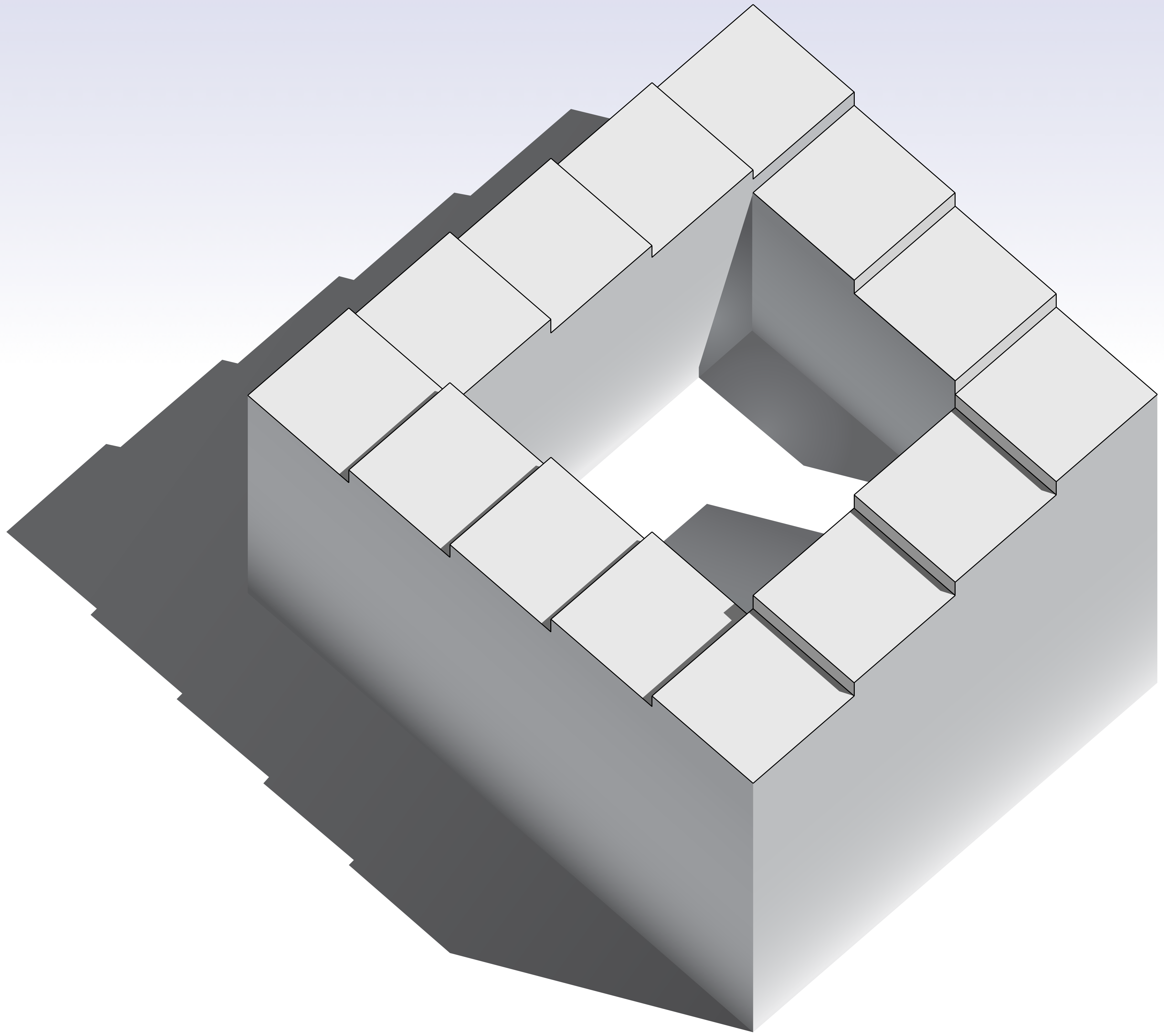
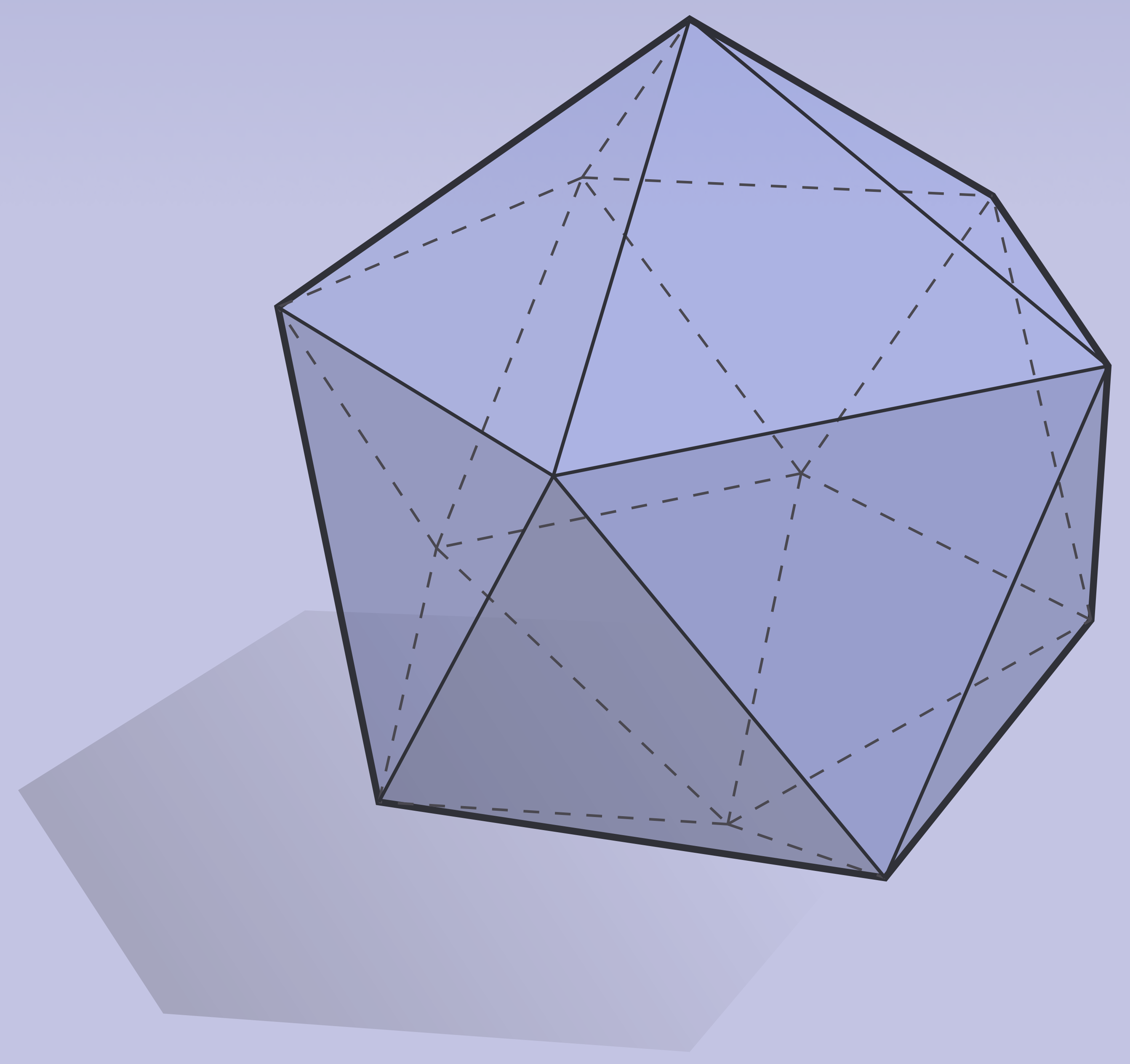


Discrete Differential Geometry: An Applied Introduction

Keenan Crane • CMU 15-458/858

LECTURE 11:
DISCRETE CURVES

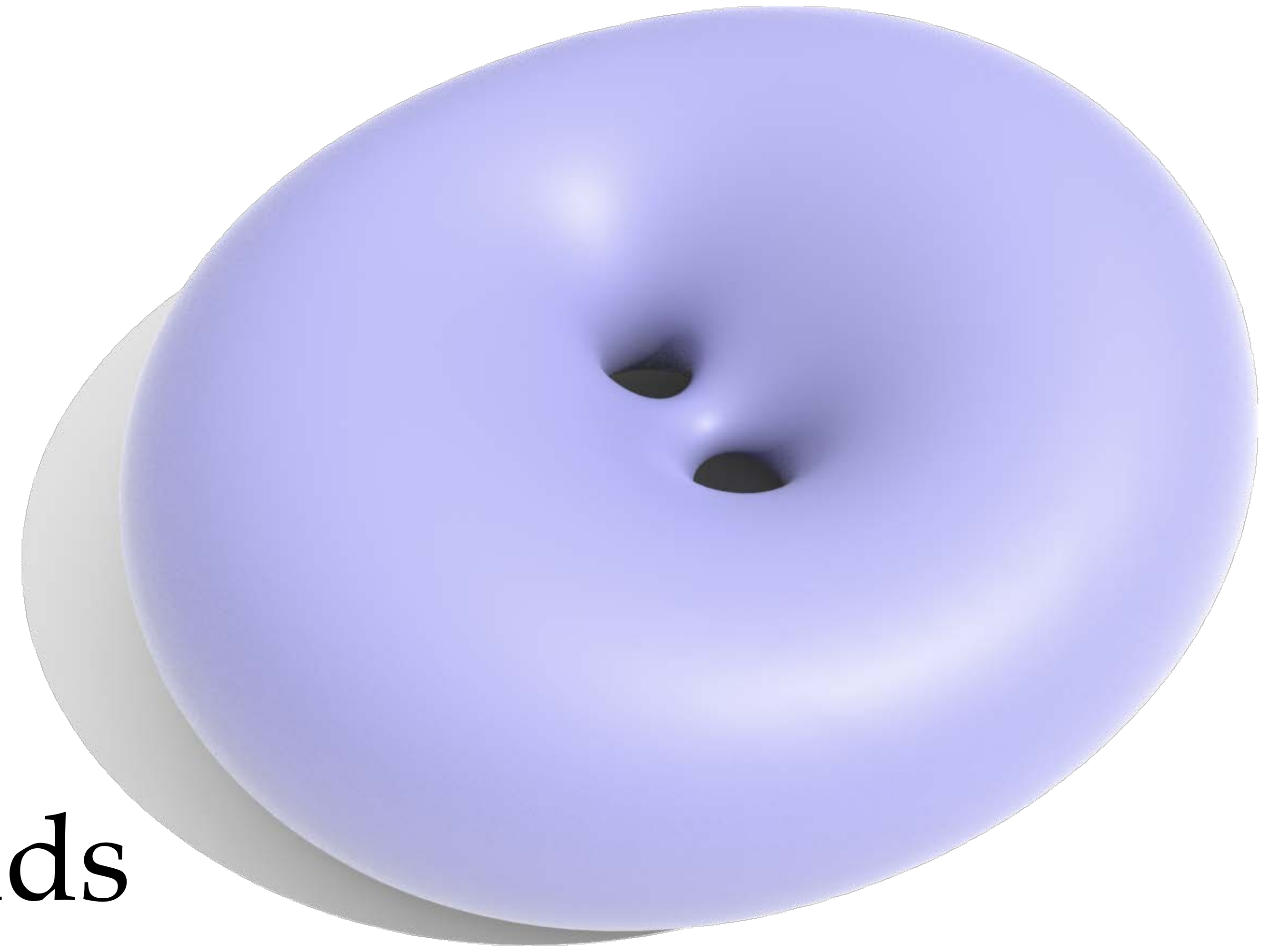


DISCRETE DIFFERENTIAL GEOMETRY: AN APPLIED INTRODUCTION

Keenan Crane • CMU 15-458/858

Curves, Surfaces, and Volumes

- In general, differential geometry studies n-dimensional manifolds; we'll focus mostly on low dimensions: curves (n=1), surfaces (n=2), and volumes (n=3)
- Why? Geometry we encounter in "every day life" (Common in applications!)
- Low-dimensional manifolds are not baby stuff! :-)
  - ***n*=1:** unknot recognition (open as of July 2017)
  - ***n*=2:** Willmore conjecture (2012 for genus 1)
  - ***n*=3:** Geometrization conjecture (2003, $1 million)

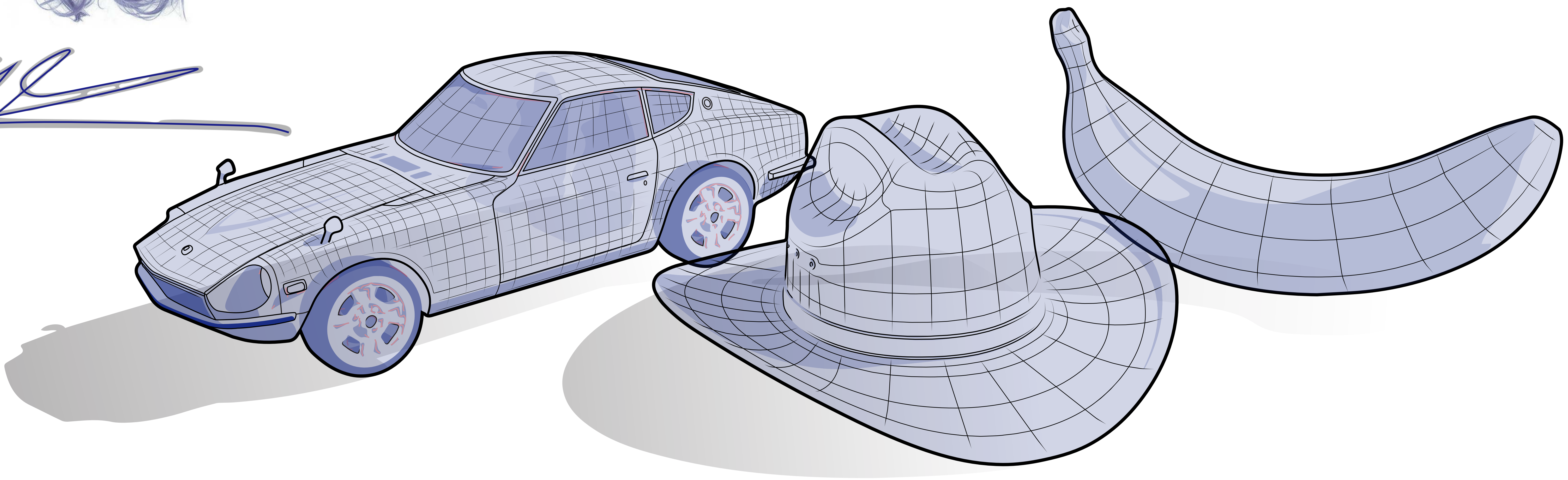


- Serious intuition gained by studying low-dimensional manifolds
- Conversely, problems involving very high-dimensional manifolds (e.g., statistics/ machine learning) involve less "deep" geometry than you might imagine!
  - *fiber bundles, Lie groups, curvature flows, spinors, symplectic structure, ...*
- Moreover... curves and surfaces are beautiful! (And sometimes boring for large n…)

Curves & Surfaces

- Much of the geometry we encounter in life well-described by *curves* and *surfaces**

(Curves)

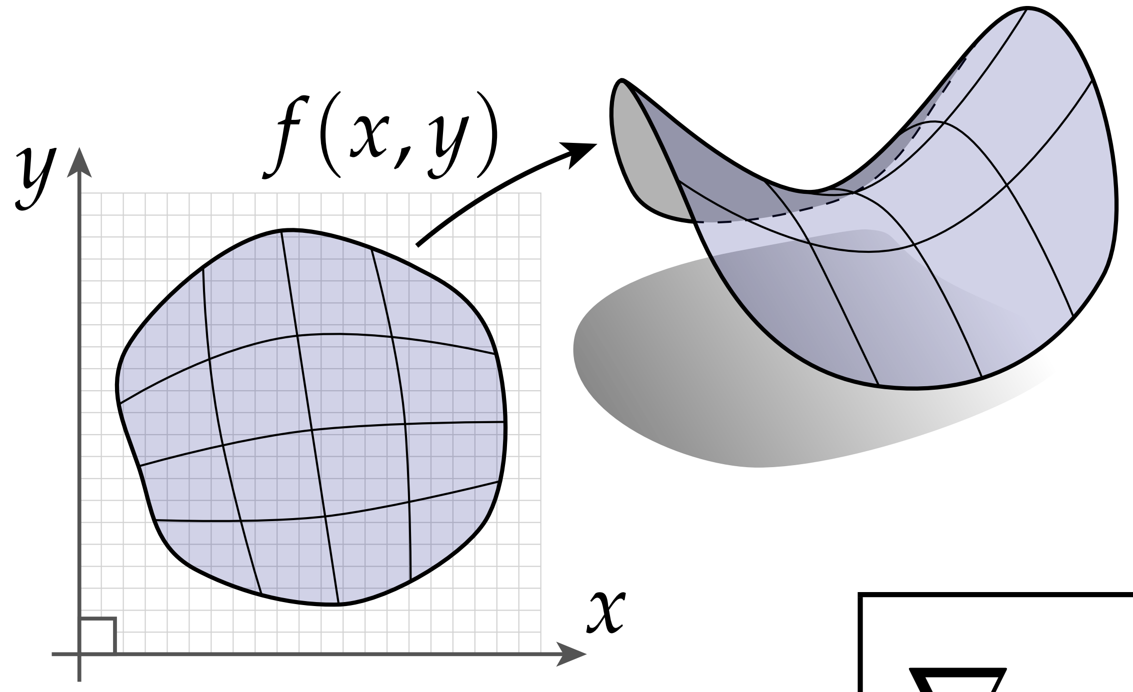


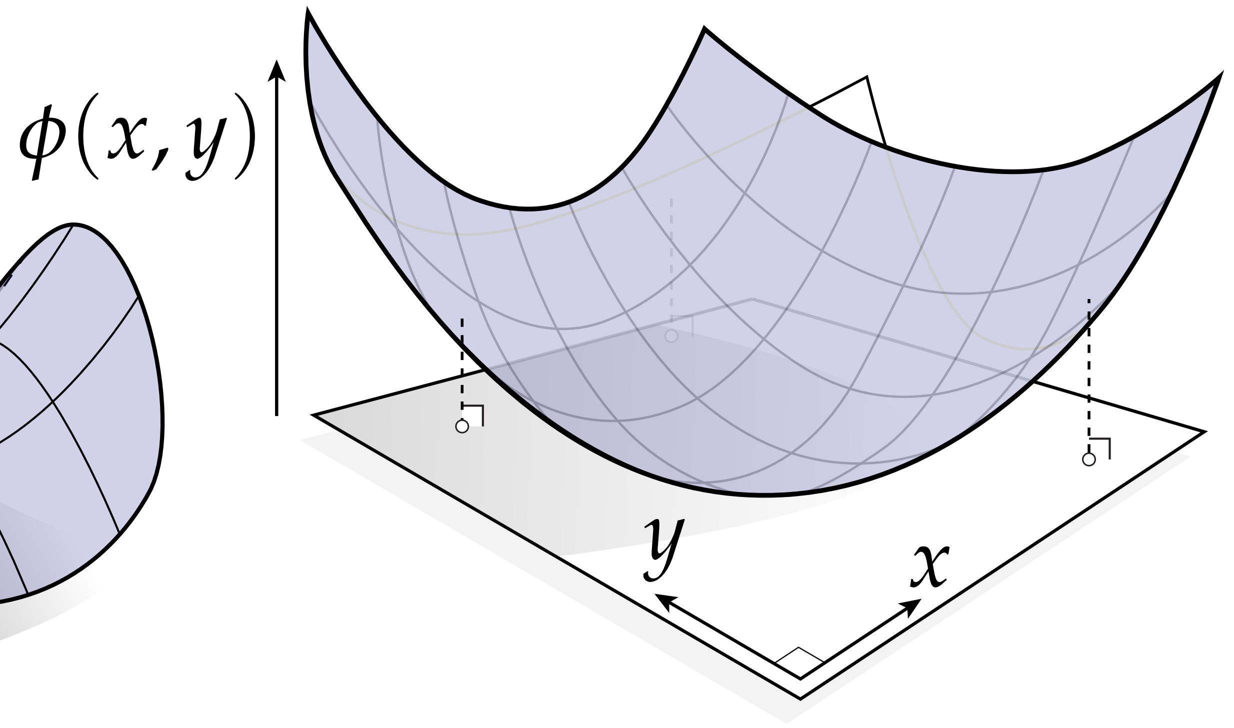
*Or solids… but the boundary of a solid is a surface!

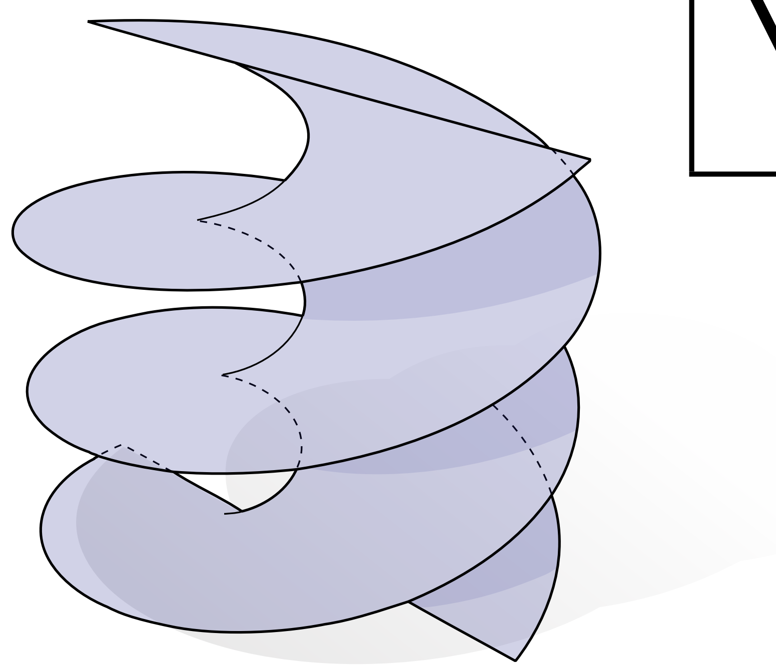
(Surfaces)

Smooth Descriptions of Curves & Surfaces

- Many ways to express the geometry of a curve or surface:
  - height function over tangent plane
  - local parameterization
  - Christoffel symbols — coordinates/indices
  - **differential forms** — "coordinate free"
  - moving frames — change in *adapted frame*
  - Riemann surfaces (*local*); Quaternionic functions (*global*)
- People can get very religious about these different "dialects"... best to be multilingual!
- We'll dive deep into one description (**differential forms**) and touch on others





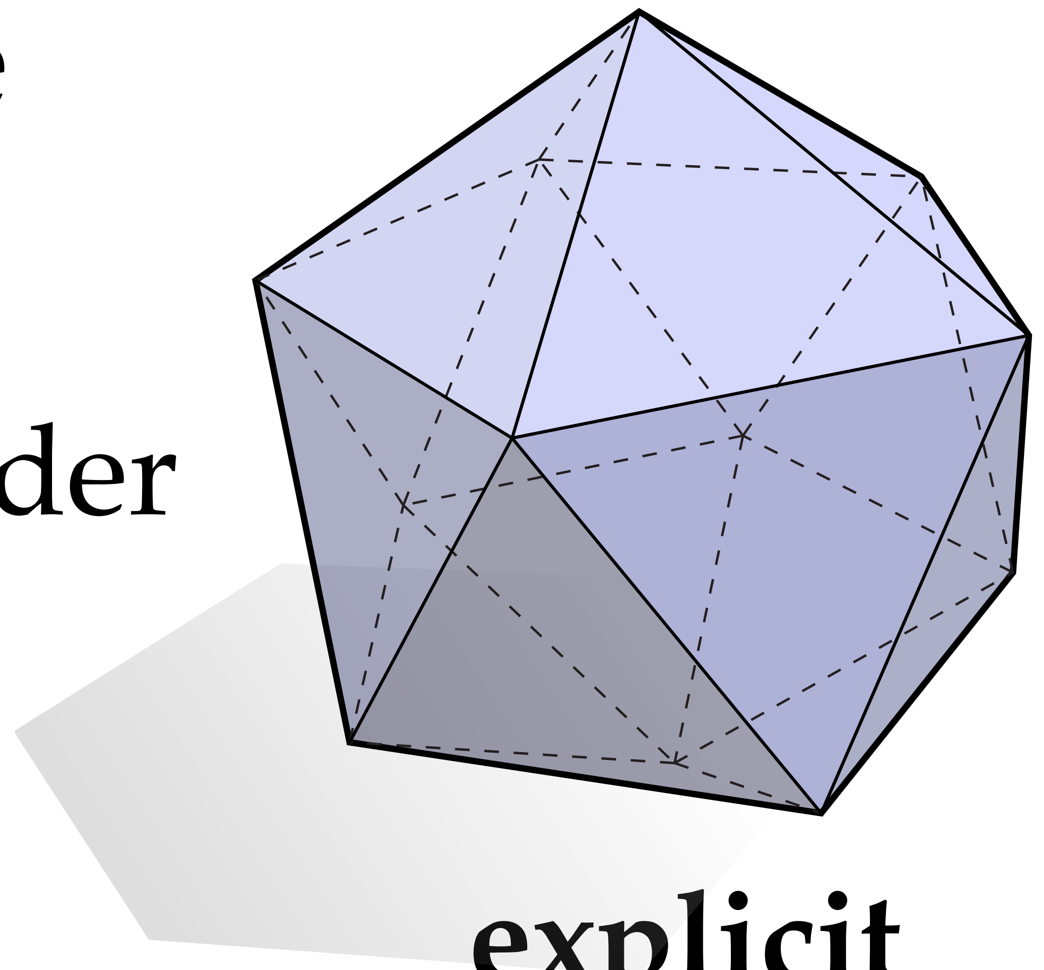


$$\nabla_i e_j = \Gamma_{ij}^k e_k$$

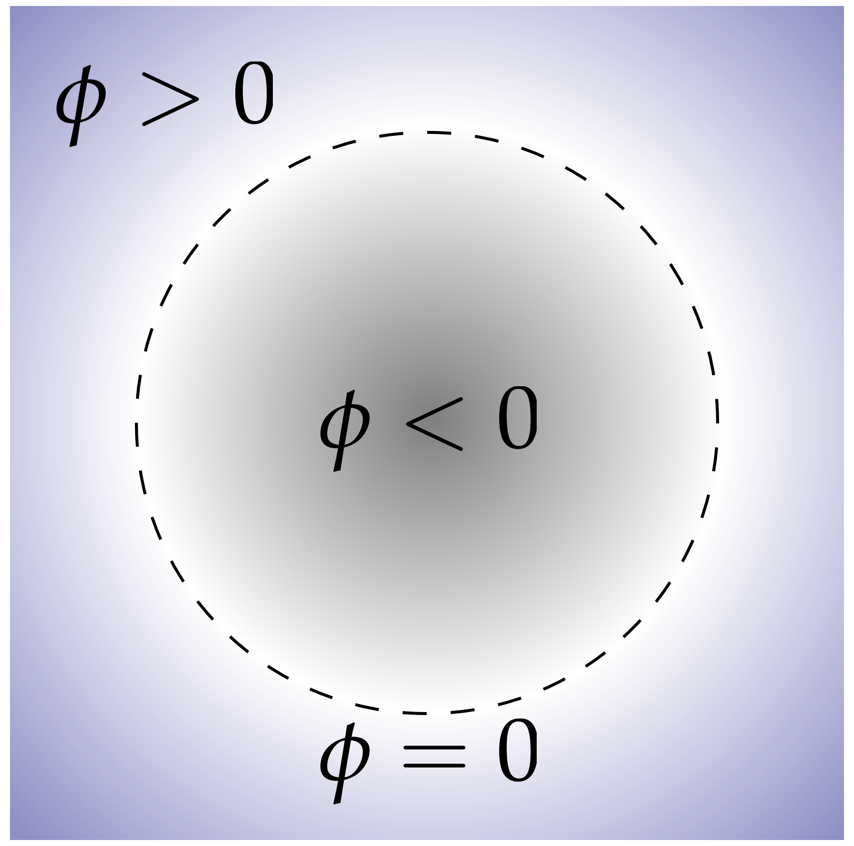
$$\begin{aligned} d\sigma &= \omega \wedge \sigma \\ d\omega &= \omega \wedge \omega \end{aligned}$$

Discrete Descriptions of Curves & Surfaces

- Also *many* ways to discretize a surface
- For instance:
  - **implicit** — *e.g.*, zero set of scalar function on a grid
    - good for changing topology, high accuracy
    - expensive to store/adaptivity is harder
    - hard to solve sophisticated equations *on* surface
  - **explicit** — *e.g.*, polygonal surface mesh
    - changing topology, high-order continuity is harder
    - cheaper to store / adaptivity is much easier
    - more mature tools for equations *on* surfaces
- Don't be "religious"; use the right tool for the job!



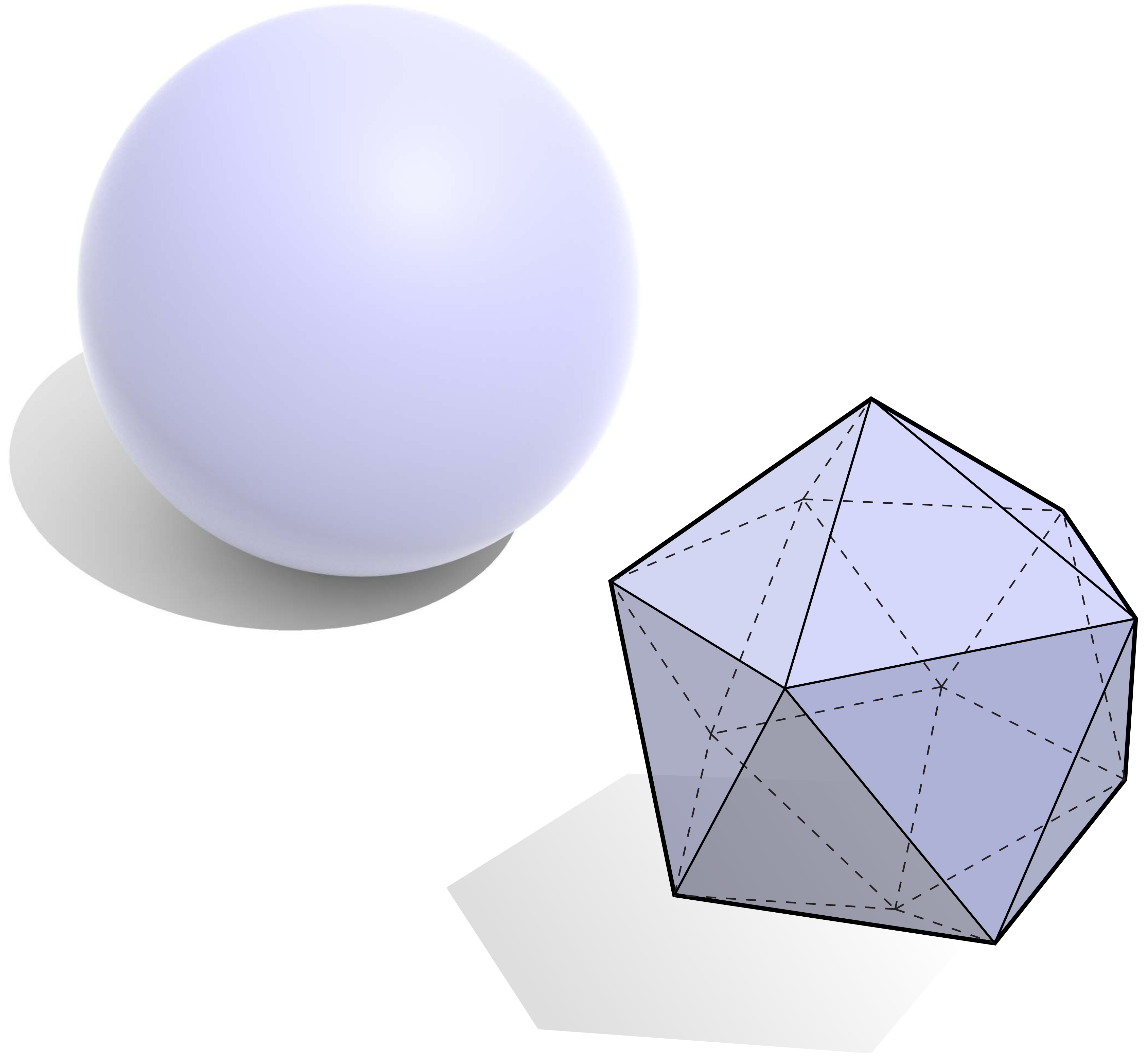
explicit



implicit

Curves & Surfaces — Overview

- **Goal:** understand curves & surfaces from complementary smooth and discrete points of view.
- **<u>Smooth setting:</u>**
  - express geometry via differential forms
  - will first need to think about *vector-valued* forms
- **<u>Discrete setting:</u>**
  - use explicit mesh as domain
  - express geometry via discrete differential forms

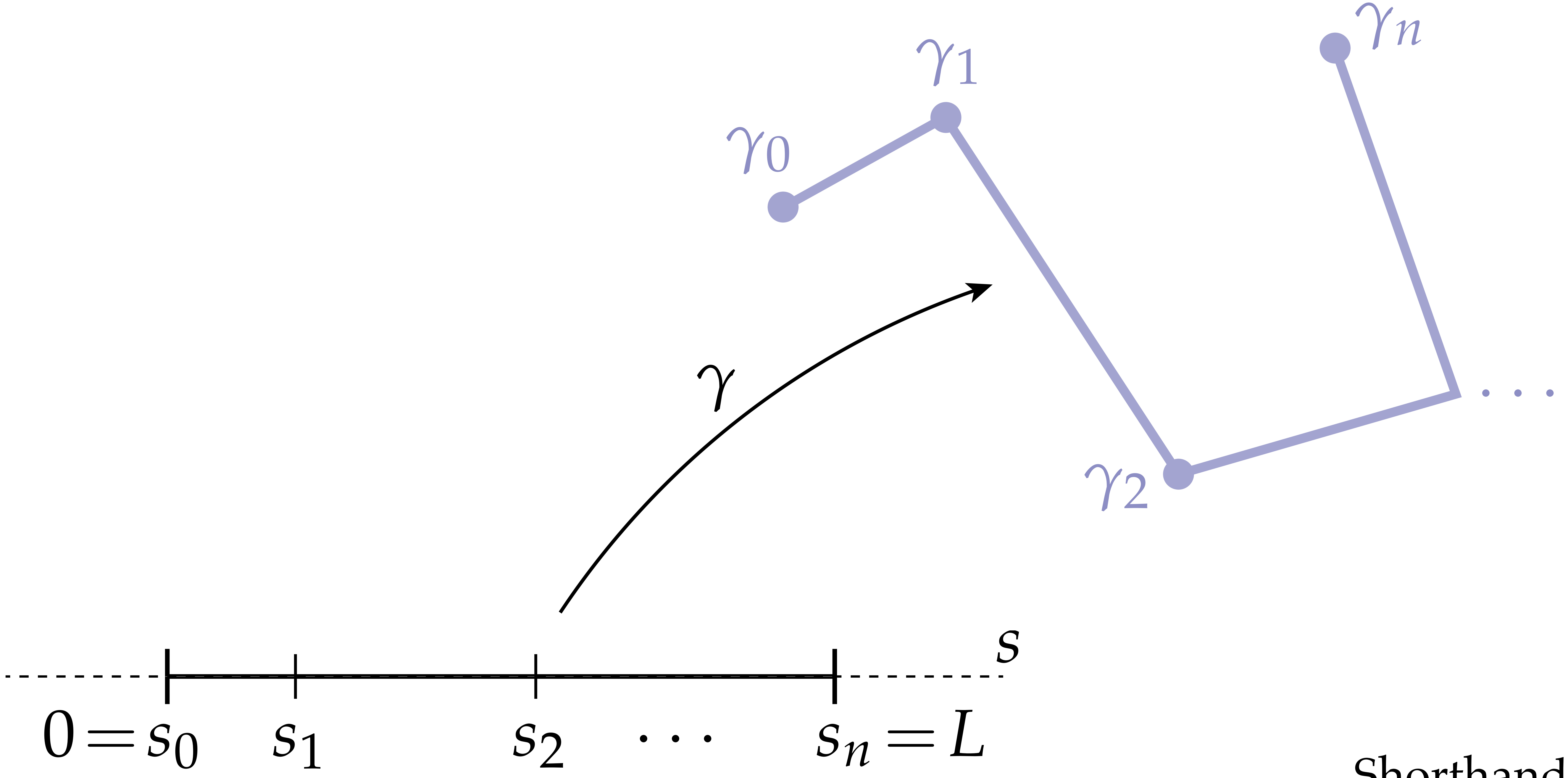


- **Payoff:** will become very easy to switch back & forth between smooth setting (*scribbling in a notebook*) and discrete setting (*running algorithms on real data!*)

Discrete Curves

Discrete Curves in the Plane

- We'll define a **discrete curve** as a *piecewise linear* parameterized curve, *i.e.*, a sequence of points connected by straight line segments:

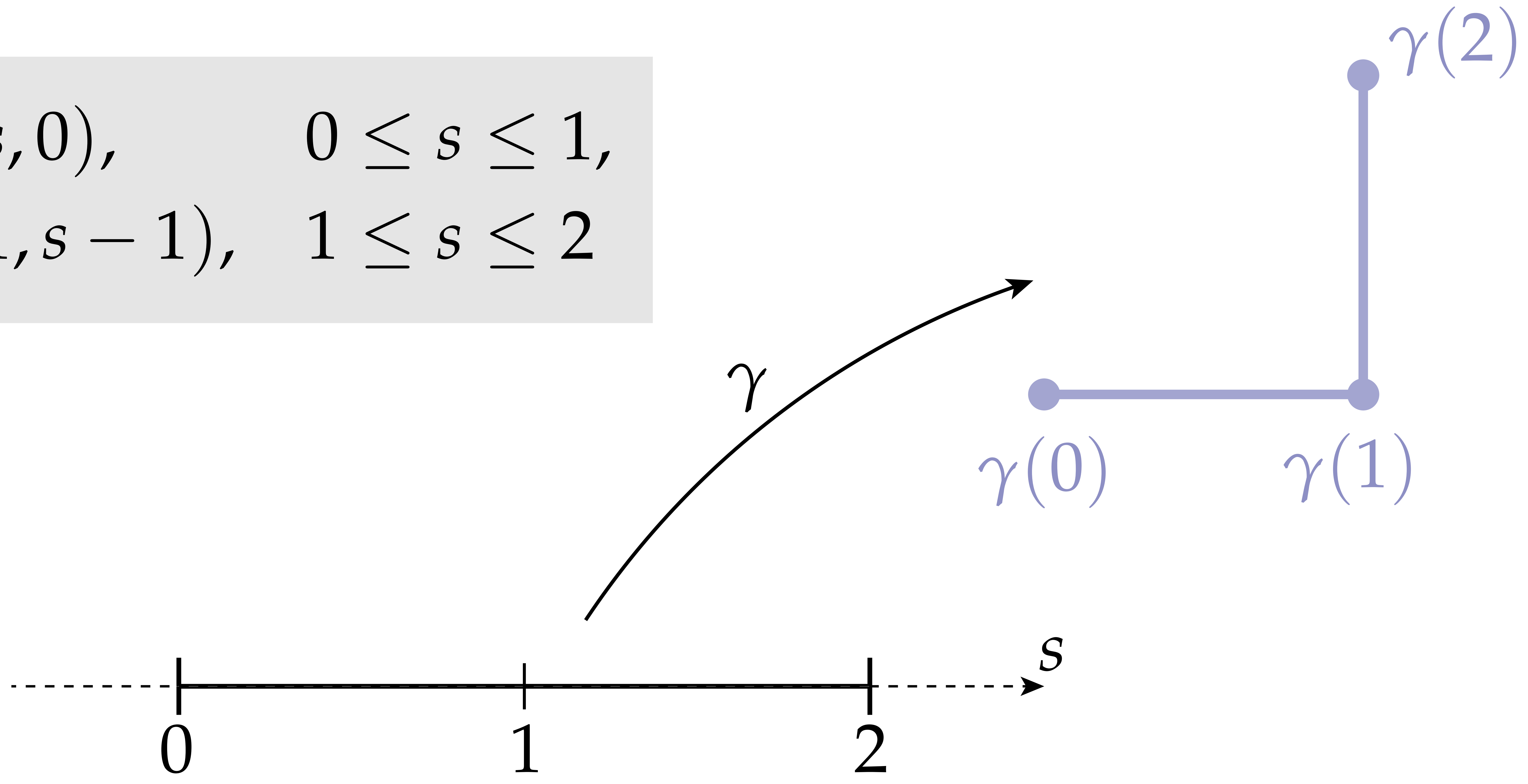


Shorthand: $\gamma_i := \gamma(s_i)$

Discrete Curves in the Plane—Example

- A simple example is a curve comprised of two segments:

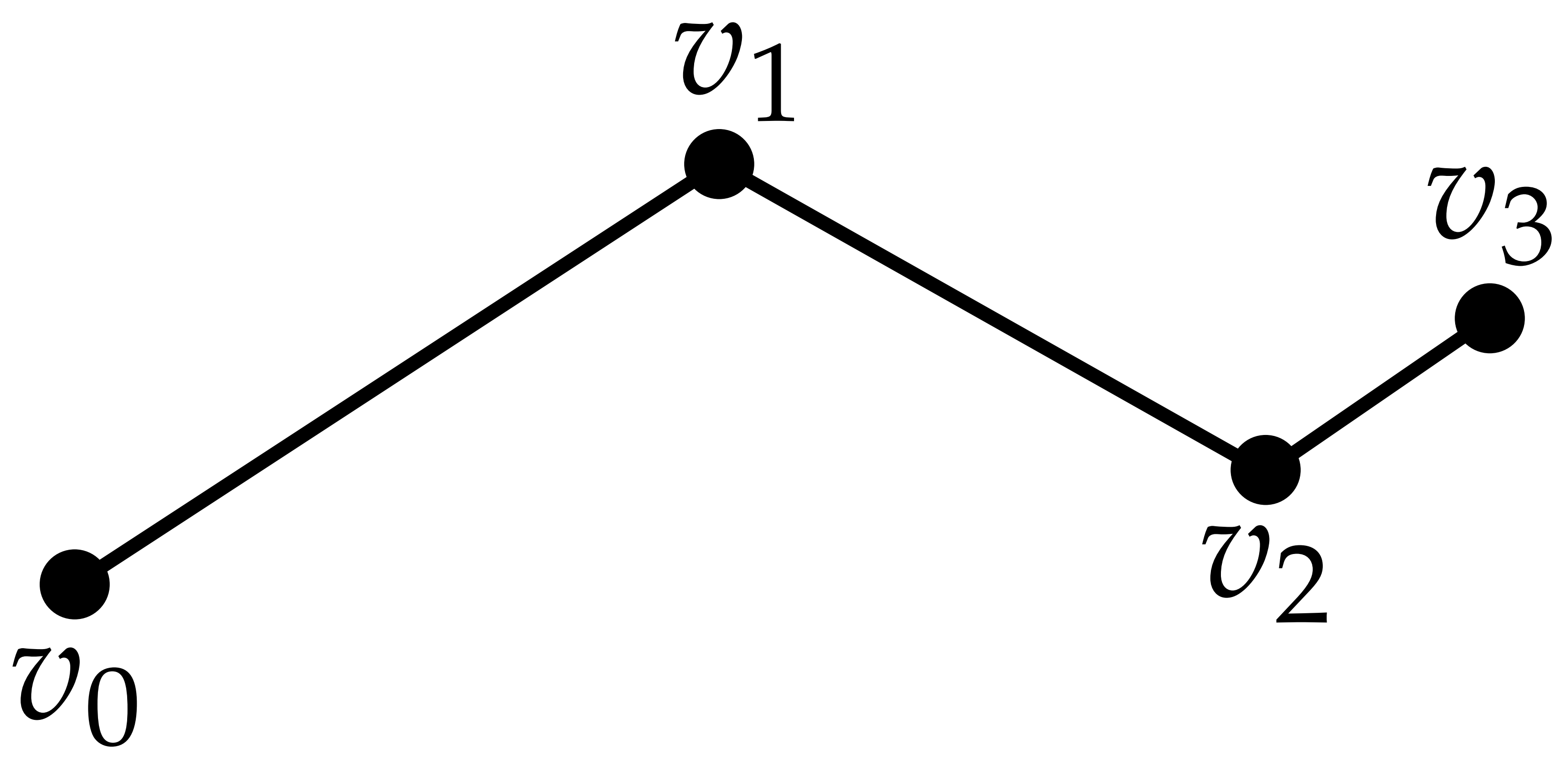
$$\gamma(s) := \begin{cases} (s, 0), & 0 \leq s \leq 1, \\ (1, s-1), & 1 \leq s \leq 2 \end{cases}$$



Discrete Curves and Discrete Differential Forms

- Equivalently, a discrete curve is determined by a discrete, R^n-valued 0-form on a manifold simplicial 1-complex

- The 0-form values give the location of the vertices; interpolation by Whitney bases (hat functions) gives the map from each edge to R^n

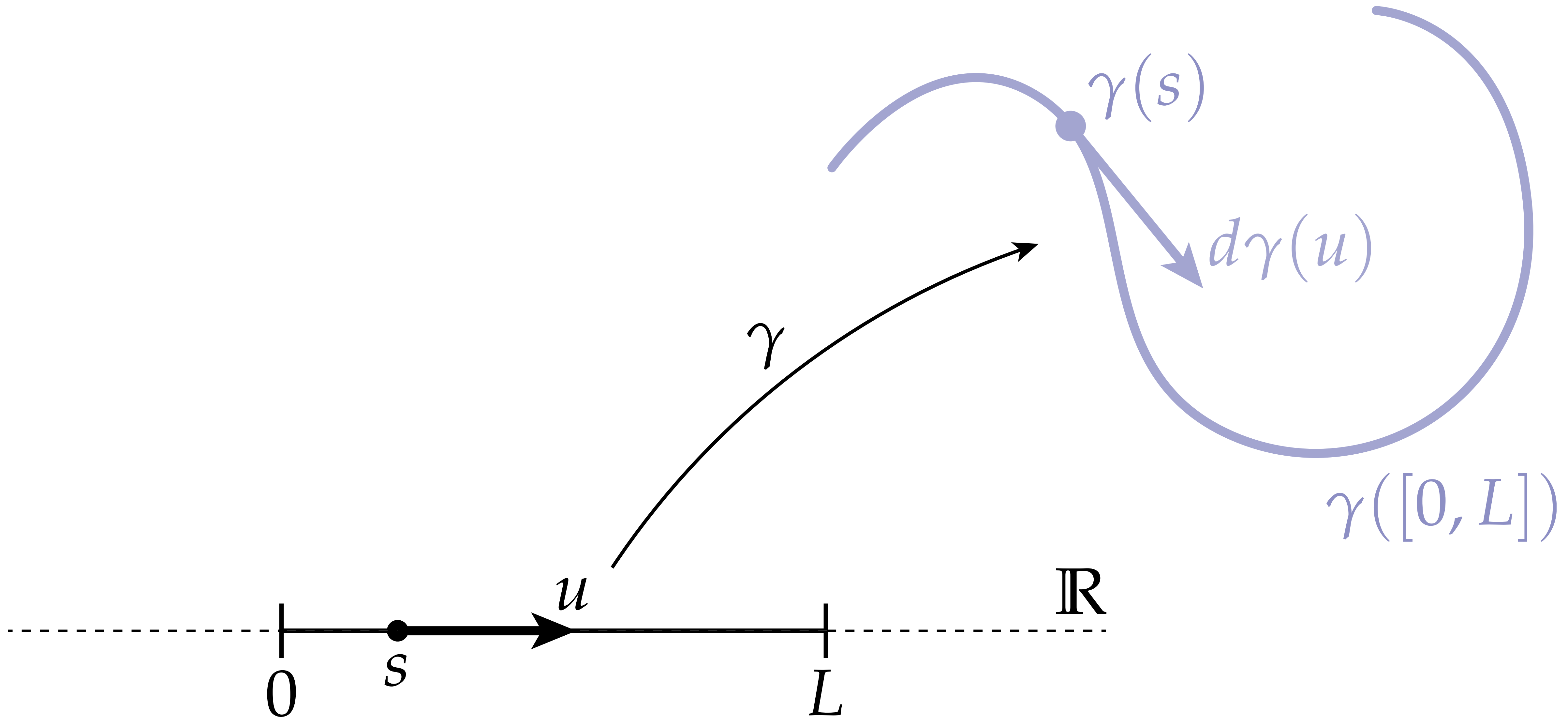


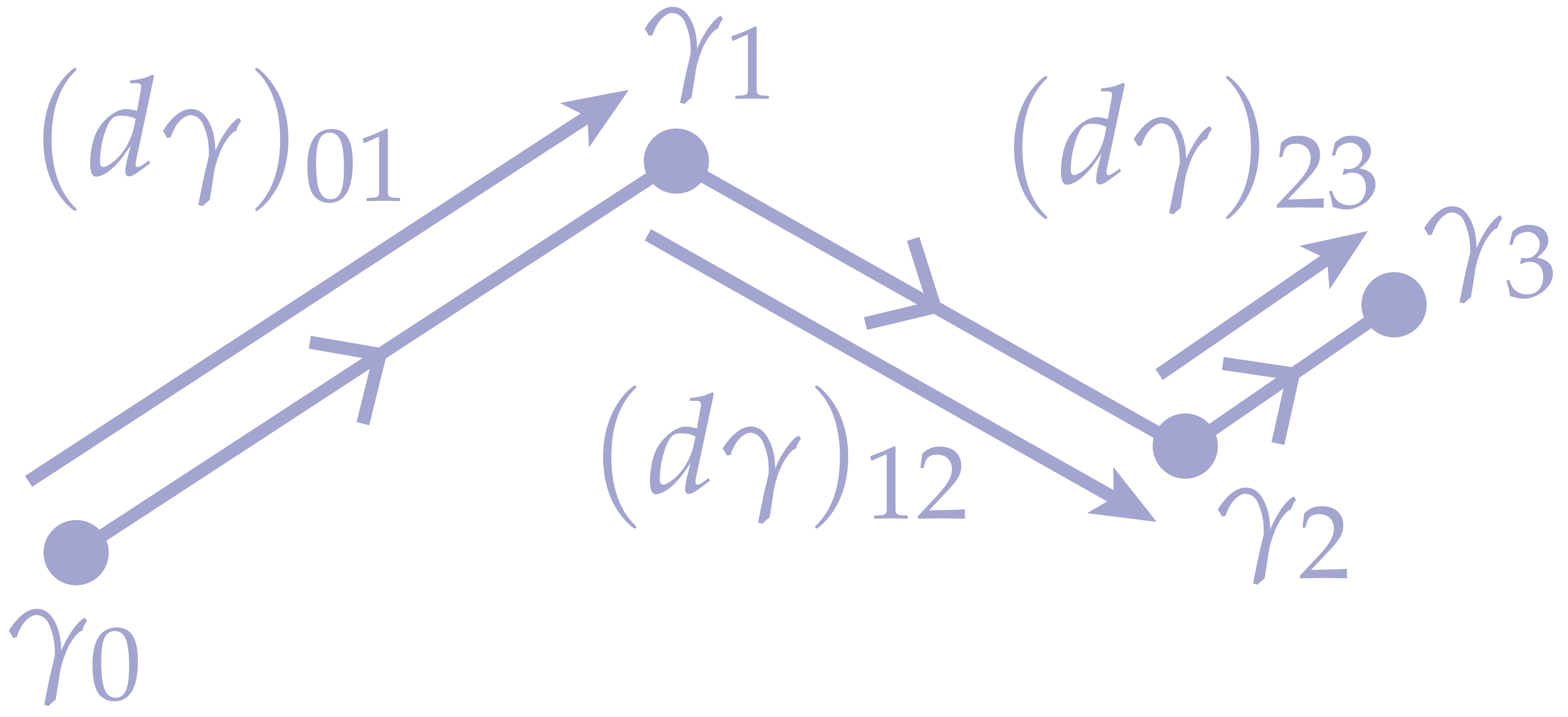
$$K = \{ (v_0,v_1), (v_1,v_2), (v_2,v_3), (v_0), (v_1), (v_2), (v_3), \varnothing \}$$

$$\begin{aligned} \gamma(v_0) &= (33, 66) \\ \gamma(v_1) &= (79, 36) \\ \gamma(v_2) &= (118, 58) \\ \gamma(v_3) &= (134, 47) \end{aligned}$$

Differential of a Discrete Curve

- We can now directly translate statements about **smooth** curves expressed via **smooth** exterior calculus into statements about **discrete** curves expressed using **discrete** exterior calculus
- Simple example: the *differential* just becomes the edge vectors:

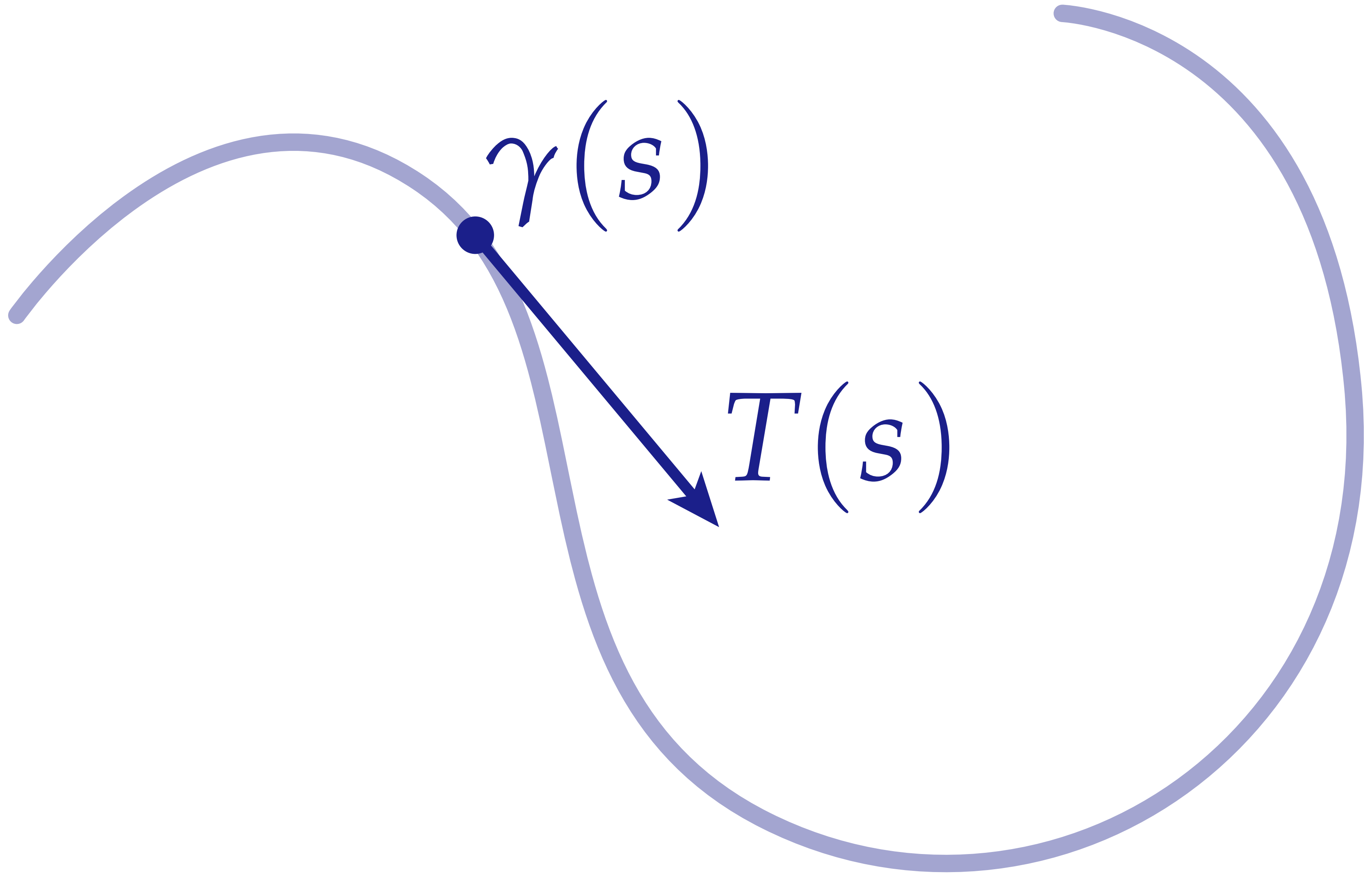


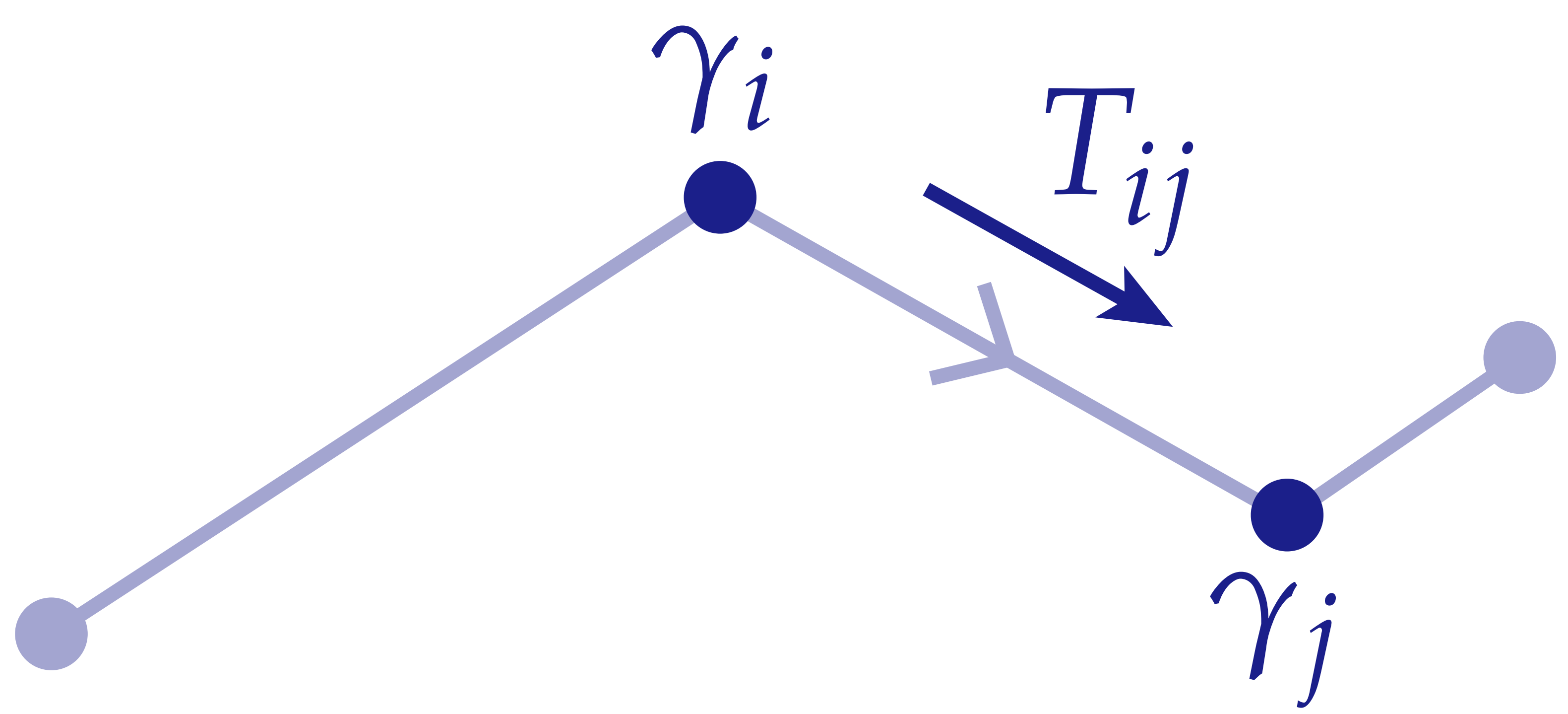


$$(d\gamma)_{ij} = \gamma_j - \gamma_i$$

Discrete Tangent

- As in smooth setting, can simply normalize differential to obtain tangents, yielding a vector per edge*





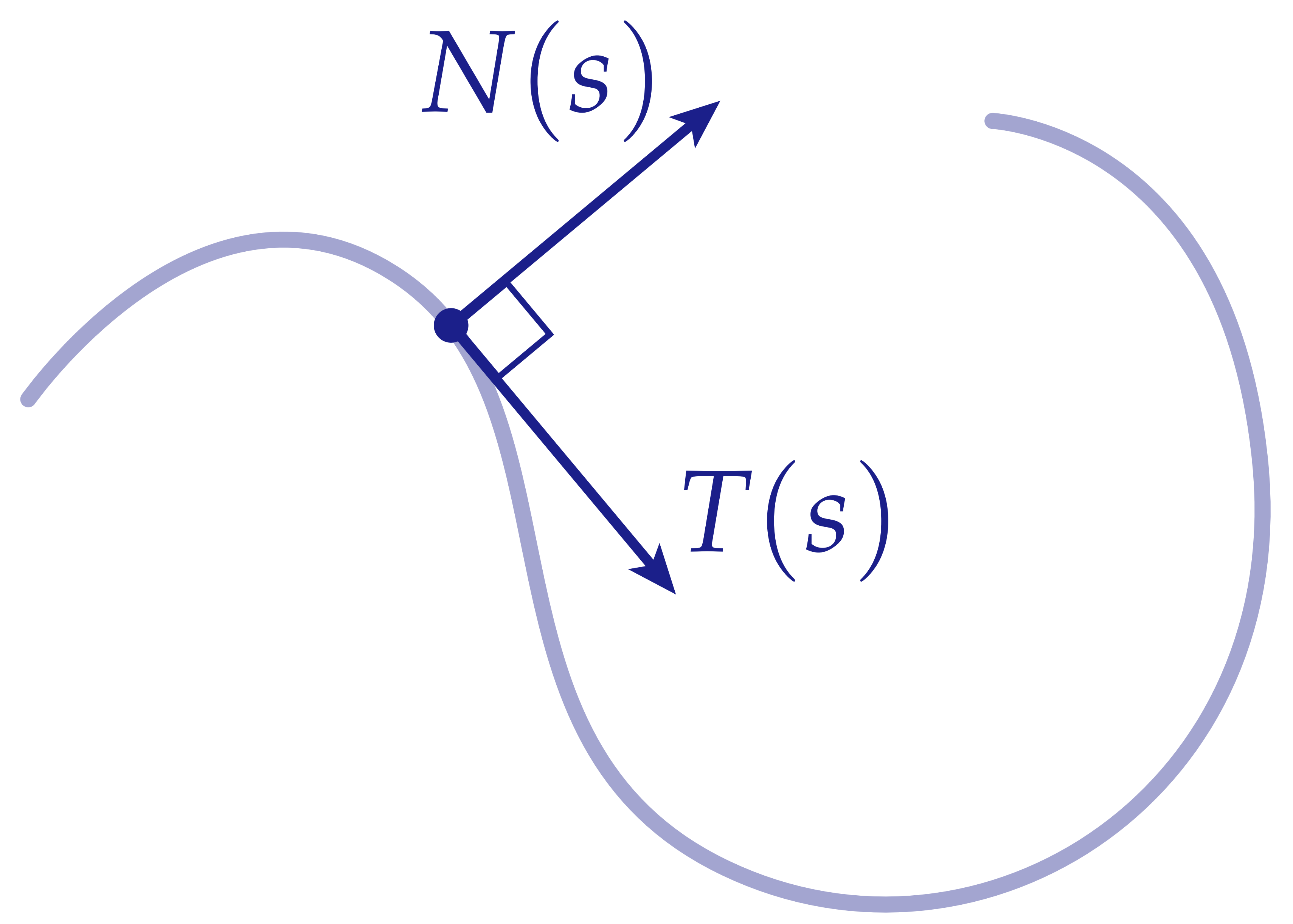
$$T(s) := d\gamma(\tfrac{d}{ds})/|d\gamma(\tfrac{d}{ds})|$$

$$T_{ij} := (d\gamma)_{ij}/|(d\gamma)_{ij}|$$

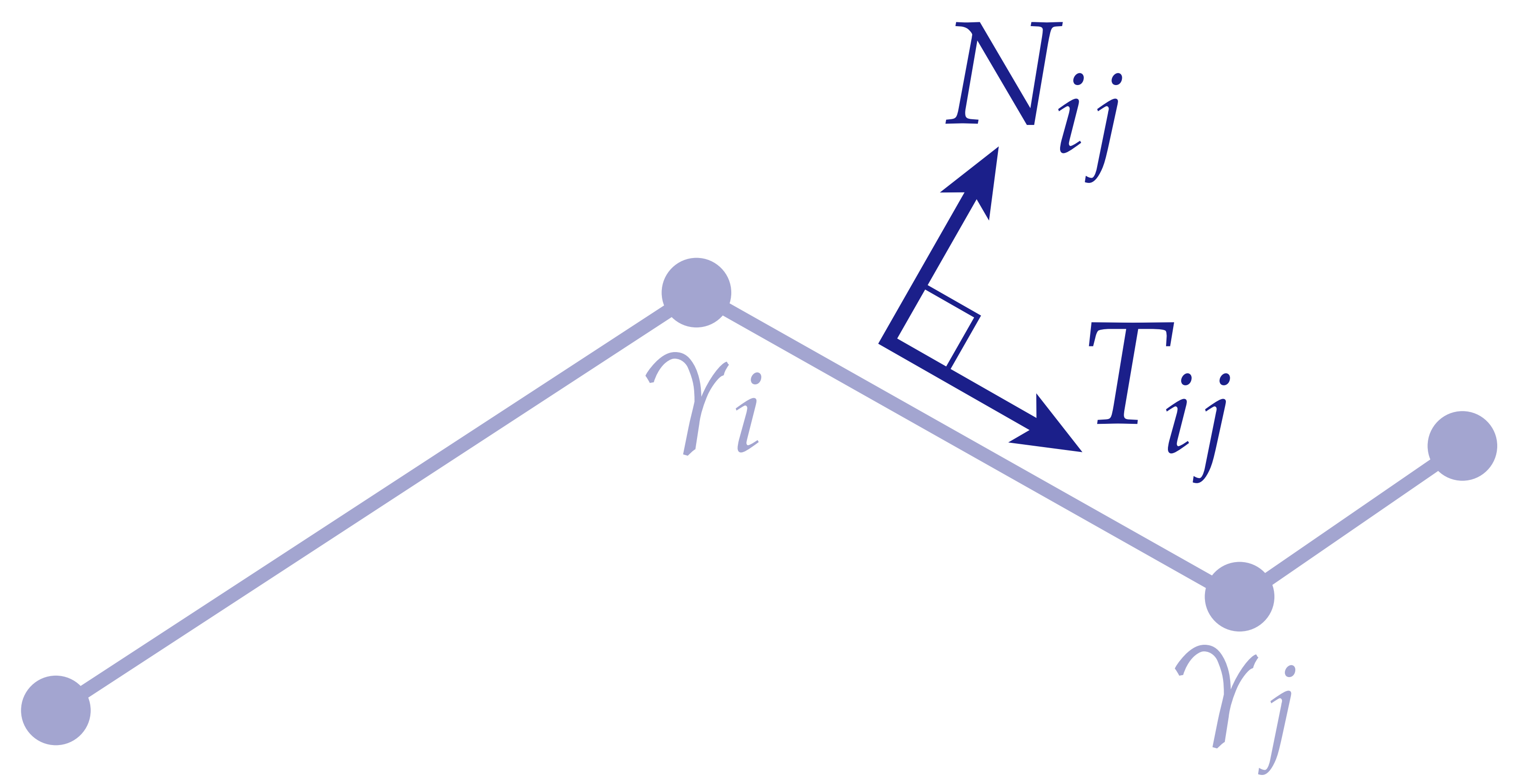
*And no definition of the tangent at vertices!

Discrete Normal

- As in the smooth setting, we can express the (discrete) normals of a planar curve as a 90-degree rotation of the (discrete) tangent:



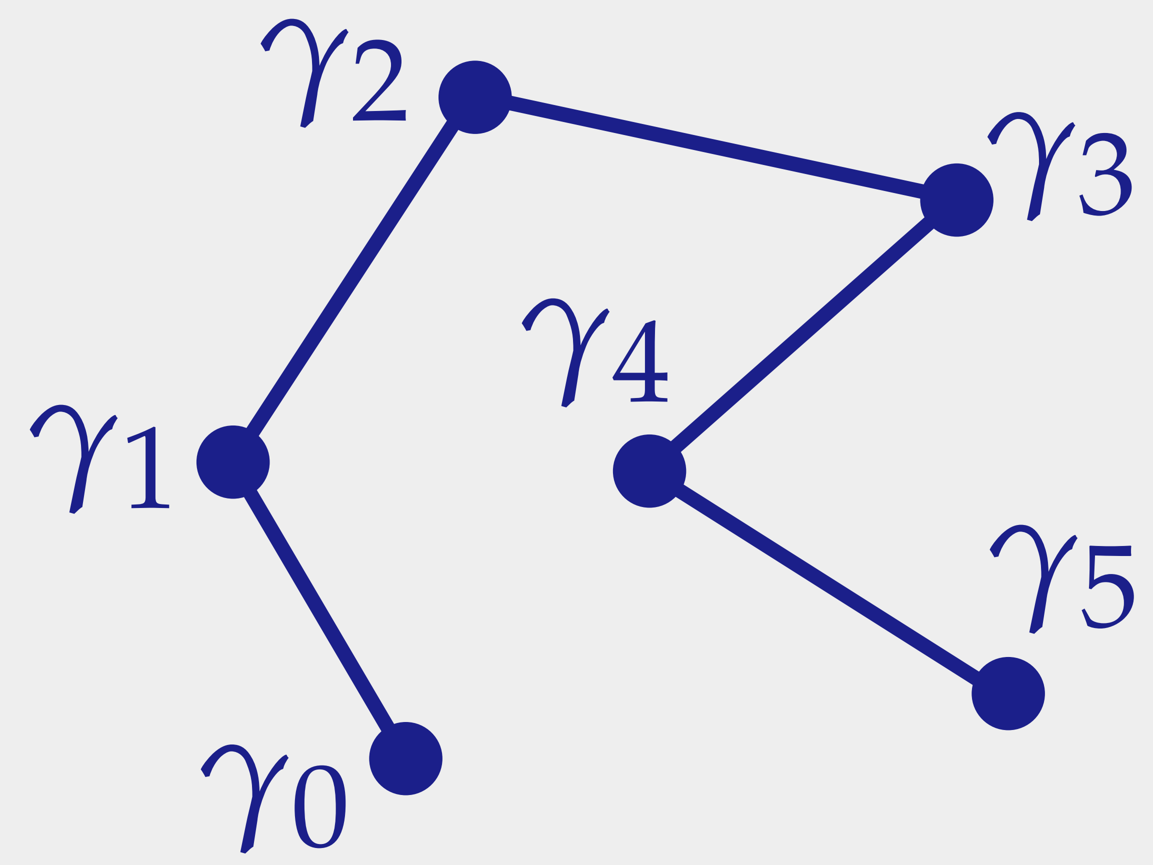
$$N(s) = \mathcal{J}T(s)$$



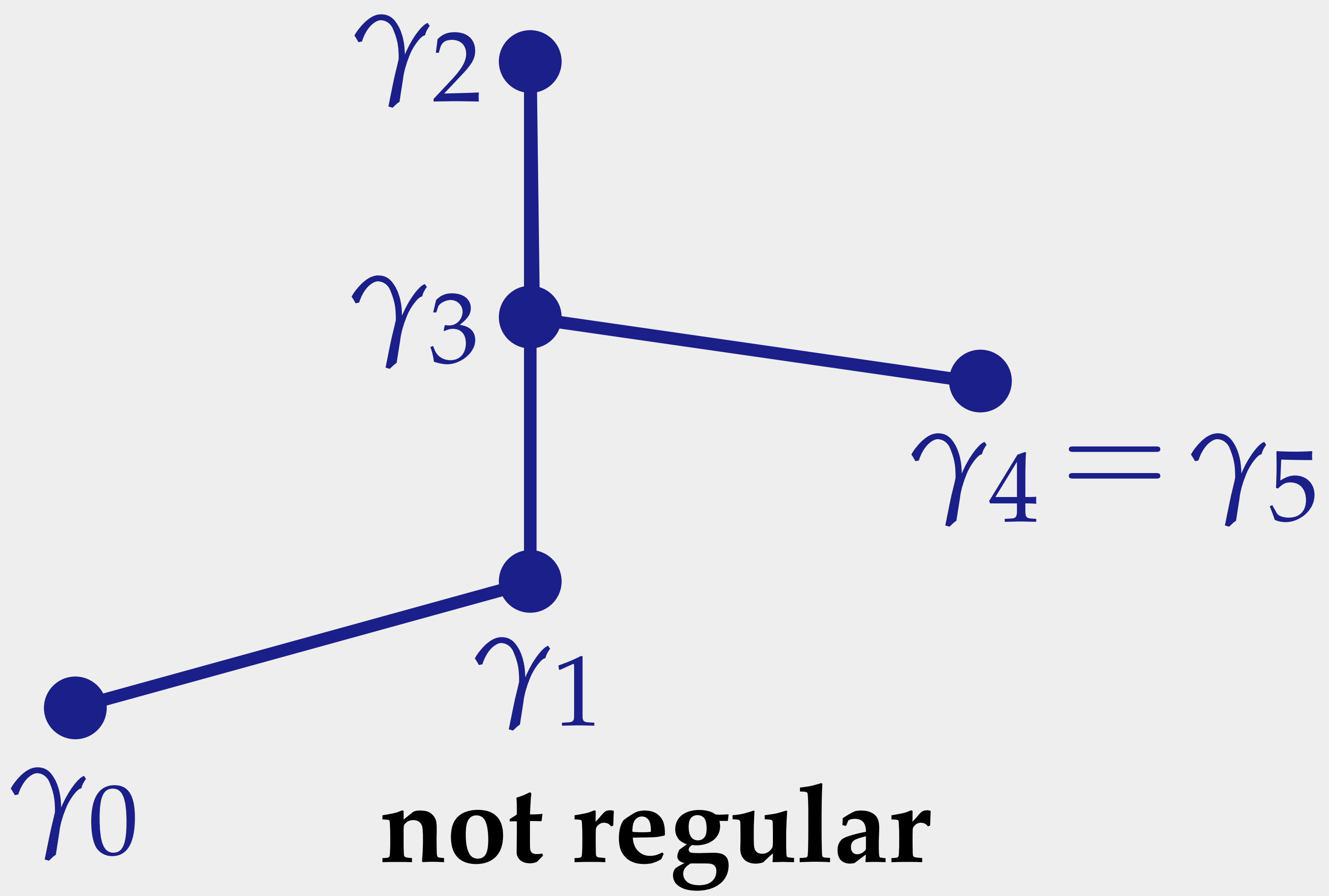
$$N_{ij} = \mathcal{J}T_{ij}$$

Regular Discrete Curve / Discrete Immersion

- Recall that a smooth curve is *regular* if its differential is nonzero; this condition helps avoid "bad behavior" like sharp cusps
- For a discrete curve, a nonzero differential merely prevents zero edge lengths; need something stronger to get "nice" curves
- In particular, a *regular discrete curve* or *discrete immersion* is a discrete curve that is a **locally injective map**
- Rules out zero edge lengths *and* zero angles



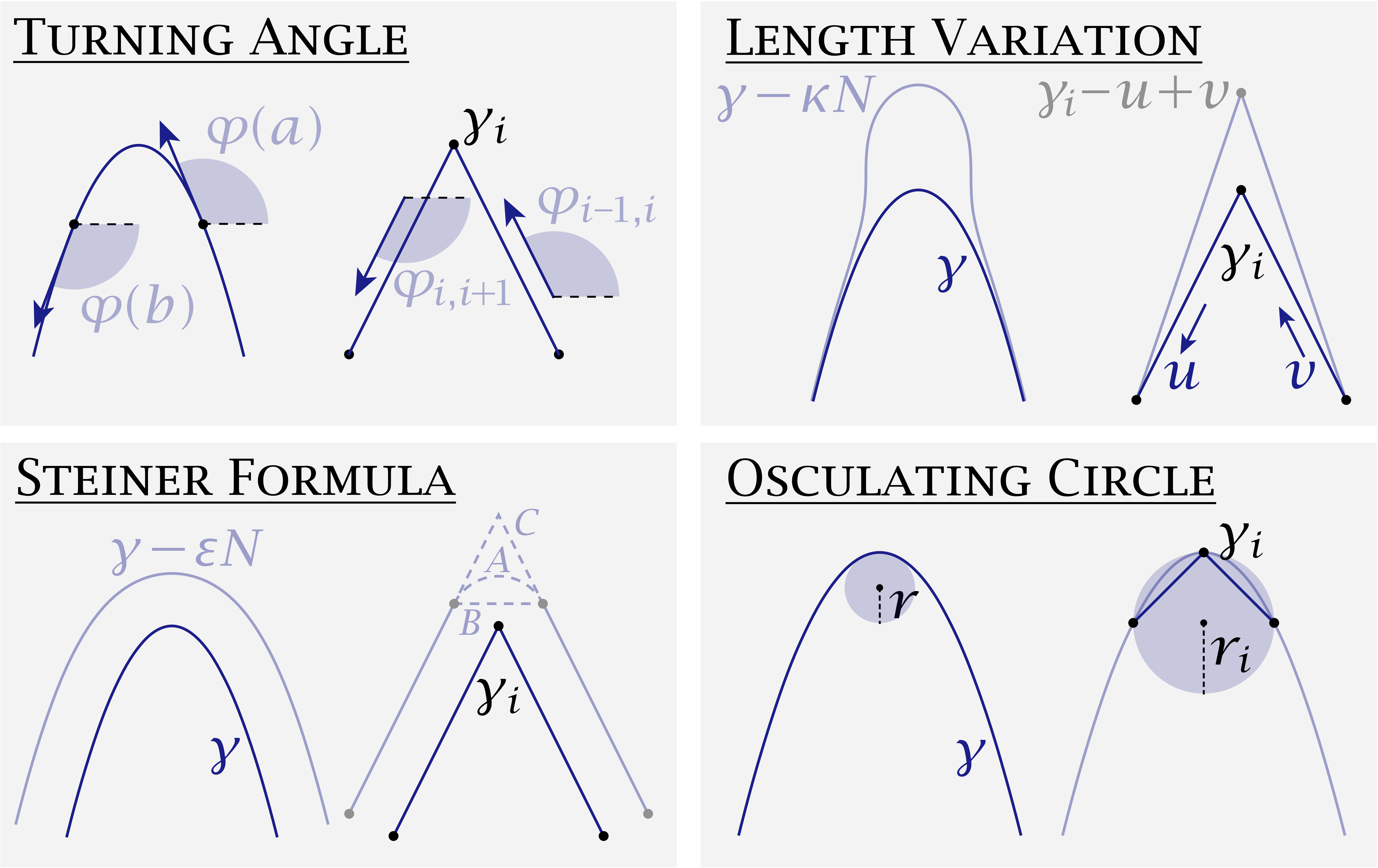
regular



not regular

Discrete Curvature

- For a regular discrete curve, discrete curvature has several definitions

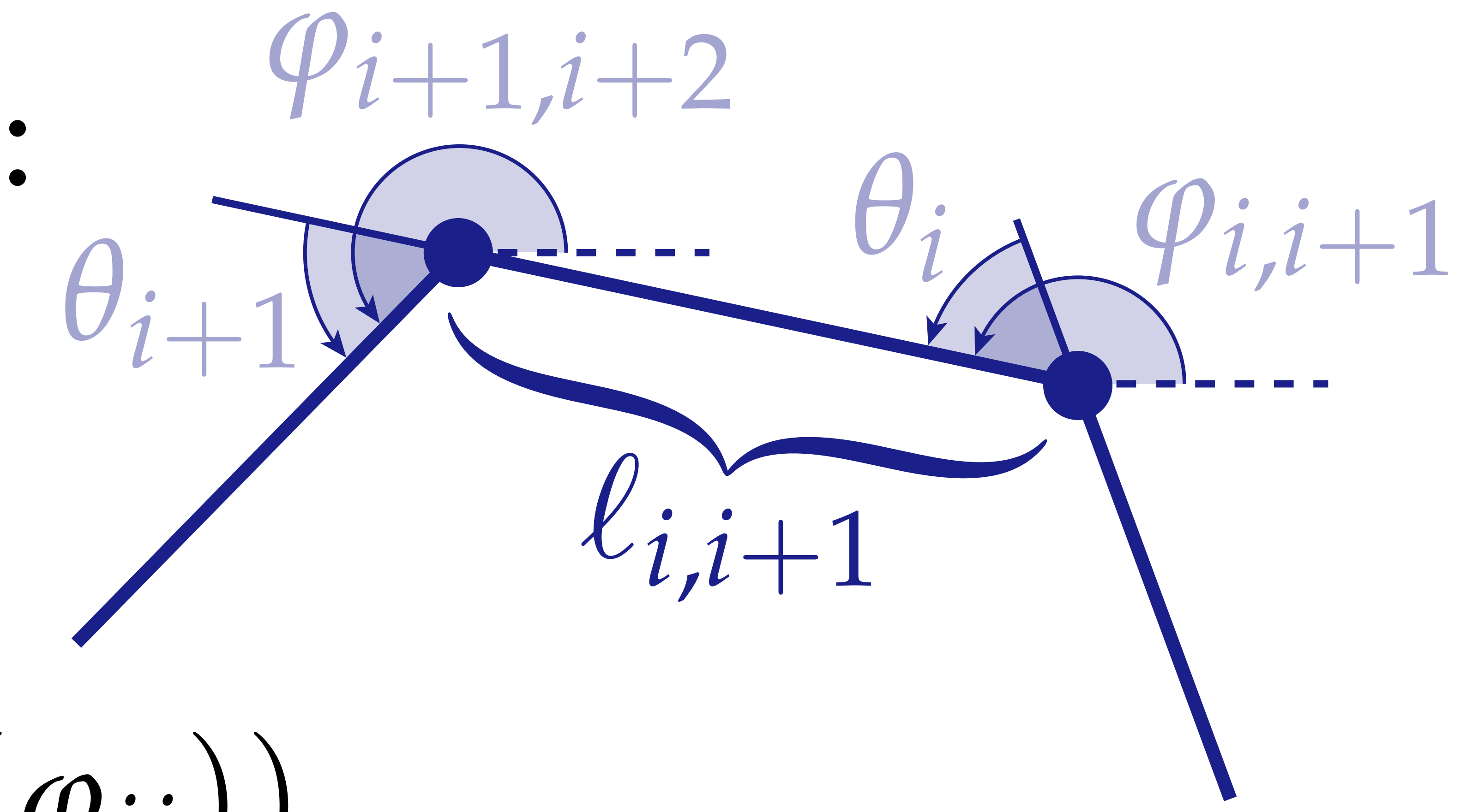


Fundamental Theorem of Discrete Plane Curves

Fact. Up to rigid motions, a regular discrete plane curve is uniquely determined by its edge lengths and turning angles.

Q: Given only this data, how can we recover the curve?

A: Mimic the procedure from the smooth setting:



Sum curvatures to get angles: $\varphi_{i,i+1} := \sum_{k=1}^{i} \theta_k$

Evaluate unit tangents: $T_{ij} := (\cos(\varphi_{ij}), \sin(\varphi_{ij}))$

Sum tangents to get curve: $\gamma_i := \sum_{k=1}^{i} \ell_{k,k+1} T_{k,k+1}$

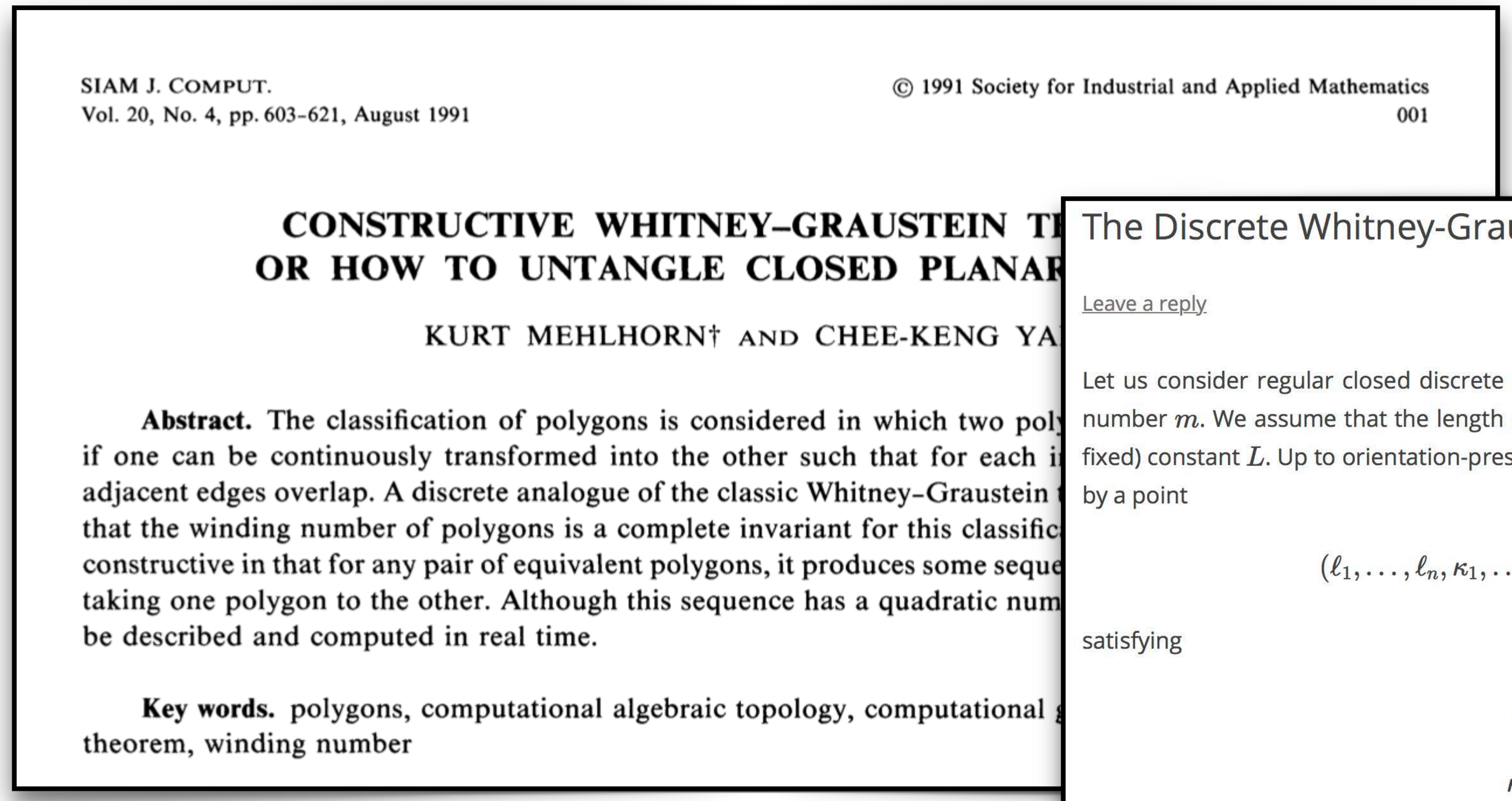
Q: Rigid motions?

Discrete Whitney Graustein

- If we adopt the definition of a discrete regular curve as one that is *locally injective*, then there is a discrete version of Whitney-Graustein that exactly mirrors the smooth one

- Has been carefully studied from several perspectives:
  - Constructive algorithm (case analysis) by Mehlhorn & Yap (1991)
  - Much simpler argument by Pinkall in terms of convex polyhedron: https://bit.ly/2BFtywA

- Both use powerful idea from (discrete) differential geometry: to find a "path" connecting two objects, find path from both objects to a canonical one, then compose… (uniformization, Delaunay, …)


SIAM J. COMPUT.
Vol. 20, No. 4, pp. 603–621, August 1991

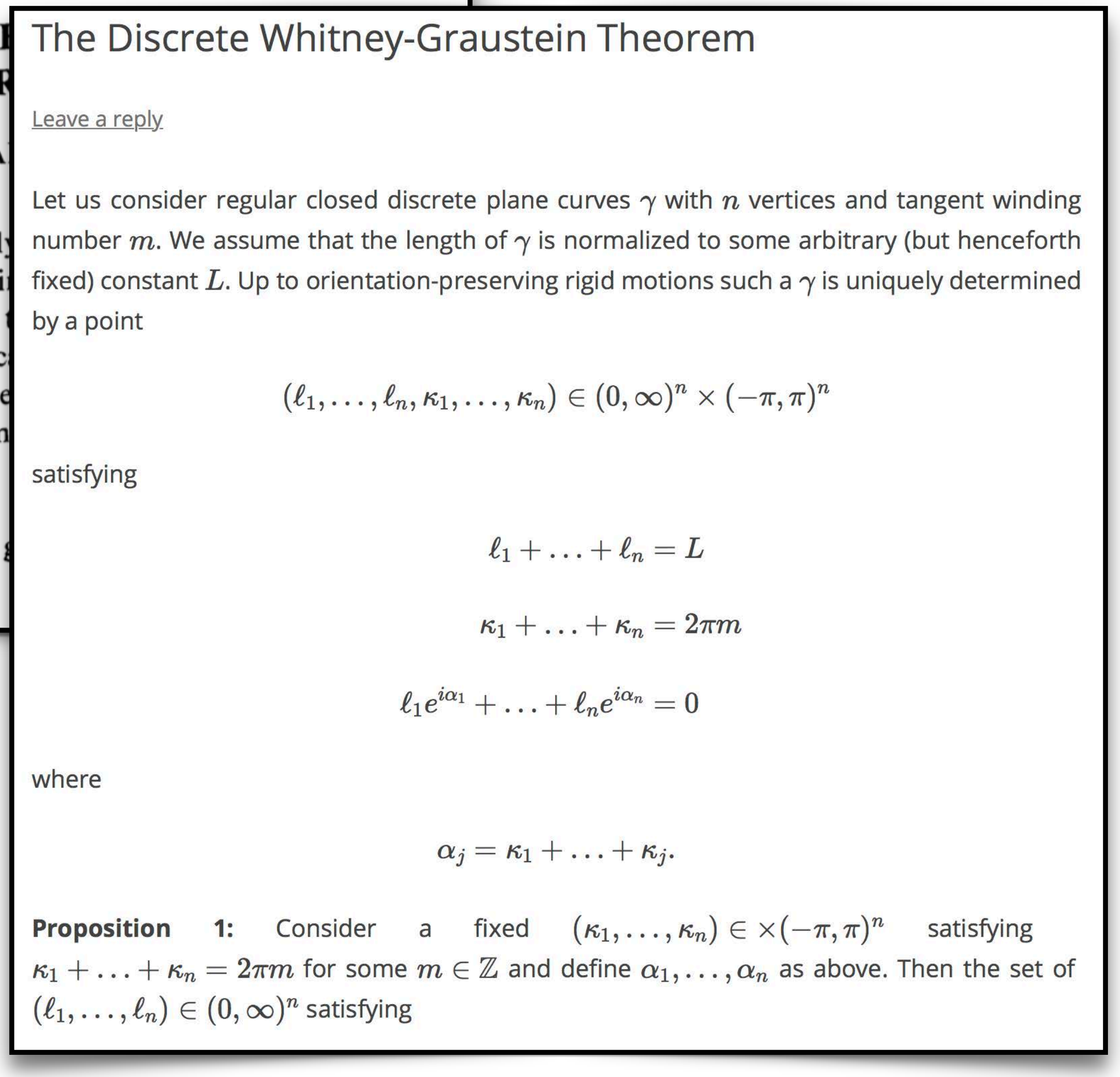
© 1991 Society for Industrial and Applied Mathematics
001

CONSTRUCTIVE WHITNEY–GRAUSTEIN TH
OR HOW TO UNTANGLE CLOSED PLANAR

KURT MEHLHORN† AND CHEE-KENG YA

Abstract. The classification of polygons is considered in which two poly
if one can be continuously transformed into the other such that for each i
adjacent edges overlap. A discrete analogue of the classic Whitney–Graustein t
that the winding number of polygons is a complete invariant for this classific
constructive in that for any pair of equivalent polygons, it produces some seque
taking one polygon to the other. Although this sequence has a quadratic num
be described and computed in real time.

Key words. polygons, computational algebraic topology, computational
theorem, winding number


The Discrete Whitney-Graustein Theorem

Leave a reply

Let us consider regular closed discrete plane curves γ with n vertices and tangent winding number m. We assume that the length of γ is normalized to some arbitrary (but henceforth fixed) constant L. Up to orientation-preserving rigid motions such a γ is uniquely determined by a point

$$(\ell_1, \ldots, \ell_n, \kappa_1, \ldots, \kappa_n) \in (0,\infty)^n \times (-\pi,\pi)^n$$

satisfying

$$\ell_1 + \ldots + \ell_n = L$$

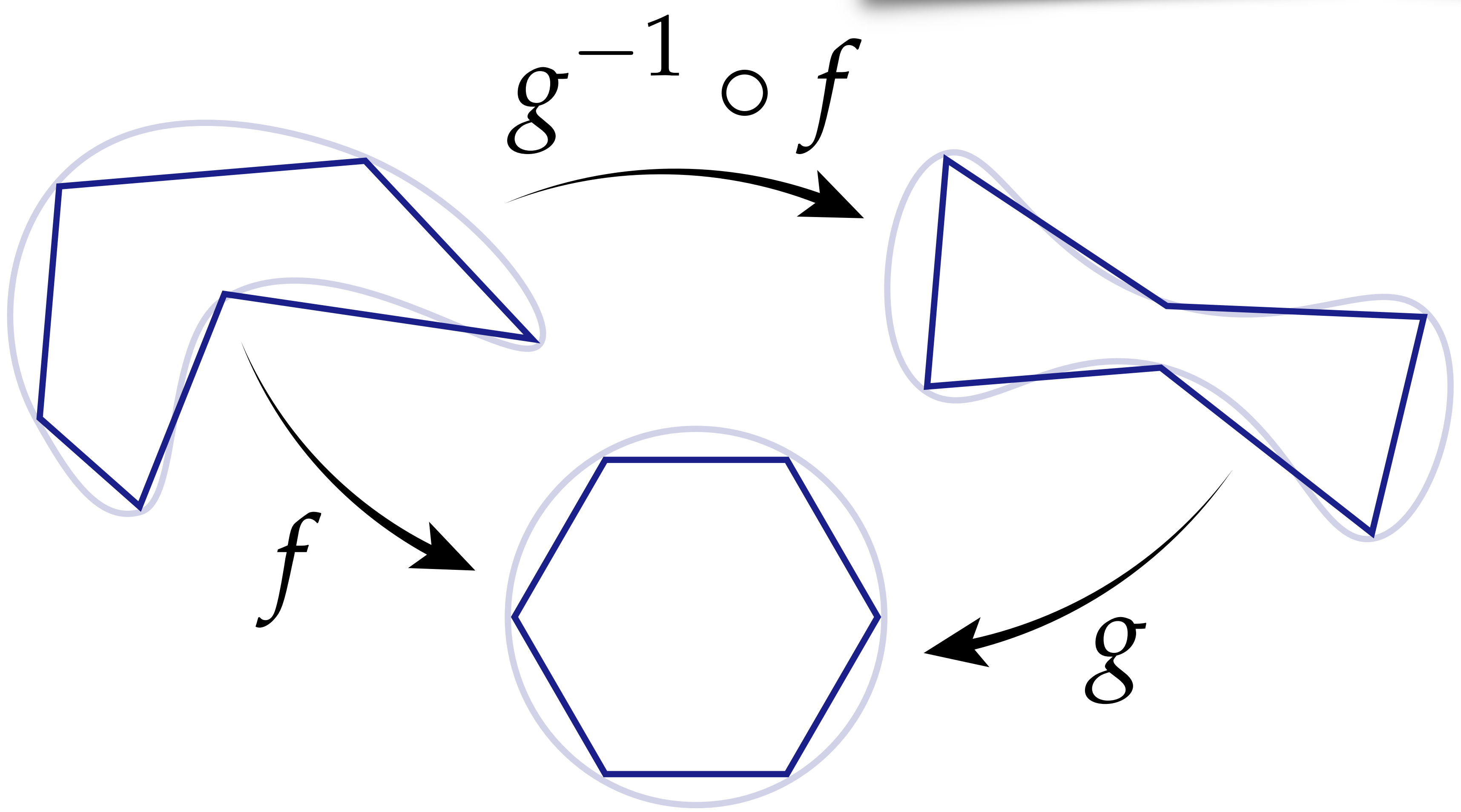
$$\kappa_1 + \ldots + \kappa_n = 2\pi m$$

$$\ell_1 e^{i\alpha_1} + \ldots + \ell_n e^{i\alpha_n} = 0$$

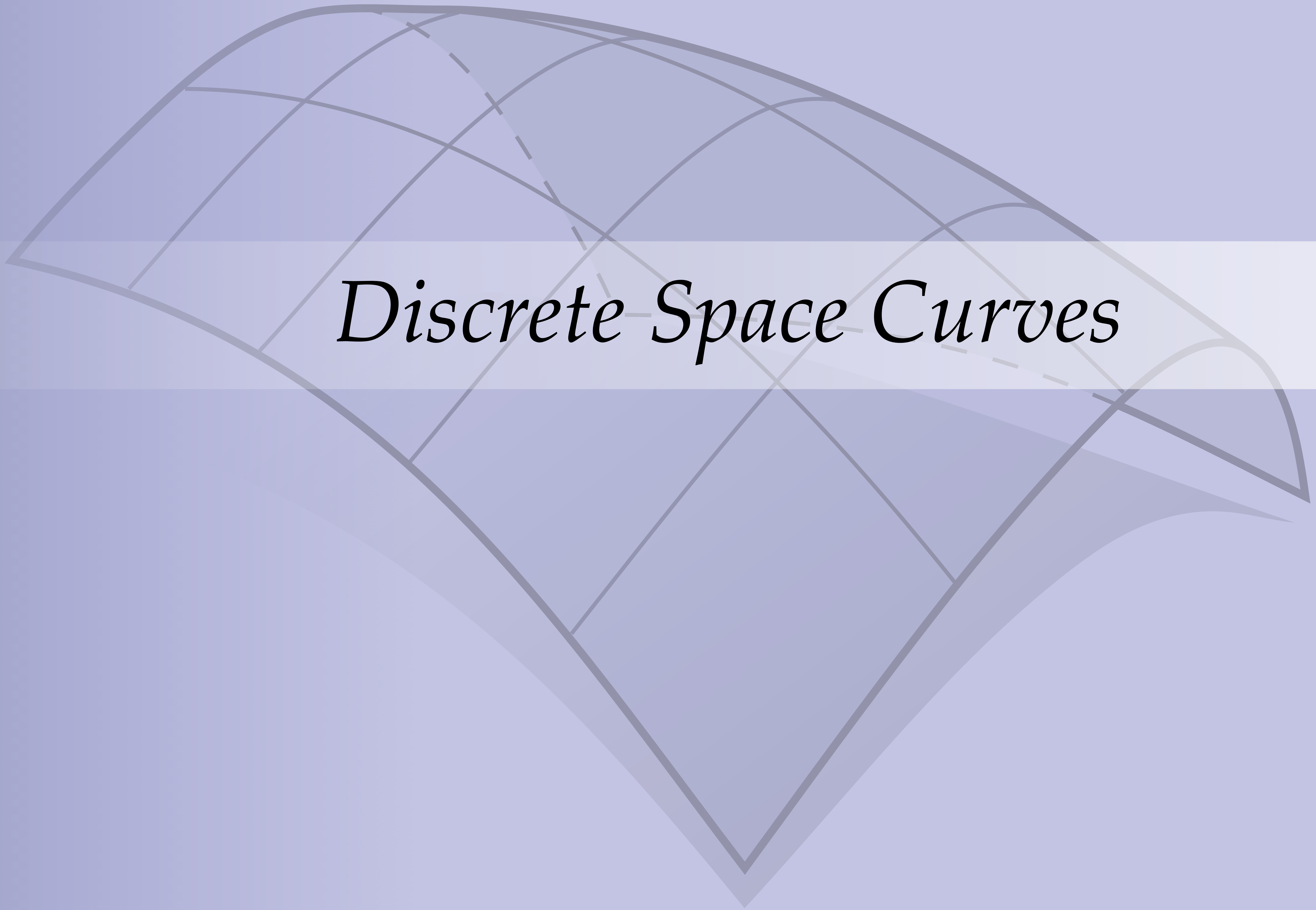
where

$$\alpha_j = \kappa_1 + \ldots + \kappa_j.$$

Proposition 1: Consider a fixed $(\kappa_1, \ldots, \kappa_n) \in \times(-\pi,\pi)^n$ satisfying $\kappa_1 + \ldots + \kappa_n = 2\pi m$ for some $m \in \mathbb{Z}$ and define $\alpha_1, \ldots, \alpha_n$ as above. Then the set of $(\ell_1, \ldots, \ell_n) \in (0,\infty)^n$ satisfying



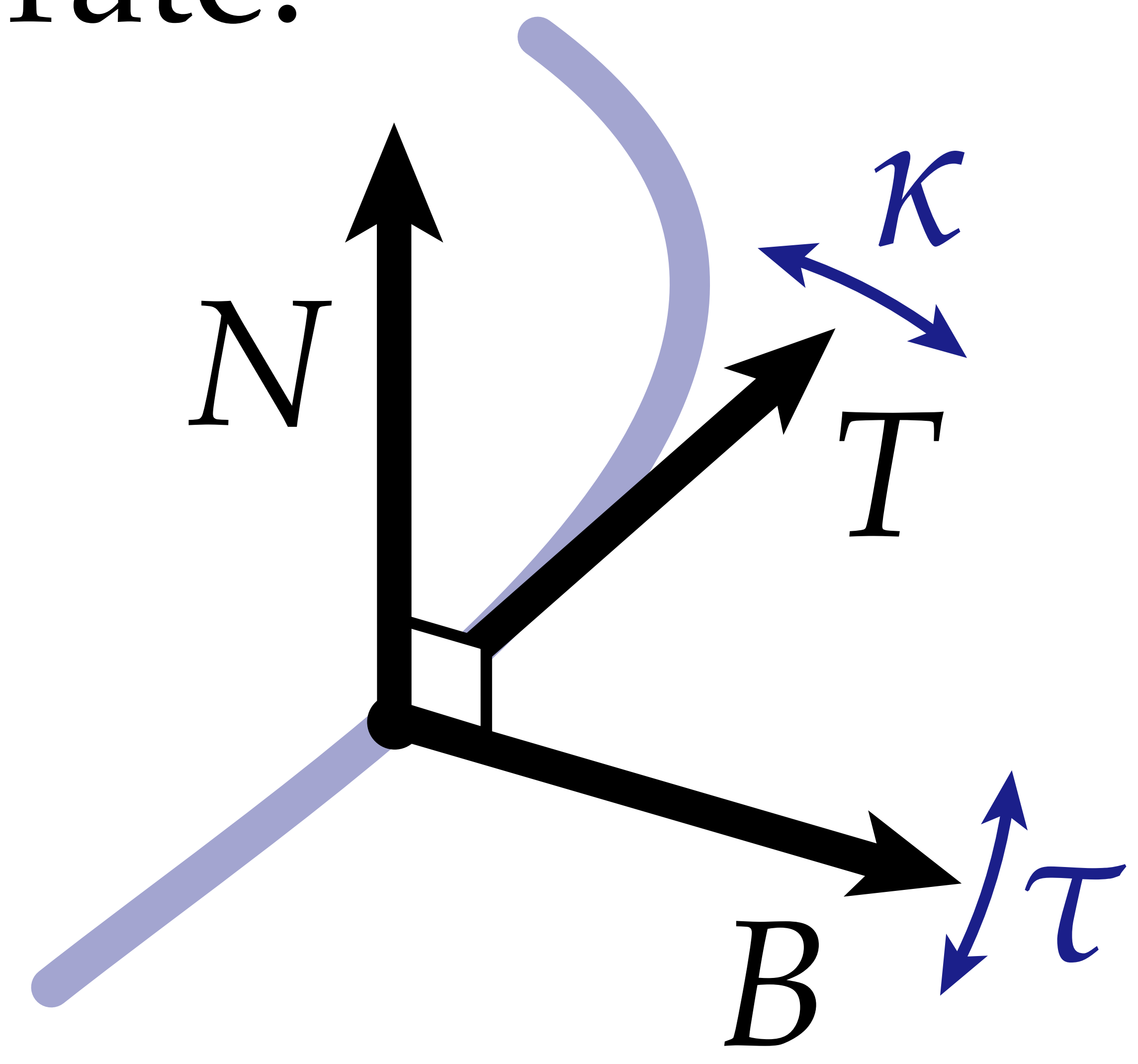
Discrete Space Curves



Review: Fundamental Theorem of Space Curves

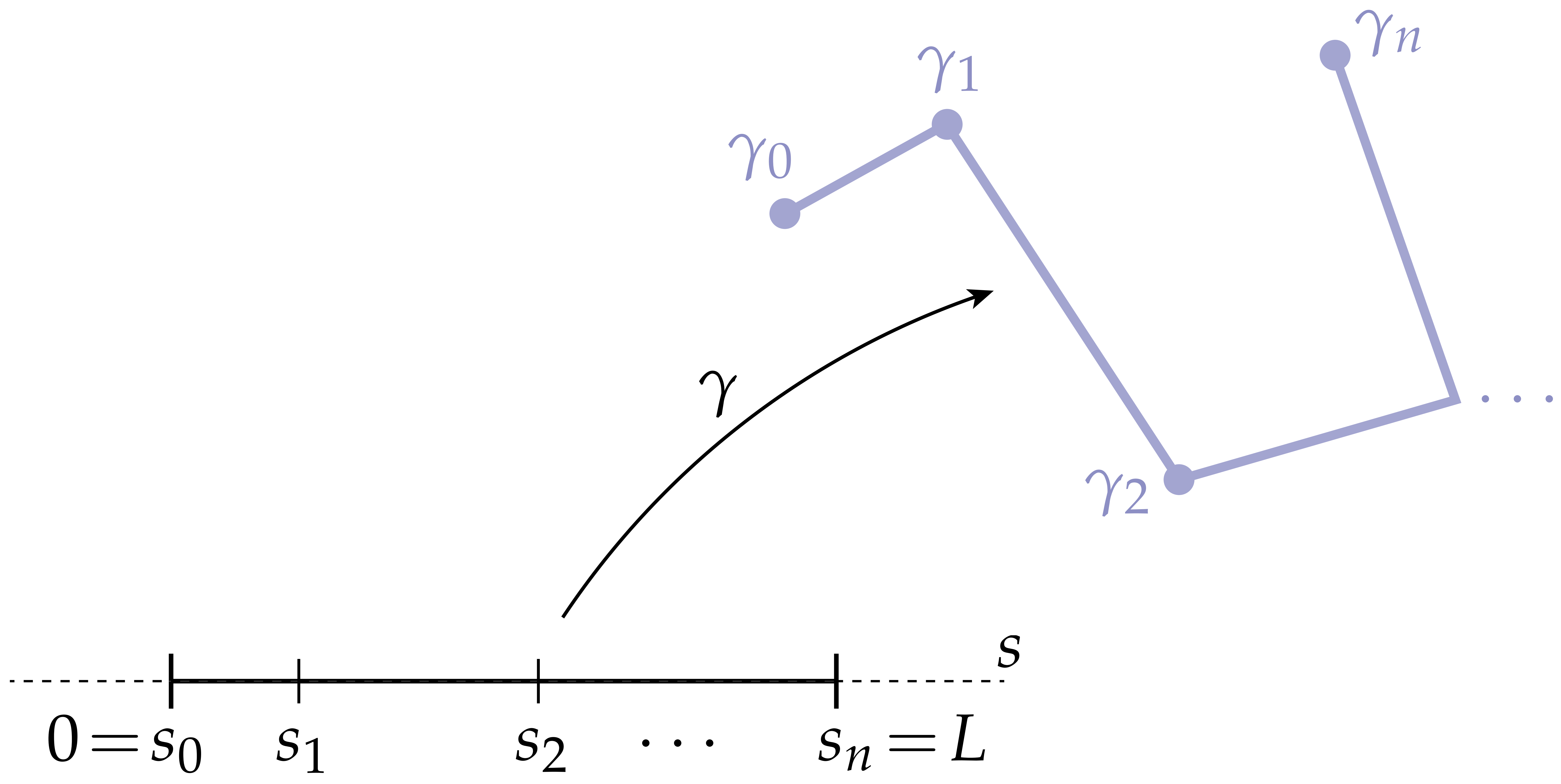
- The *fundamental theorem of space curves* tells that given the curvature κ and torsion τ of an arc-length parameterized space curve, we can recover the curve itself
- Formally: integrate the *Frenet-Serret equations*; intuitively: start drawing a curve, bend & twist at prescribed rate.

$$\frac{d}{ds}\begin{bmatrix} T \\ N \\ B \end{bmatrix} = \begin{bmatrix} 0 & -\kappa & 0 \\ \kappa & 0 & -\tau \\ 0 & \tau & 0 \end{bmatrix}\begin{bmatrix} T \\ N \\ B \end{bmatrix}$$



Discrete Space Curve

- A **discrete space curve** is simply a discrete curve in *R3* rather than R2; described by vertex positions



Fundamental Theorem of Discrete Space Curves

Q: How can we discretize the fundamental theorem for space curves?

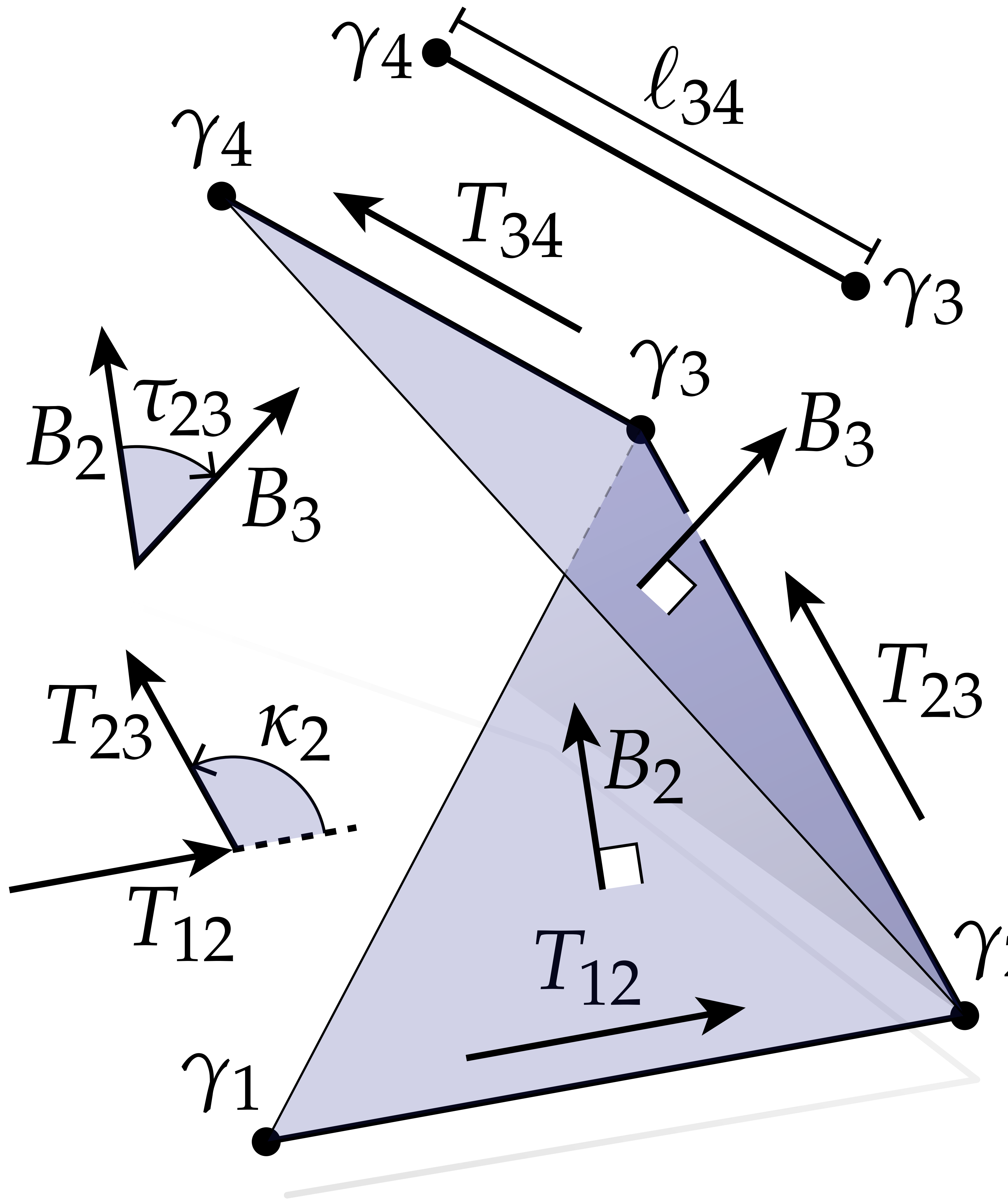
A: One possibility (*"reduced coordinates"*):

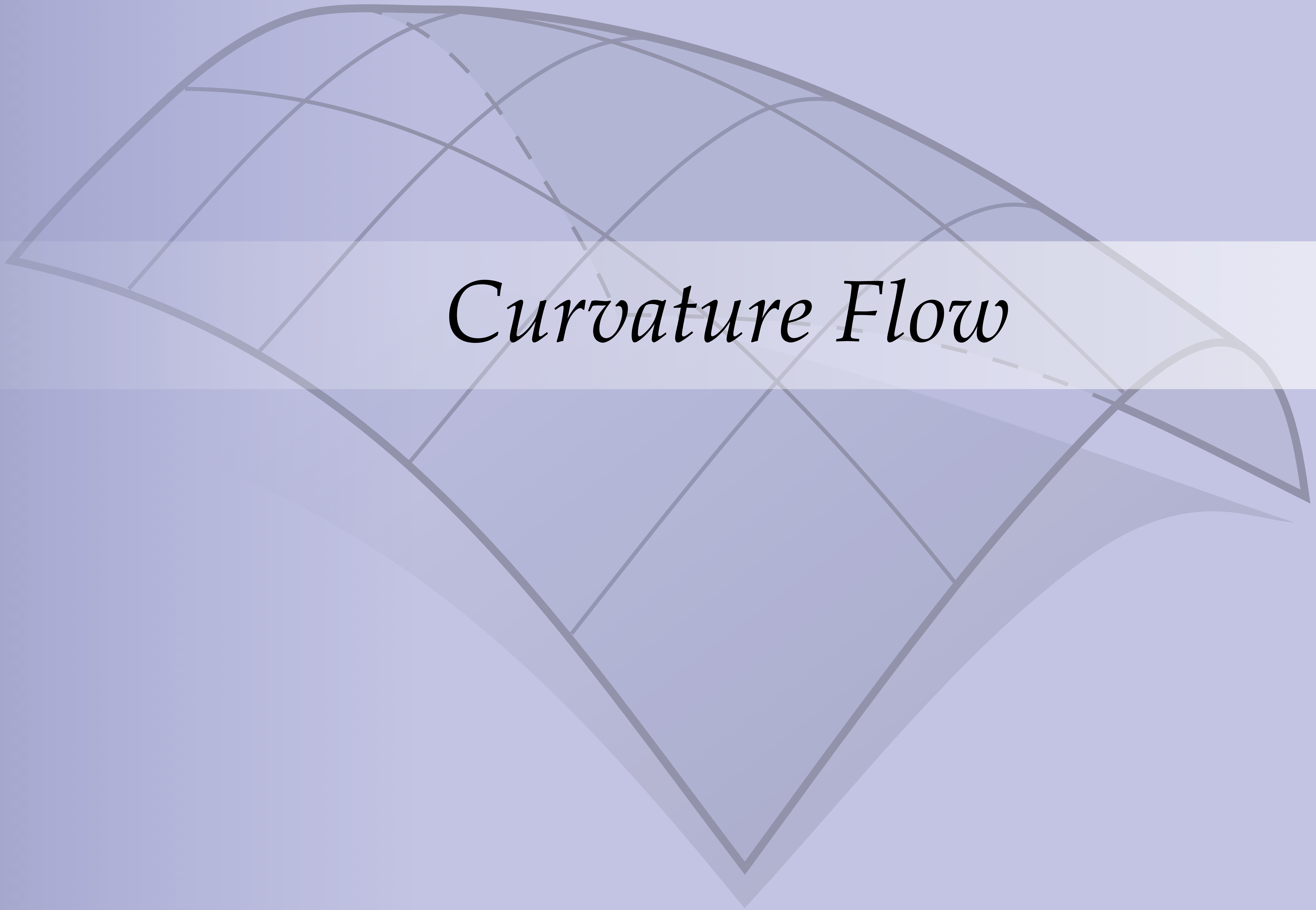
– arc length ⇒ lengths ℓ_{ij} at edges ij

– curvature ⇒ exterior angles κ_i at vertices i

– torsion ⇒ angles τ_{ij} at edges ij

Q: Reconstruction procedure?

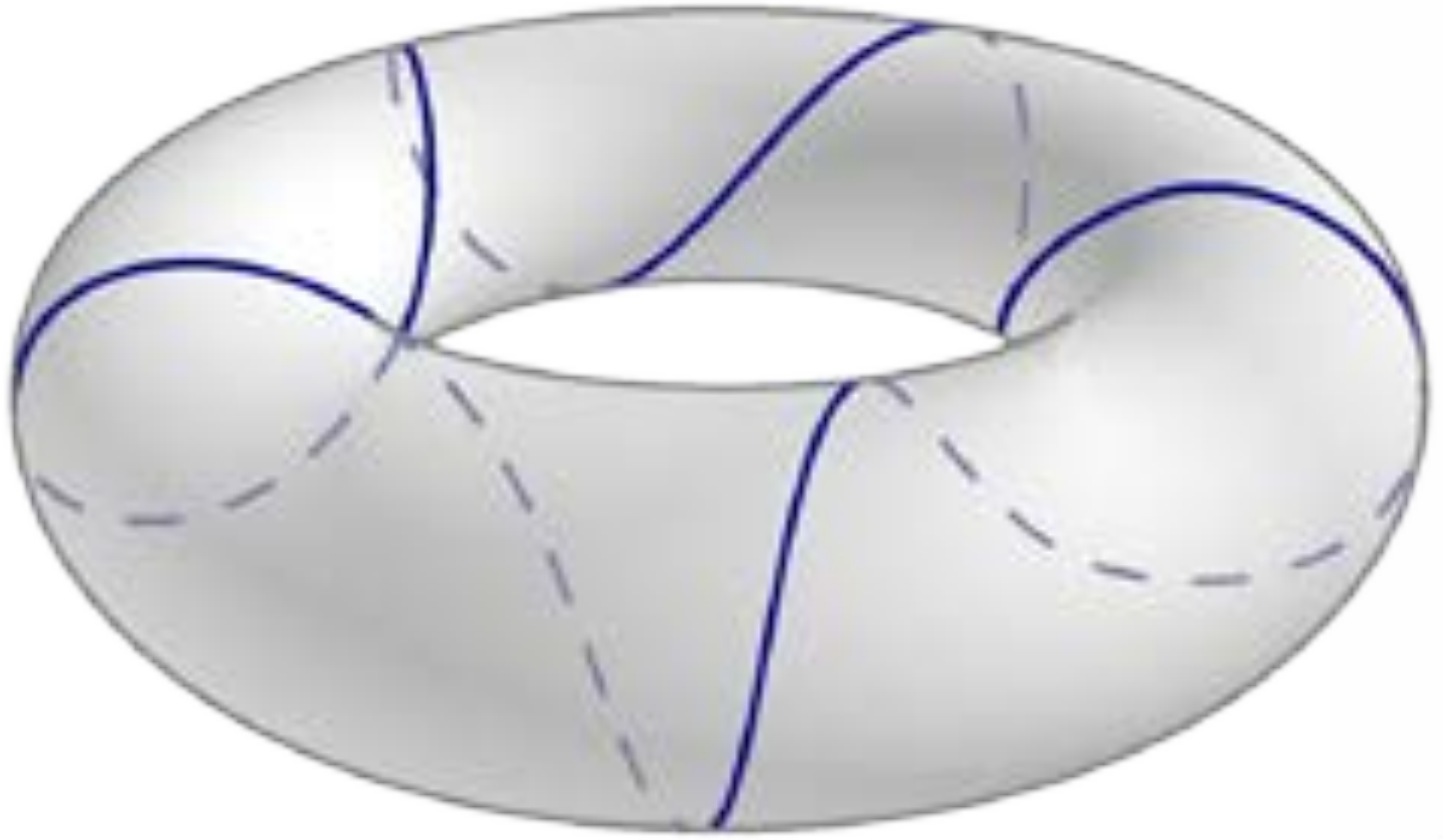


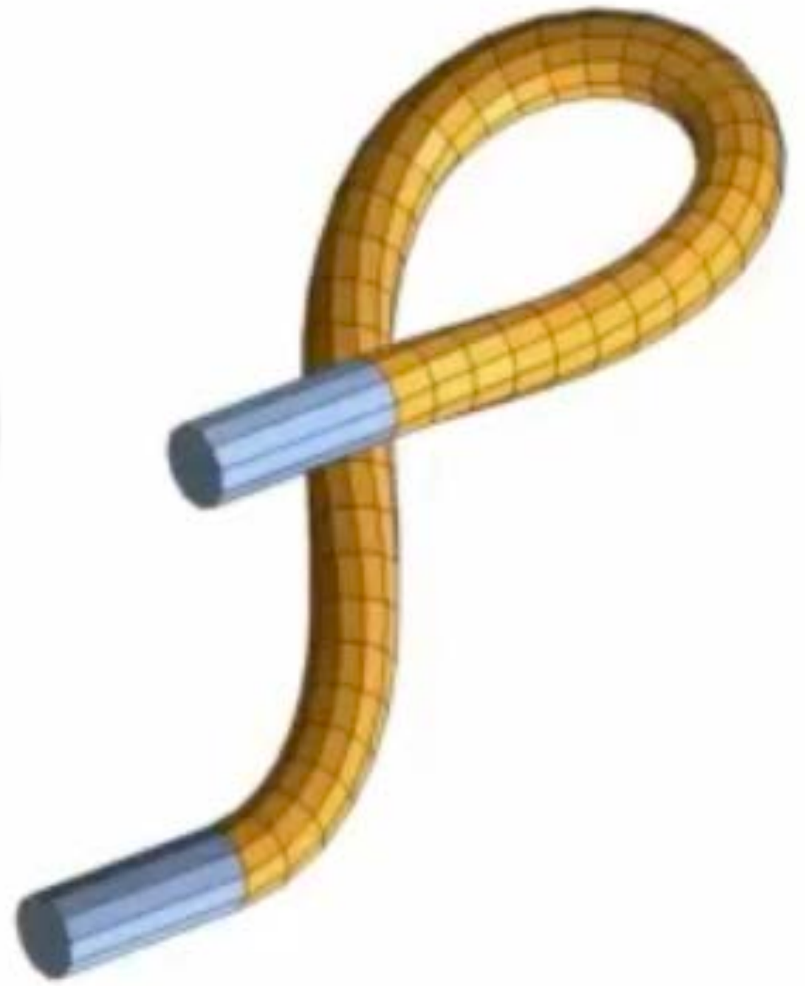


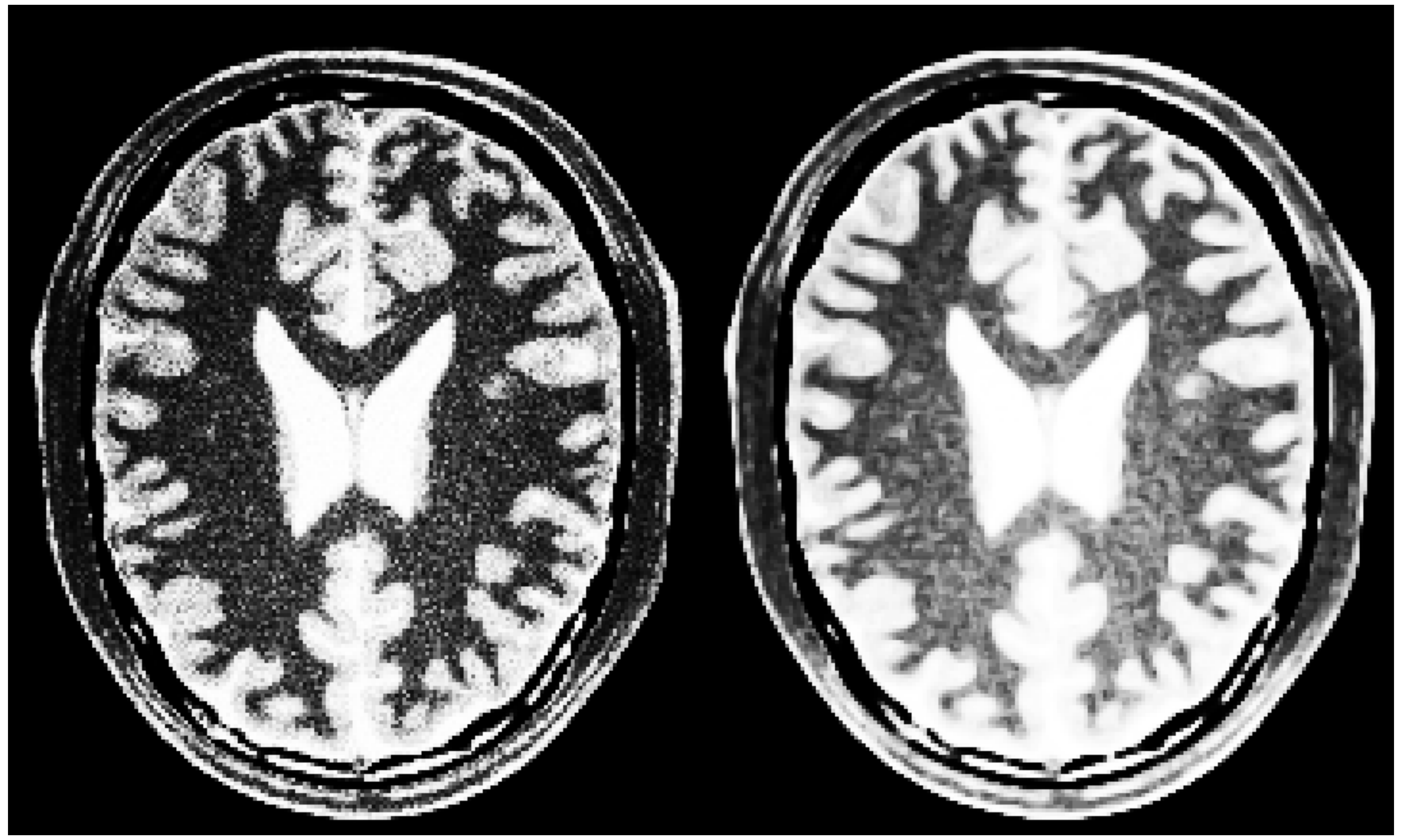
Curvature Flow

Curvature Flow on Curves

- A *curvature flow* is a time evolution of a curve (or surface) driven by some function of its curvature.
- Such flows model physical *elastic rods,* can be used to find shortest curves (*geodesics*) on surfaces, or might be used to smooth noisy data (*e.g.*, image contours).
- Two common examples: *length-shortening flow* and *elastic flow*.

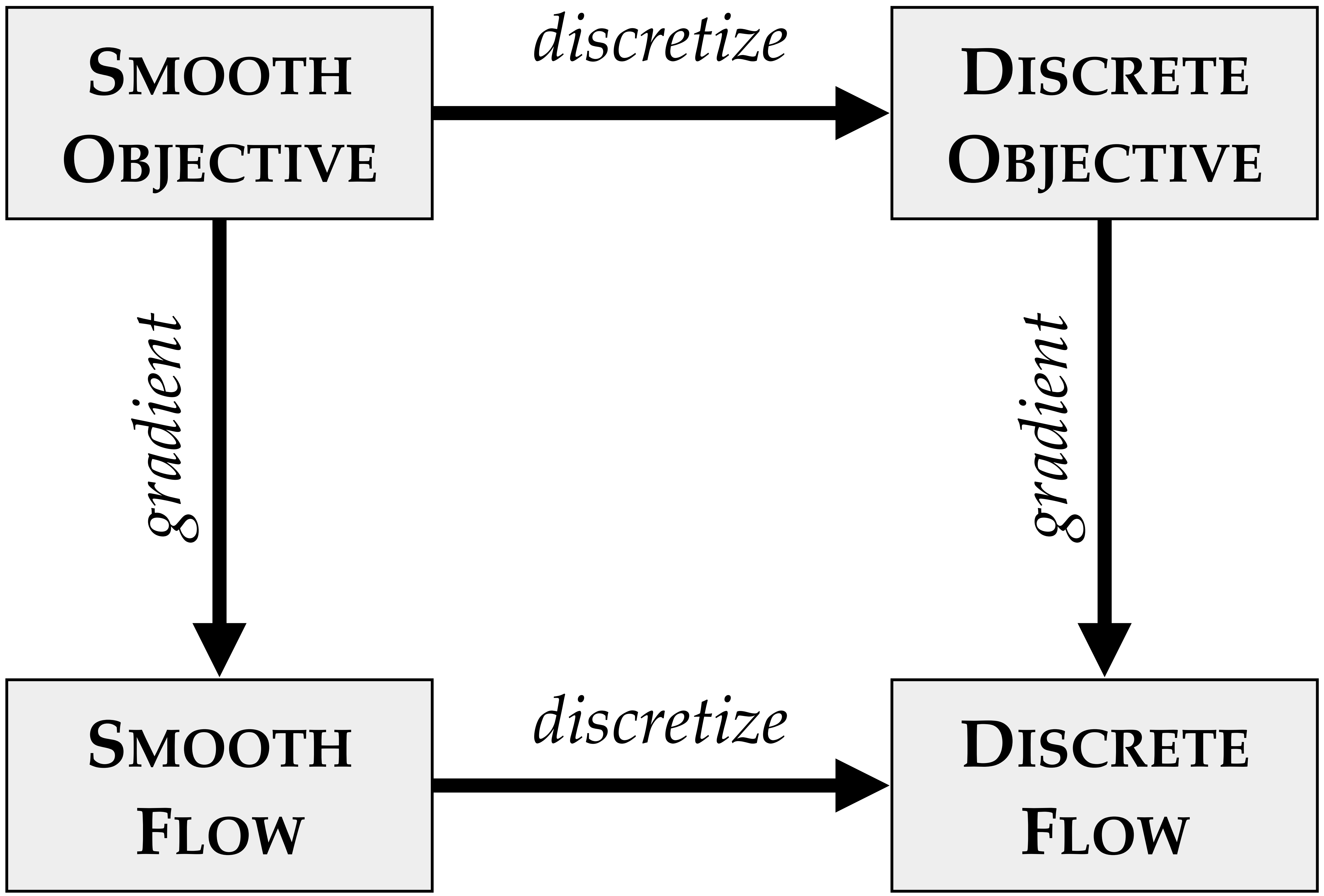






Discretizing a Gradient Flow

- Two possible paths for discretizing any gradient flow:
  1. **First** derive the gradient of the objective in the smooth setting, **then** discretize the resulting evolution equation.
  2. **First** discretize the objective itself, **then** take the gradient of the resulting discrete objective.
- In general, ***will not*** *lead to the same numerical scheme/algorithm!*



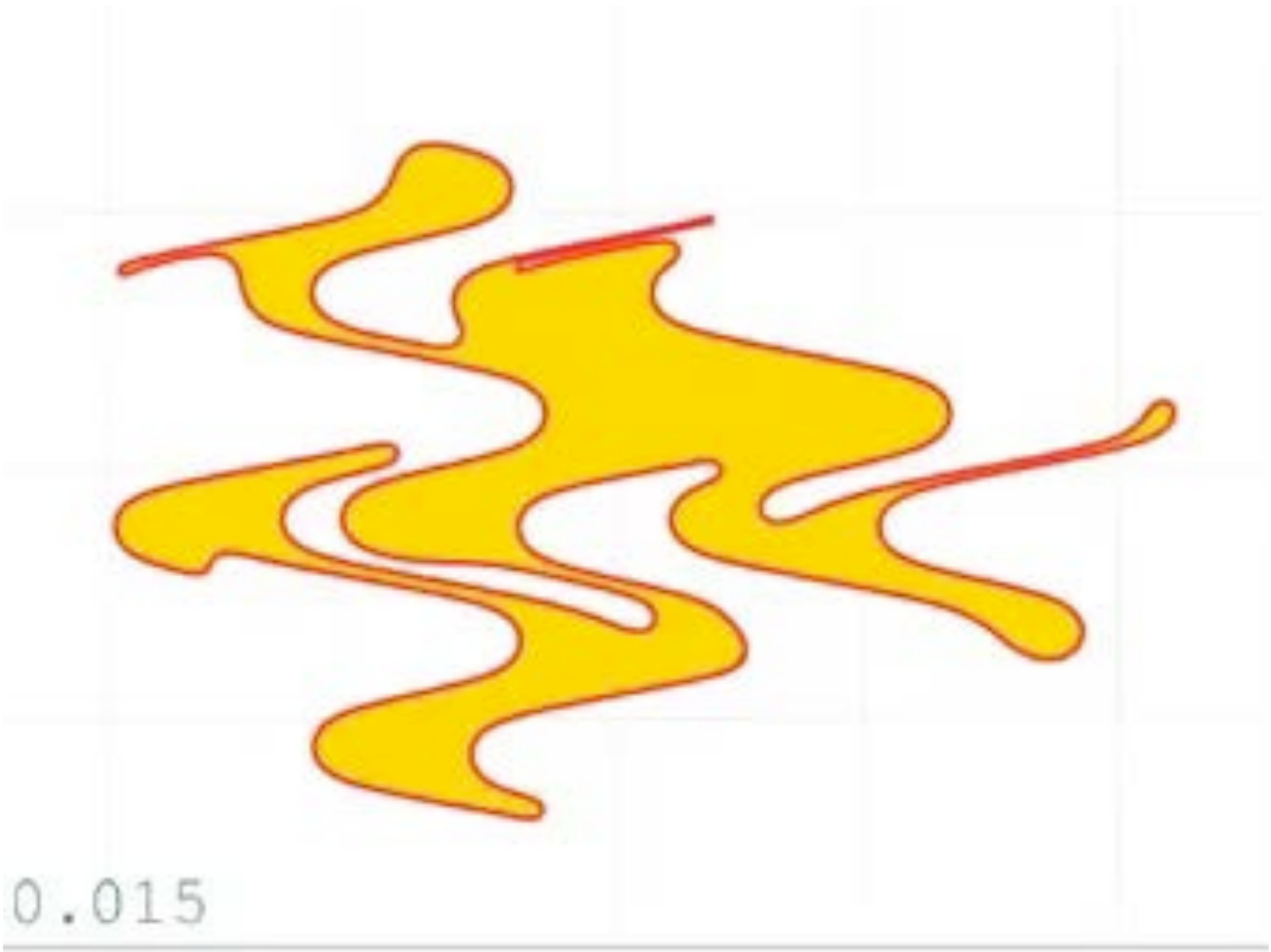
(Does **NOT** commute in general.)

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

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

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

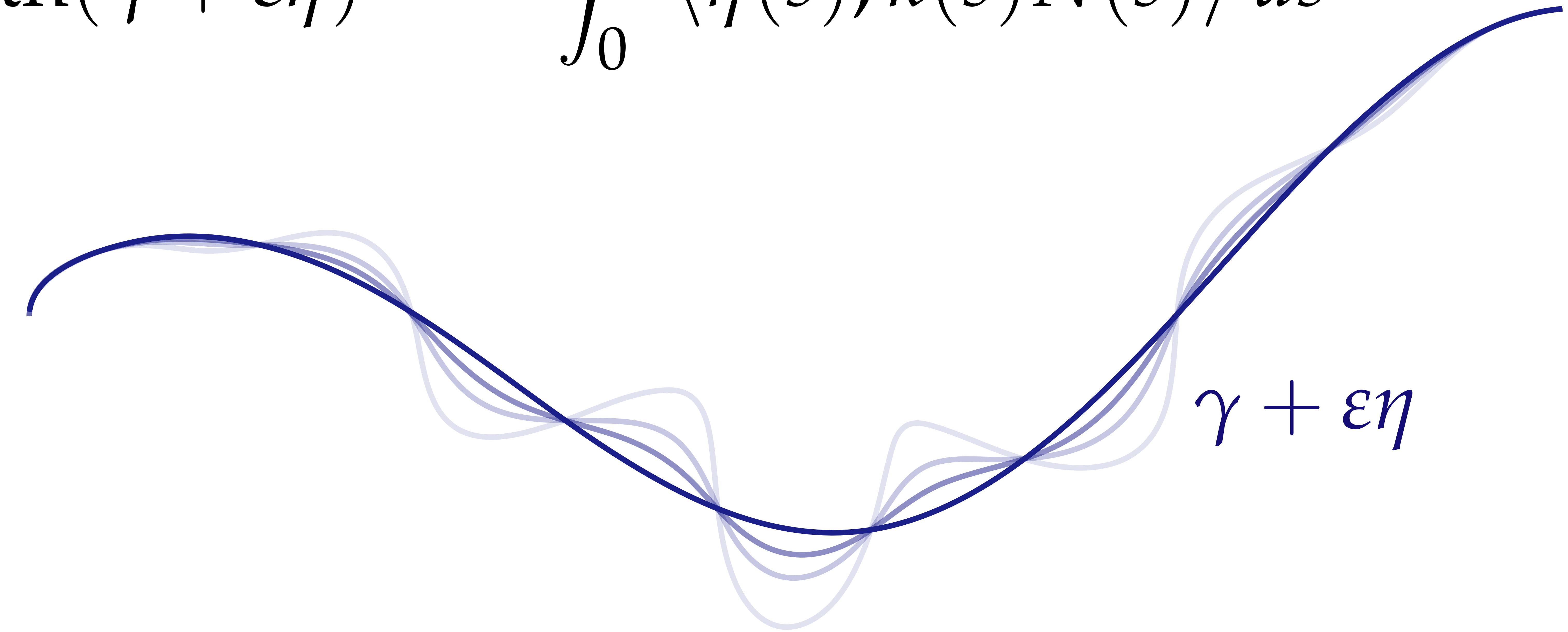


credit: Sigurd Angenent

Length Shortening Flow

Let length(γ) 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.

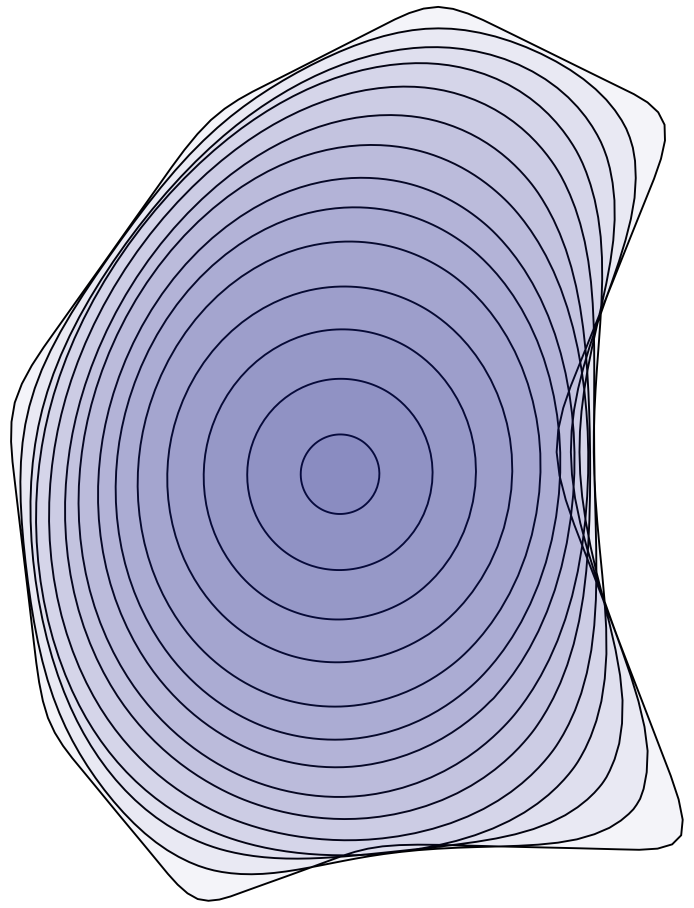
Length Shortening Flow—Forward Euler

- 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"*)

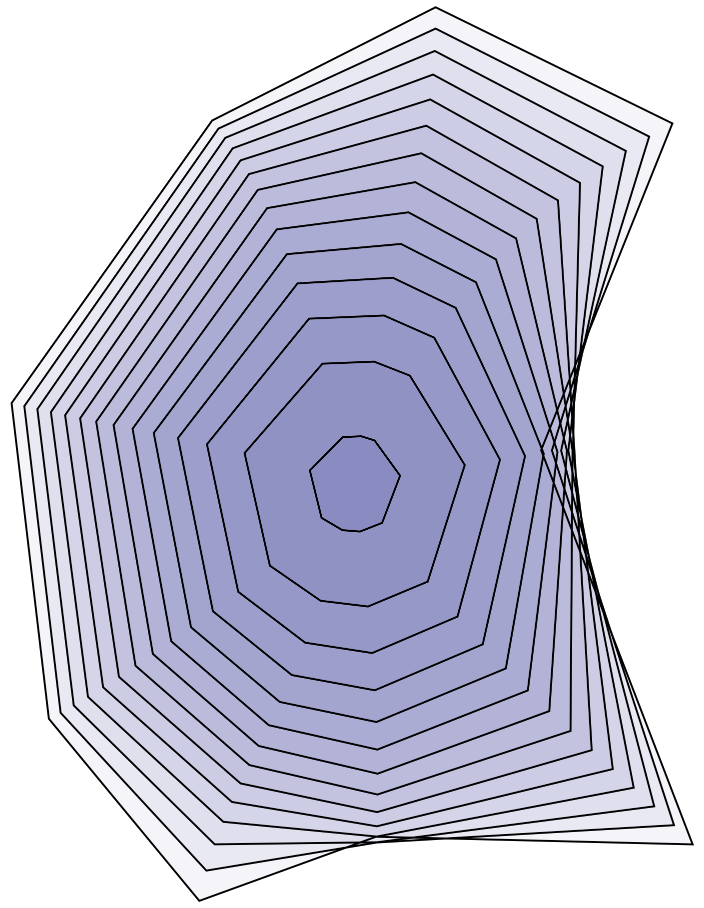
$$\frac{d}{dt}\gamma(s,t) = -\kappa(s,t)N(s,t)$$

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

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



smooth



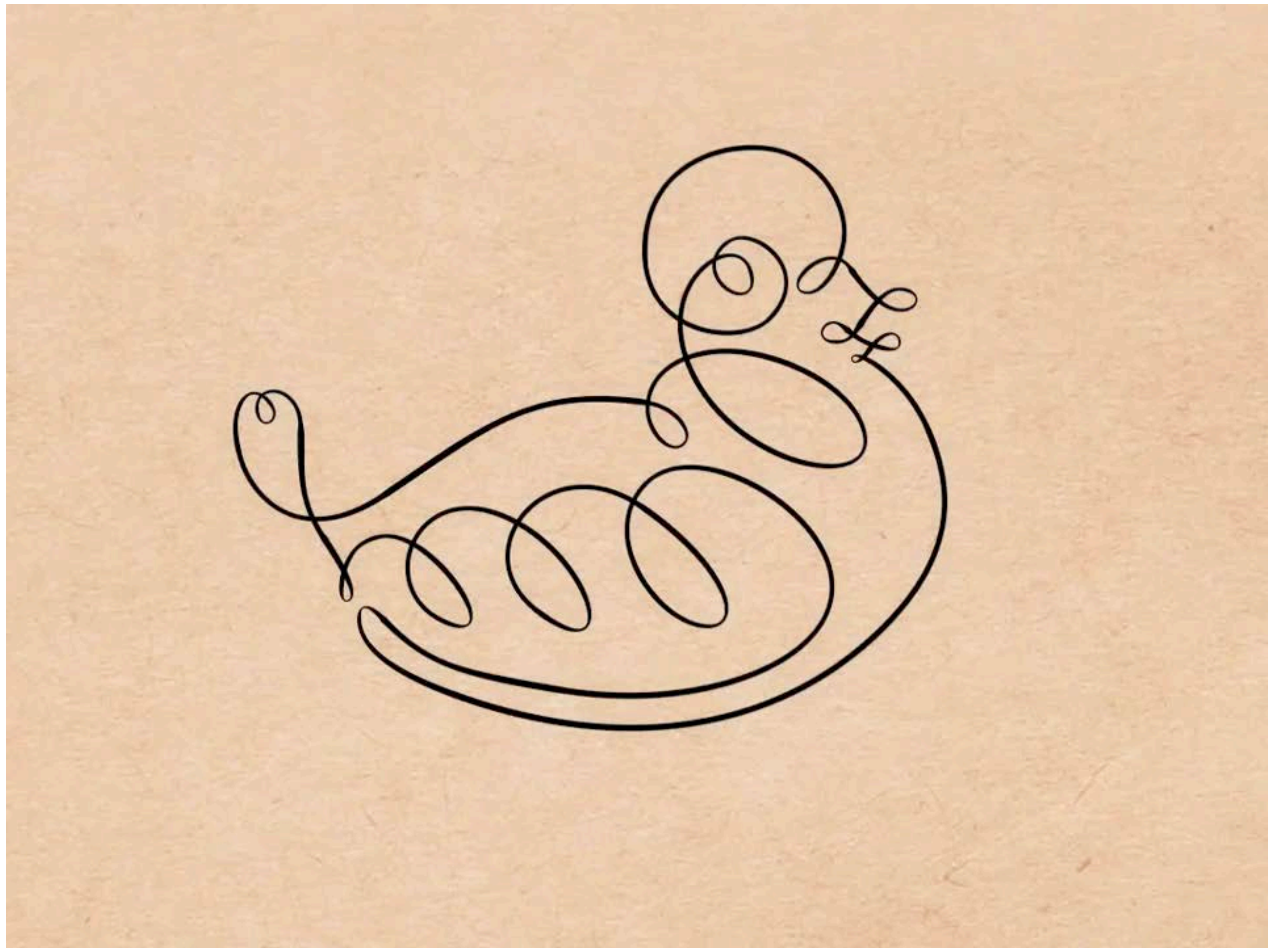
discrete

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)

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

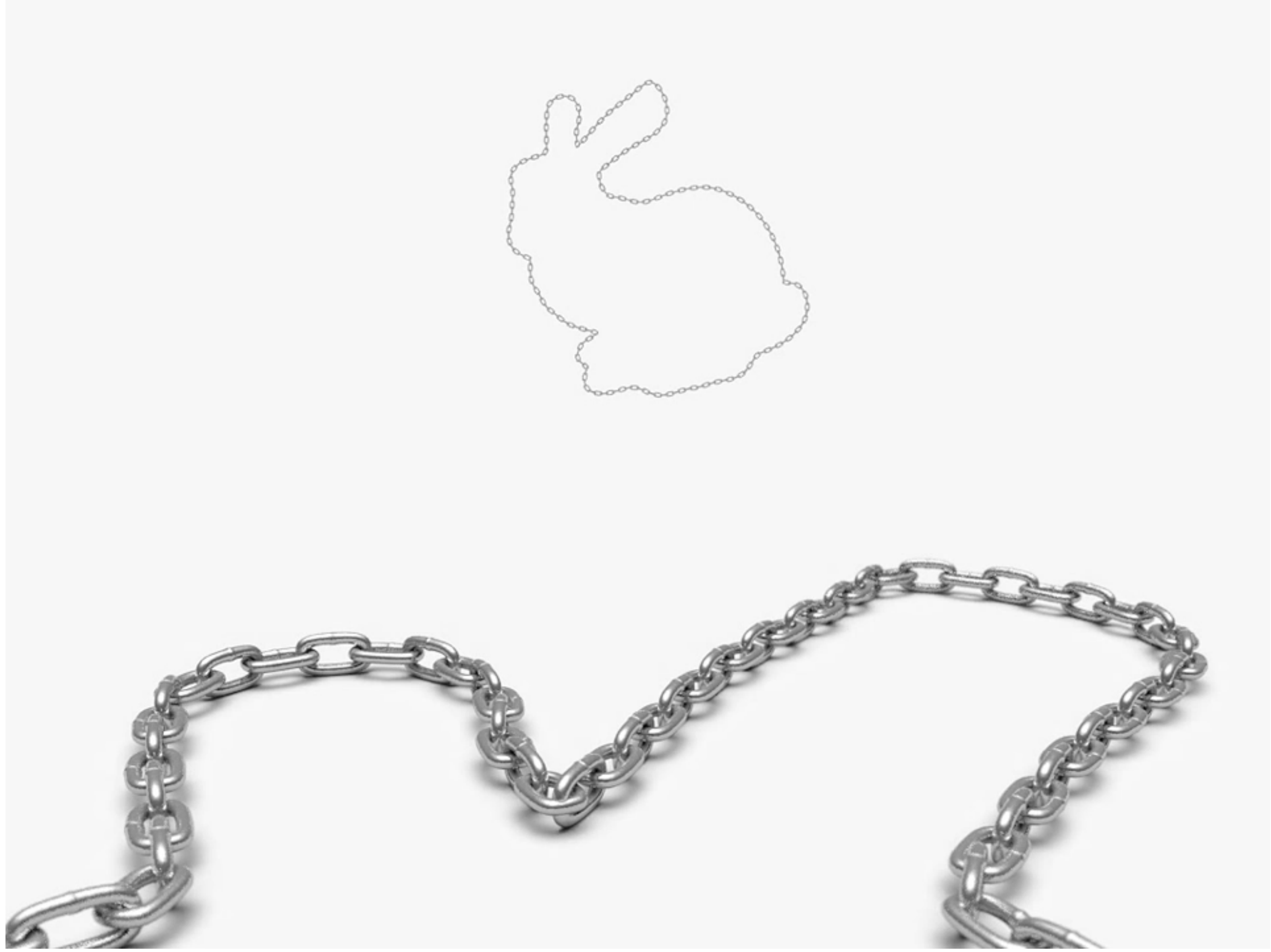
$$\tfrac{d}{dt}\gamma = -\nabla_\gamma E(\gamma)$$



http://brickisland.net/cs177fa12/?p=320

Isometric Elastic Flow

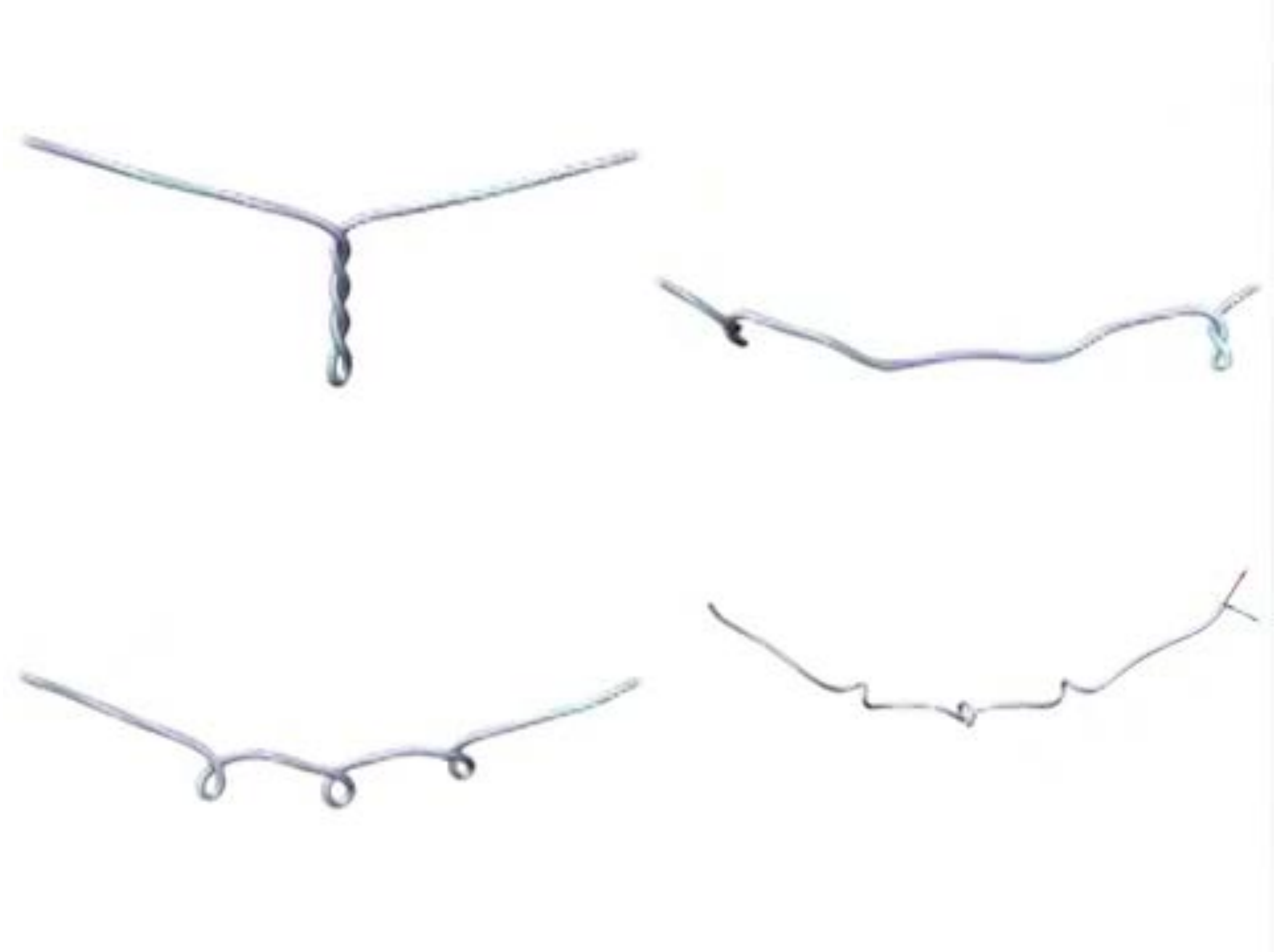
- Different way to smooth out a curve is to directly "shrink" curvature
- Discrete case: "scale down" turning angles, then use the fundamental theorem of discrete plane curves to reconstruct
- Extremely stable numerically; exactly preserves edge lengths
- Challenge: how do we make sure closed curves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



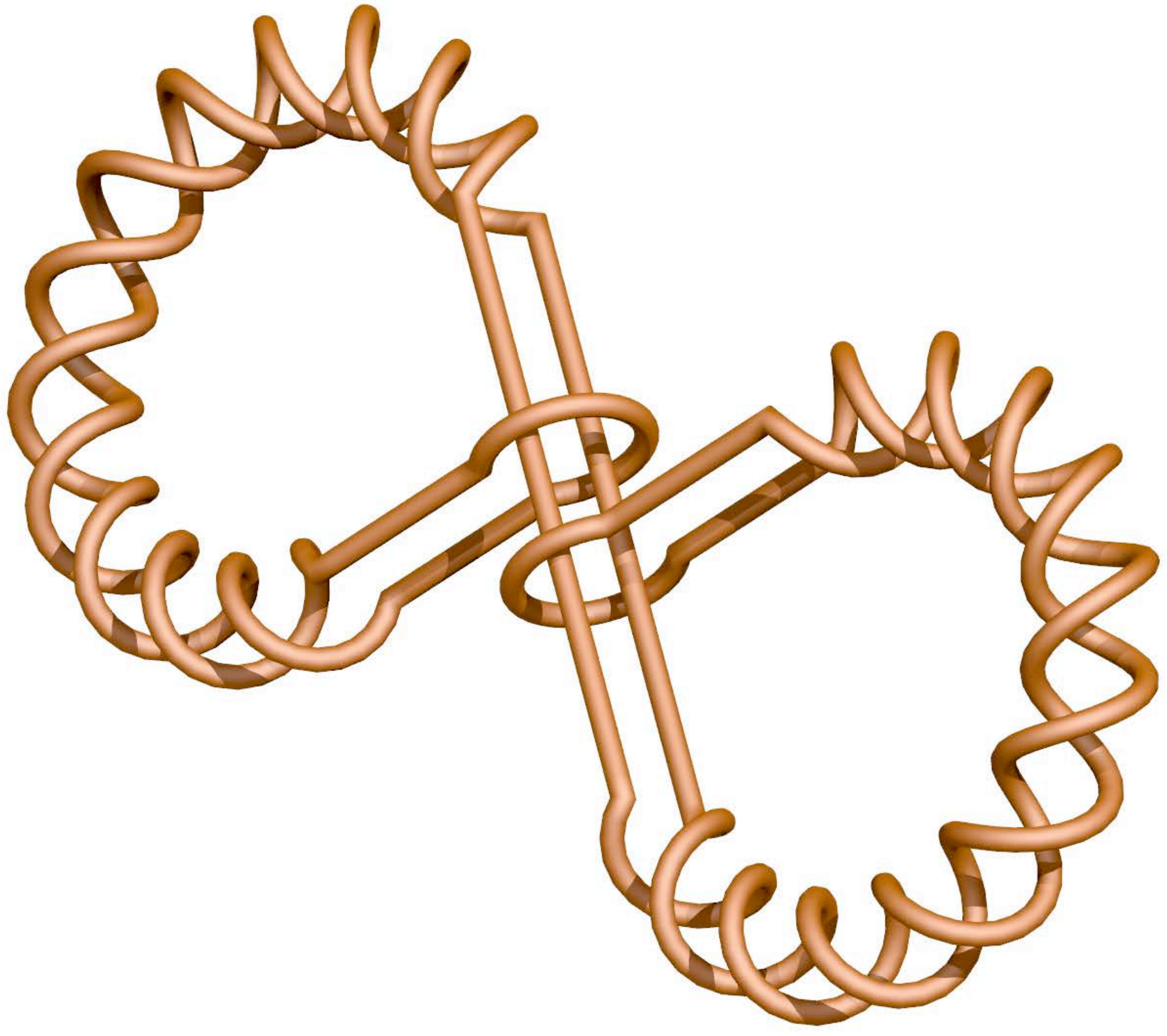
From Bergou et al, *"Discrete Elastic Rods"*

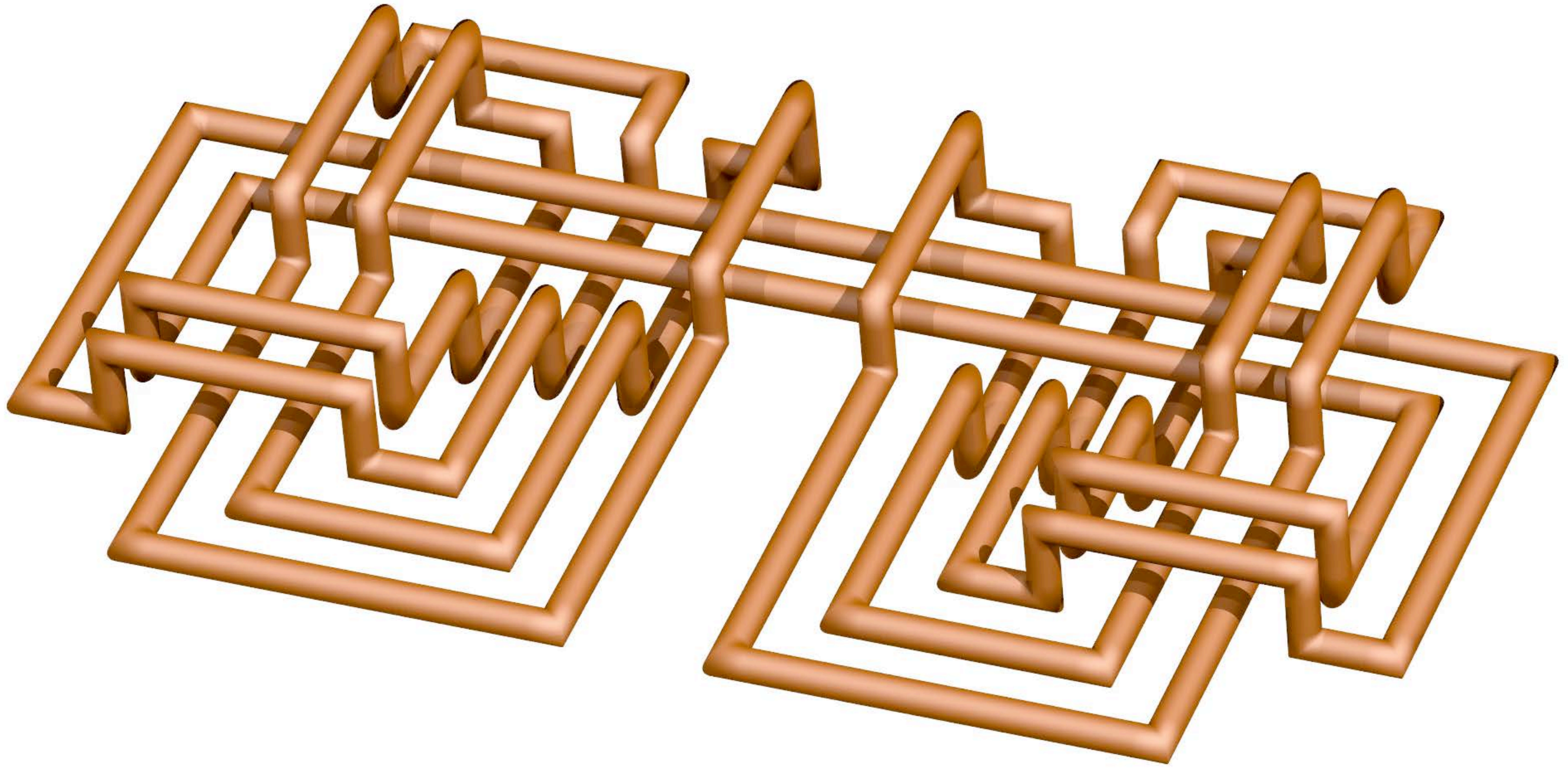
Untangling Knots

- Is a given curve "knotted?"
- Minimize elastic energy *and* penalize self-collision
- *Might* go to smoothest curve in same isotopy class

$$\int_0^L \int_0^L \frac{1}{|\gamma(s)-\gamma(t)|^2} - \frac{1}{d(s,t)^2}\, ds\, dt$$

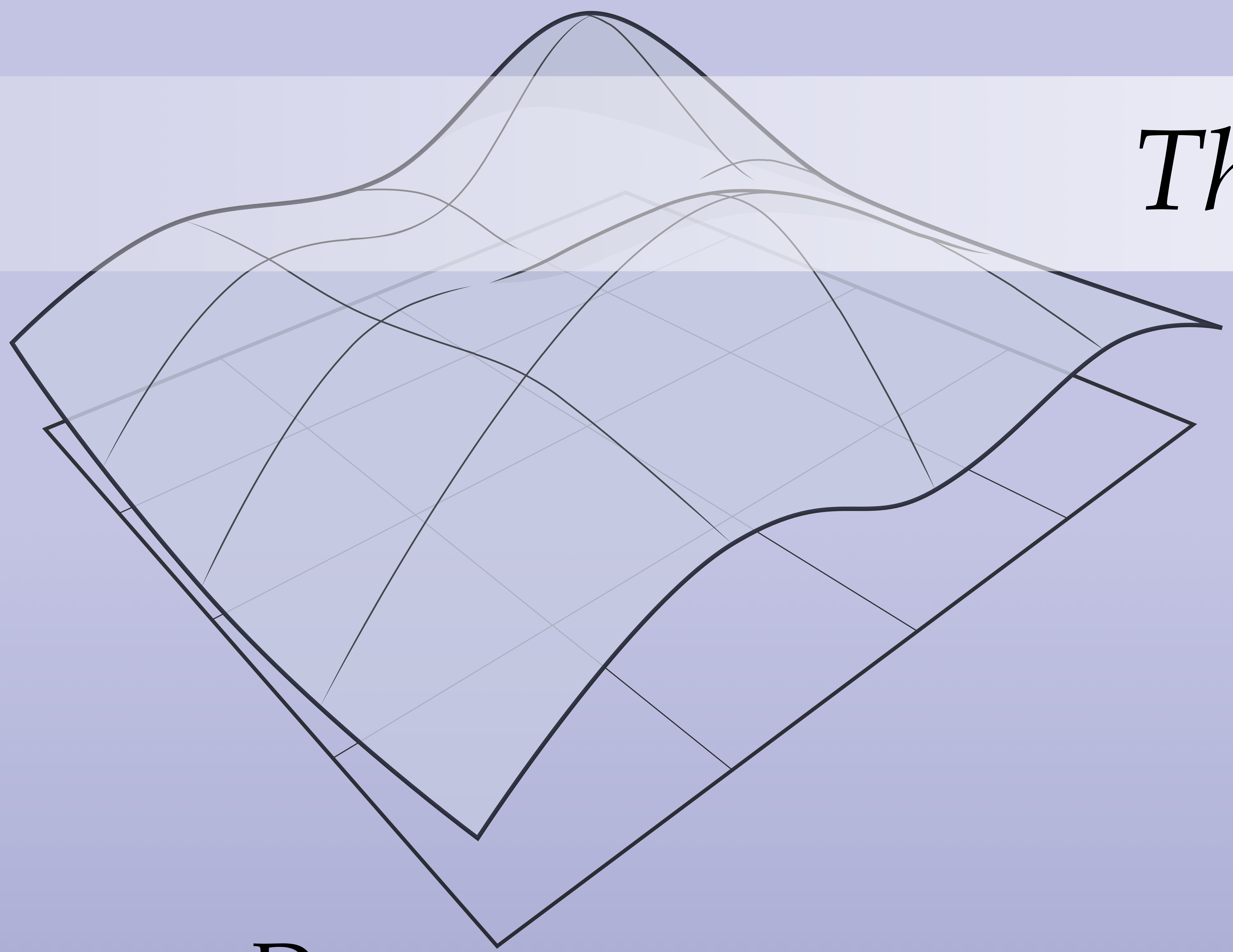
Möbius energy





Credit: Henrik Schumacher

Thanks!



Discrete Differential Geometry: An Applied Introduction

Keenan Crane • CMU 15-458/858