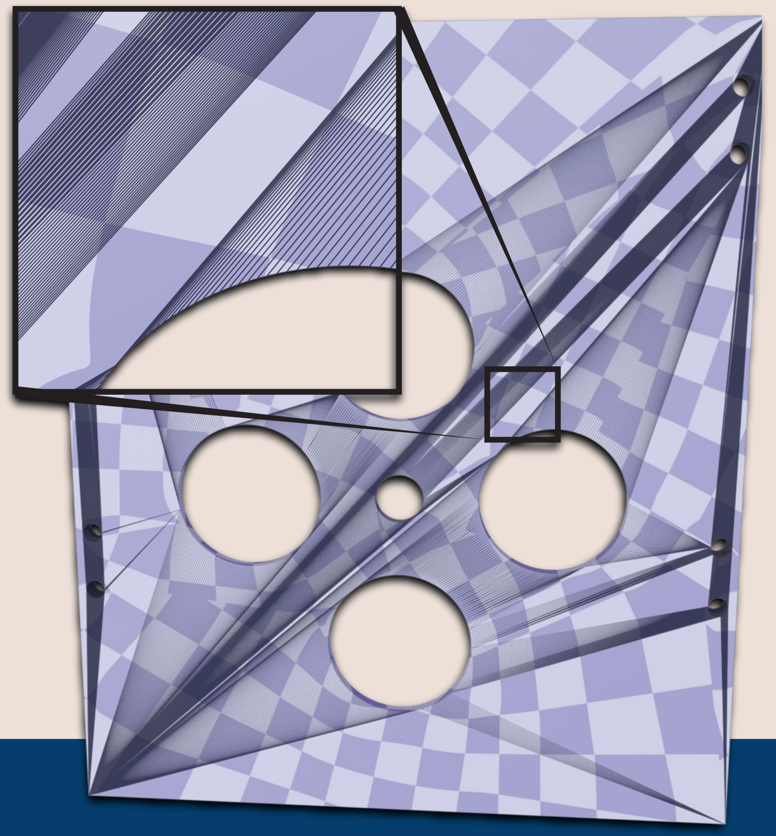
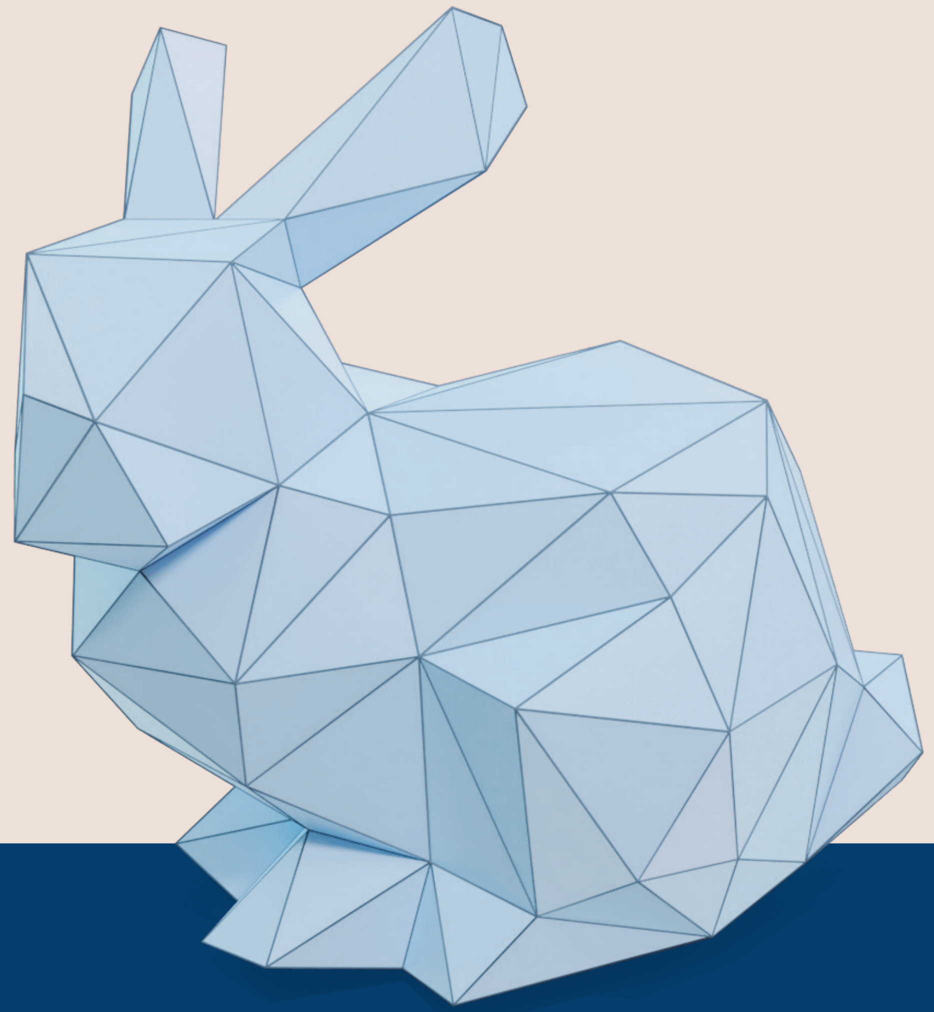




Discrete Conformal Equivalence of Polyhedral Surfaces


Mark Gillespie

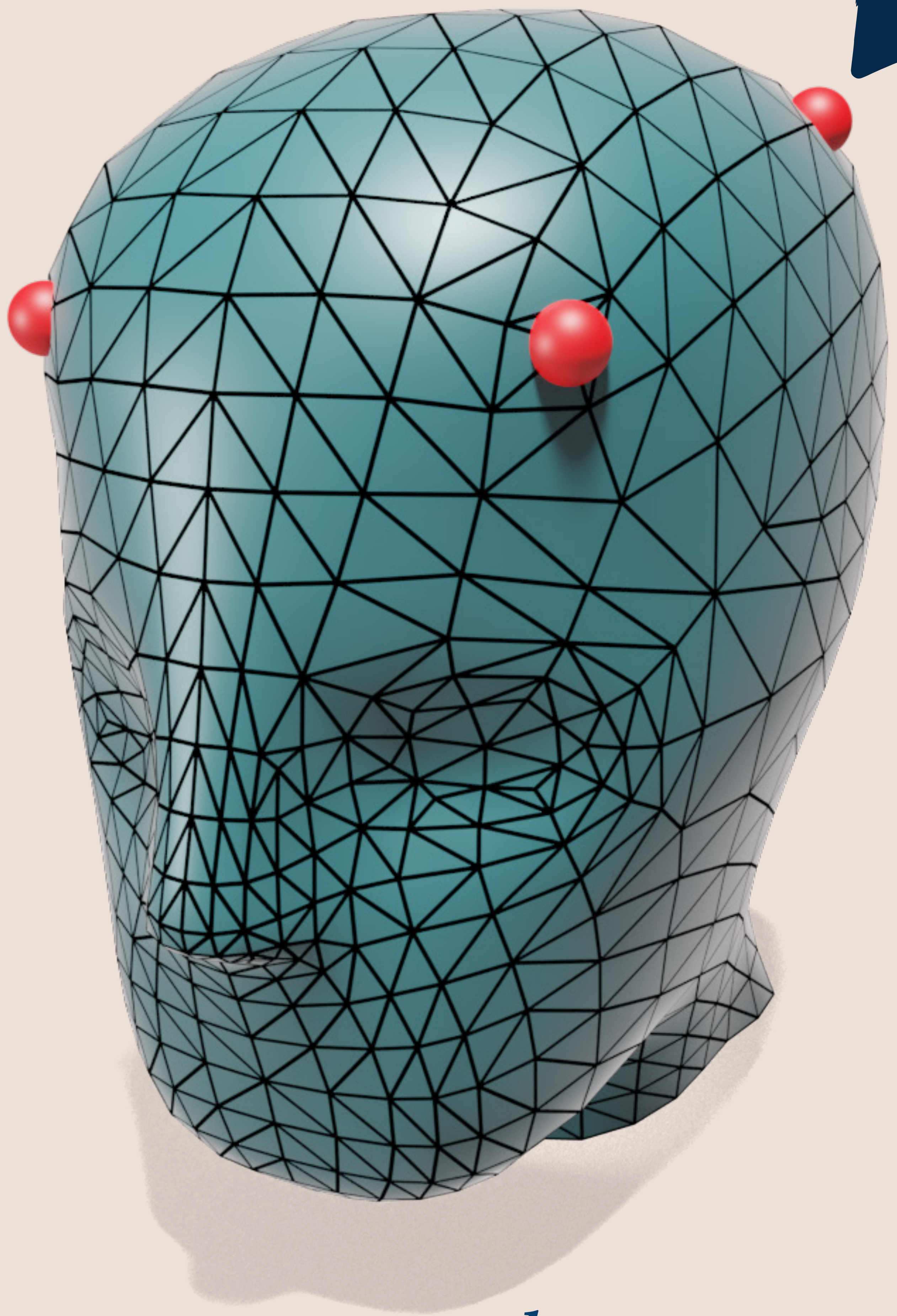
Boris Springborn

Keenan Crane

Goal: high-quality surface parameterization

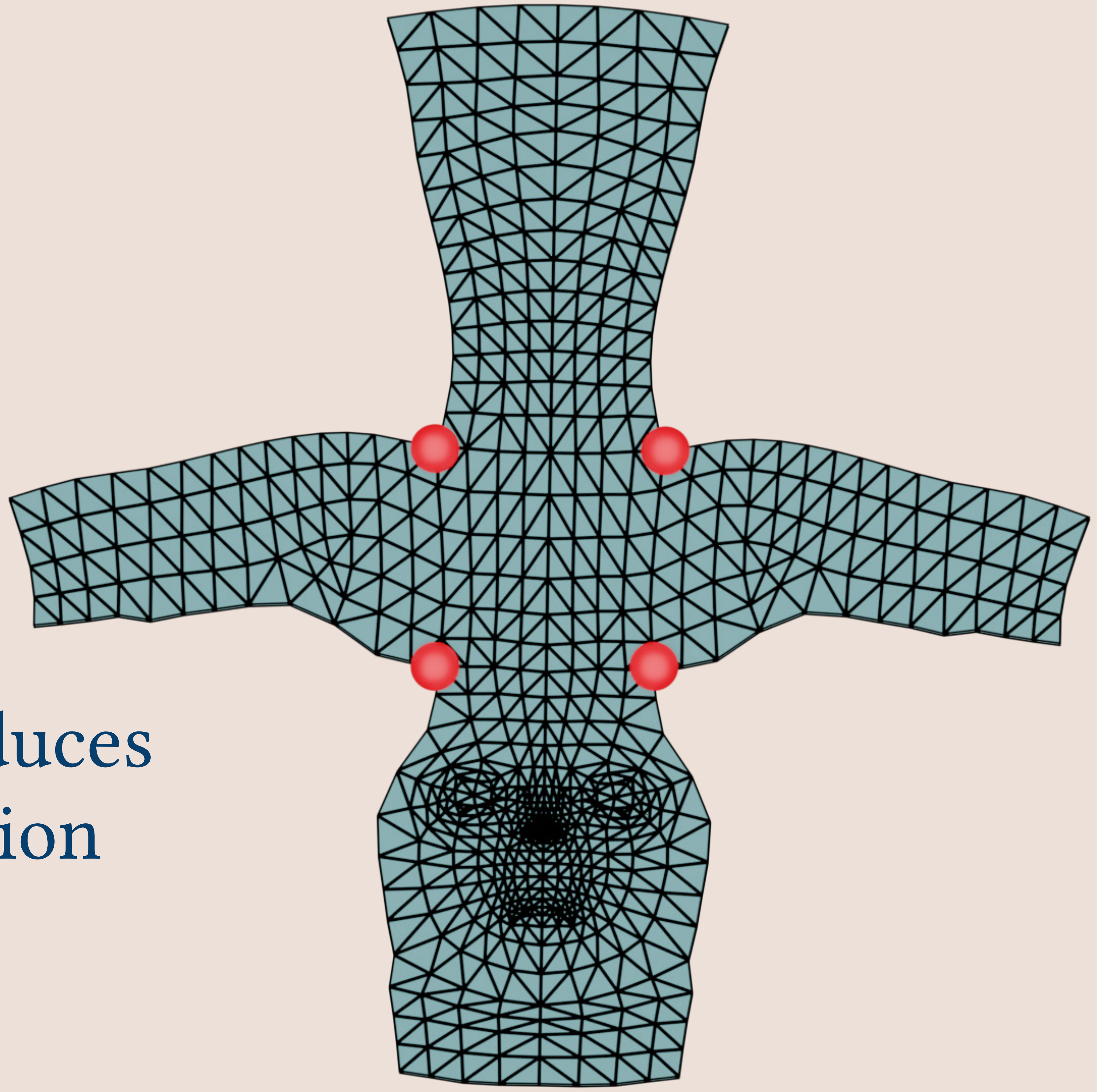
use *cone flattening*

Cones reduce area distortion



input mesh

Conformal reduces angle distortion

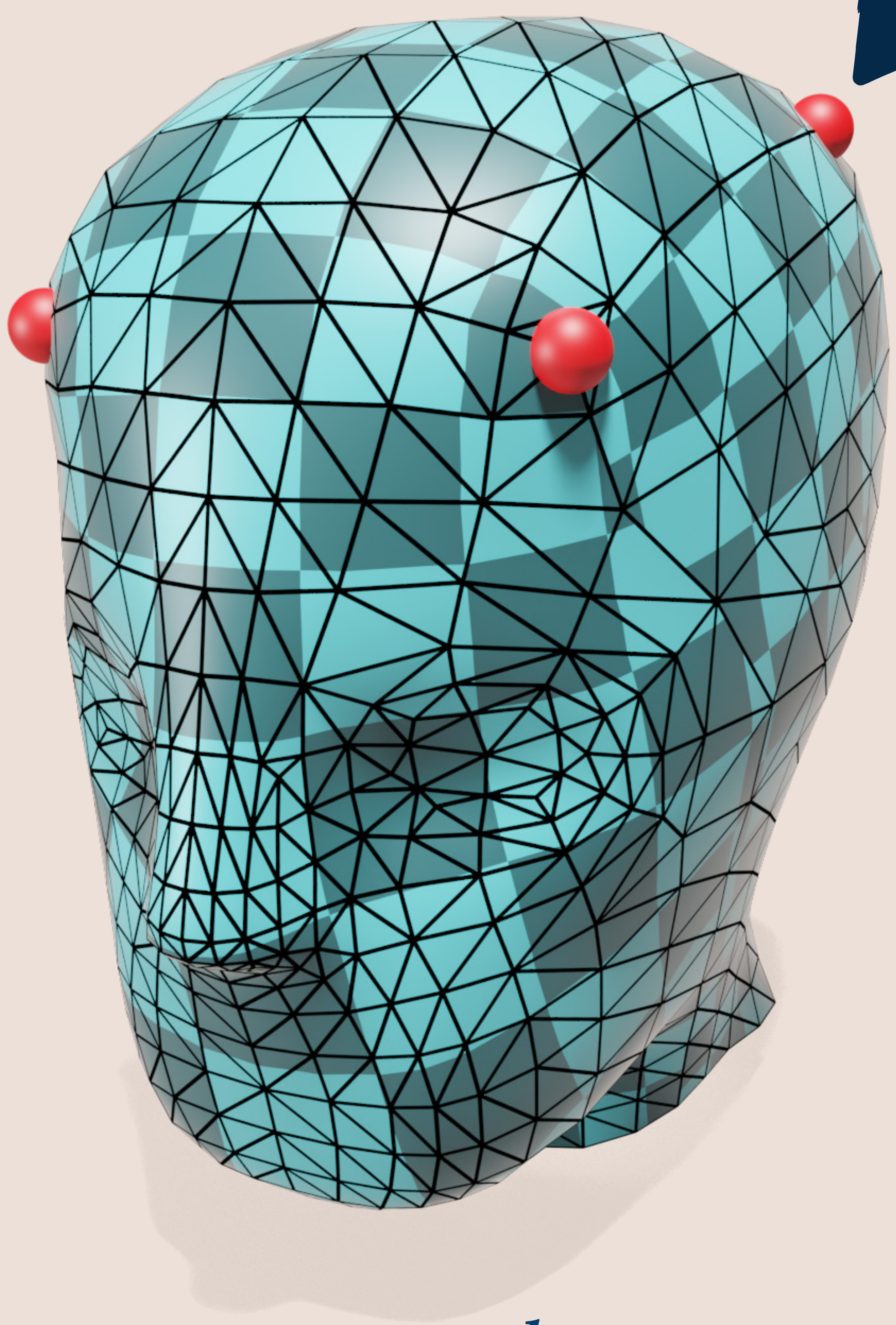


output parameterization

Goal: high-quality surface parameterization

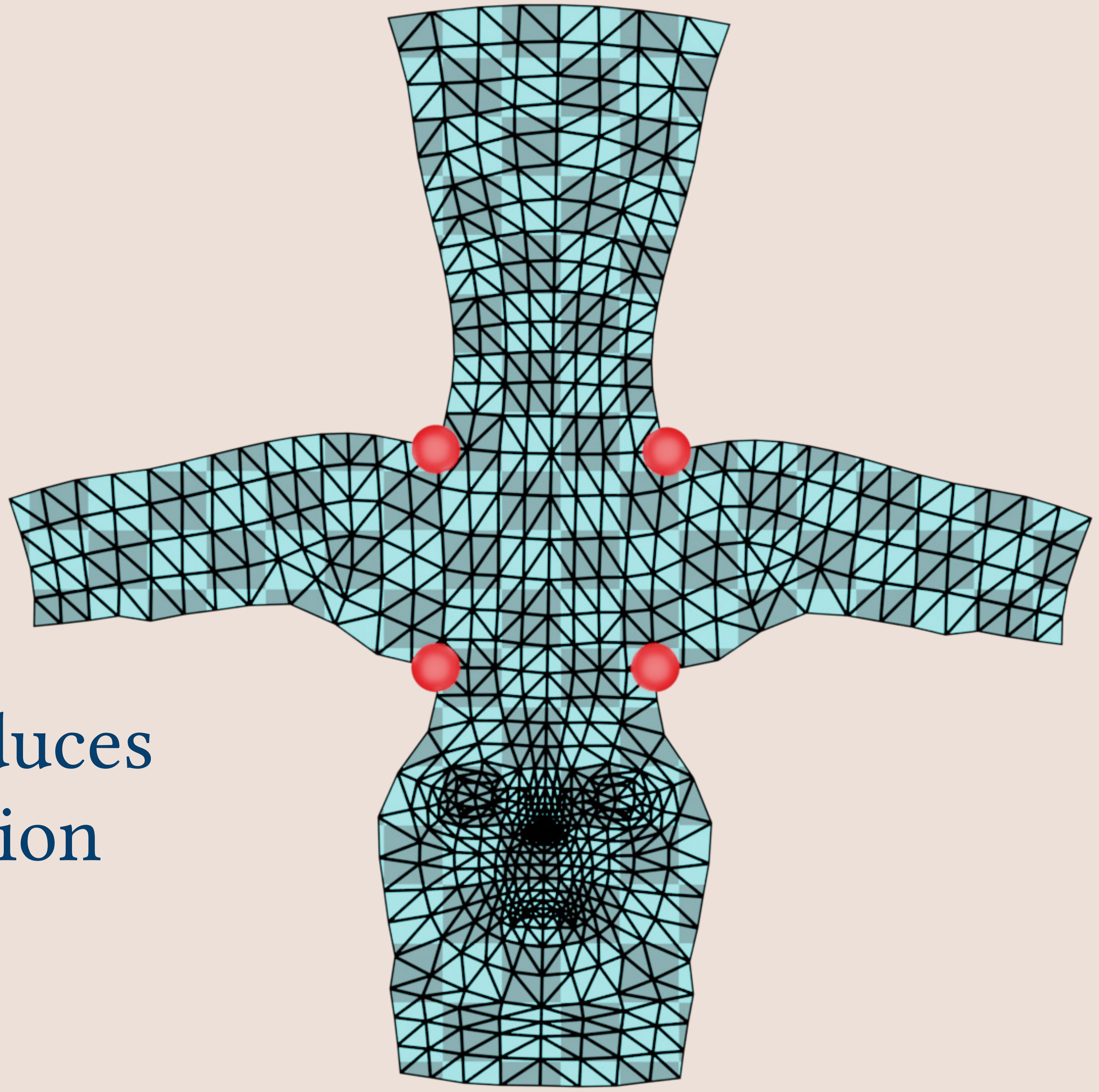
use *cone flattening*

Cones reduce area distortion



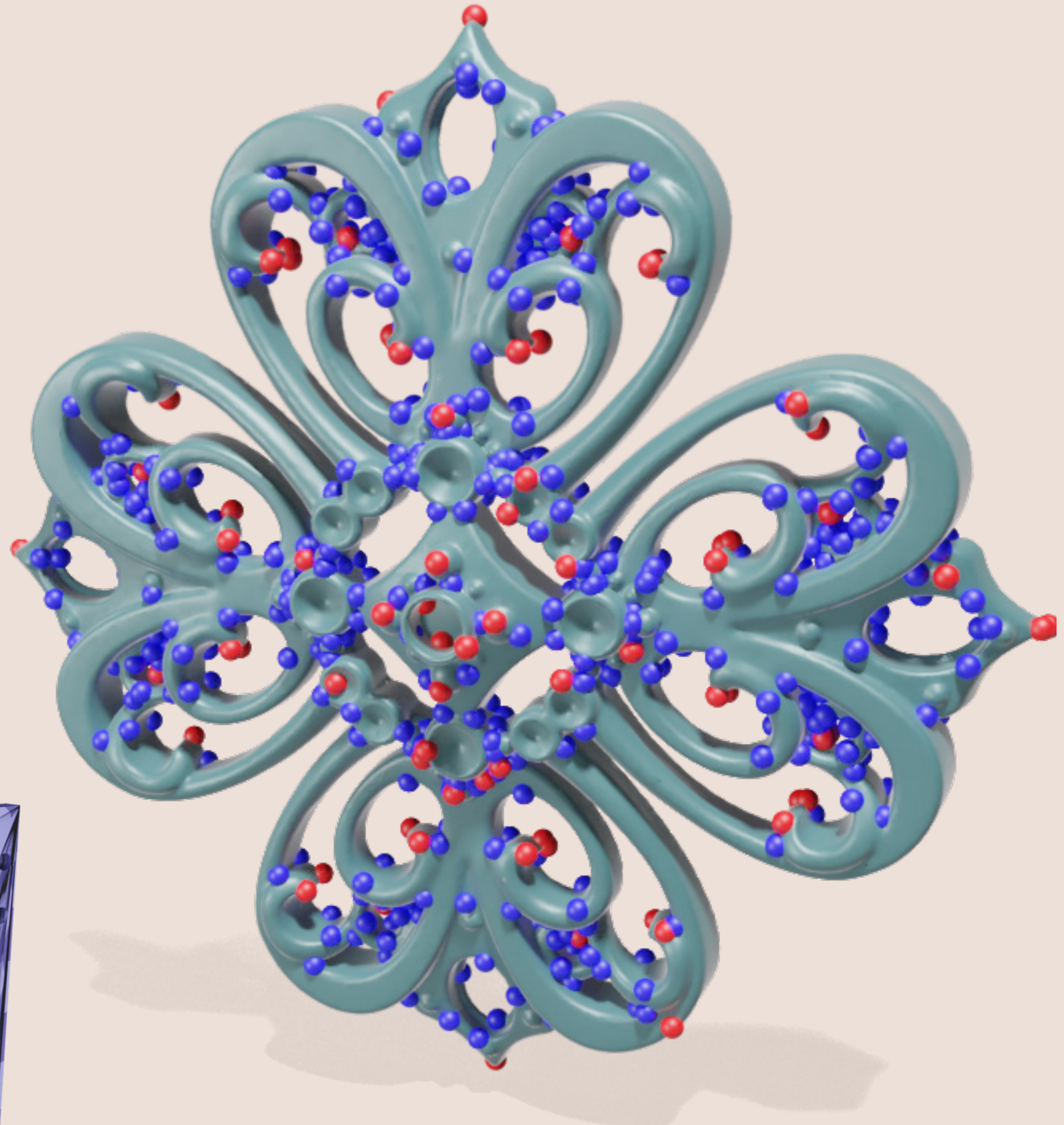
input mesh

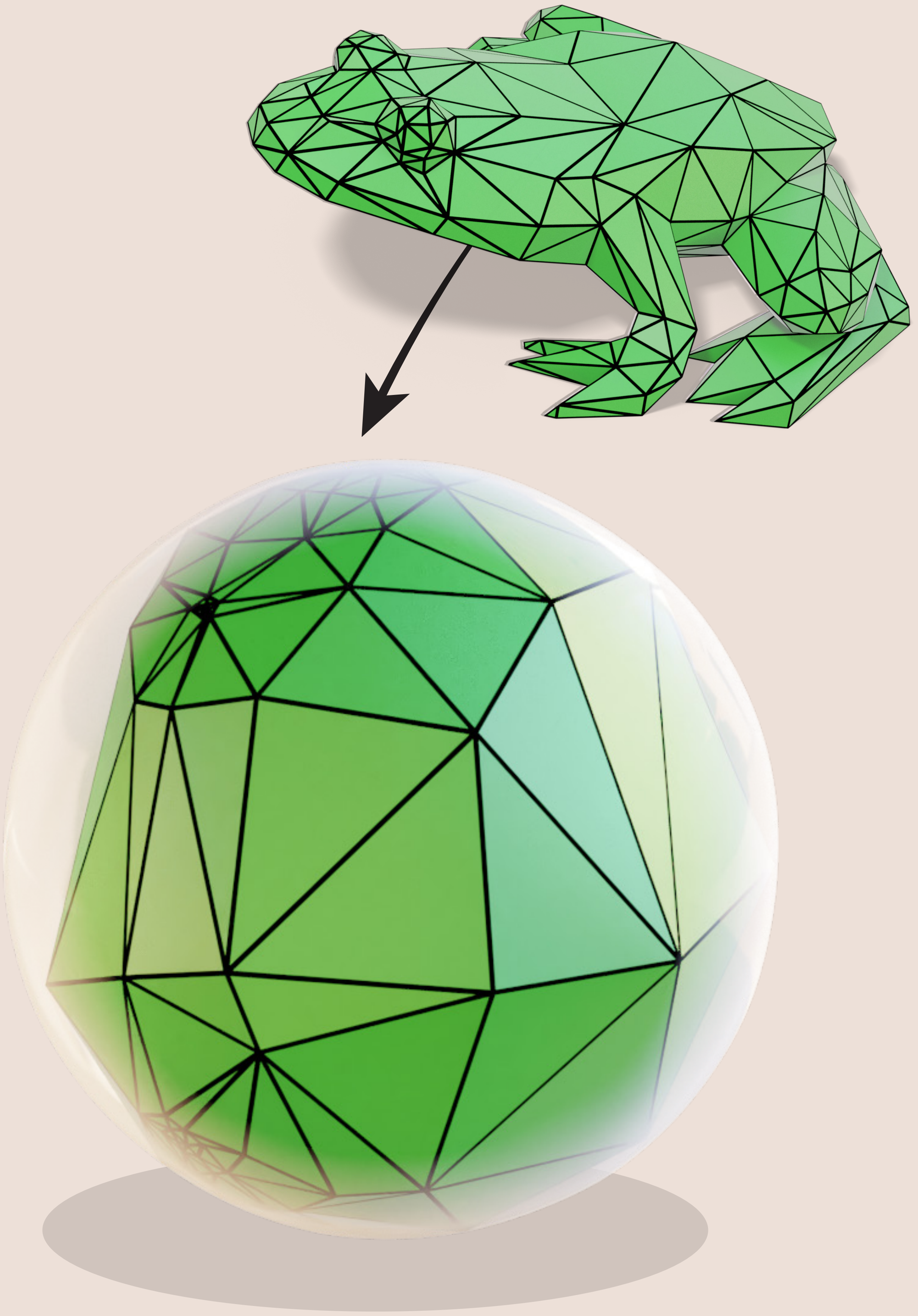
Conformal reduces angle distortion



output parameterization

Why is this hard?



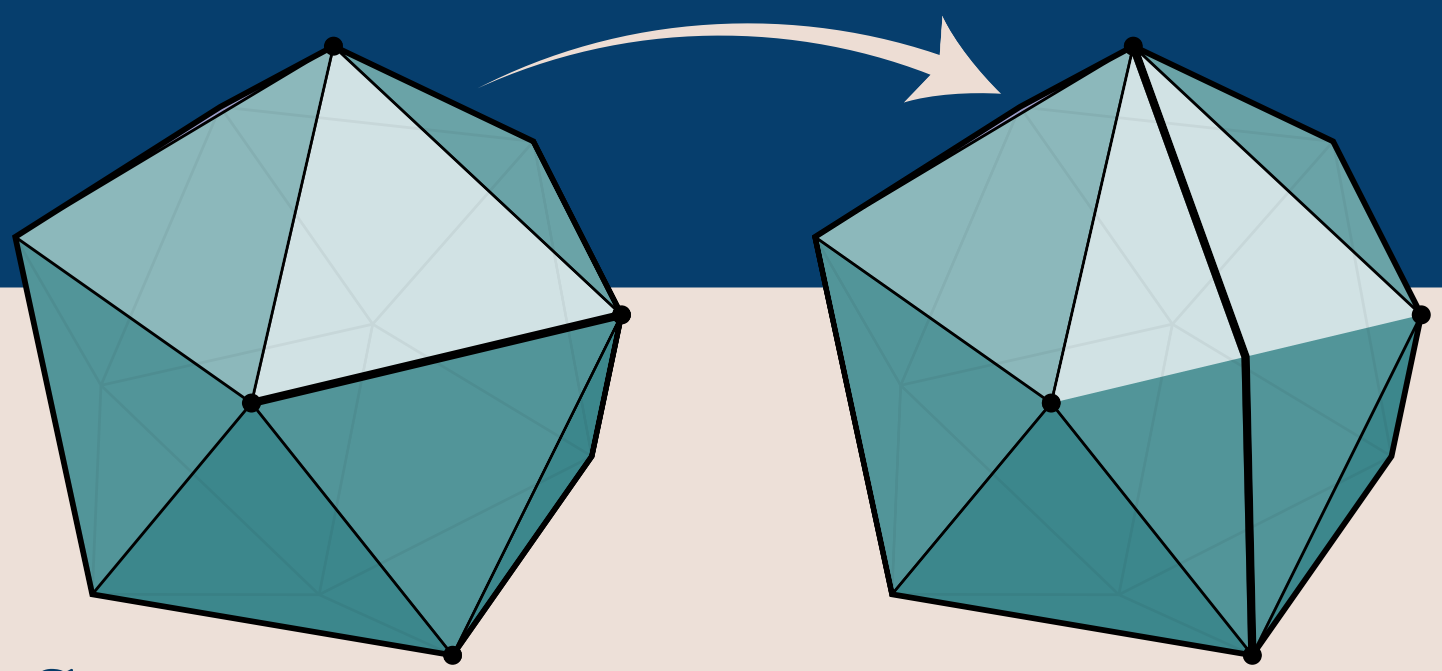


Reliable surface parameterization

via the *discrete uniformization theorem*

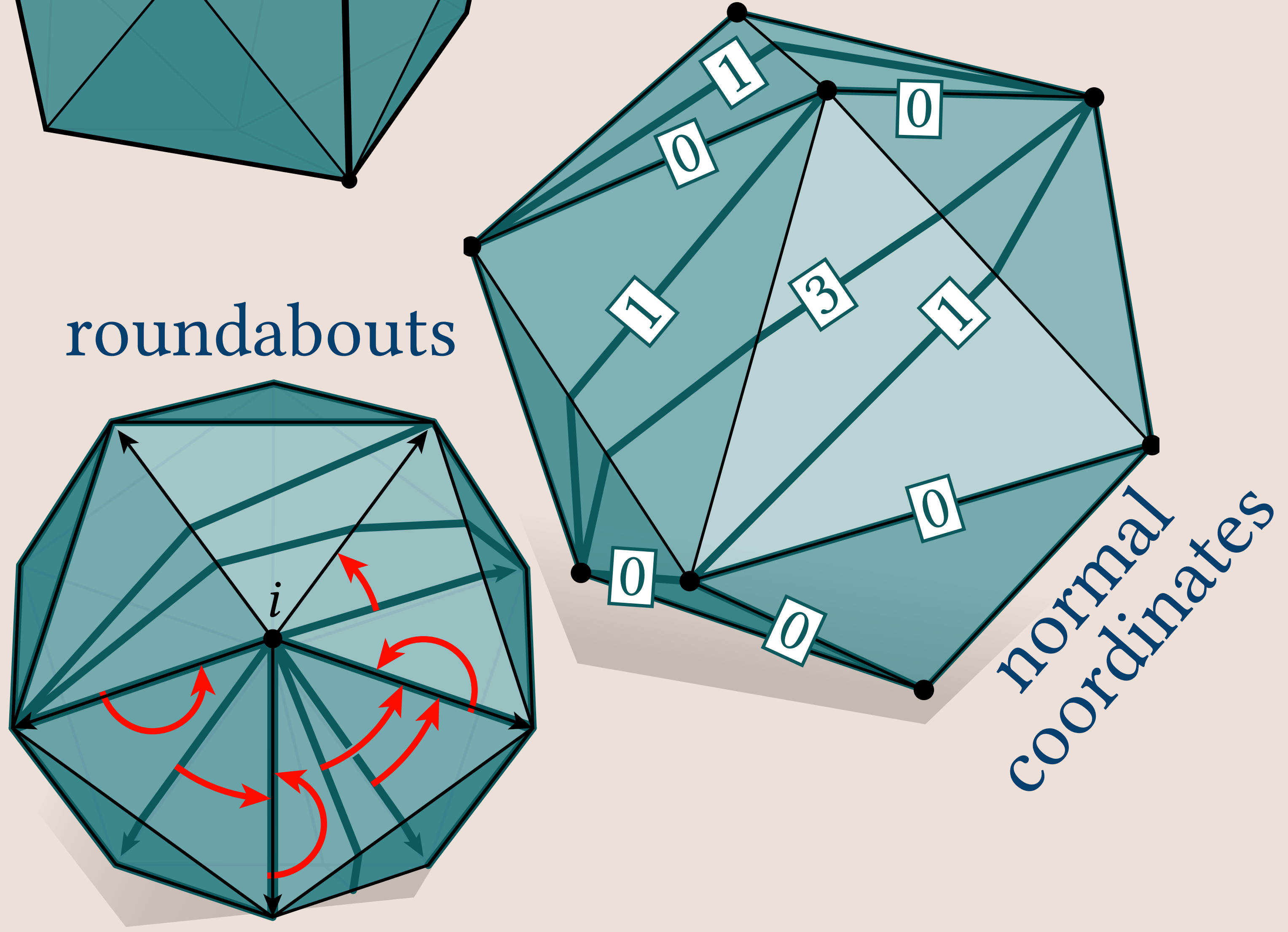
bad meshes | difficult cones | spherical maps

Contributions

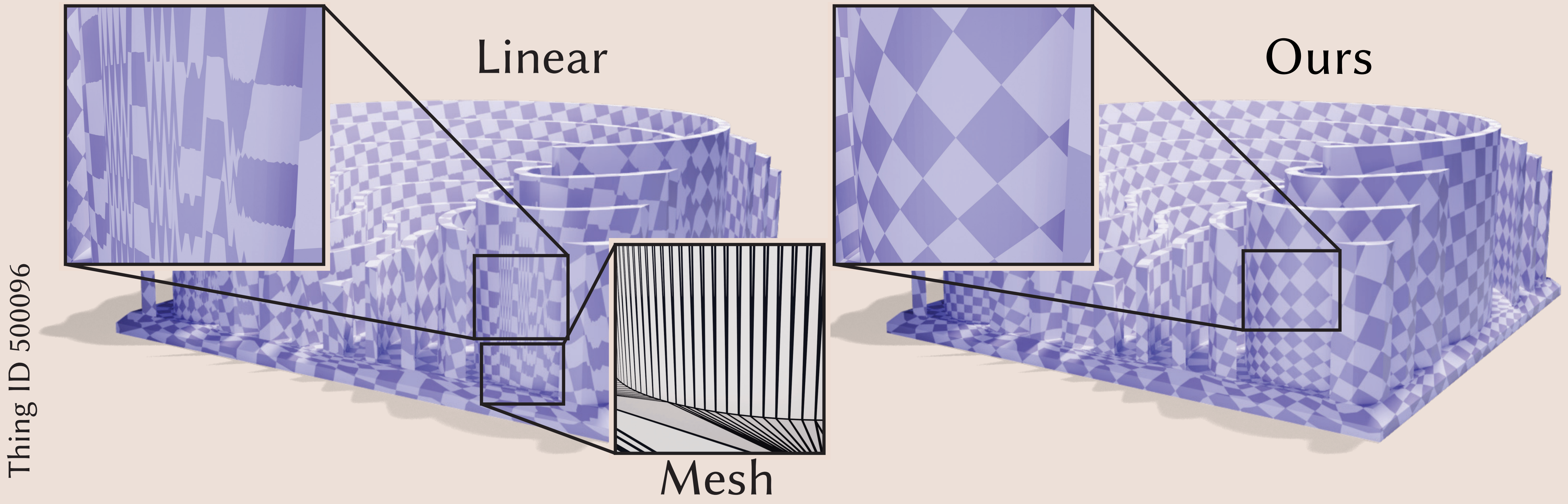


- Generalize CETM [Springborn+ 2008]

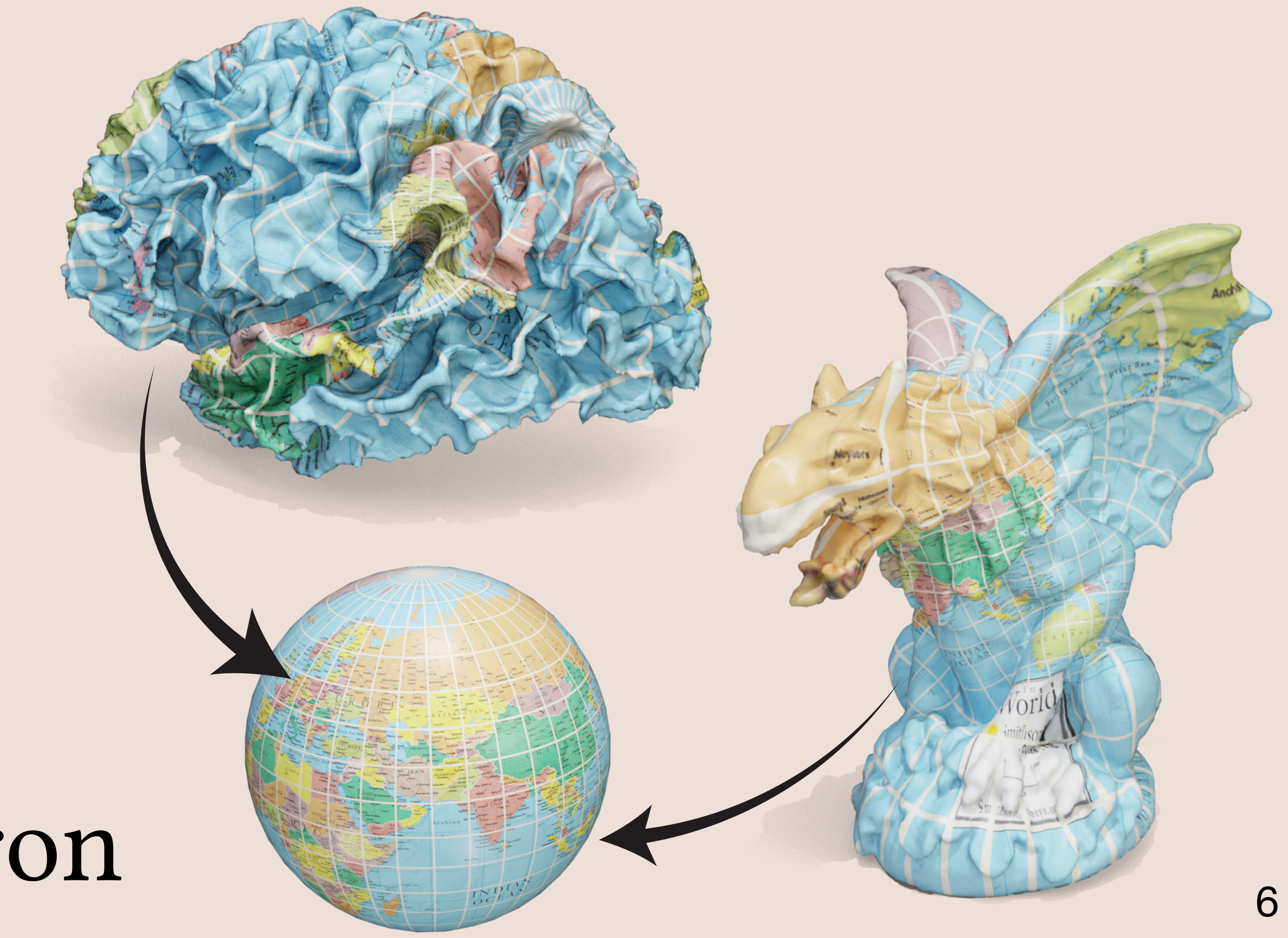
1. Change mesh connectivity ⟶ use Ptolemy flips
   - Ensures that we find a valid parameterization



2. Correspondence ⟶ normal coordinates & roundabouts
3. Interpolation ⟶ calculate in the hyperboloid model



4. Spherical case (guaranteed)
   - Discrete conformal map to convex, sphere-inscribed polyhedron

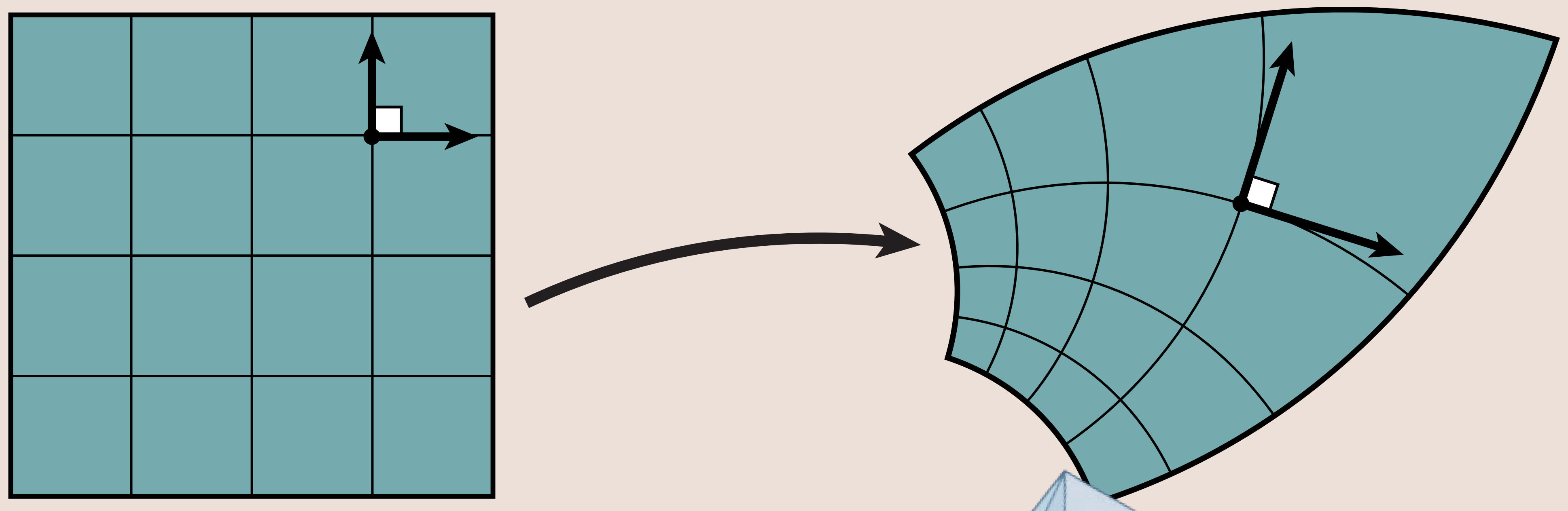


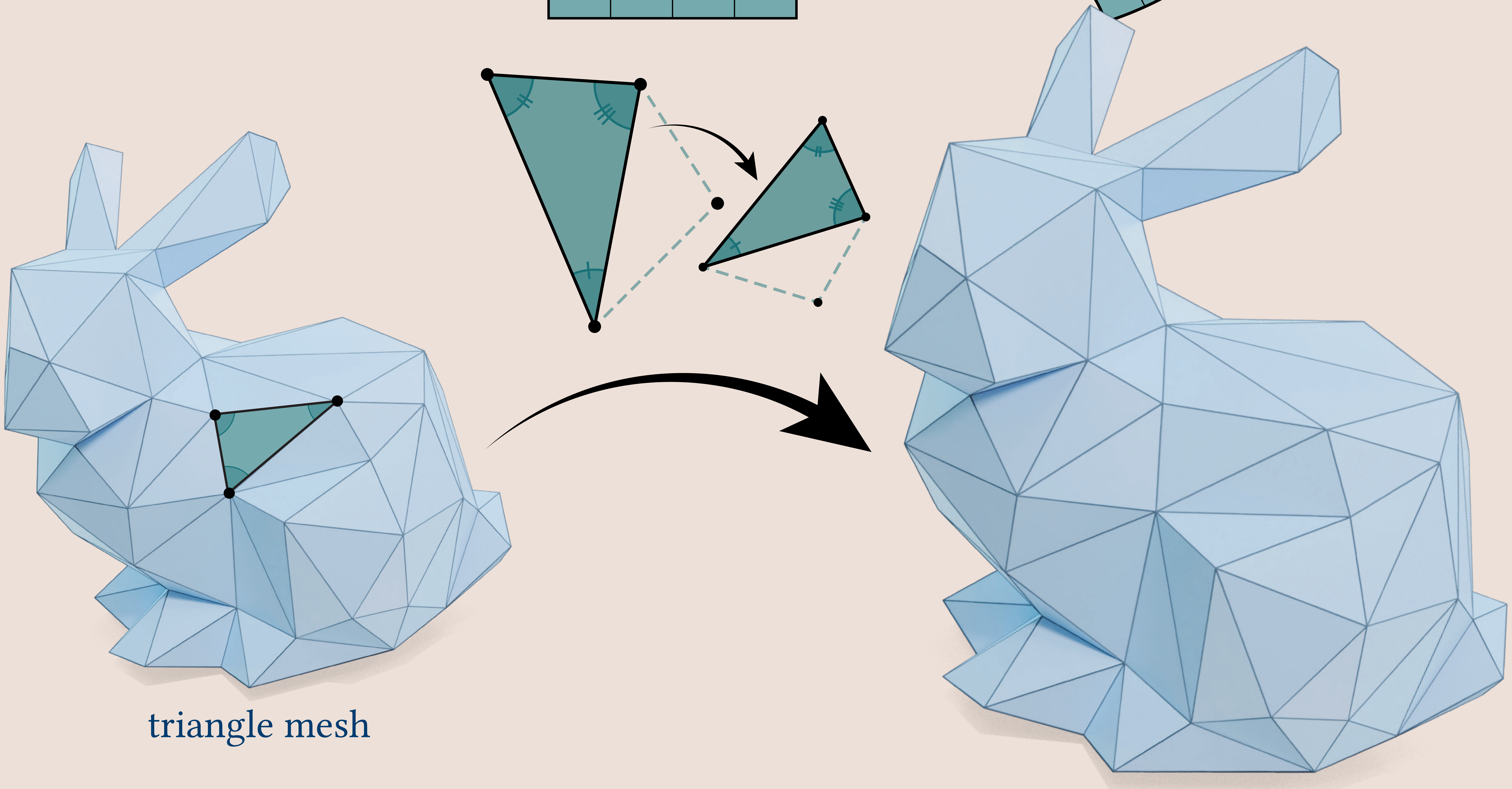
Optimization

with Ptolemy flips

What is a discrete conformal map?

- *"Conformal maps preserve angles"*
  - Really easy to apply to meshes

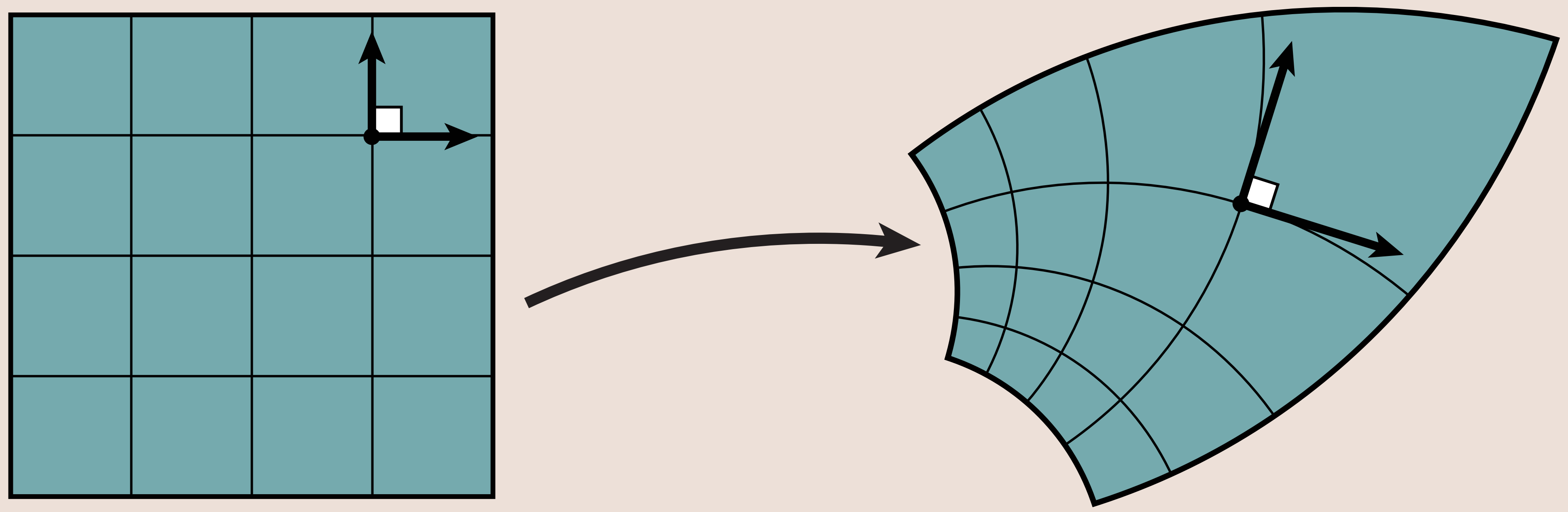




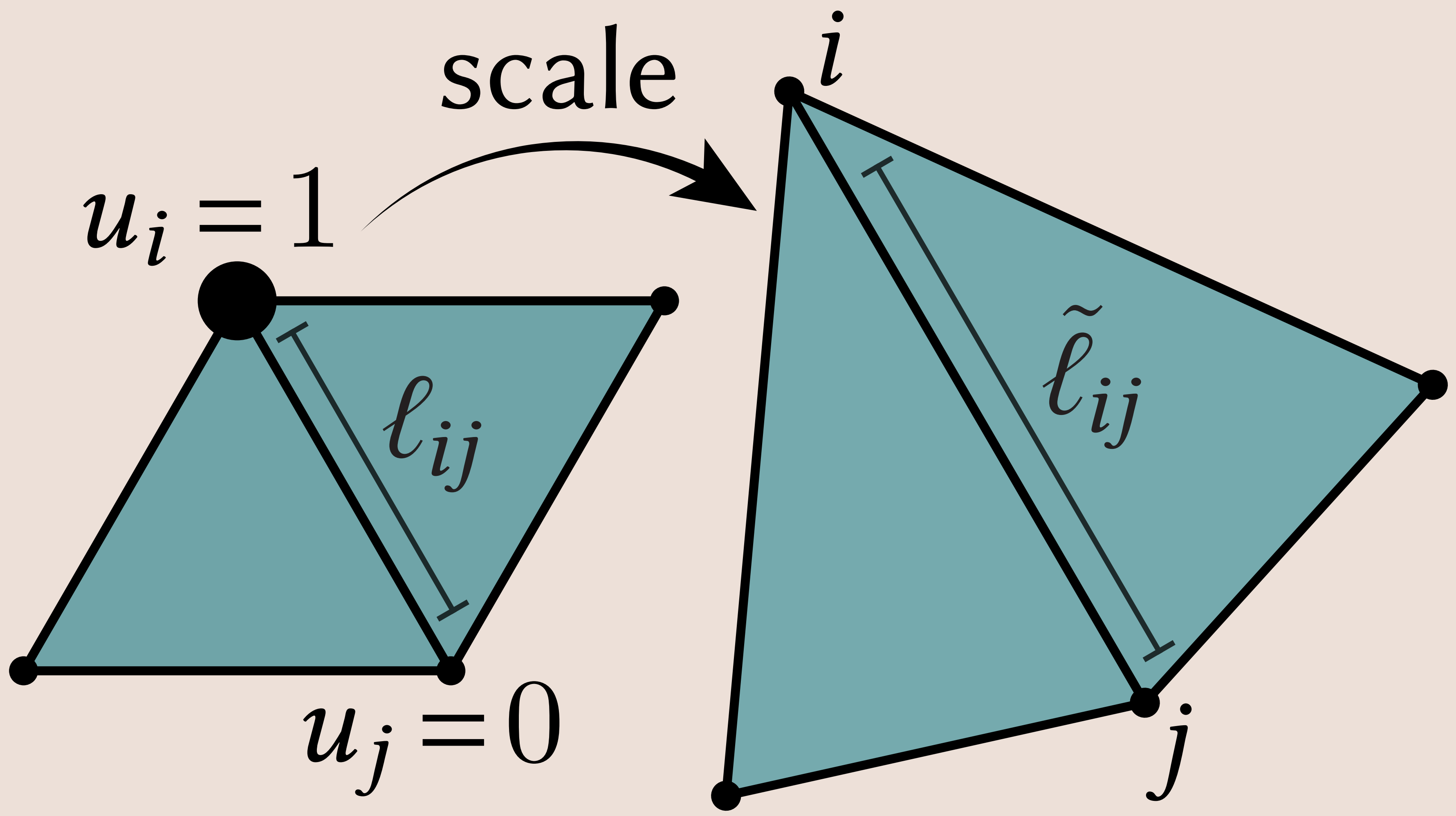
triangle mesh

What is a discrete conformal map?

- ~~*"Conformal maps preserve angles"*~~
  - Too strict

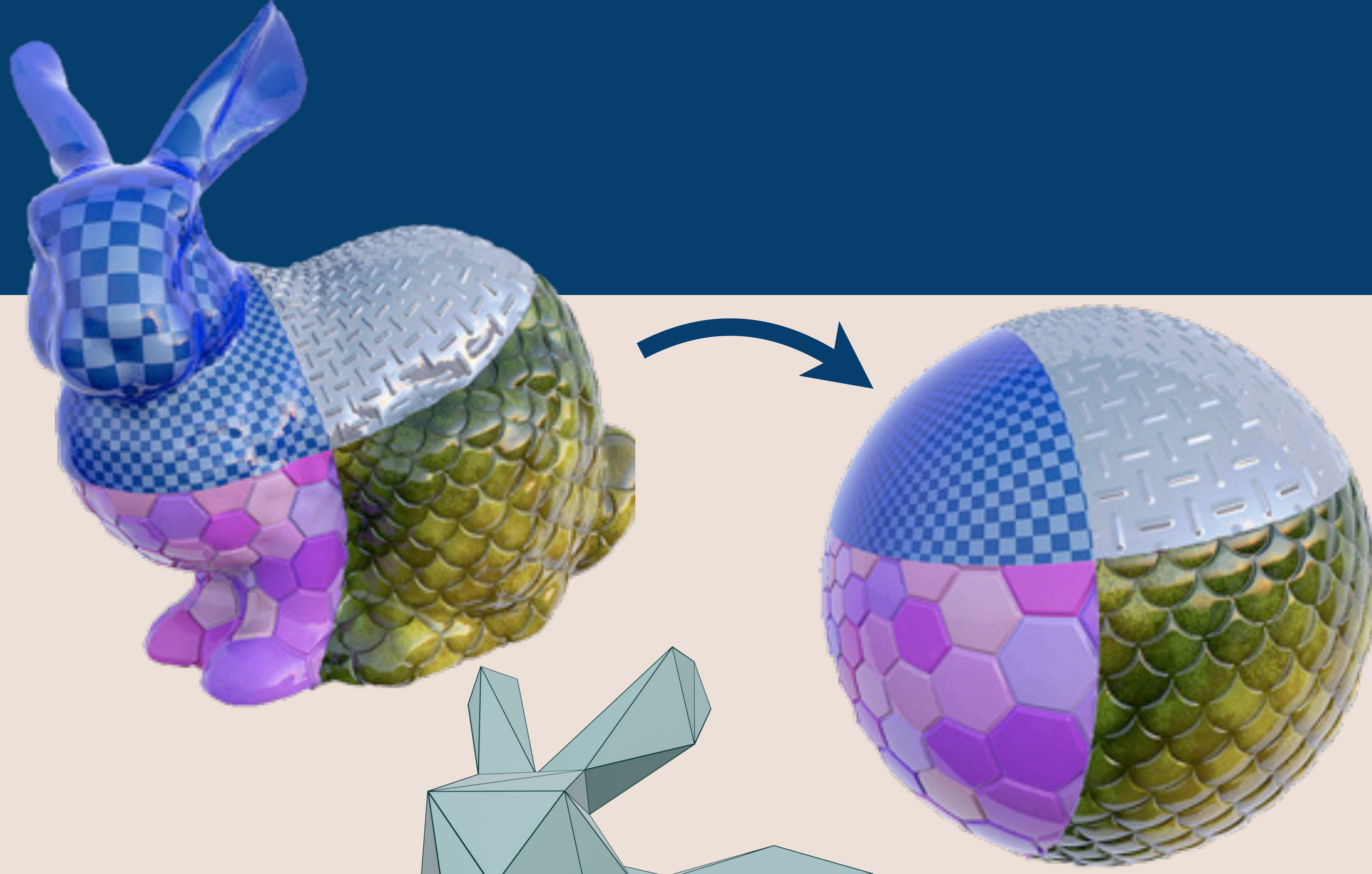


- Metric scaling
  - Locally, a conformal map just scales
- Discrete analogue: *vertex scaling*
  - log scale factor $u : V \to \mathbb{R}$
  - $\tilde{\ell}_{ij} = e^{(u_i+u_j)/2}\ell_{ij}$
  - Captures rich mathematical theory [Bobenko+ 2011]

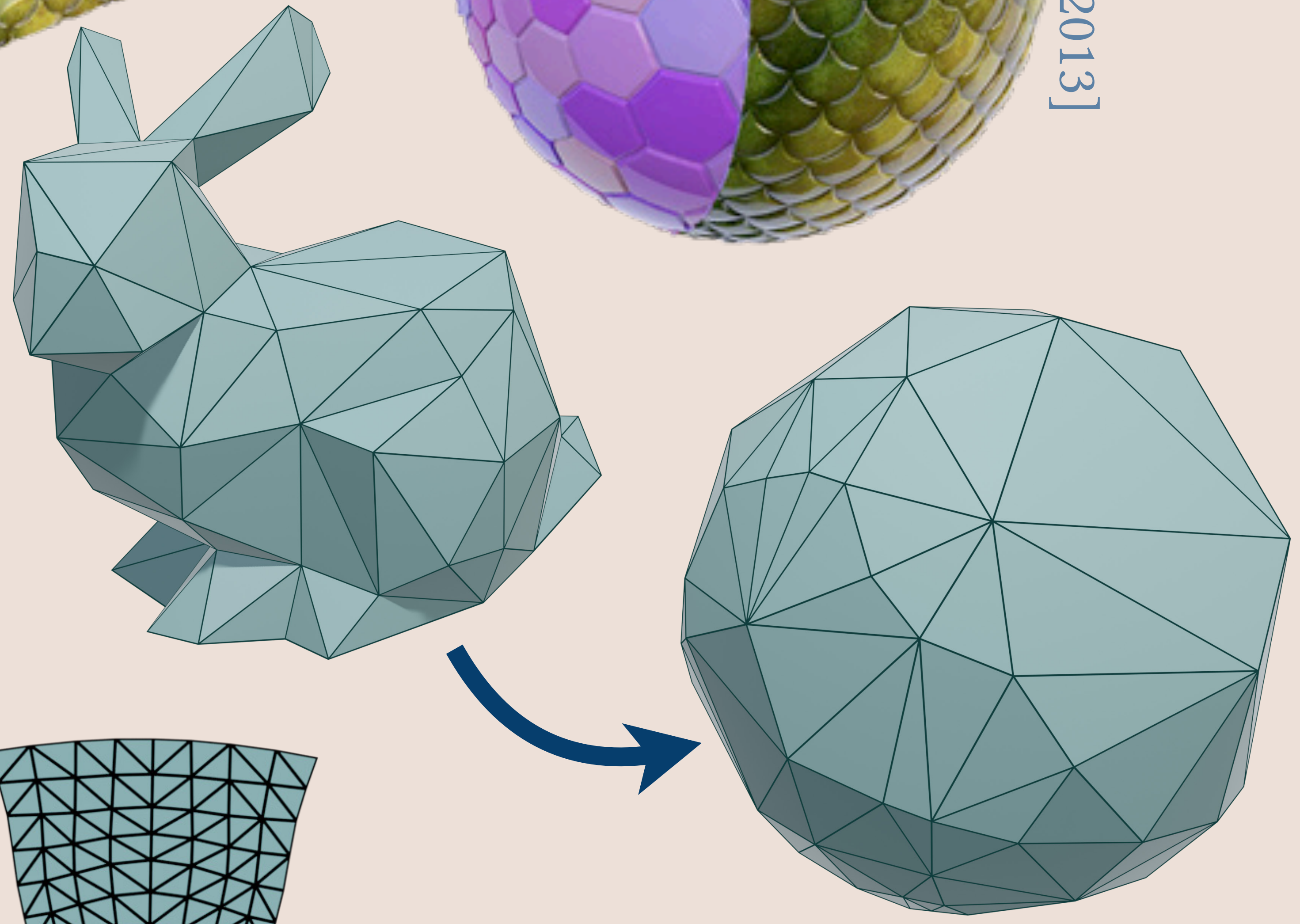


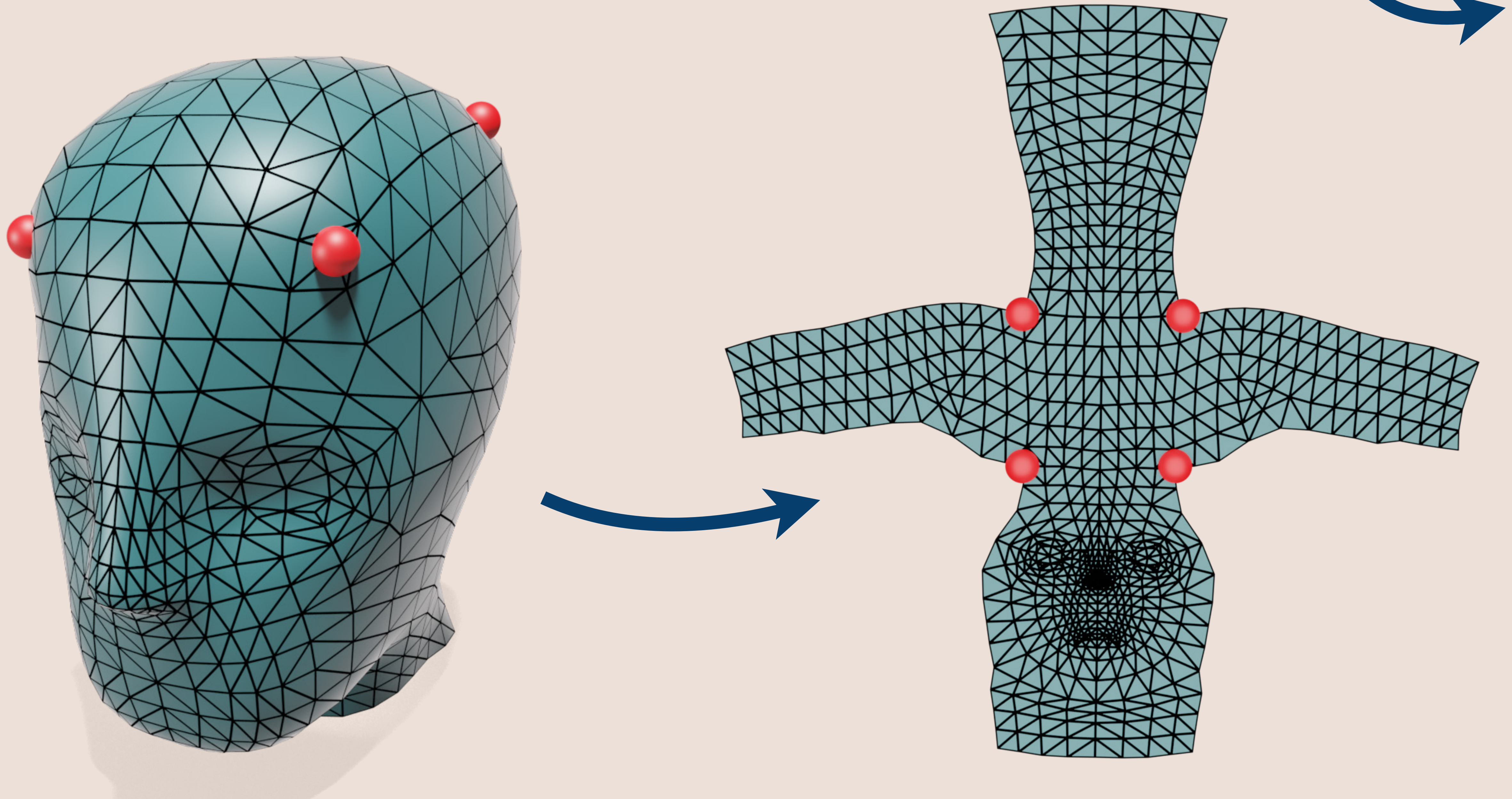
Uniformization

- Smooth uniformization [Poincaré 1907; Abikoff 1981]
  - *Any surface can be conformally mapped to one of constant curvature*
- Discrete uniformization [Gu+ 2018ab; Springborn 2019]
  - *Any triangle mesh can be discretely conformally mapped to one of constant curvature (or any valid target curvature)*
  - Perfect tool for cone flattening

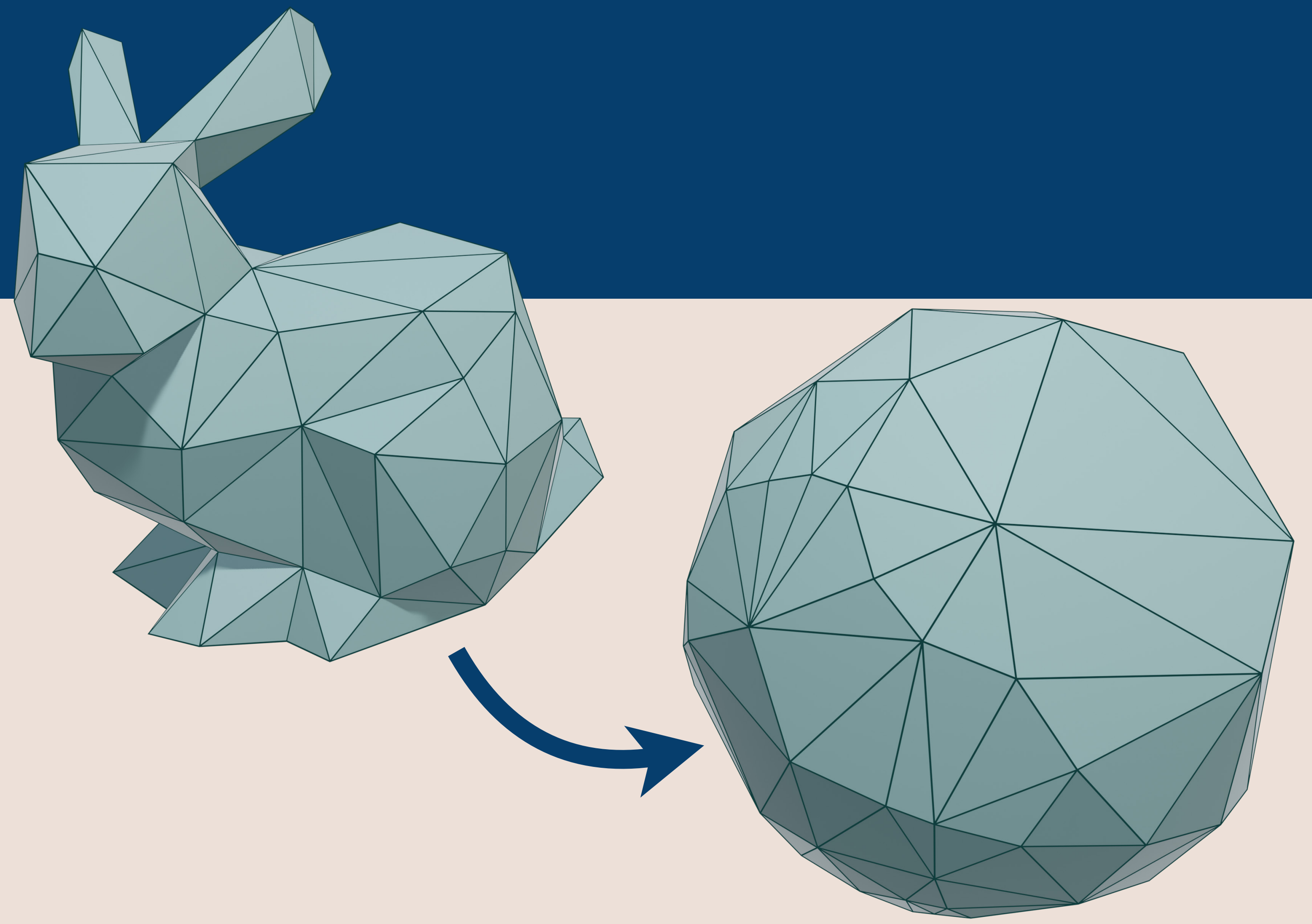


[Crane+ 2013]

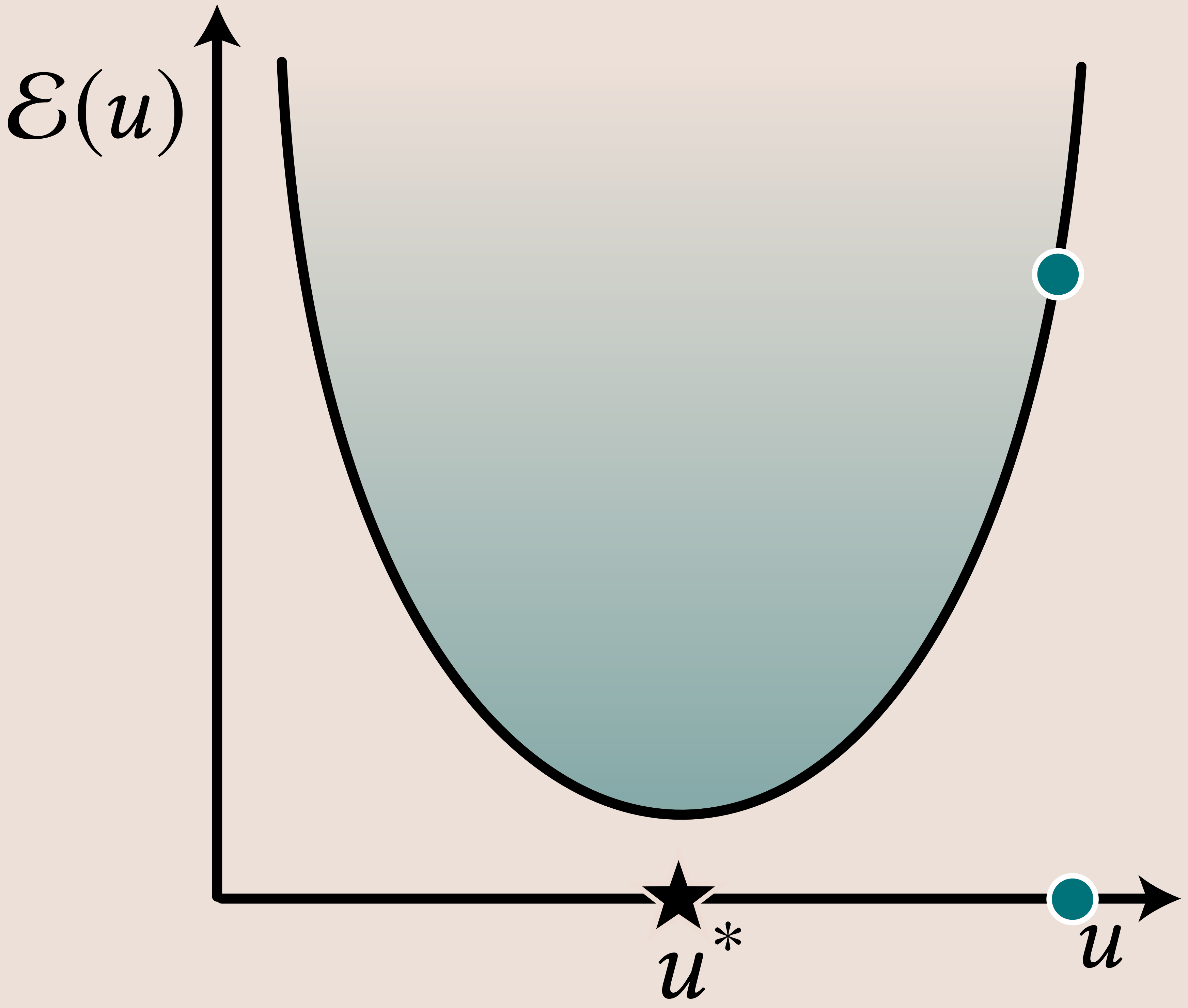




Discrete uniformization



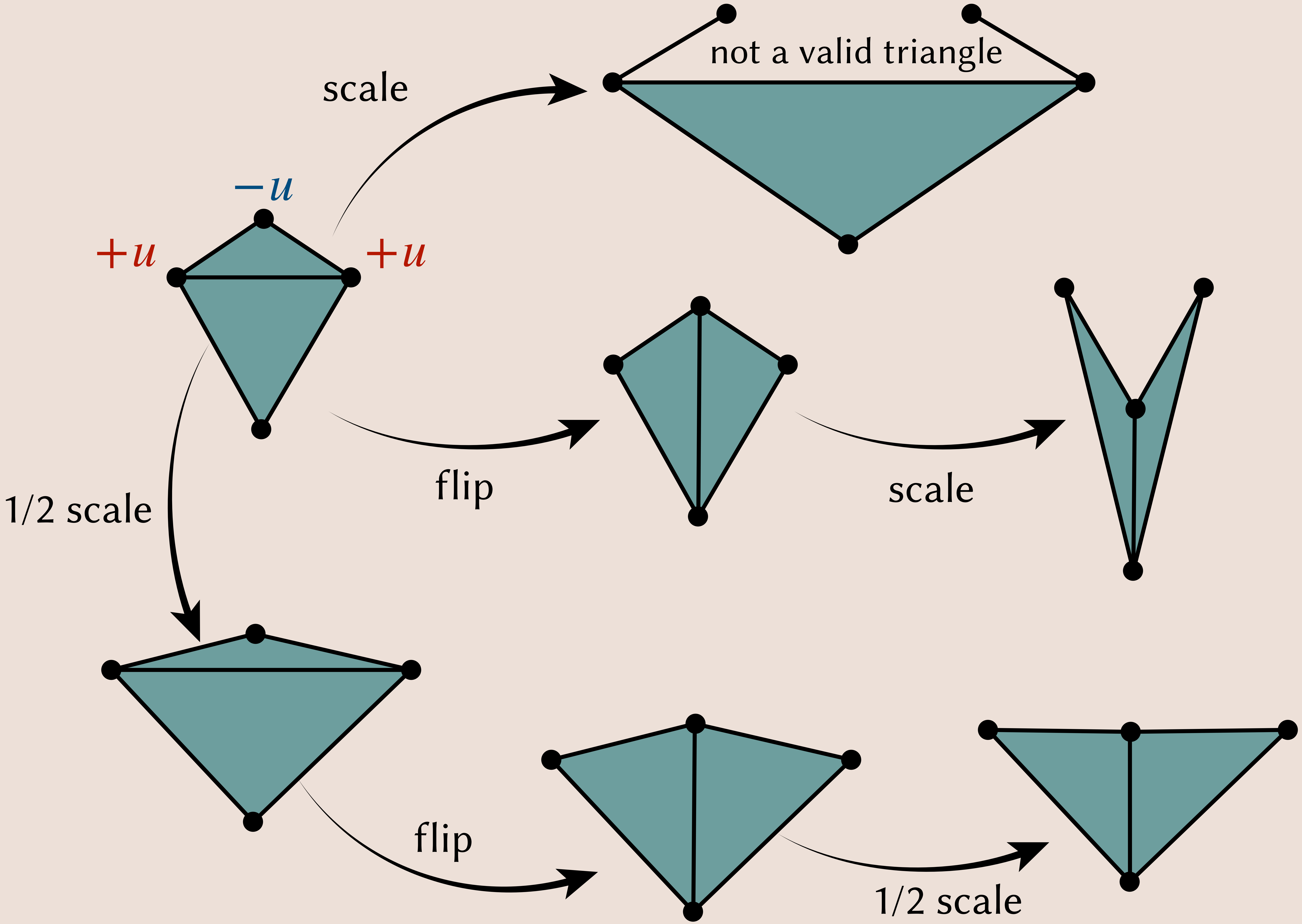
- Discrete uniformization [Gu+ 2018ab; Springborn 2019]
  - *Any triangle mesh can be discretely conformally mapped to one of constant curvature (or any valid target curvature)*
- [Luo 04]: follow flow
- [Springborn+ 2008]: minimize energy



Main idea: find discrete conformal maps by minimizing a convex energy

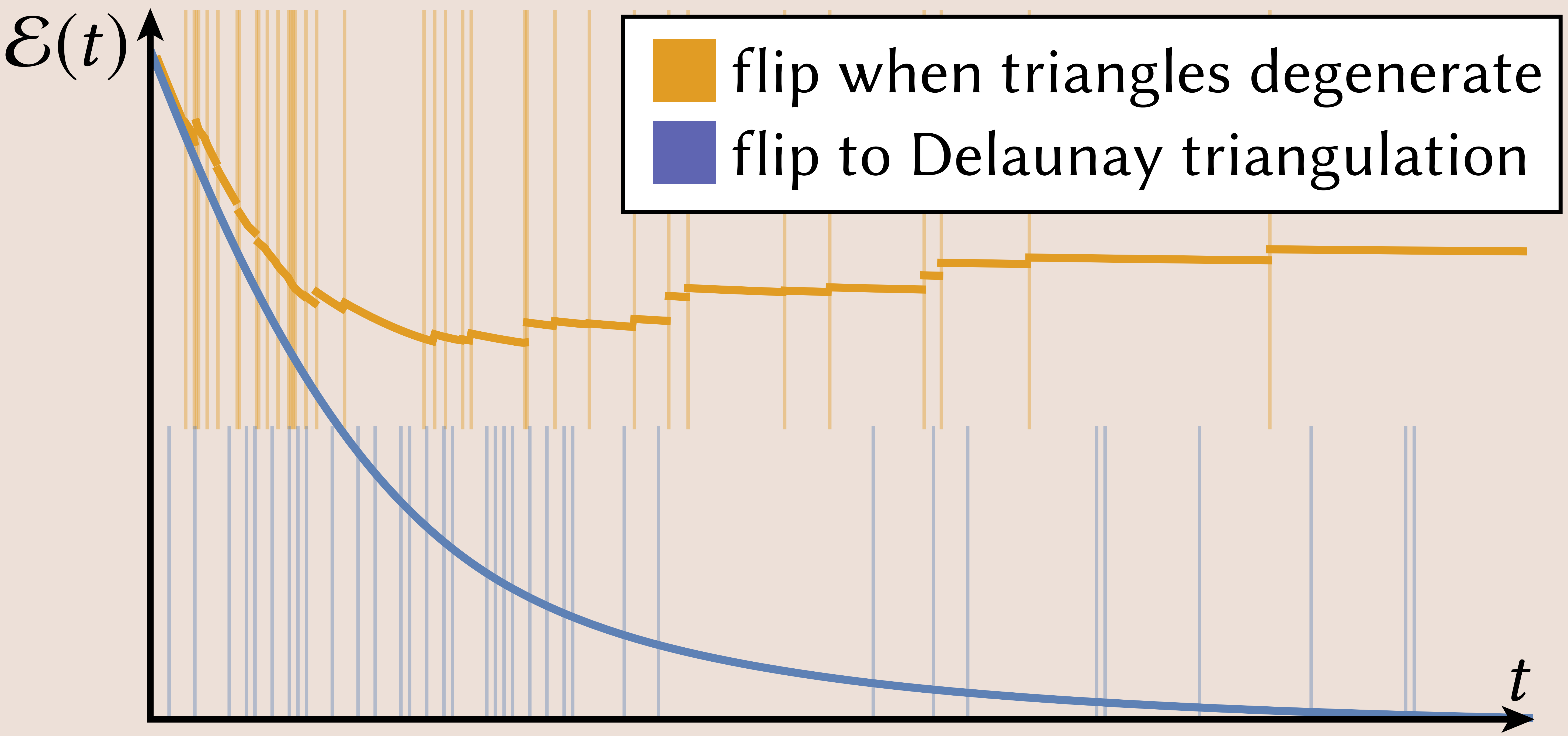
Challenges with discrete uniformization

Discrete uniformization doesn't always work on a fixed mesh because triangles can degenerate



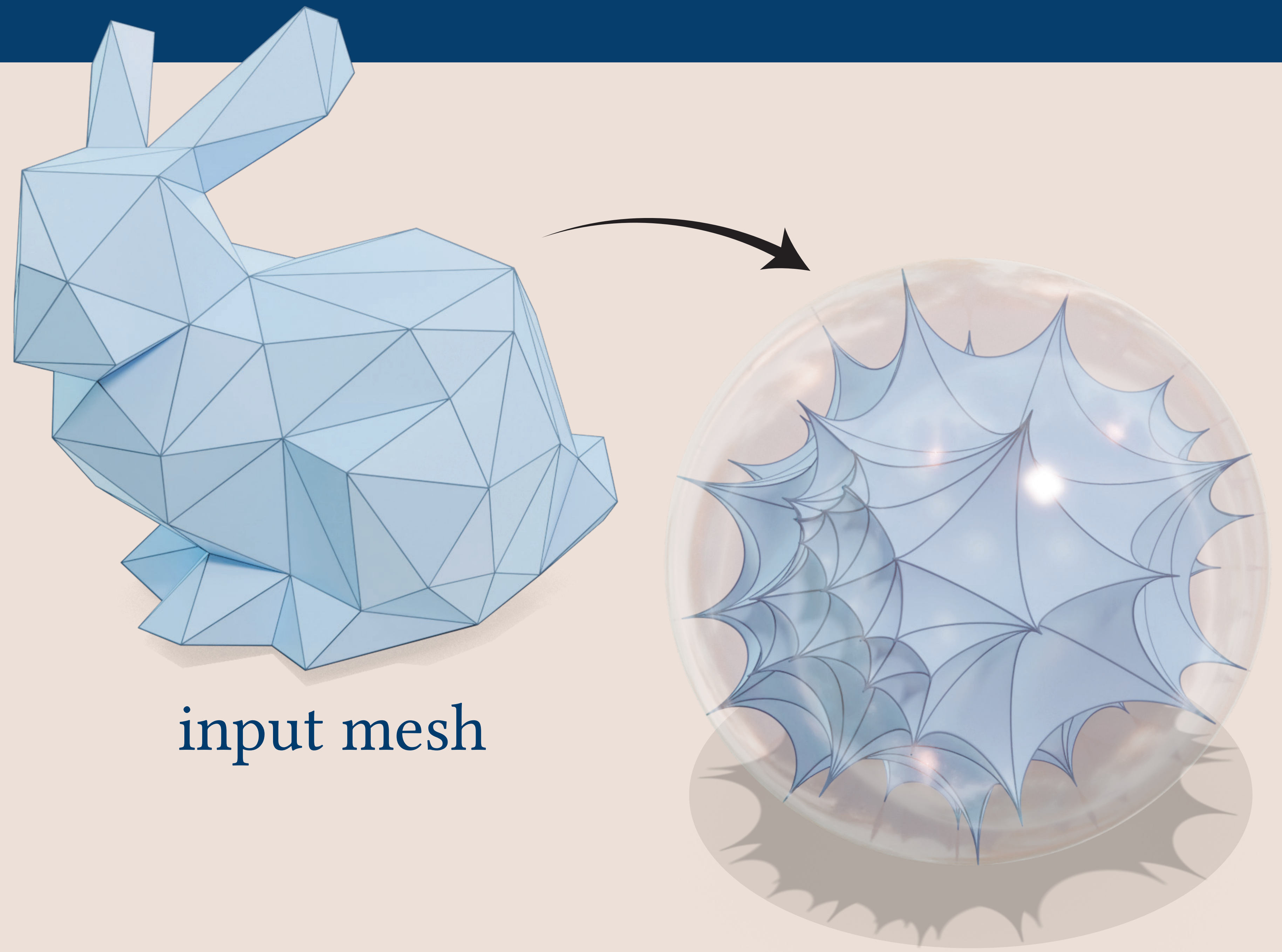
Challenges with discrete uniformization

- Idea: flip edges when triangles break
  - Problem: energy discontinuous at flips (vertical lines)
- [Gu+ 2018a]: maintain Delaunay
  - Problem: stop to flip



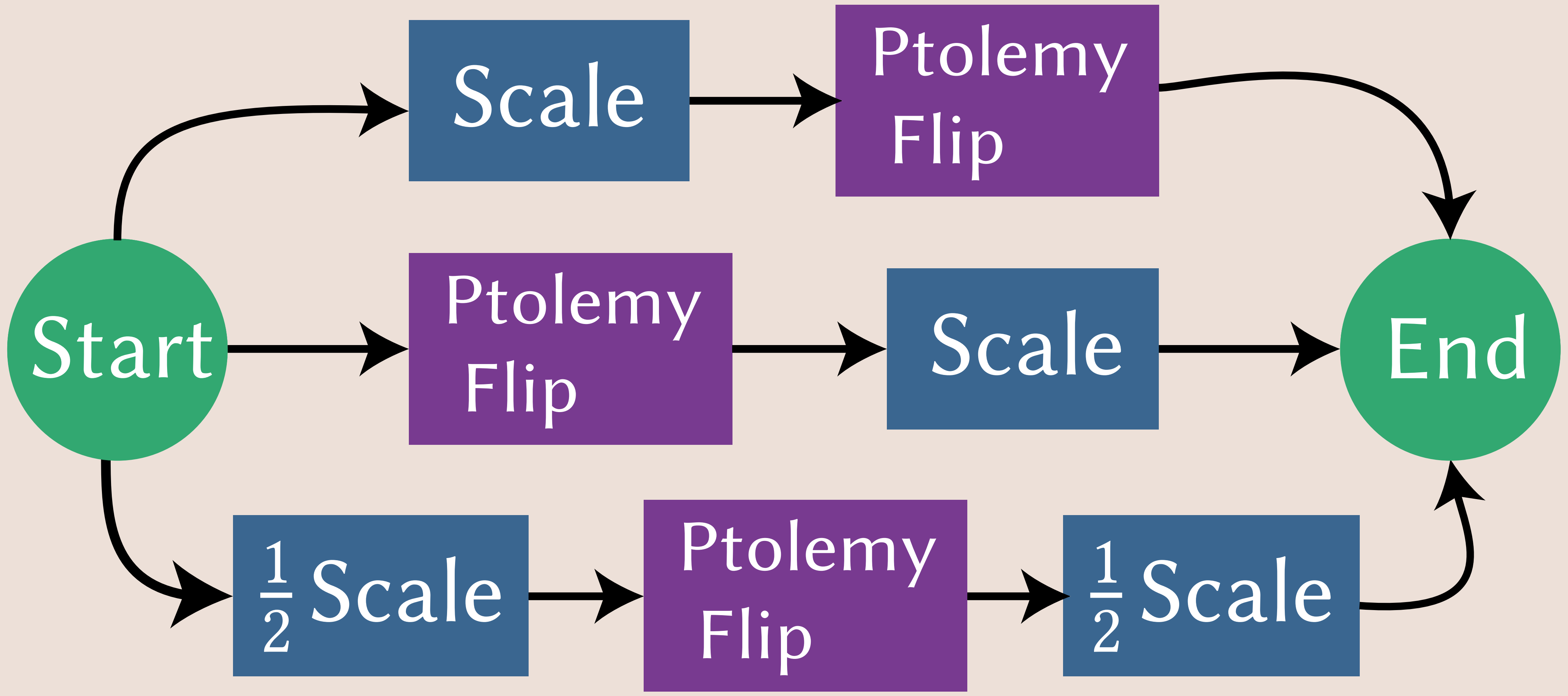
Hyperbolic geometry to the rescue

- Reinterpret mesh as ideal polyhedron [Bobenko+ 2010]
- Compute flipped edge lengths via *Ptolemy's formula*
  - $\ell_{ij} := (\ell_{lj}\ell_{ki} + \ell_{il}\ell_{jk})/\ell_{lk}$
  - "Ptolemy flip"
  - Well-defined for any nonzero edge lengths
- Decouples scaling and flipping [Springborn 2019]



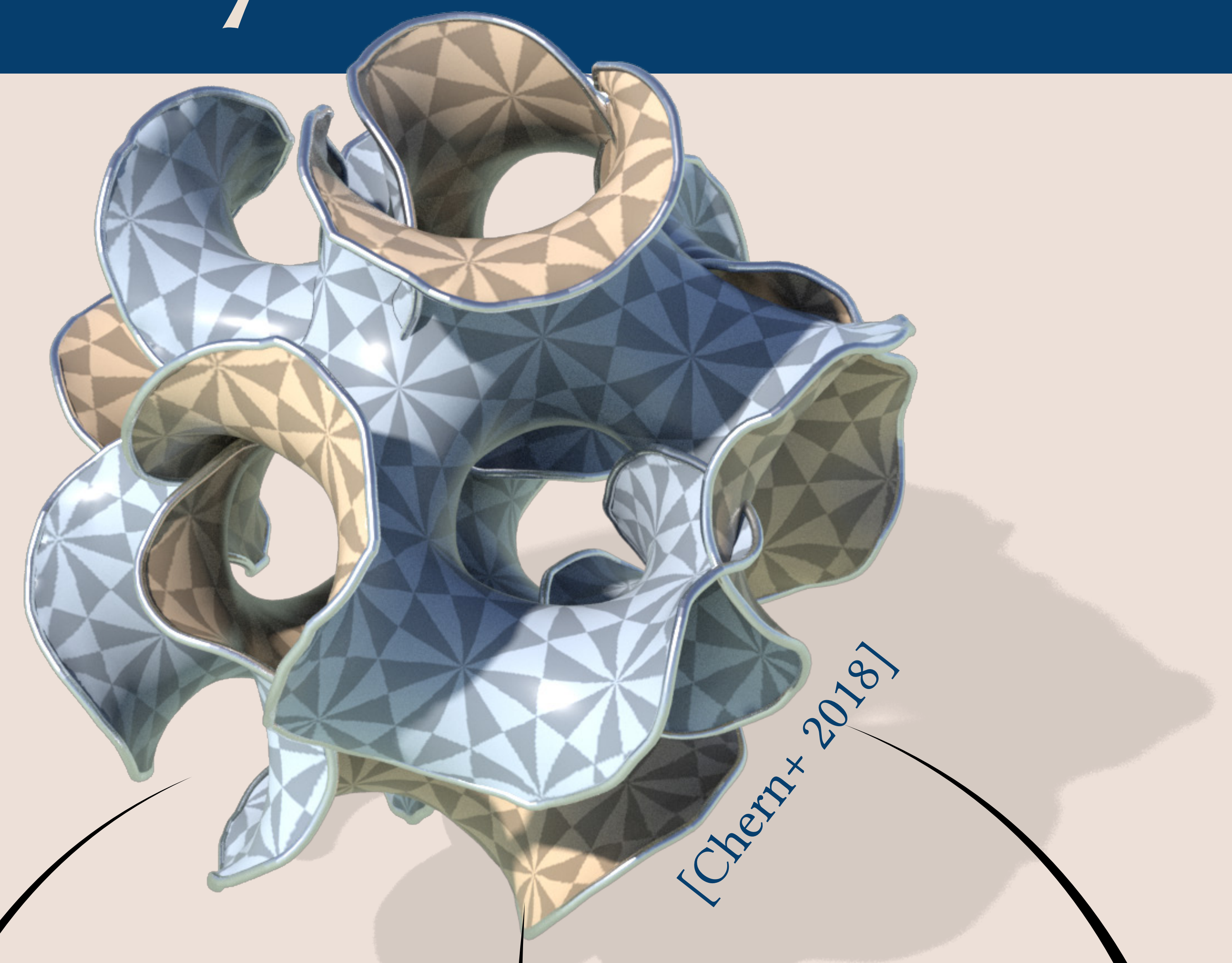
input mesh

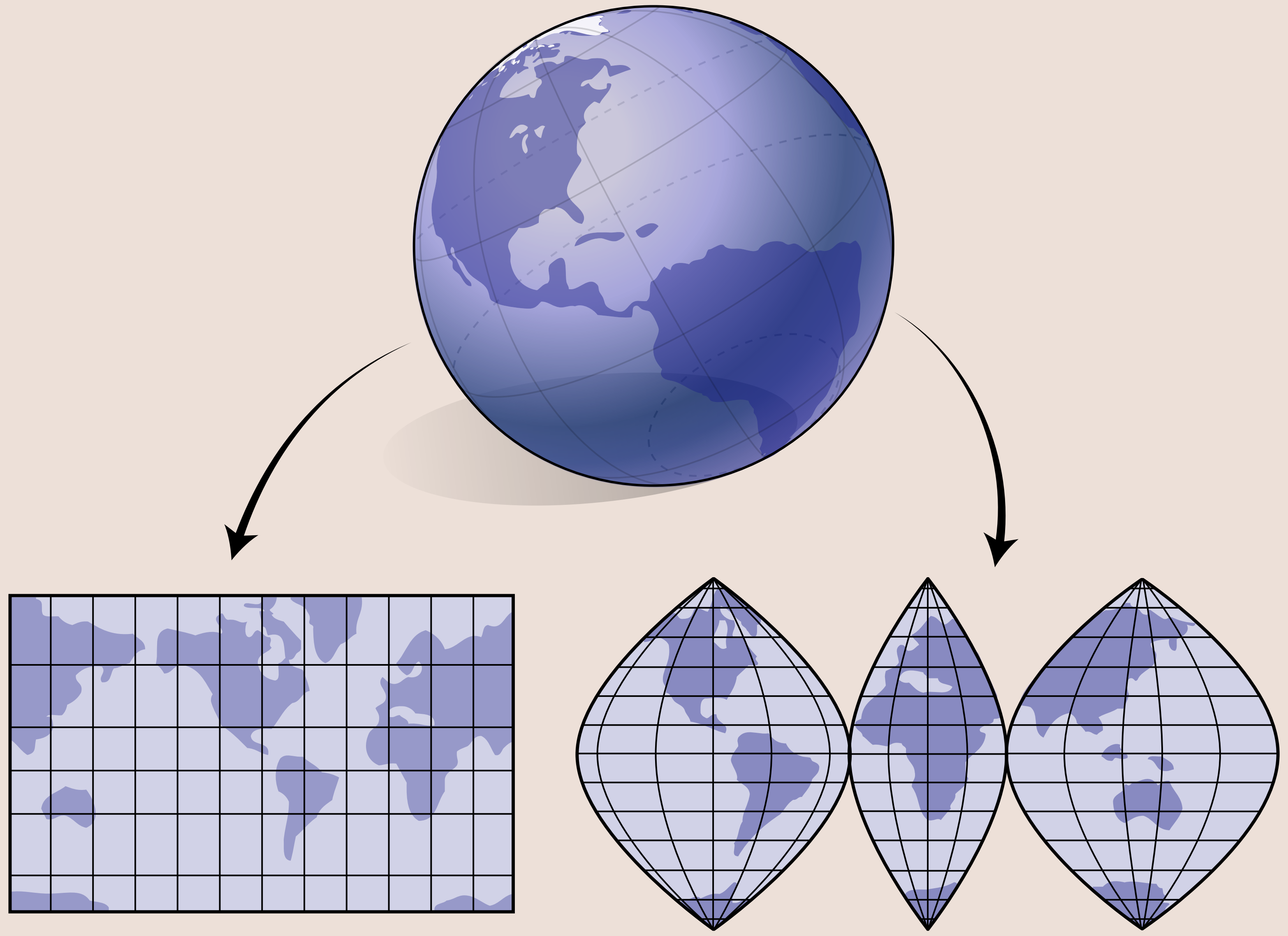
ideal hyperbolic polyhedron

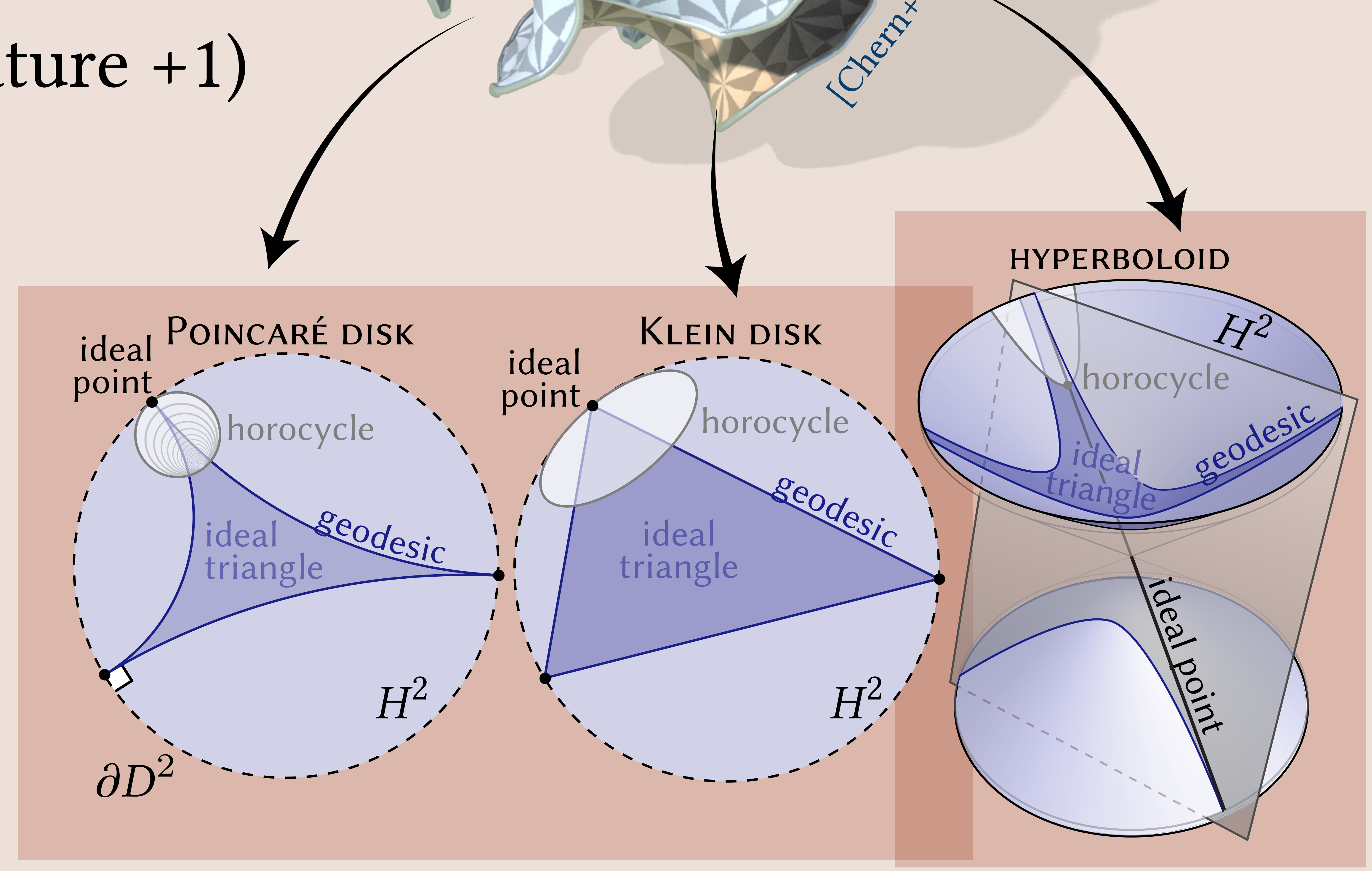


A quick primer on hyperbolic geometry

- Hyperbolic plane
  - Saddle-shaped everywhere
  - *Gaussian curvature* $= -1$
    - (for reference, sphere has constant curvature +1)
- View through models, like maps of the earth

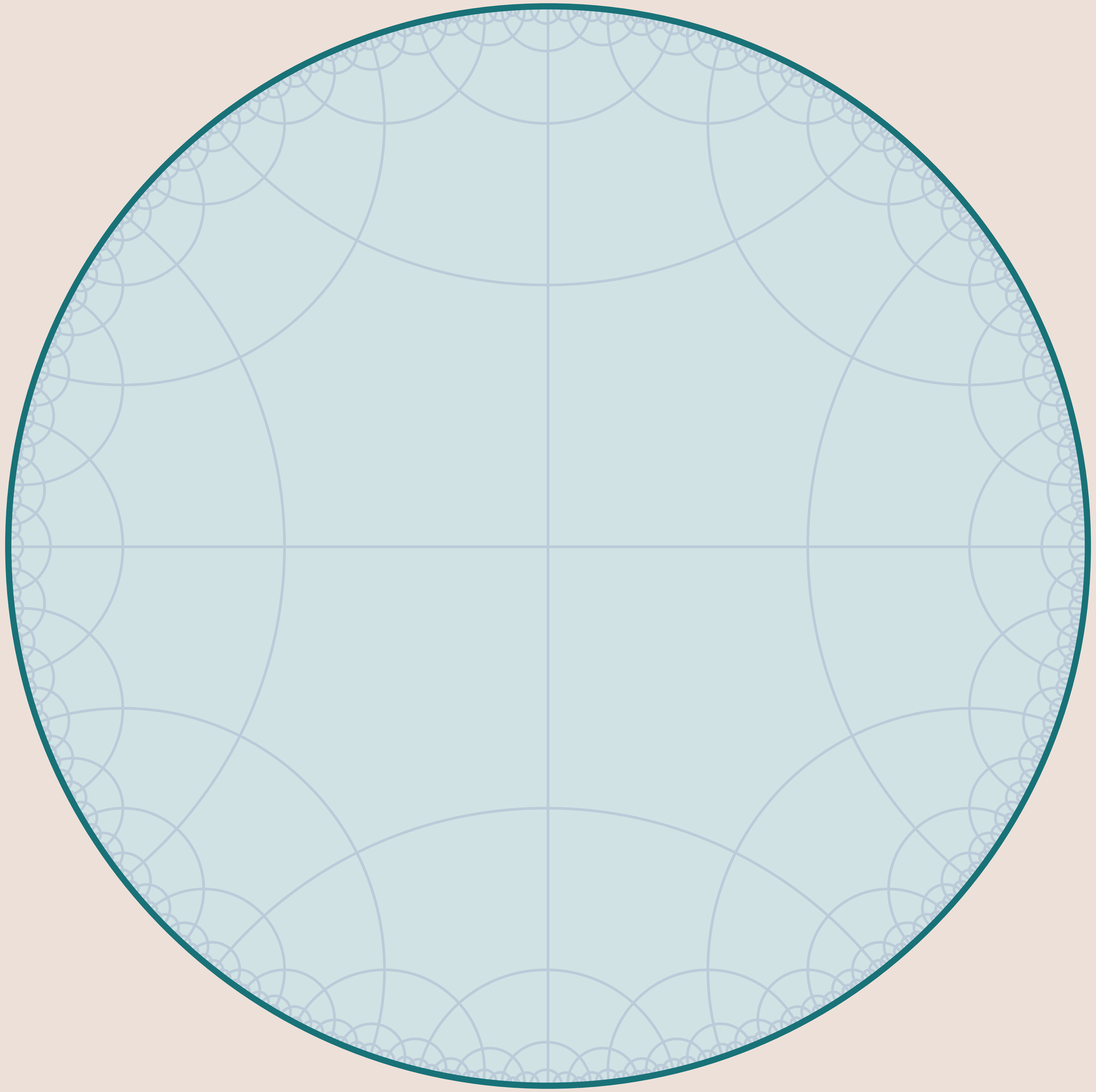






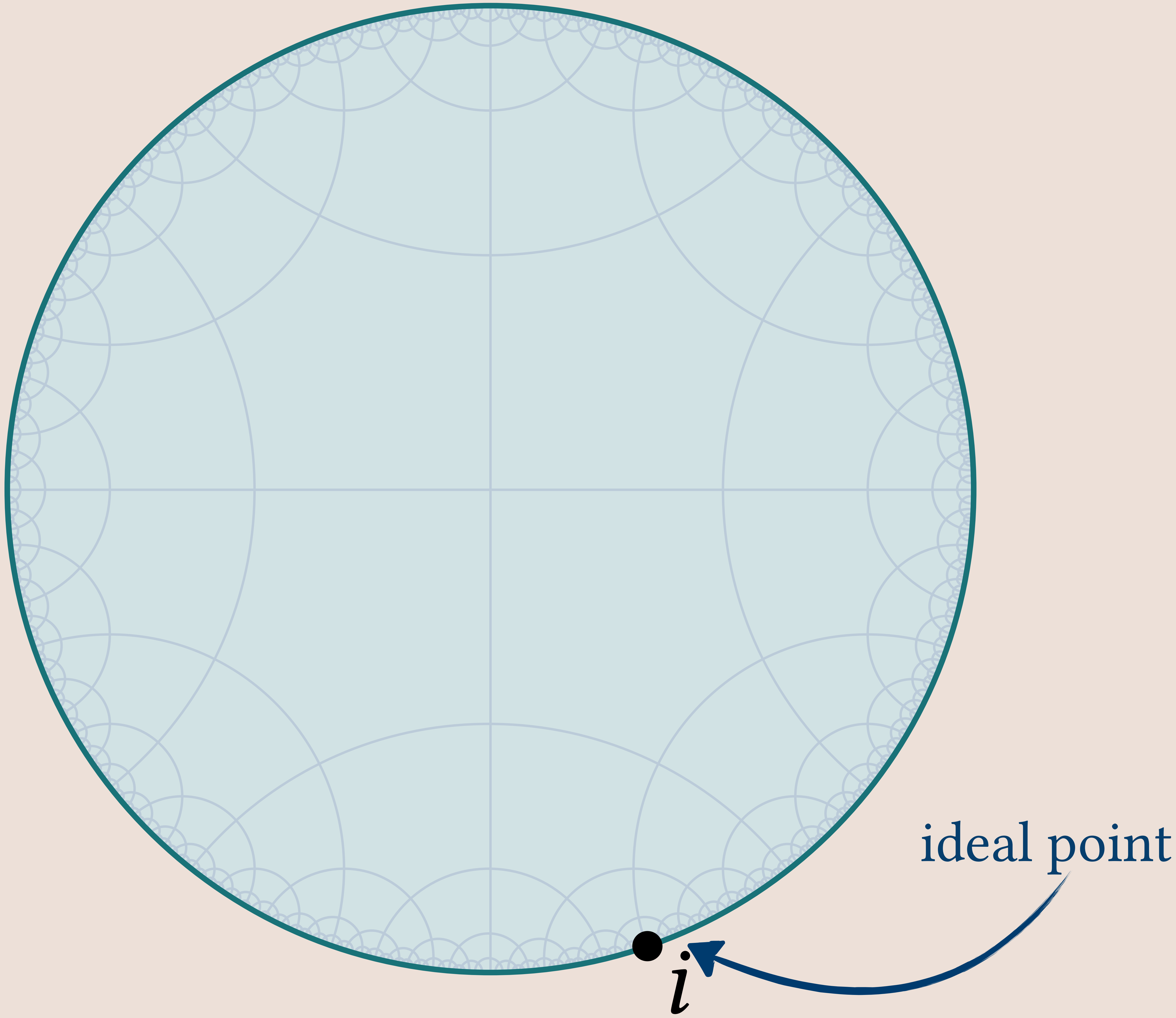
The Poincaré disk

- Represent hyperbolic plane inside unit disk
  - Conformal model:
    - Angles are preserved
    - Regions are scaled up or down



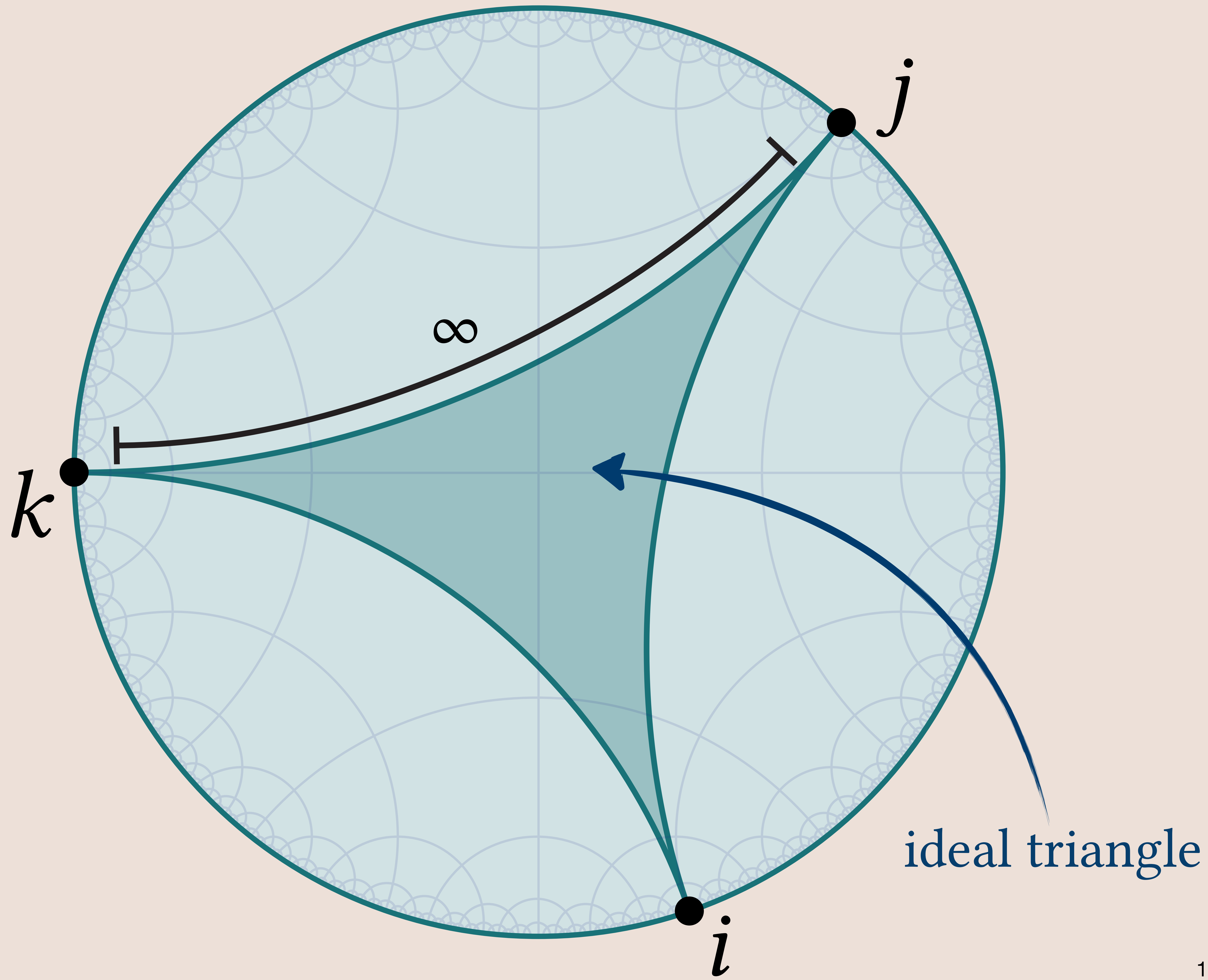
The Poincaré disk

- Represent hyperbolic plane inside unit disk
  - Conformal model:
    - Angles are preserved
    - Regions are scaled up or down



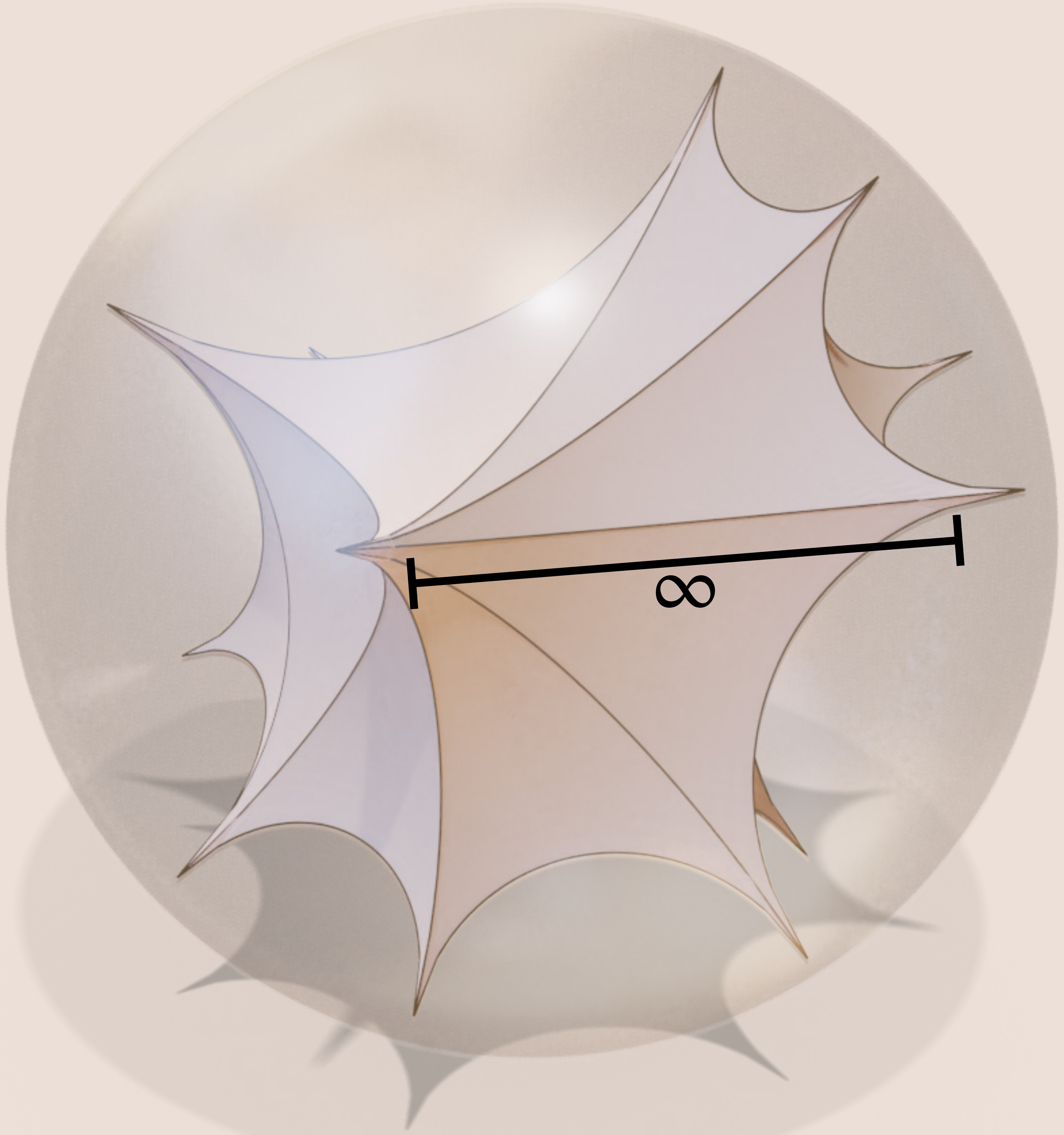
The Poincaré disk

- Represent hyperbolic plane inside unit disk
  - Conformal model:
    - Angles are preserved
    - Regions are scaled up or down

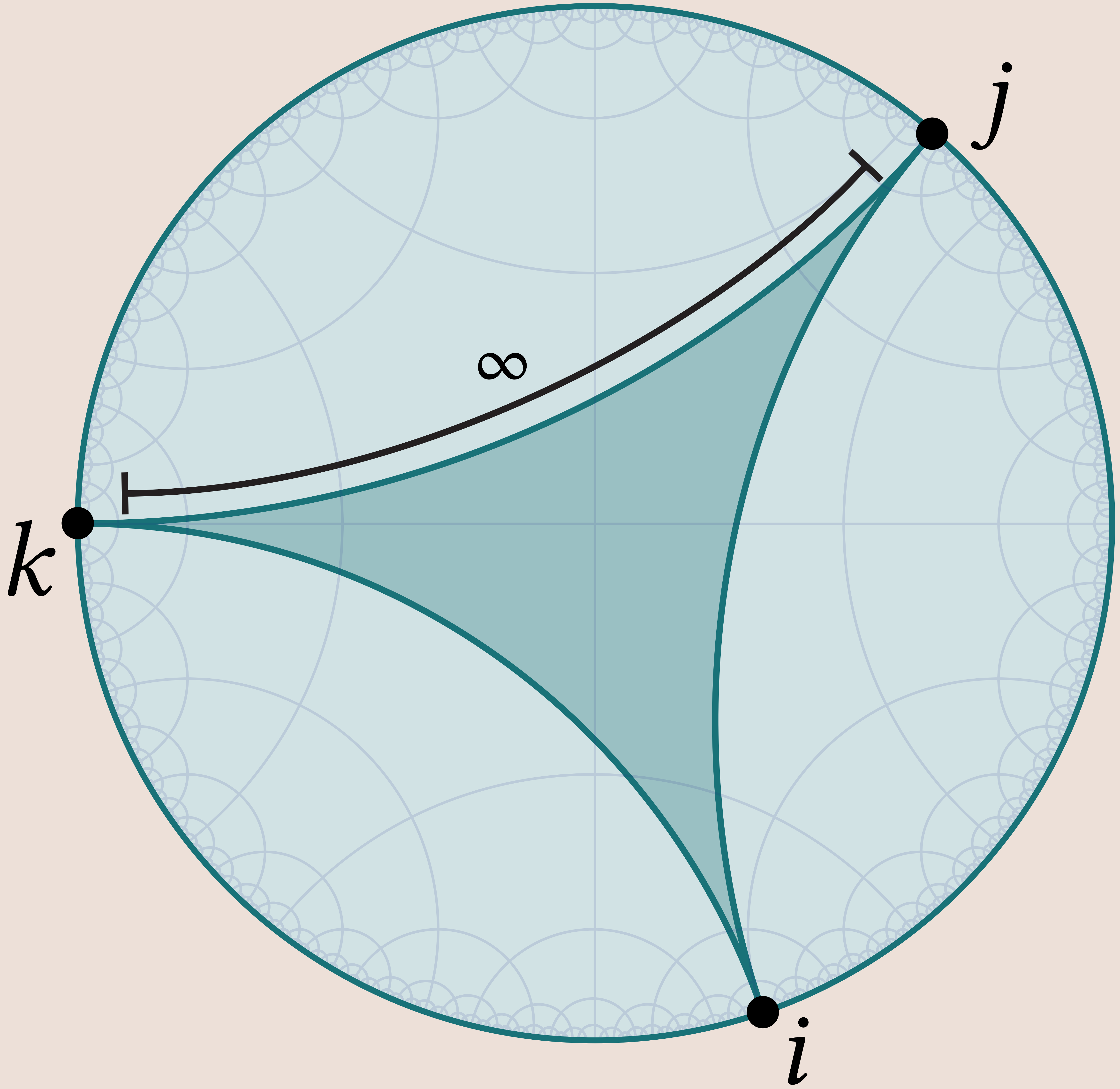


Ideal hyperbolic polyhedra

- Glue together several ideal triangles

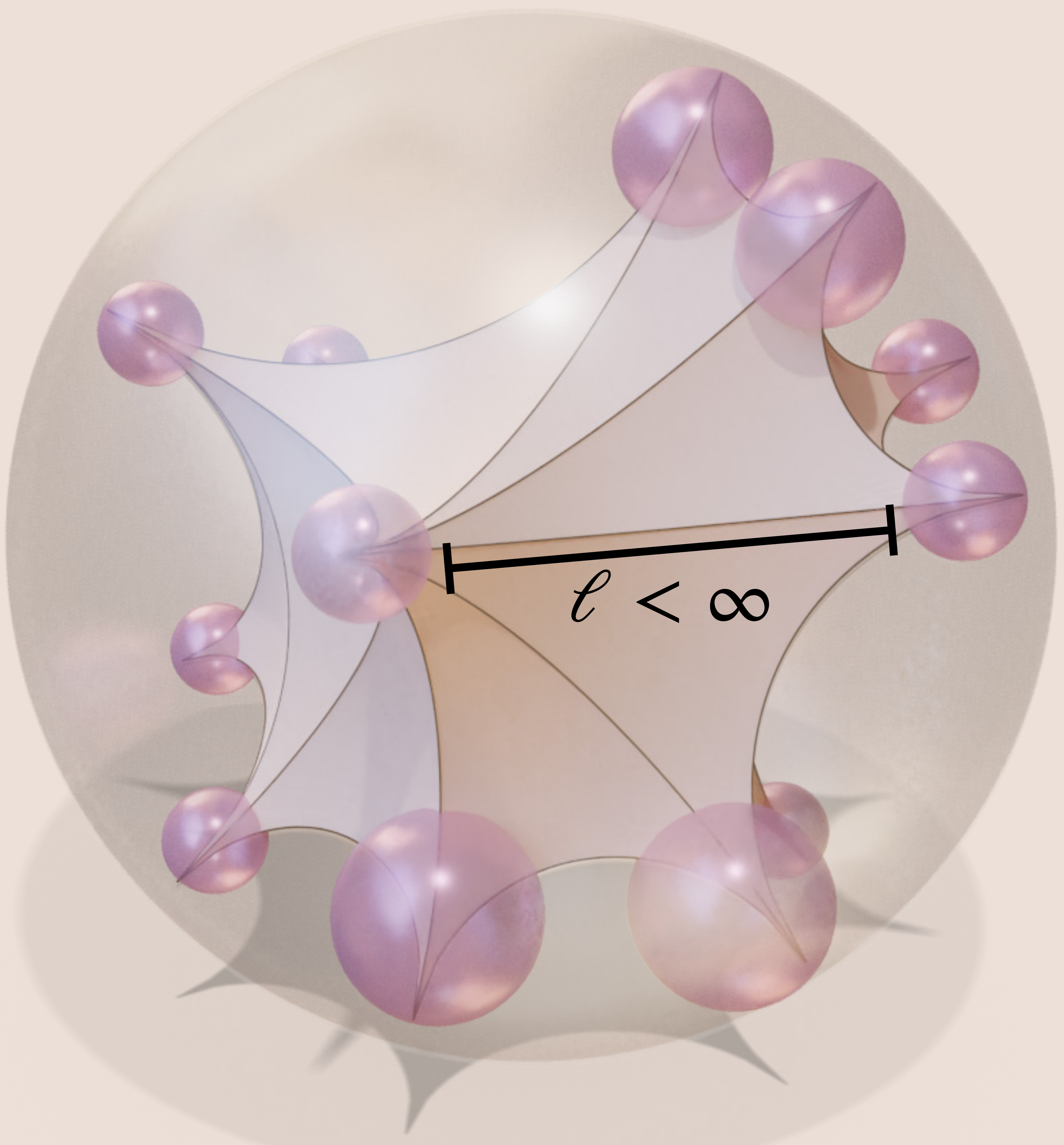


ideal hyperbolic polyhedron

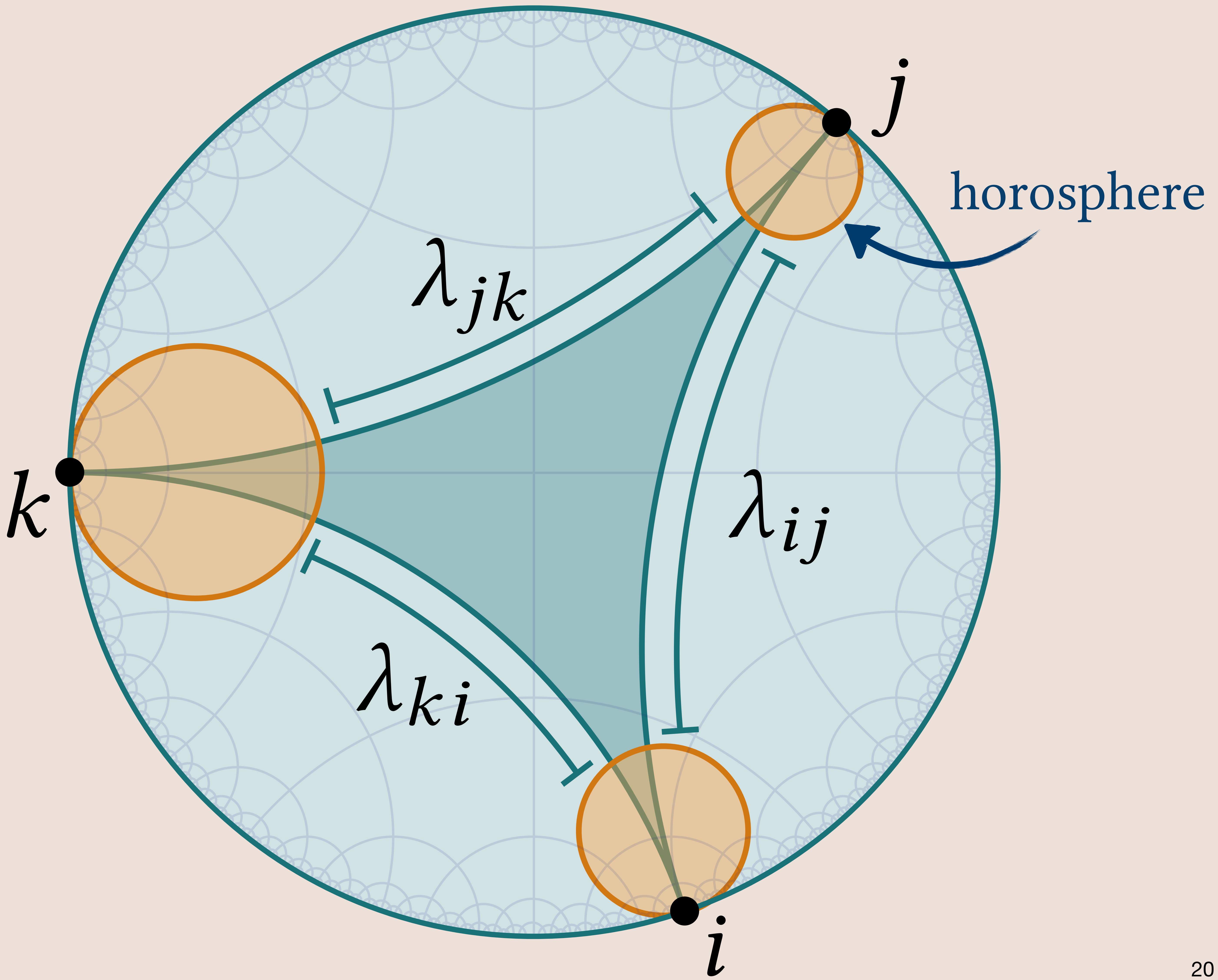


Ideal hyperbolic polyhedra

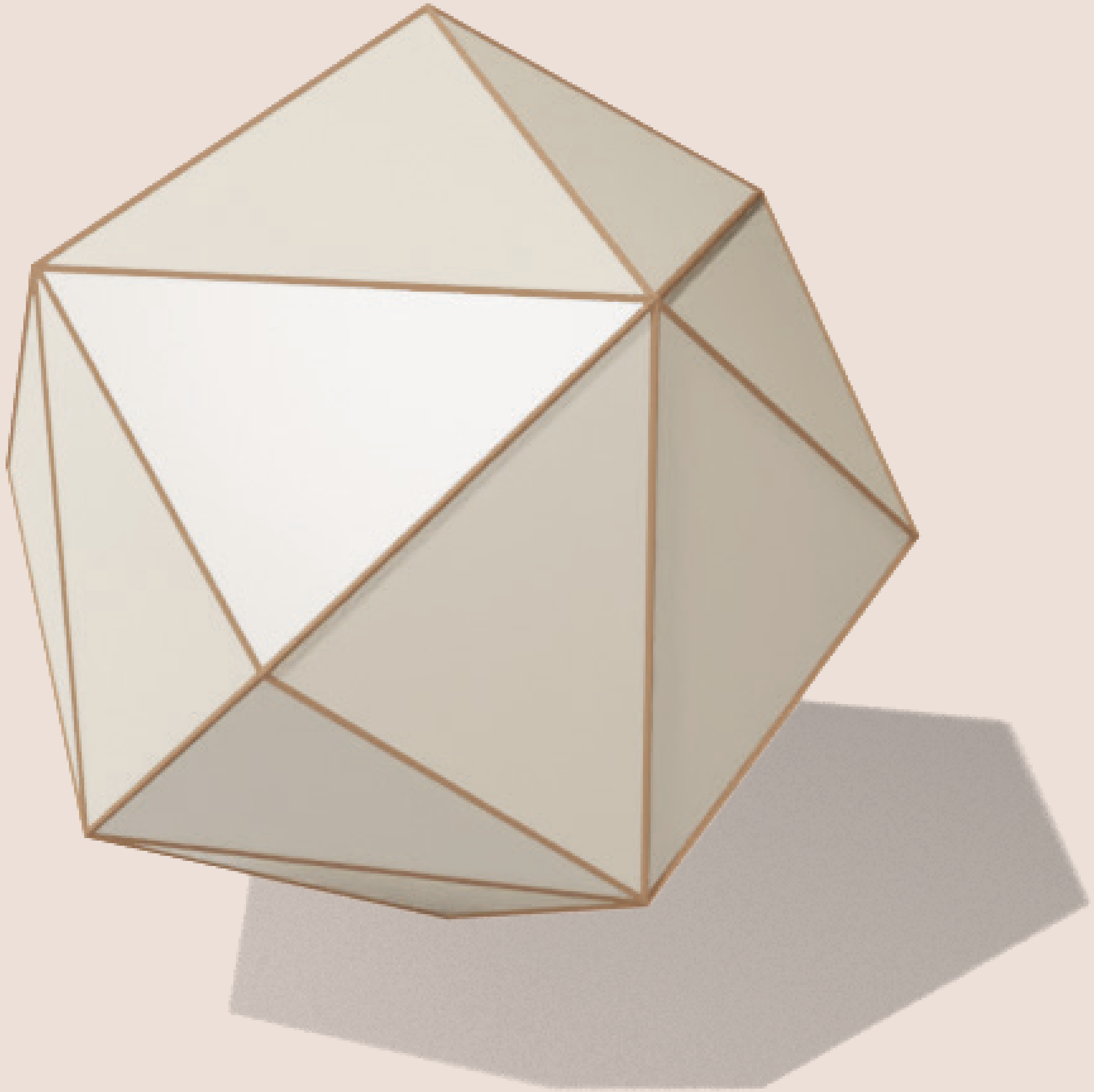
- Glue together several ideal triangles

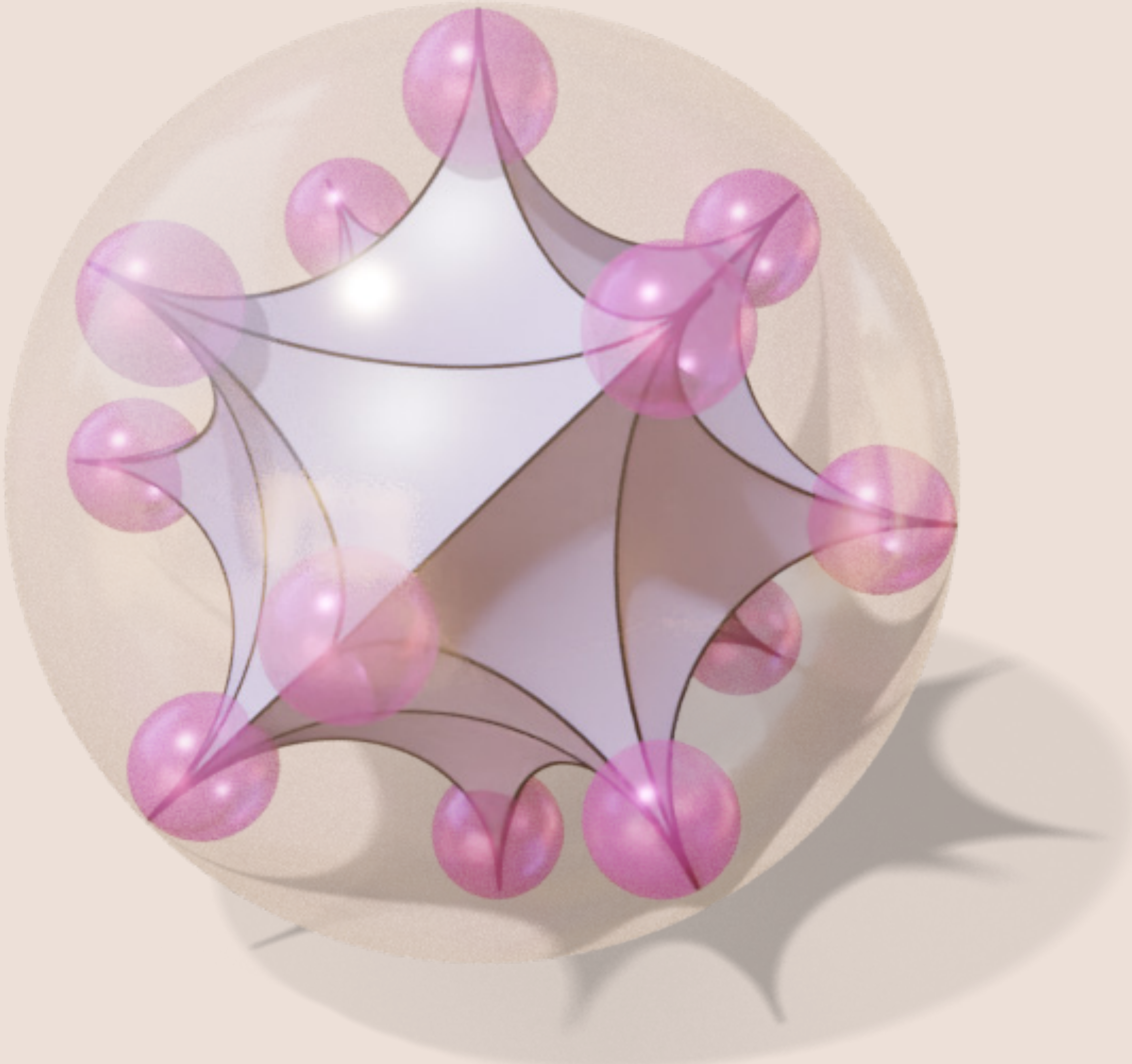


ideal hyperbolic polyhedron

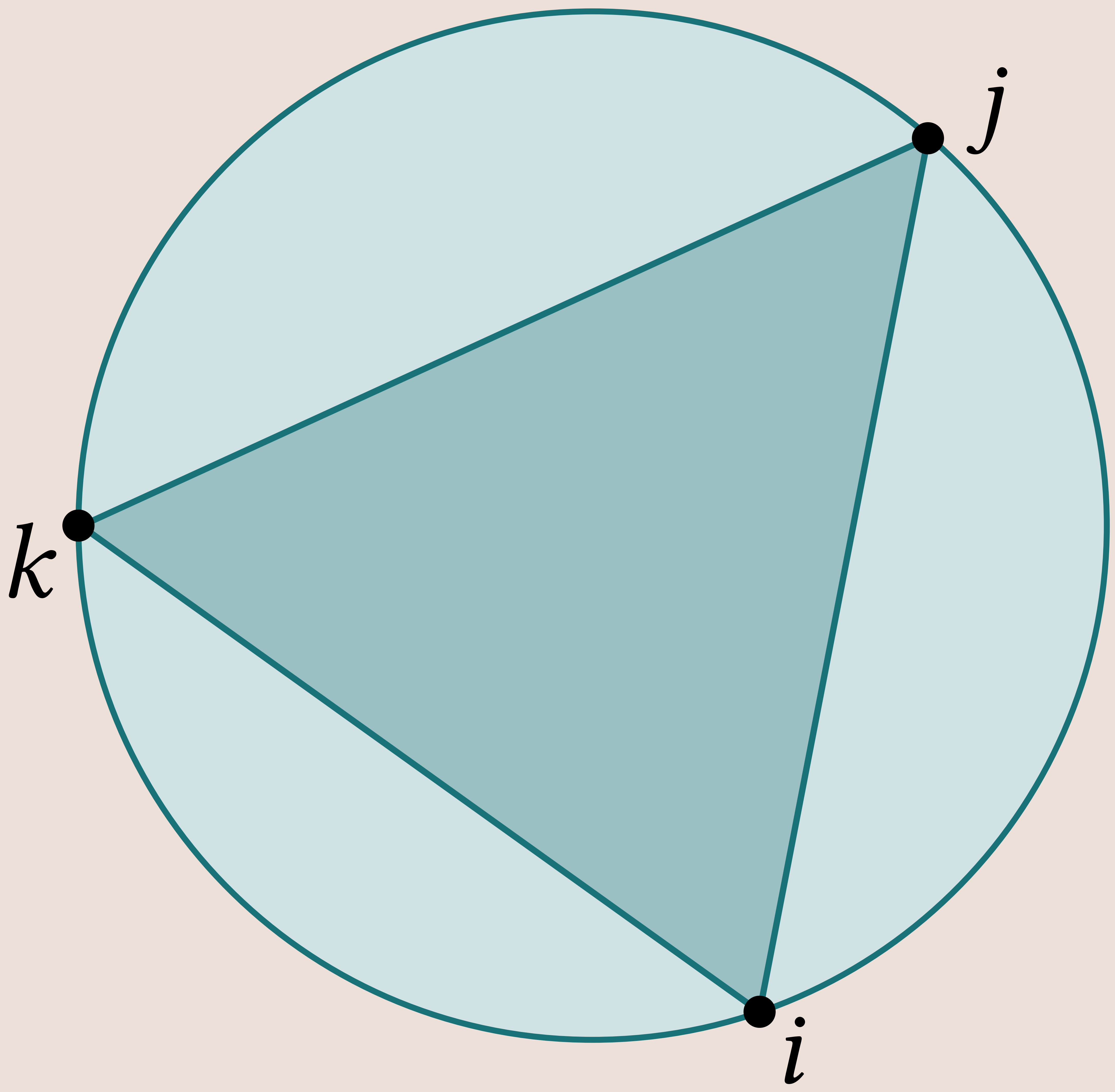


Triangle mesh ⟷ ideal polyhedron



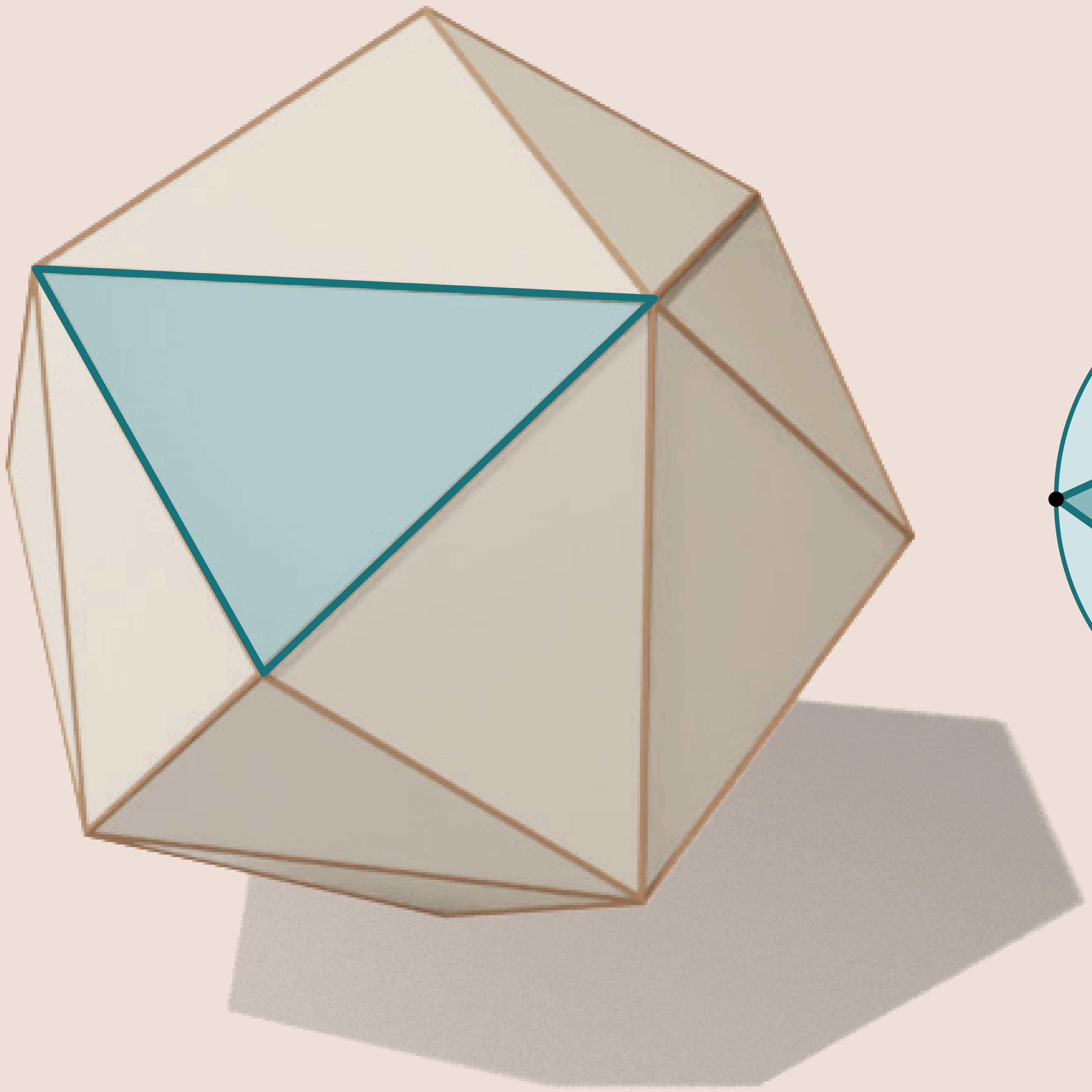


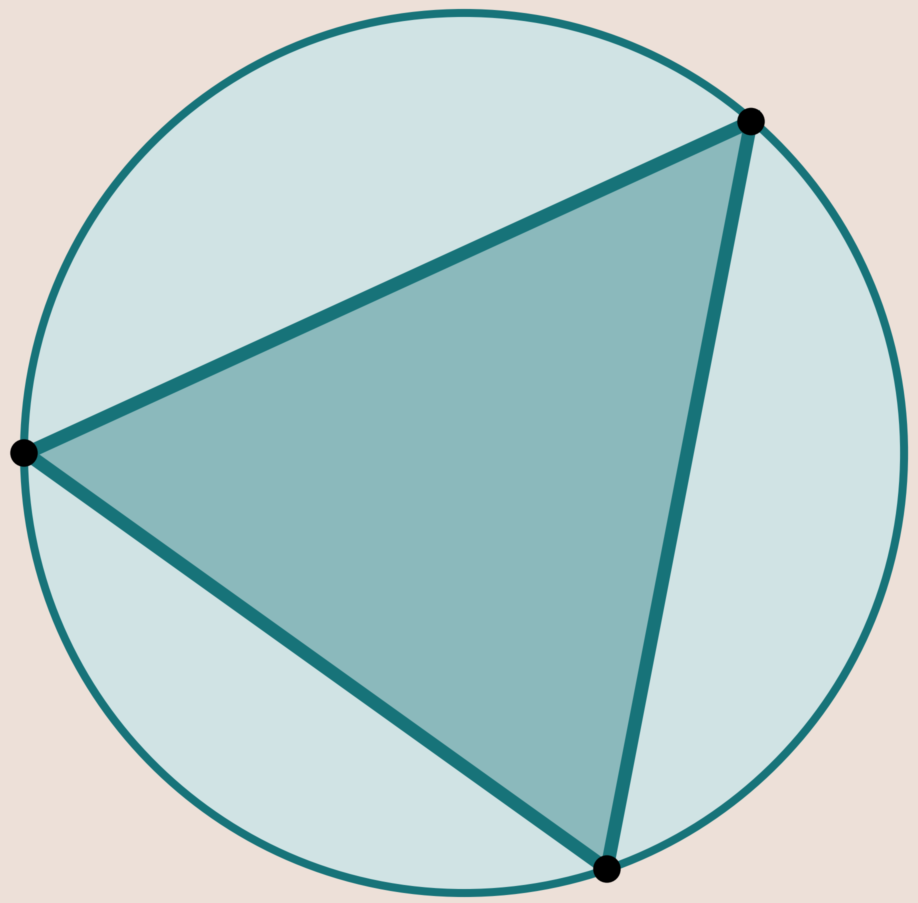
The Beltrami-Klein model



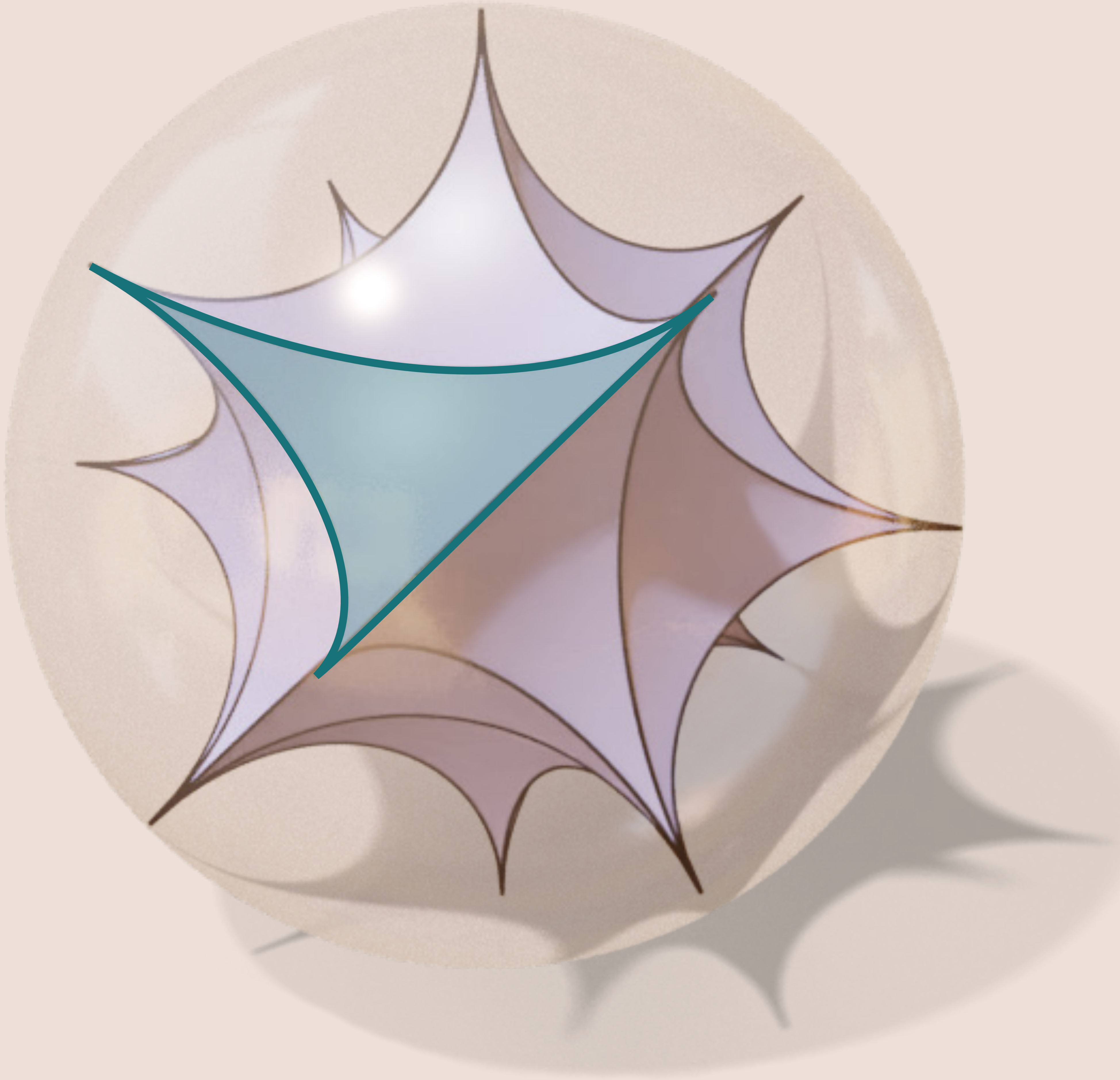
Klein ideal triangle $\longleftrightarrow$ Euclidean triangle inside circle

Triangle mesh $\longleftrightarrow$ ideal polyhedron

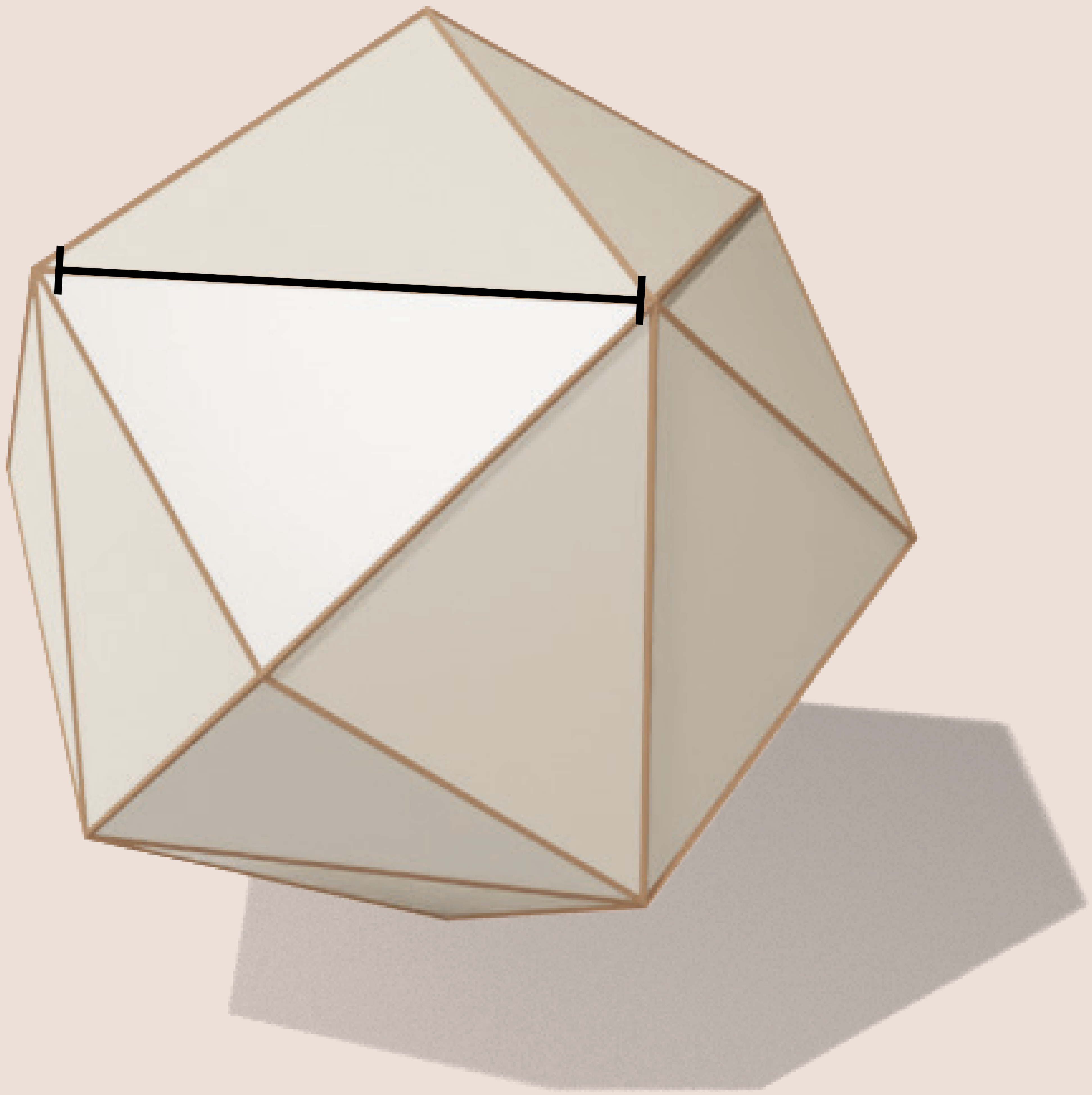


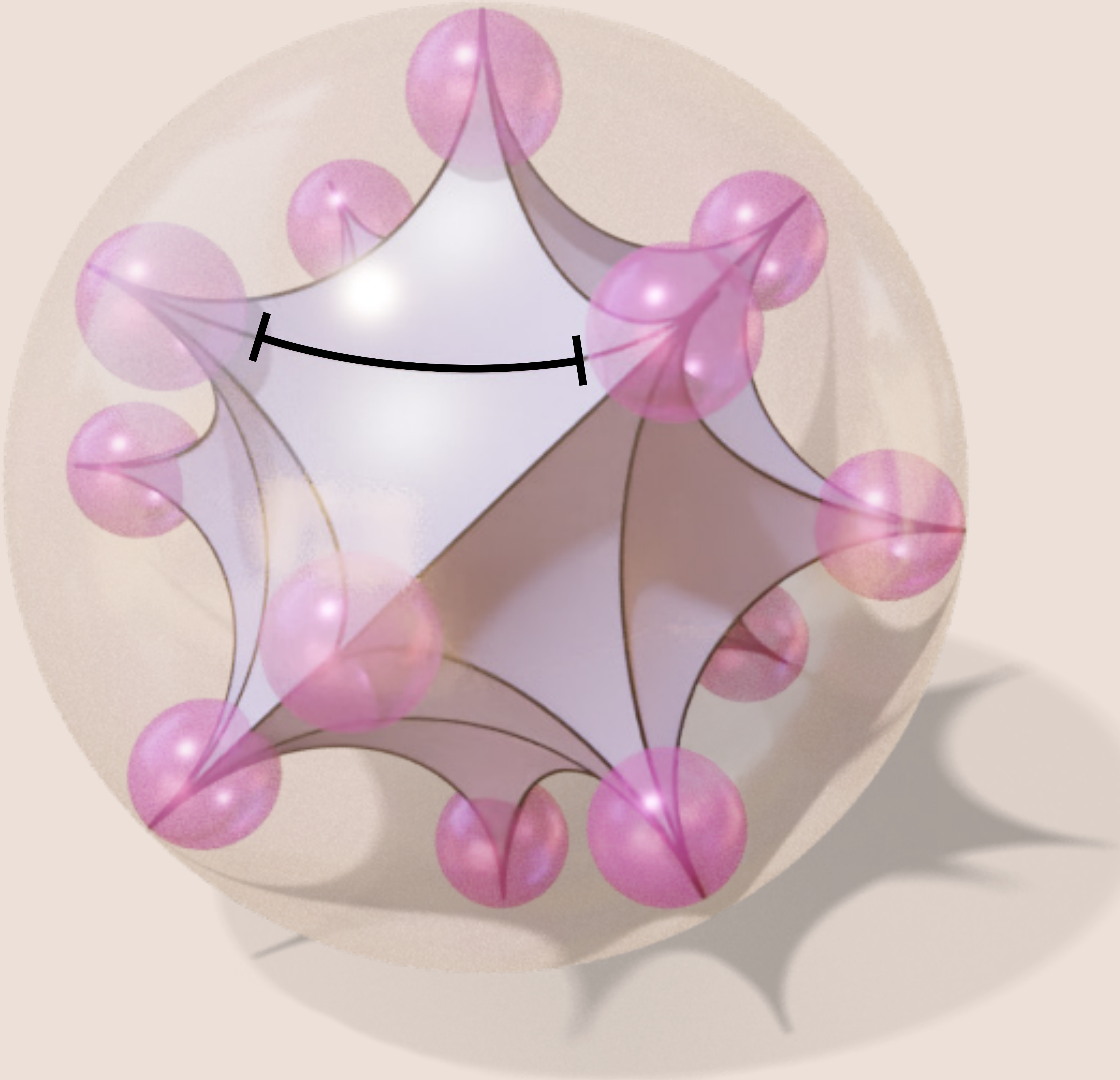


ideal triangle



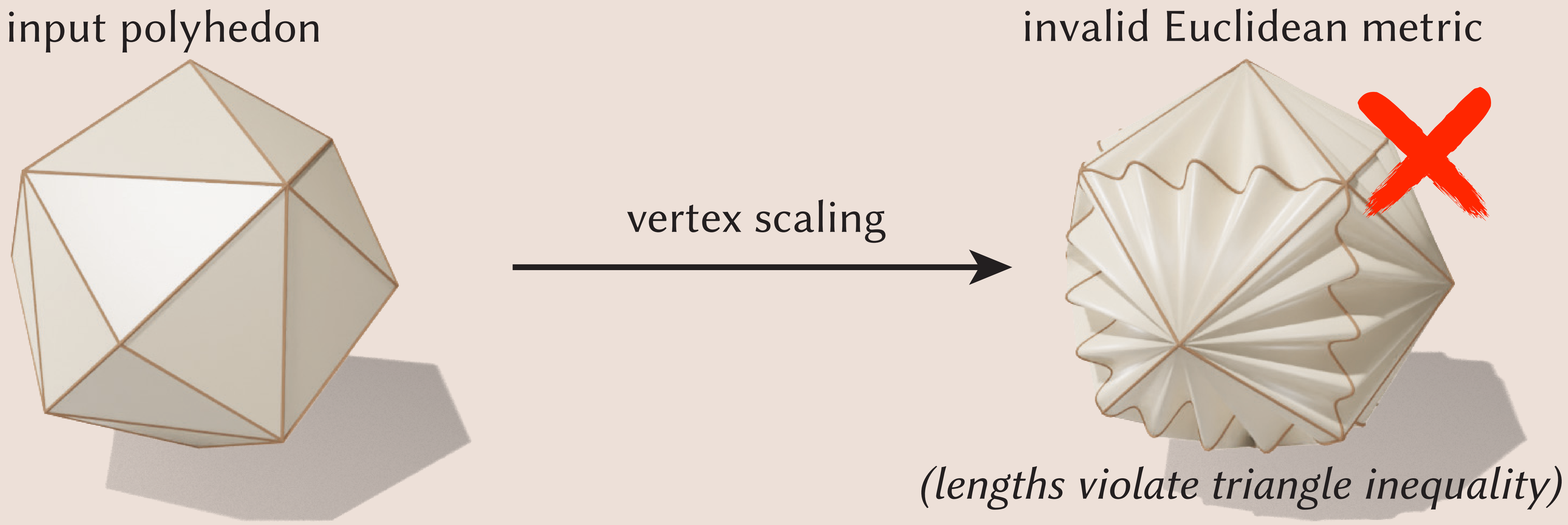
Triangle mesh $\longleftrightarrow$ ideal polyhedron



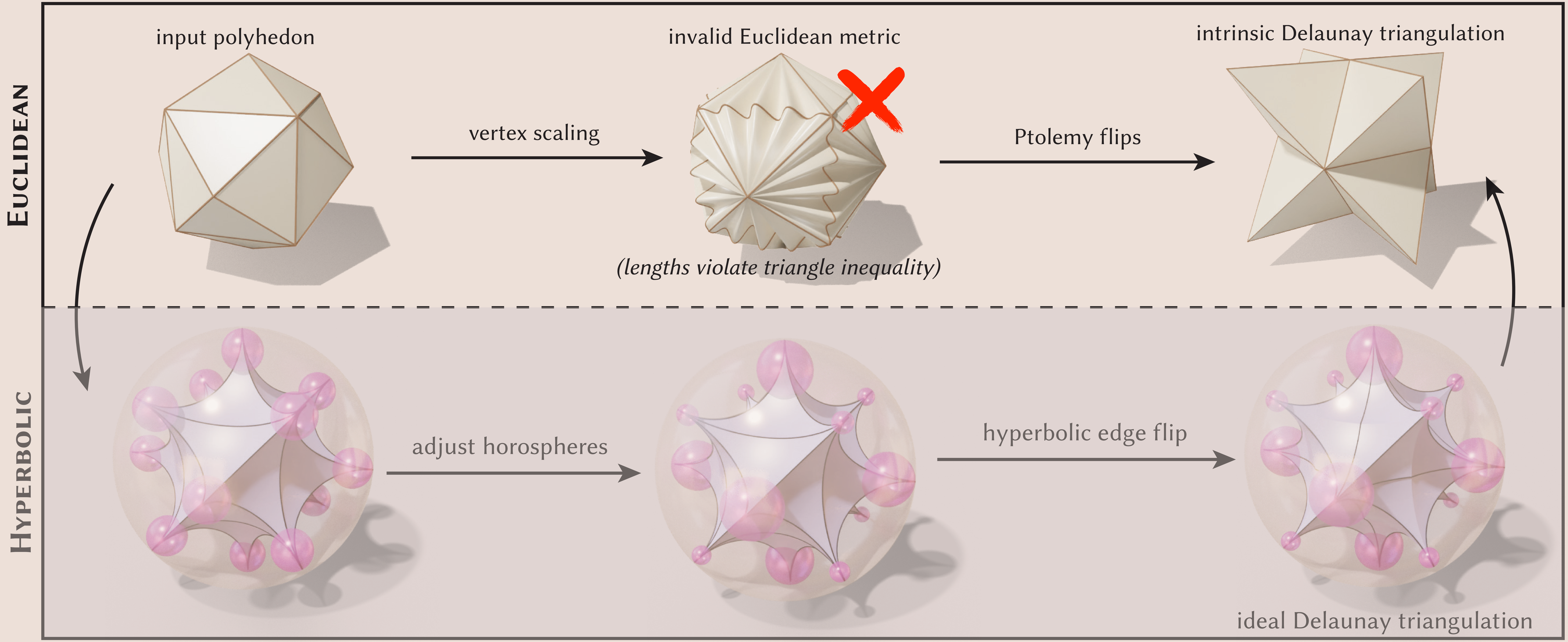


Discrete conformal maps across triangulations

Euclidean



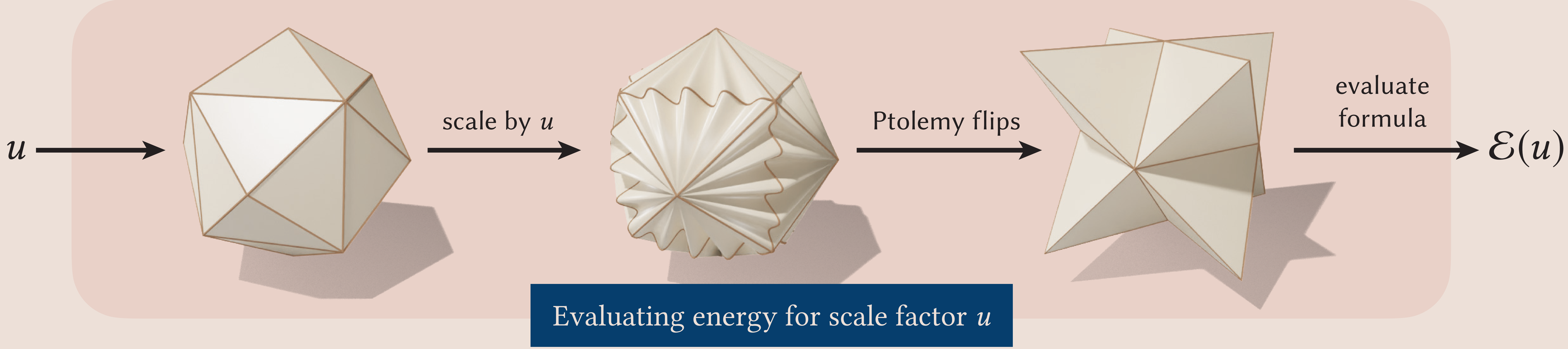
Discrete conformal maps across triangulations



Gives same result as pausing during scaling process to maintaining Delaunay condition

Optimization with Ptolemy flips

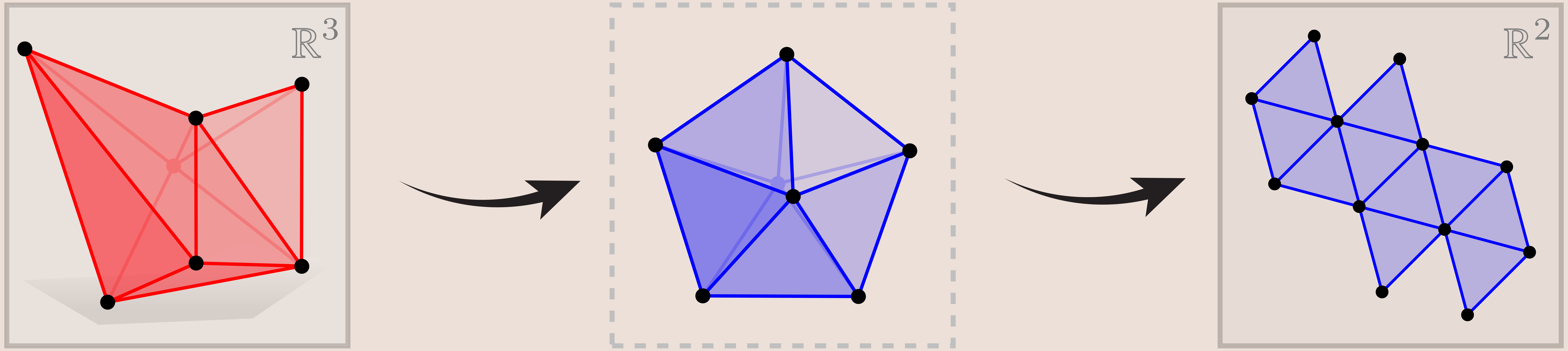
- Finding discrete conformal parameterization $\longleftrightarrow$ minimizing energy $\mathcal{E}(u)$
- Have expressions for energy and derivatives in terms of edge lengths [Springborn+ 2008]



- Hand to any optimization algorithm

Energy remains convex and C^2

The procedure so far



Find optimal scale factors

Lay out triangles in plane

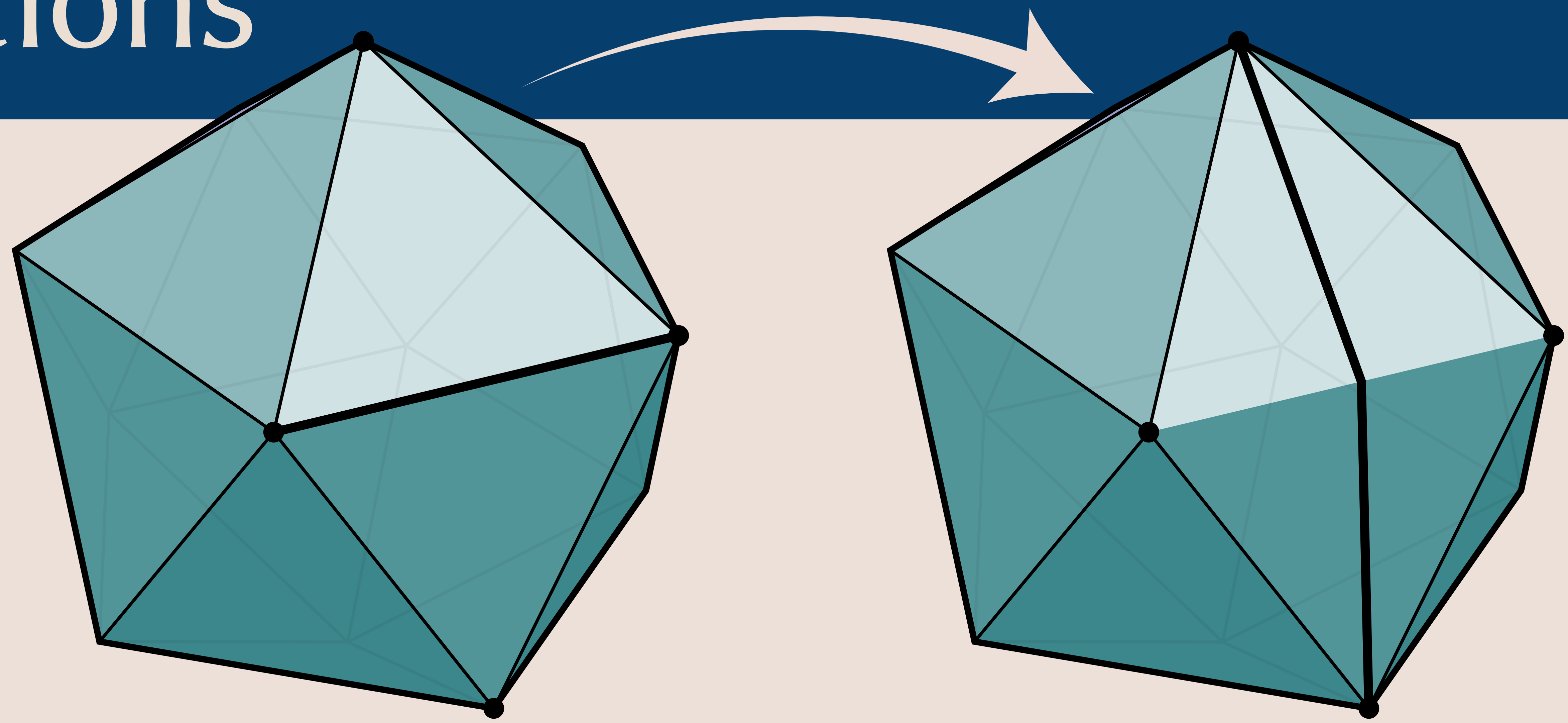
Challenge: connectivity might have changed

Correspondence Tracking

with normal coordinates and roundabouts

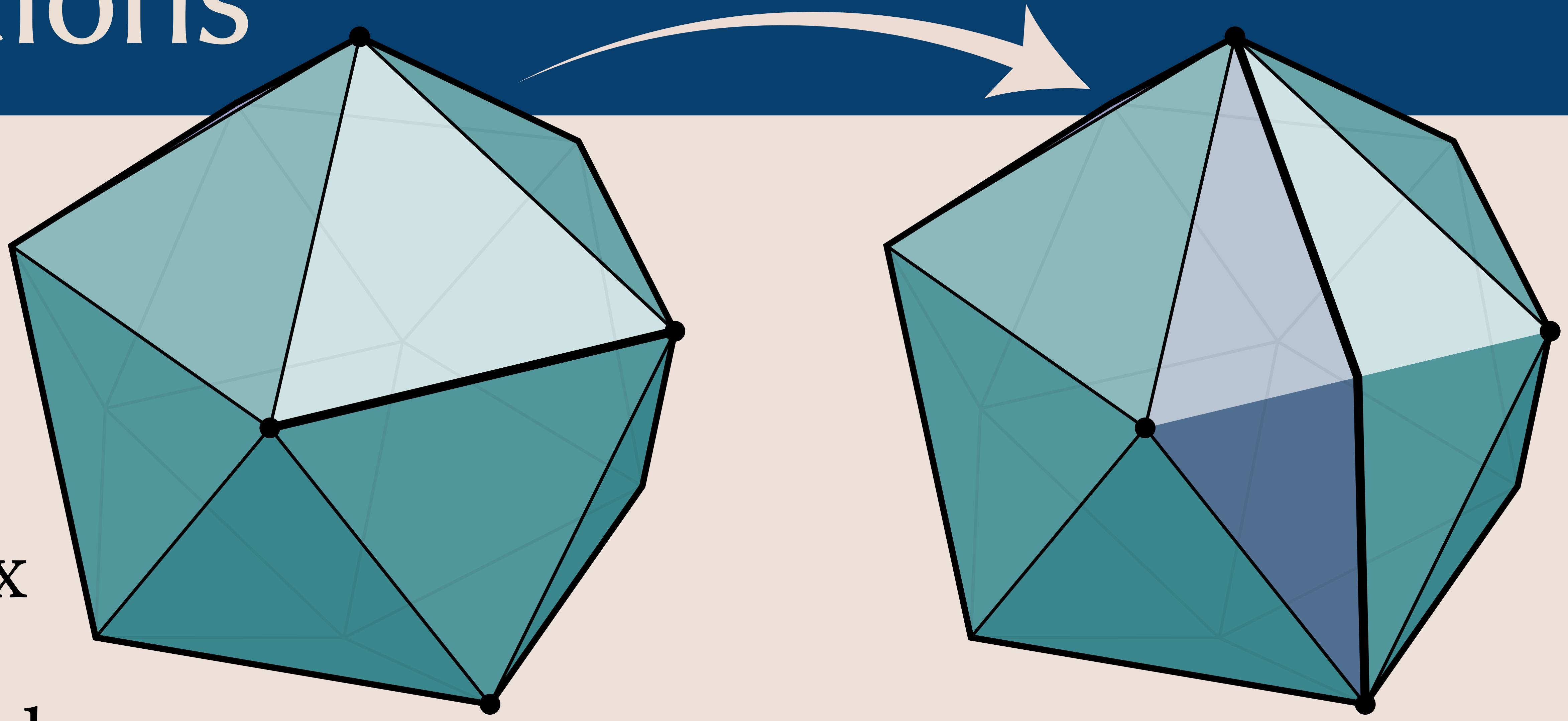
Intrinsic triangulations

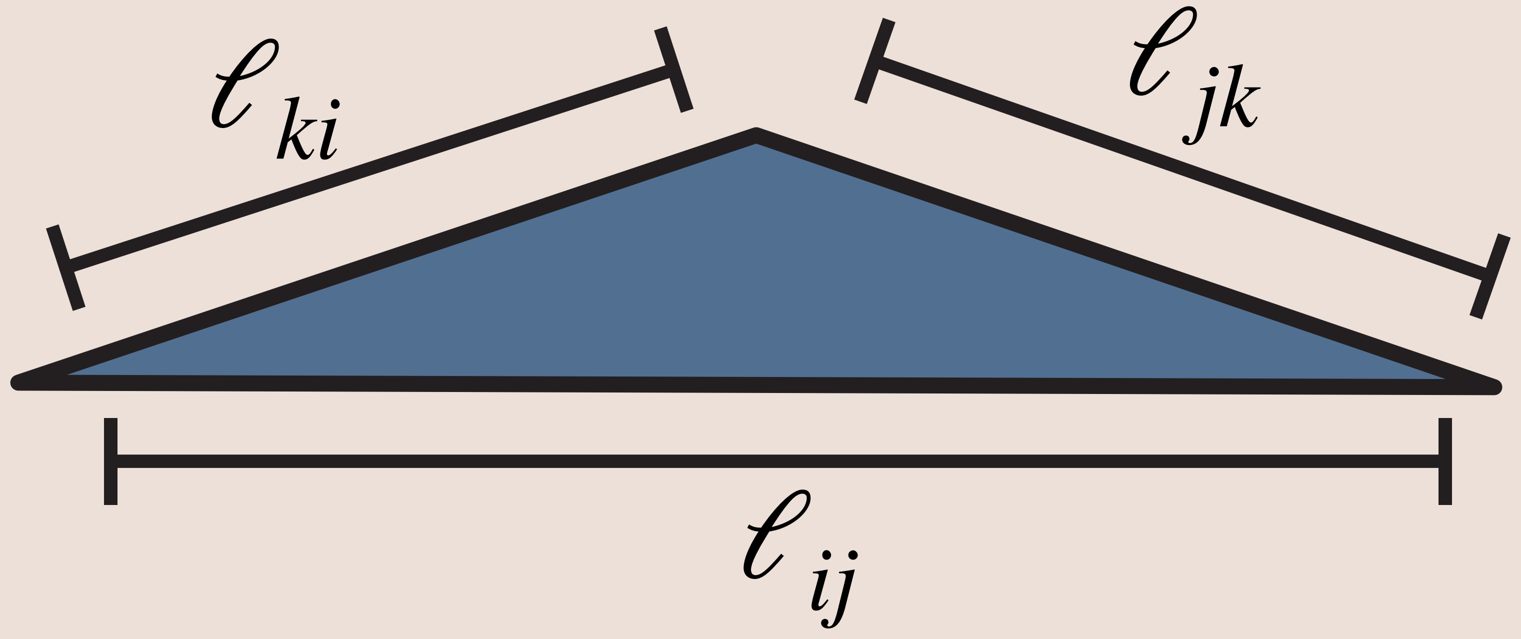
- Intrinsic edge flips

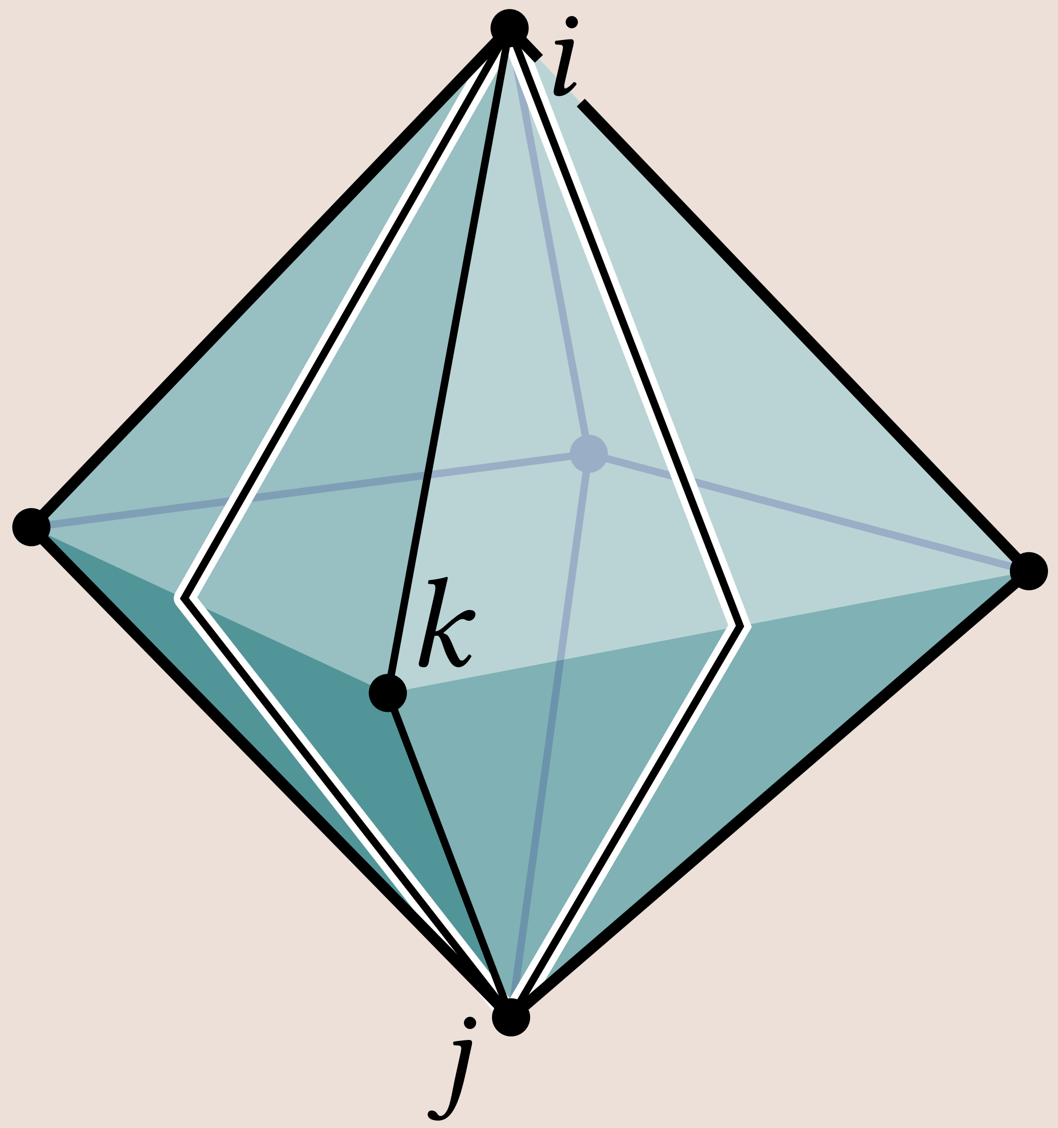


Intrinsic triangulations

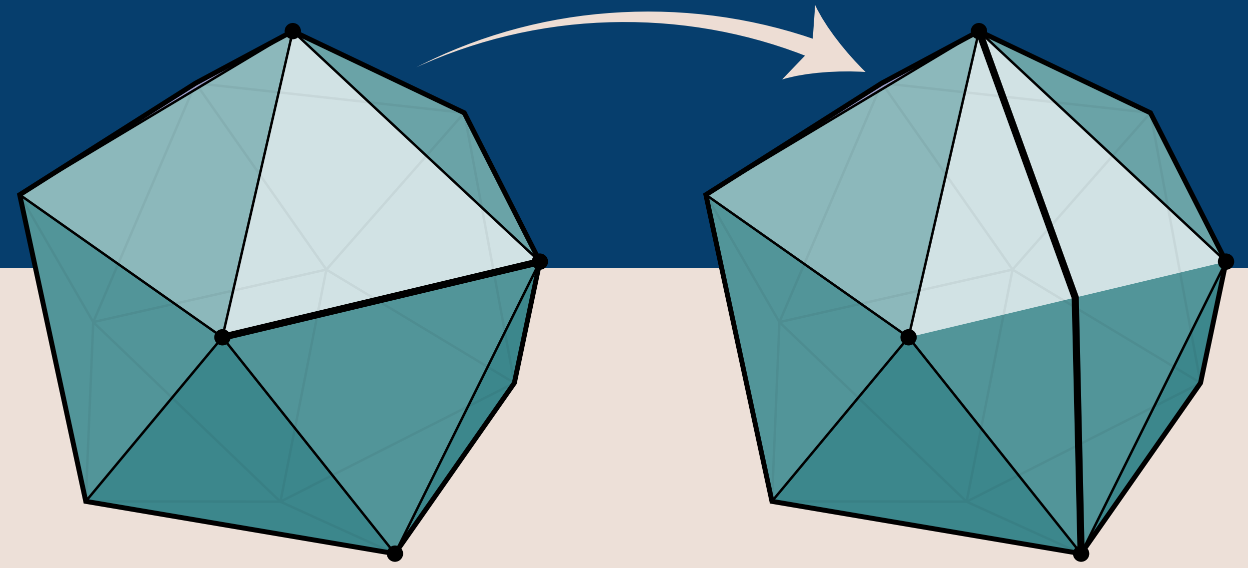
- Intrinsic edge flips
- Basic data: edge lengths
- Mesh is a general Δ-complex
  - Allows self edges, multi edges





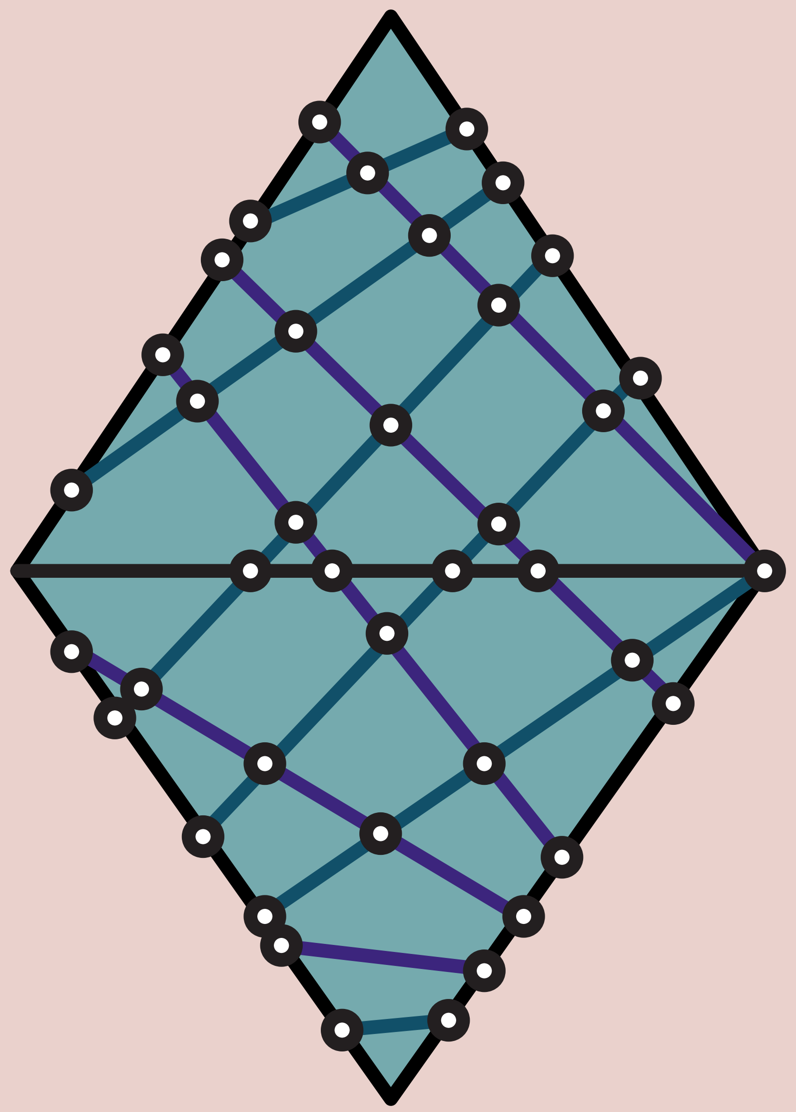


Correspondence data structures



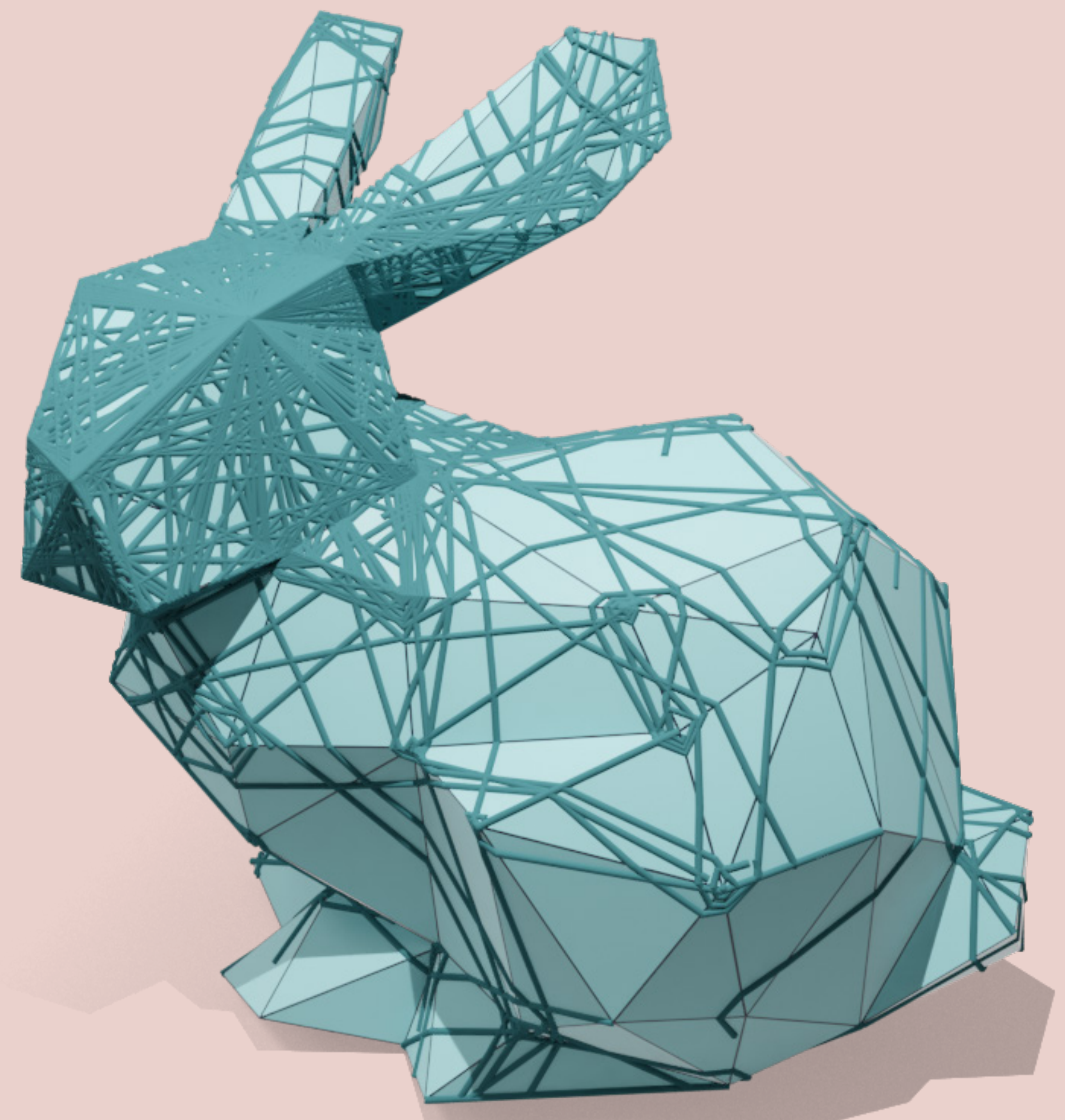
- How does triangulation sit over input?
  - Existing schemes don't suffice in this hyperbolic setting

[Fisher+ 2007]



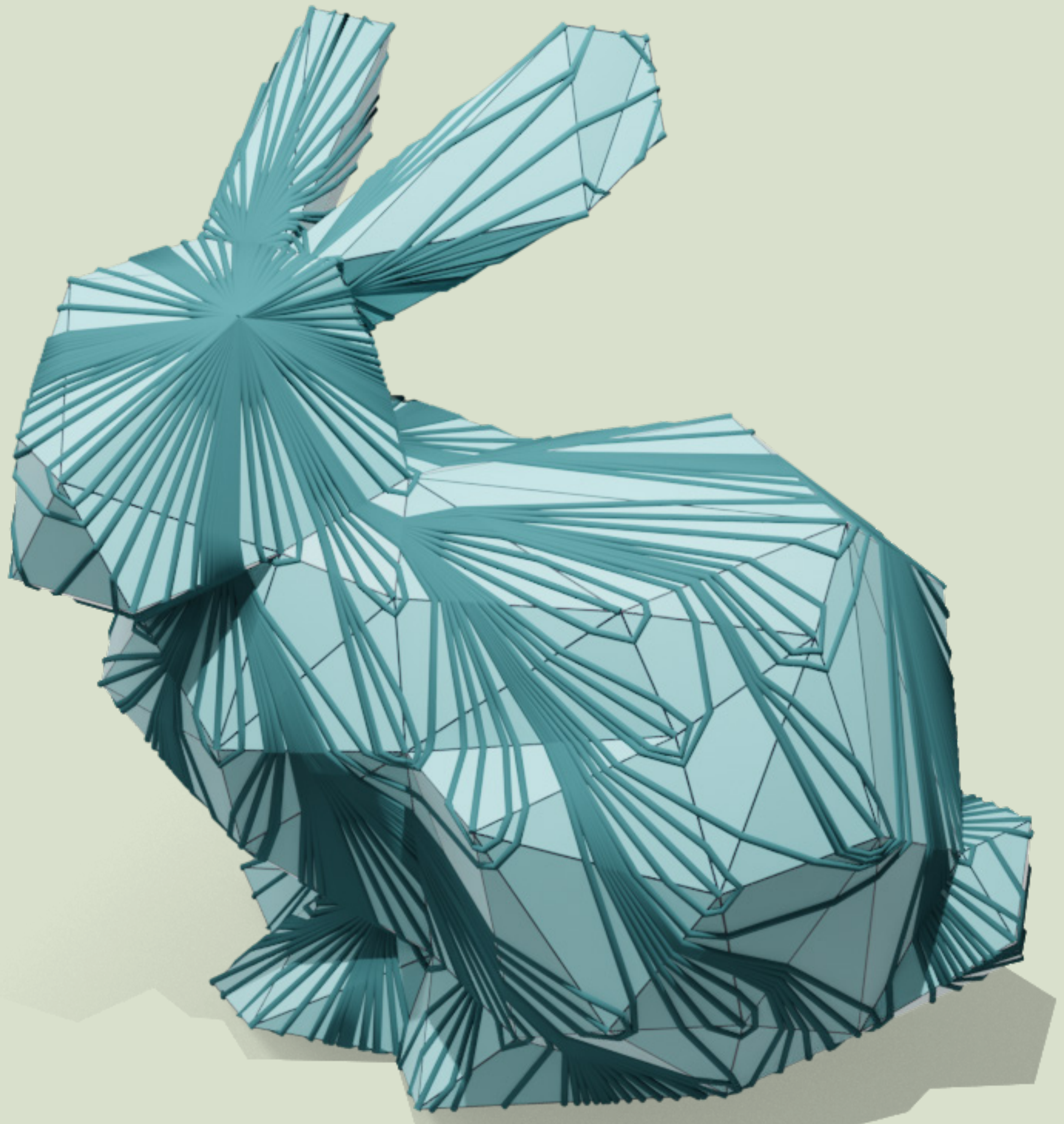
✗ prohibitively complex

[Sharp+ 2019]



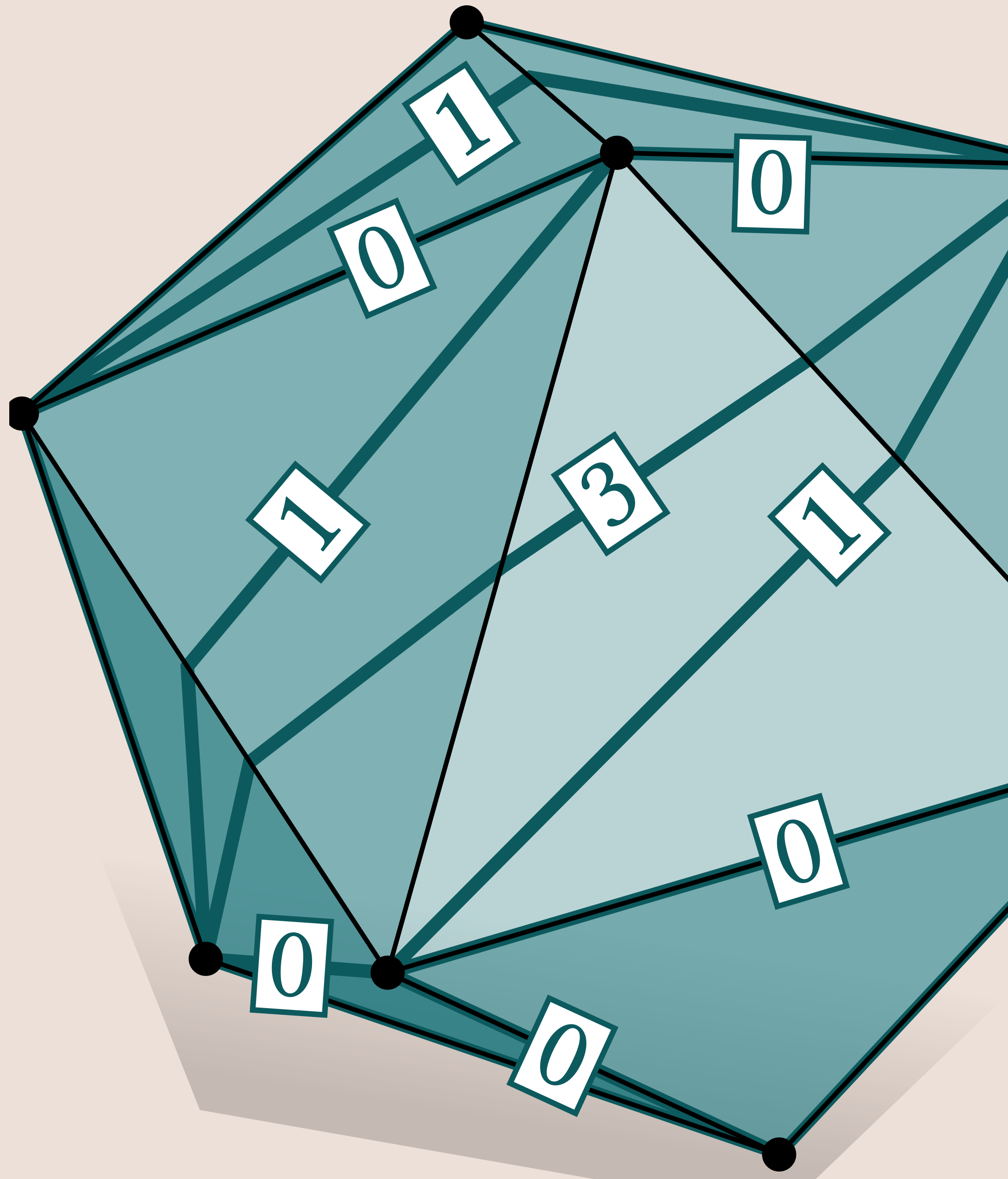
✗ floating point error

new data structure



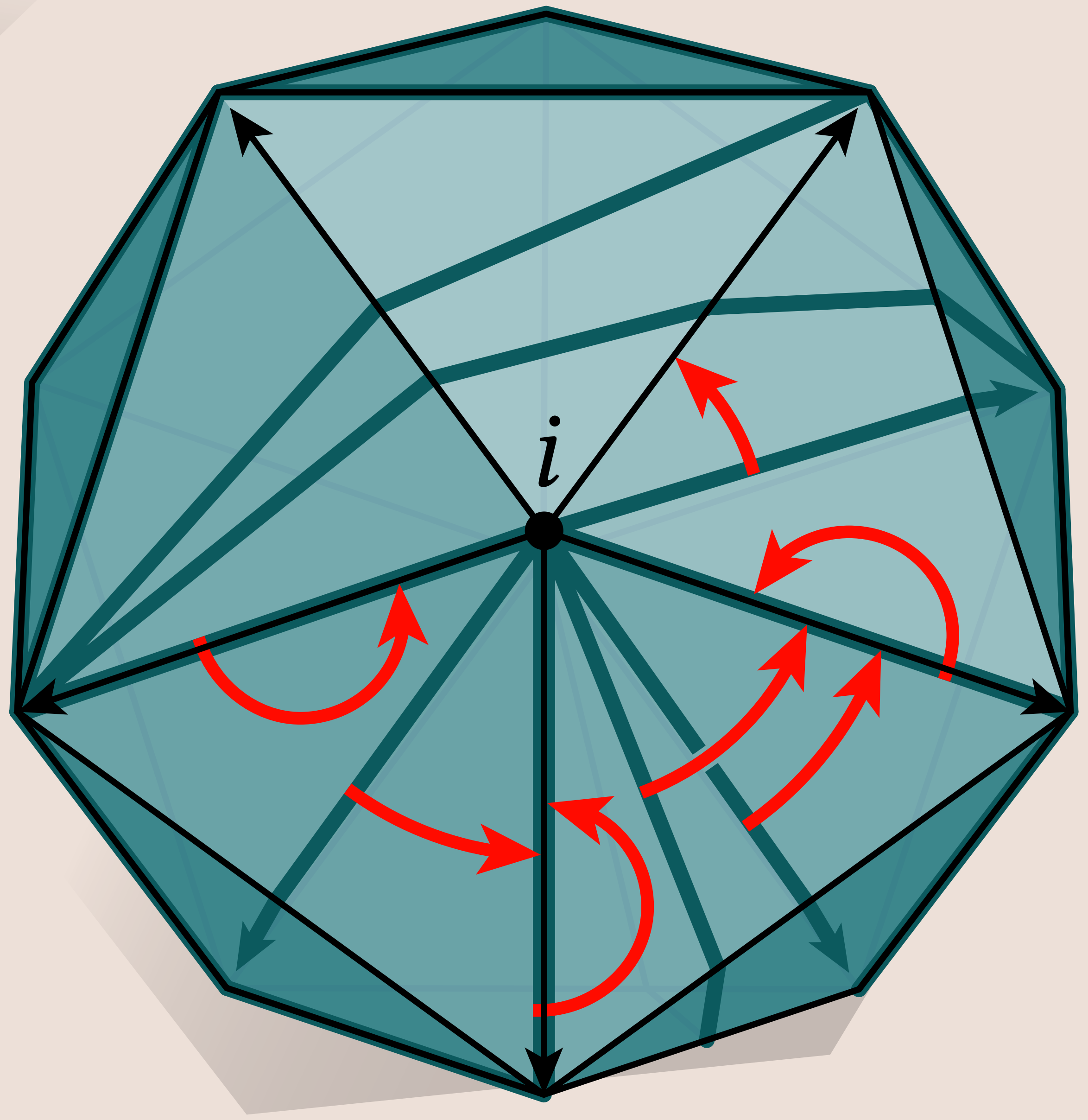
Normal coordinates & roundabouts

- Integer encoding of correspondence
- Fully determines geometry of intersections
- Easy to update



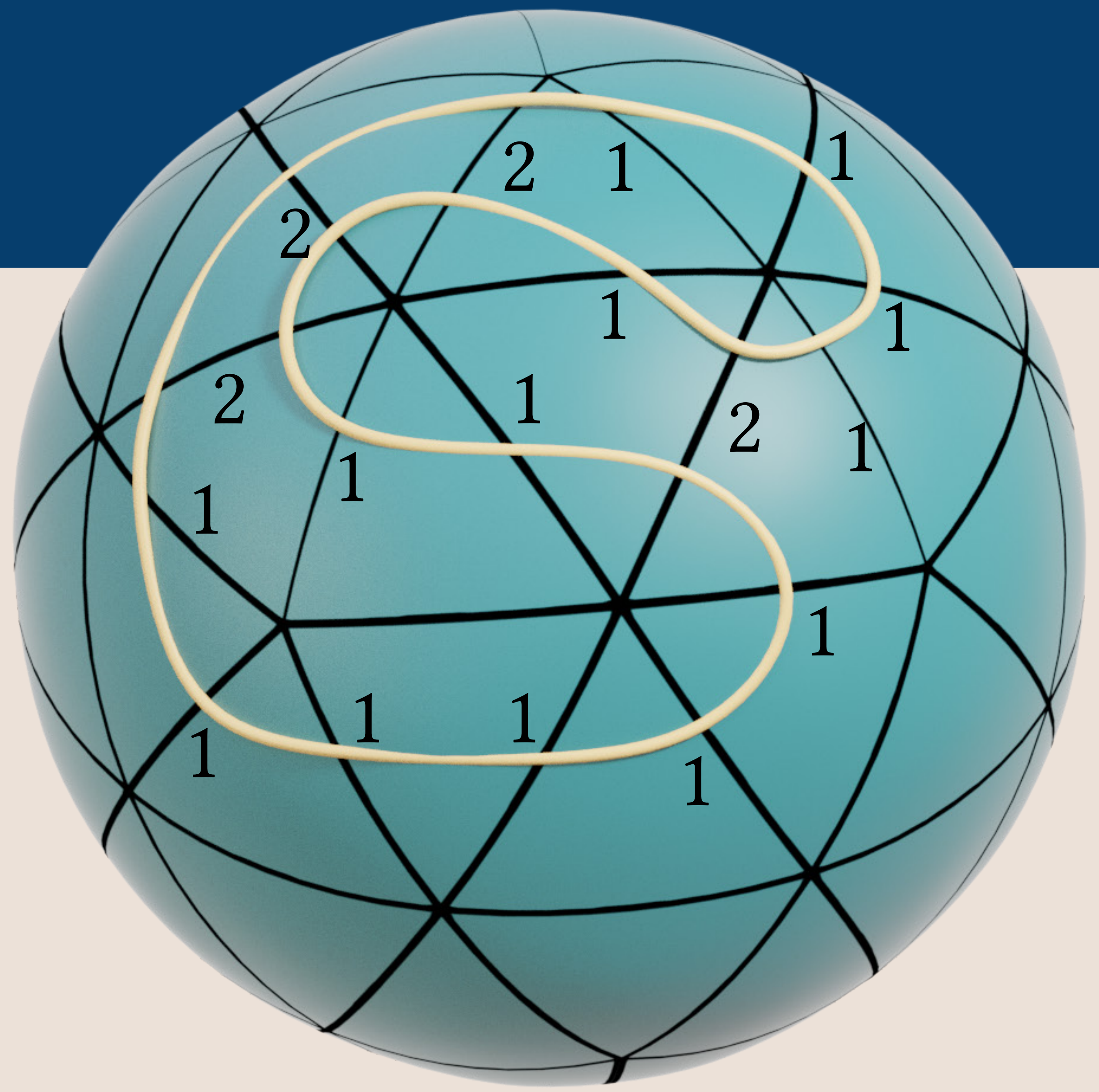
normal coordinates*

roundabouts

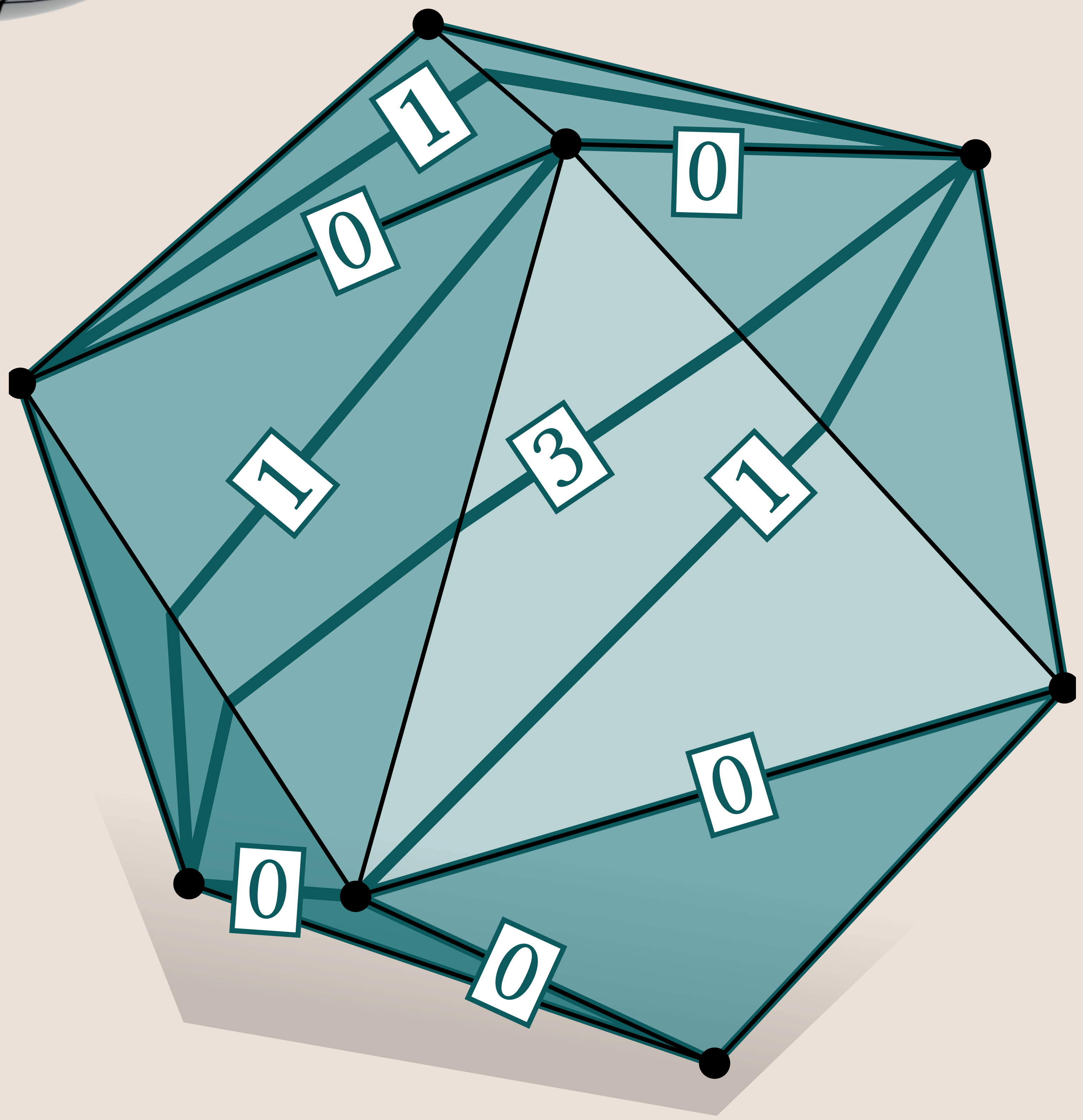


* not the same thing as "geodesic normal coordinates" from Riemannian geometry

Normal coordinates

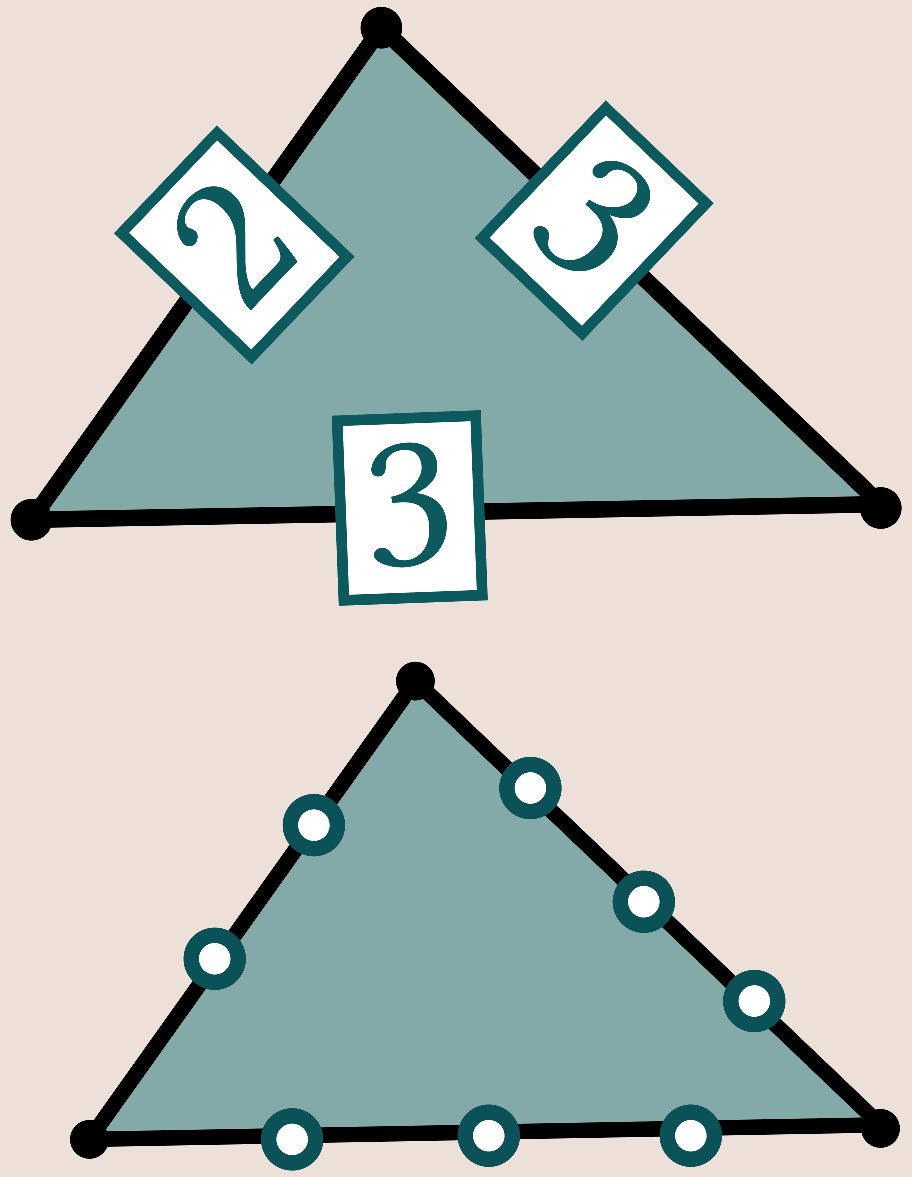


- Tool from geometric topology
  [Kneser 1929; Haken 1961]
- Encode curve sitting along a triangulation
  - Just count crossings
  - Determines curves up to homotopy
- We diverge from standard usage
  - Geodesic *triangulations* on triangle meshes
  - Determines curve geometry
  - New edge flip formula



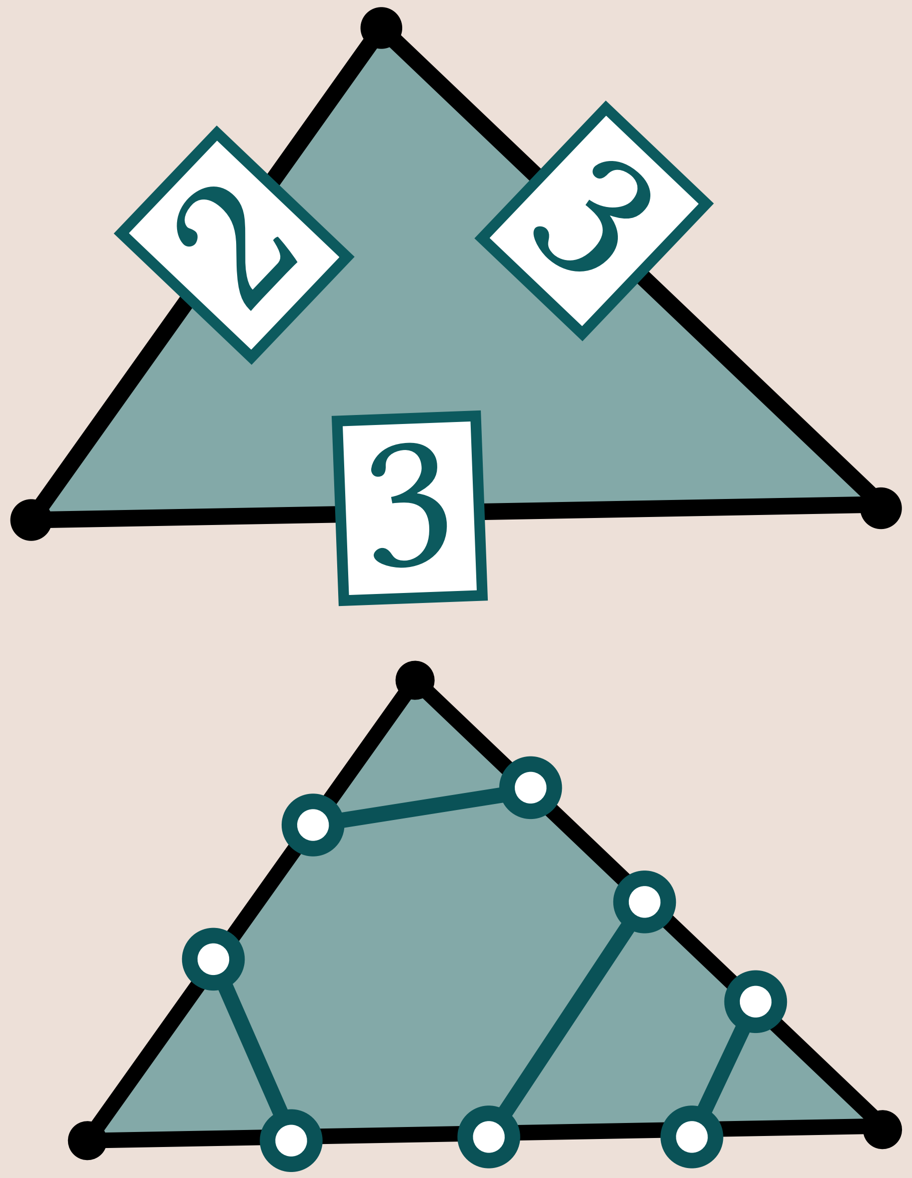
Recovering curves from normal coordinates

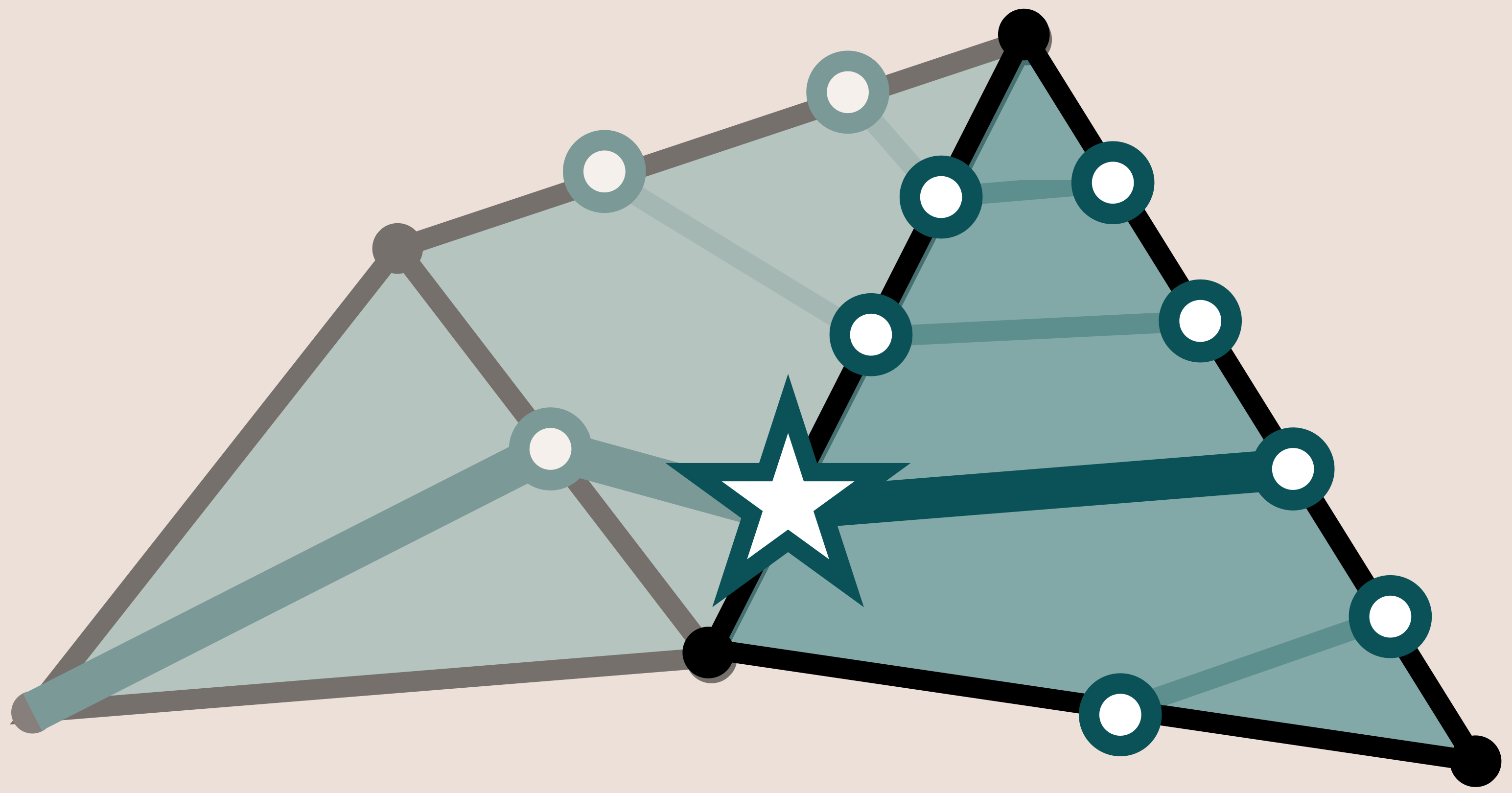
- Trace curve along mesh



Recovering curves from normal coordinates

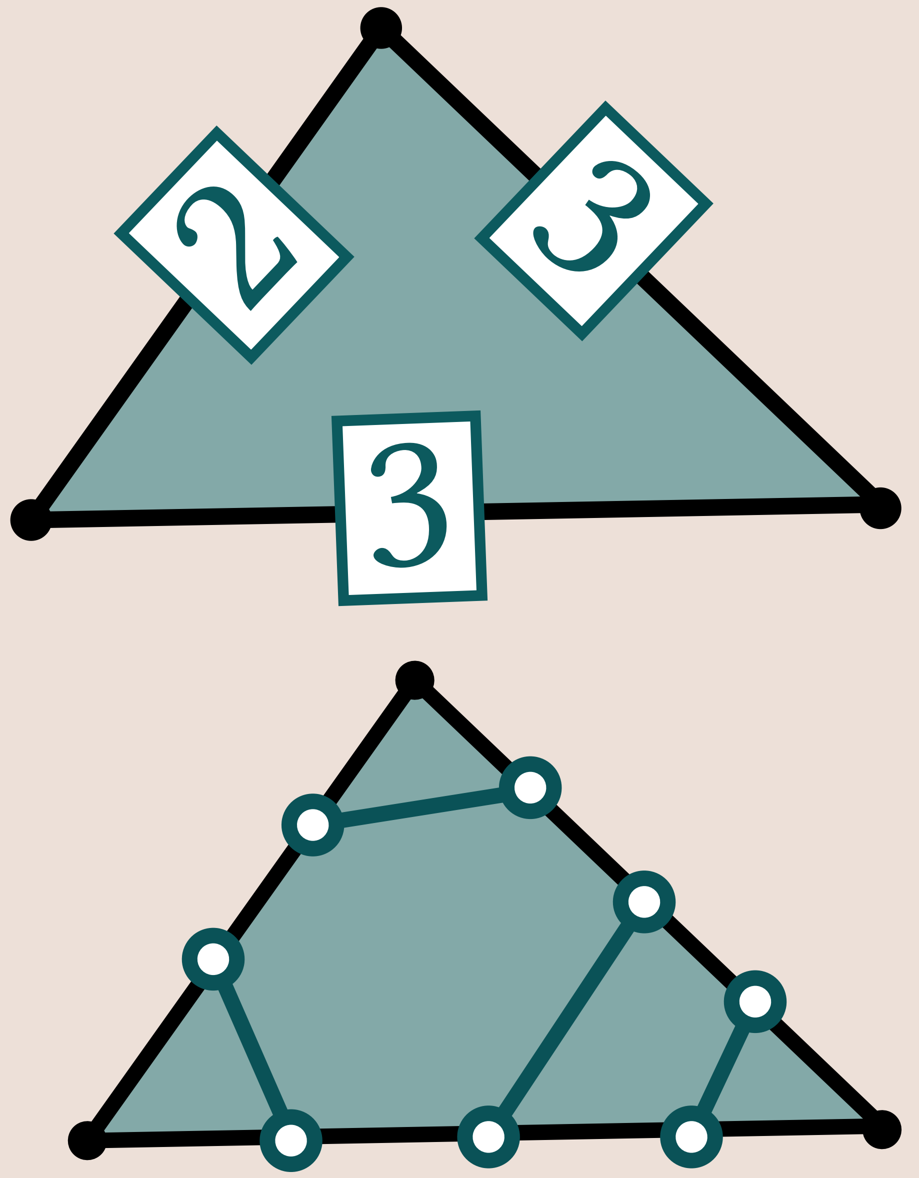
- Trace curve along mesh
  - Step one triangle at a time

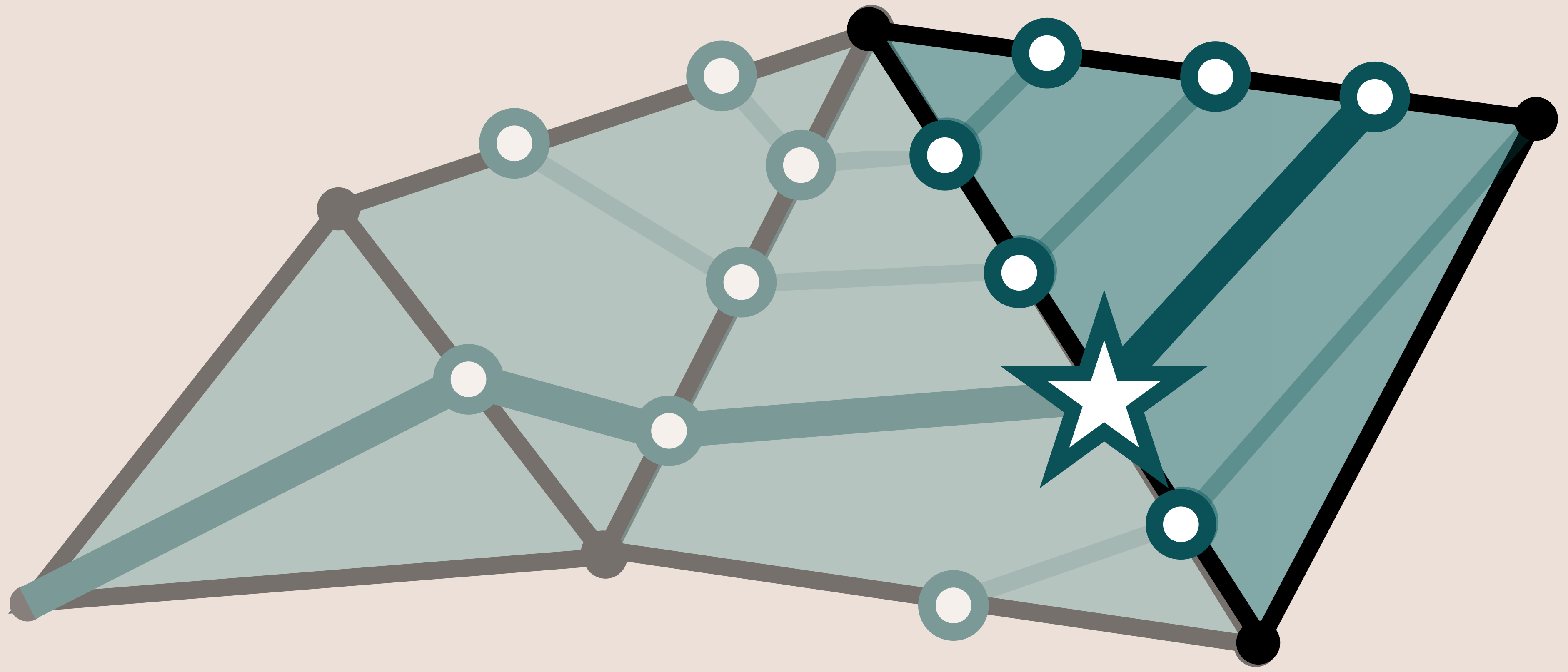




Recovering curves from normal coordinates

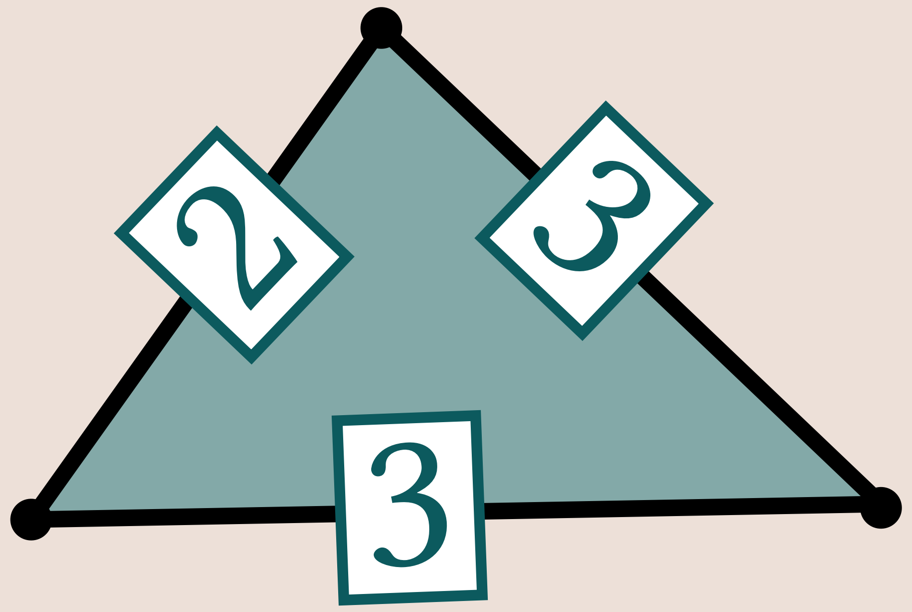
- Trace curve along mesh
  - Step one triangle at a time

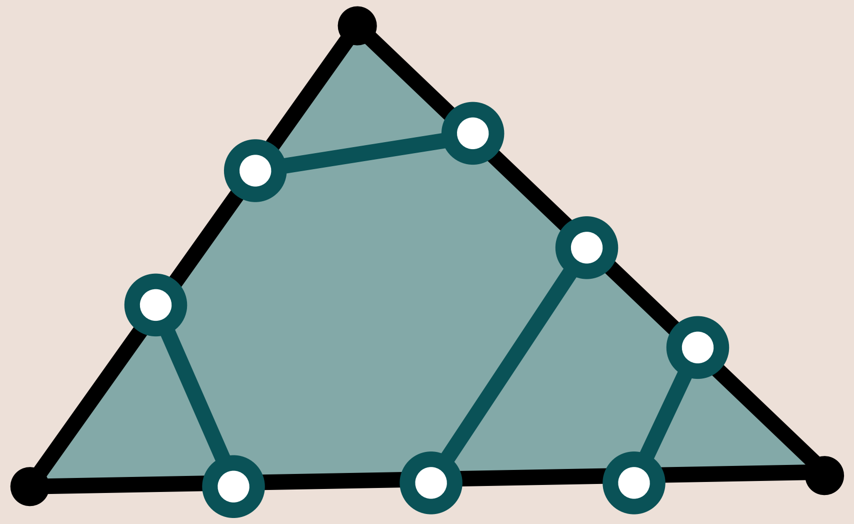


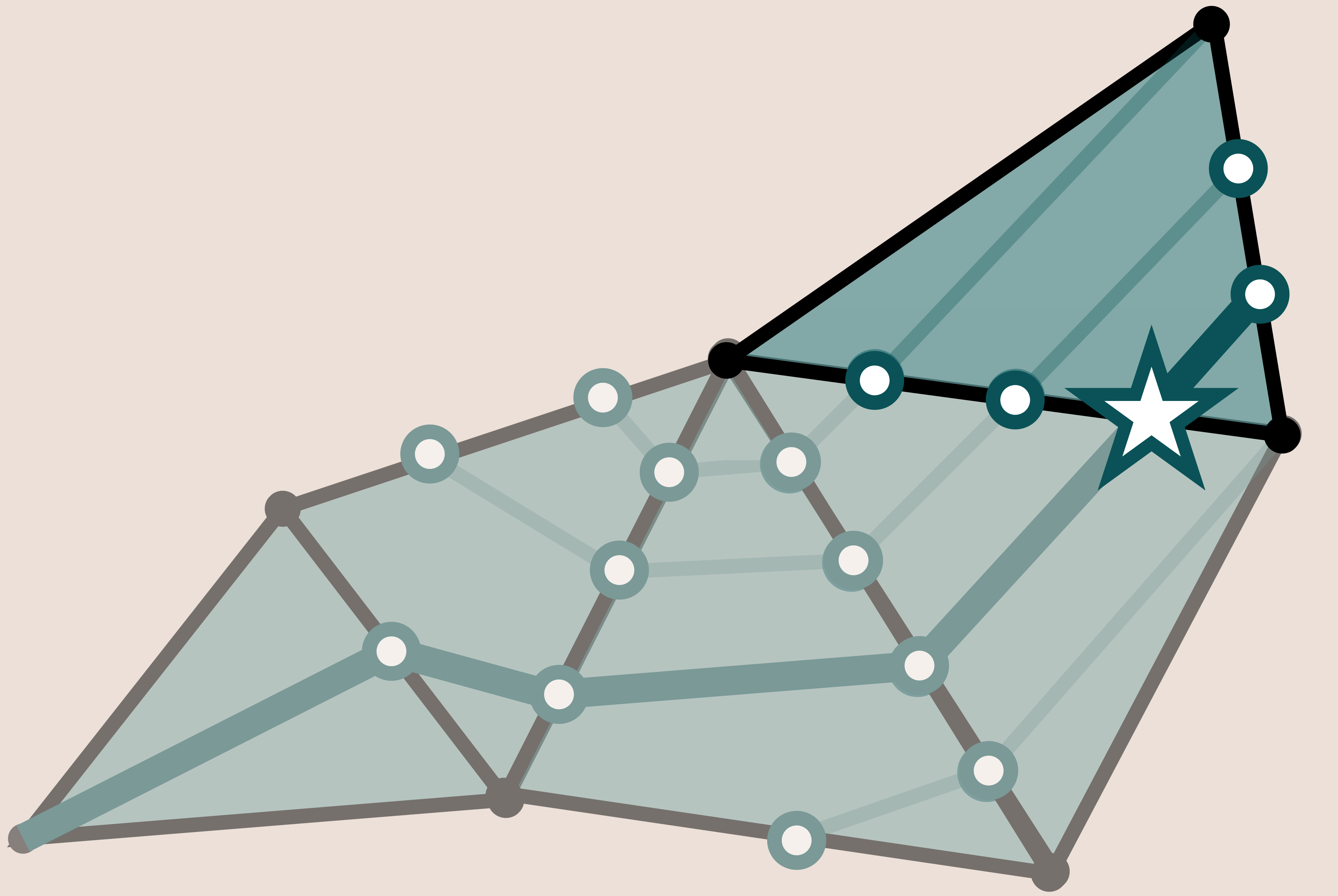


Recovering curves from normal coordinates

- Trace curve along mesh
  - Step one triangle at a time

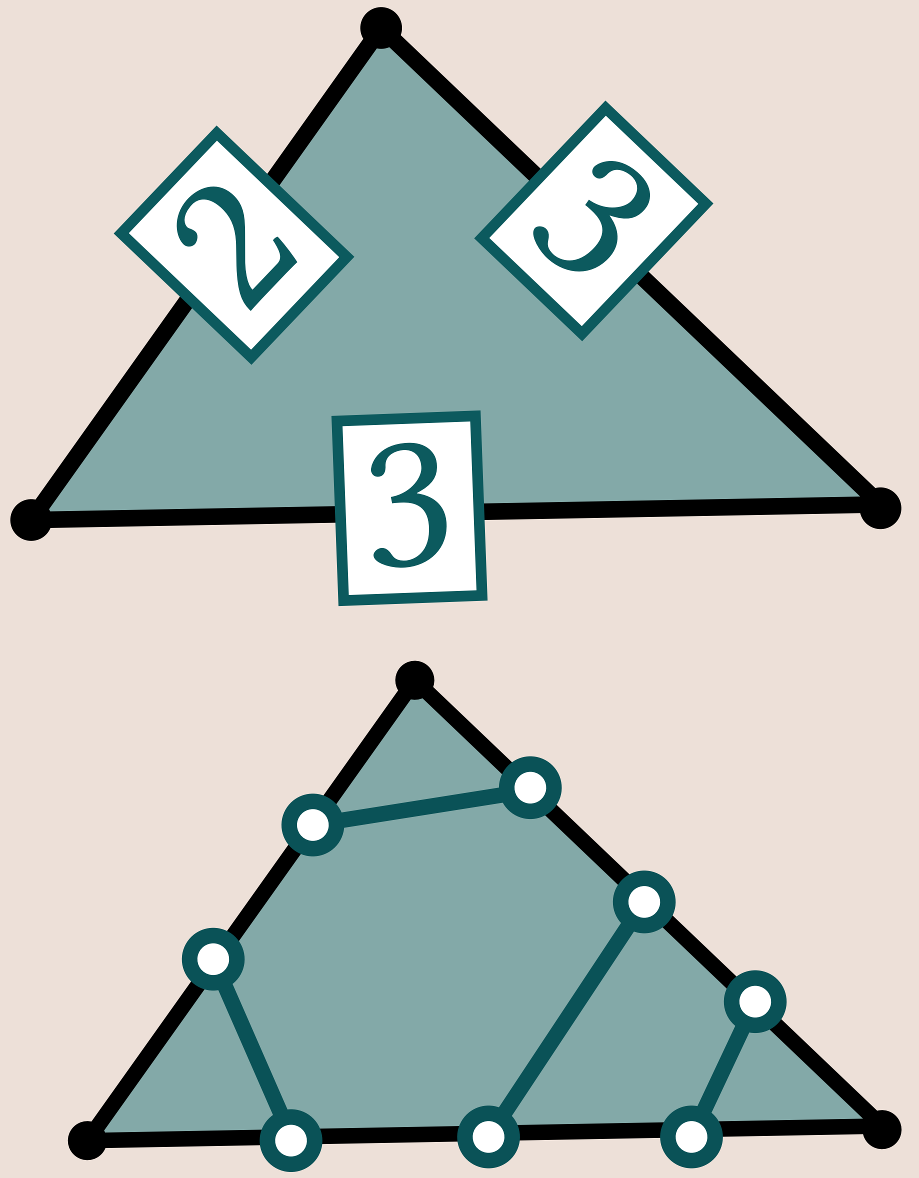


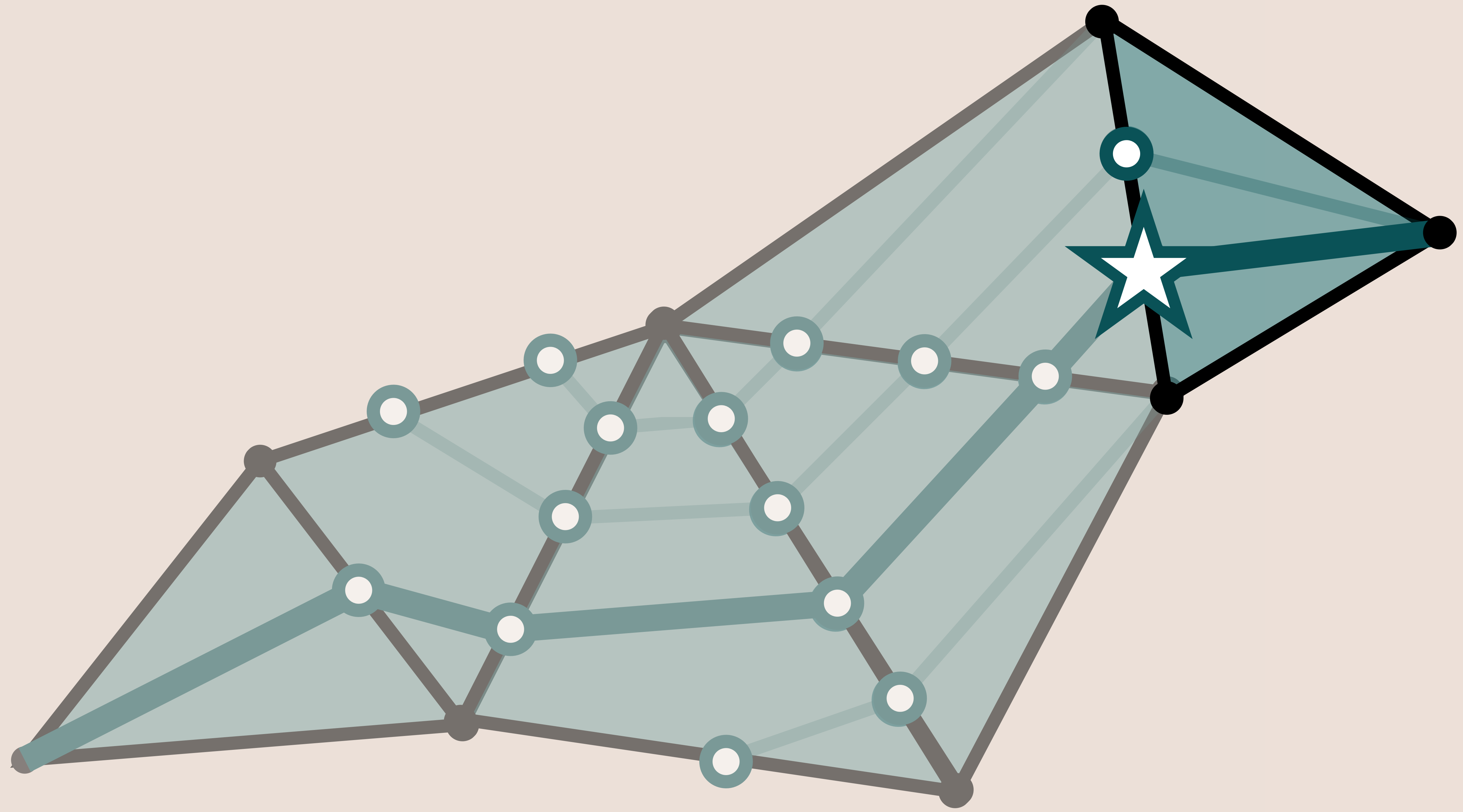




Recovering curves from normal coordinates

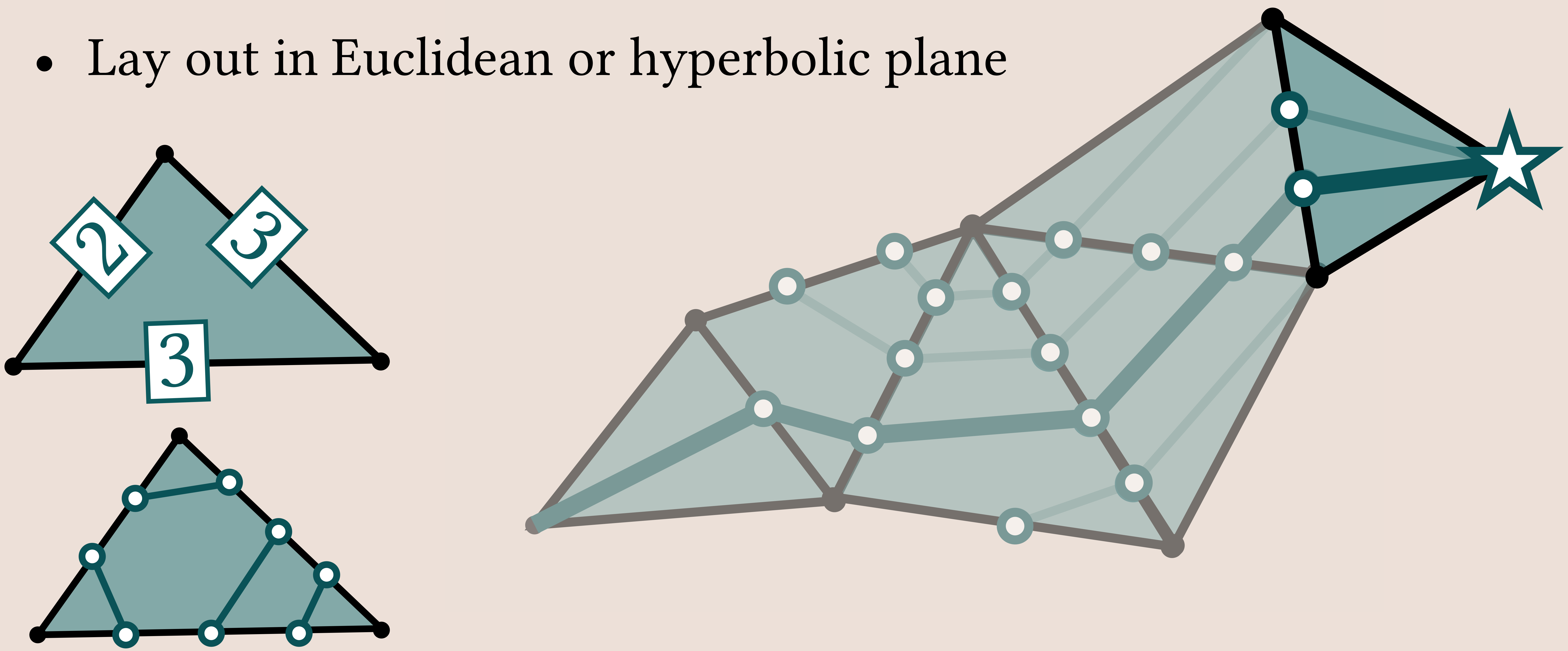
- Trace curve along mesh
  - Step one triangle at a time

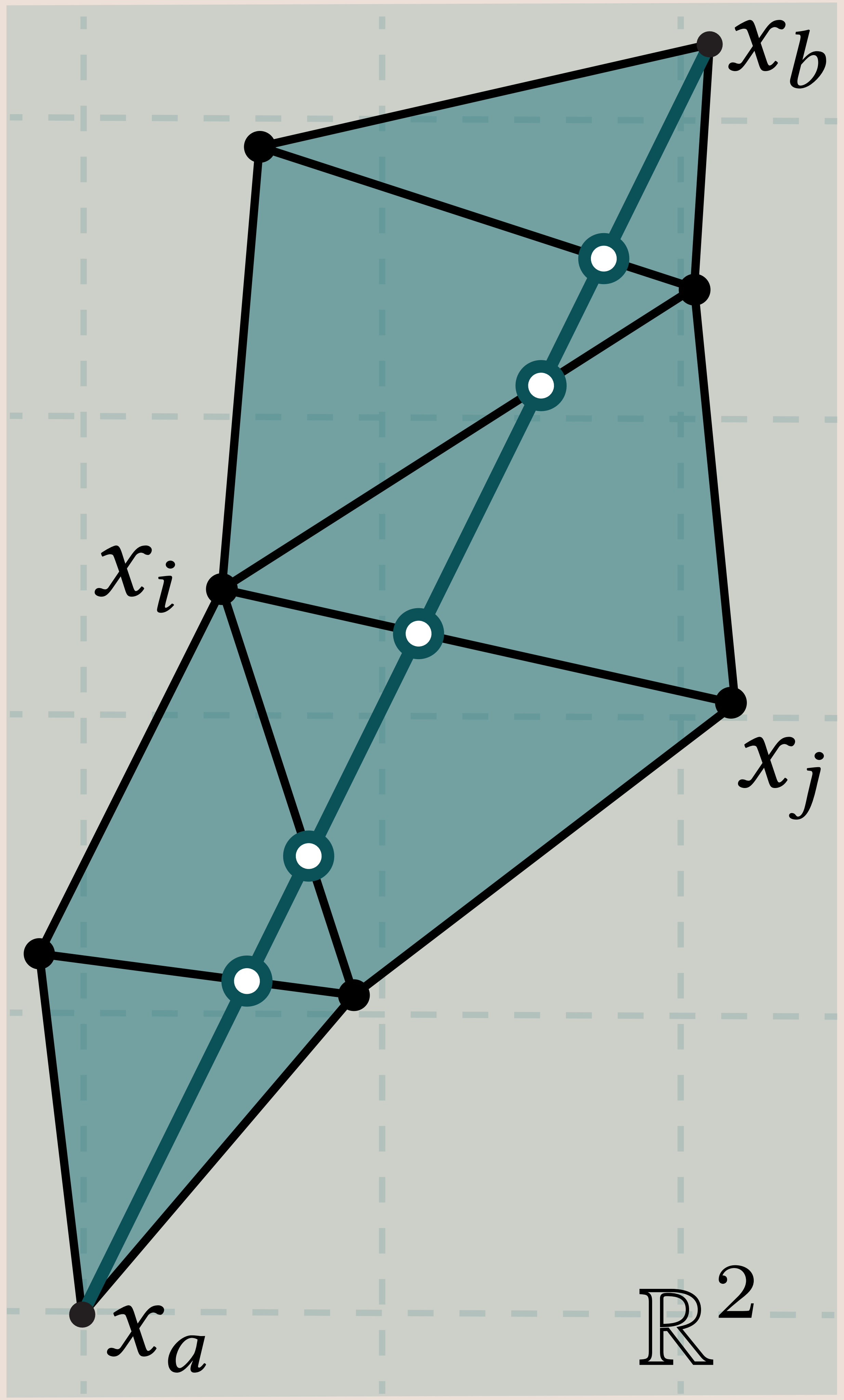




Recovering curves from normal coordinates

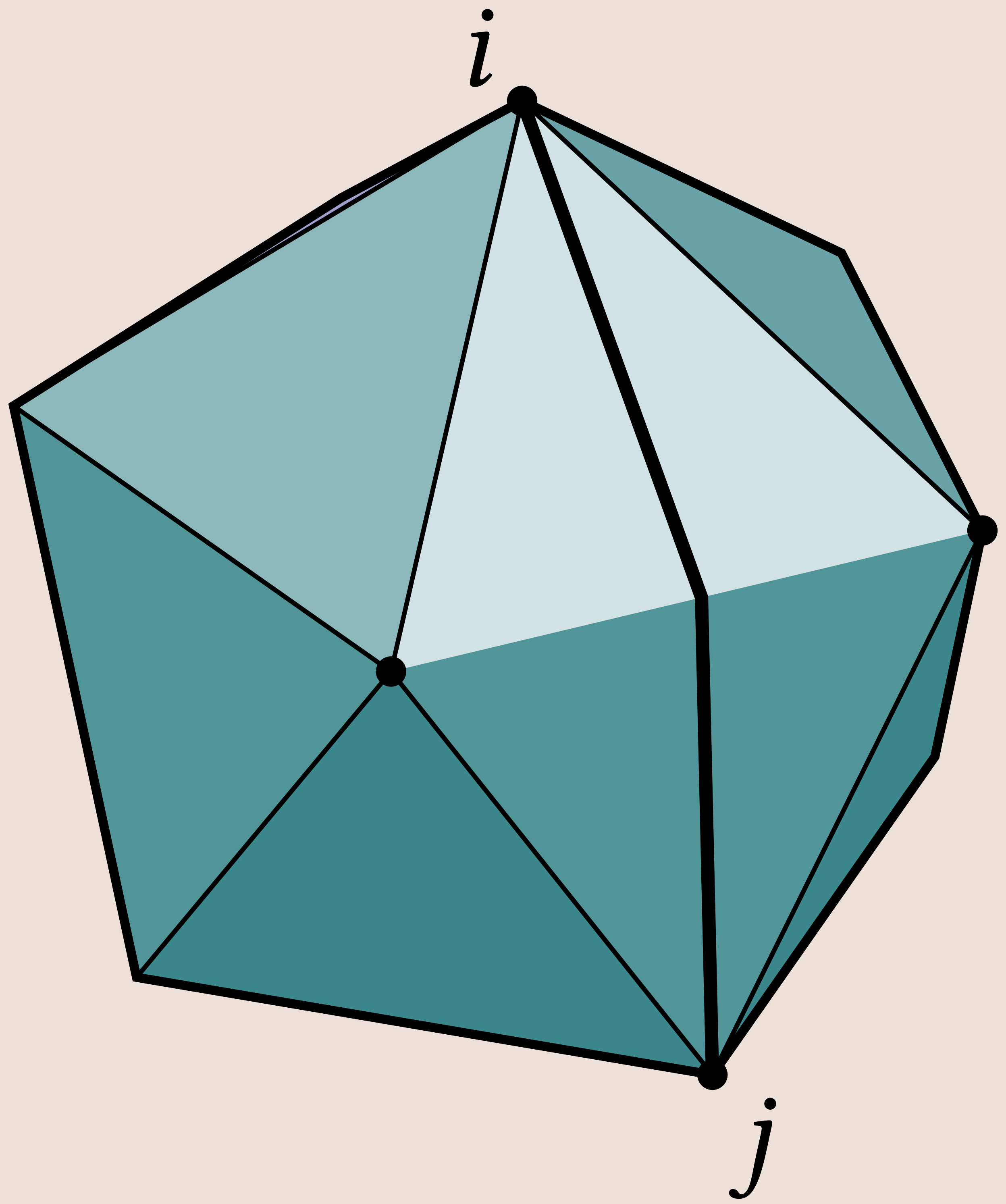
- Trace curve along mesh
  - Step one triangle at a time
  - Guaranteed to be correct triangle strip (depends only on integer data)
- Lay out in Euclidean or hyperbolic plane

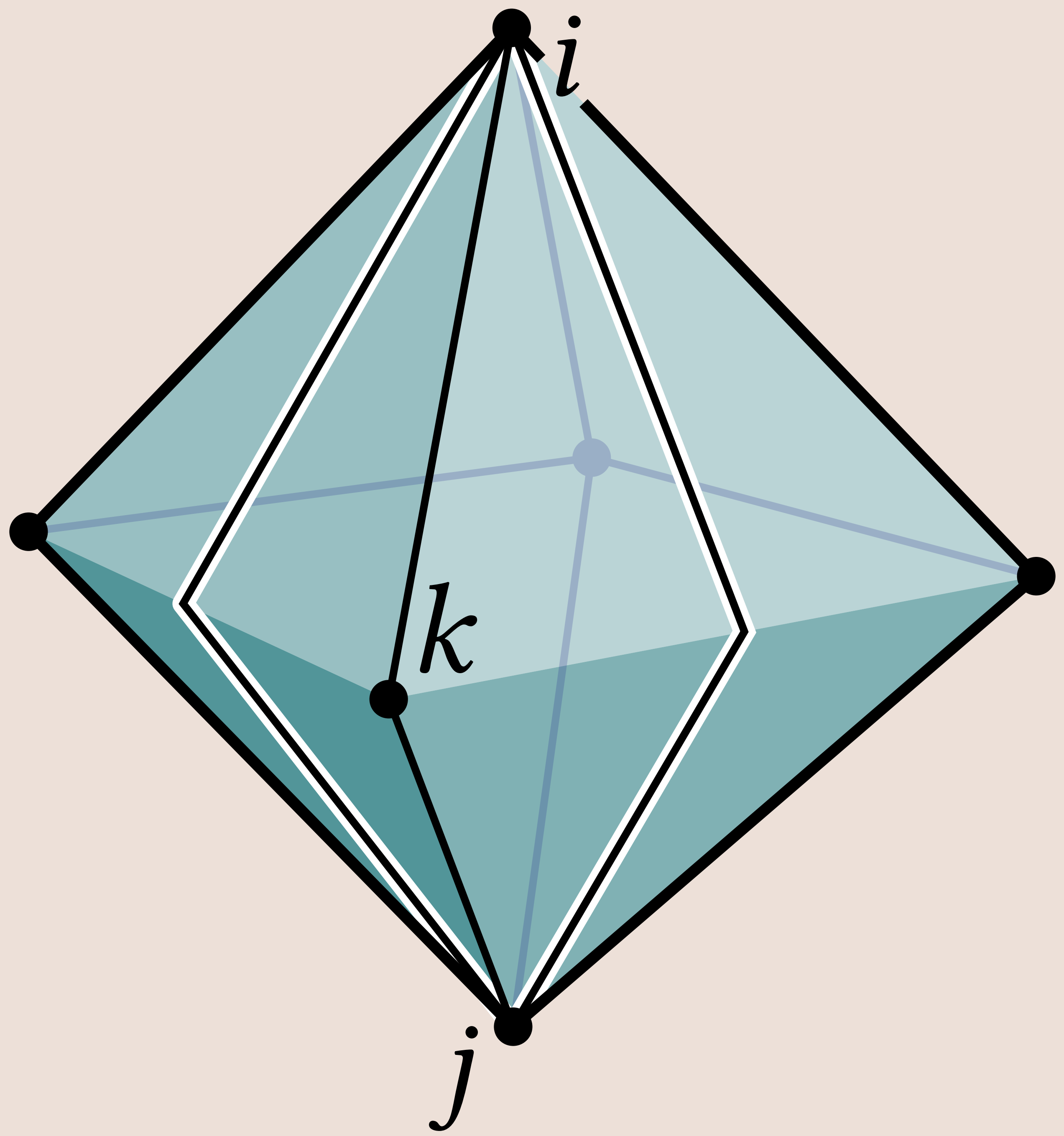


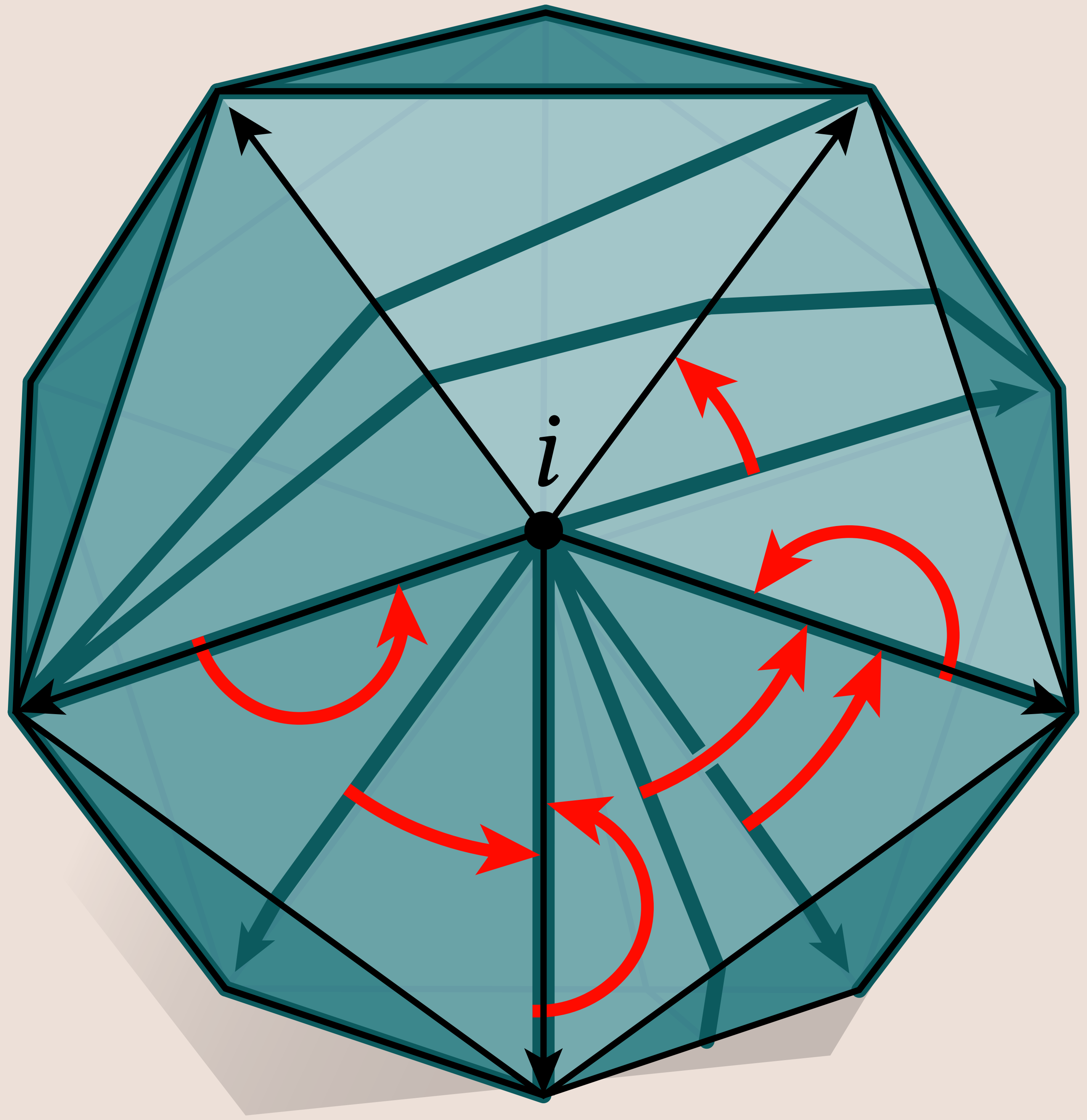


Roundabouts

- Problem: can't always tell which edge the traced curve corresponds to
  - Attempt 1: Inspect curve endpoints
  - New idea: roundabouts - encode which edge each traced curve corresponds to







roundabout at vertex i

Final algorithm

1. Find scale factors
   - Maintain edge lengths, normal coordinates, roundabouts

2. Planar layout

3. Trace edges to get explicit correspondence

4. Interpolate texture coordinates

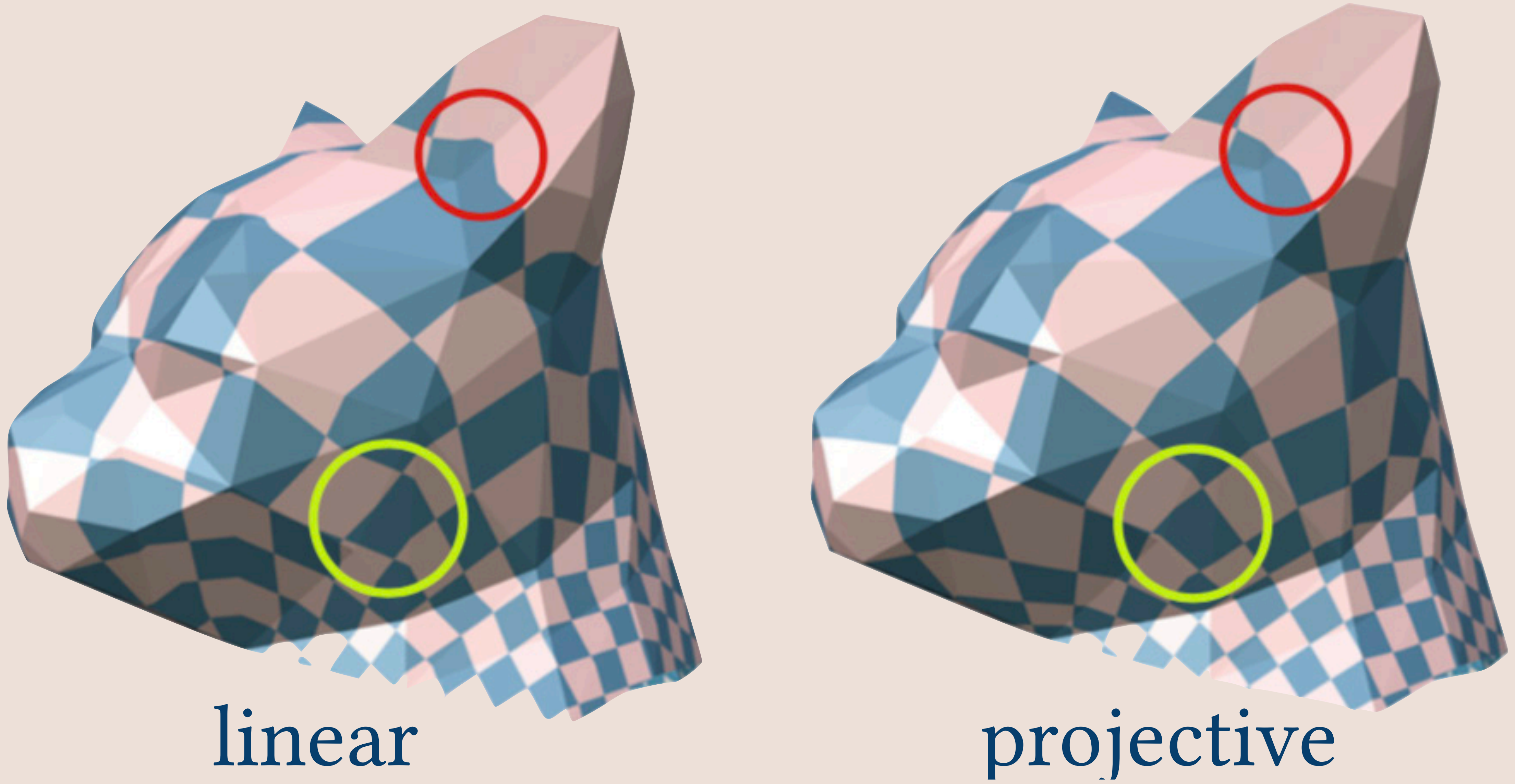
Texture Interpolation

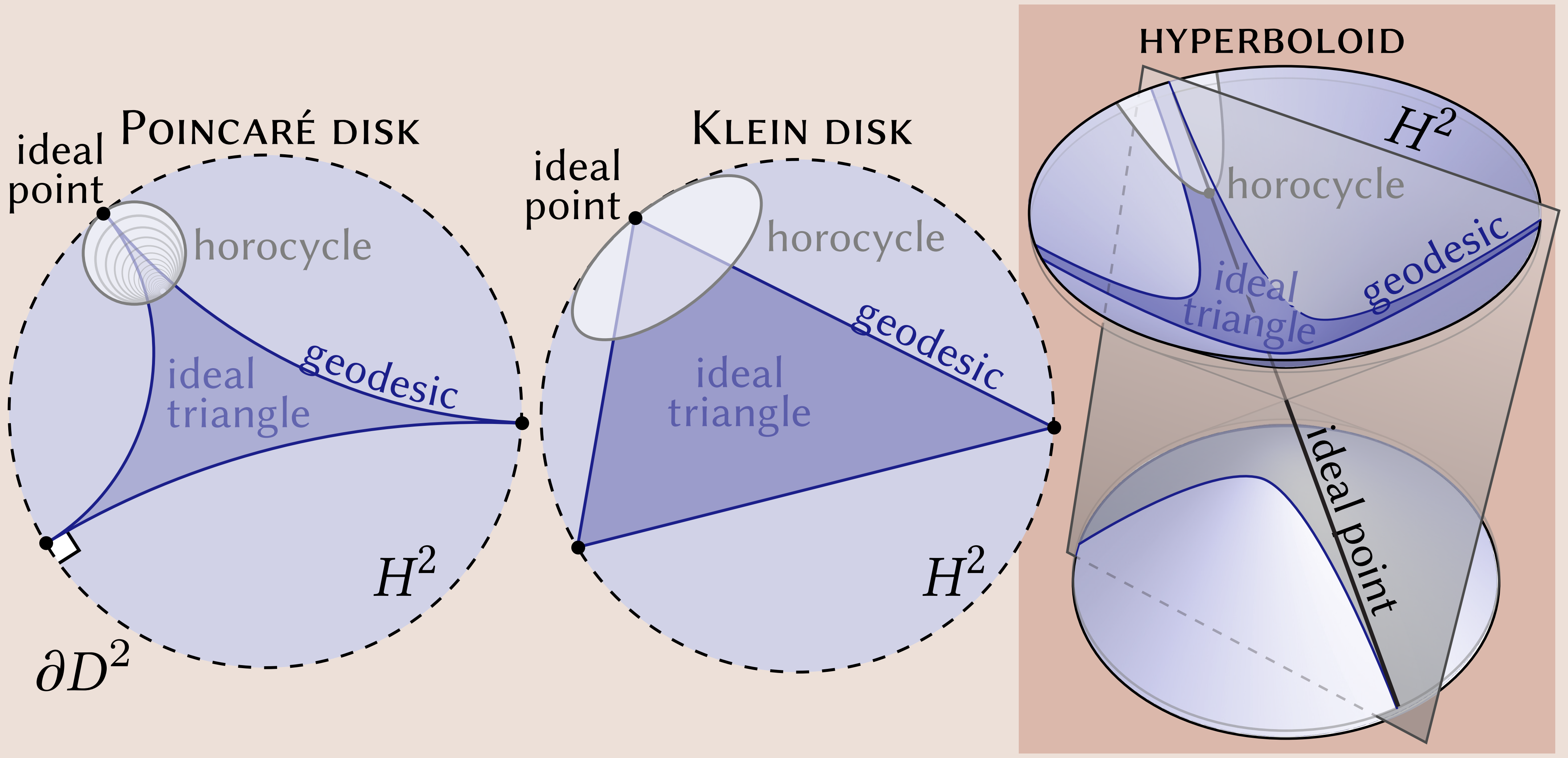
in the light cone

Projective interpolation

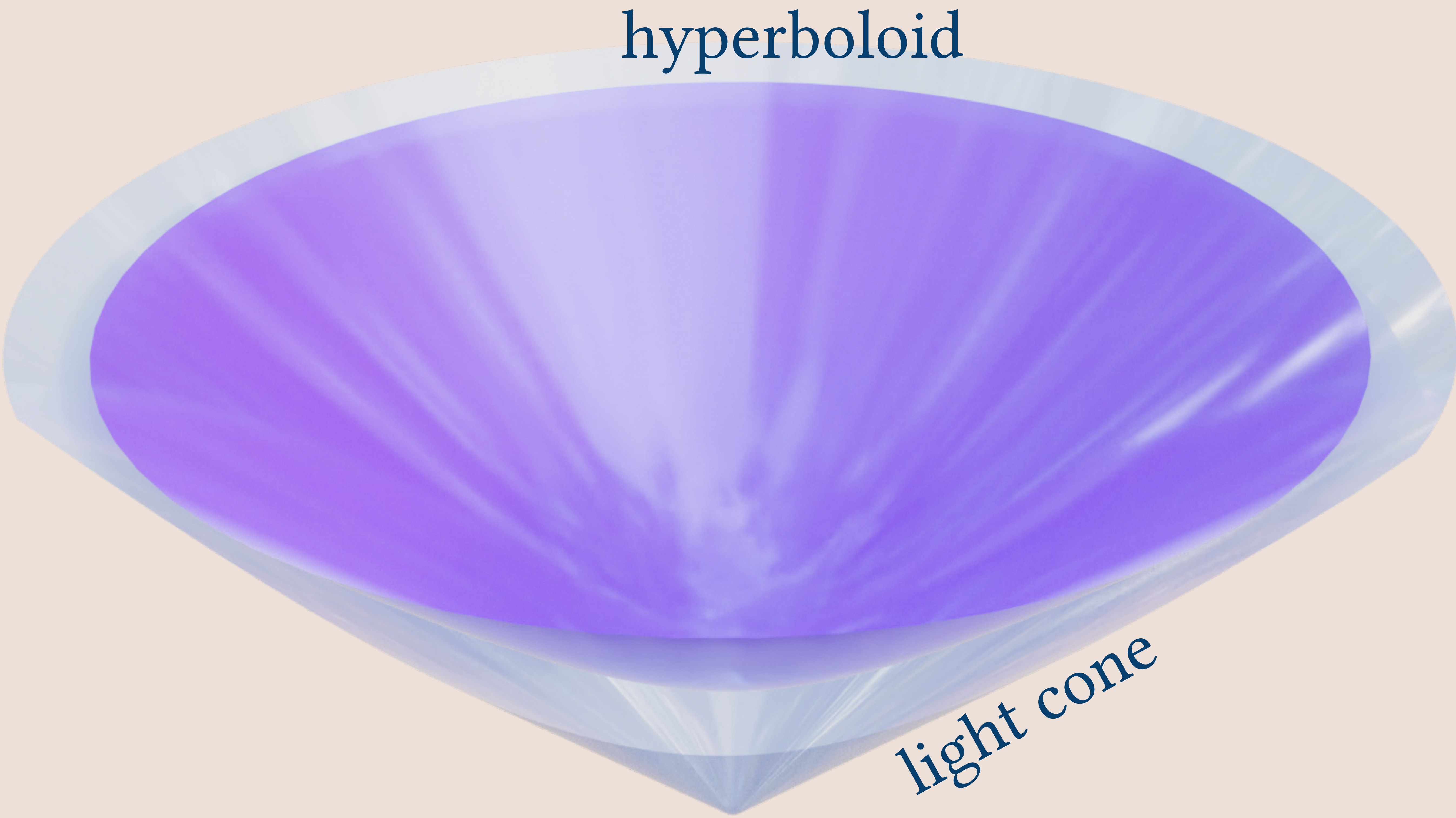
- [Springborn+ 2008]: projective interpolation
- Problem: what should you do in variable triangulation case?
  - Solution: lay out in hyperboloid

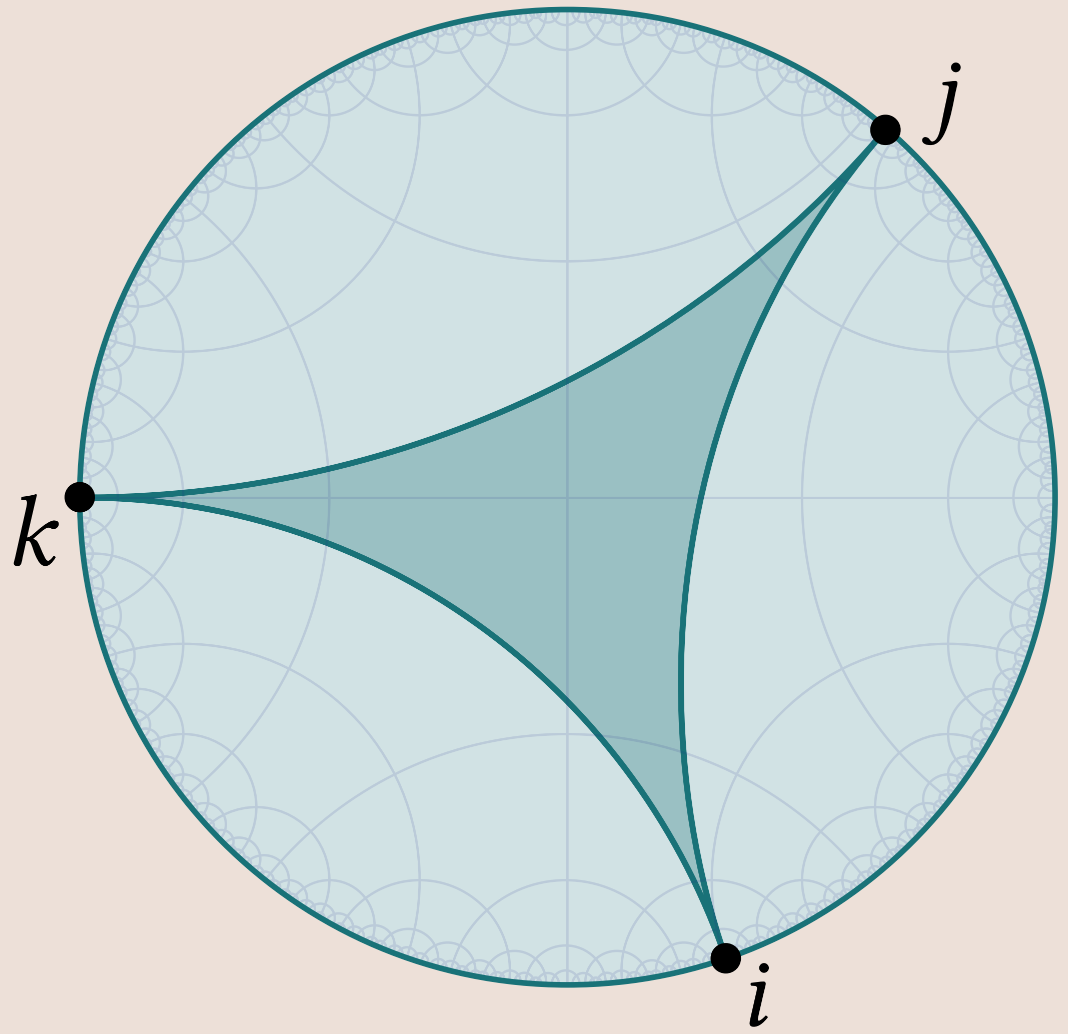
[Springborn+ 2008]



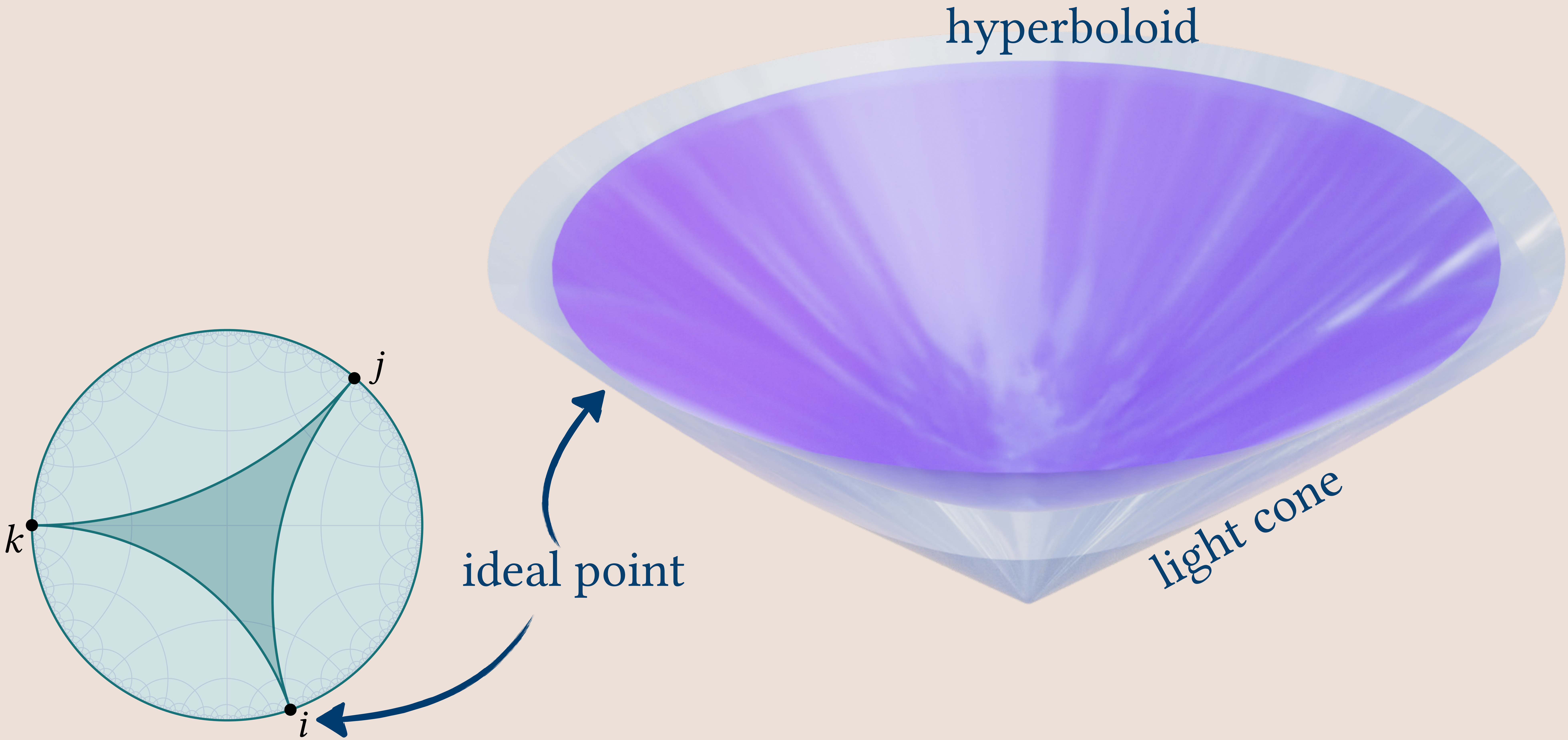


The hyperboloid and the light cone

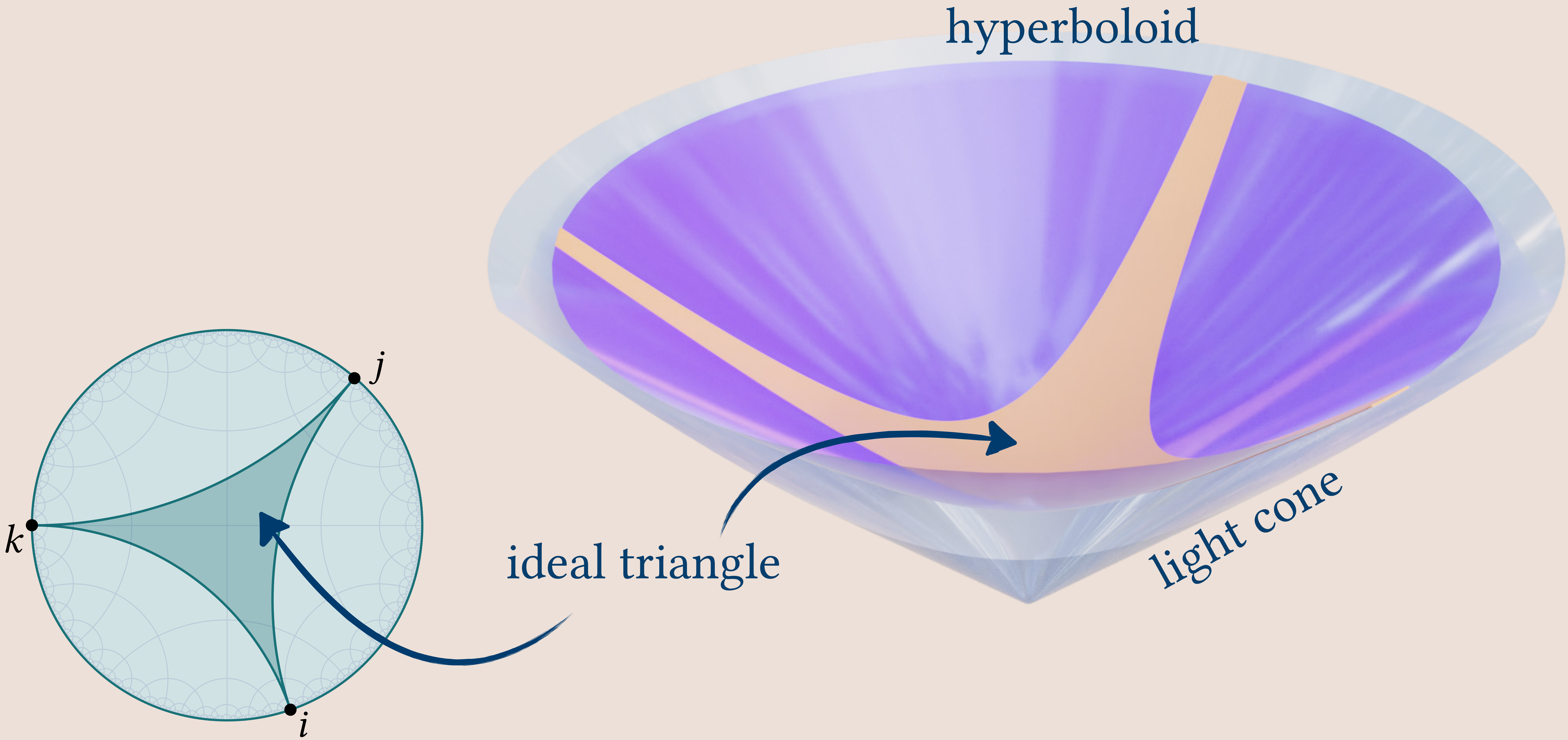




The hyperboloid and the light cone

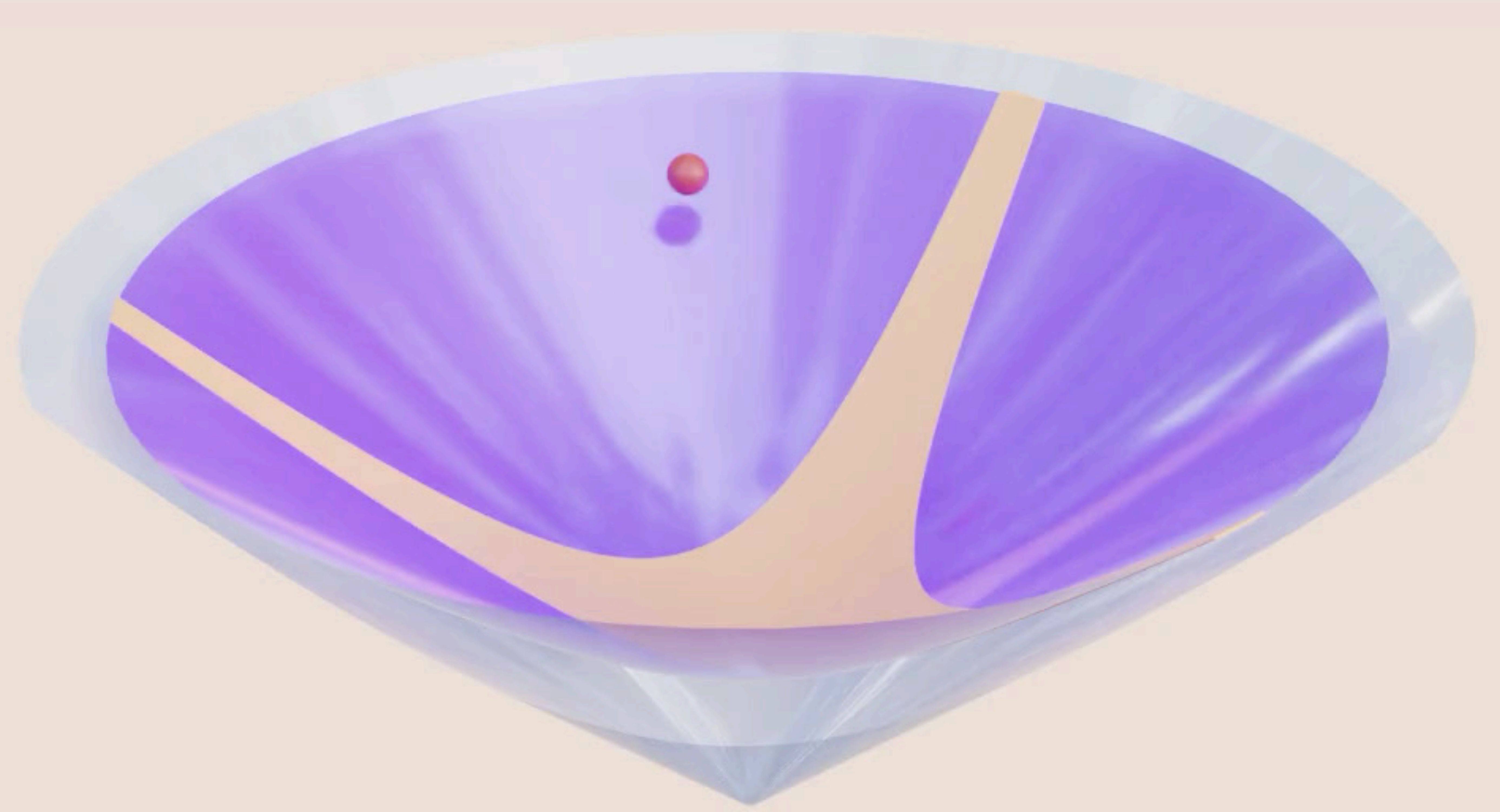


The hyperboloid and the light cone



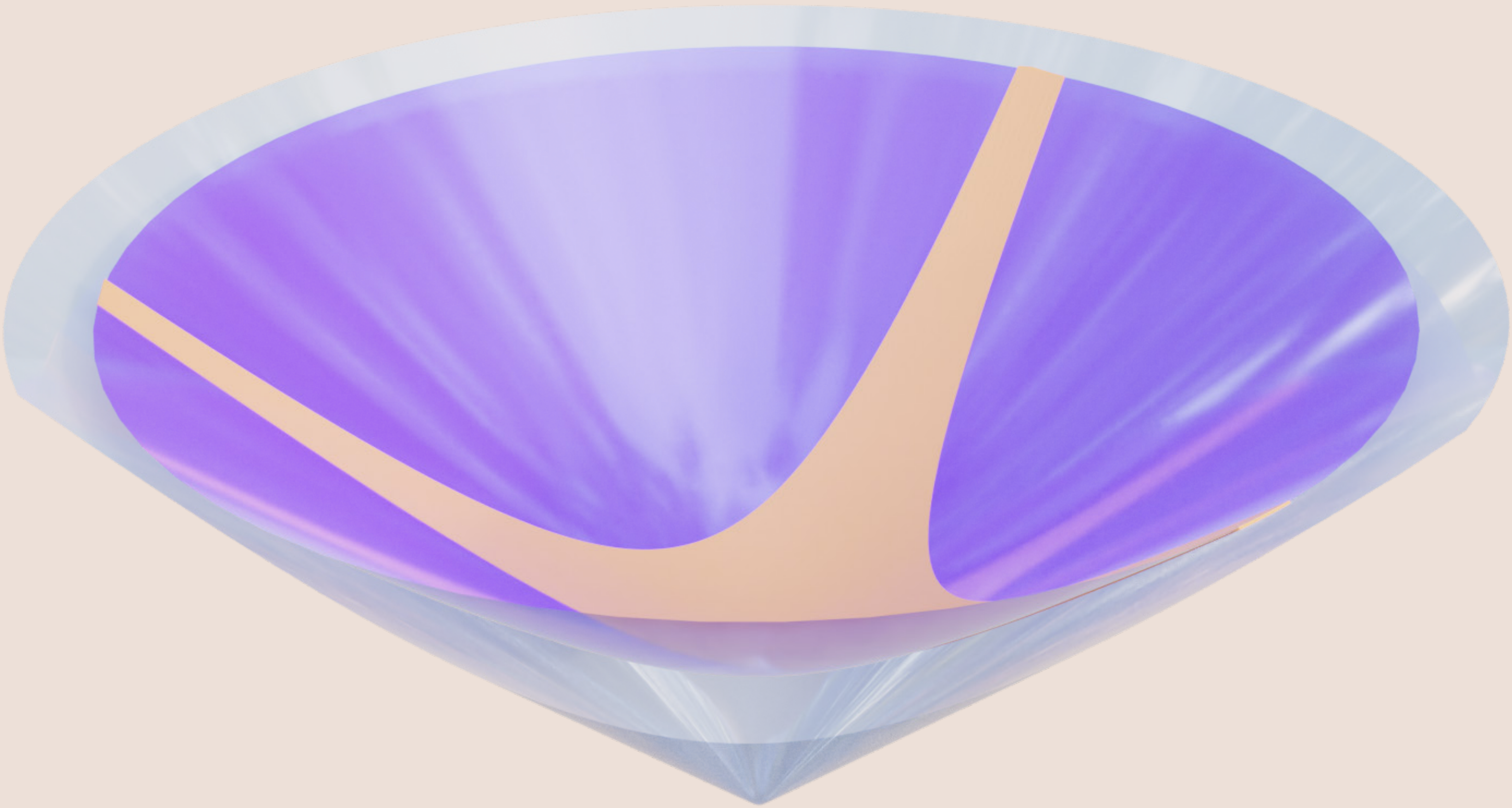
The hyperboloid and the light cone

- Normalize points



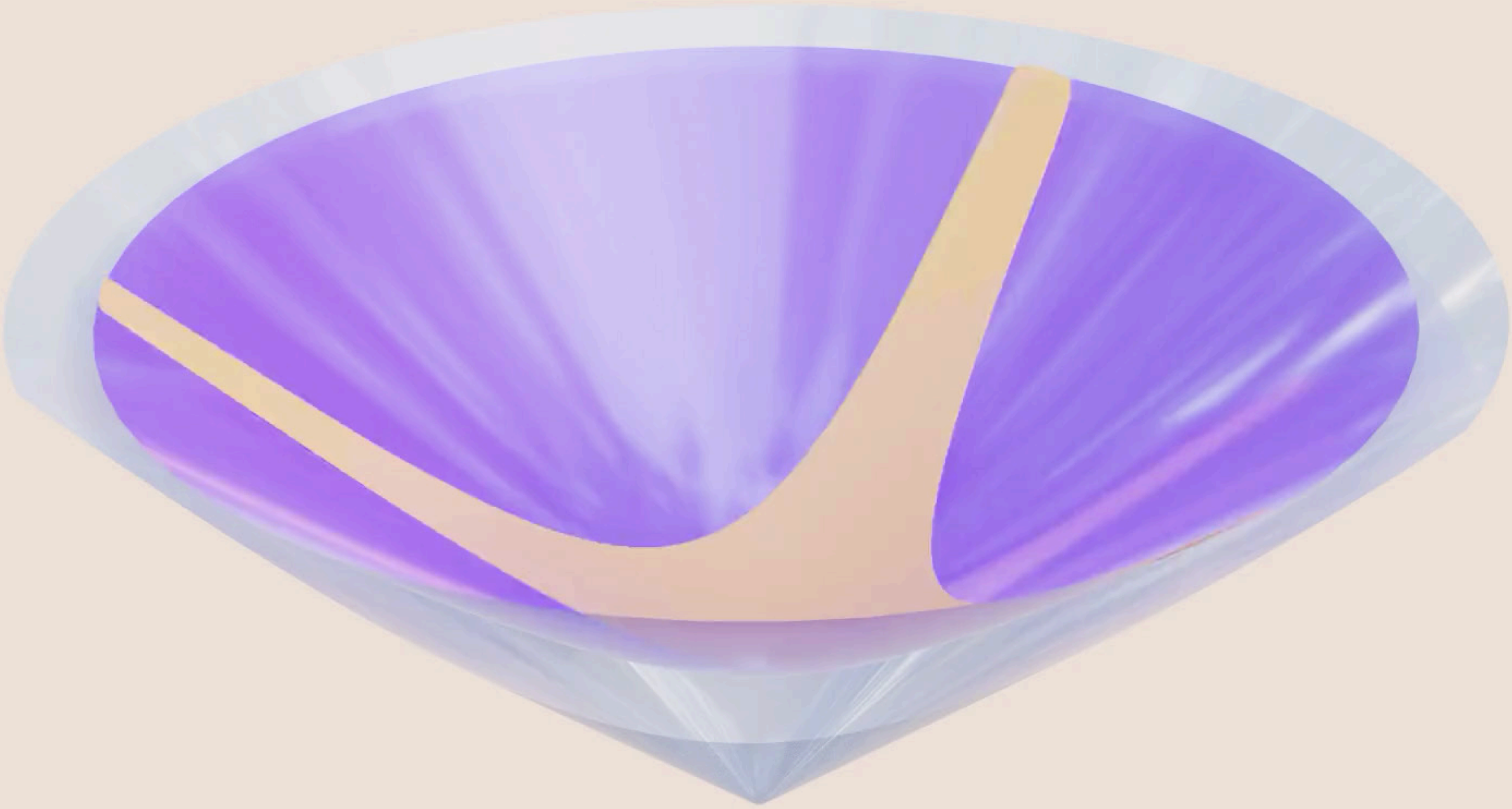
The hyperboloid and the light cone

- Normalize points
- Ideal triangle $\longleftrightarrow$ inscribed Euclidean triangle



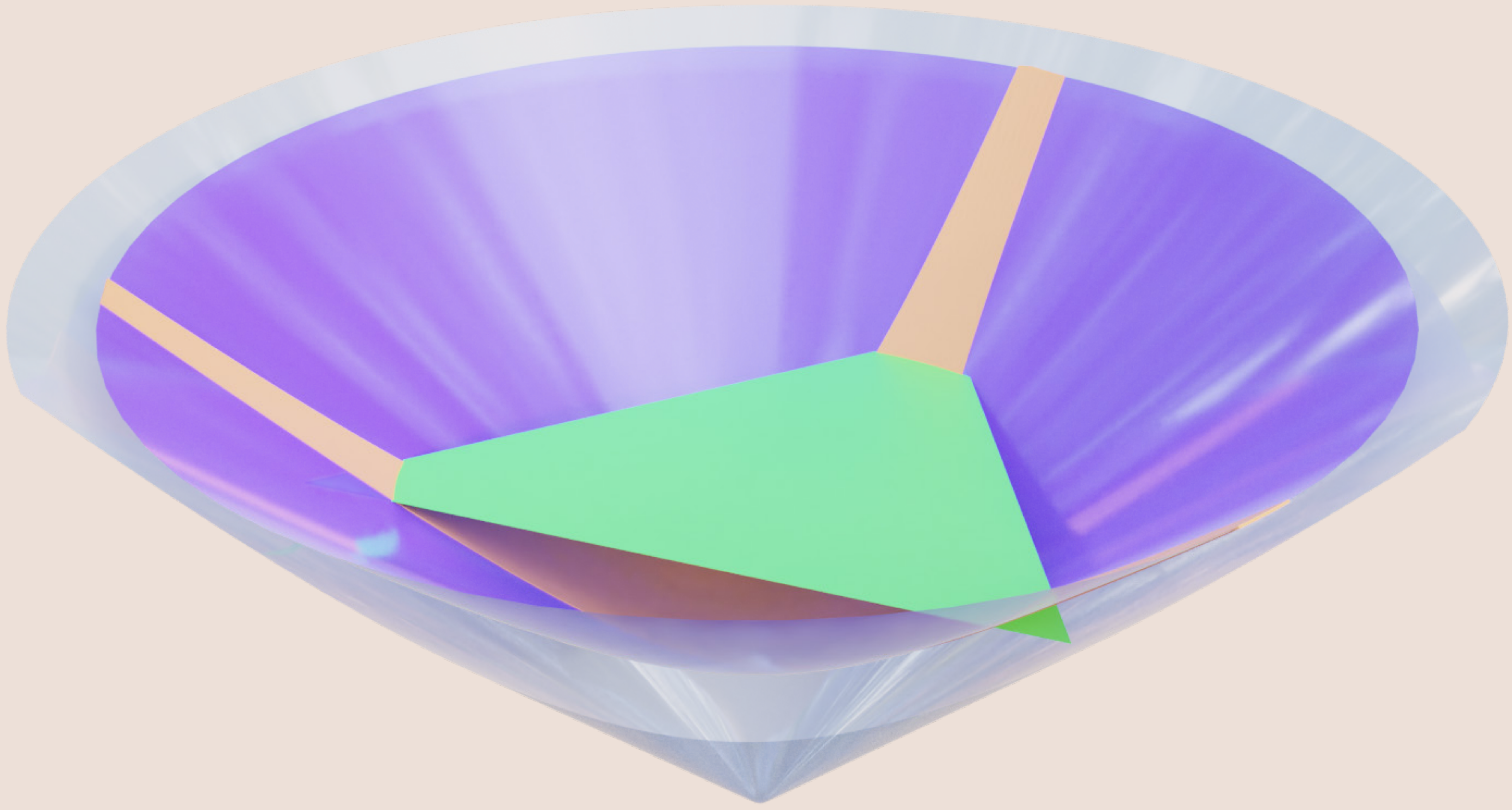
The hyperboloid and the light cone

- Normalize points
- Ideal triangle $\longleftrightarrow$ inscribed Euclidean triangle



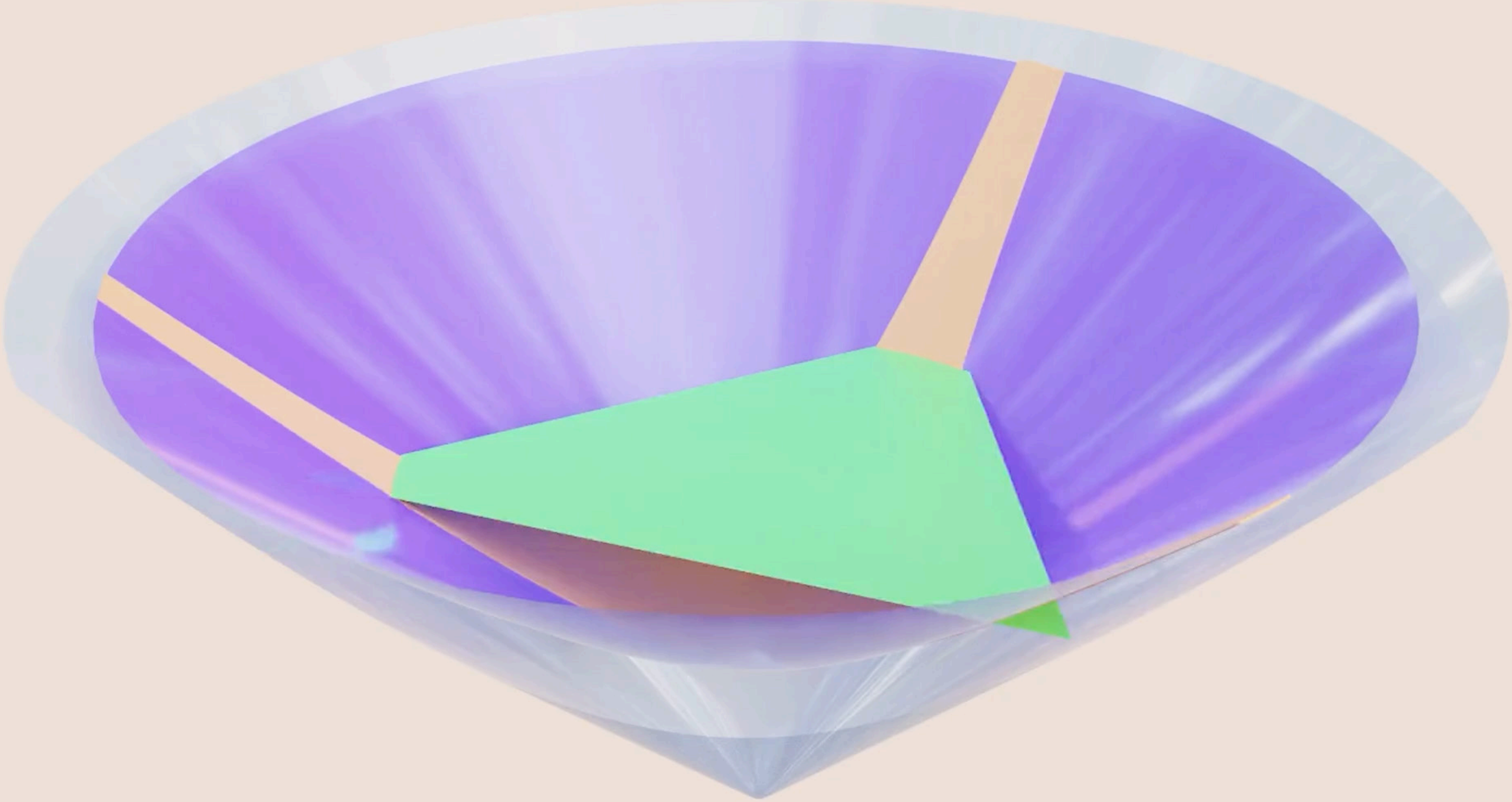
The hyperboloid and the light cone

- Normalize points
- Ideal triangle $\longleftrightarrow$ inscribed Euclidean triangle

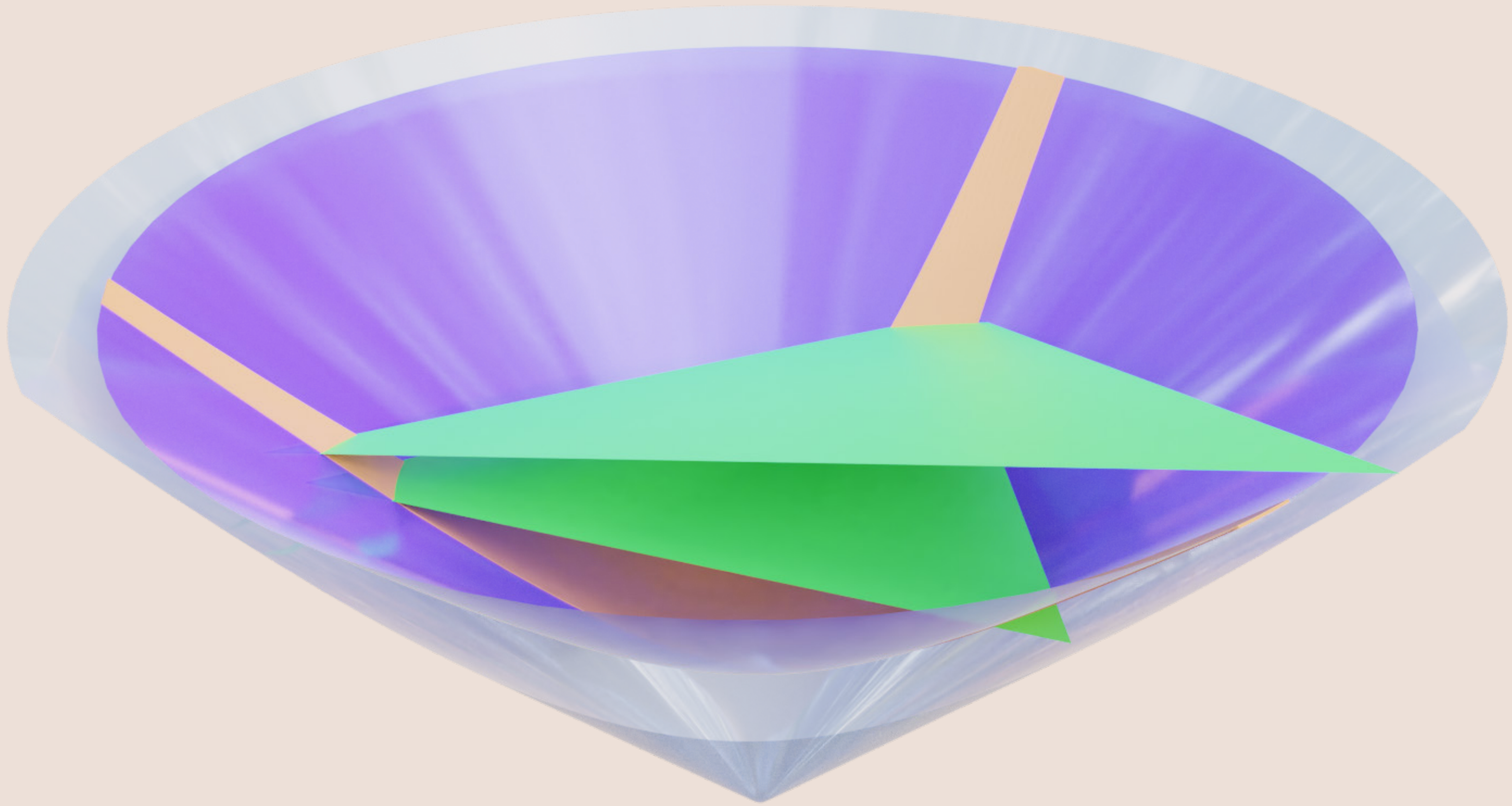


The hyperboloid and the light cone

- Normalize points
- Ideal triangle $\longleftrightarrow$ inscribed Euclidean triangle
- Vertex scaling $\longleftrightarrow$ scaling vertices along light cone

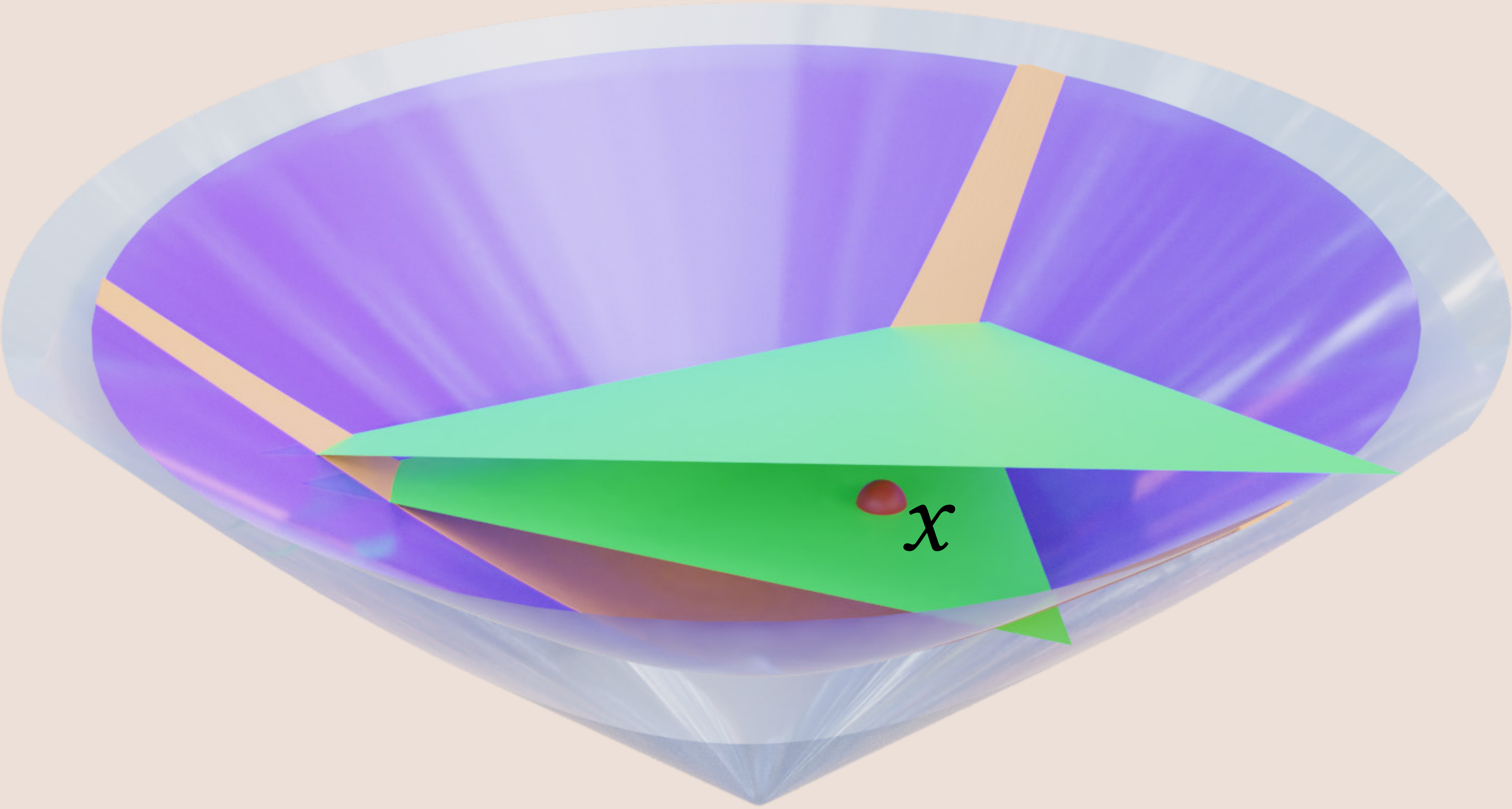


Projective maps



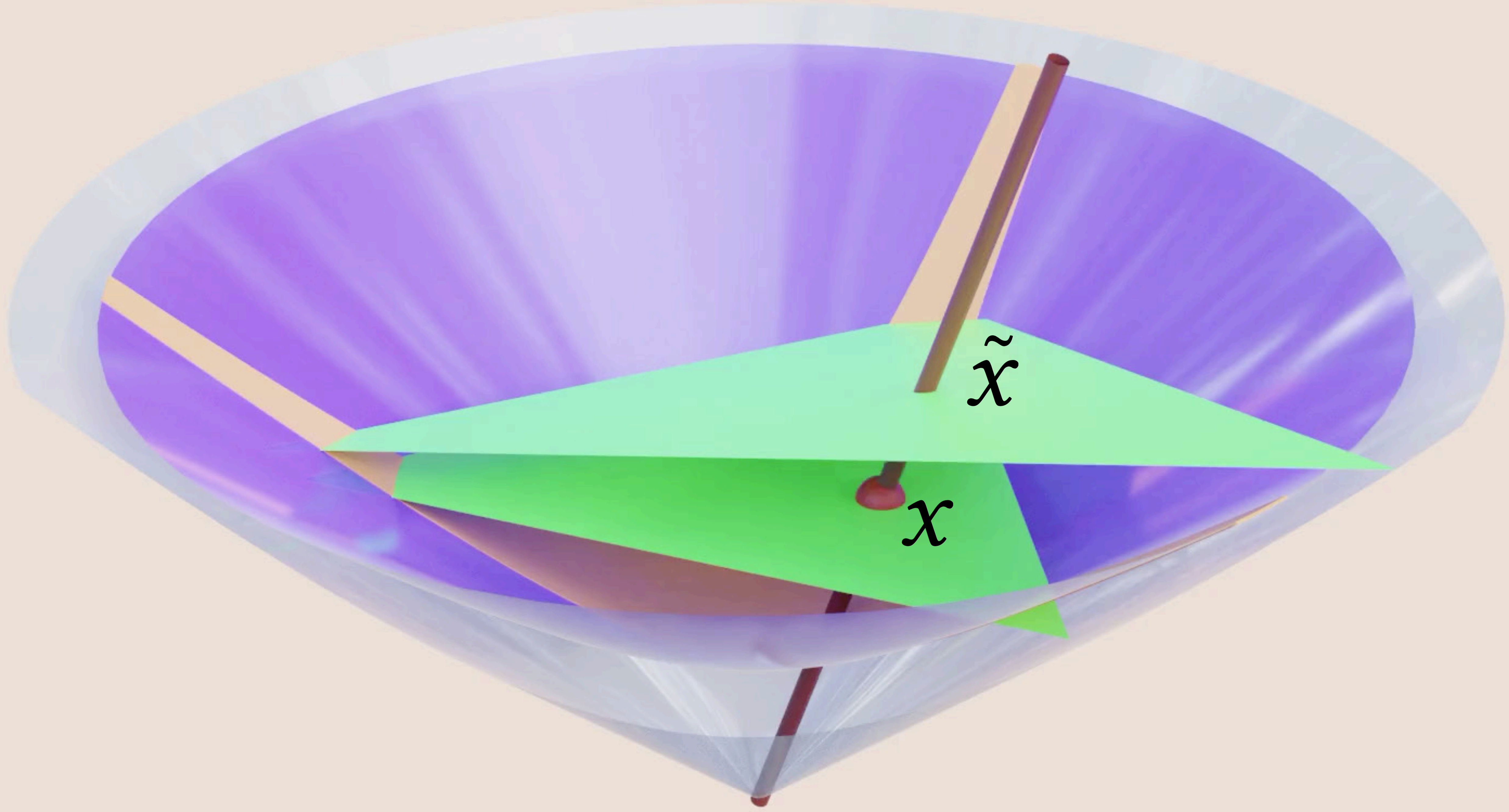
fixed triangulation

Projective maps



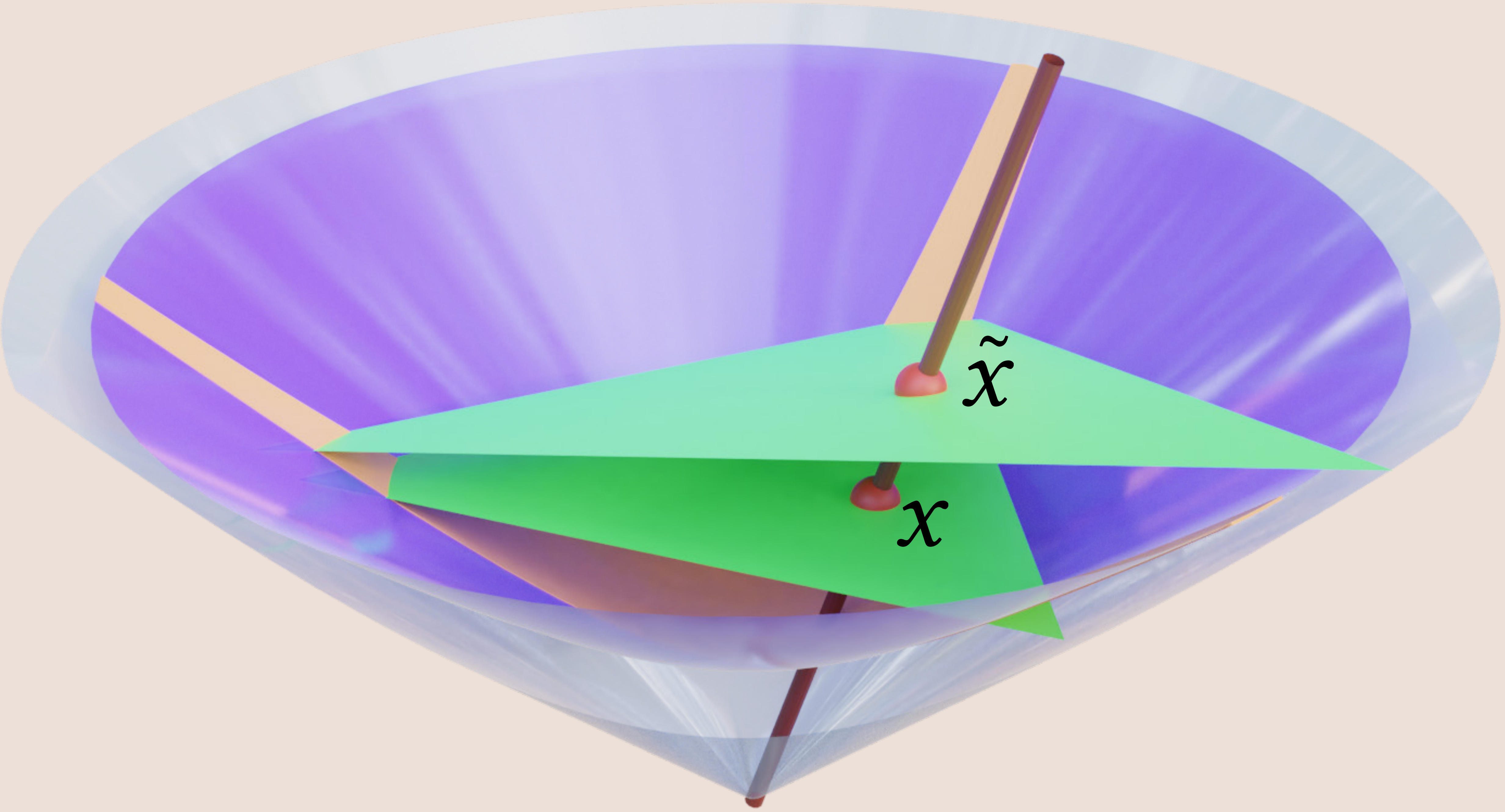
fixed triangulation

Projective maps

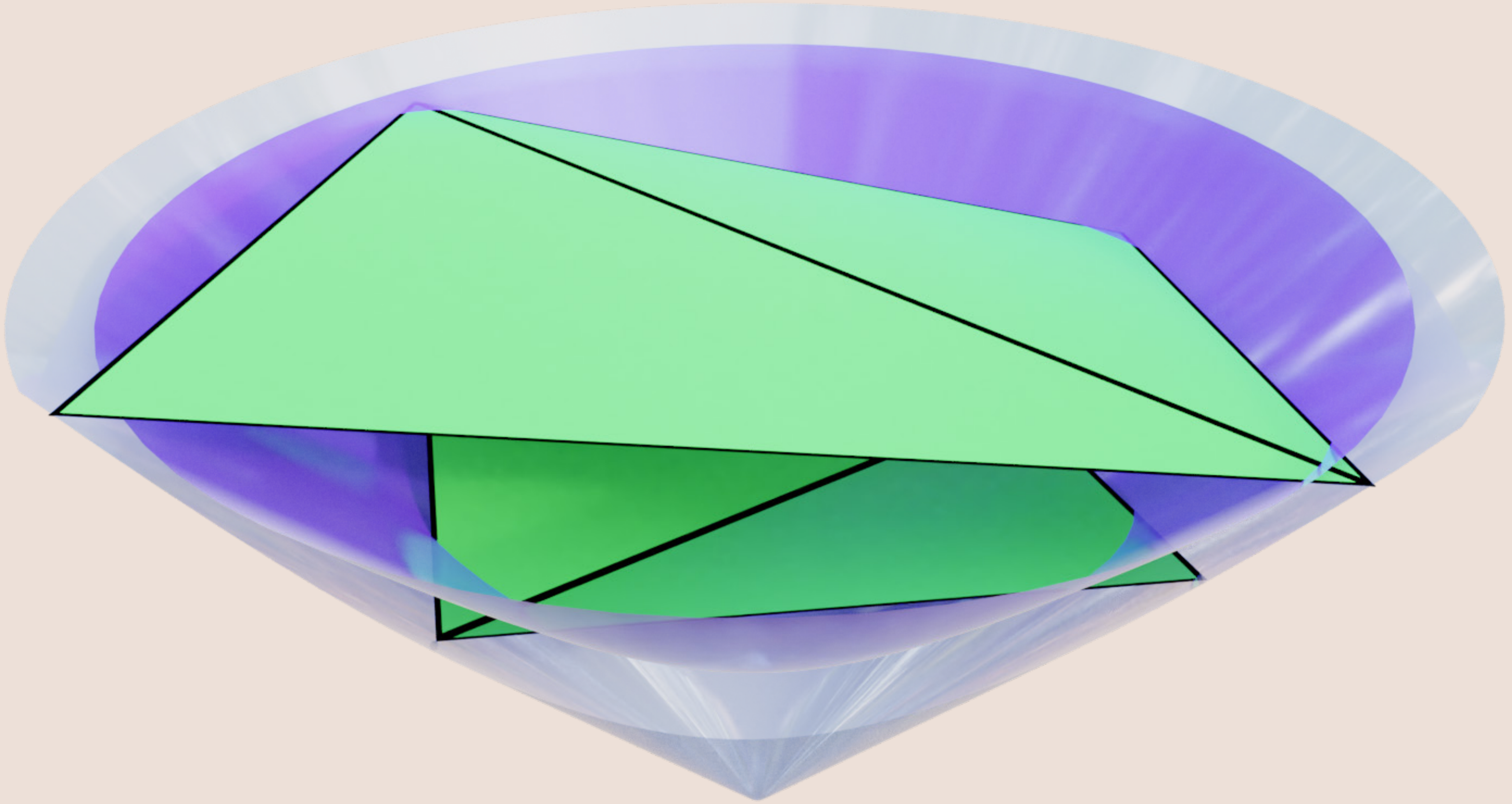


fixed triangulation

Projective maps

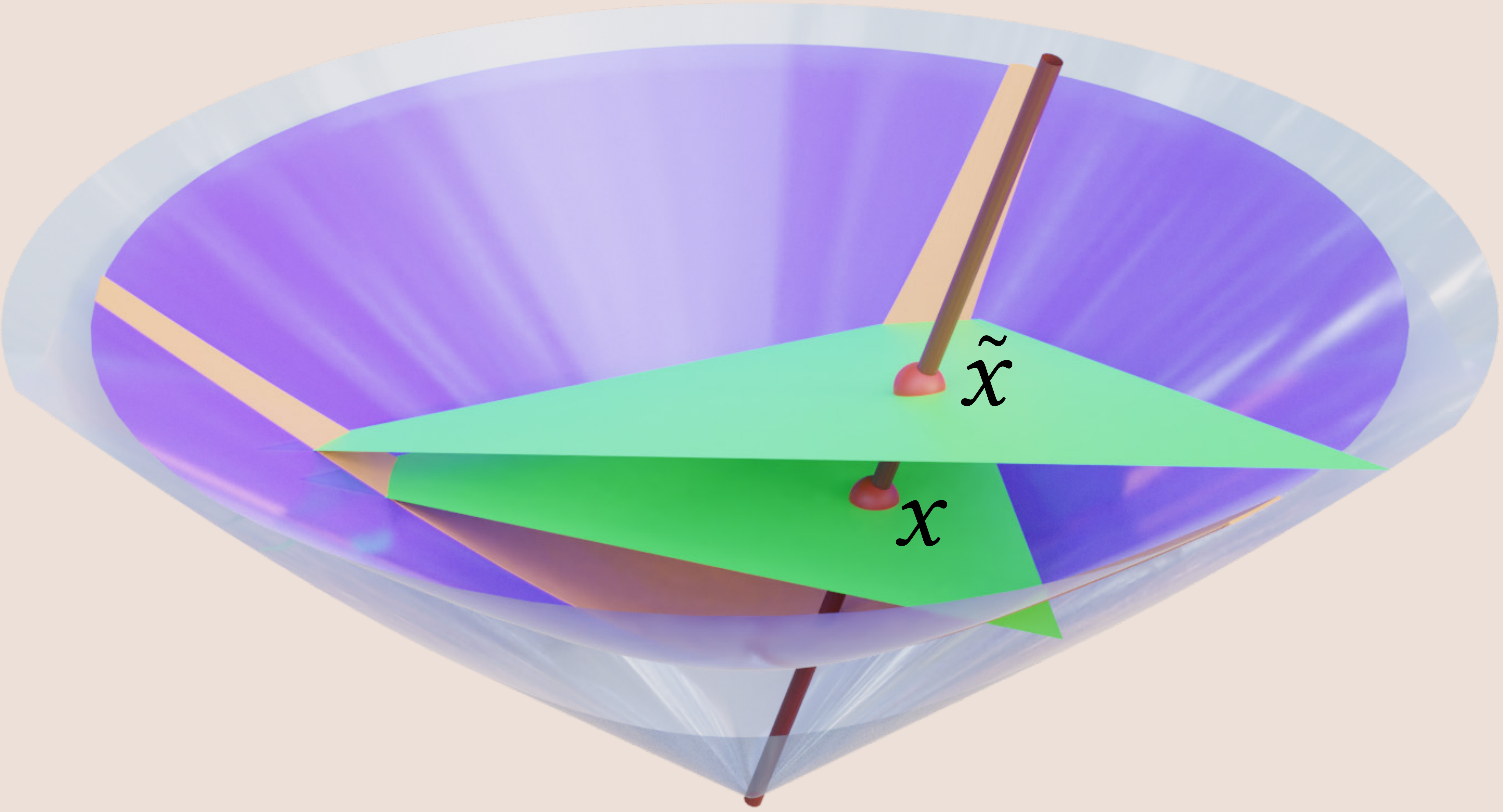


fixed triangulation

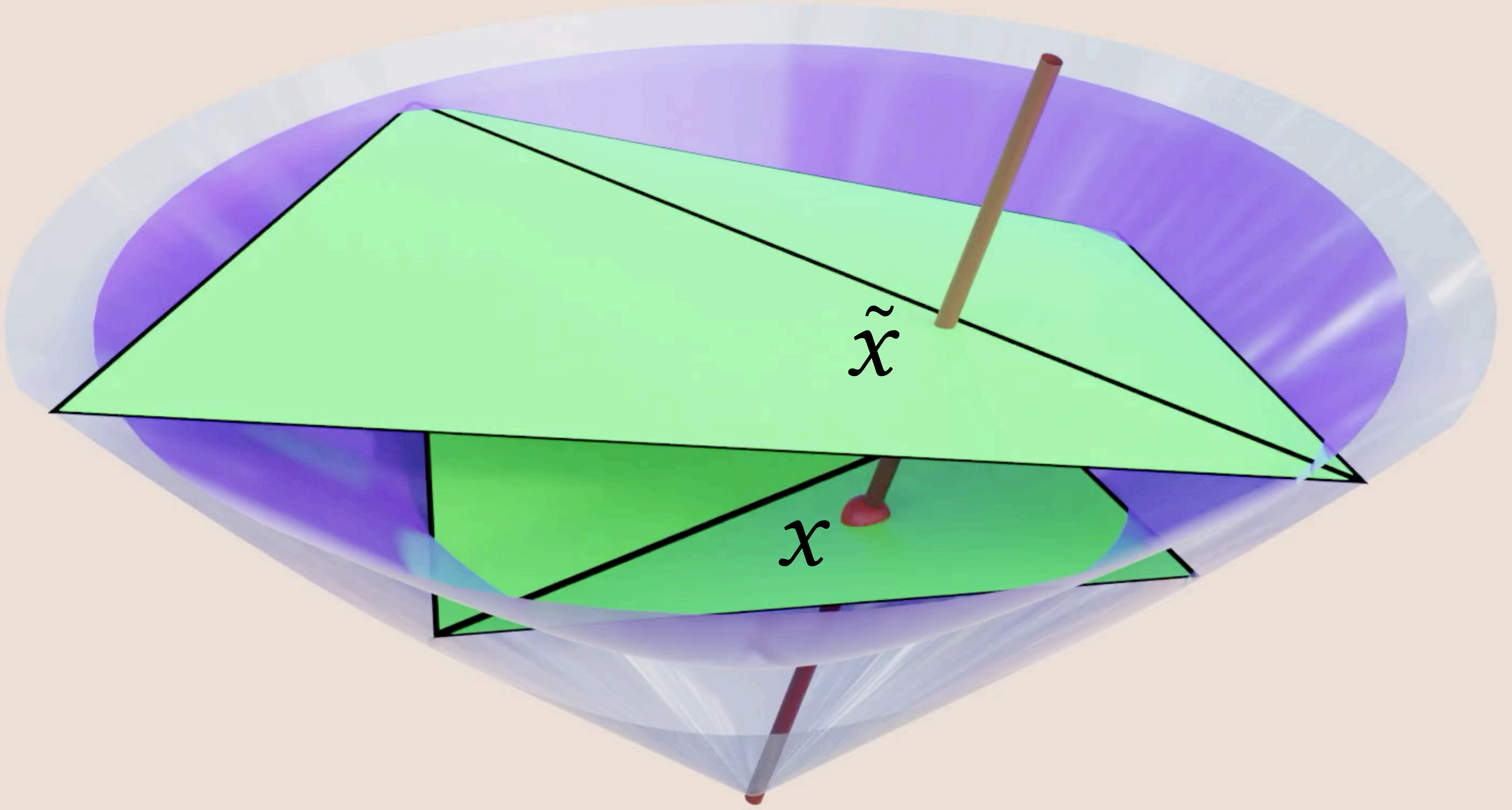


variable triangulation

Projective maps

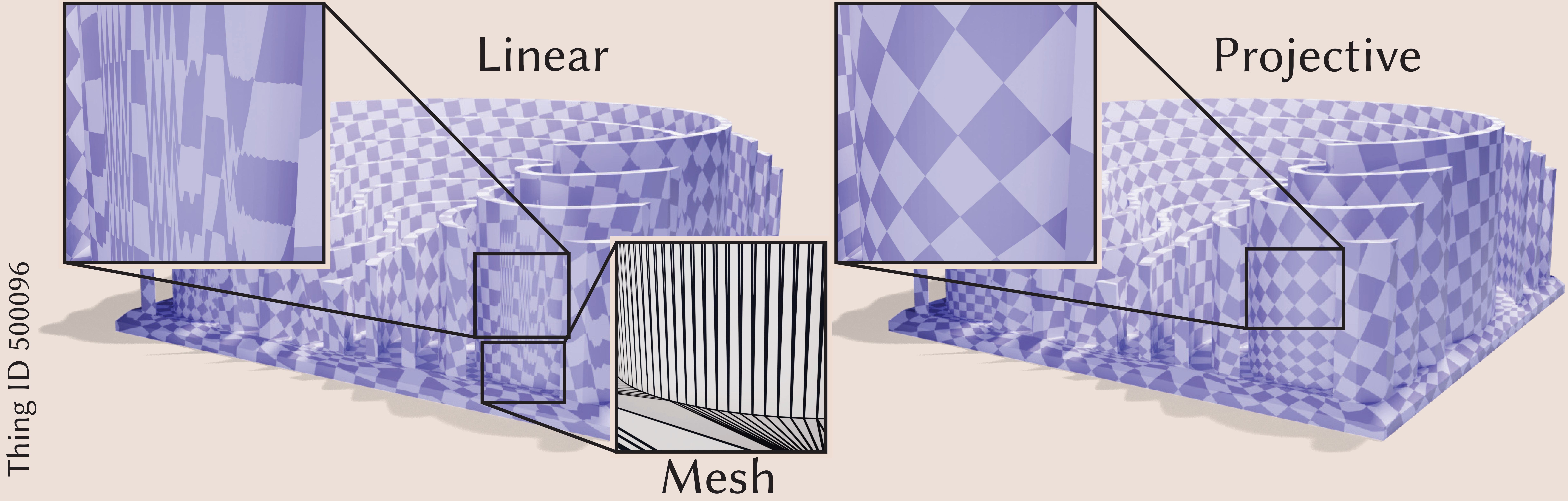


fixed triangulation



variable triangulation

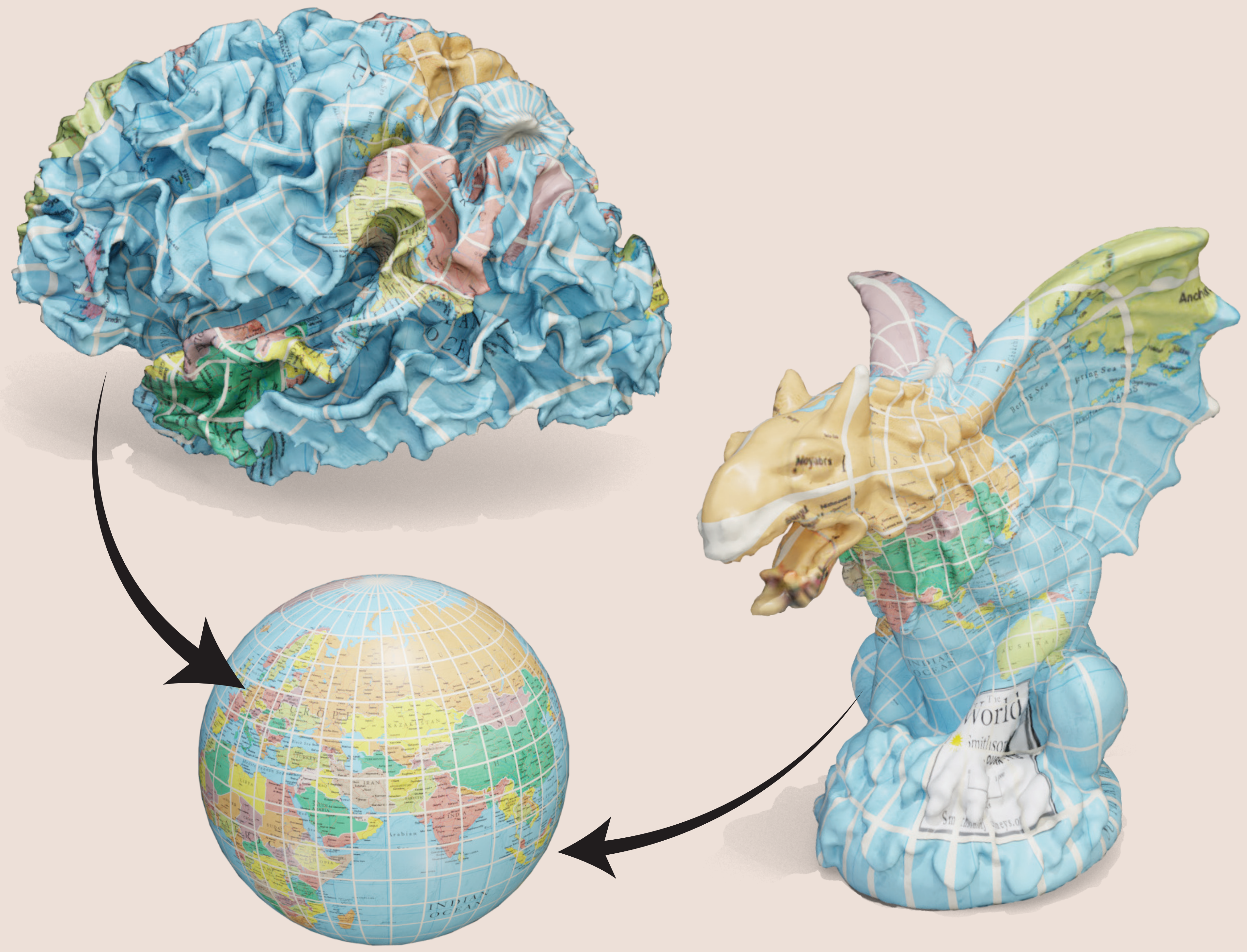
Projective interpolation improves quality

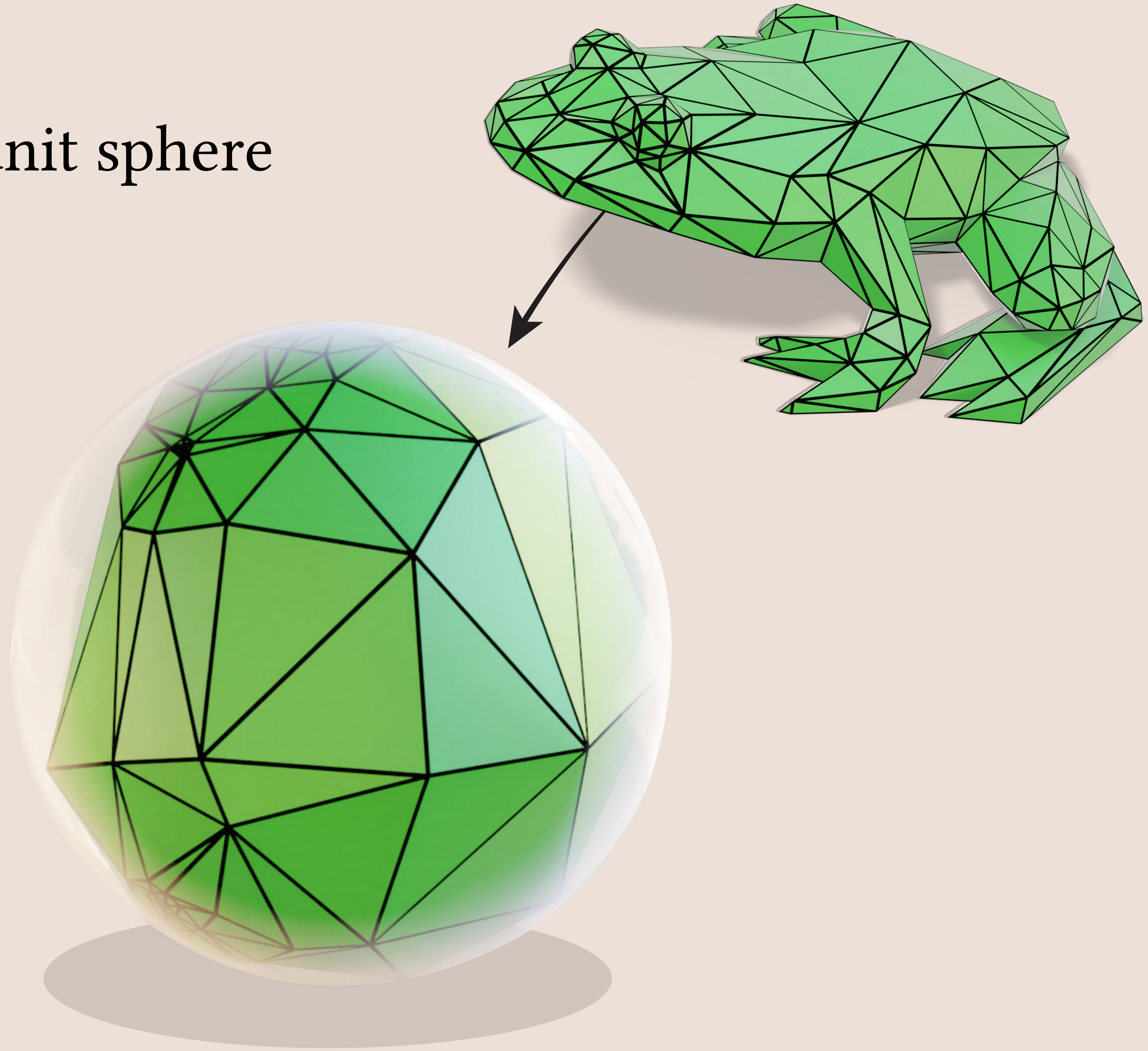


Discrete spherical uniformization

Discrete spherical uniformization

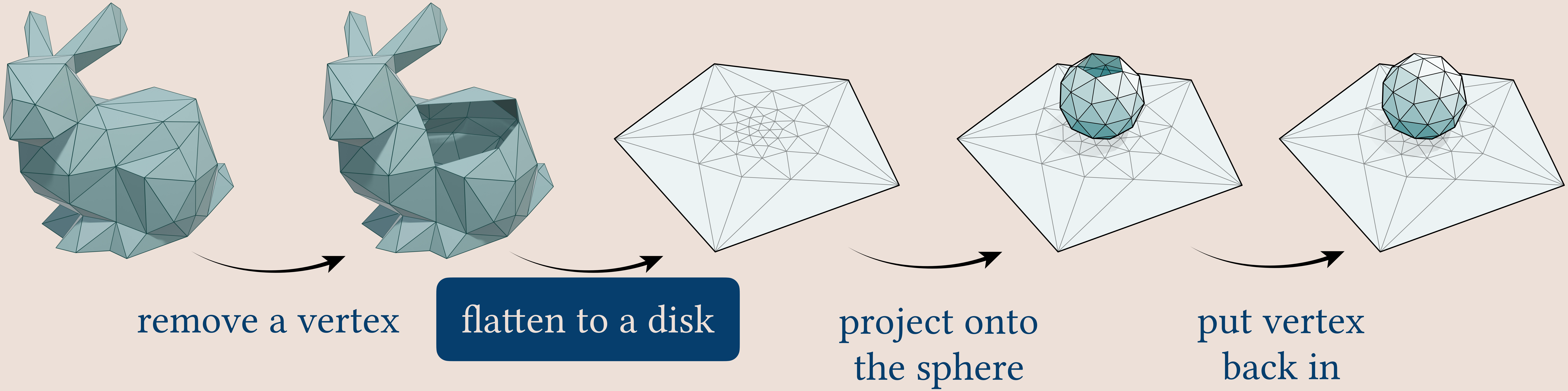
- So far: cone flattenings
- Now: map genus-0 surfaces to sphere
  - Explicitly, convex polyhedron w/ vertices on unit sphere





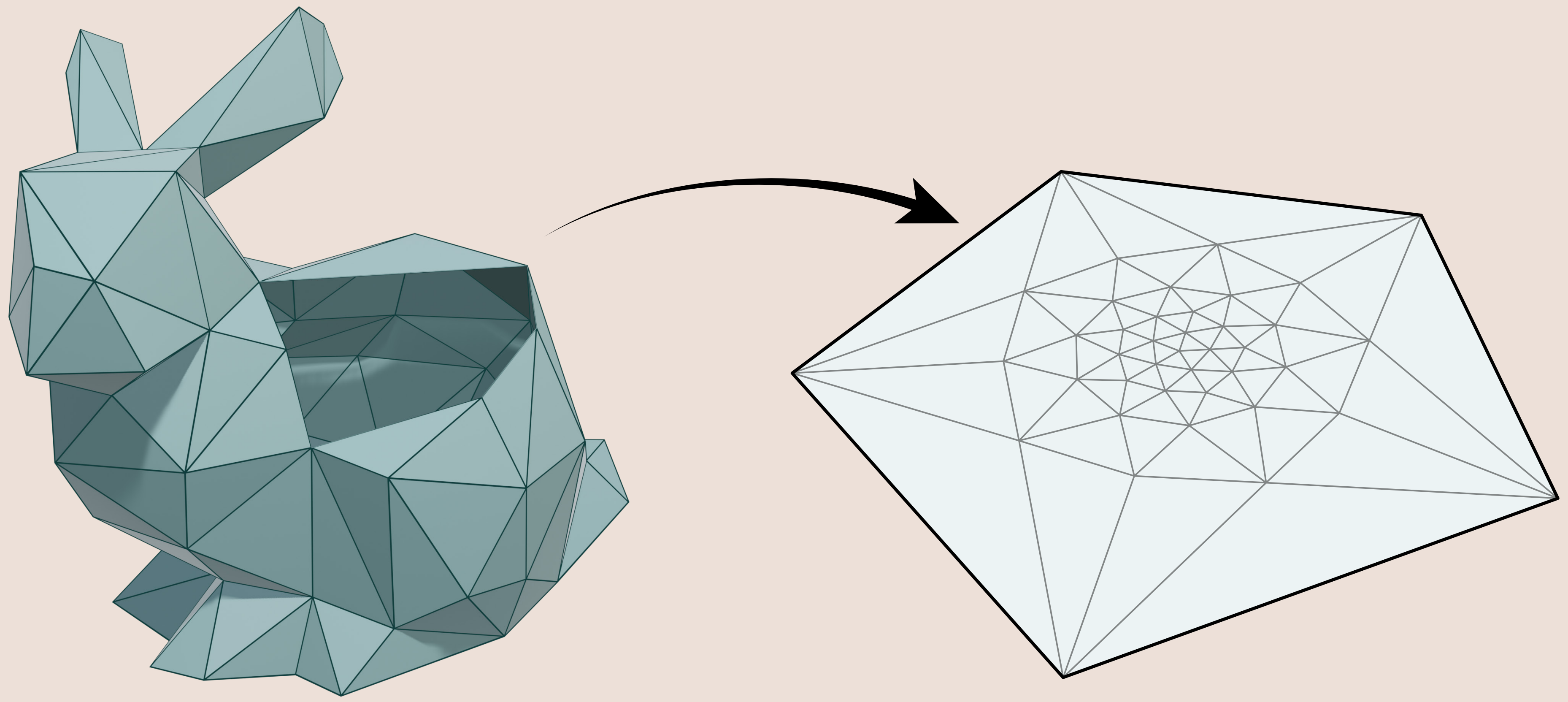
Discrete spherical uniformization

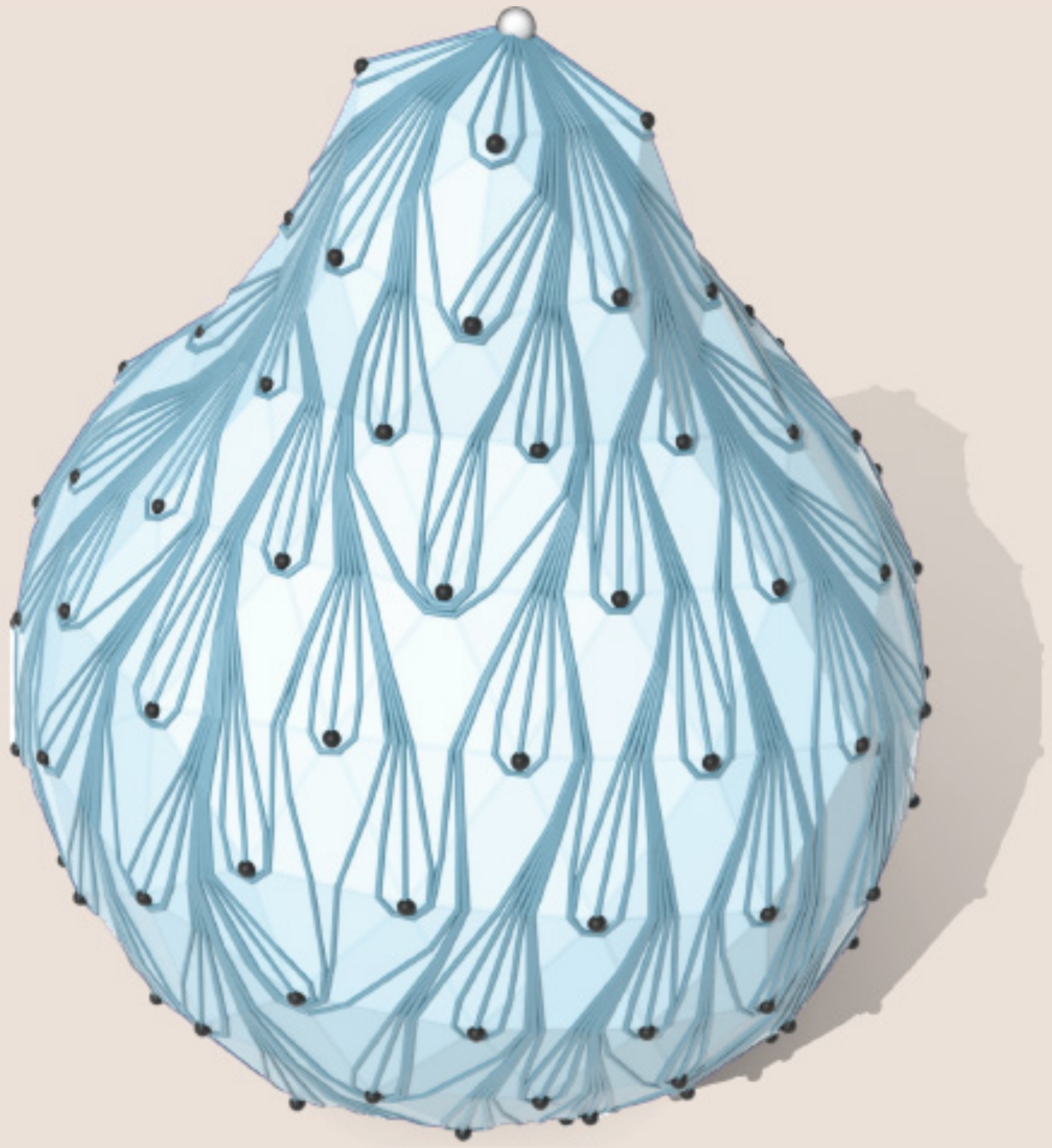
- Idea [Springborn 2019]:



Mapping to a polygon

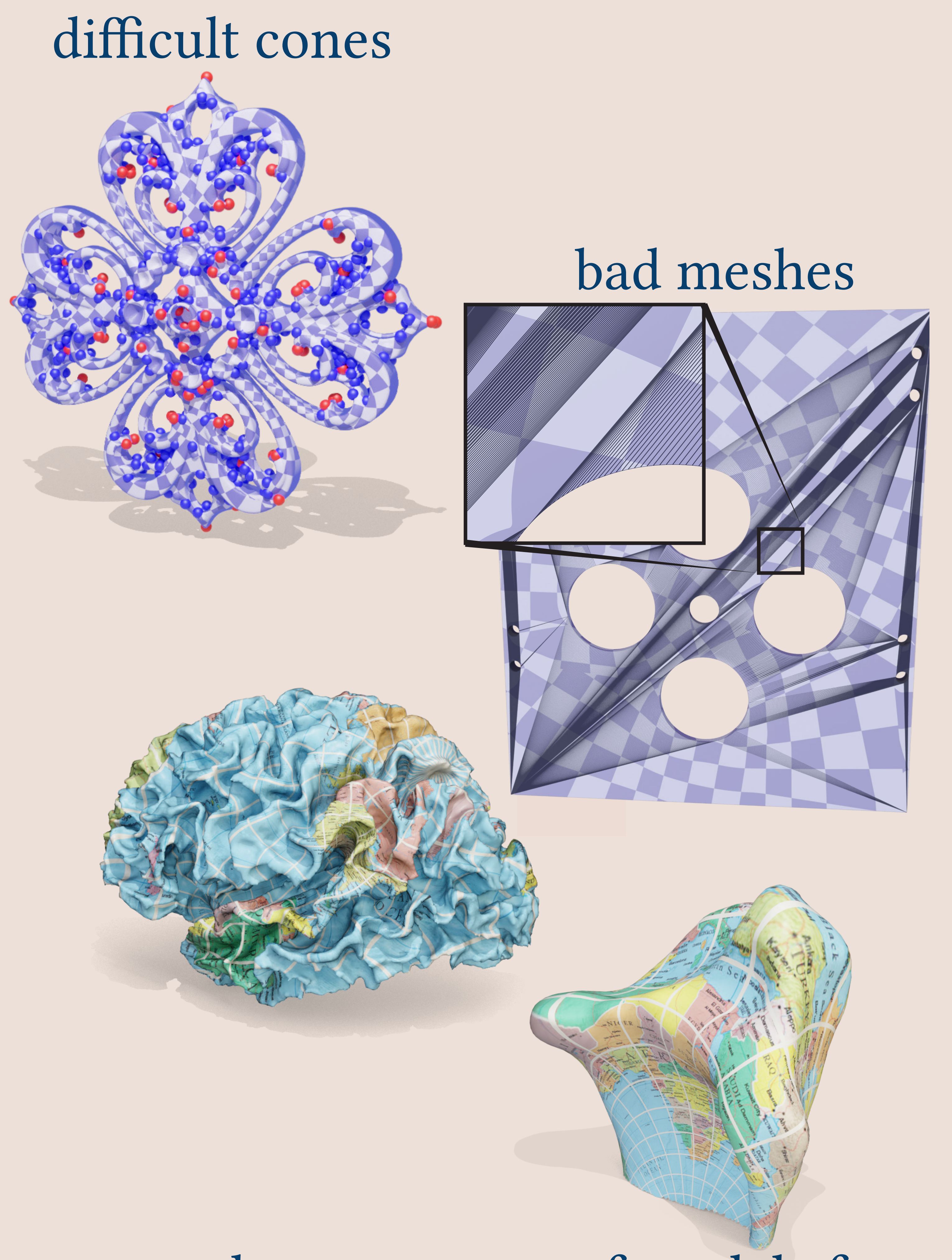
- Mapping to polygon requires a similar optimization problem
- Problem: what if you need to flip a boundary edge?
- Hyperbolic perspective saves us again
  - Previous vertex scaling methods couldn't guarantee success
  - Fun fact: compute (hyperbolic) geodesic distance via Delaunay flipping [Springborn 2019]





Results

Challenging datasets

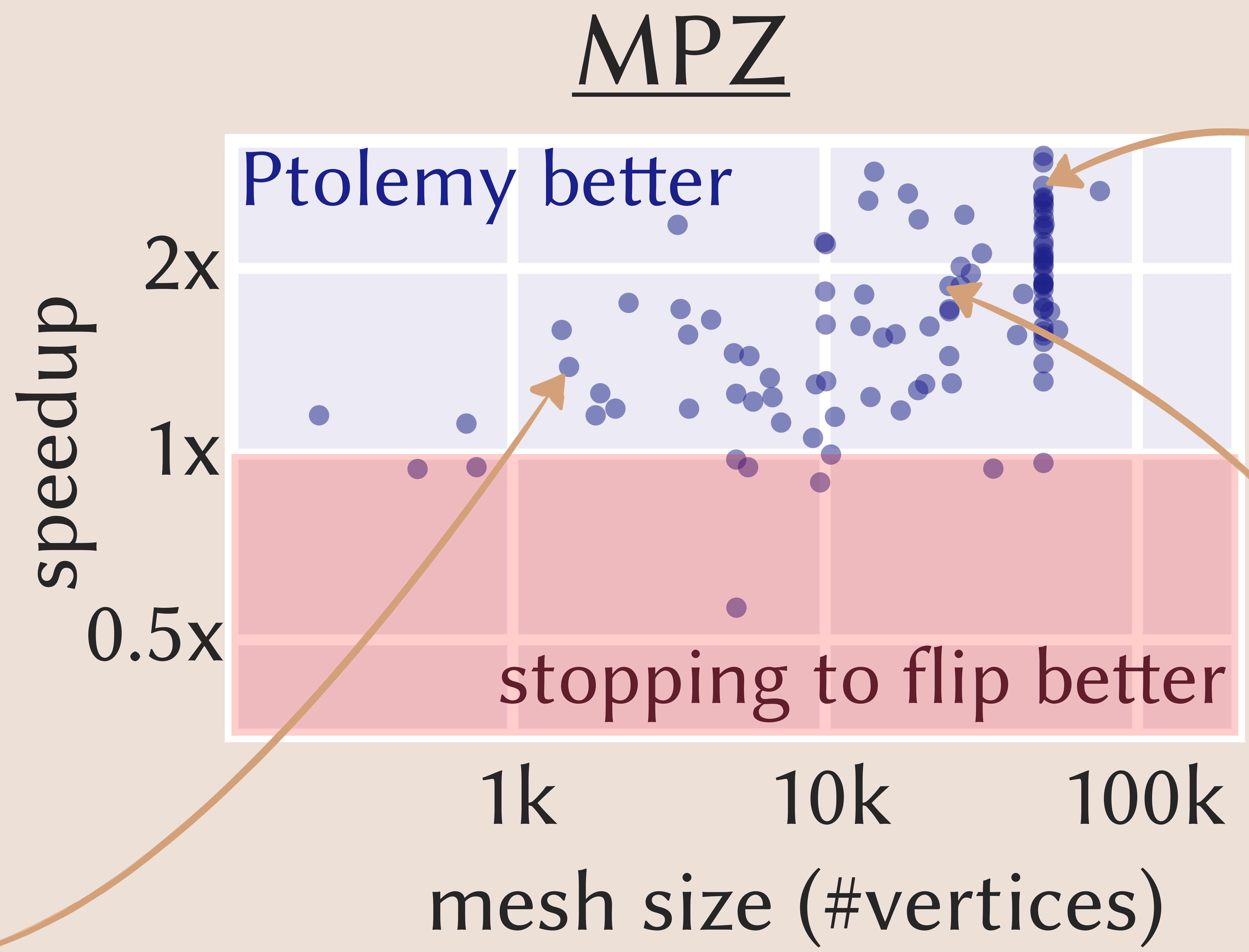


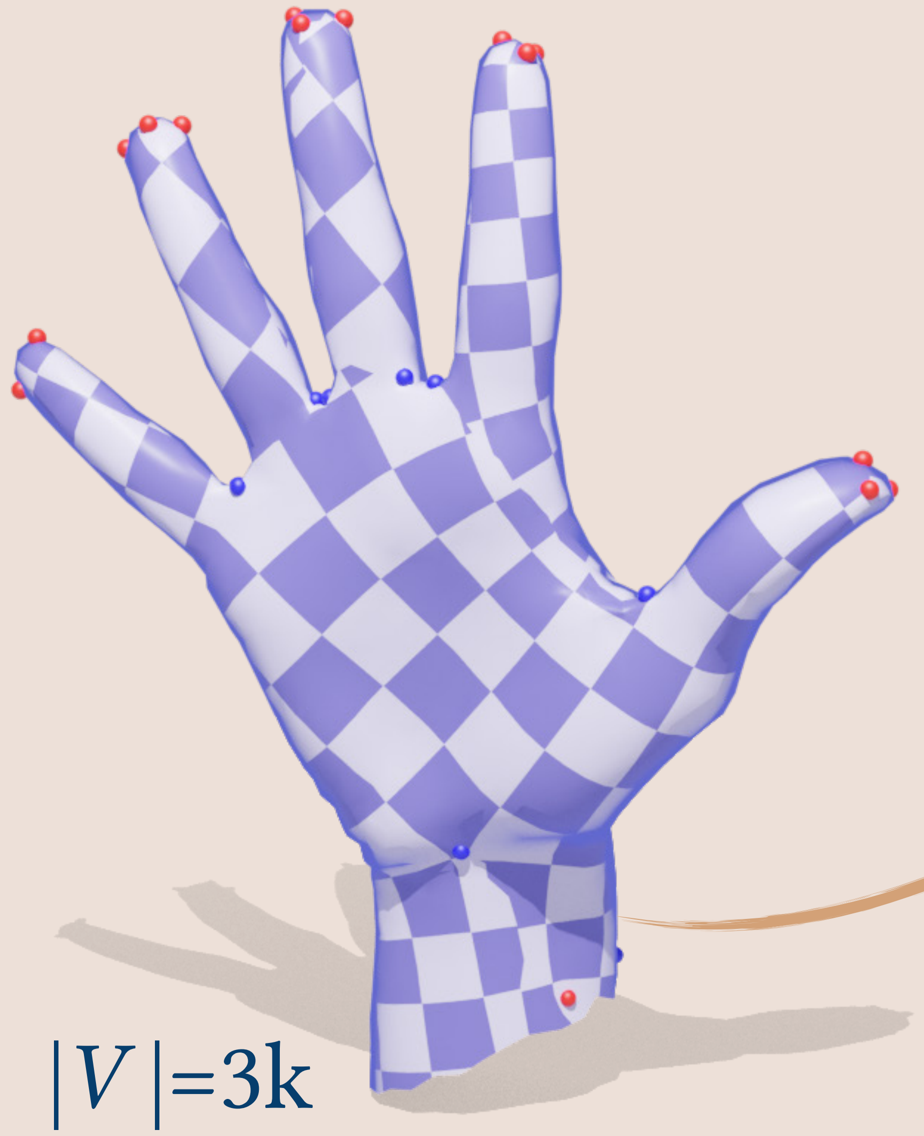
Dataset	# Models	Success rate	Average time
MPZ [Myles+ 2014]	114	100%	8s
Thingi10k [Zhou+ 2016]	32,744*	97.7%	57s†
brain scans [Yeo+ 2009]	78	100%	493s
anatomical surfaces [Boyer+ 2011]	187	100%	15s

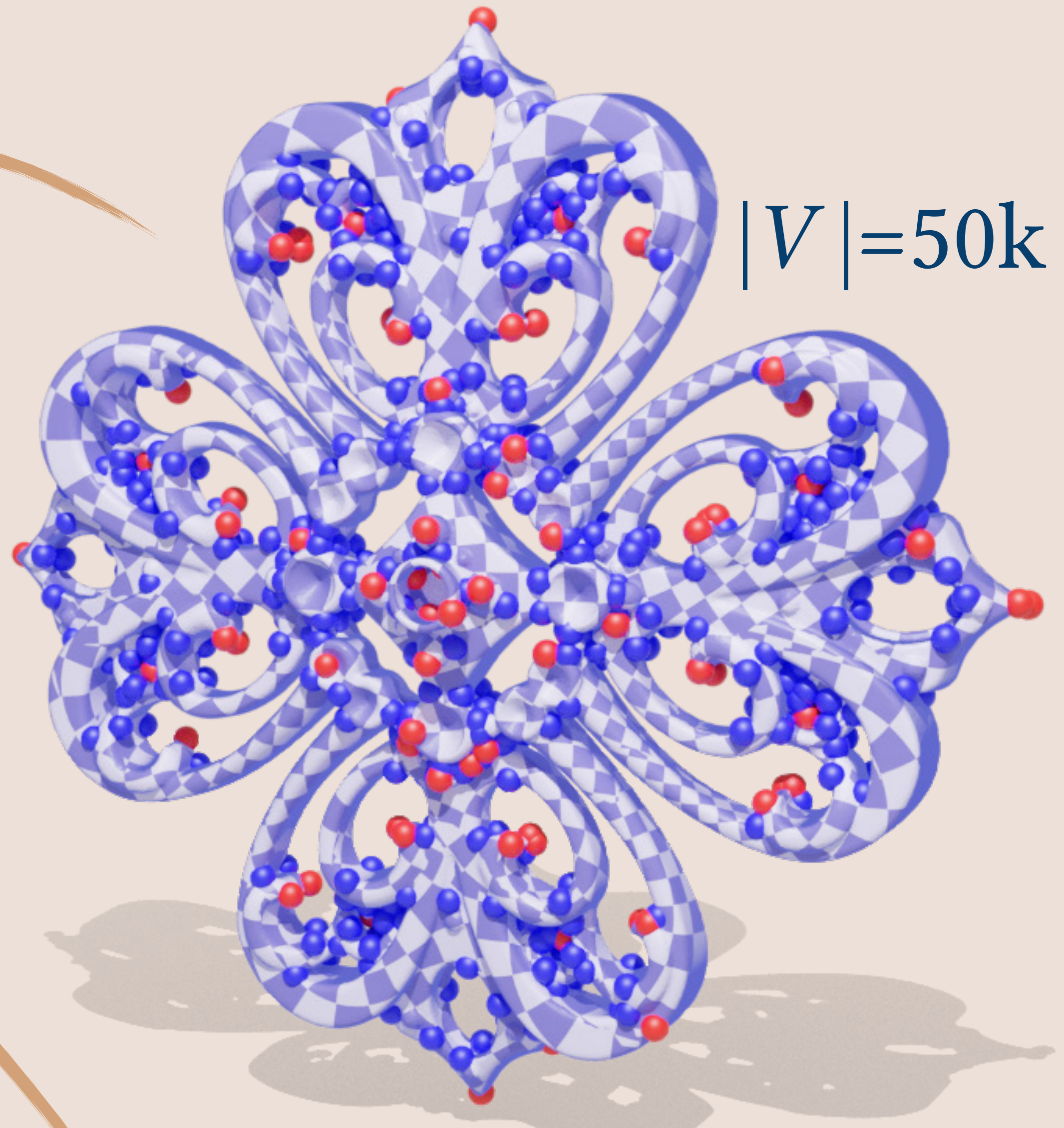
* connected components of models from Thingi10k

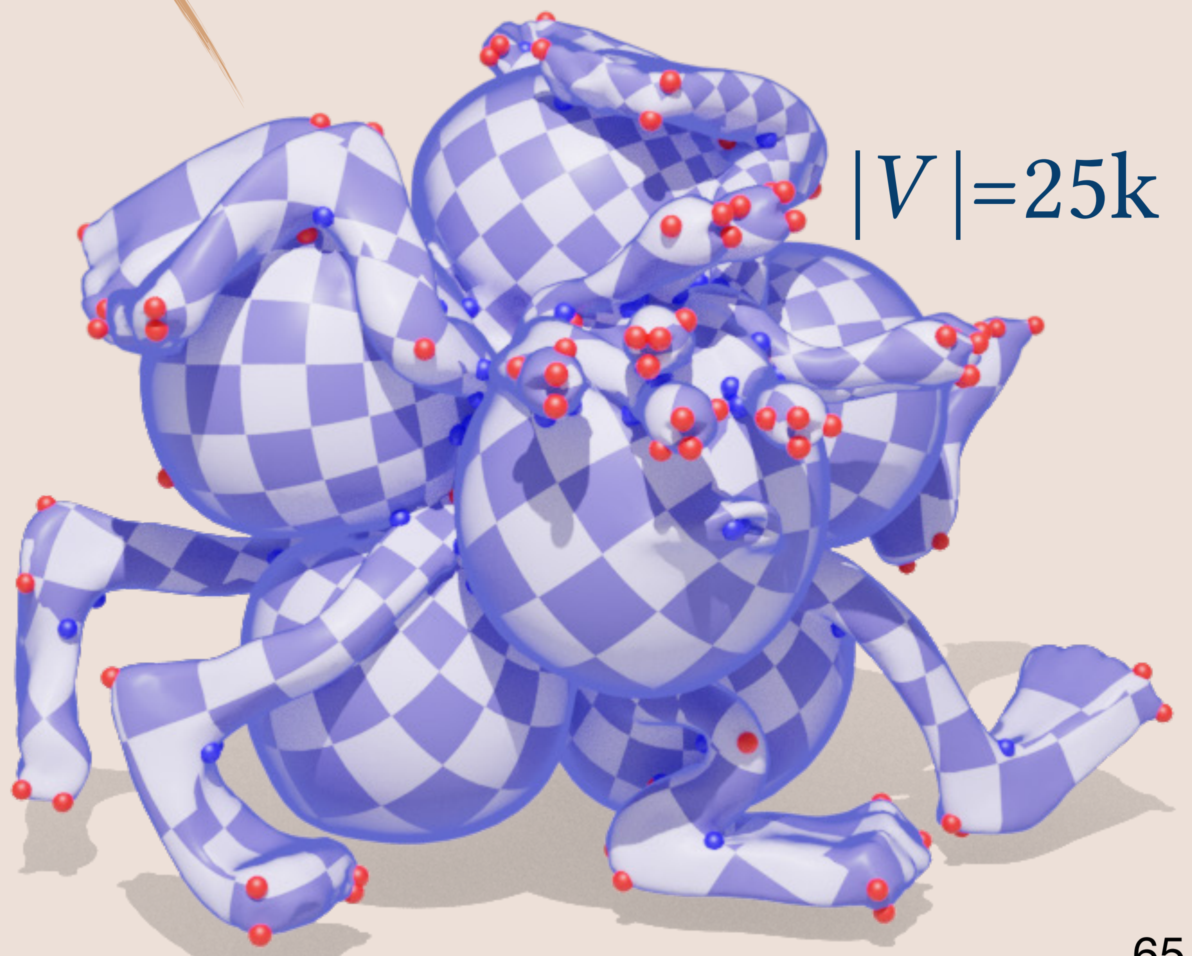
† average time on models with > 1000 vertices

Ptolemy flips improve performance

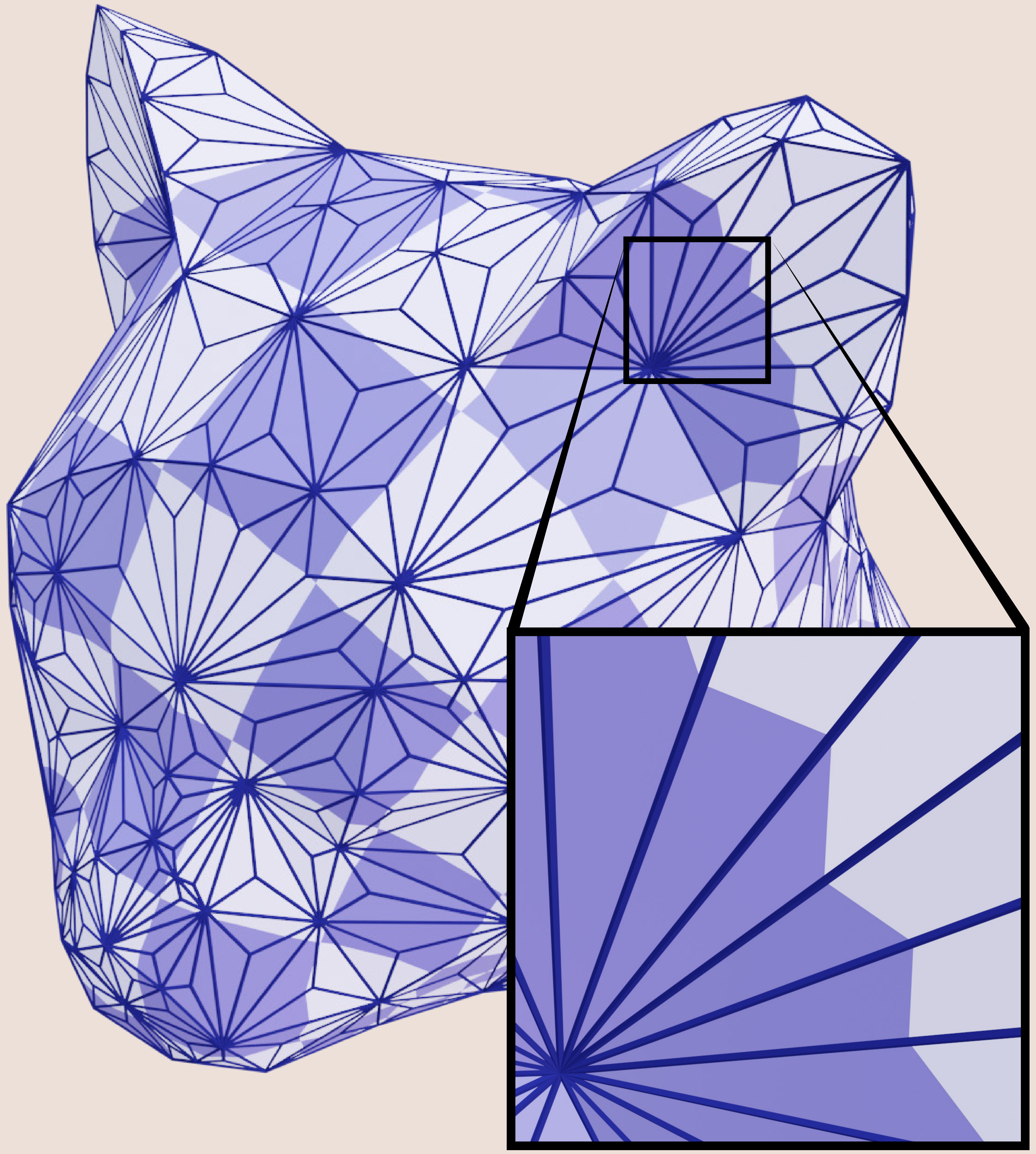




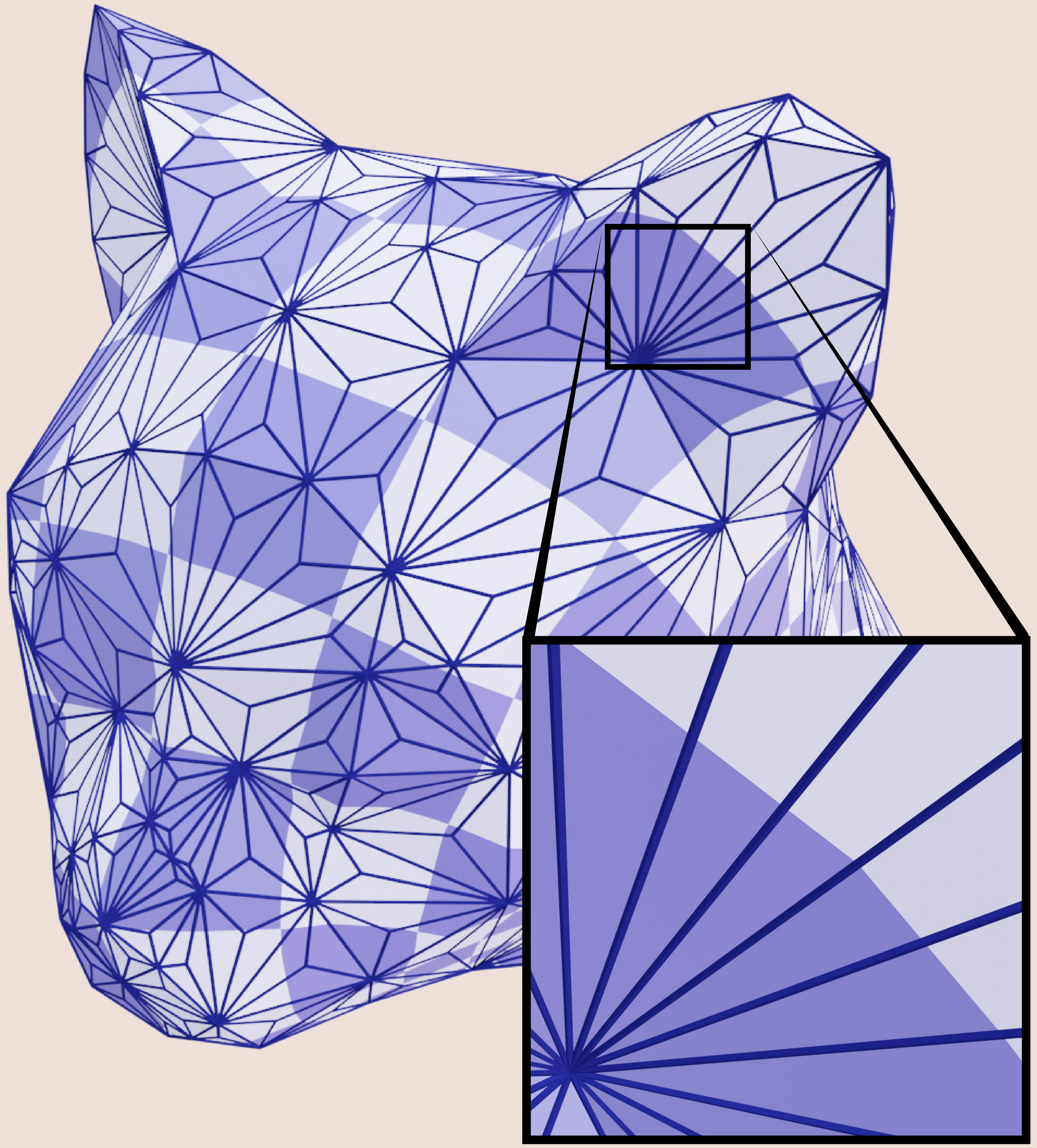




Variable triangulation > fixed triangulation



Fixed triangulation (CETM)



Variable triangulation (CEPS)

Even when fixed triangulation succeeds, variable triangulation is better

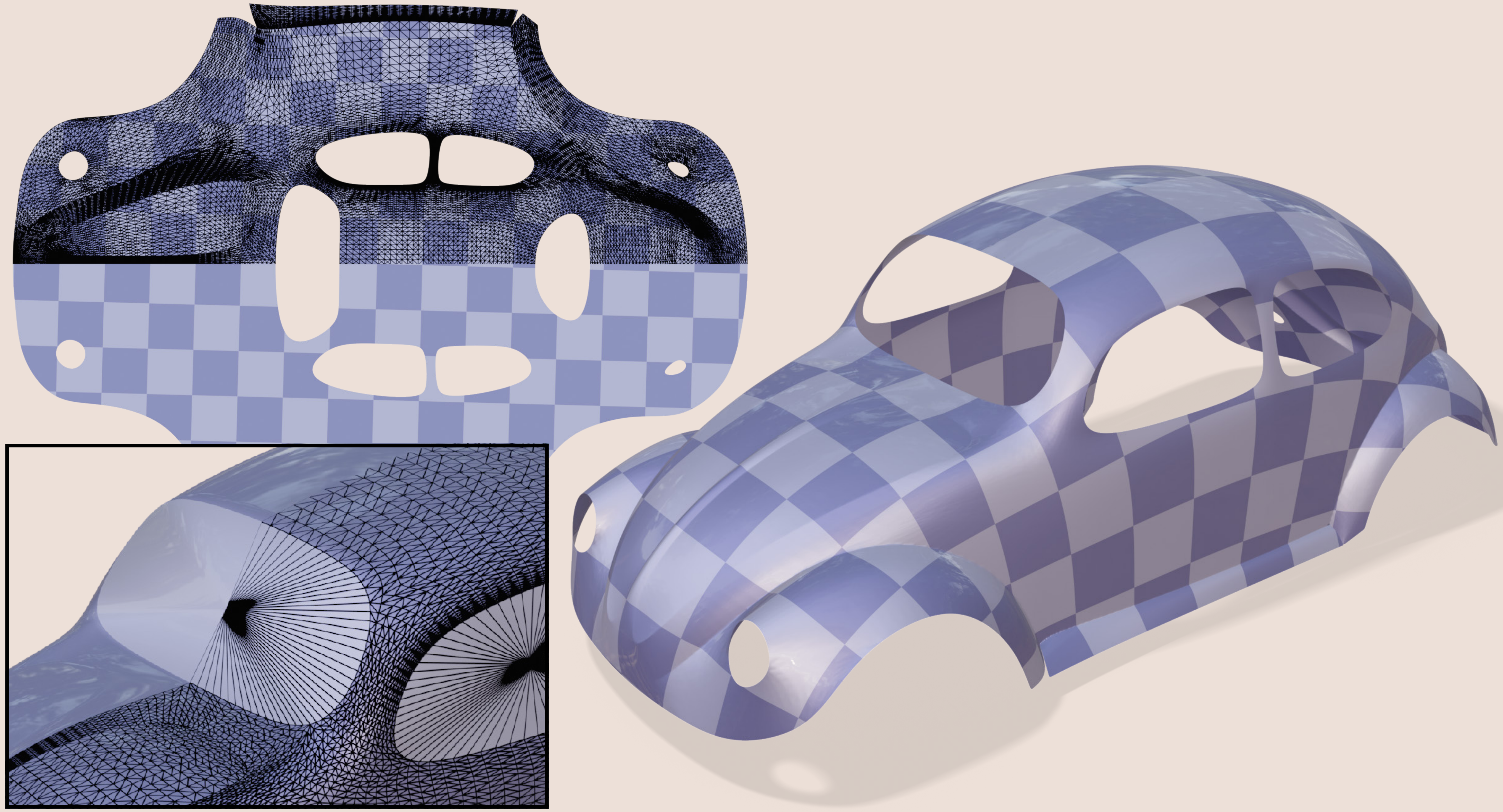
Boundary conditions

- Prescribe boundary curvature or scale factor
- Key idea: eliminate boundary by doubling surface

Multiply-connected domains

No hole filling

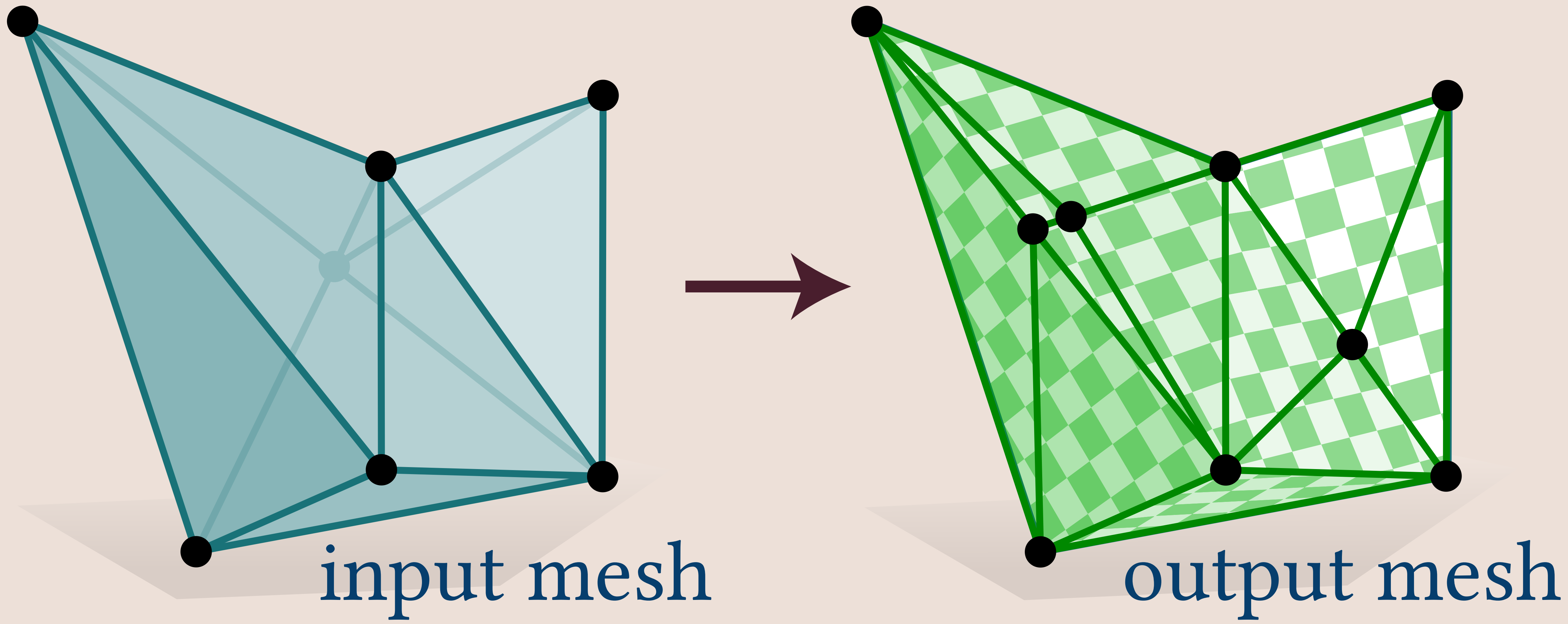
[Hefetz+2019]



Hole filling

Limitations and future work

- Output is refined mesh
  - Could you unflip all flipped edges?
- If all you care about is injectivity, correspondence is simpler
- Going beyond 2D
  - 2D uniformization theorem $\longrightarrow$ 3D geometrization theorem
  - 2D Delaunay triangulations $\longrightarrow$ 3D Delaunay tetrahedralizations



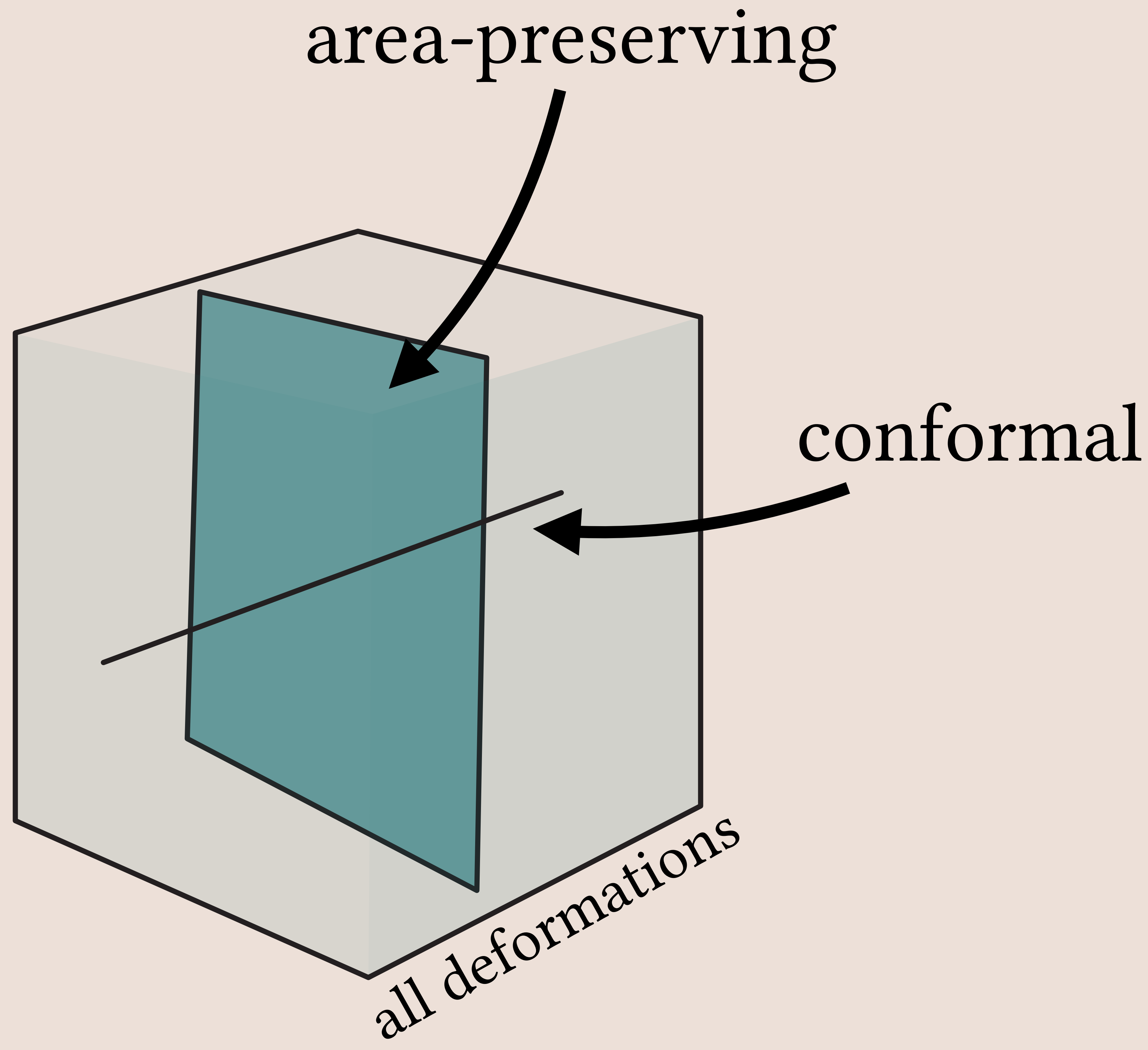
Work in Progress: Discrete Area Equivalence

- Conformal deformations are a subspace of all deformations
  - What is the complementary subspace?
- E.g. consider linear maps in the plane

$$\begin{pmatrix} s & 0 \\ 0 & s \end{pmatrix} \quad \begin{pmatrix} a & -b \\ b & a \end{pmatrix} \quad \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \qquad\qquad \begin{pmatrix} c & d \\ d & -c \end{pmatrix}$$

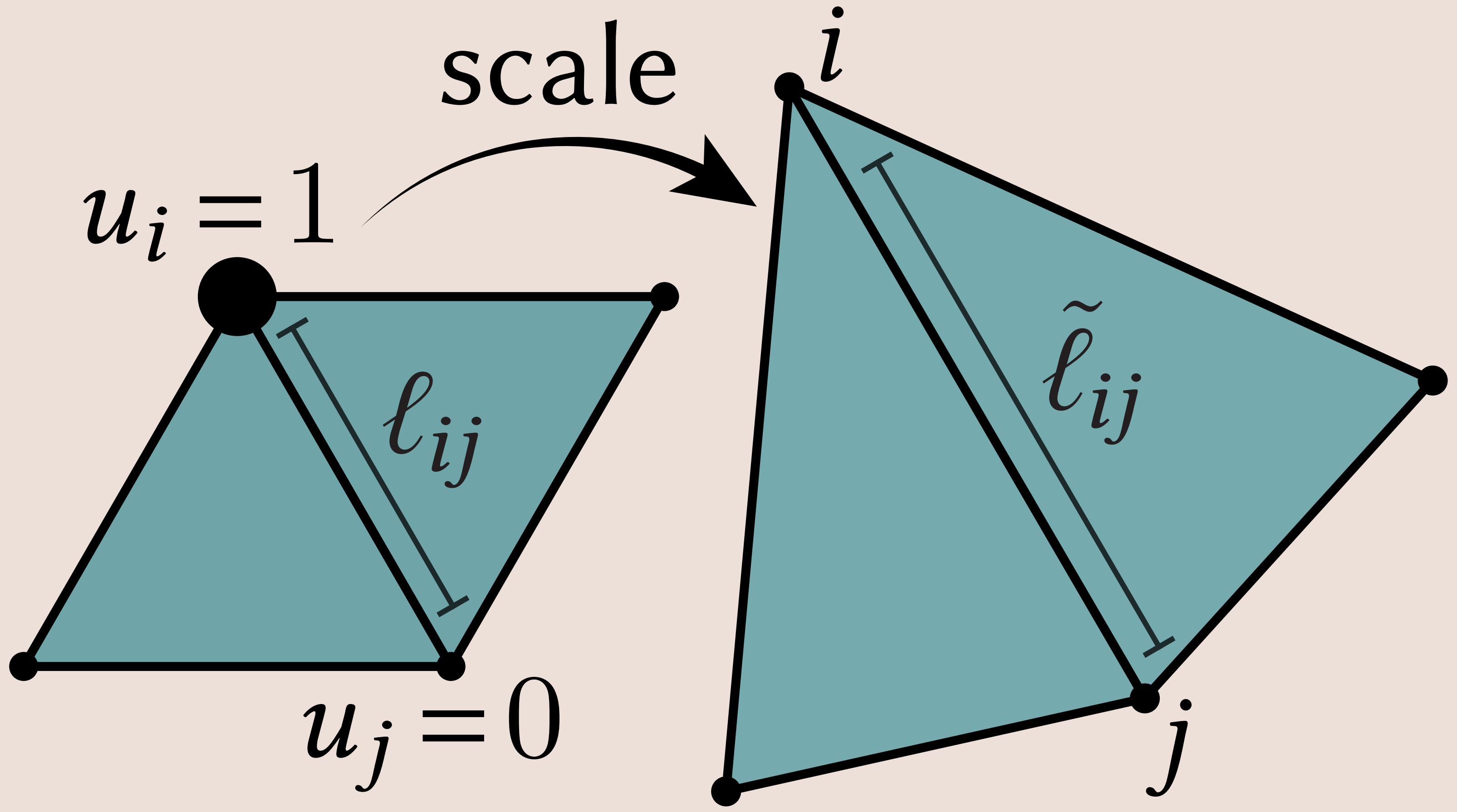
conformal

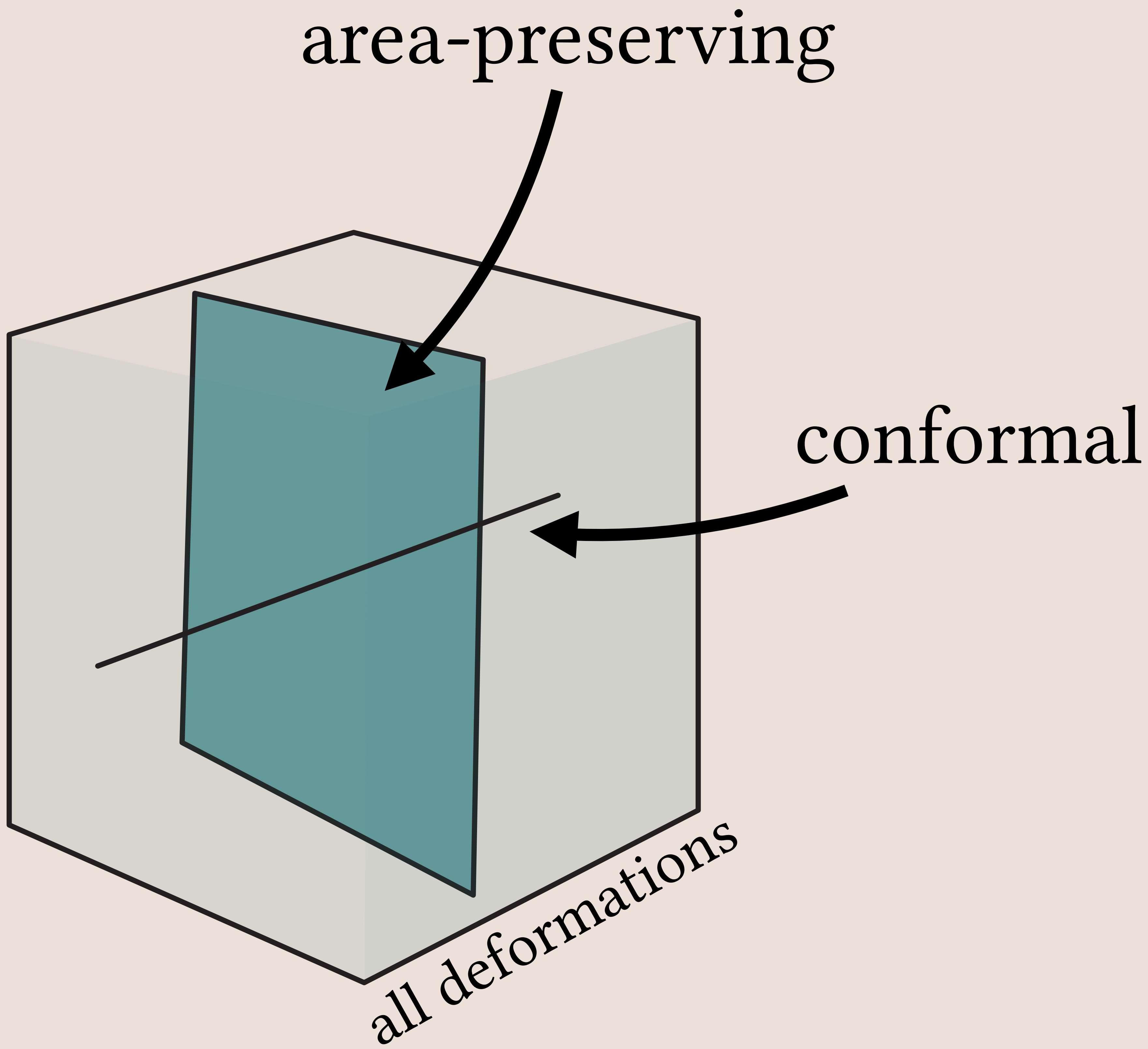
orthogonal to conformal
$\longleftrightarrow$ symmetric, trace-free
$\longleftrightarrow$ derivative of an area-preserving map



Work in Progress: Discrete Area Equivalence

- What is the complement of discrete conformal maps?
  - Discrete conformal maps scale edges equally around vertices



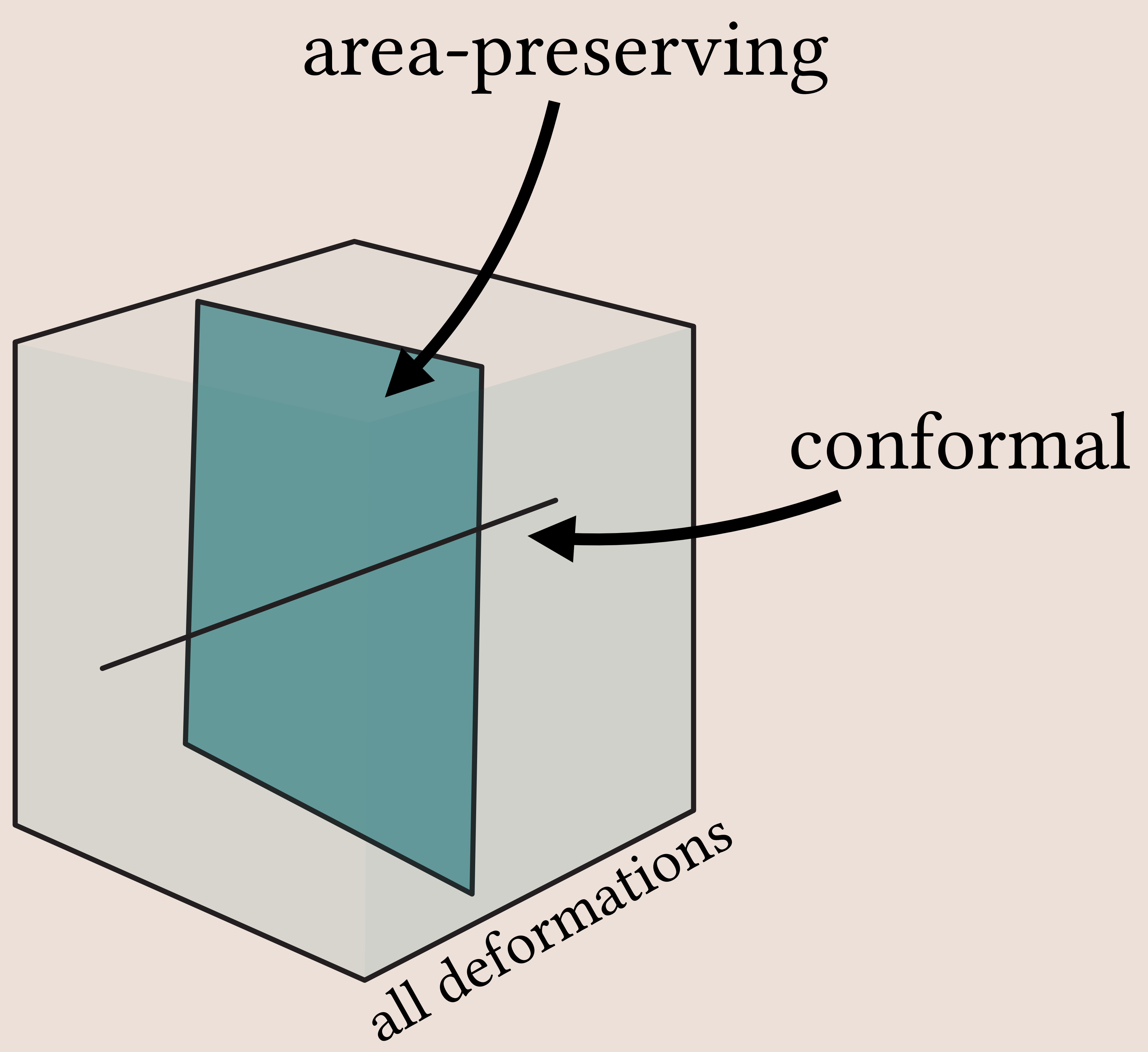


  - Complementary maps should preserve the average edge length around a vertex

$$\ell \rightsquigarrow \tilde{\ell} \qquad \left(\prod_{j=1}^{d} \ell_{ij}\right)^{1/d} = \left(\prod_{j=1}^{d} \tilde{\ell}_{ij}\right)^{1/d}$$

Work in Progress: Discrete Area Equivalence $\prod_j \ell_{ij} = \prod_j \tilde{\ell}_{ij}$

- Doesn't seem to work out so well in practice
  - Discretizes *infinitesimal* area-preserving maps
  - Sees a lot of "distortion" in finite area-preserving maps
- Also some open theoretical questions:
  - What do you do if the triangulation changes?



Thanks!

Code is available at github.com/MarkGillespie/CEPS

Carnegie Mellon University

berlin

THE GEOMETRY COLLECTIVE AT CARNEGIE MELLON UNIVERSITY

This work was supported by a Packard Fellowship, NSF Award 1717320, DFG TRR 109, an NSF Graduate Research Fellowship, and gifts from Autodesk, Adobe, and Facebook.