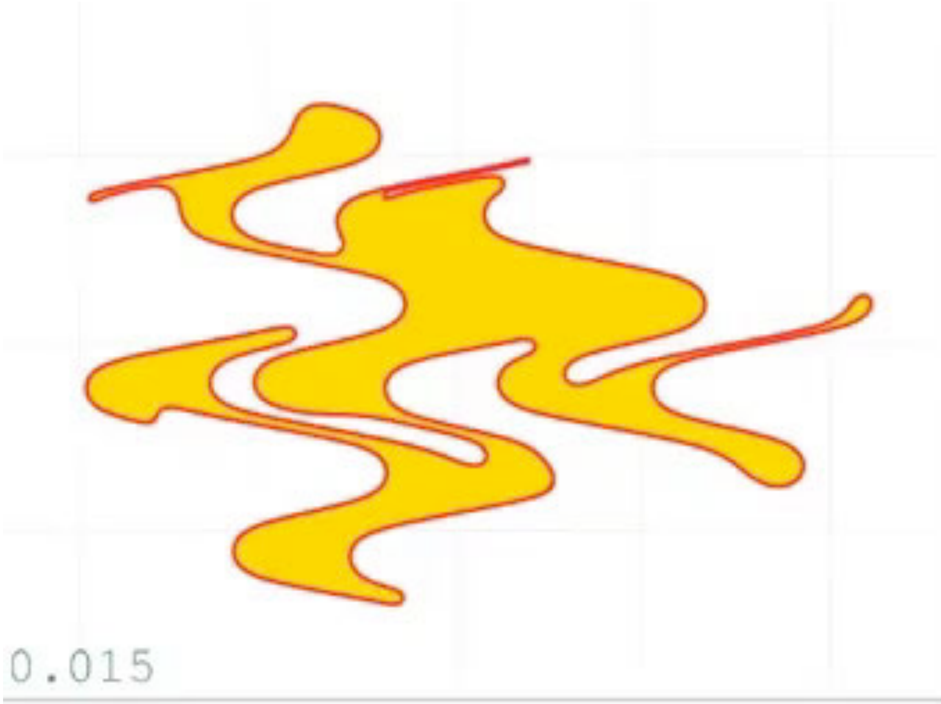
Length Shortening Flow

- The objective for length shortening flow is simply the total length of the curve; the flow is then the (L^2) gradient flow.

$$\text{length}(\gamma) := \int_0^L |\tfrac{d}{ds}\gamma|\, ds$$

$$\tfrac{d}{dt}\gamma = -\nabla_\gamma \text{length}(\gamma)$$

- For closed curves, several interesting features (Gage-Grayson-Hamilton):
  - Center of mass is preserved
  - Curves flow to “round points”
  - Embedded curves remain embedded

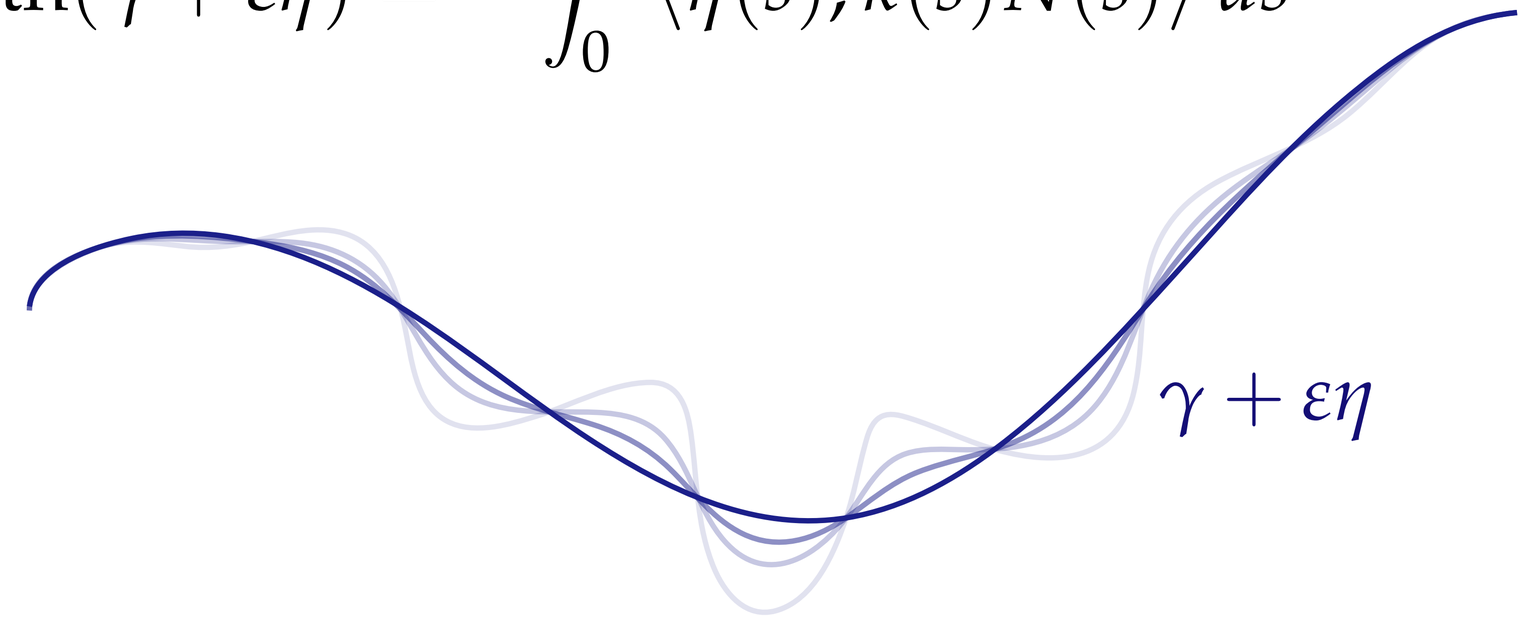


credit: Sigurd Angenent

Length Shortening Flow

Let $\text{length}(\gamma)$ denote the total length of a regular plane curve $\gamma : [0, L] \to \mathbb{R}^2$, and consider a variation $\eta : [0, L] \to \mathbb{R}^2$ vanishing at endpoints. One can then show that

$$\frac{d}{d\varepsilon}\Big|_{\varepsilon=0} \text{length}(\gamma + \varepsilon\eta) = -\int_0^L \langle \eta(s), \kappa(s)N(s) \rangle \, ds$$



Key idea: quickest way to reduce length is to move in the direction κN.

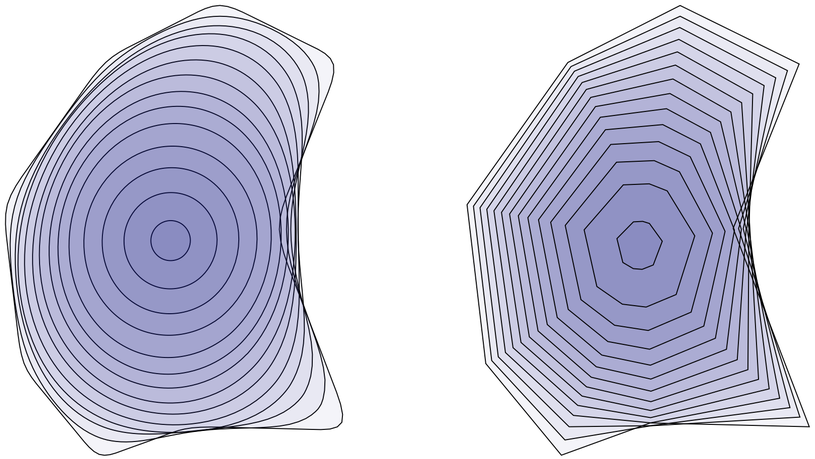
Length Shortening Flow—Forward Euler

- At each moment in time, move curve in normal direction with speed proportional to curvature
- "Smooths out" curve (*e.g.*, noise), eventually becoming circular
- Discretize by replacing time derivative with difference in time; smooth curvature with one (of many) curvatures
- Repeatedly add a little bit of κN (*"forward Euler method"*)

$$\frac{d}{dt}\gamma(s,t) = -\kappa(s,t)N(s,t)$$

$$\frac{\gamma_i^{t+1} - \gamma_i^t}{\tau} = -\kappa_i^t N_i^t$$

$$\Rightarrow \gamma_i^{t+1} = \gamma_i^t - \tau\kappa_i^t N_i^t$$



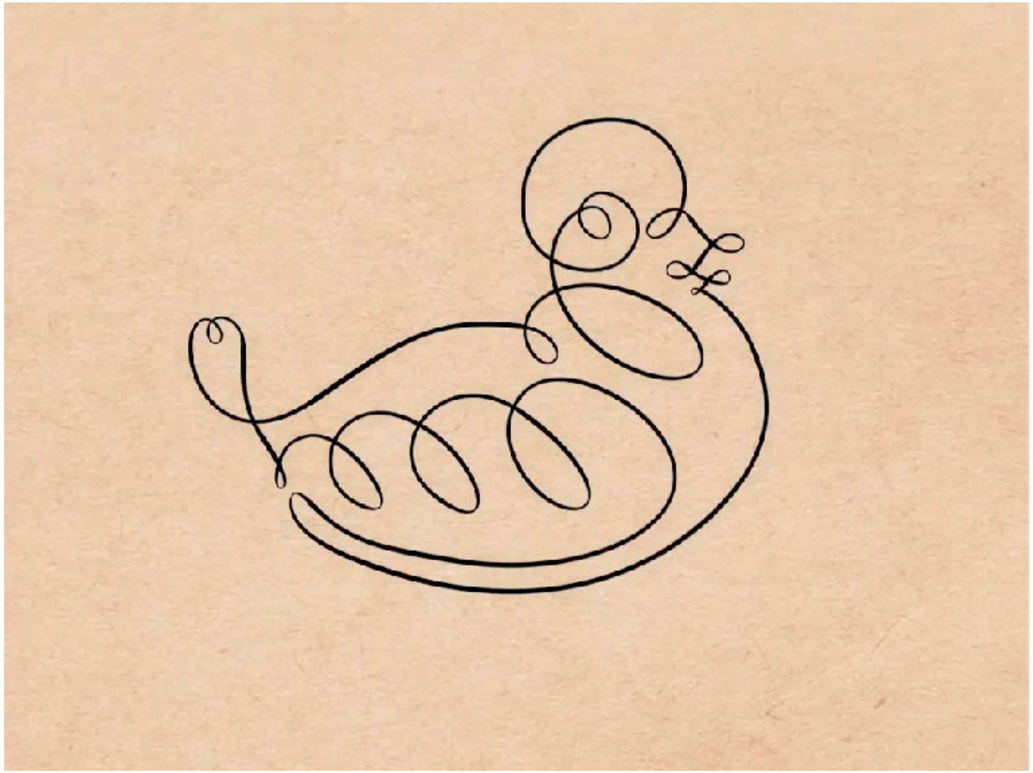
smooth **discrete**

Elastic Flow

- Basic idea: rather than shrinking length, try to reduce *bending* (curvature)
- Objective is integral of squared curvature; elastic flow is then gradient flow on this objective
- Minimizers are called *elastic curves*
- More interesting w/ constraints (e.g., endpoint positions & a tangents)

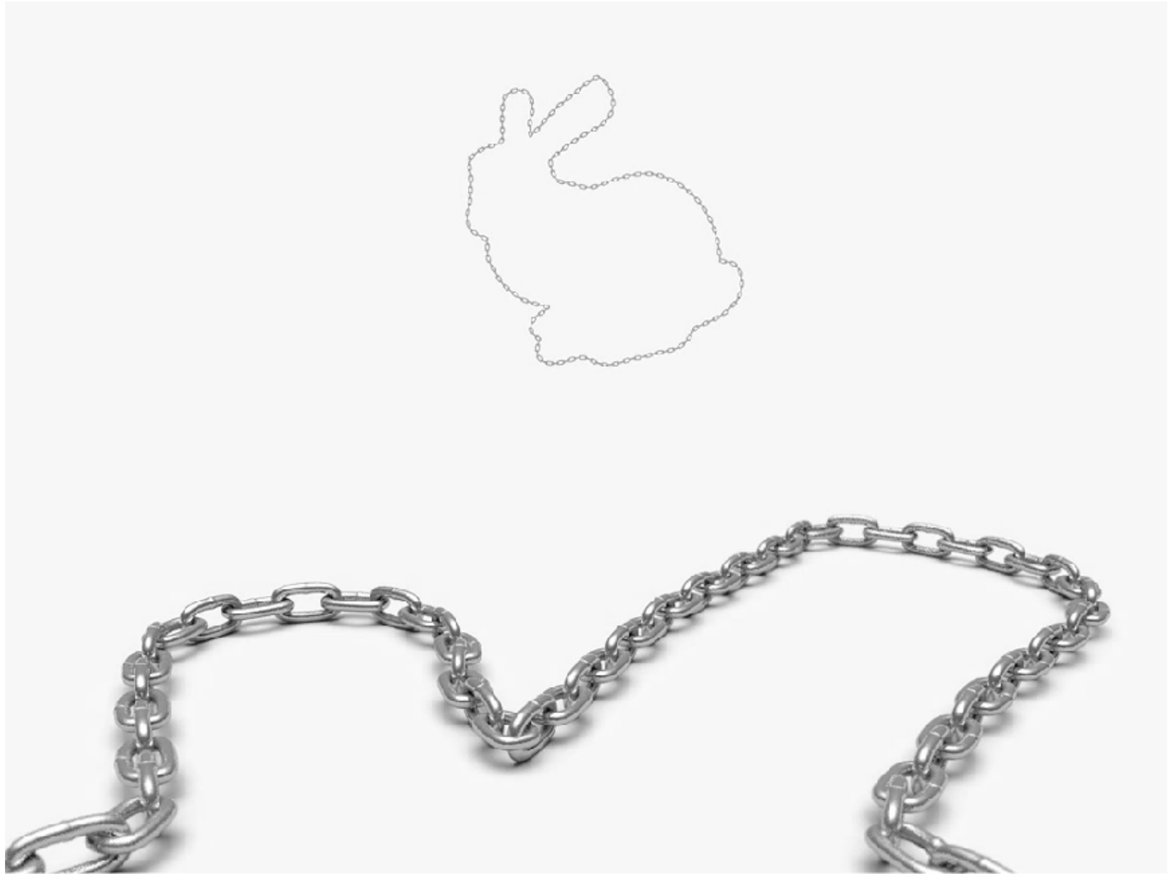
$$E(\gamma) := \int_0^L \kappa(s)^2 \, ds$$

$$\frac{d}{dt}\gamma = -\nabla_\gamma E(\gamma)$$



Isometric Elastic Flow

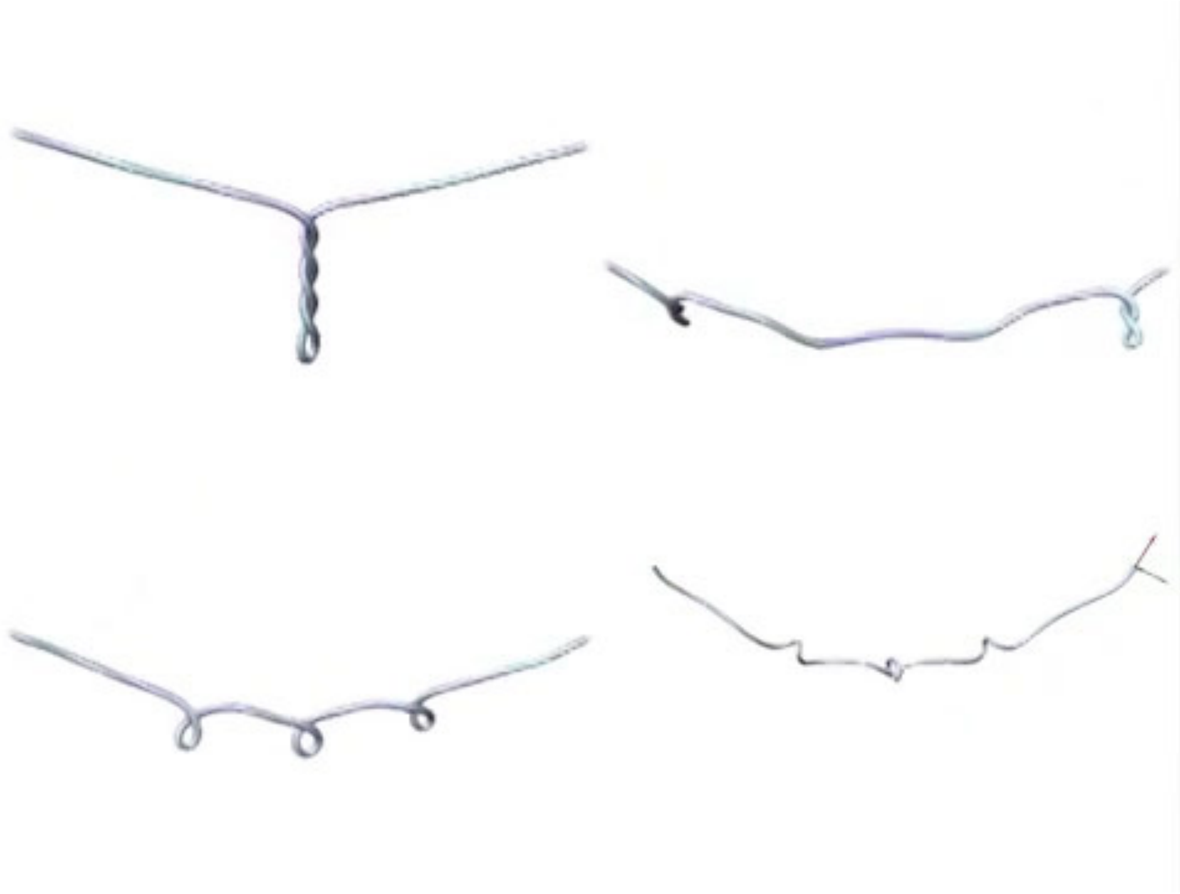
- Different way to smooth out a curve is to directly "shrink" curvature
- Discrete case: "scale down" turning angles, then use the fundamental theorem of discrete plane curves to reconstruct
- Extremely stable numerically; exactly preserves edge lengths
- Challenge: how do we make sure closed curves remain closed?



From Crane et al, *"Robust Fairing via Conformal Curvature Flow"*

Elastic Rods

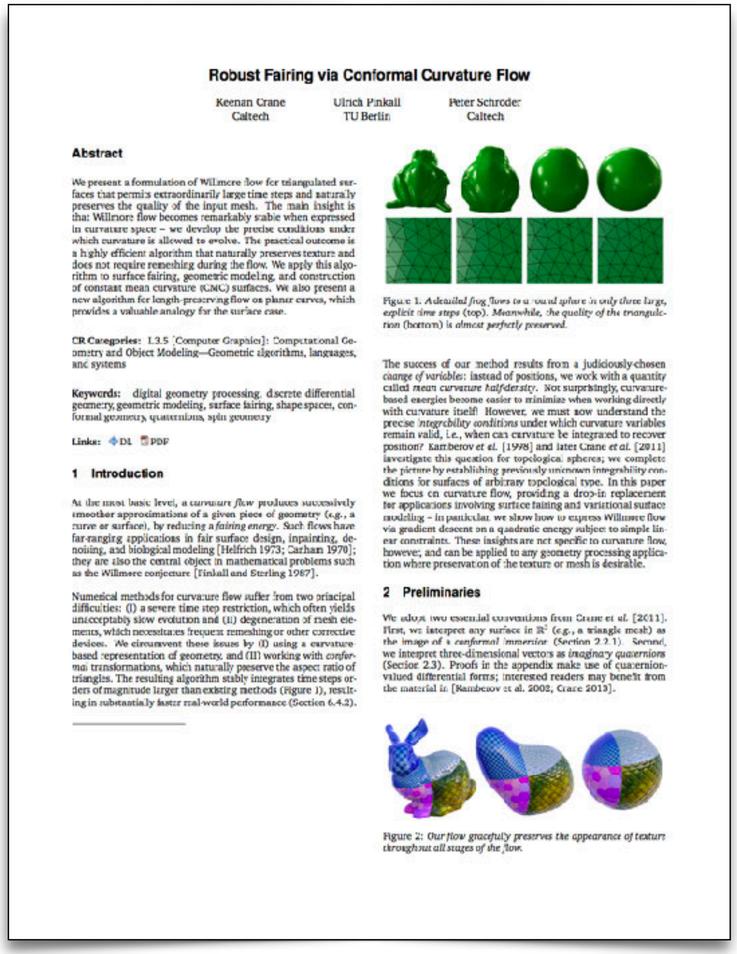
- For space curve, can also try to minimize both *curvature* **and** *torsion*
- Both in some sense measure "non-straightness" of curve
- Provides rich model of *elastic rods*
- Lots of interesting applications (simulating hair, laying cable, …)



From Bergou et al, *"Discrete Elastic Rods"*

Reading Assignment

- Readings from papers on curve algorithms (will be posted online)



Robust Fairing via Conformal Curvature Flow

Keenan Crane, Caltech · Ulrich Pinkall, TU Berlin · Peter Schröder, Caltech

Abstract

We present a formulation of Willmore flow for triangulated surfaces that permits extraordinarily large time steps and naturally preserves the quality of the input mesh. The main insight is that Willmore flow becomes remarkably stable when expressed in curvature space – we develop the precise conditions under which curvature is allowed to evolve. The practical outcome is a highly efficient algorithm that naturally preserves texture and does not require remeshing during the flow. We apply this algorithm to surface fairing, geometric modeling, and construction of constant mean curvature (CMC) surfaces. We also present a new algorithm for length-preserving flow on planar curves, which provides a valuable analogy for the surface case.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Geometric algorithms, languages, and systems

Keywords: digital geometry processing, discrete differential geometry, geometric modeling, surface fairing, shape spaces, conformal geometry, quaternions, spin geometry

Links: DL PDF

1 Introduction

At the most basic level, a *curvature flow* produces successively smoother approximations of a given piece of geometry (e.g., a curve or surface), by reducing a *fairing energy*. Such flows have far-ranging applications in fair surface design, inpainting, denoising, and biological modeling [Helfrich 1973; Canham 1970]; they are also the central object in mathematical problems such as the Willmore conjecture [Pinkall and Sterling 1987].

Numerical methods for curvature flow suffer from two principal difficulties: (I) a severe time step restriction, which often yields unacceptably slow evolution and (II) degeneration of mesh elements, which necessitates frequent remeshing or other corrective devices. We circumvent these issues by (I) using a curvature-based representation of geometry, and (II) working with *conformal* transformations, which naturally preserve the aspect ratio of triangles. The resulting algorithm stably integrates time steps orders of magnitude larger than existing methods (figure 1), resulting in substantially faster real-world performance (section 6.4.2).

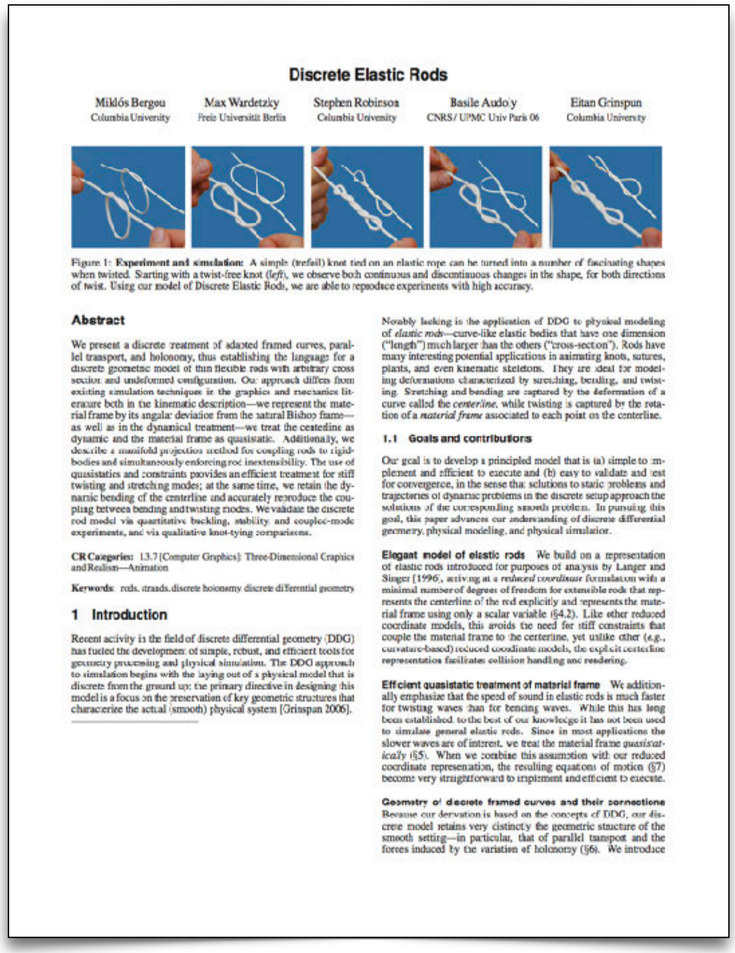
Figure 1: *A detailed frog flows to a round sphere in only three large, explicit time steps (top). Meanwhile, the quality of the triangulation (bottom) is almost perfectly preserved.*

The success of our method results from a judiciously-chosen *change of variables*: instead of positions, we work with a quantity called *mean curvature half-density*. Not surprisingly, curvature-based energies become easier to minimize when working directly with curvature itself. However, we must now understand the precise *integrability conditions* under which curvature variables remain valid, i.e., when can curvature be integrated to recover position? Kamberov *et al.* [1998] and later Crane *et al.* [2011] investigate this question for topological spheres; we complete the picture by establishing previously unknown integrability conditions for surfaces of arbitrary topological type. In this paper we focus on curvature flow, providing a drop-in replacement for applications involving surface fairing and variational surface modeling – in particular, we show how to express Willmore flow via gradient descent on a quadratic energy subject to simple linear constraints. These insights are not specific to curvature flow, however, and can be applied to any geometry processing application where preservation of the texture or mesh is desirable.

2 Preliminaries

We adopt two essential conventions from Crane et al. [2011]. First, we interpret any surface in $\mathbb{R}^3$ (e.g., a triangle mesh) as the *image of a conformal immersion* (section 2.2.1). Second, we interpret three-dimensional vectors as *imaginary quaternions* (section 2.3). Proofs in the appendix make use of quaternion-valued differential forms; interested readers may benefit from the material in [Kamberov et al. 2002, Crane 2013].

Figure 2: *Our flow gracefully preserves the appearance of texture throughout all stages of the flow.*



Discrete Elastic Rods

Miklós Bergou, Columbia University · Max Wardetzky, Freie Universität Berlin · Stephen Robinson, Columbia University · Basile Audoly, CNRS / UPMC Univ Paris 06 · Eitan Grinspun, Columbia University

Figure 1: **Experiment and simulation:** *A simple (trefoil) knot tied on an elastic rope can be turned into a number of fascinating shapes when twisted. Starting with a twist-free knot (left), we observe both continuous and discontinuous changes in the shape, for both directions of twist. Using our model of Discrete Elastic Rods, we are able to reproduce experiments with high accuracy.*

Abstract

We present a discrete treatment of adapted framed curves, parallel transport, and holonomy, thus establishing the language for a discrete geometric model of thin flexible rods with arbitrary cross section and undeformed configuration. Our approach differs from existing simulation techniques in the graphics and mechanics literature both in the kinematic description—we represent the material frame by its angular deviation from the natural Bishop frame—as well as in the dynamical treatment—we treat the centerline as dynamic and the material frame as quasistatic. Additionally, we describe a *multigrid projection method* for coupling rods to rigid-bodies and simultaneously enforcing rod inextensibility. The use of quasistatics and constraints provides an efficient treatment for stiff twisting and stretching modes; at the same time, we retain the dynamic bending of the centerline and accurately reproduce the coupling between bending and twisting modes. We validate the discrete rod model via quantitative buckling, stability and coupled-mode experiments, and via qualitative knot-tying comparisons.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

Keywords: rods, strands, discrete holonomy, discrete differential geometry

1 Introduction

Recent activity in the field of discrete differential geometry (DDG) has fueled the development of simple, robust, and efficient tools for geometry processing and physical simulation. The DDG approach to simulation begins with the laying out of a physical model that is discrete from the ground up; the primary directive in designing this model is a focus on the preservation of key geometric structures that characterize the actual (smooth) physical system [Grinspun 2006].

Notably lacking is the application of DDG to physical modeling of *elastic rods*—curve-like elastic bodies that have one dimension ("length") much larger than the others ("cross-section"). Rods have many interesting potential applications in animating knots, sutures, plants, and even kinematic skeletons. They are ideal for modeling deformations characterized by stretching, bending, and twisting. Stretching and bending are captured by the deformation of a curve called the *centerline*, while twisting is captured by the rotation of a *material frame* associated to each point on the centerline.

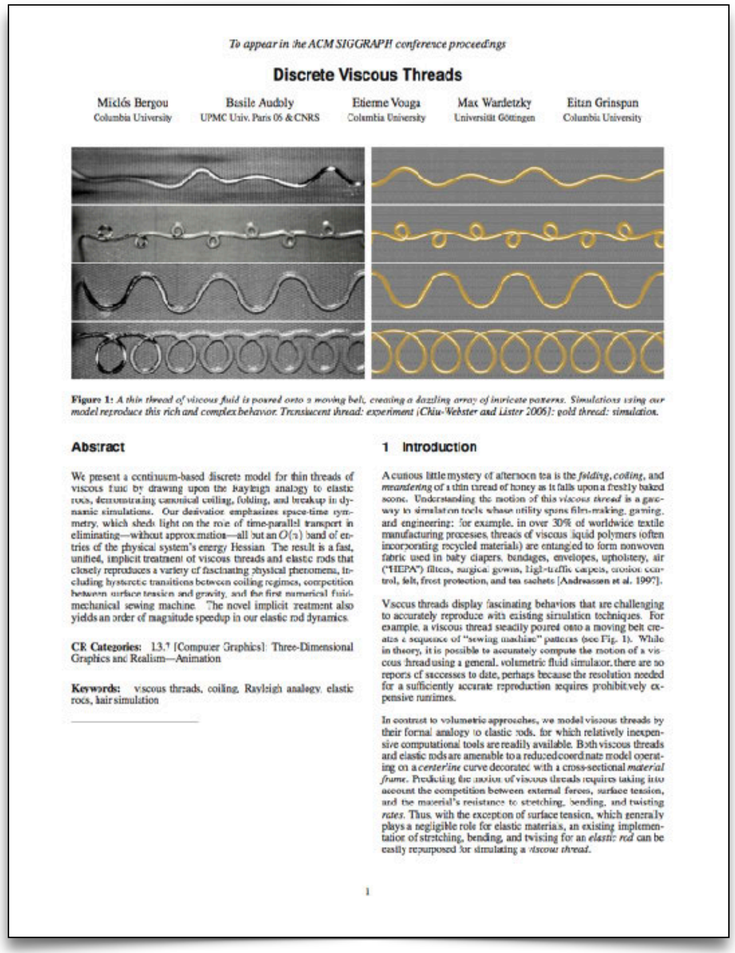
1.1 Goals and contributions

Our goal is to develop a principled model that is (a) simple to implement and efficient to execute and (b) easy to validate and test for convergence, in the sense that solutions to static problems and trajectories of dynamic problems in the discrete setup approach the solutions of the corresponding smooth problem. In pursuit of this goal, this paper advances our understanding of discrete differential geometry, physical modeling, and physical simulation.

Elegant model of elastic rods We build on a representation of elastic rods introduced for purposes of analysis by Langer and Singer [1996], arriving at a *reduced coordinate* formulation with a minimal number of degrees of freedom for extensible rods that represents the centerline of the rod explicitly and represents the material frame using only a scalar variable (§4.2). Like other reduced coordinate models, this avoids the need for stiff constraints that couple the material frame to the centerline, yet unlike other (e.g., curvature-based) reduced coordinate models, the explicit centerline representation facilitates collision handling and rendering.

Efficient quasistatic treatment of material frame We additionally emphasize that the speed of sound in elastic rods is much faster for twisting waves than for bending waves. While this has long been established to the best of our knowledge it has not been used to simulate general elastic rods. Since in most applications the slower waves are of interest, we treat the material frame *quasistatically* (§5). When we combine this assumption with our reduced coordinate representation, the resulting equations of motion (§7) become very straightforward to implement and efficient to execute.

Geometry of discrete framed curves and their connections Because our derivation is based on the concepts of DDG, our discrete model retains very distinctly the geometric structure of the smooth setting—in particular, that of parallel transport and the forces induced by the variation of holonomy (§6). We introduce



To appear in the ACM SIGGRAPH conference proceedings

Discrete Viscous Threads

Miklós Bergou, Columbia University · Basile Audoly, UPMC Univ. Paris 05 & CNRS · Etienne Vouga, Columbia University · Max Wardetzky, Universität Göttingen · Eitan Grinspun, Columbia University

Figure 1: *A thin thread of viscous fluid is poured onto a moving belt, creating a dazzling array of intricate patterns. Simulations using our model reproduce this rich and complex behavior. Translucent thread: experiment (Chiu-Webster and Lister 2006); gold thread: simulation.*

Abstract

We present a continuum-based discrete model for thin threads of viscous fluid by drawing upon the Rayleigh analogy to elastic rods, demonstrating canonical coiling, folding, and breakup in dynamic simulations. Our derivation emphasizes space-time symmetry, which sheds light on the role of time-parallel transport in eliminating—without approximation—all but an $O(n)$ band of entries of the physical system's energy Hessian. The result is a fast, unified, implicit treatment of viscous threads and elastic rods that closely reproduces a variety of fascinating physical phenomena, including hysteretic transitions between coiling regimes, competition between surface tension and gravity, and the first numerical fluid-mechanical sewing machine. The novel implicit treatment also yields an order of magnitude speedup in our elastic rod dynamics.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

Keywords: viscous threads, coiling, Rayleigh analogy, elastic rods, hair simulation

1 Introduction

A curious little mystery of afternoon tea is the *folding*, *coiling*, and *meandering* of a thin thread of honey as it falls upon a freshly baked scone. Understanding the motion of this *viscous thread* is a gateway to simulation tools whose utility spans film-making, gaming, and engineering: for example, in over 30% of worldwide textile manufacturing processes, threads of viscous liquid polymers (often incorporating recycled materials) are entangled to form nonwoven fabric used in baby diapers, bandages, envelopes, upholstery, air ("HEPA") filters, surgical gowns, light-traffic carpets, erosion control, felt, frost protection, and tea sachets [Andreassen et al. 1997].

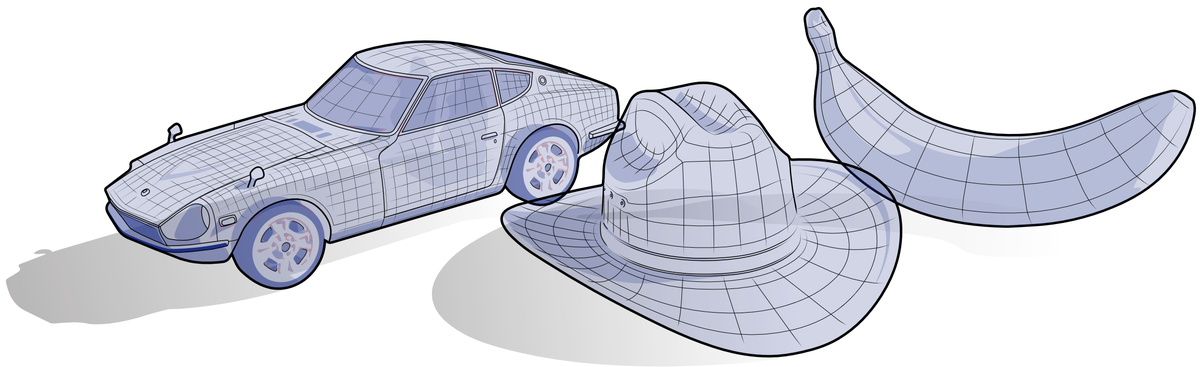
Viscous threads display fascinating behavior that are challenging to accurately reproduce with existing simulation techniques. For example, a viscous thread steadily poured onto a moving belt creates a sequence of "sewing machine" patterns (see Fig. 1). While in theory, it is possible to accurately compute the motion of a viscous thread using a general, volumetric fluid simulator, there are no reports of successes to date, perhaps because the resolution needed for a sufficiently accurate reproduction requires prohibitively expensive runtimes.

In contrast to volumetric approaches, we model viscous threads by their formal analogy to elastic rods, for which relatively inexpensive computational tools are readily available. Both viscous threads and elastic rods are amenable to a reduced coordinate model operating on a *centerline* curve decorated with a cross-sectional *material frame*. Predicting the motion of viscous threads requires taking into account the competition between external forces, surface tension, and the material's resistance to stretching, bending, and twisting *rates*. Thus, with the exception of surface tension, which generally plays a negligible role for elastic materials, an existing implementation of stretching, bending, and twisting for an *elastic rod* can be easily repurposed for simulating a *viscous thread*.

From Curves to Surfaces

- **Previously:** saw how to talk about 1D curves (both smooth and discrete)
- **Today:** will study 2D curved surfaces (both smooth and discrete)
  - Some concepts remain the same (*e.g.*, differential); others need to be generalized (*e.g.*, curvature)
  - Still use exterior calculus as our *lingua franca*

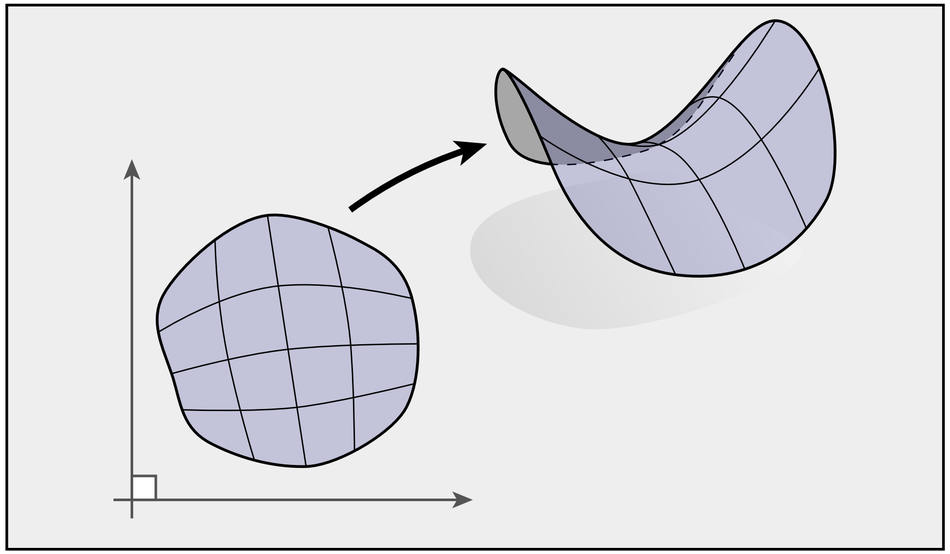
(Curves)

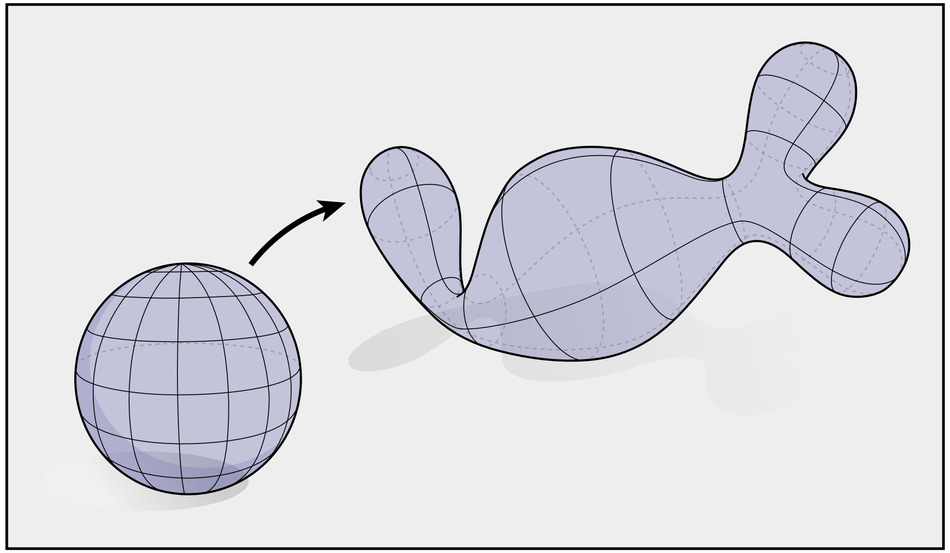


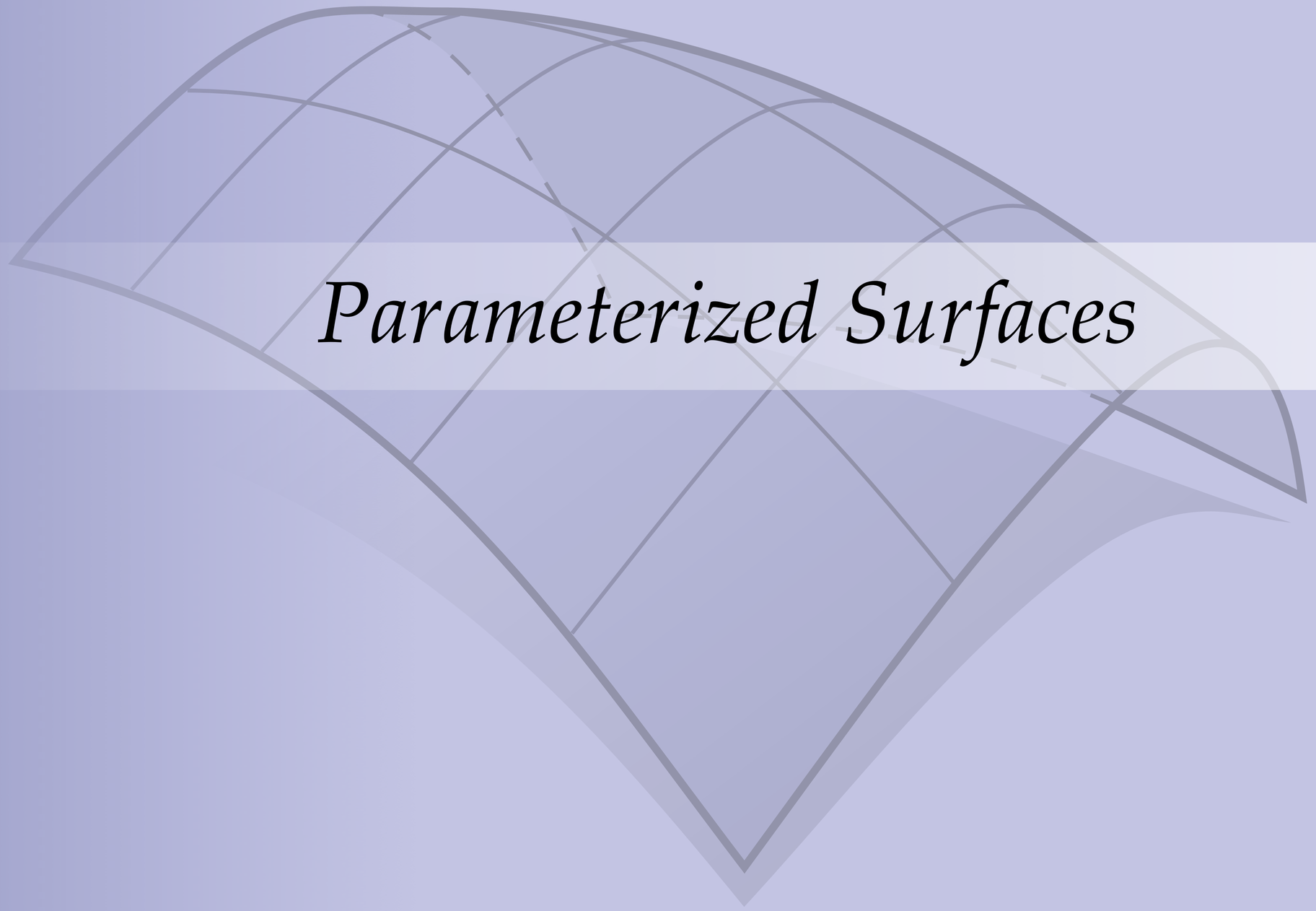
(Surfaces)

Surfaces—Local vs. Global View

- So far, we've only studied exterior calculus in R^n
- Will therefore be easiest to think of surfaces expressed in terms of patches of the plane (**local picture**)
- Later, when we study topology & smooth manifolds, we'll be able to more easily think about "whole surfaces" all at once (**global picture**)
- Global picture is *much* better model for **discrete** surfaces (meshes)…





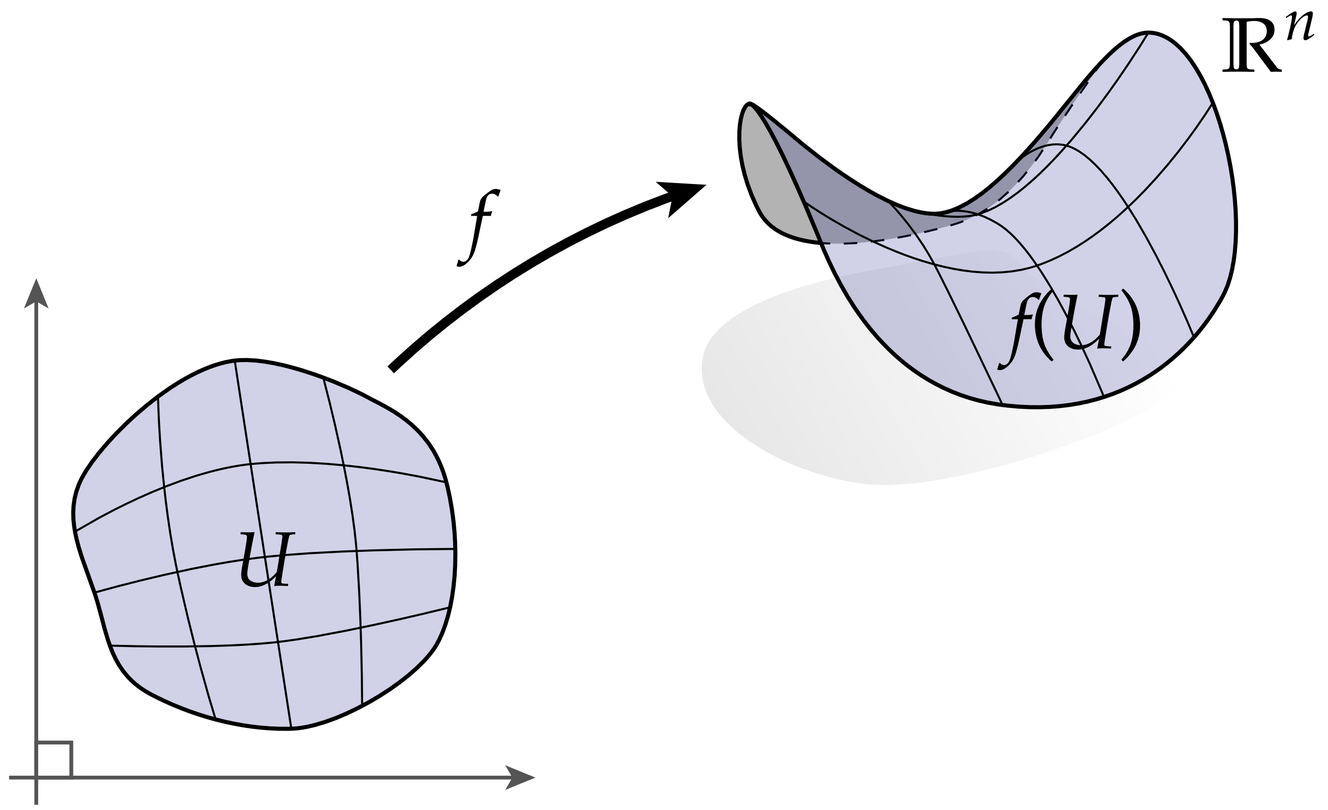


Parameterized Surfaces

Parameterized Surface

A **parameterized surface** is a map from a two-dimensional region $U \subset \mathbb{R}^2$ into $\mathbb{R}^2$:

$$f : U \to \mathbb{R}^n$$



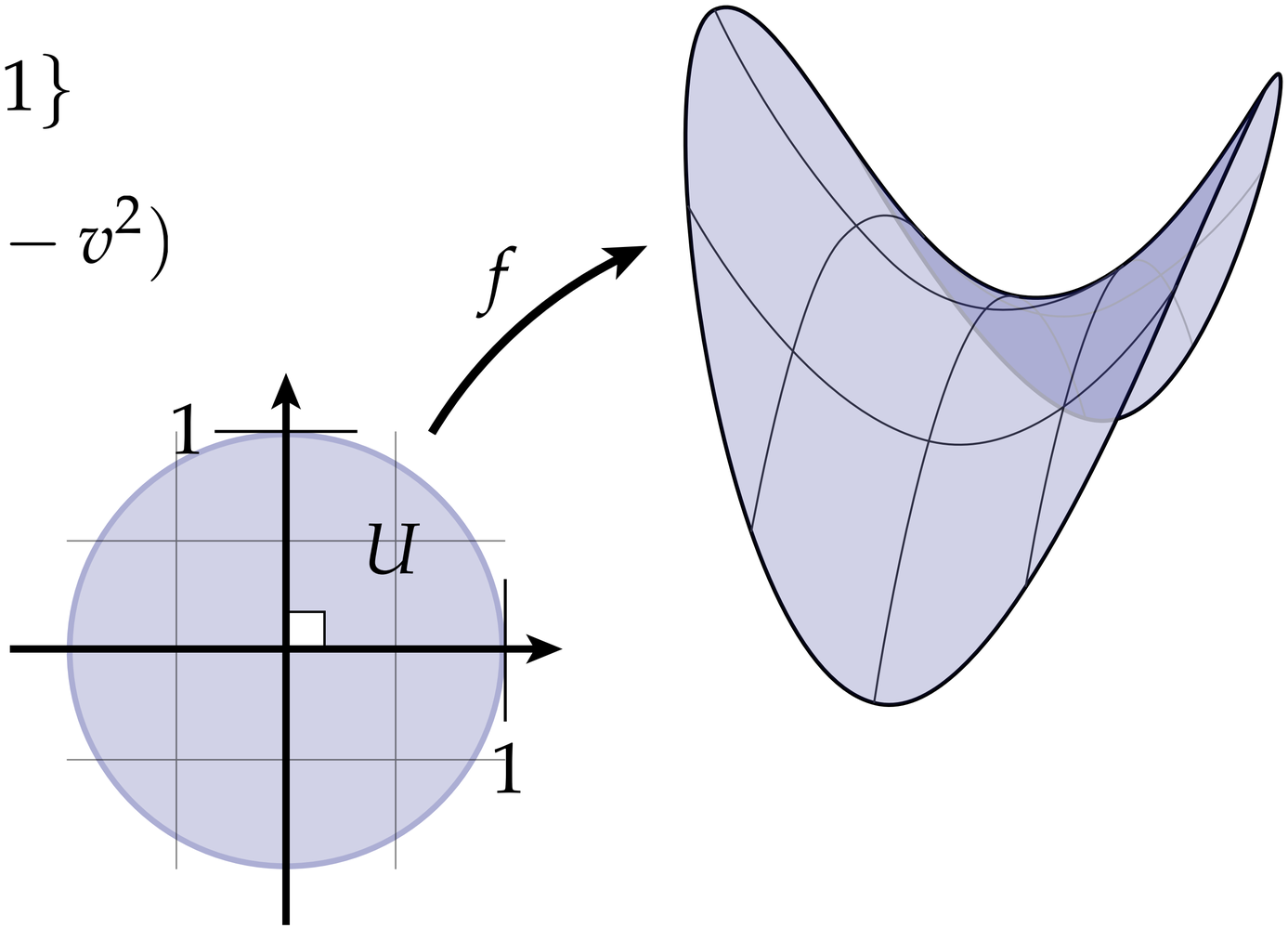
The set of points $f(U)$ is called the **image** of the parameterization.

Parameterized Surface—Example

- As an example, we can express a *saddle* as a parameterized surface:

$$U := \{(u, v) \in \mathbb{R}^2 : u^2 + v^2 \leq 1\}$$

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

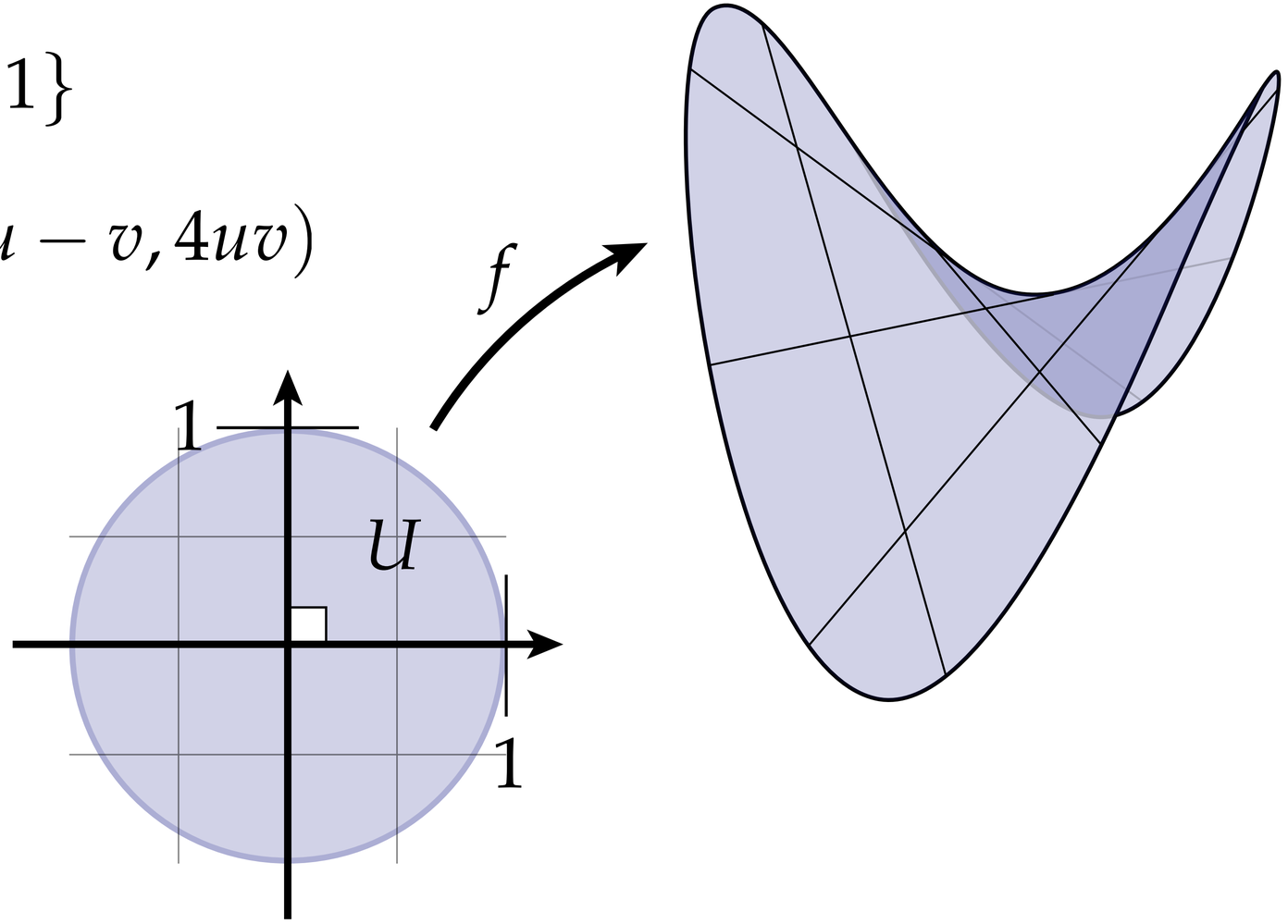


Reparameterization

- Many different parameterized surfaces can have the same image:

$$U := \{(u,v) \in \mathbb{R}^2 : u^2 + v^2 \leq 1\}$$

$$f : U \to \mathbb{R}^3;\ (u,v) \mapsto (u+v, u-v, 4uv)$$

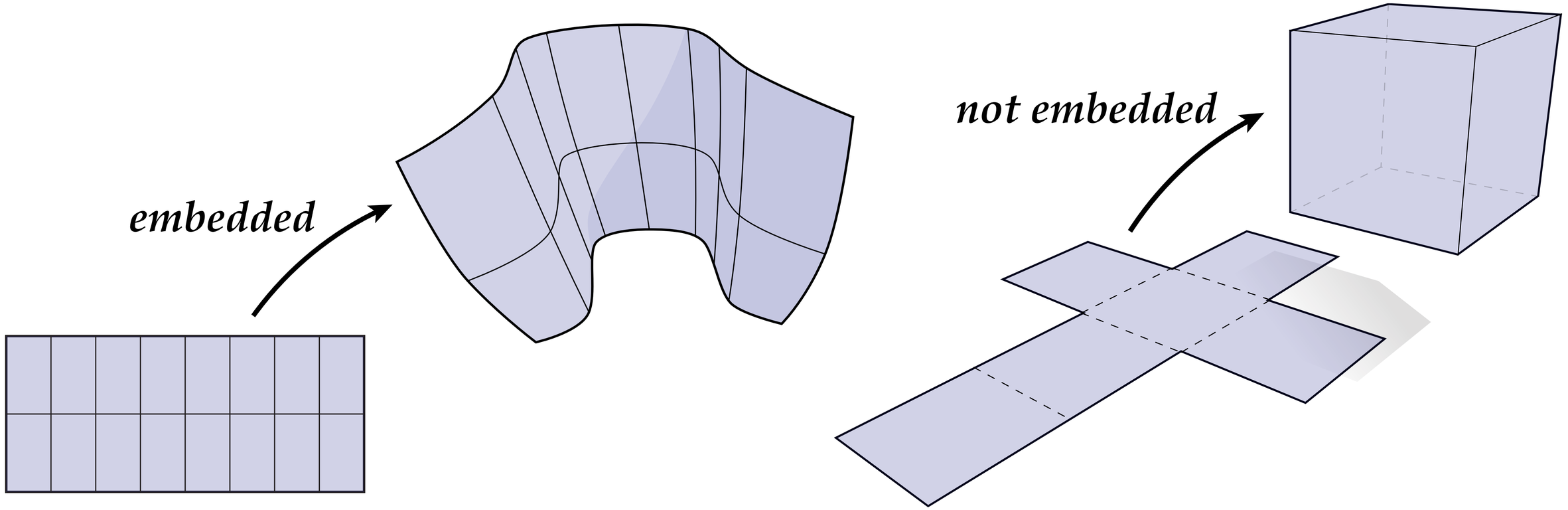


This *"reparameterization symmetry"* can be a major challenge in applications—*e.g.*, trying to decide if two parameterized surfaces (or meshes) describe the same shape.

Analogy: graph isomorphism

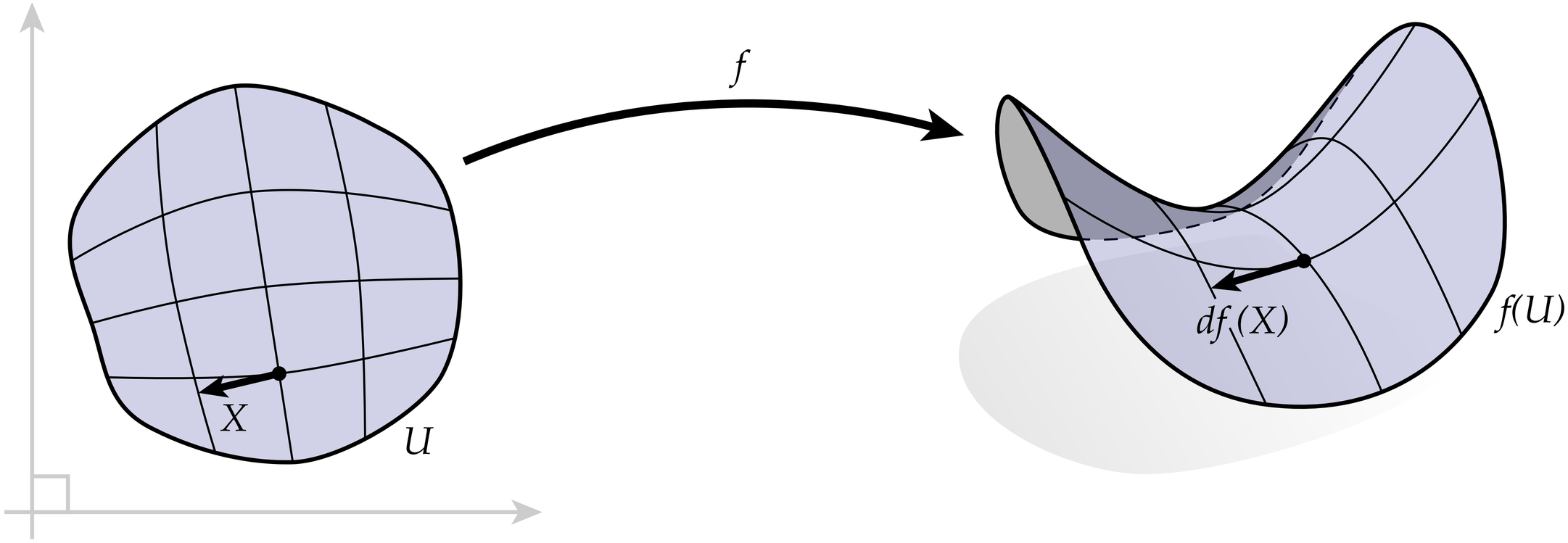
Embedded Surface

- Roughly speaking, an **embedded** surface does not self-intersect
- More precisely, a parameterized surface is an embedding if it is a continuous injective map, and has a continuous inverse on its image



Differential of a Surface

Intuitively, the *differential* of a parameterized surface tells us how tangent vectors on the domain get mapped to vectors in space:



We say that *df "pushes forward"* vectors X into R^n, yielding vectors $df(X)$

Differential in Coordinates

In coordinates, the differential is simply the exterior derivative:

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

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

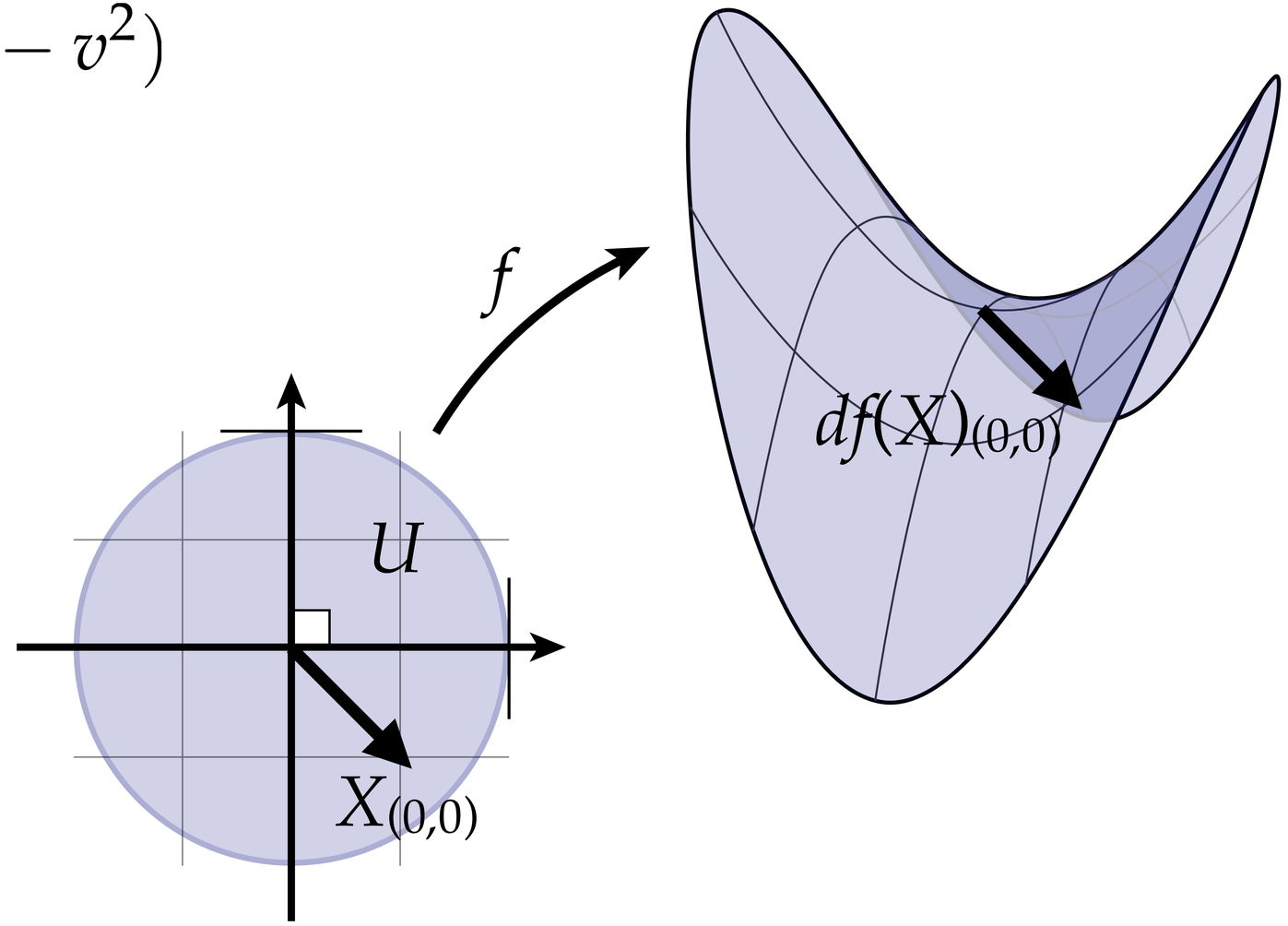
$$(1, 0, 2u)du + (0, 1, -2v)dv$$

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

$$X := \tfrac{3}{4}\left(\frac{\partial}{\partial x} - \frac{\partial}{\partial y}\right)$$

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

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



Differential—Matrix Representation (Jacobian)

Definition. Consider a map $f : \mathbb{R}^n \to \mathbb{R}^m$, and let $x_1, \ldots, x_n$ be coordinates on $\mathbb{R}^n$. Then the *Jacobian* of f is the matrix

$$J_f := \begin{bmatrix} \partial f^1/\partial x^1 & \cdots & \partial f^1/\partial x^n \\ \vdots & \ddots & \vdots \\ \partial f^m/\partial x^1 & \cdots & \partial f^m/\partial x^n \end{bmatrix},$$

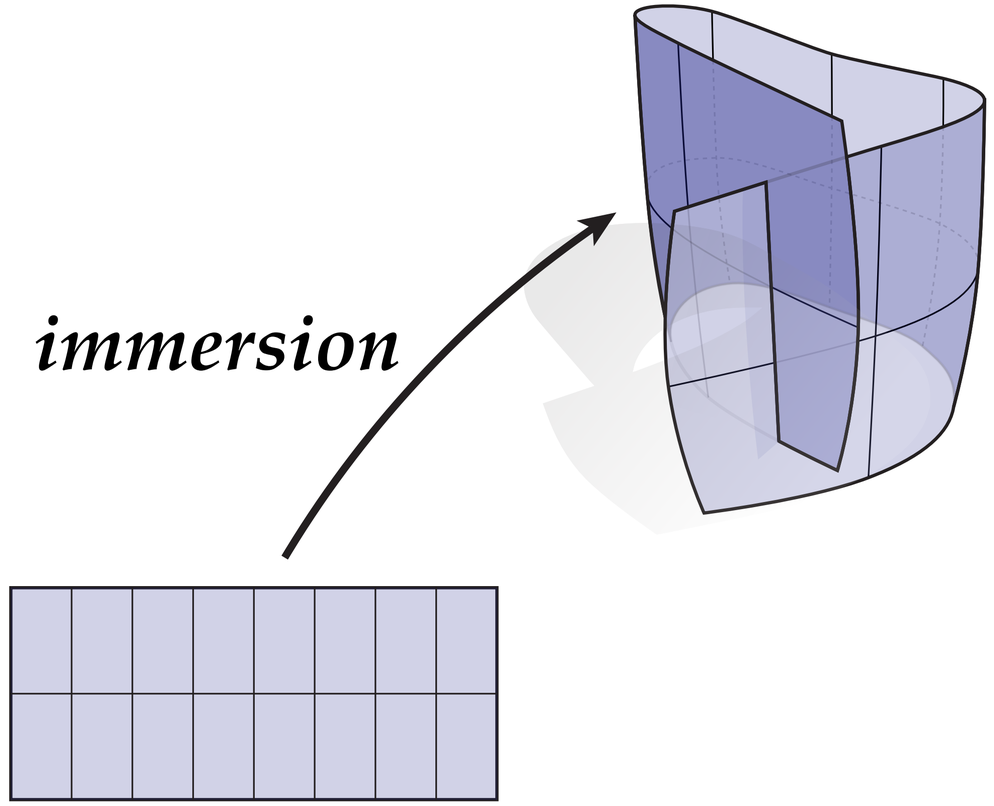
where $f^1, \ldots, f^m$ are the components of f w.r.t. some coordinate system on $\mathbb{R}^m$. This matrix represents the differential in the sense that $df(X) = J_f X$.

(In solid mechanics, also known as the *deformation gradient*.)

Note: does not generalize to infinite dimensions! (E.g., maps between functions.)

Immersed Surface

- A parameterized surface f is an *immersion* if its differential is nondegenerate, *i.e.*, if $df(X) = 0$ if and only if $X = 0$.



Intuition: no region of the surface gets “pinched”

Immersion — Example

Consider the standard parameterization of the sphere:

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

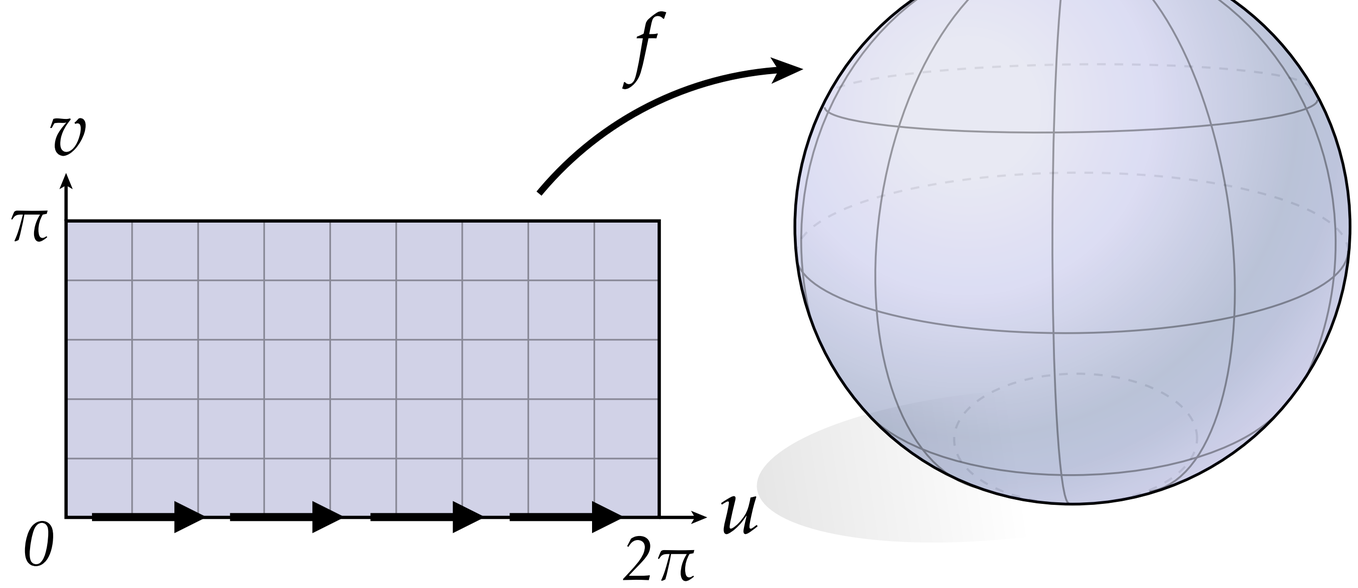
$$df = \frac{\partial f}{\partial u}du + \frac{\partial f}{\partial v}dv = \begin{pmatrix} -\sin(u)\sin(v), & \cos(u)\sin(v), & 0 \end{pmatrix} du + \begin{pmatrix} \cos(u)\cos(v), & \cos(v)\sin(u), & -\sin(v) \end{pmatrix} dv$$

Q: Is f an immersion?

A: No: when $v = 0$ we get

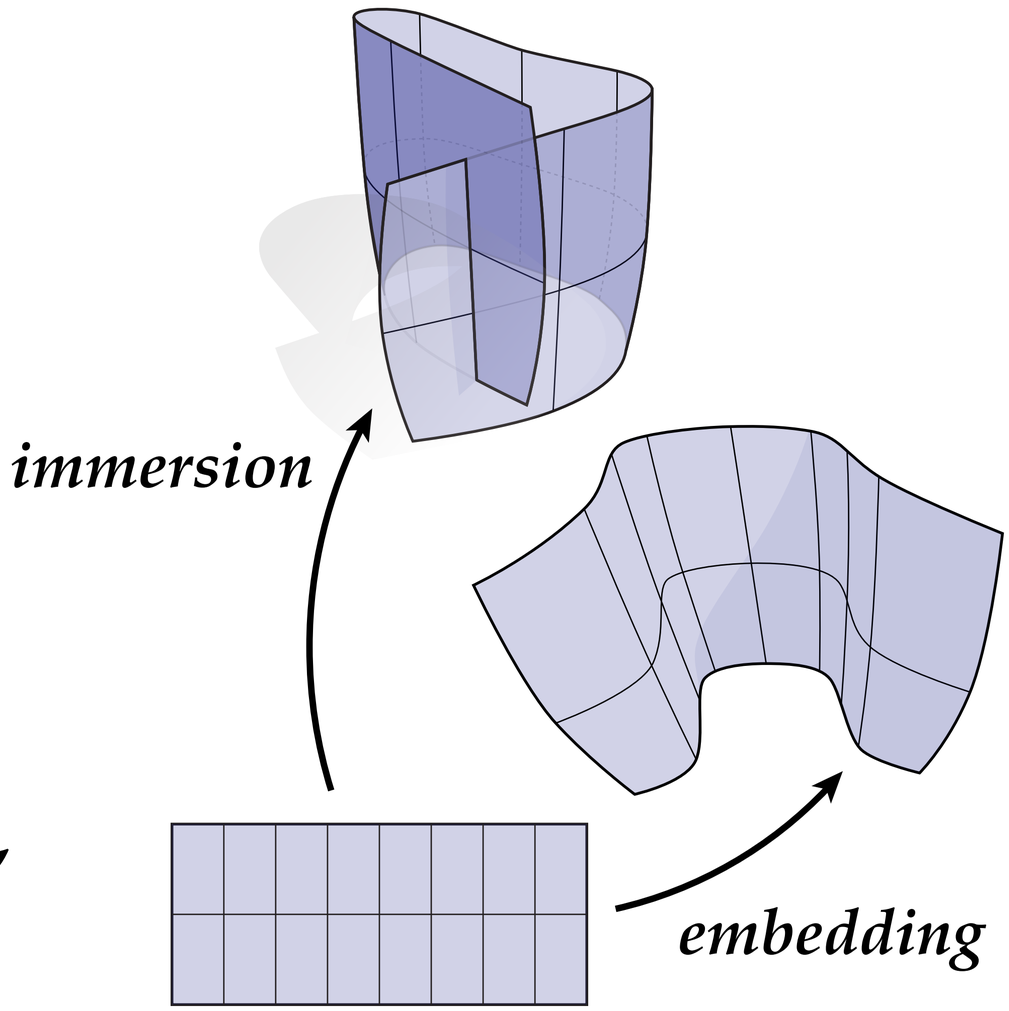
$$\begin{pmatrix} 0, & 0, & 0 \end{pmatrix} du + \begin{pmatrix} \cos(u), & \sin(u), & -\sin(v) \end{pmatrix} dv$$

Nonzero tangents mapped to zero!



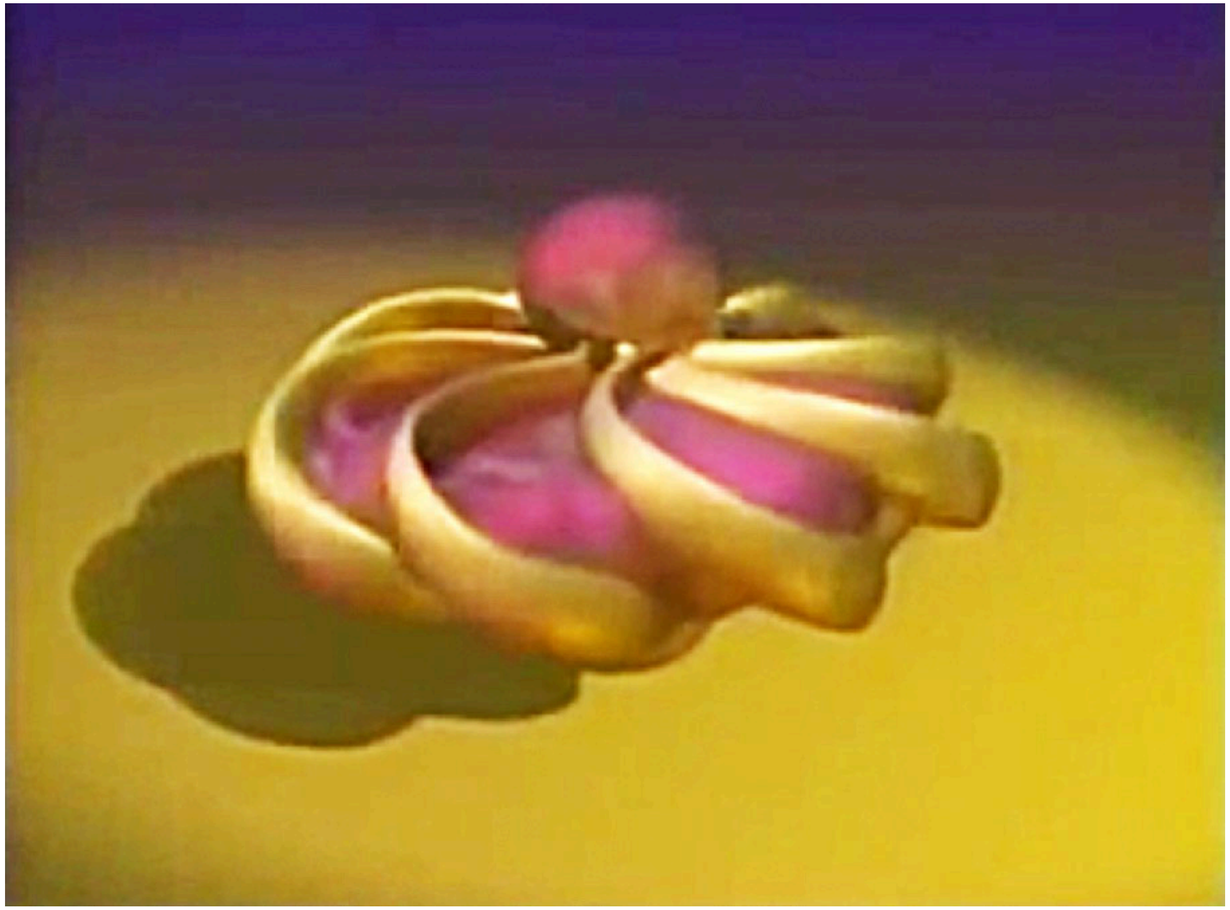
Immersion vs. Embedding

- In practice, ensuring that a surface is globally embedded can be challenging
- Immersions are typically “nice enough” to define local quantities like tangents, normals, metric, etc.
- Immersions are also a natural model for the way we typically think about meshes: most quantities (angles, lengths, etc.) are perfectly well-defined, even if there happen to be self-intersections

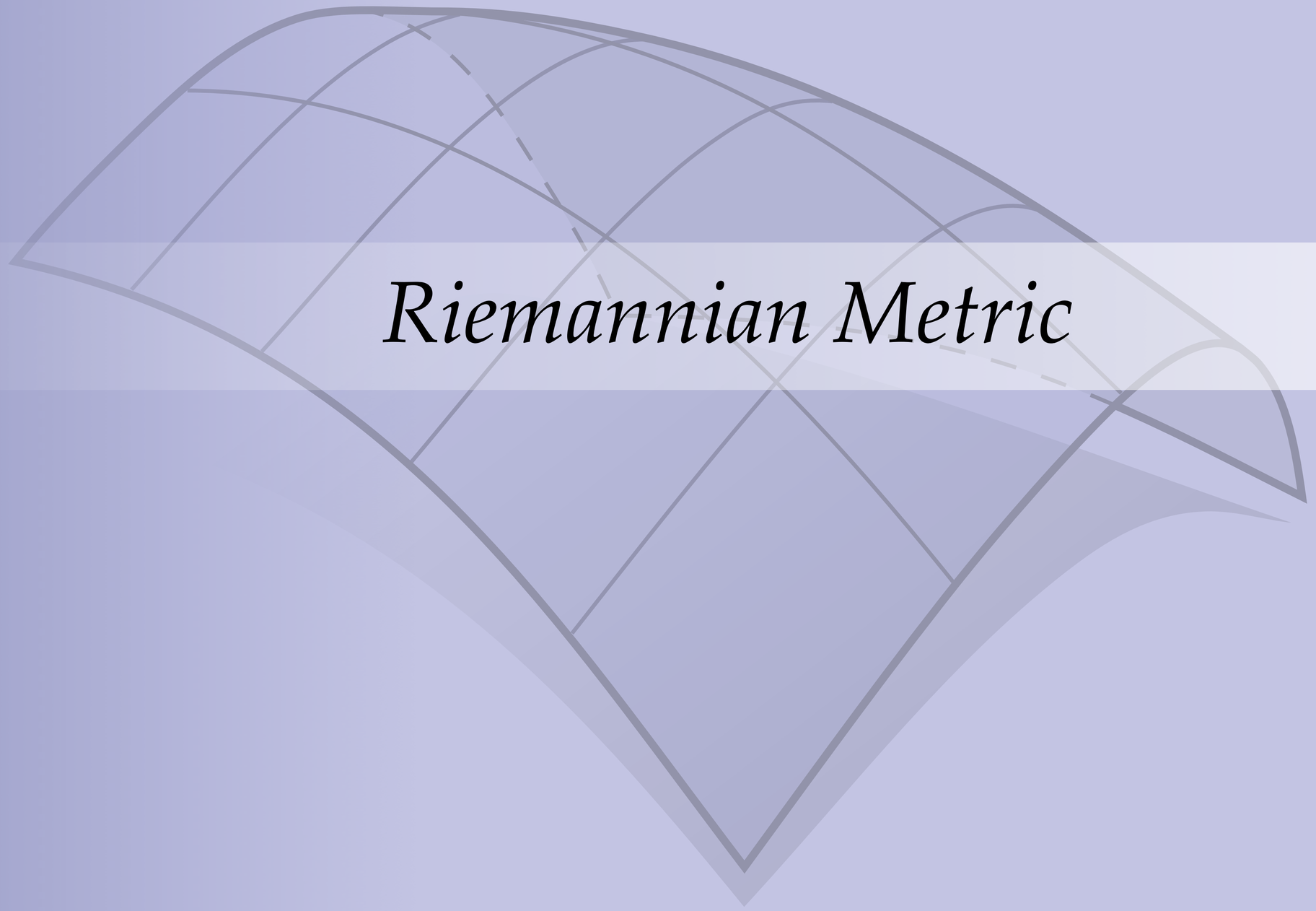


Sphere Eversion

Turning a Sphere Inside-Out (1994)



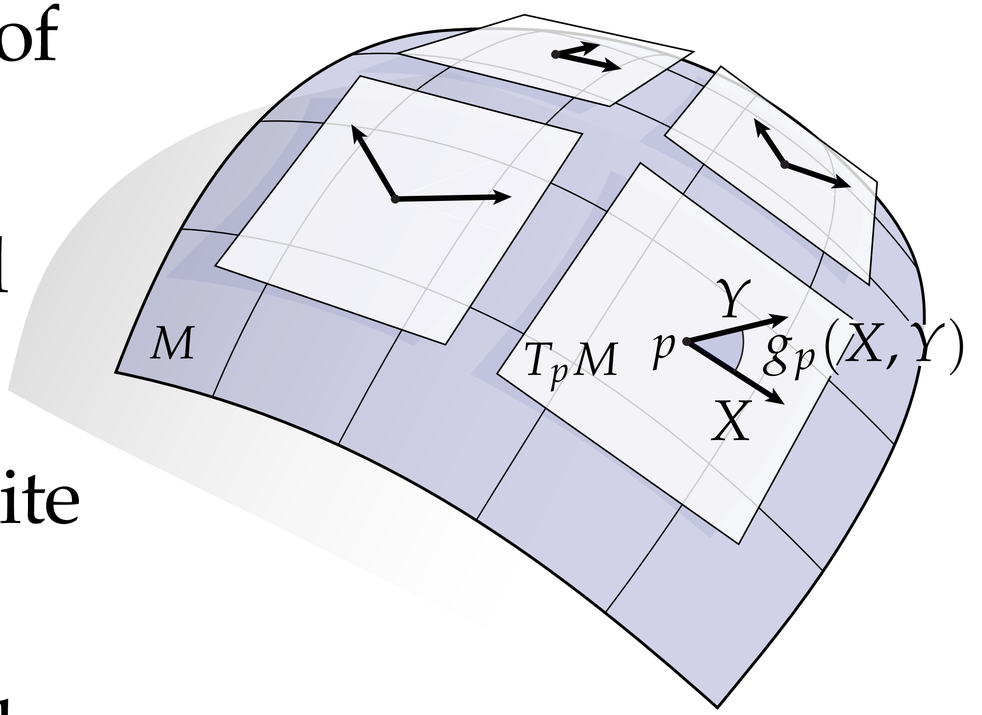
`https://youtu.be/-6g3ZcmjJ7k`



Riemannian Metric

Riemann Metric

- Many quantities on manifolds (curves, surfaces, *etc.*) ultimately boil down to measurements of *lengths* and *angles* of tangent vectors
- This information is encoded by the so-called *Riemannian metric**
- Abstractly: smoothly-varying positive-definite bilinear form
- For immersed surface, can (and will!) describe more concretely / geometrically

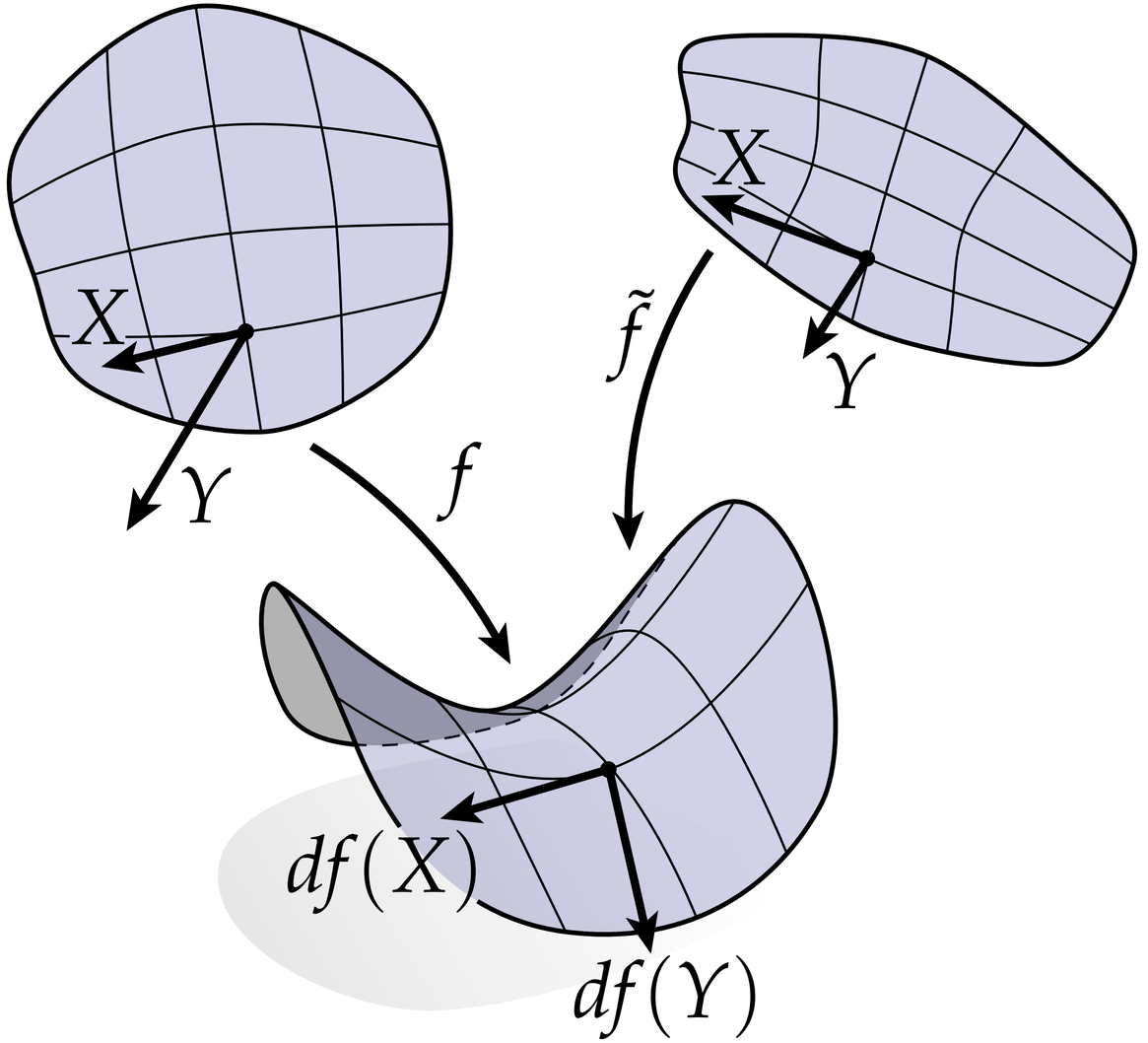


***Note:** *not* the same as a point-to-point distance metric $d(x,y)$

Metric Induced by an Immersion

- Given an immersed surface f, how should we measure inner product of vectors X, Y on its domain U?
- We should **not** use the usual inner product on the plane! (Why not?)
- Planar inner product tells us *nothing* about actual length & angle on the surface (and changes depending on choice of parameterization!)
- Instead, use **induced metric**

$$g(X,Y) := \langle df(X), df(Y) \rangle$$



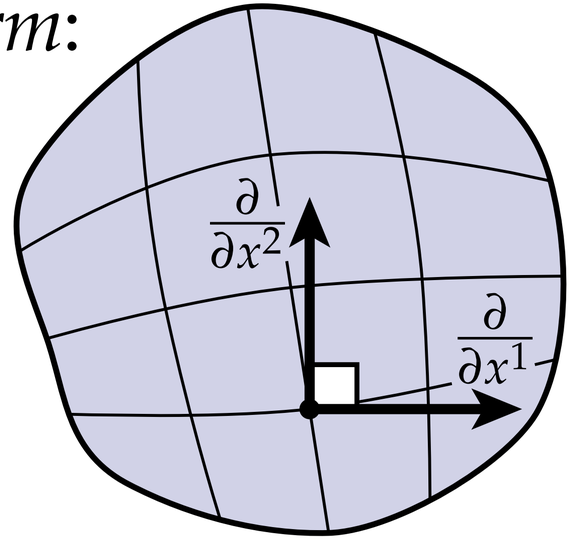
Key idea: must account for "stretching"

Induced Metric—Matrix Representation

- Metric is a bilinear map from a pair of vectors to a scalar, which we can represent as a 2x2 matrix $\mathbf{I}$ called the *first fundamental form*:

$$g(X, Y) = X^T \mathbf{I} Y$$

$$\Rightarrow \mathbf{I}_{ij} = g\left(\frac{\partial}{\partial x^i}, \frac{\partial}{\partial x^j}\right) = \left\langle df\left(\frac{\partial}{\partial x^i}\right), df\left(\frac{\partial}{\partial x^j}\right)\right\rangle$$



- Alternatively, can express first fundamental form via Jacobian:

$$g(X, Y) = \langle df(X), df(Y)\rangle = (J_f X)^{\mathsf{T}}(J_f Y) = X^{\mathsf{T}}(J_f^{\mathsf{T}} J_f)Y$$

$$\Rightarrow \mathbf{I} = J_f^{\mathsf{T}} J_f$$

Induced Metric—Example

Can use the differential to obtain the induced metric:

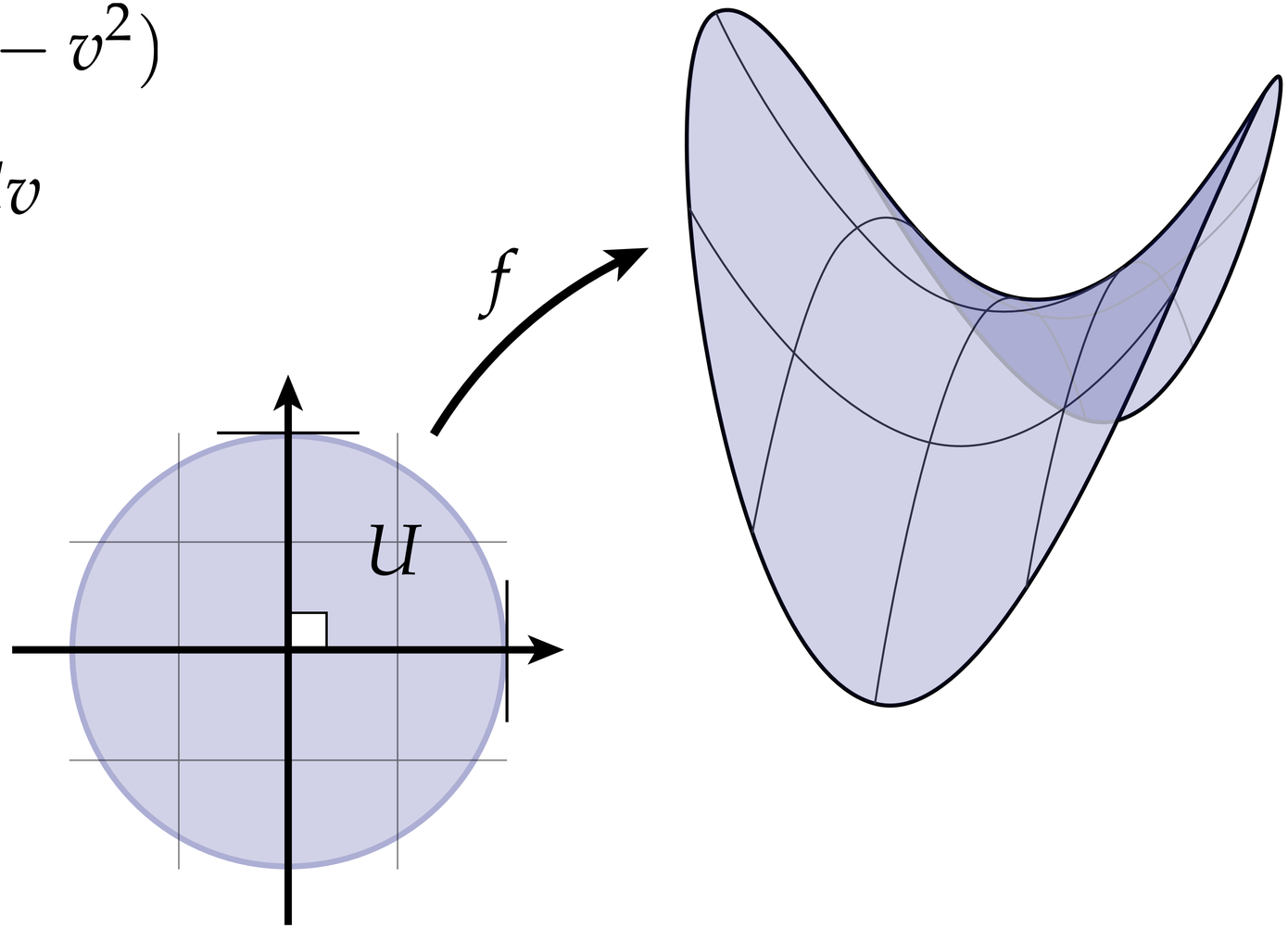
$$f : U \to \mathbb{R}^3;\ (u, v) \mapsto (u, v, u^2 - v^2)$$

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

$$J_f = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 2u & -2v \end{bmatrix}$$

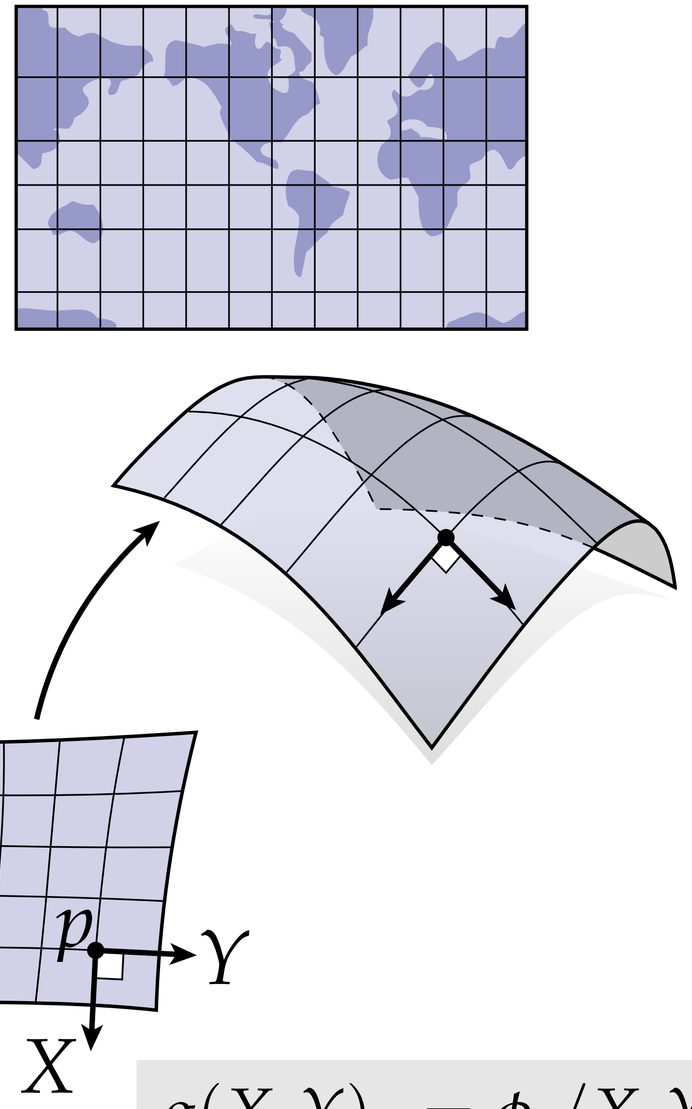
$$\mathbf{I} = J_f^{\mathsf{T}} J_f$$

$$= \begin{bmatrix} 1 + 4u^2 & -4uv \\ -4uv & 1 + 4v^2 \end{bmatrix}$$



Conformal Coordinates

- As we've just seen, there can be a complicated relationship between length & angle on the domain (2D) and the image (3D)
- For curves, we simplified life by using an *arc-length* or *isometric* parameterization: lengths on domain are identical to lengths along curve
- For surfaces, usually not possible to preserve all *lengths* (e.g., globe). Remarkably, however, can always preserve *angles* (**conformal**)
- Equivalently, a parameterized surface is *conformal* if at each point the induced metric is simply a positive rescaling of the 2D Euclidean metric

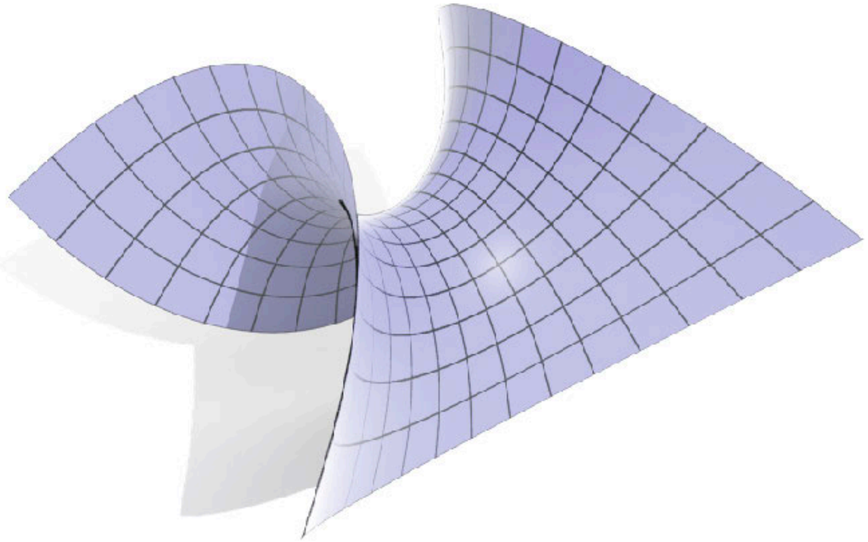


$$g(X,Y)_p = \phi_p \langle X, Y \rangle$$

Example (Enneper Surface)

Consider the surface

$$f(u,v) := \begin{bmatrix} uv^2 + u - \frac{1}{3}u^3 \\ \frac{1}{3}v\left(v^2 - 3u^2 - 3\right) \\ (u-v)(u+v) \end{bmatrix}$$



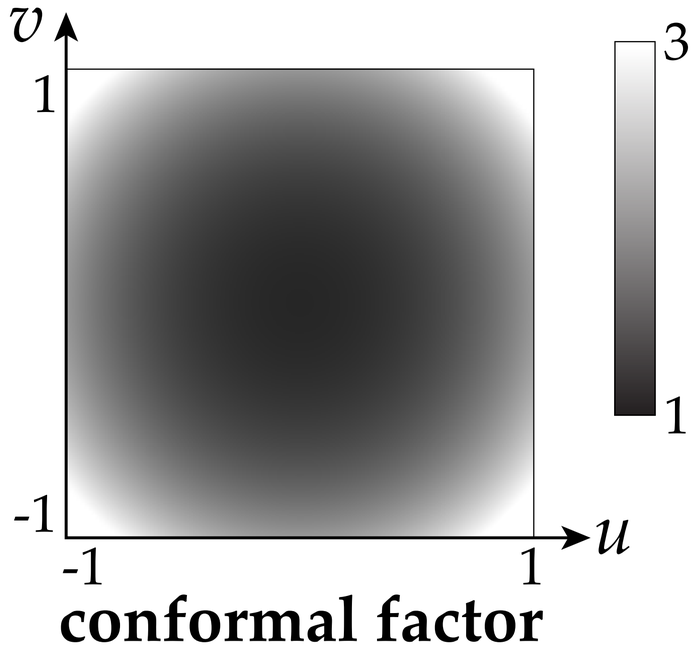
Its Jacobian matrix is

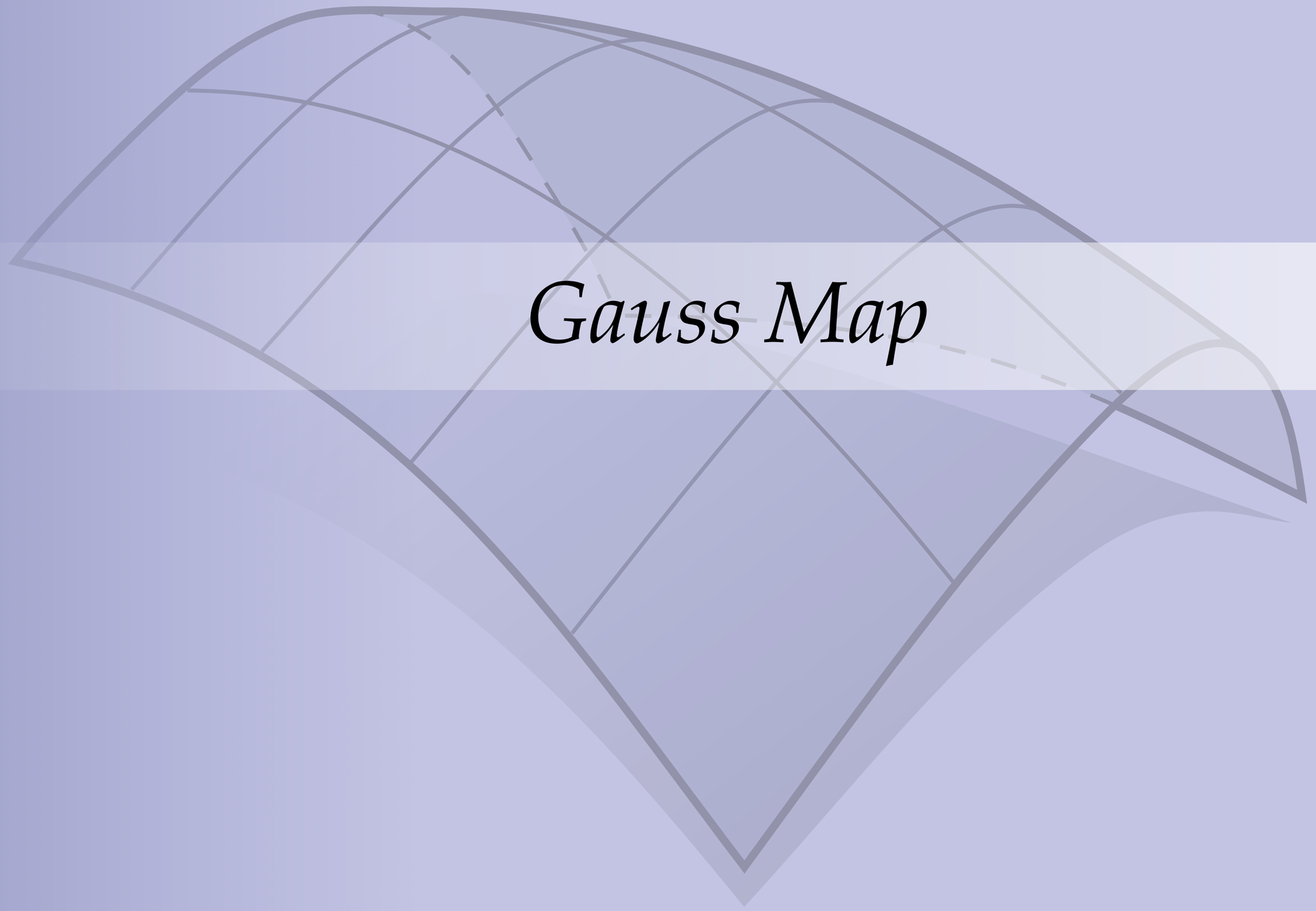
$$J_f = \begin{bmatrix} -u^2 + v^2 + 1 & 2uv \\ -2uv & -u^2 + v^2 - 1 \\ 2u & -2v \end{bmatrix}$$

Its metric then works out to be just a scalar function times the usual metric of the Euclidean plane:

$$\mathbf{I} = J_f^T J_f = \left(u^2 + v^2 + 1\right)^2 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

This function is called the *conformal scale factor*.



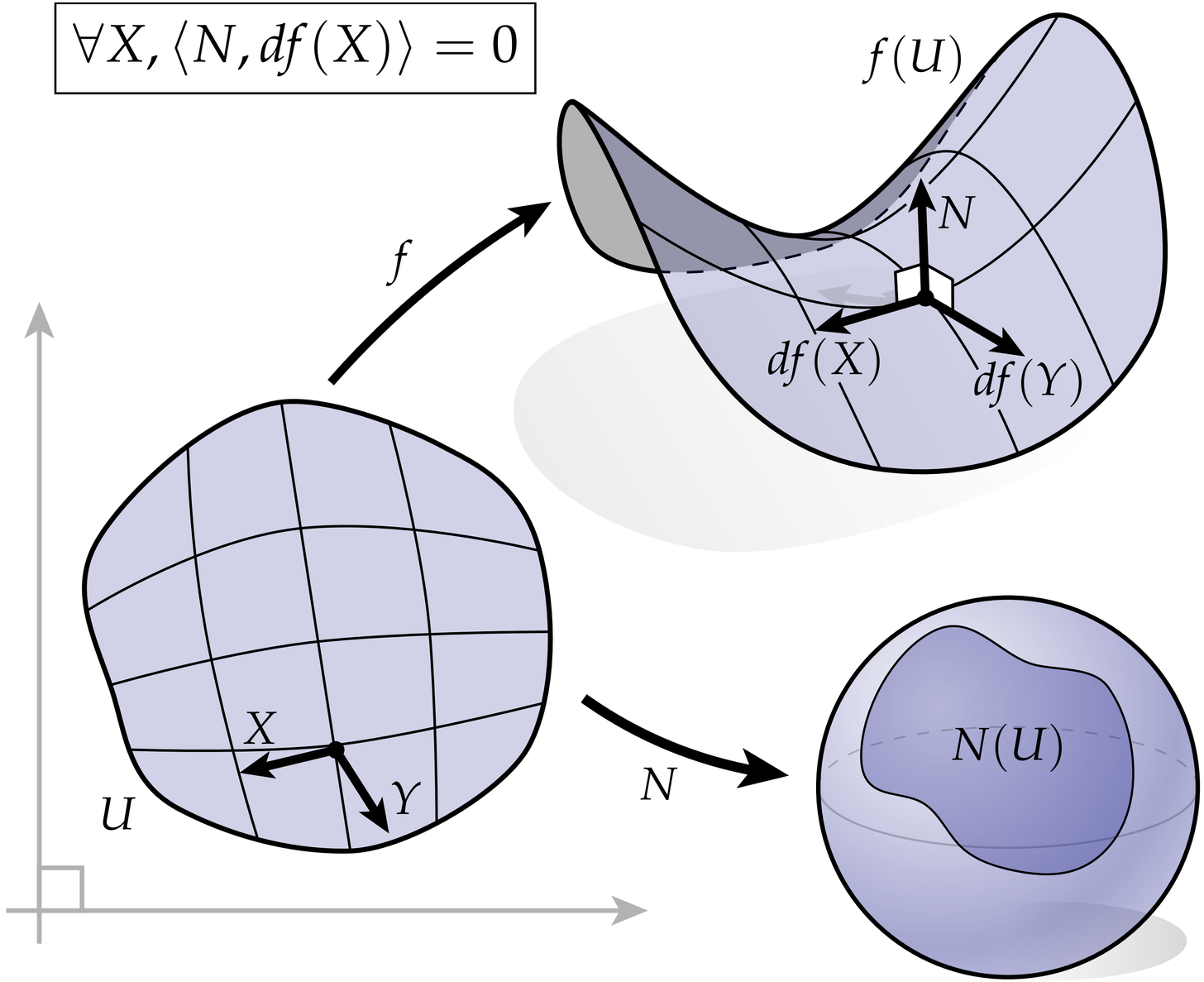


Gauss Map

Gauss Map

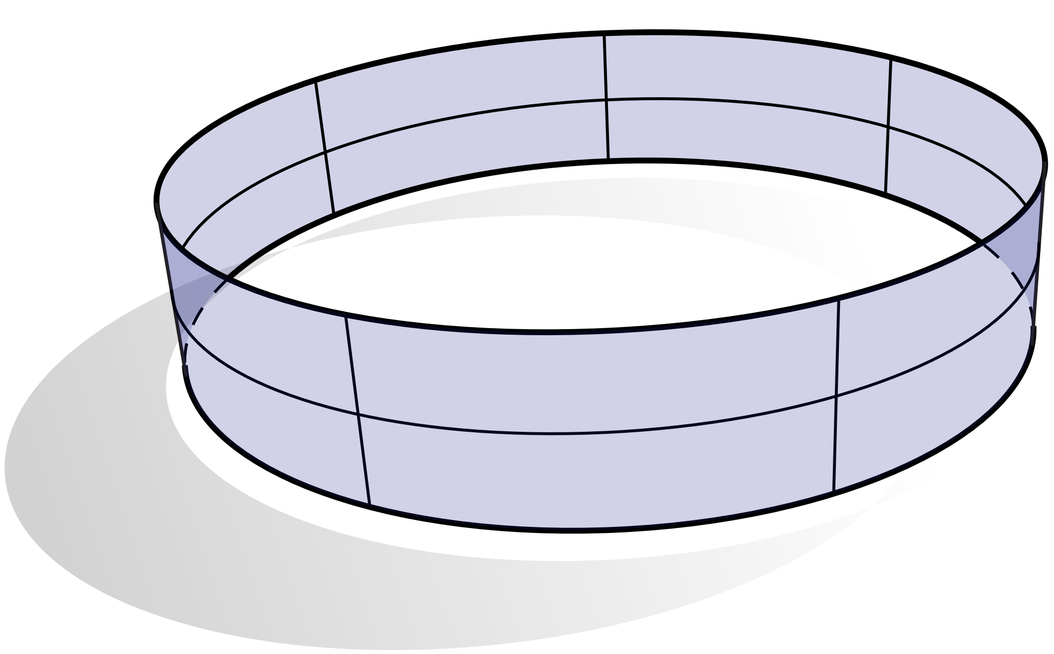
- A vector is **normal** to a surface if it is orthogonal to all tangent vectors
- **Q:** Is there a *unique* normal at a given point?
- **A:** No! Can have different magnitudes/directions.
- The **Gauss map** is a *continuous* map taking each point on the surface to a *unit* normal vector
- Can visualize Gauss map as a map from the surface to the unit sphere

$$\forall X, \langle N, df(X) \rangle = 0$$

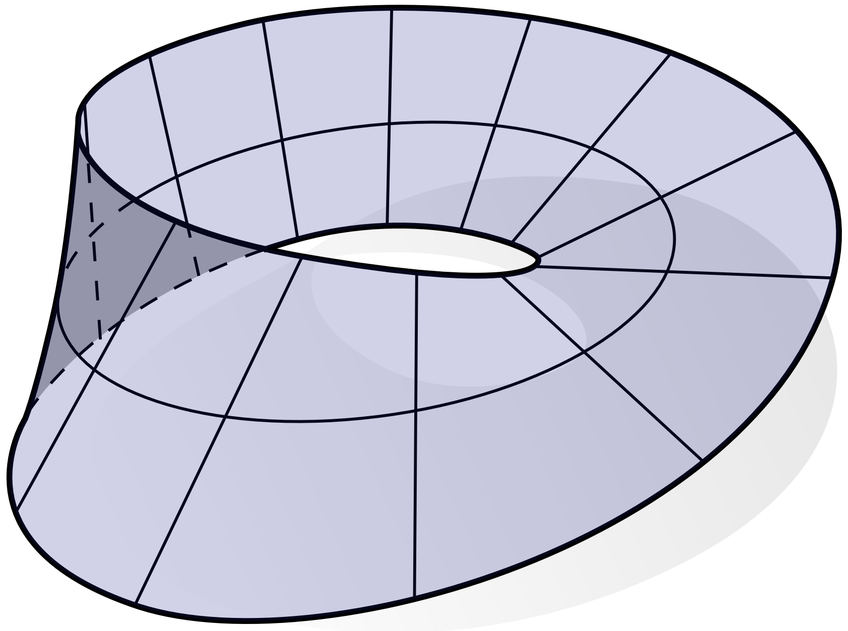


Orientability

Not every surface admits a Gauss map (globally):



orientable



nonorientable

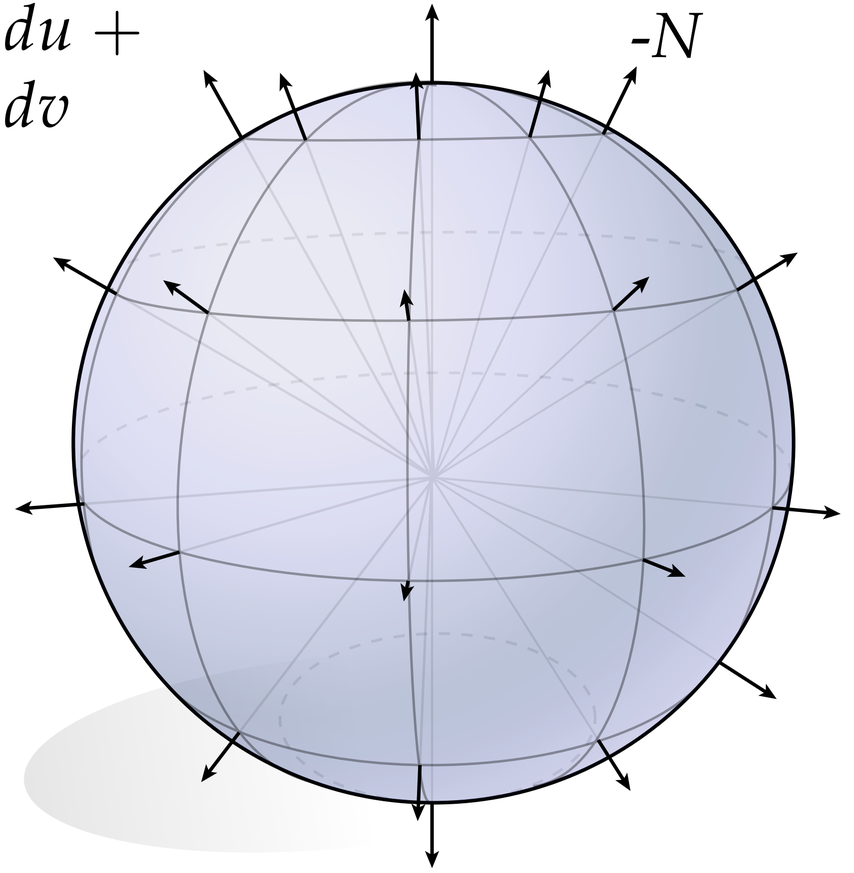
Gauss Map — Example

Can obtain unit normal by taking the cross product of two tangents*:

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

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

$$df\left(\frac{\partial}{\partial u}\right) \times df\left(\frac{\partial}{\partial v}\right) = \begin{bmatrix} -\cos(u)\sin^2(v) \\ -\sin(u)\sin^2(v) \\ -\cos(v)\sin(v) \end{bmatrix}$$



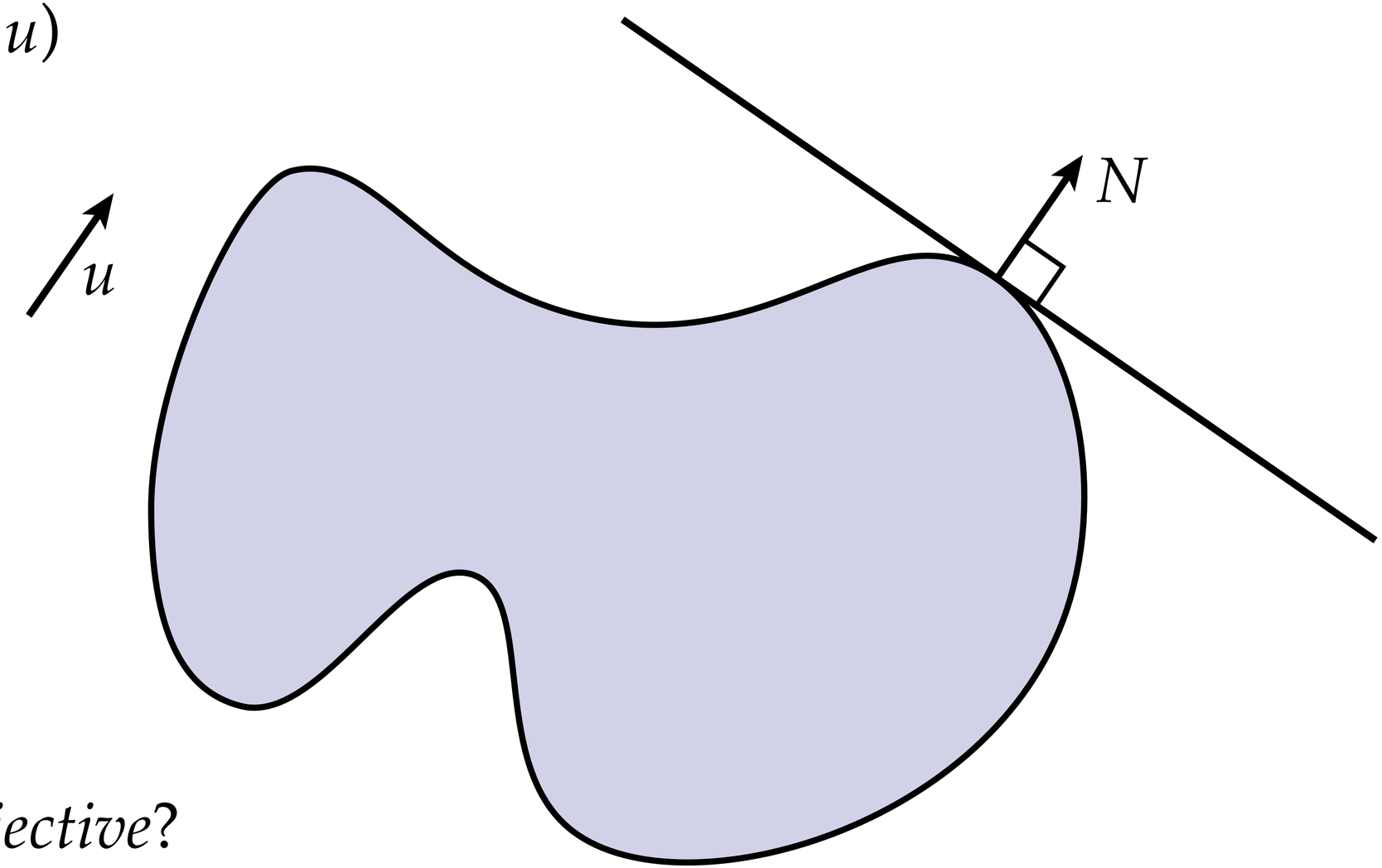
To get *unit* normal, divide by length. In this case, can just notice we have a constant multiple of the sphere itself:

$$\Rightarrow N = -f$$

*Must not be parallel!

Surjectivity of Gauss Map

- Given a unit vector u, can we always find some point on a surface that has this normal? ($N = u$)
- Yes! **Proof** (Hilbert):



Q: Is the Gauss map *injective*?

Vector Area

- Given a little patch of surface Ω, what's the "average normal"?
- Can simply integrate normal over the patch, divide by area:

$$\frac{1}{\text{area}(\Omega)} \int_\Omega N \, dA$$

- Integrand $N\,dA$ is called the **vector area**. (Vector-valued 2-form)
- Can be easily expressed via exterior calculus*:

$$\begin{aligned} df \wedge df(X,Y) &= df(X) \times df(Y) - df(Y) \times df(X) = \\ &2df(X) \times df(Y) = \\ &2N dA(X,Y) \end{aligned}$$

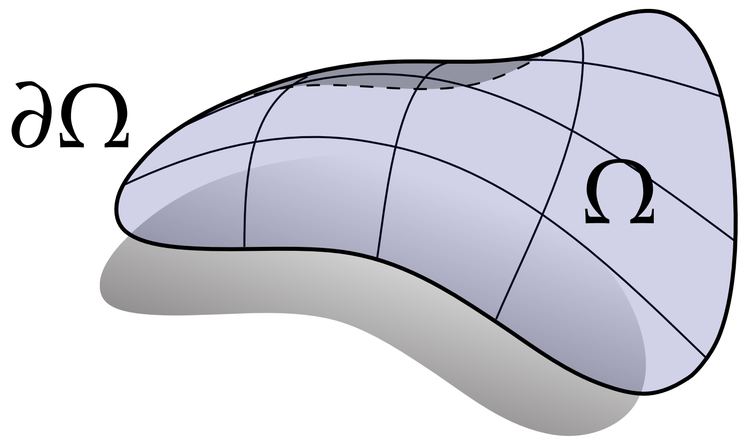
$$\implies \boxed{\mathcal{A} = \tfrac{1}{2} df \wedge df}$$

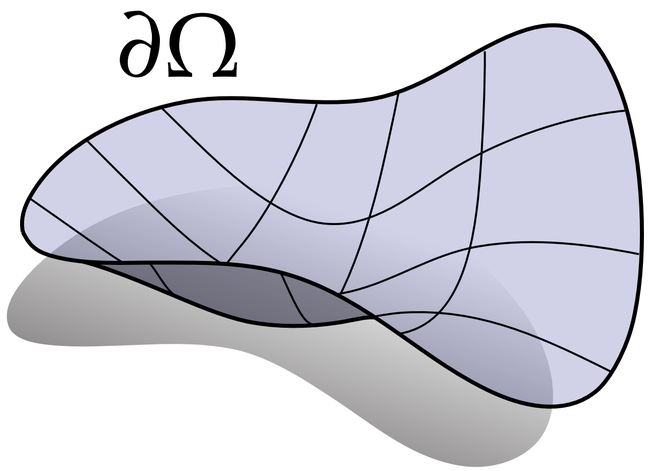
Vector Area, continued

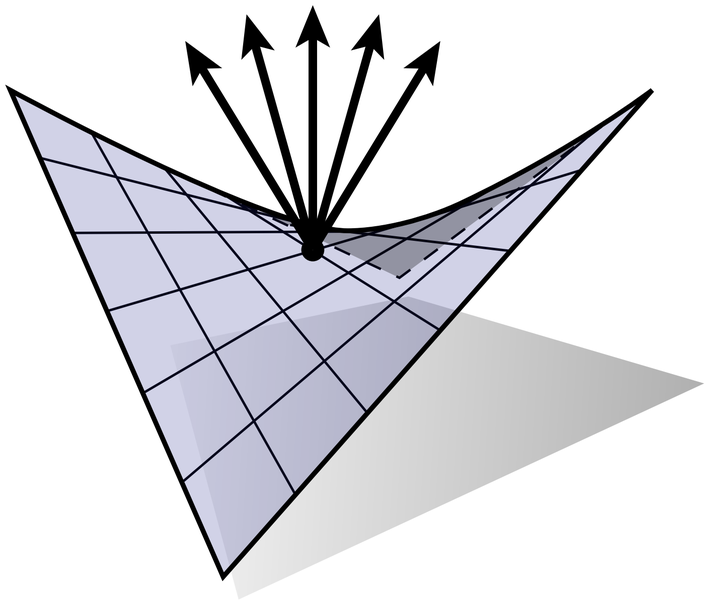
- By expressing vector area this way, we make an interesting observation:

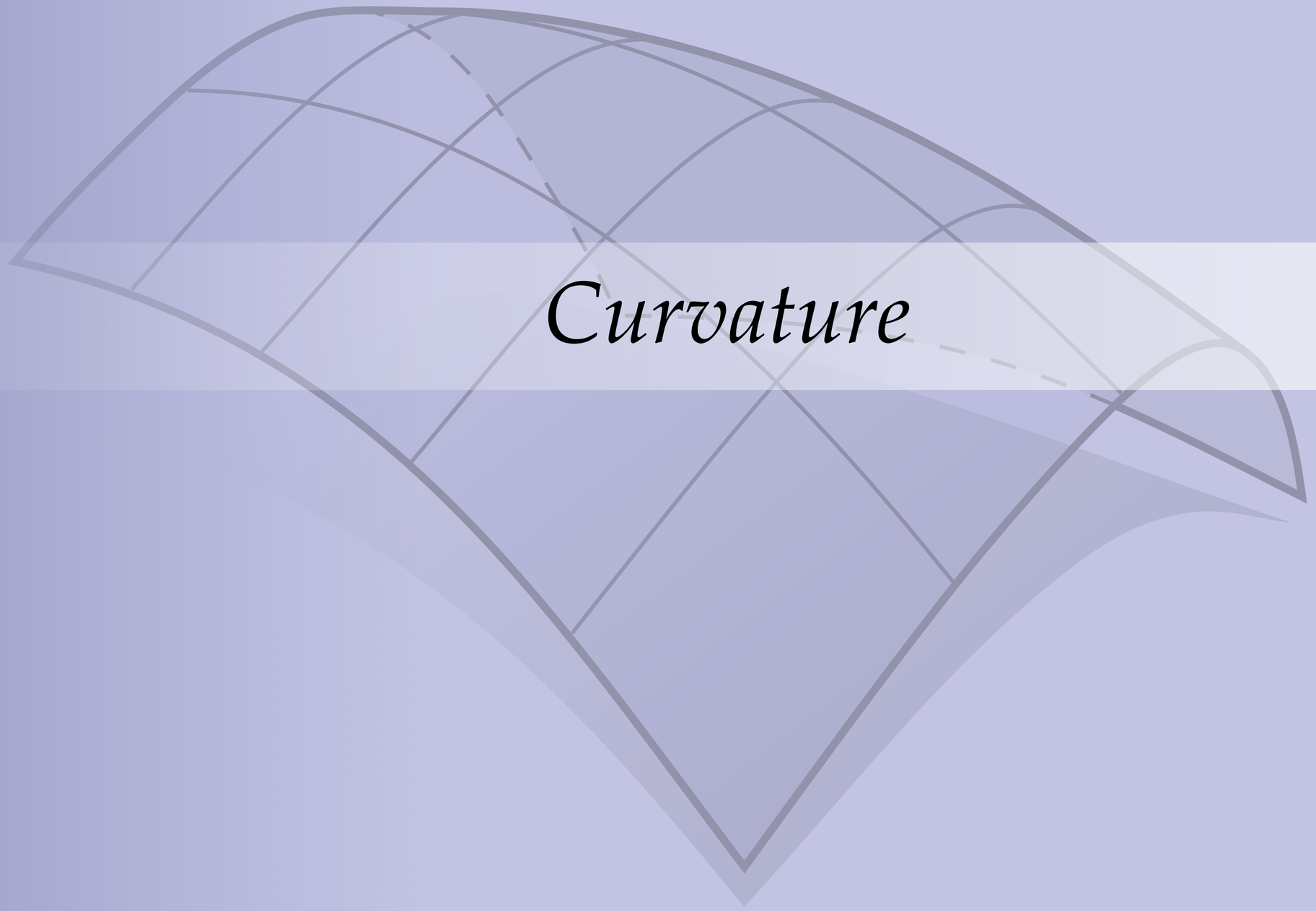
$$2\int_\Omega N\,dA = \int_\Omega df \wedge df = \int_\Omega d(f\,df) = \int_{\partial\Omega} f\,df = \int_{\partial\Omega} f(s) \times df(T(s))\,ds$$

- Hence, vector area is the same for any two patches w/ same boundary
- Can define "normal" given **only** boundary (*e.g.*, nonplanar polygon)
- **Corollary:** *integral of normal vanishes for any closed surface*





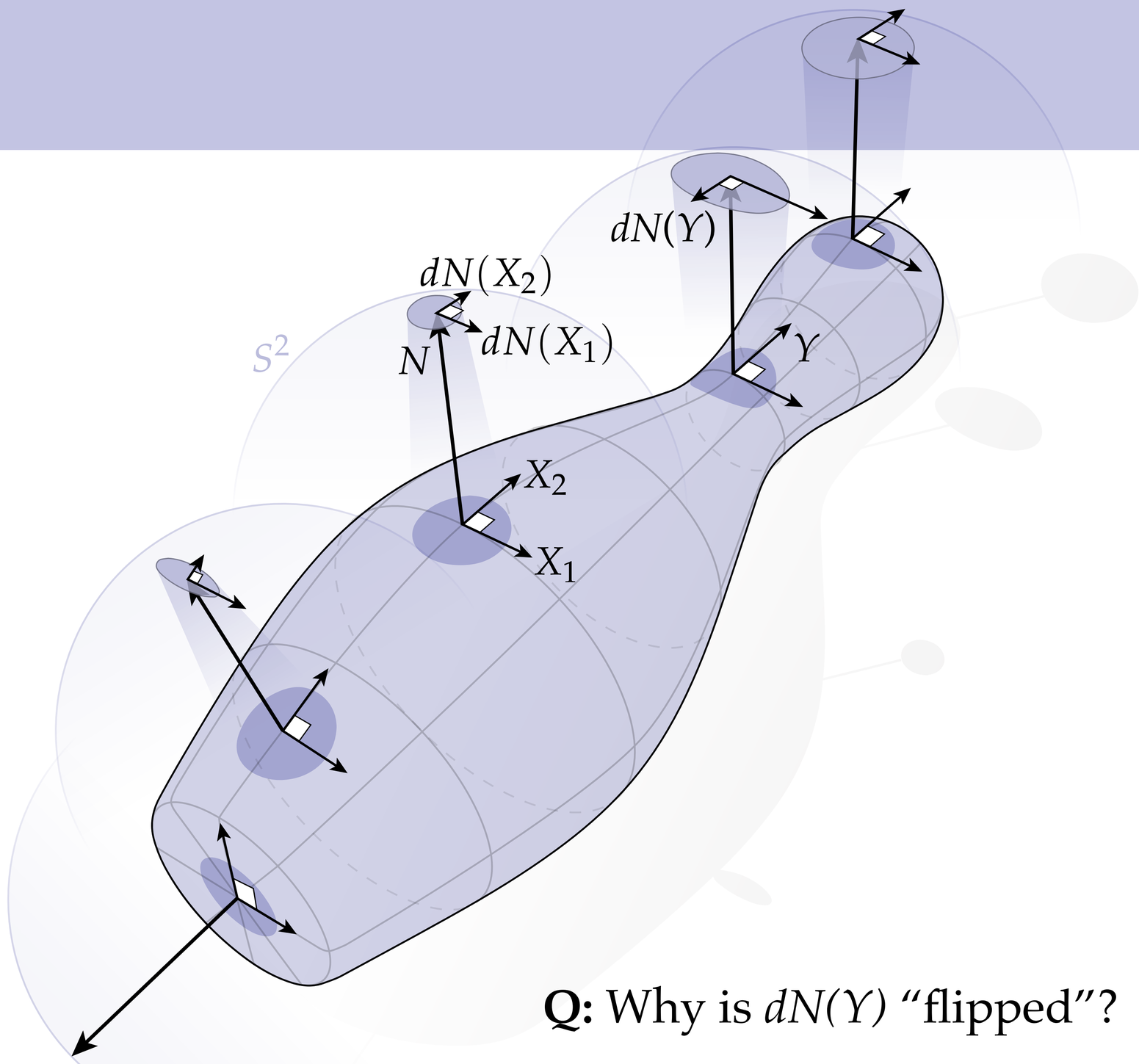




Curvature

Weingarten Map

- The **Weingarten map** dN is the differential of the Gauss map N
- At each point, tells us the change in the normal vector along any given direction X
- Since change in *unit* normal cannot have any component in the normal direction, $dN(X)$ is always tangent to the surface
- Can also think of it as a vector tangent to the unit sphere S^2



Q: Why is $dN(Y)$ "flipped"?

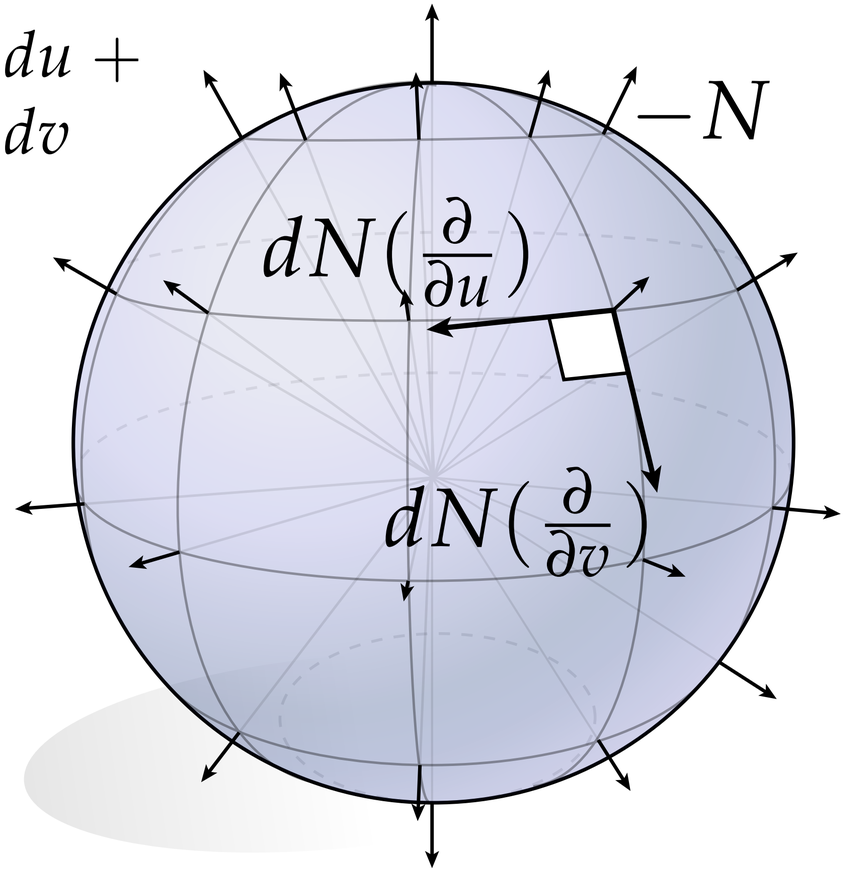
Weingarten Map—Example

- Recall that for the sphere, $N = -f$. Hence, Weingarten map dN is just $-df$:

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

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

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

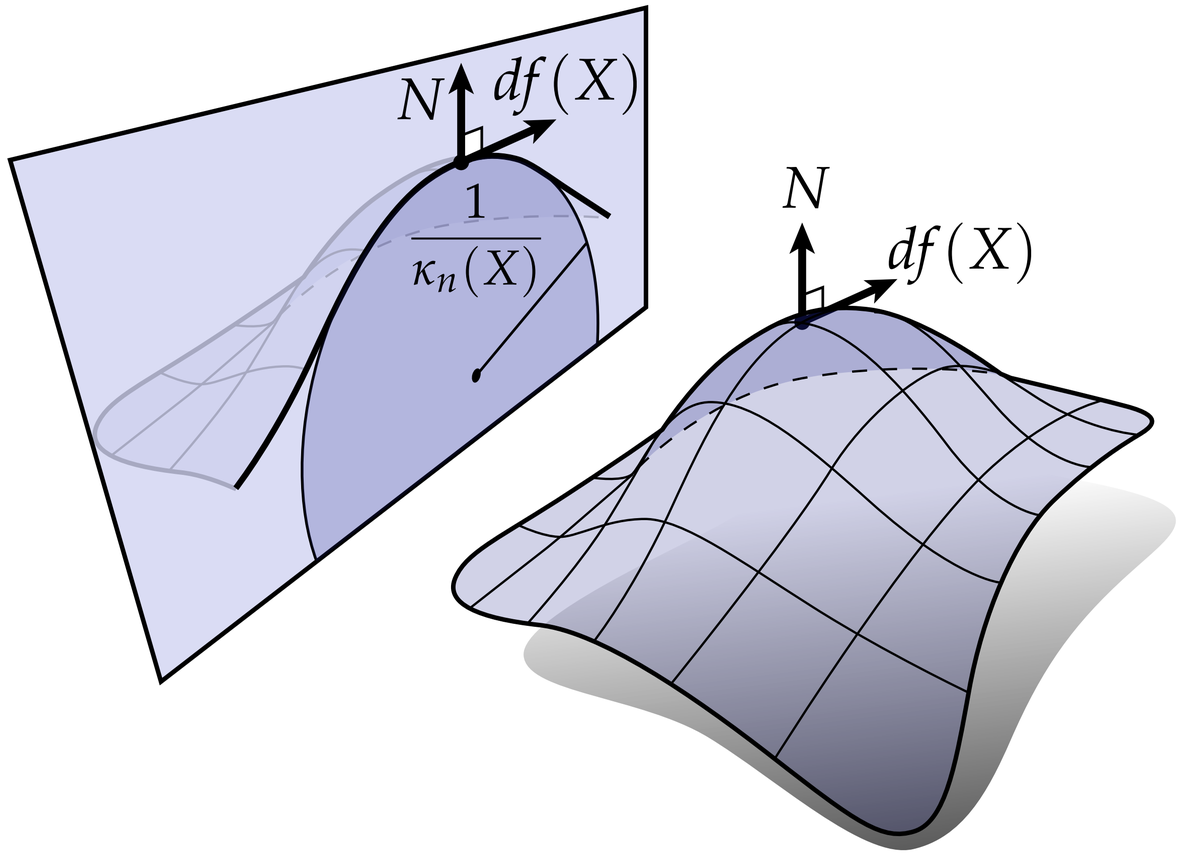


Key idea: computing the Weingarten map is no different from computing the differential of a surface.

Normal Curvature

- For curves, curvature was the rate of change of the *tangent*; for immersed surfaces, we'll instead consider how quickly the *normal* is changing.*
- In particular, **normal curvature** is rate at which normal is bending along a given tangent direction:

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



- Equivalent to intersecting surface with normal-tangent plane and measuring the usual curvature of a plane curve

*For plane curves, what would happen if we instead considered change in N?

Normal Curvature—Example

Consider a parameterized cylinder:

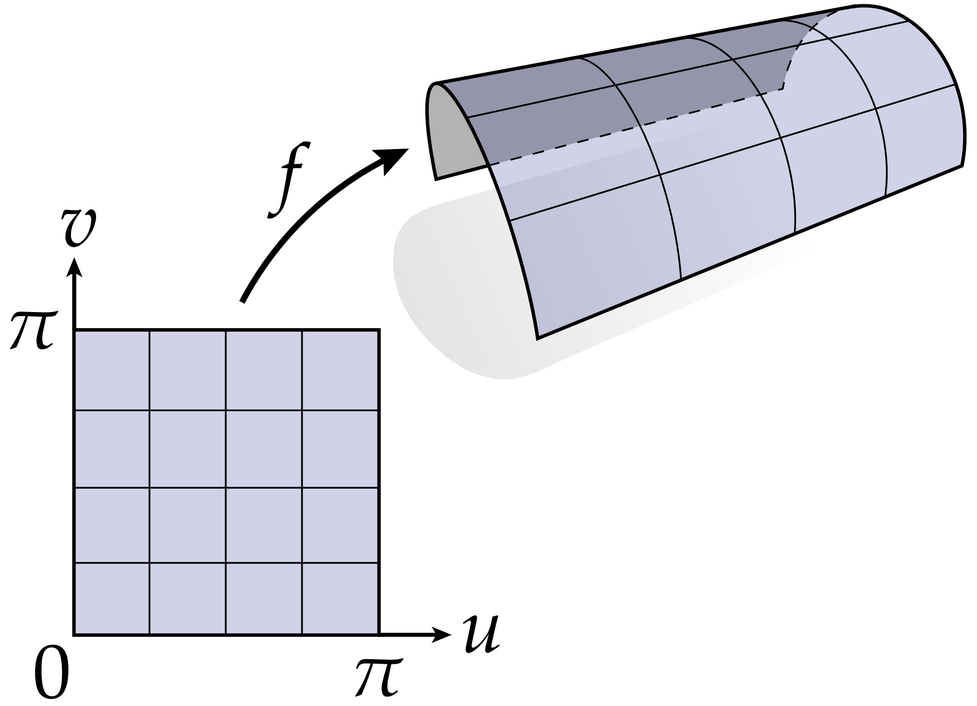
$$f(u,v) := (\cos(u), \sin(u), v)$$

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

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

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

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



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

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

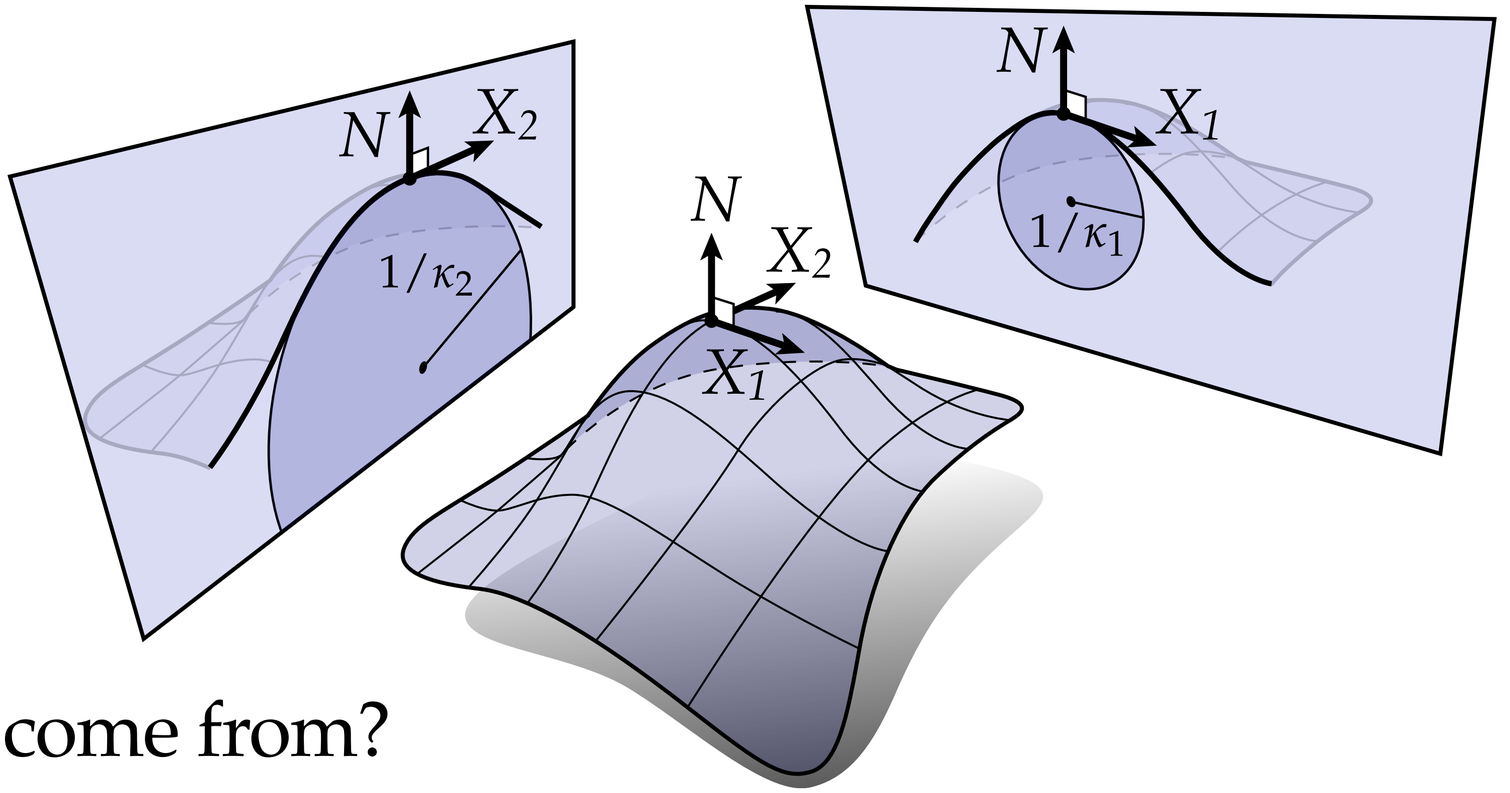
Q: Does this result make sense geometrically?

Principal Curvature

- Among all directions X, there are two **principal directions** X_1, X_2 where normal curvature has minimum/maximum value (respectively)
- Corresponding normal curvatures are the **principal curvatures**
- Two critical facts*:

1. $g(X_1, X_2) = 0$

2. $dN(X_i) = \kappa_i df(X_i)$



Where do these relationships come from?

Shape Operator

- The change in the normal N is always *tangent* to the surface
- Must therefore be some linear map S from tangent vectors to tangent vectors, called the **shape operator**, such that

$$df(SX) = dN(X)$$

- Principal directions are the *eigenvectors* of S
- Principal curvatures are *eigenvalues* of S
- **Note:** S is not a symmetric matrix! Hence, eigenvectors are not orthogonal in R^2; only orthogonal with respect to induced metric g.

Shape Operator — Example

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

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

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

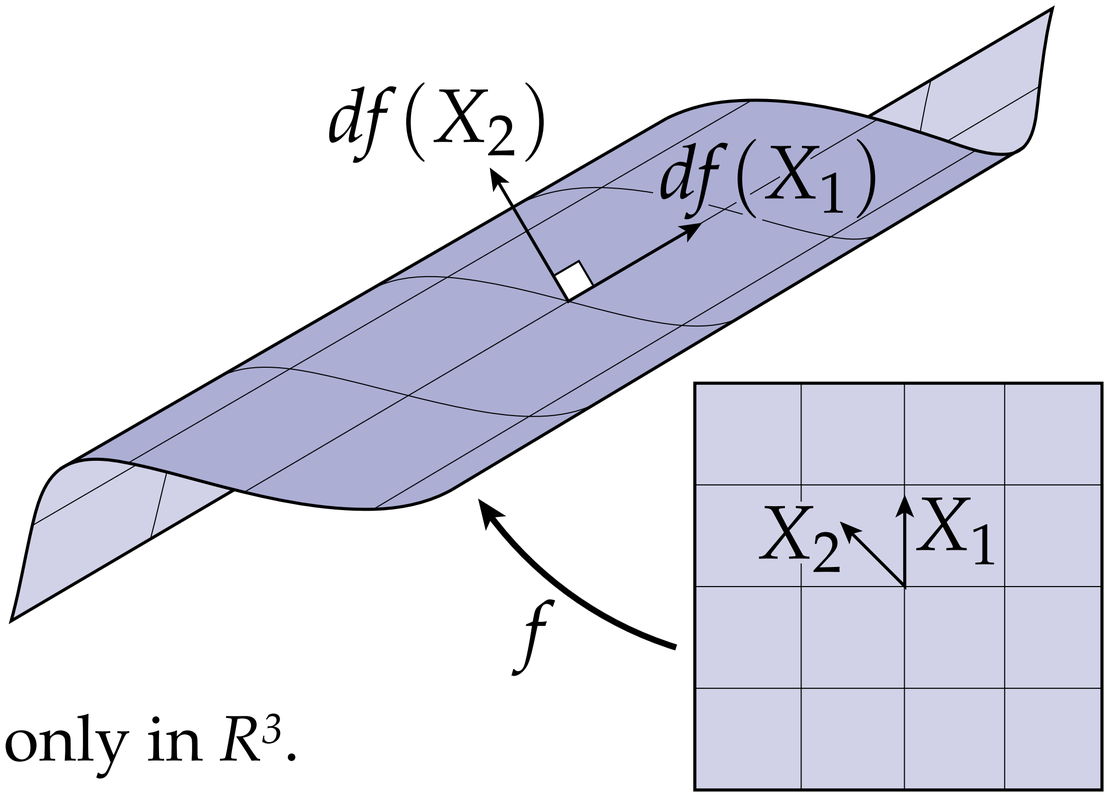
$$\boxed{df \circ S = dN}$$

$$\begin{bmatrix} -\sin(u) & 0 \\ \cos(u) & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} = \begin{bmatrix} -\sin(u) & 0 \\ \cos(u) & 0 \\ 0 & 0 \end{bmatrix}$$

$$\Rightarrow S = \begin{bmatrix} 1 & 0 \\ -1 & 0 \end{bmatrix} \quad X_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad X_2 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

$$df(X_1) = (0,0,1) \qquad \kappa_1 = 0$$

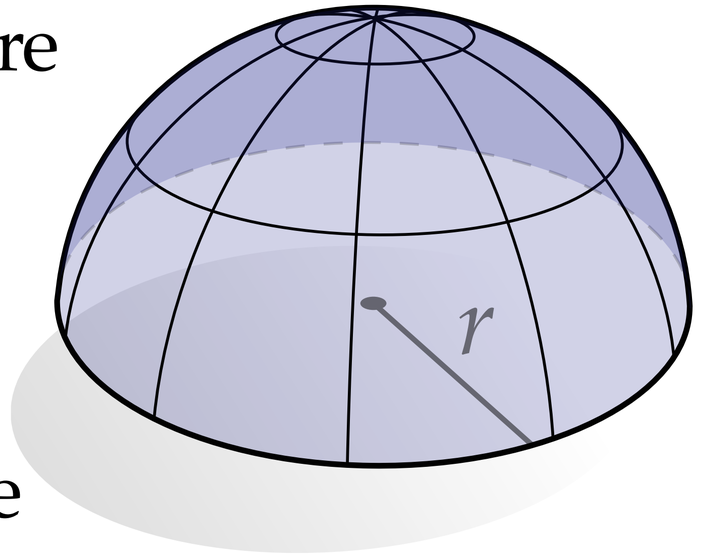
$$df(X_2) = (\sin(u), -\cos(u), 0) \qquad \kappa_2 = 1$$



Key observation: principal directions orthogonal only in R^3.

Umbilic Points

- Points where principal curvatures are equal are called **umbilic points**
- Principal *directions* are not uniquely determined here
- What happens to the shape operator S?
  - May still have full rank!
  - Just have repeated eigenvalues, 2-dim. eigenspace

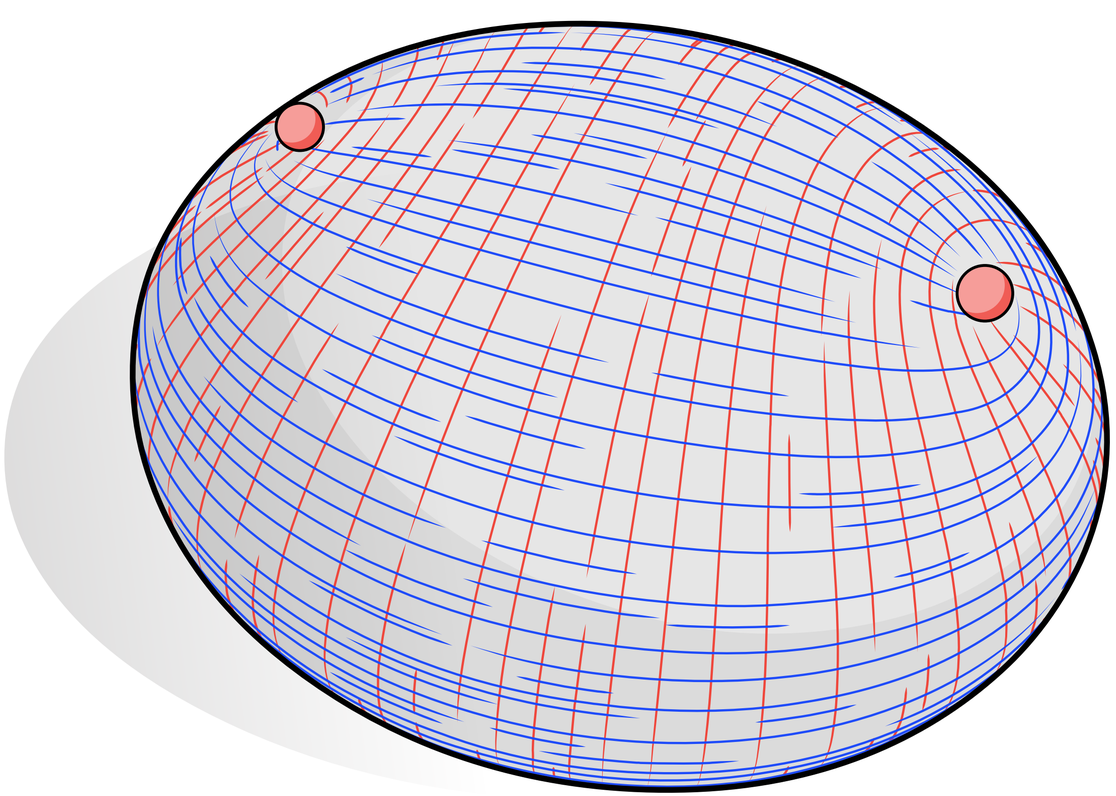


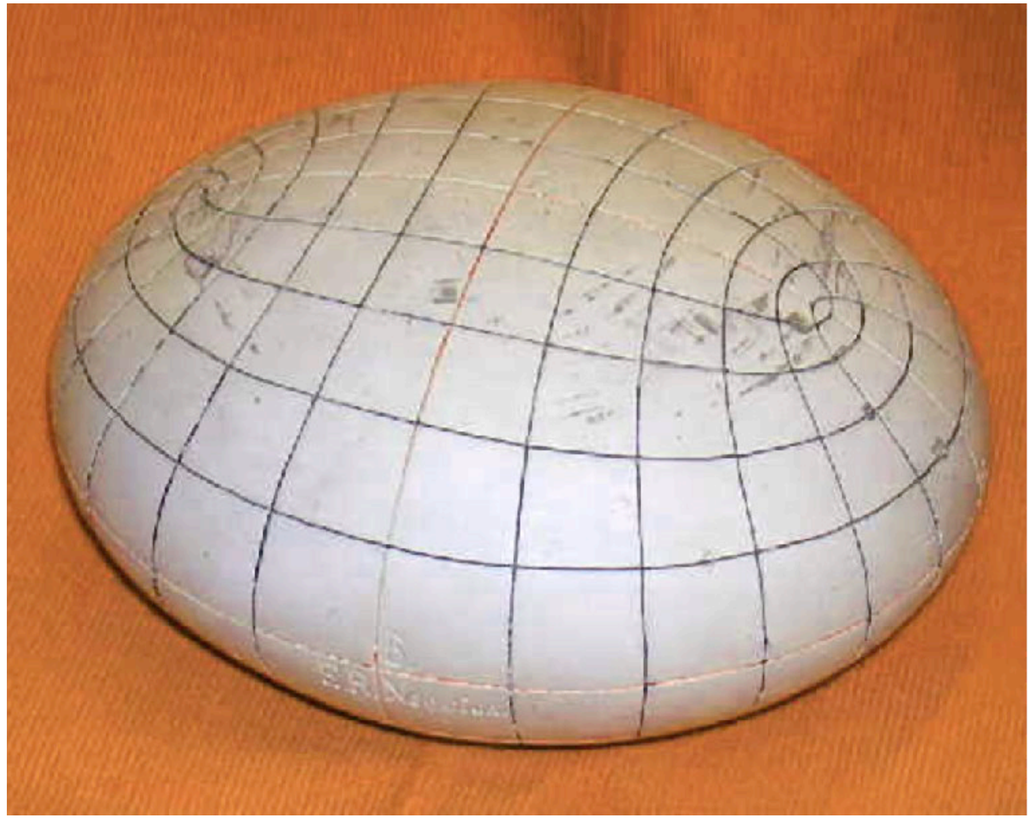
$$S = \begin{bmatrix} 1/r & 0 \\ 0 & 1/r \end{bmatrix} \qquad \kappa_1 = \kappa_2 = \frac{1}{r} \qquad \forall X,\ SX = \tfrac{1}{r}X$$

Could still of course choose (arbitrarily) an orthonormal pair X_1, X_2…

Principal Curvature Nets

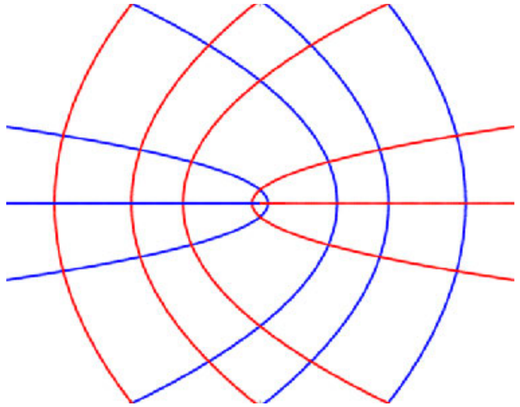
- Walking along principal direction field yields **principal curvature lines**
- Collection of all such lines is called the **principal curvature network**



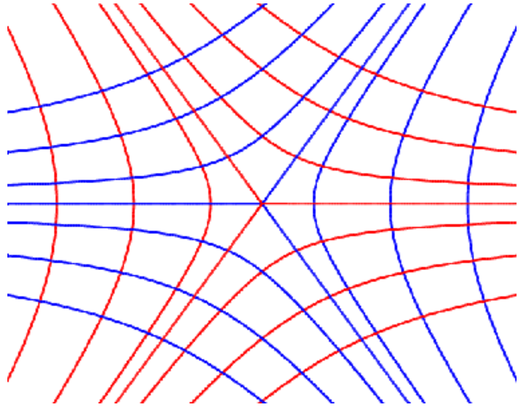


Topological Invariance of Umbilic Count

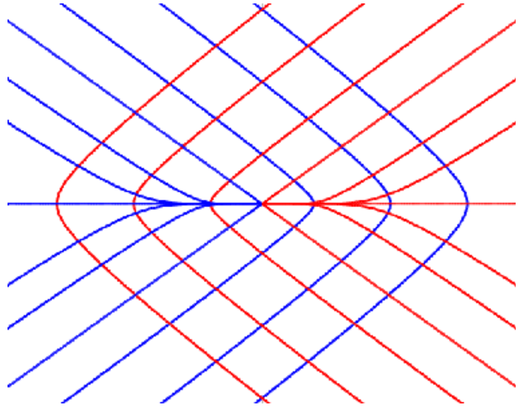
Can classify regions around umbilics into three types based on behavior of principal network: *lemon*, *star*, and *monstar*



lemon (k_1)



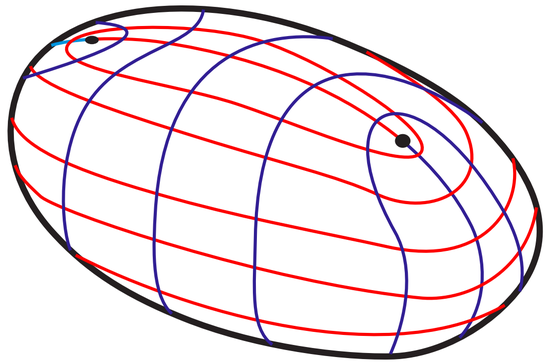
star (k_2)



monstar (k_3)

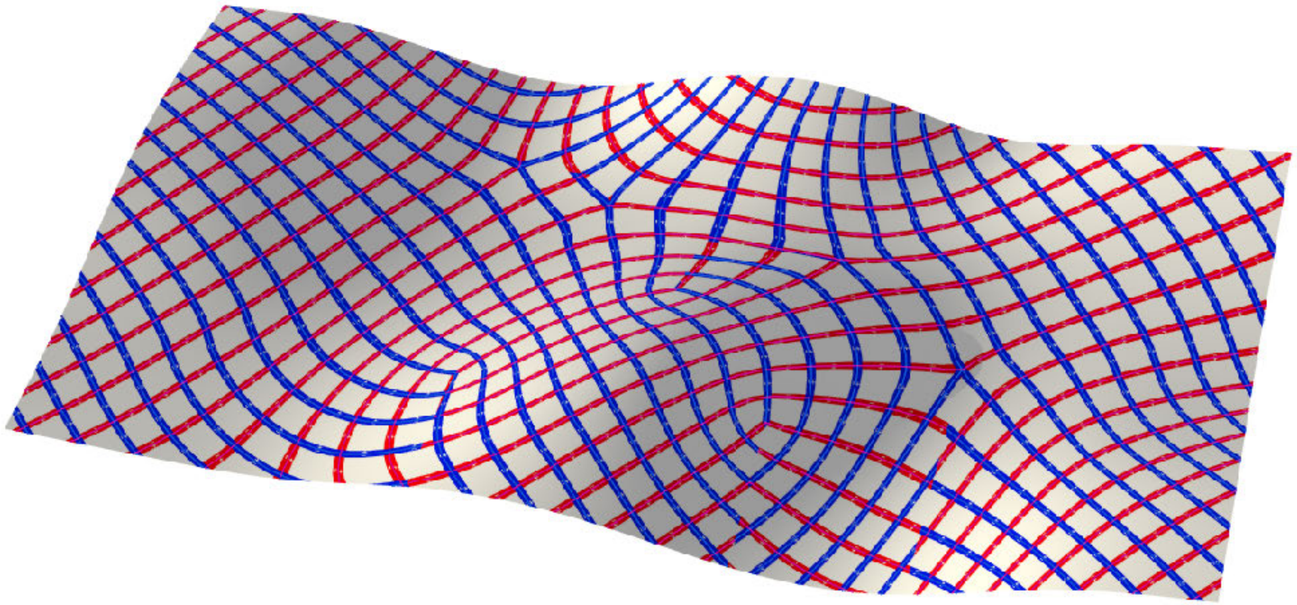
Fact. If k_1, k_2, k_3 are number of umbilics of each type, then

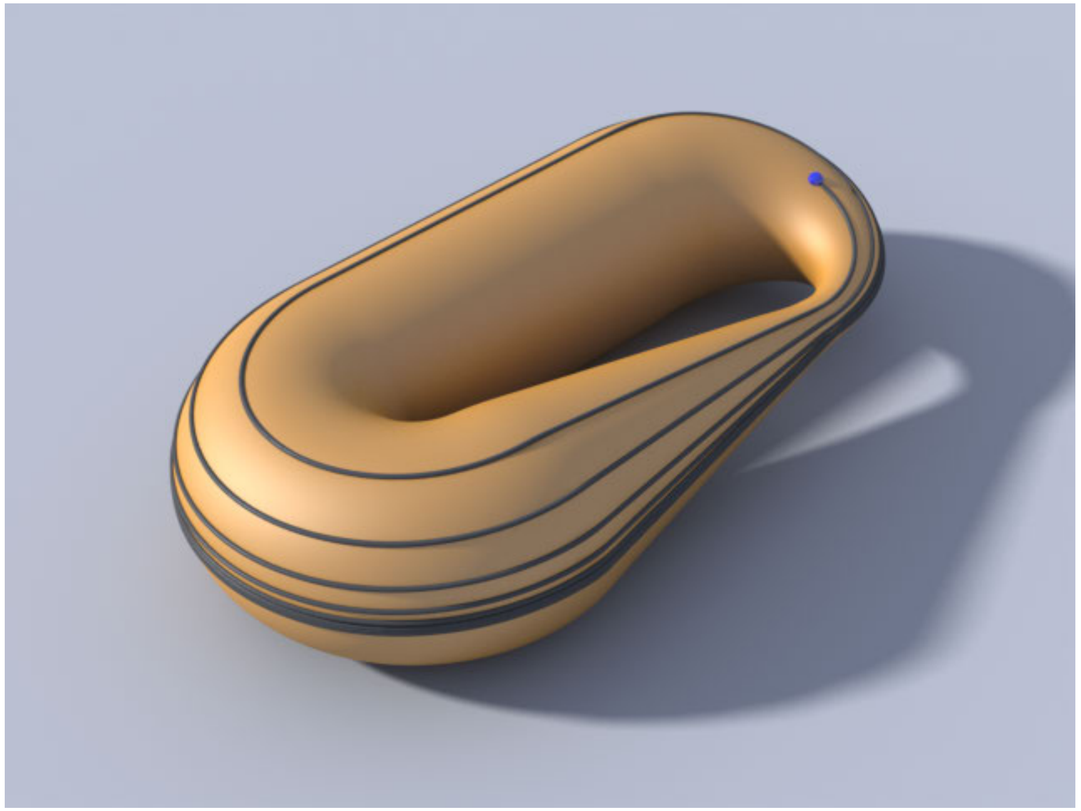
$$\kappa_1 - \kappa_2 + \kappa_3 = 2\chi$$



Separatrices and Spirals

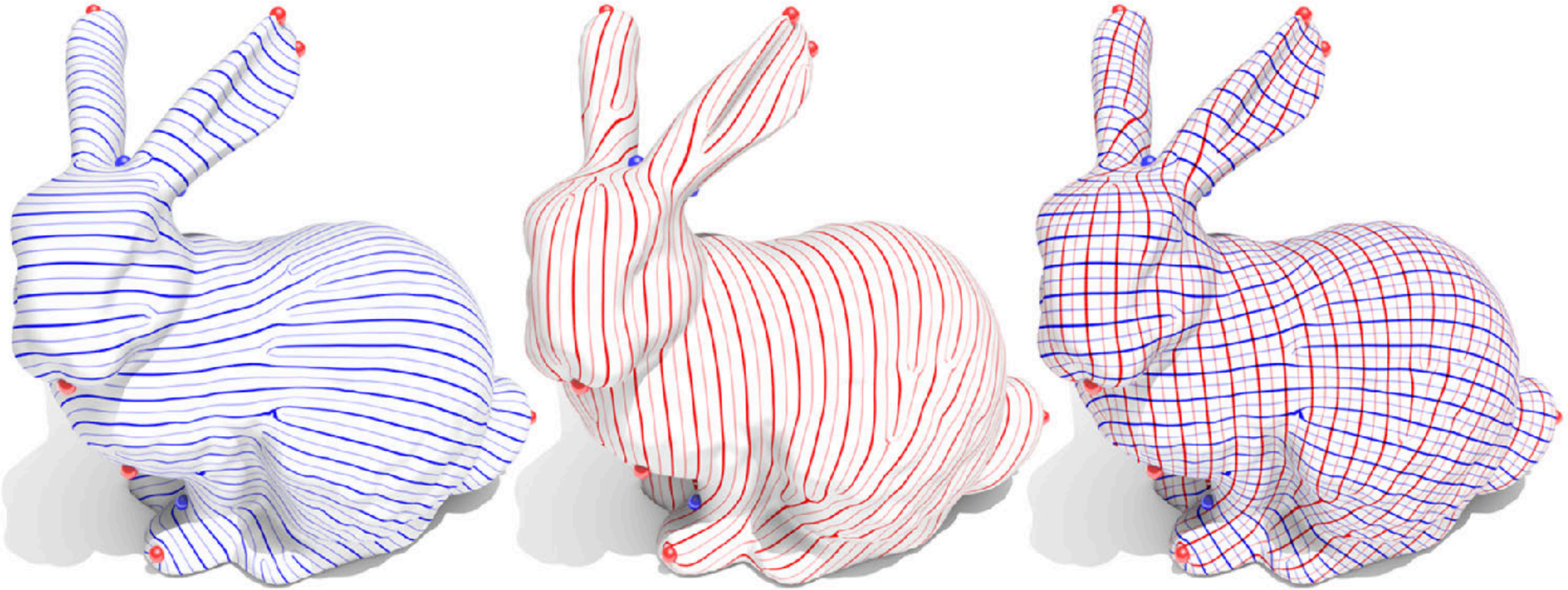
- If we walk along a principal curvature line, where do we end up?
- Sometimes, a curvature line terminates at an umbilic point in both directions; these so-called **separatrices** (can) split network into regular patches.
- Other times, we make a closed loop. More often, however, behavior is *not* so nice!





Application—Quad Remeshing

- Recent approach to meshing: construct net *roughly* aligned with principal curvature—but with separatrices & loops, not spirals.



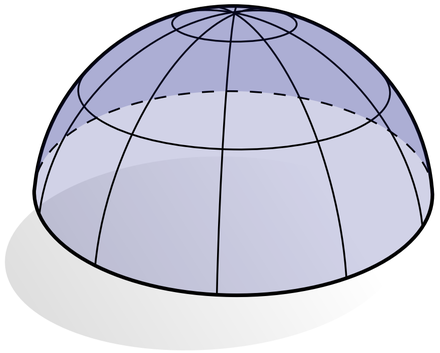
from Knöppel, Crane, Pinkall, Schröder, *"Stripe Patterns on Surfaces"*

Gaussian and Mean Curvature

Gaussian and *mean* curvature also fully describe local bending:

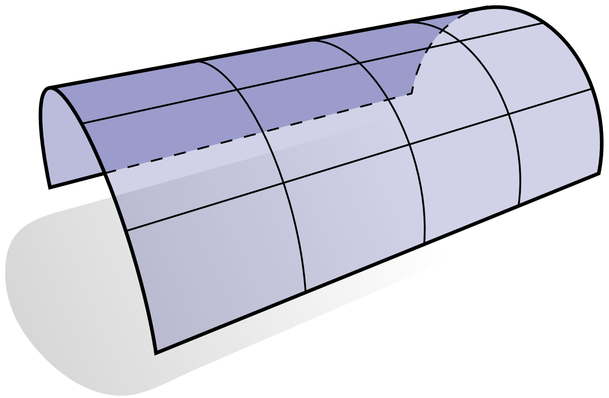
Gaussian $K := \kappa_1 \kappa_2$

mean* $H := \frac{1}{2}(\kappa_1 + \kappa_2)$



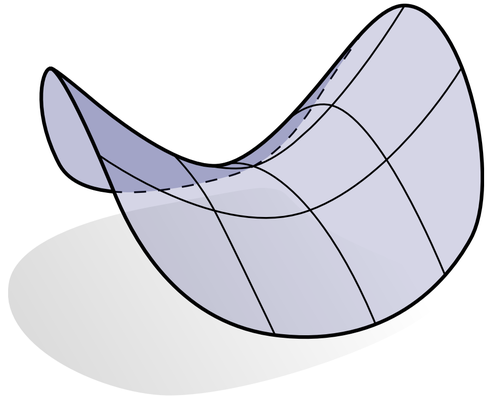
$K > 0$

$H \neq 0$



"developable" $K = 0$

$H \neq 0$



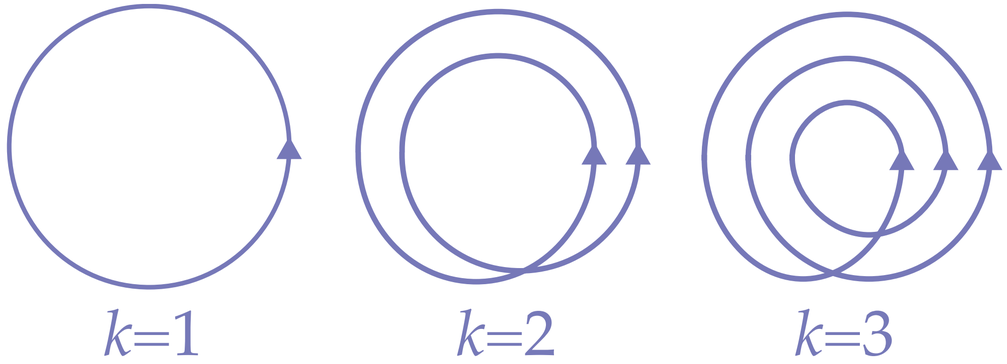
$K < 0$

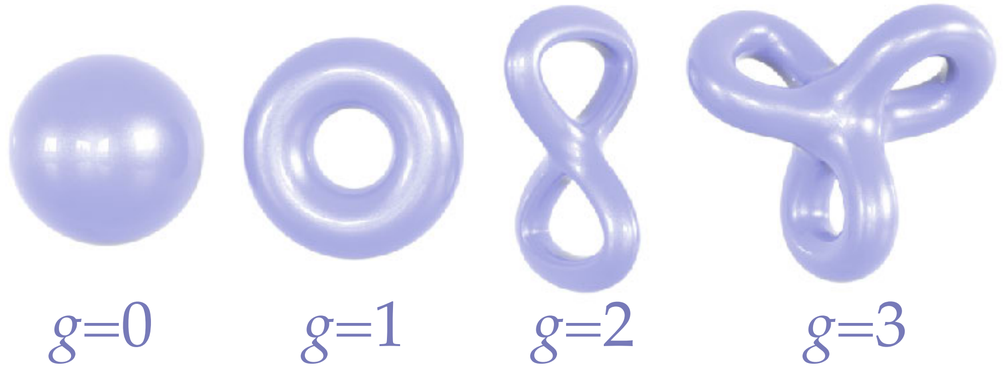
"minimal" $H = 0$

*__Warning:__ another common convention is to omit the factor of $1/2$

Gauss-Bonnet Theorem

- Recall that the total curvature of a closed plane curve was always equal to 2π times *turning number k*
- **Q:** Can we make an analogous statement about surfaces?
- **A:** Yes! Gauss-Bonnet theorem says total Gaussian curvature is always 2π times *Euler characteristic* χ
- Euler characteristic can be expressed in terms of the *genus* (number of "handles")





Curves

$$\int_0^L \kappa \, ds = 2\pi k$$

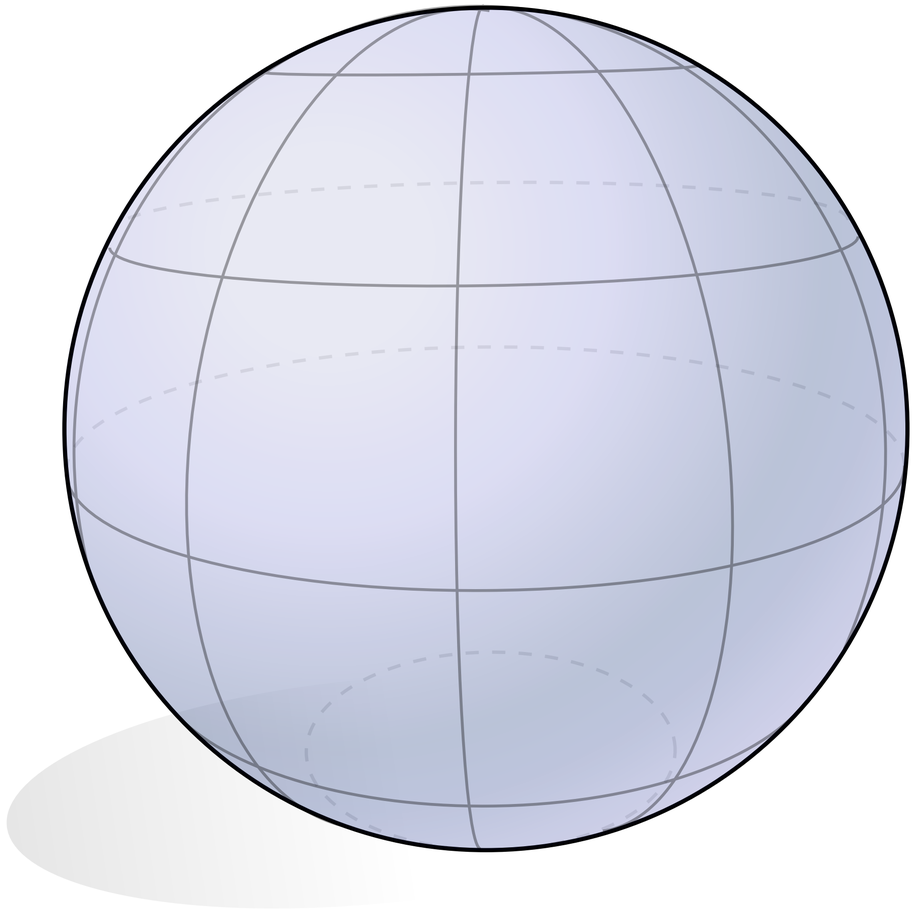
Surfaces

$$\int_M K \, dA = 2\pi\chi$$

Total Mean Curvature?

Theorem (Minkowski): for a regular closed embedded surface,

$$\int_M H \, dA \geq \sqrt{4\pi A}$$



Q: When do we get equality?

A: For a sphere.

Curvature of a Curve in a Surface

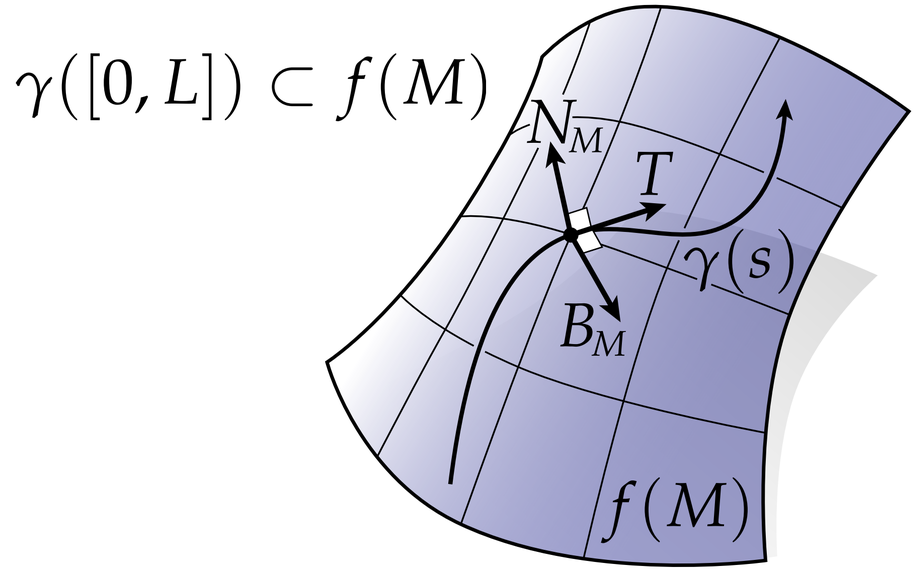
- Earlier, broke the "bending" of a space curve into curvature (κ) and torsion (τ)
- For a curve *in a surface*, can instead break into *normal* and *geodesic* curvature:

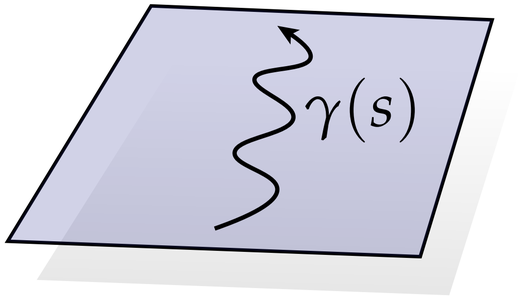
$$\kappa_n := \langle N_M, \tfrac{d}{ds} T \rangle$$

$$\kappa_g := \langle B_M, \tfrac{d}{ds} T \rangle$$

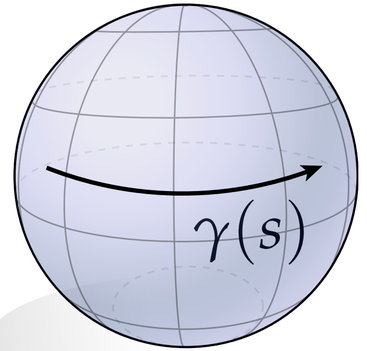
- T is still tangent of the curve; but unlike the Frenet frame, N_M is the normal of the surface and $B_M := T \times B_M$

Q: Why no third curvature $\langle T_M, \frac{d}{ds} T \rangle$?





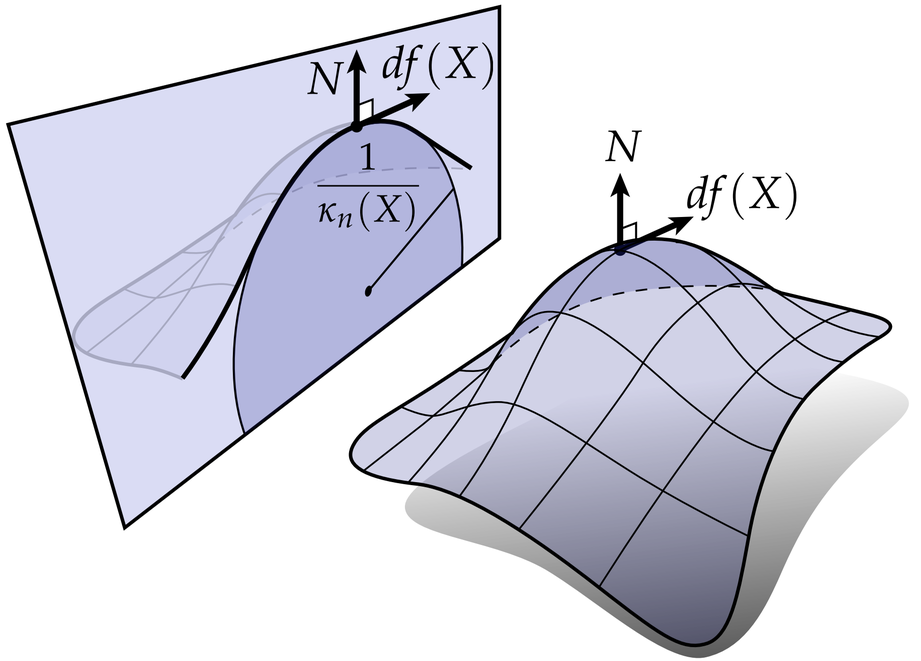
large κ_g;
small κ_n



large κ_n;
small κ_g

Second Fundamental Form

- Second fundamental form is closely related to principal curvature
- Can also be viewed as change in *first* fundamental form under motion in normal direction
- Why "fundamental?"First & second fundamental forms play role in important theorem…



$$\mathbf{II}(X,Y) := \langle dN(X), df(Y) \rangle$$

$$\kappa_N(X) := \frac{df(X), dN(X)}{|df(X)|^2} = \frac{\mathbf{II}(X,X)}{\mathbf{I}(X,X)}$$

Fundamental Theorem of Surfaces

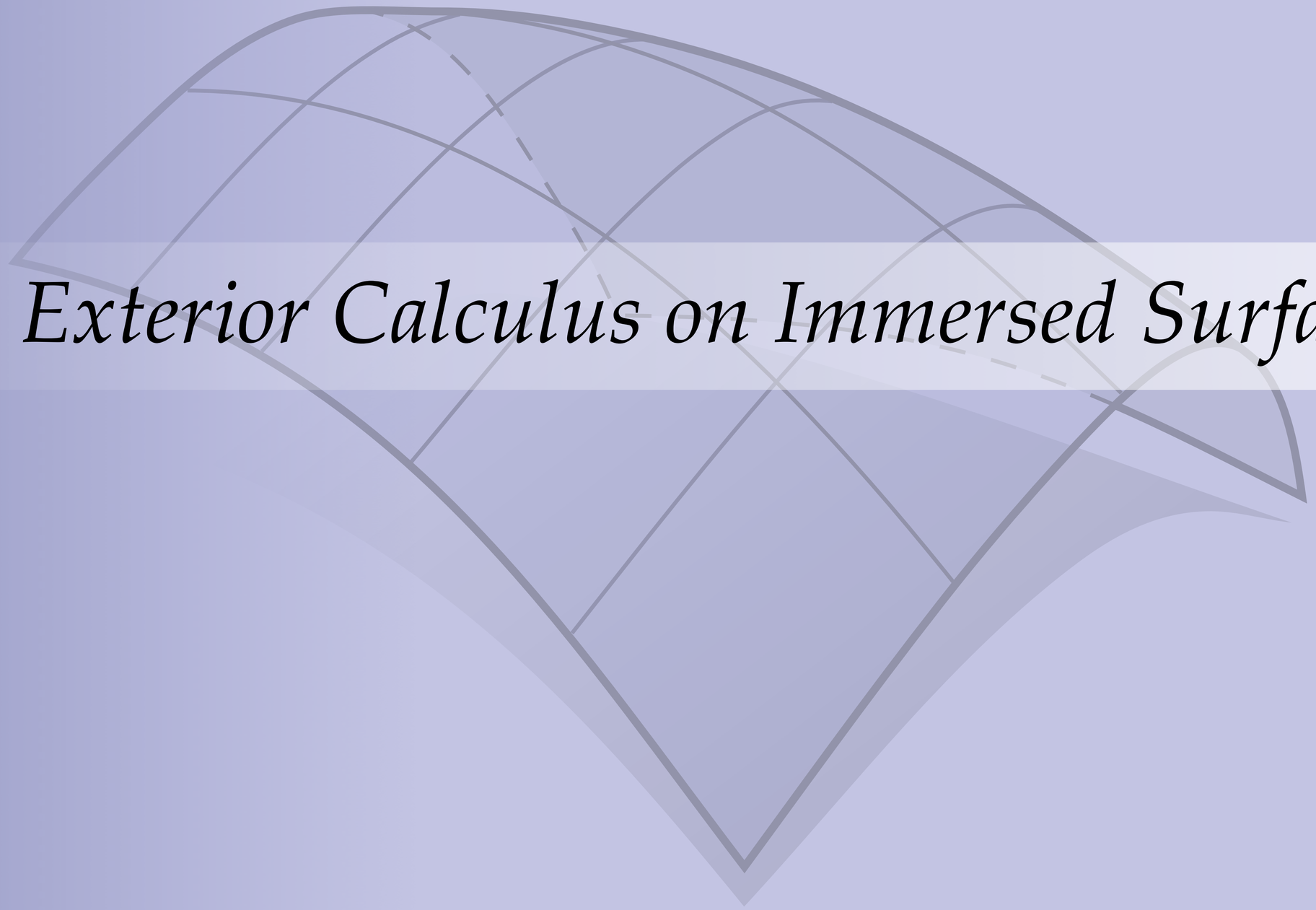
- **Fact.** Two surfaces in R^3 are congruent if and only if they have the same first and second fundamental forms
  - …However, not every pair of bilinear forms **I**, **II** on a domain U describes a valid surface—must satisfy the **Gauss Codazzi** equations
- Analogous to fundamental theorem of plane curves: determined up to rigid motion by curvature
  - …However, for *closed* curves not every curvature function is valid (*e.g.*, must integrate to $2k\pi$)

Other Descriptions of Surfaces?

- Classic question in differential geometry:

> *"What data is sufficient to completely determine a surface in space?"*

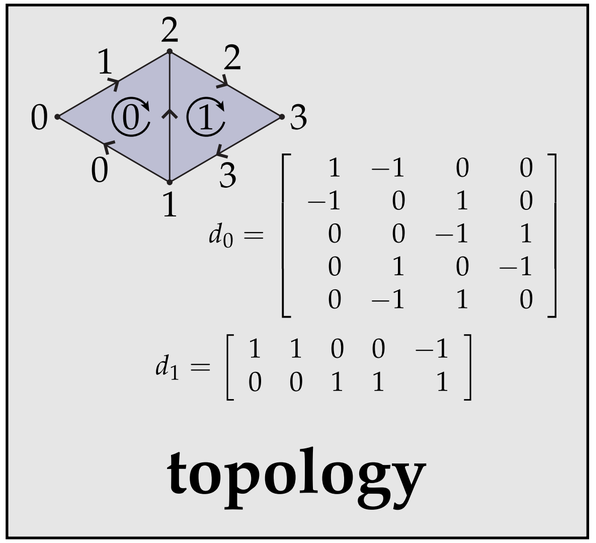
- Many possibilities…
  - First & second fundamental form (Gauss-Codazzi)
  - Mean curvature and metric (up to "Bonnet pairs")
  - Convex surfaces: metric alone is enough (Alexandrov/Pogorolev)
  - Gauss curvature essentially determines metric (Kazdan-Warner)
- …in general, still a surprisingly murky question!

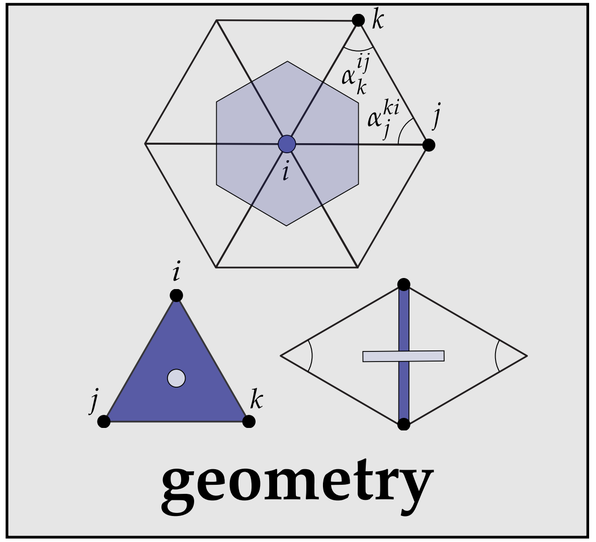


Exterior Calculus on Immersed Surfaces

Exterior Calculus on Curved Domains

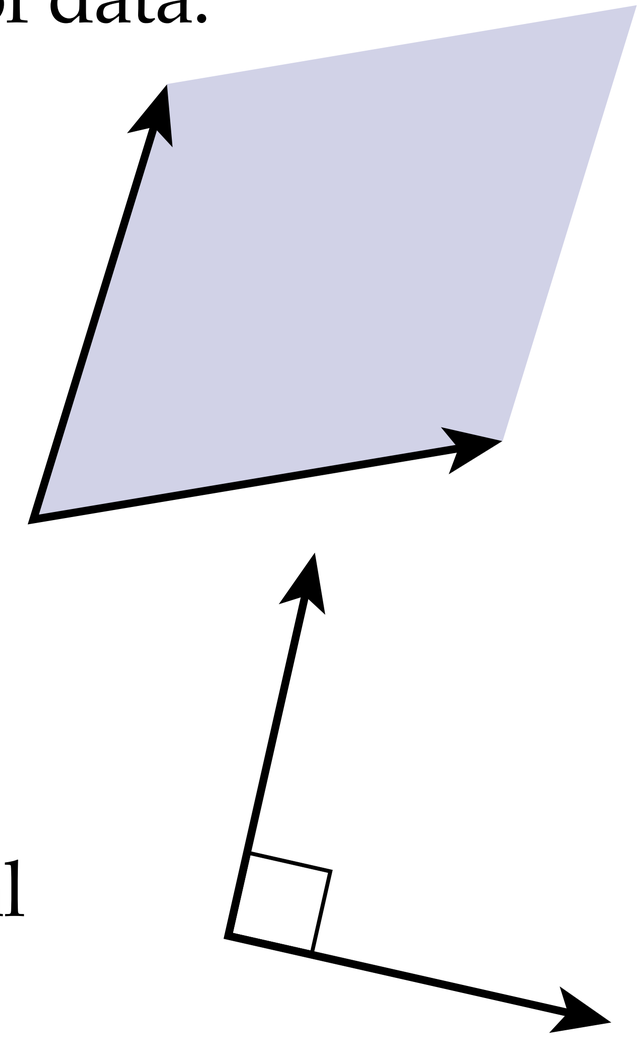
- Initial study of differential forms was in **flat** Euclidean R^n
- How do we do exterior calculus on **curved** spaces?
- Recall that operators nicely "split up" topology & geometry:
  - **(topology)** wedge product (^), exterior derivative (d)
  - **(geometry)** Hodge star (★)
- For instance, discrete d uses only mesh connectivity **(topology)**; discrete ★ involves only ratios of volumes **(geometry)**
- Therefore, to get exterior calculus to work with curved spaces, we just need to figure out what the Hodge star looks like!
- Traditionally taught from abstract **intrinsic** point of view; we'll start with the concrete **extrinsic** picture (which fewer people know… but is more directly relevant for real applications!)





Exterior Calculus on Immersed Surfaces

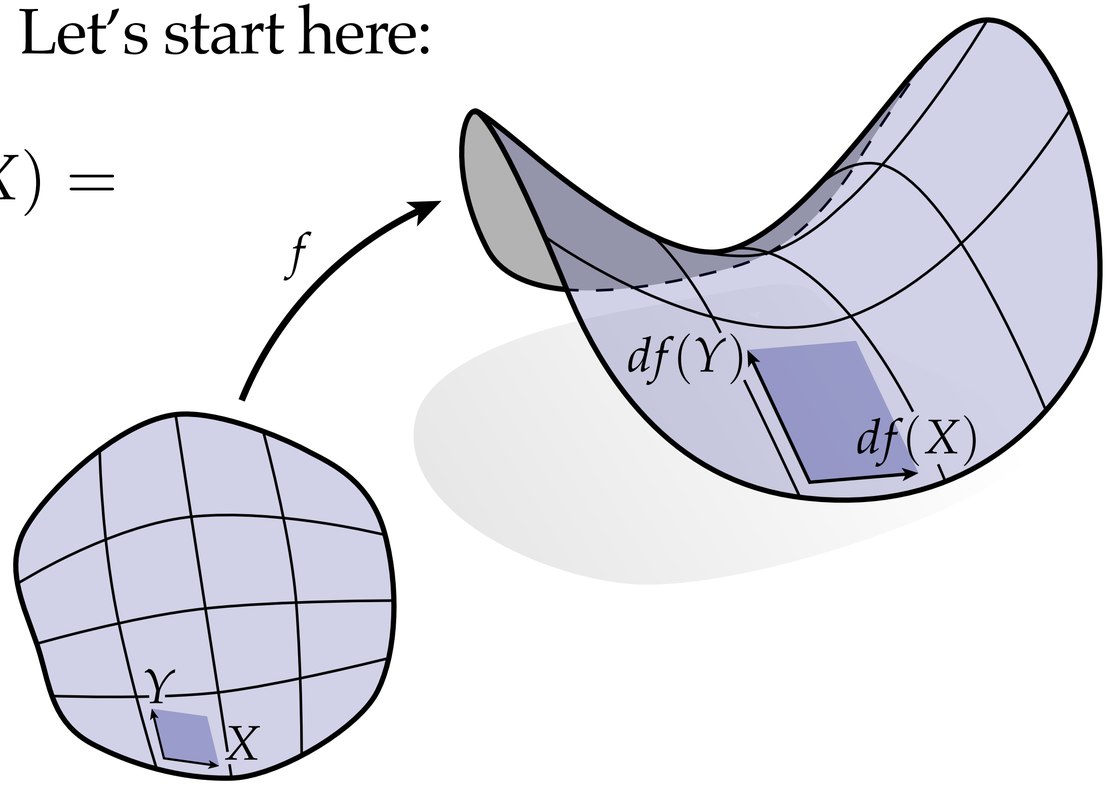
- For surface immersed in 3D, just need two pieces of data:
  - **Area form**—*"how big is a given region?"*
    - lets us define Hodge star on 0/2-forms
    - can express via cross product in R^3
  - **Complex structure**—*"how do we rotate by 90°?"*
    - lets us define Hodge star on 1-forms
    - can express via cross product w/ surface normal
- All of this data also determined by induced metric



Induced Area 2-Form

- What signed area should we associate with a pair of vectors X, Y on the domain?
- Not just their cross product! Need to account for "stretching" caused by immersion f
- What's the signed area of the stretched vector? Let's start here:

$$df \wedge df(X,Y) = df(X) \times df(Y) - df(Y) \times df(X) = 2df(X) \times df(Y)$$



Since $df(X)$ and $df(Y)$ are *tangent*, we get

$$df \wedge df(X,Y) = 2NdA(X,Y)$$

where dA is the area 2-form on $f(M)$. Hence,

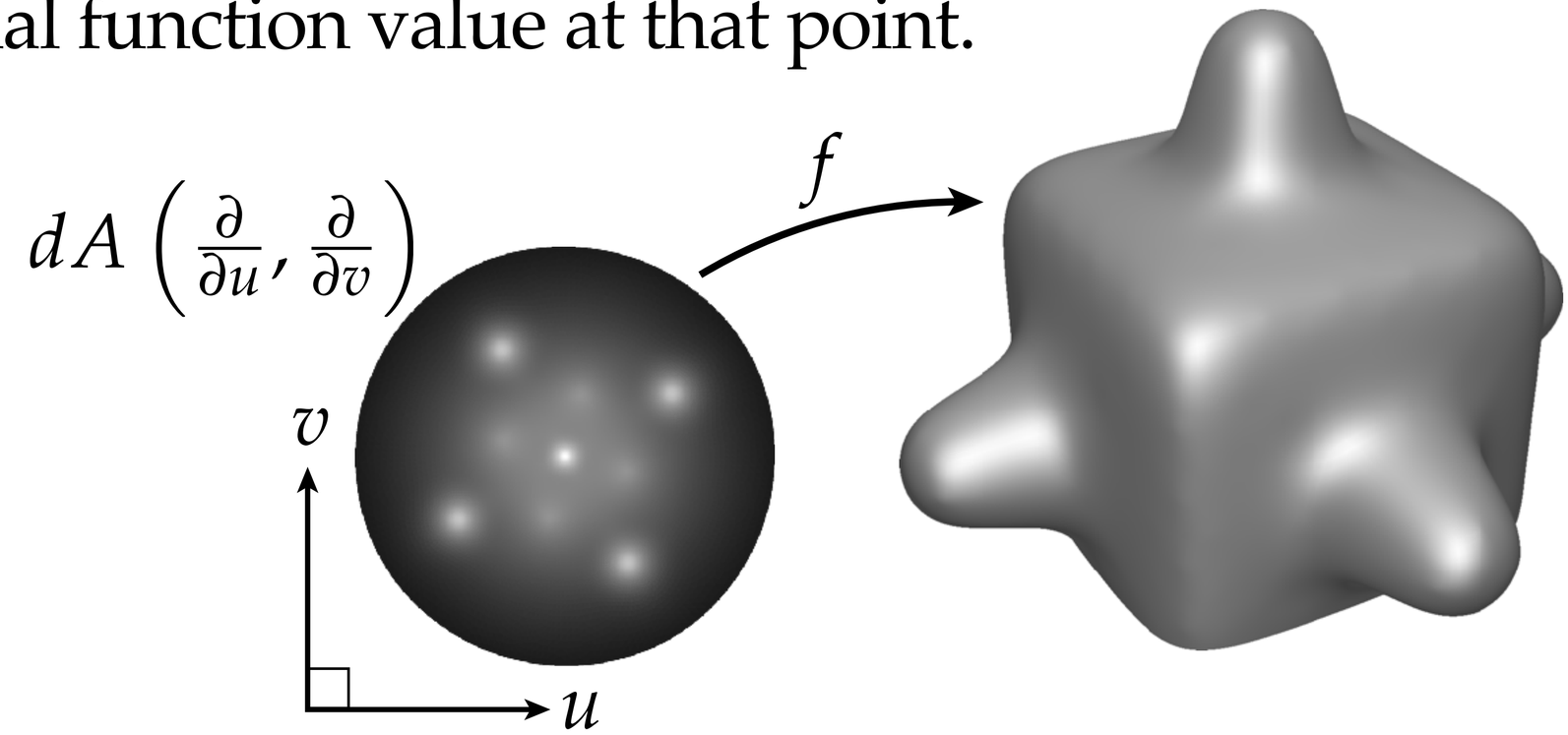
$$\boxed{dA = \tfrac{1}{2}\langle N, df \wedge df \rangle}$$

Induced Hodge Star on 0-Forms

- Given the area 2-form dA, can easily define Hodge star on 0-forms:

$$\phi \overset{\star}{\longmapsto} \phi \, dA$$

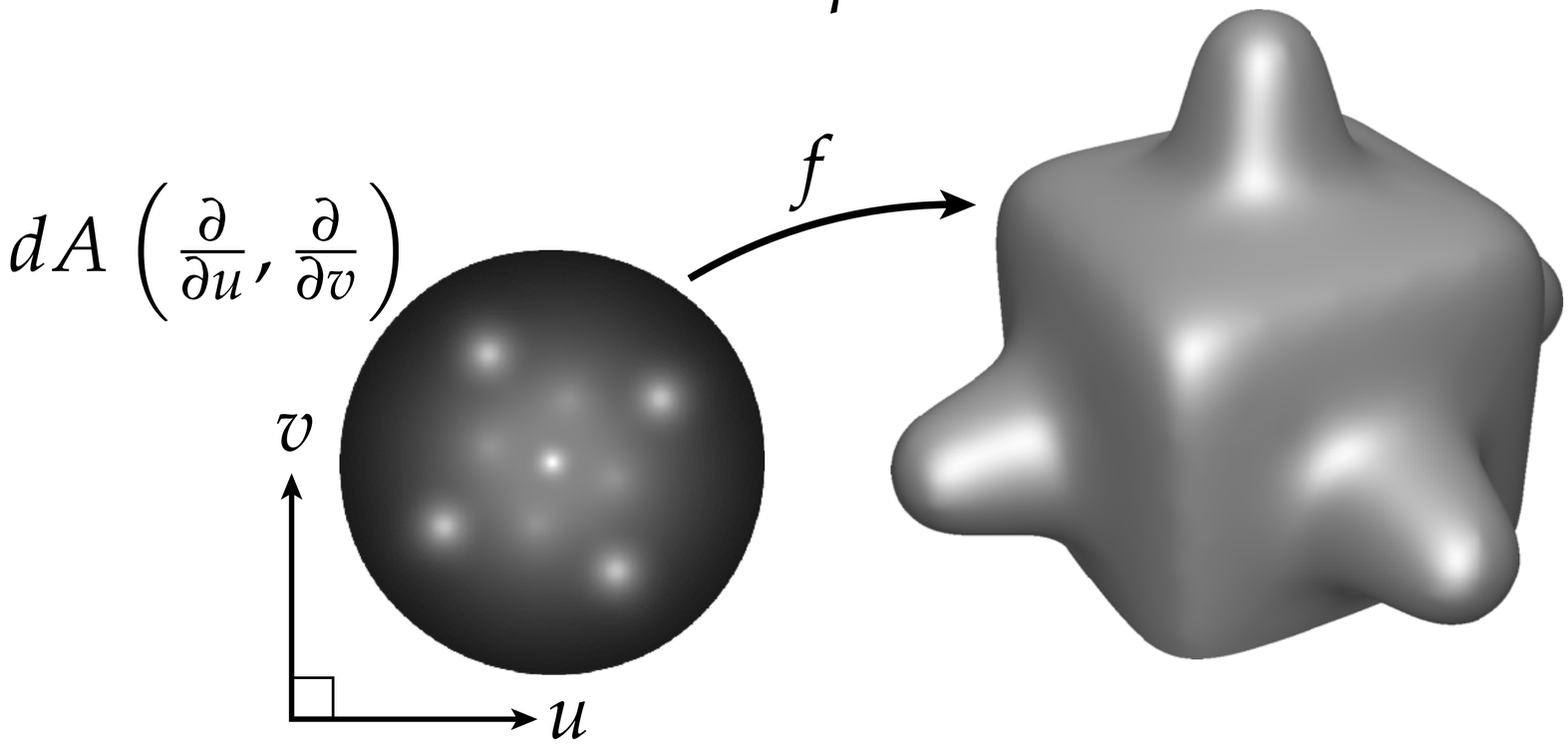
- **Meaning?** Applying this new 2-form to a unit area *on the surface* yields the original function value at that point.



Induced Hodge Star on 2-Forms

- To get the 2-form Hodge star, we just go the other way
- Suppose ω is a 2-form on $f(M)$. Then its Hodge dual is the unique 0-form ϕ such that

$$\omega = \phi \, dA$$



Complex Structure

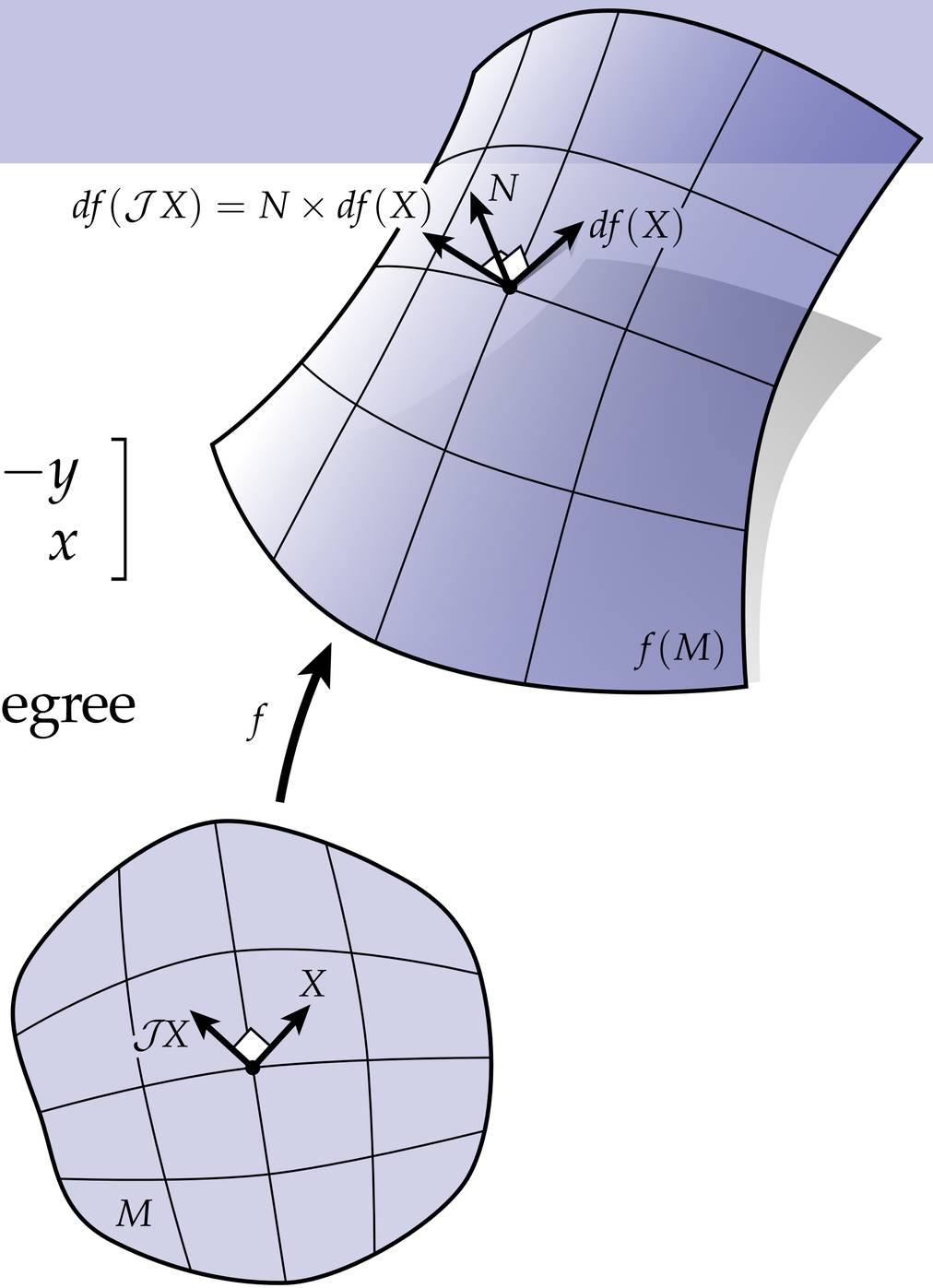
- The *complex structure** tells us how to rotate by 90°
- In R^2, we just replace (x,y) with $(-y,x)$:

$$\mathcal{J}_{\mathbb{R}^2} := \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \qquad \mathcal{J}_{\mathbb{R}^2} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} -y \\ x \end{bmatrix}$$

- For a surface immersed in R^3, we can express a 90-degree rotation via a cross product with the unit normal N:

$$df(\mathcal{J}_f X) := N \times df(X)$$

- This relationship uniquely determines $\mathcal{J}_f$
- An immersion is conformal if and only if $\mathcal{J}_f = \mathcal{J}_{\mathbb{R}^2}$



*Sometimes called *linear complex structure*; same thing for surfaces.

Complex Structure in Coordinates

- Suppose we want to explicitly compute the linear complex structure*
- Similar strategy to shape operator: solve a matrix equation for $\mathscr{J}$

$$\widehat{N} := \begin{bmatrix} 0 & -N_z & N_y \\ N_z & 0 & -N_x \\ -N_y & N_x & 0 \end{bmatrix}$$

cross product w/ normal

$$(N \times u = \widehat{N} u)$$

$$A := \begin{bmatrix} \partial f_x/\partial u & \partial f_x/\partial v \\ \partial f_y/\partial u & \partial f_y/\partial v \\ \partial f_z/\partial u & \partial f_z/\partial v \end{bmatrix}$$

Jacobian

$$J := \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix}$$

complex structure

$$\boxed{df(\mathcal{J}X) = N \times df(X)} \quad \Longrightarrow \quad \boxed{J = A^T \widehat{N} A}$$

*__Note:__ not something you do much in practice, but may help make definition feel more concrete…

Induced Hodge Star on 1-Forms

- Recall that for a 1-form α in the plane, applying $\star\alpha$ to a vector X is the same as applying α to a 90-degree rotation of X:

$$\star_{\mathbb{R}^2}\alpha(X) = \alpha(\mathcal{J}_{\mathbb{R}^2}X)$$

- For 1-forms on an immersed surface f, we instead want to apply a 90-degree rotation with respect to the surface itself:

$$\star_f\alpha(X) = \alpha(\mathcal{J}_f X)$$

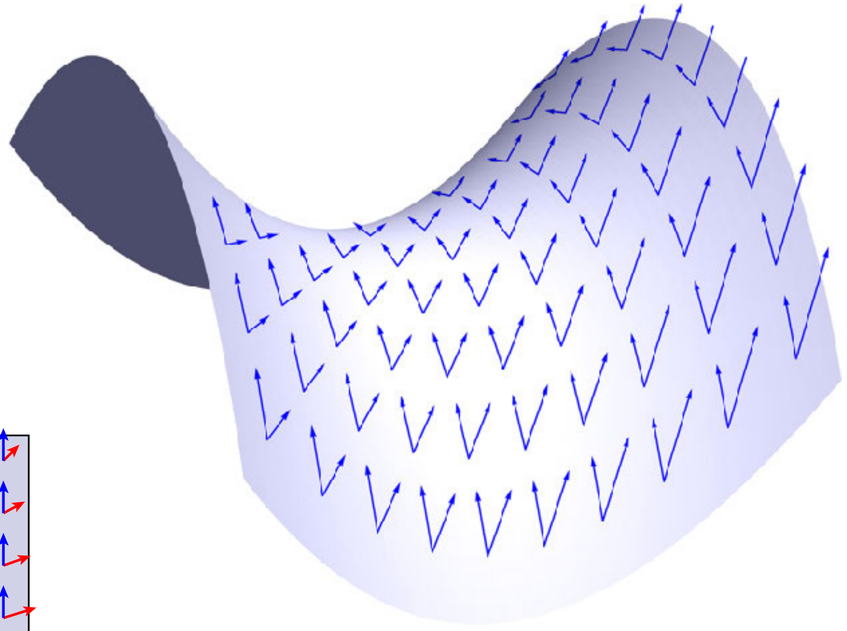
- At this point we have everything we need to do calculus on curved surfaces: 0-, 1-, and 2-form Hodge star. (Will see more general/abstract/intrinsic definitions for n-manifolds later on.)

Sharp and Flat on a Surface

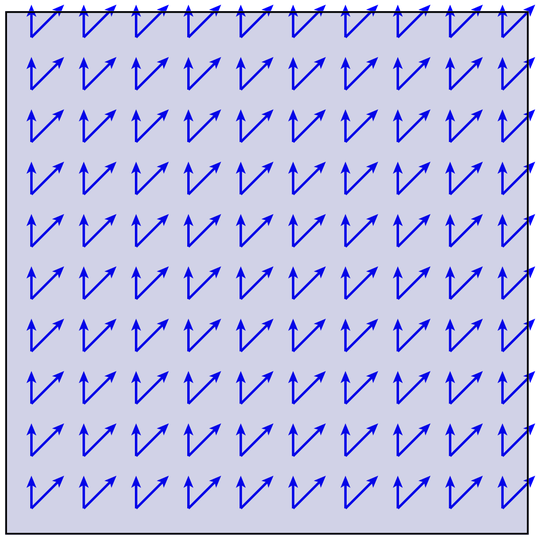
- Can use induced metric to translate between vector fields and 1-forms:

$$X^\flat(Y) := g(X,Y) \qquad g(\alpha^\sharp, Y) := \alpha(Y)$$

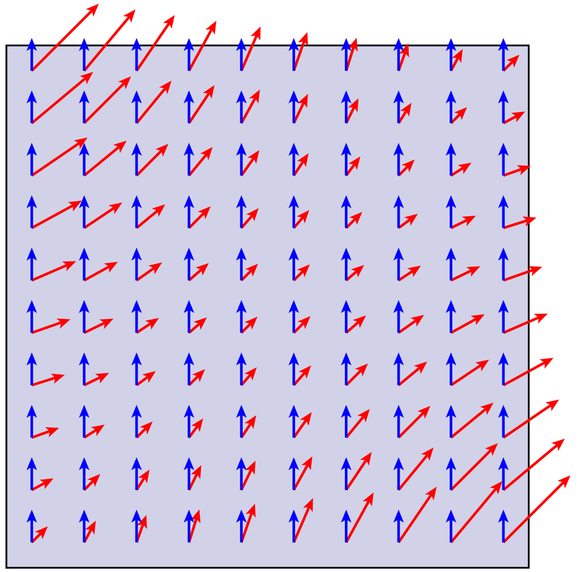
- No longer just a trivial "transpose" (as in Euclidean R^n)
- E.g., flat correctly encodes inner product on surface



$df(X) \cdot df(Y)$



$X \cdot Y \neq df(X) \cdot df(Y)$



$X^\flat(Y) = df(X) \cdot df(Y)$