{p}) = \int_{\mathbf{p}+rB} 1_D(\mathbf{x})\, d\mathbf{x}, \tag{18}$$

$$SA_r(\mathbf{p}) = \int_{\mathbf{p}+rS} 1_D(\mathbf{x})\, d\mathbf{x} = \frac{dV_r(\mathbf{p})}{dr}. \tag{19}$$

The relation of these quantities to mean curvature is discussed in [Hulin and Troyanov 2003] and [Pottmann et al. 2007]:

$$V_r = \frac{2\pi}{3}r^3 - \frac{\pi H}{4}r^4 + O(r^5), \quad SA_r = 2\pi r^2 - \pi H r^3 + O(r^4). \tag{20}$$

•