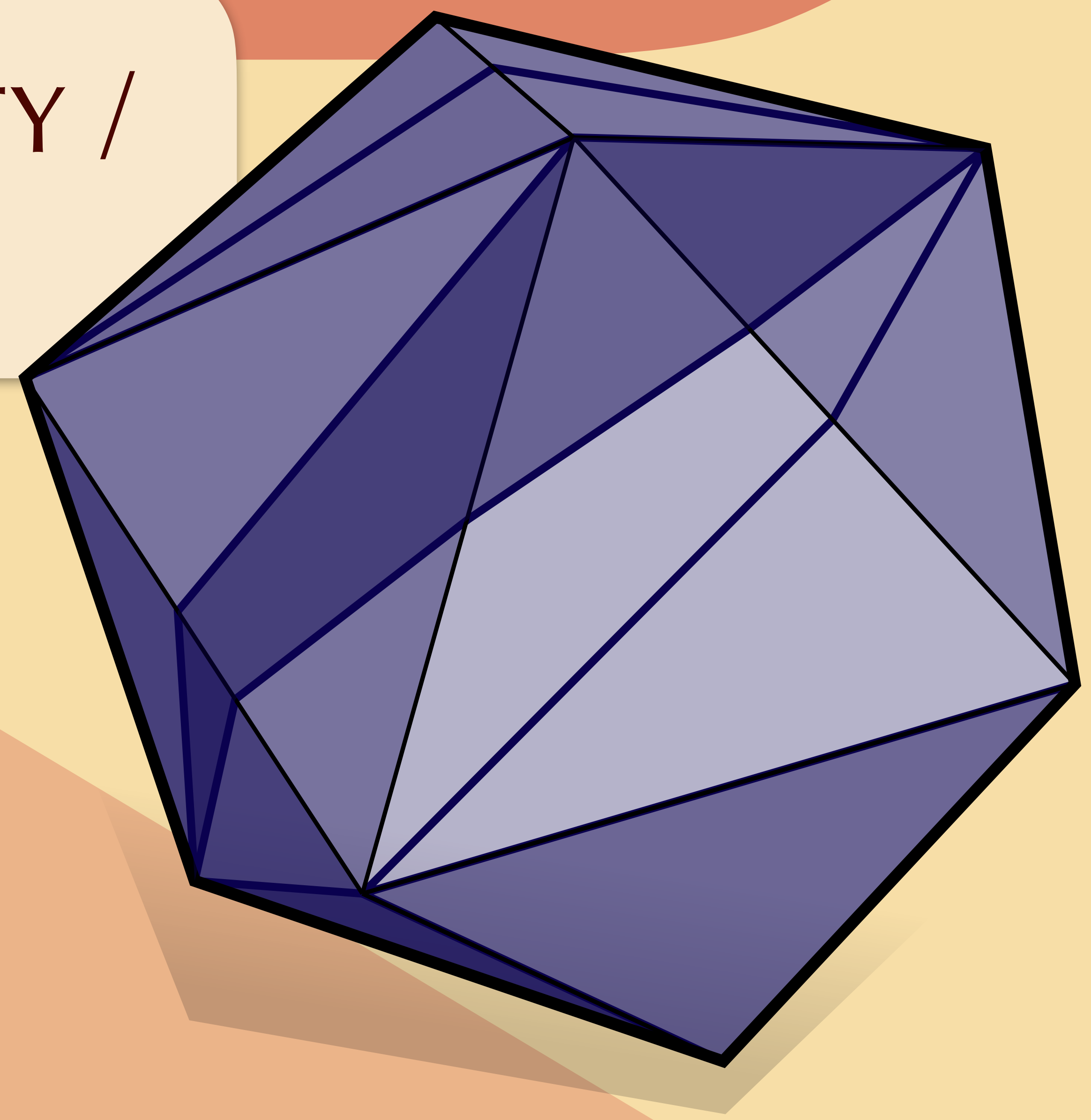
New Foundations for Robust Geometry Processing

Mark Gillespie, Carnegie Mellon University / École Polytechnique

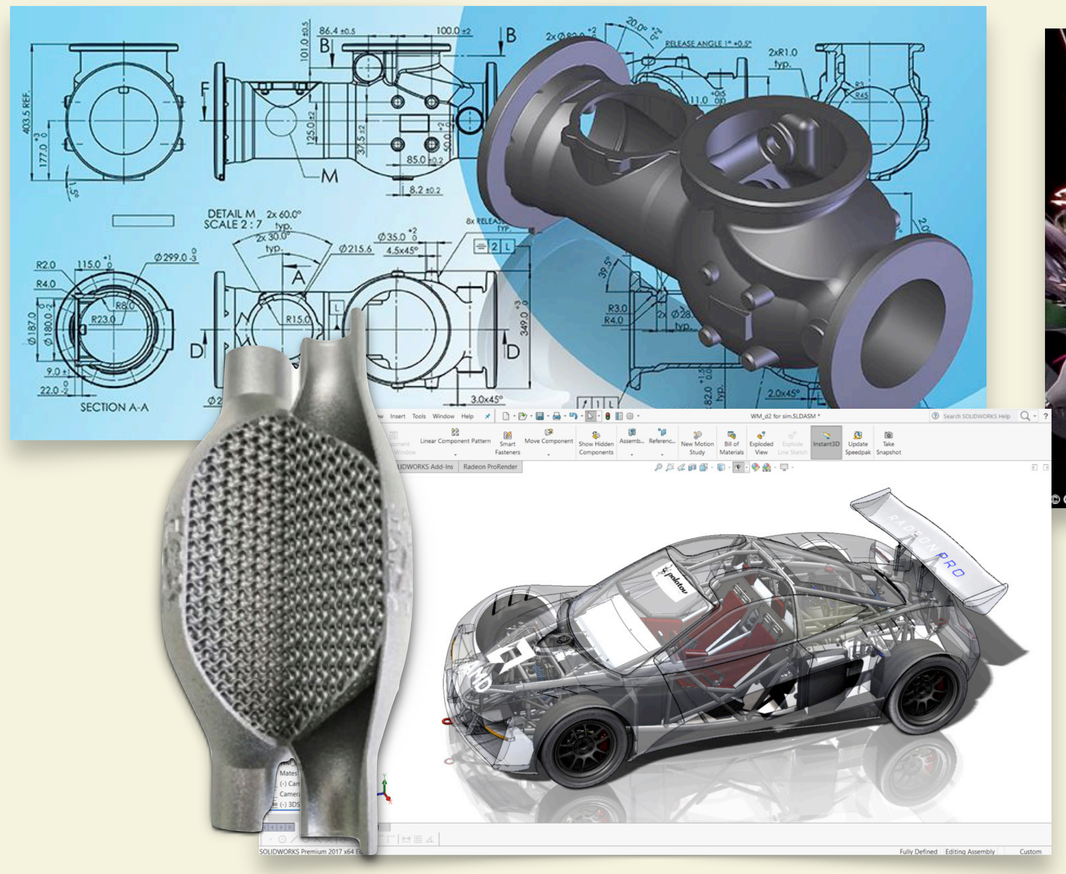


University of Utah

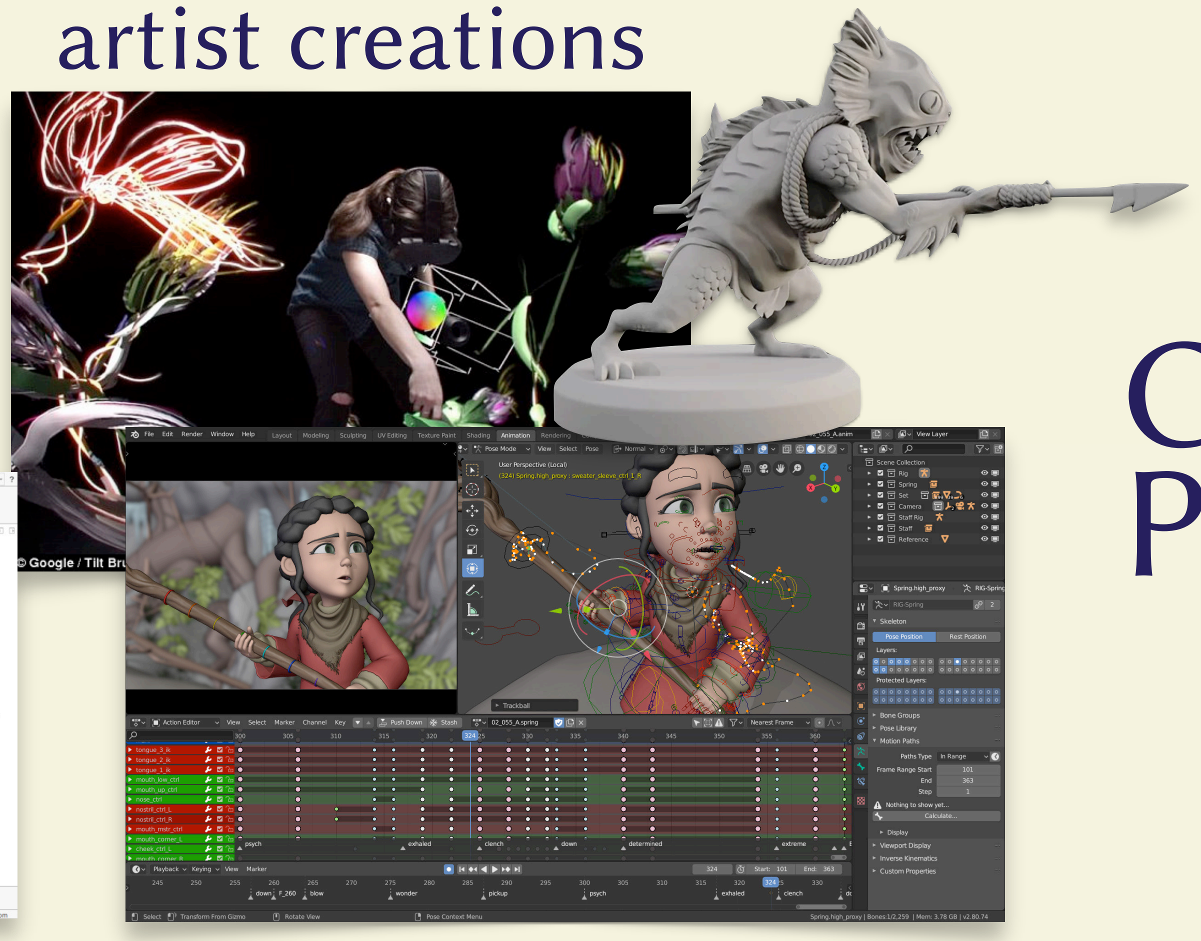
Friday, 28 Feb 2024

Geometry is all around us

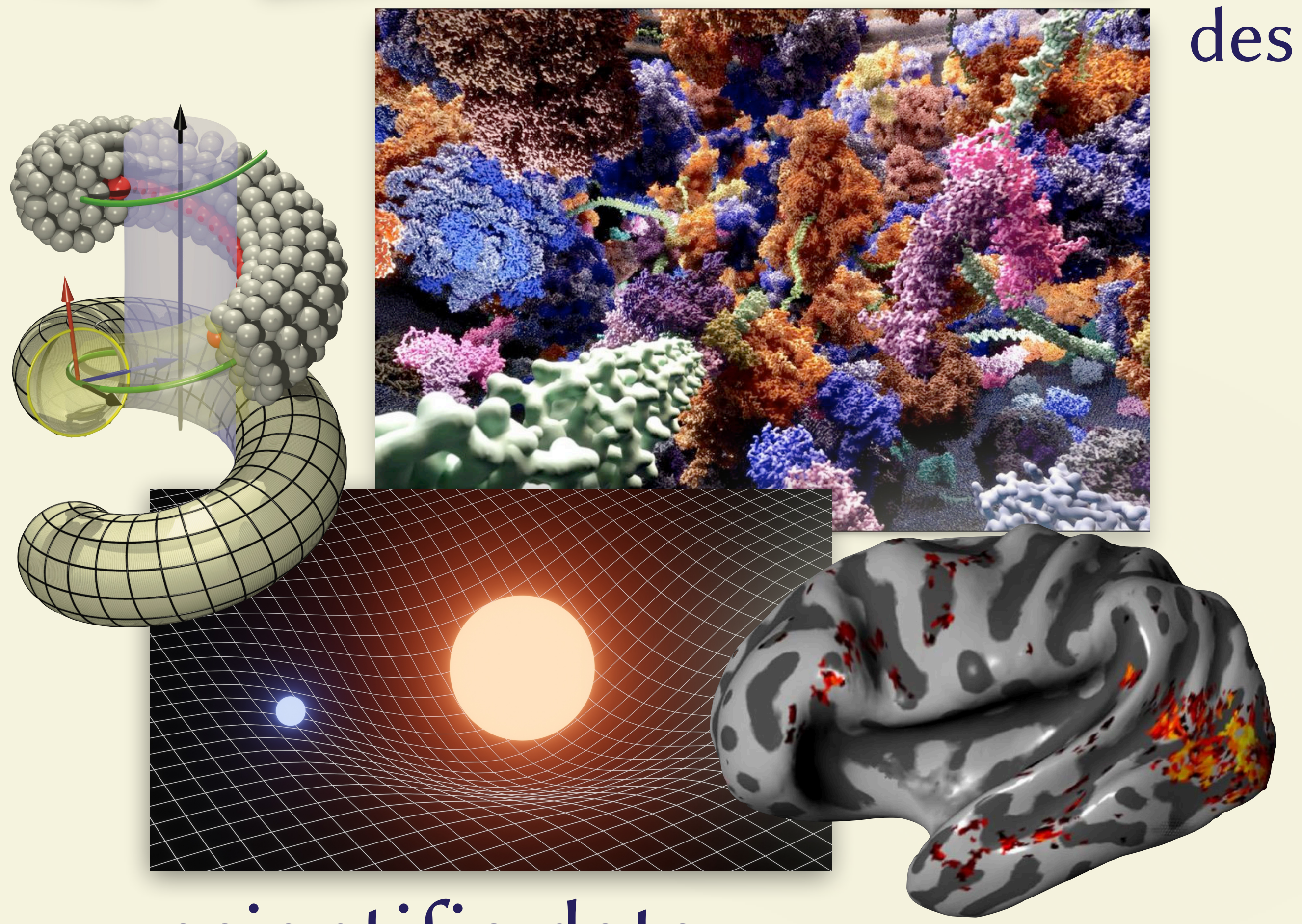
technical assets



artist creations



3D scans

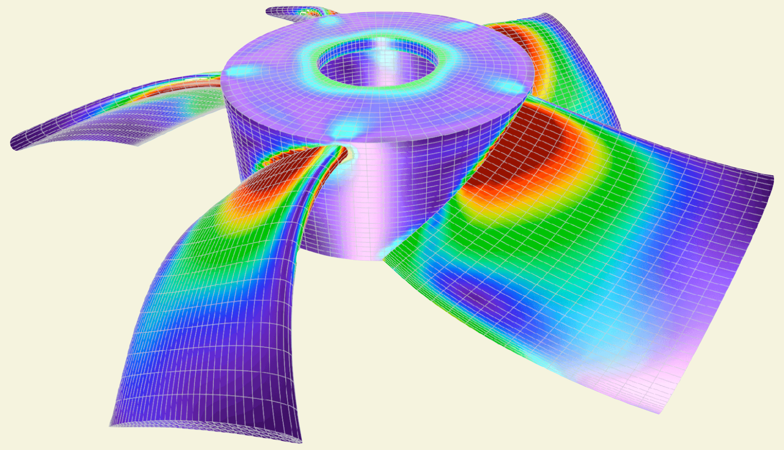


scientific data

Geometry Processing

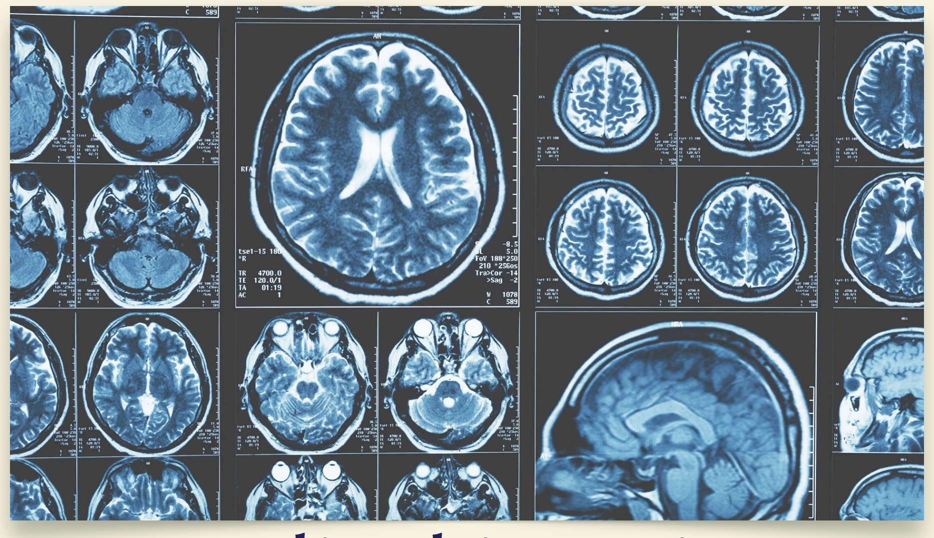
design algorithms to be:

- simple
- efficient
- robust



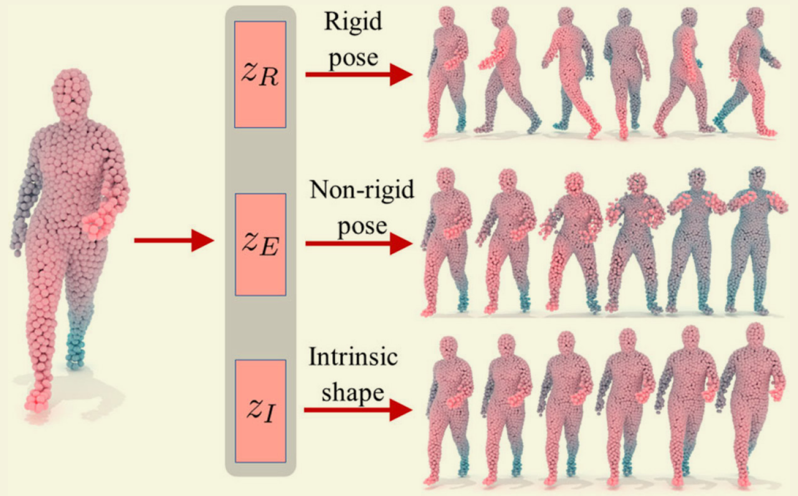
simulation

animation



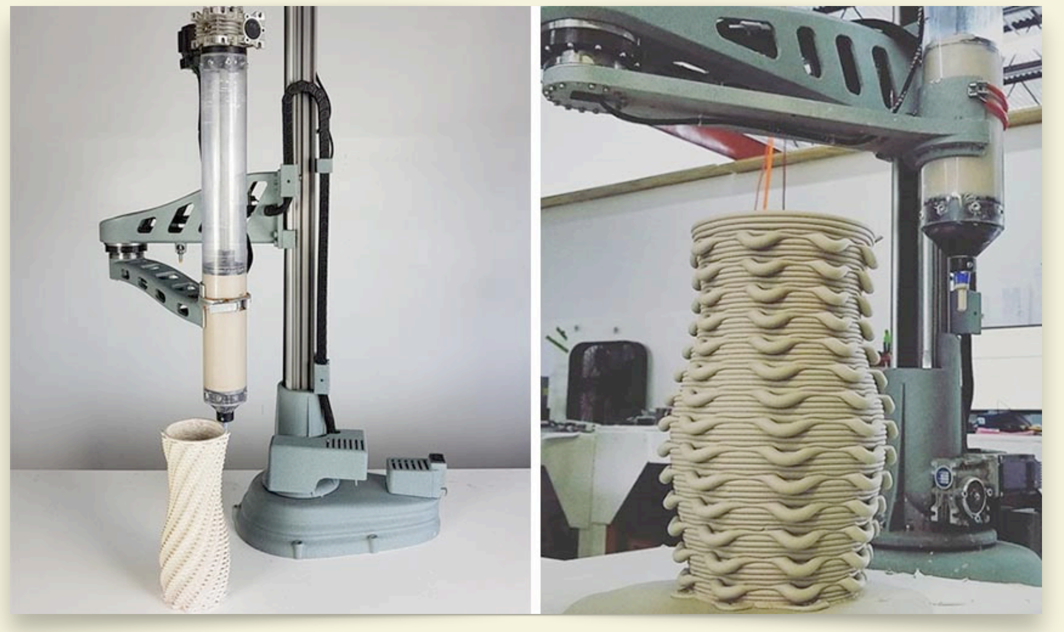
medical imaging

robotics



geometric ML

creative tools

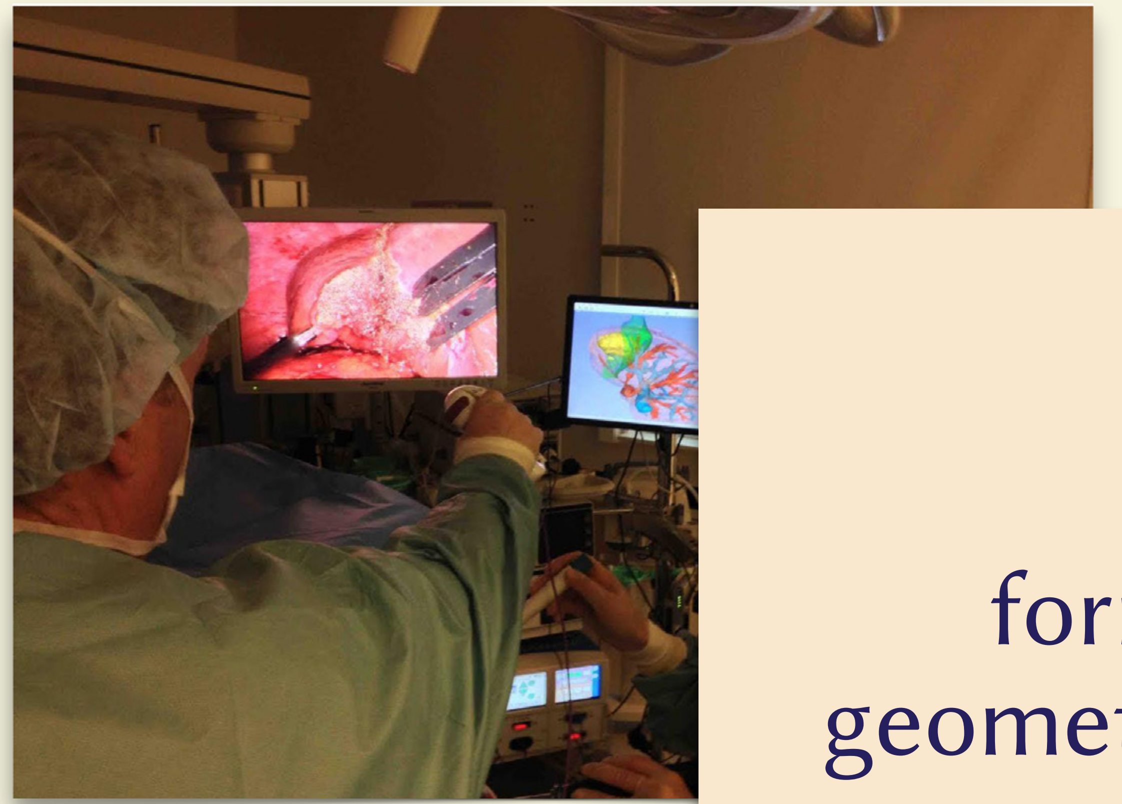


digital fabrication

Example: Poisson reconstruction

[Kazhdan *et al.* 2006]

3D scan ⟶ 3D model



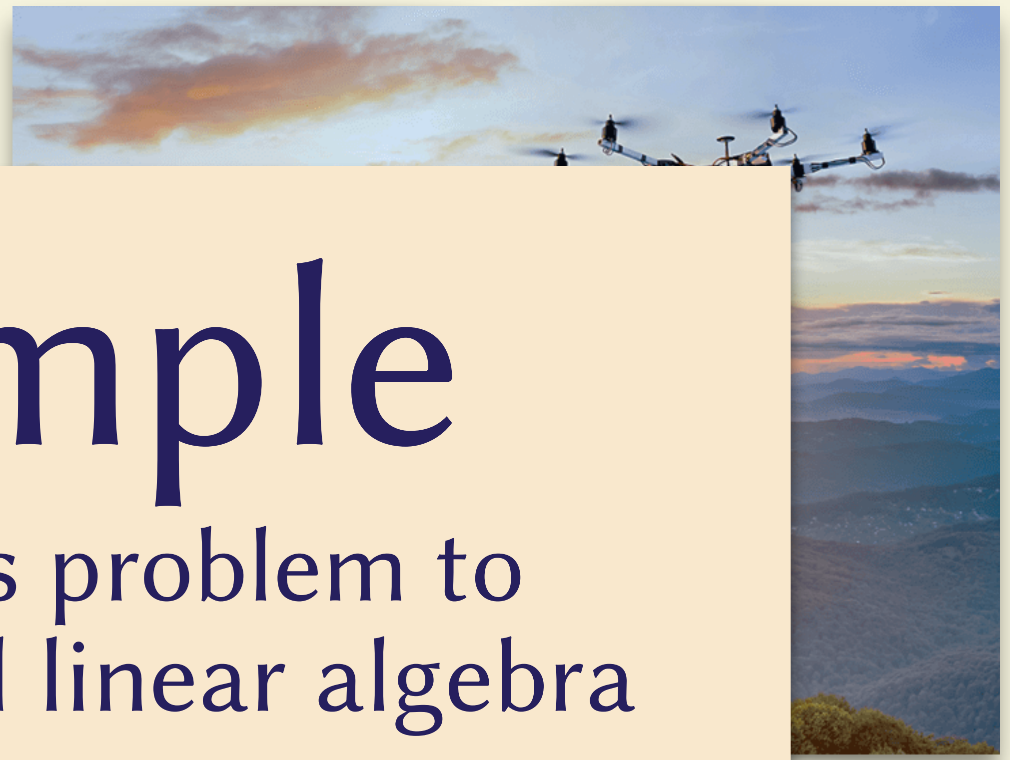
medical imaging

autonomous driving

Robust

formulated as principled geometric optimization problem

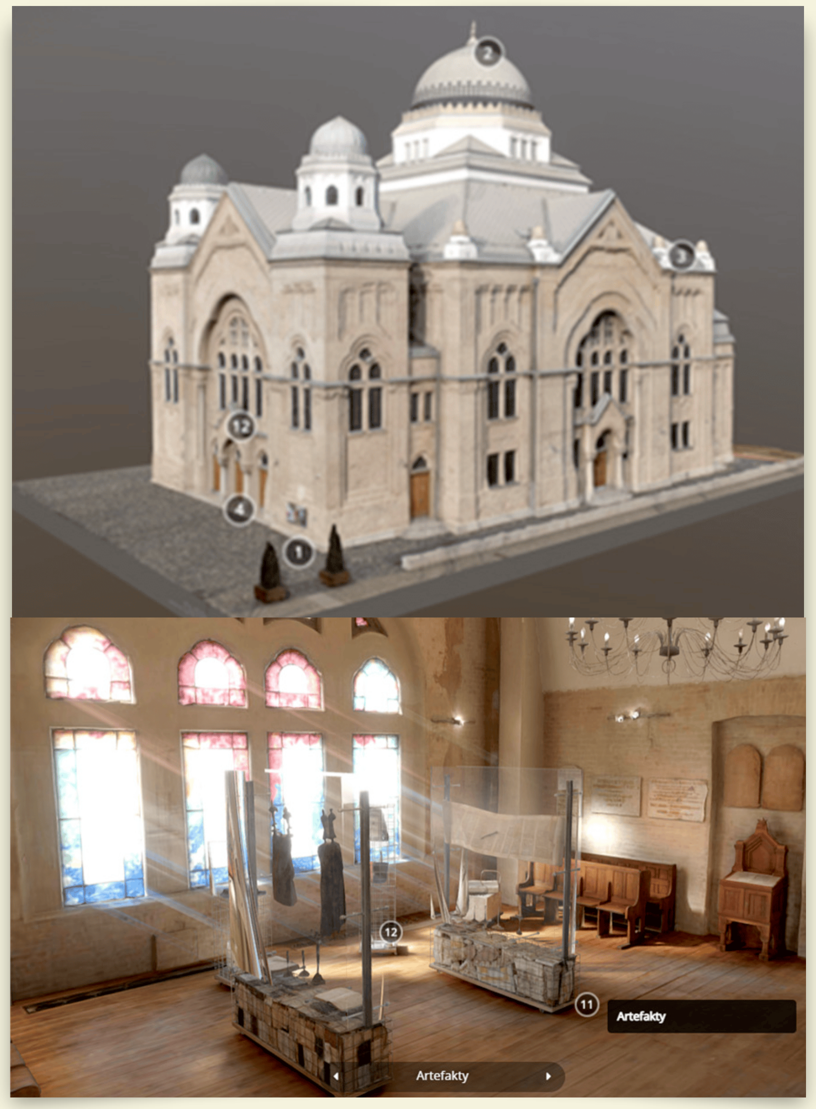
geology



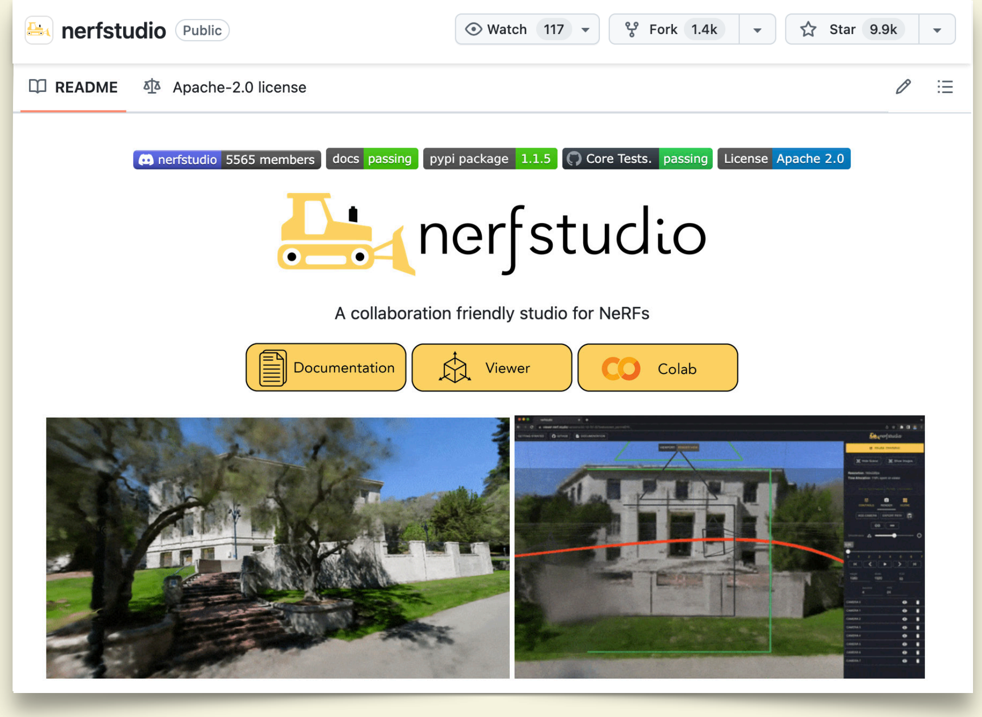
forestry

Simple

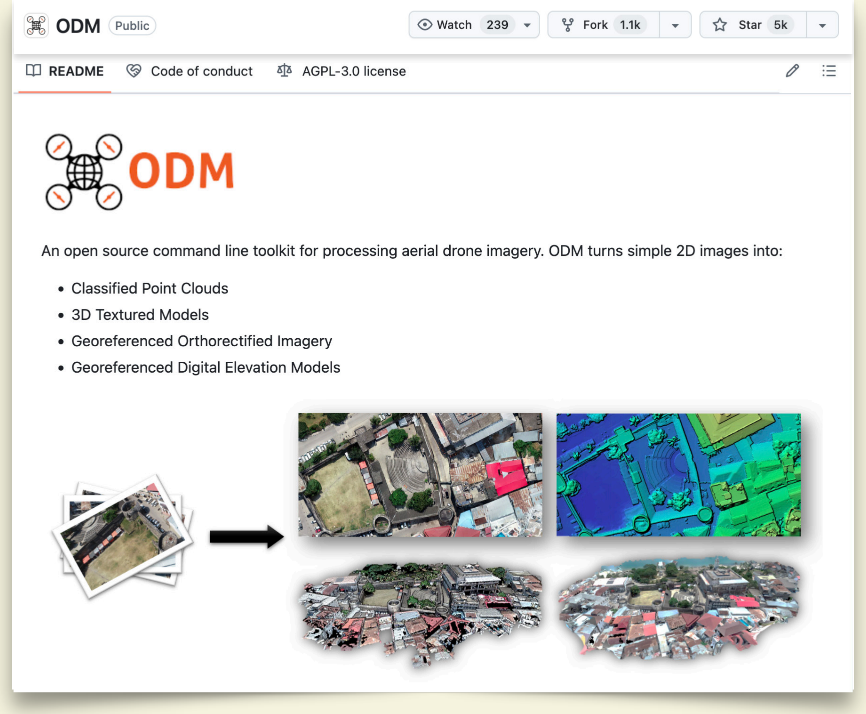
reduces problem to numerical linear algebra



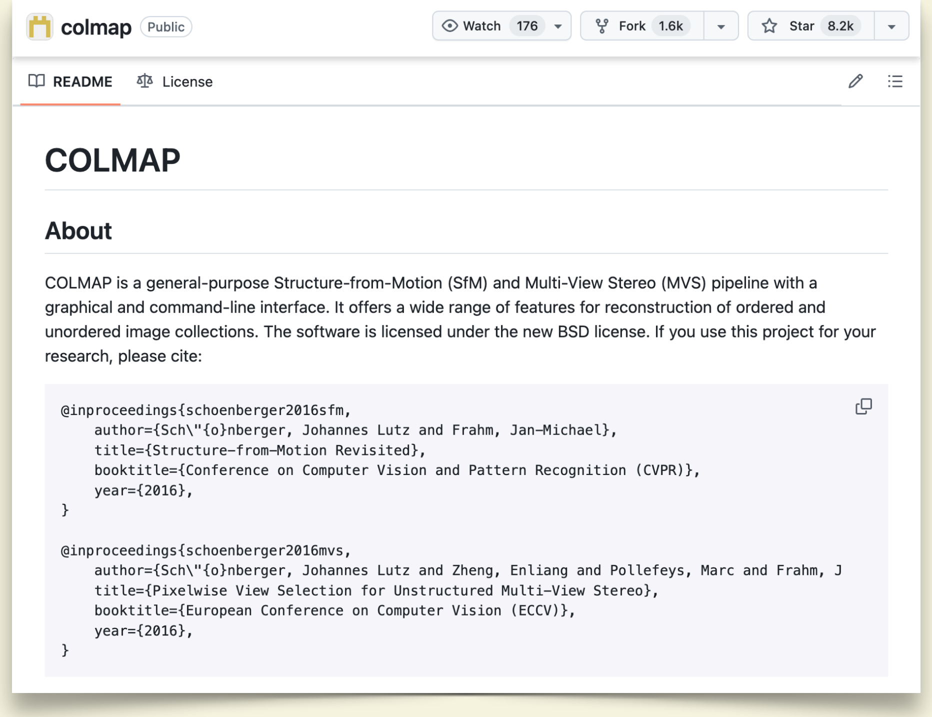
architecture



neural reconstruction



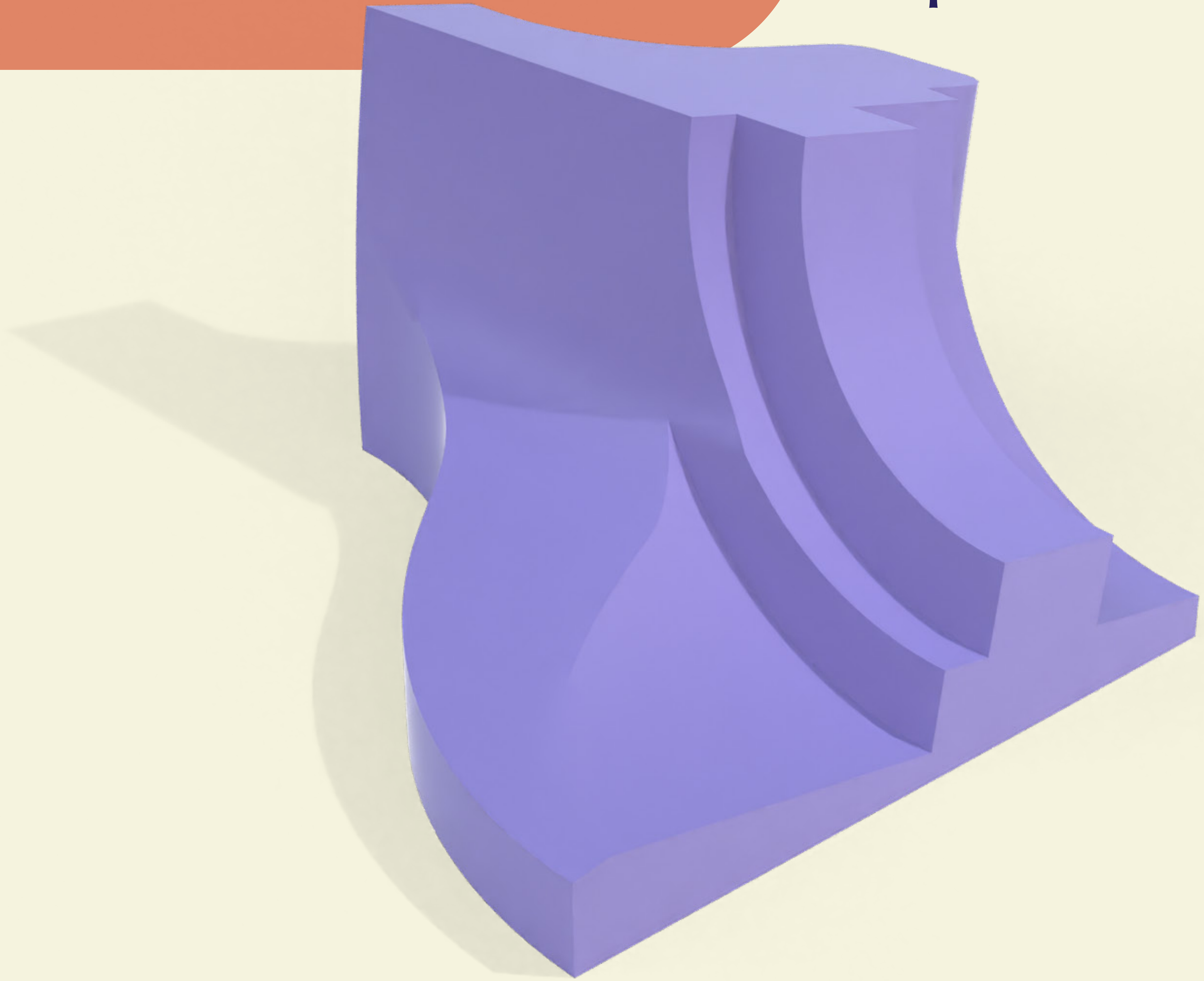
drone mapping



photogrammetry

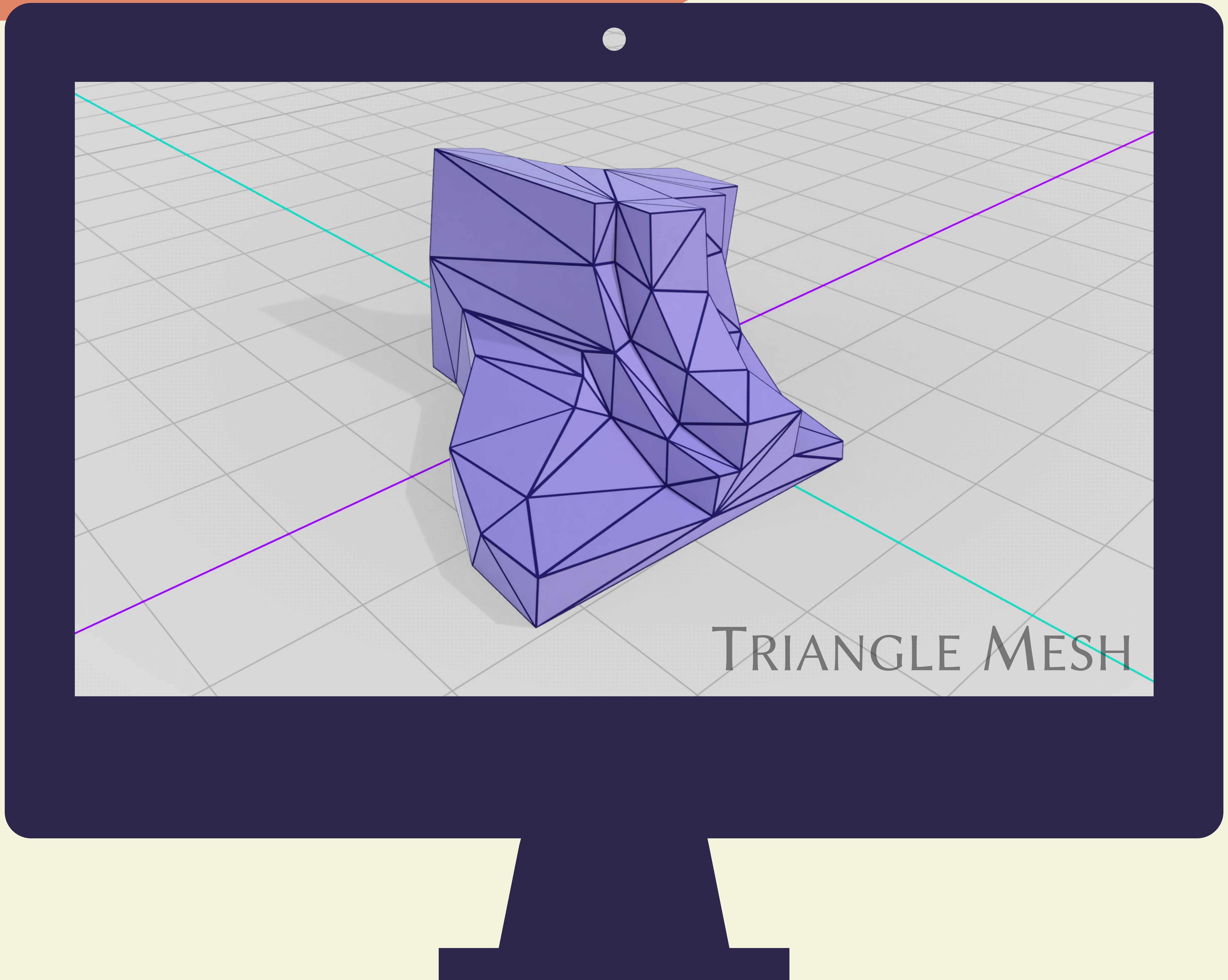
Representing surfaces

How should I
represent my surface?

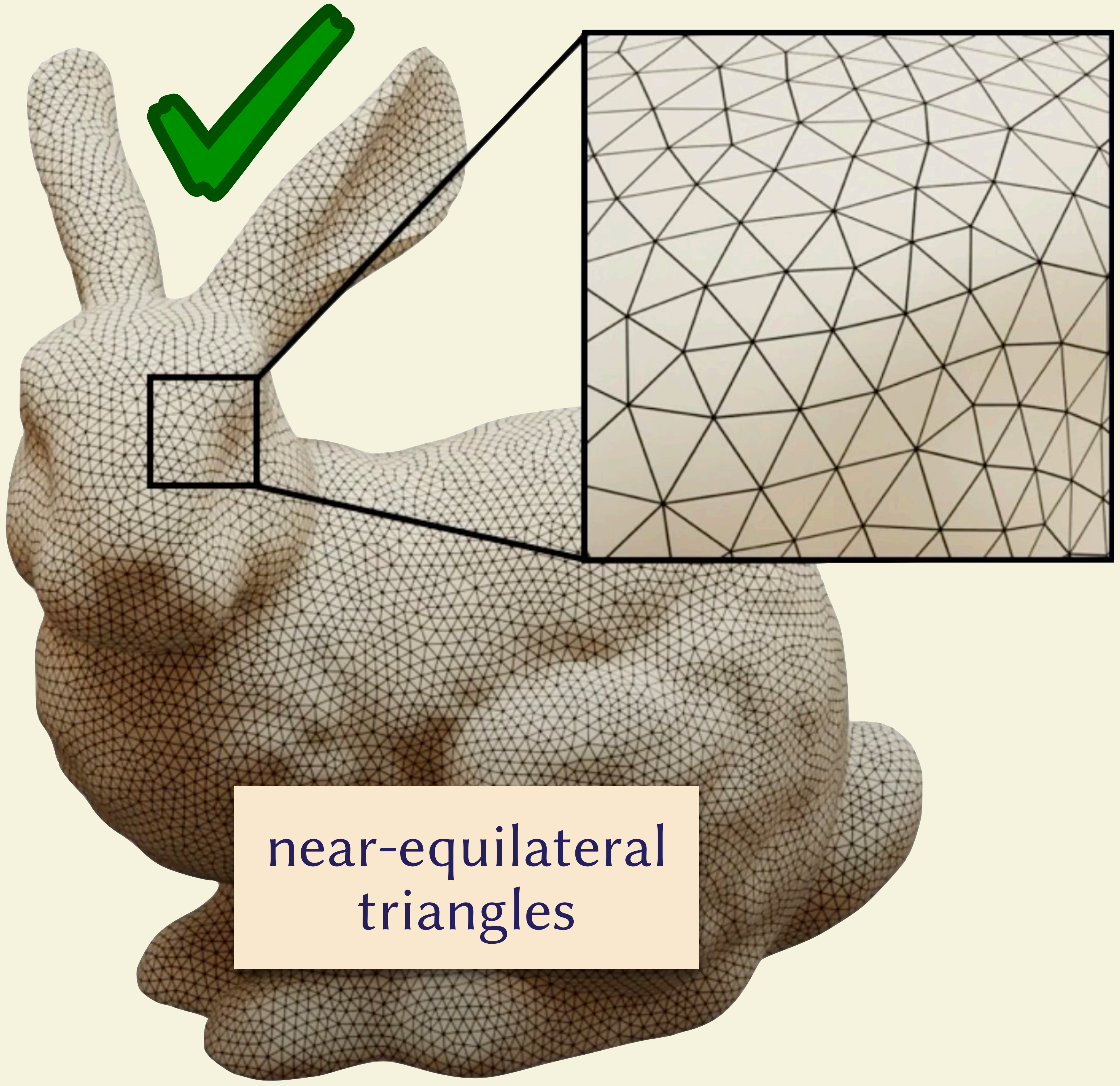


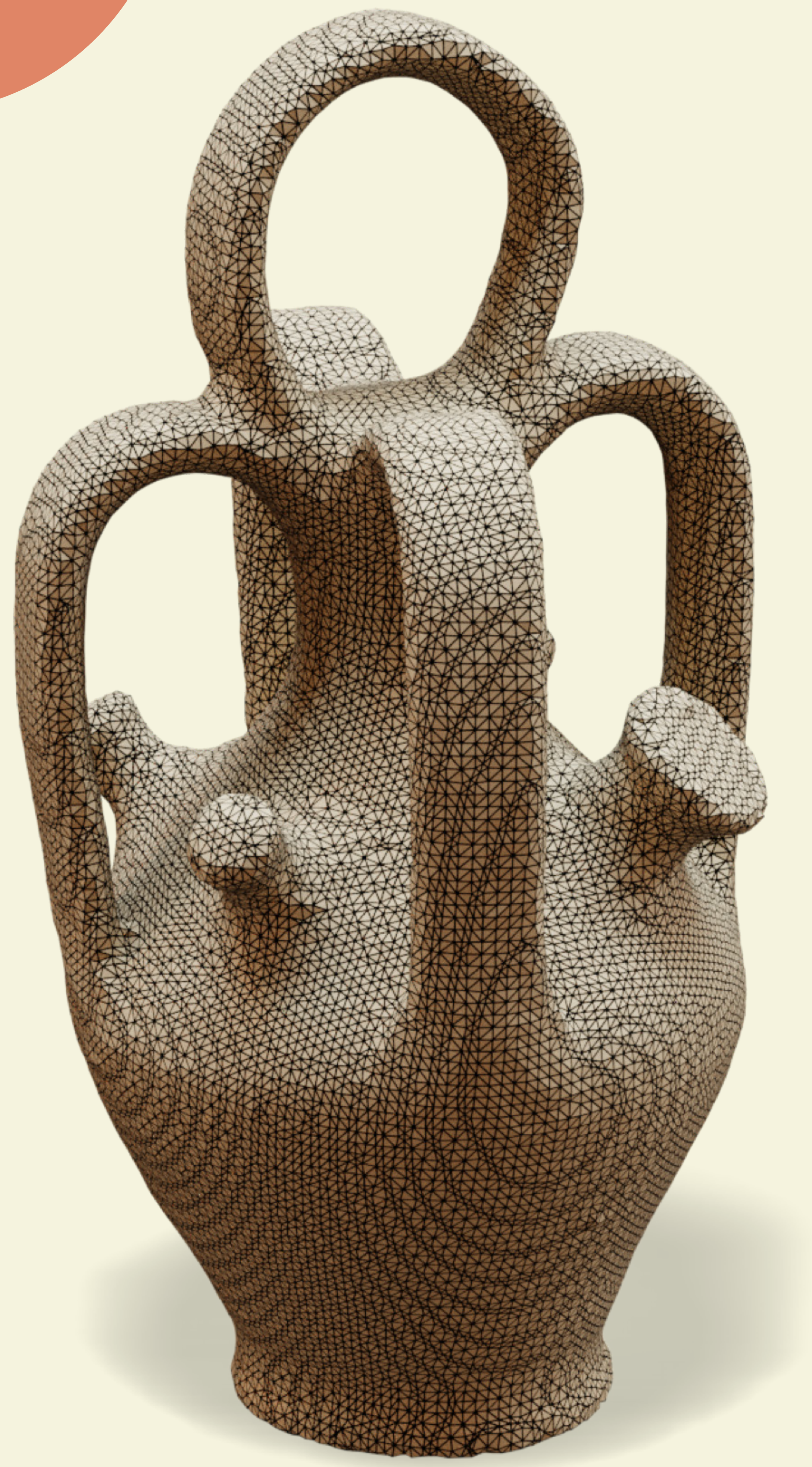
Representing surfaces

How should I represent my surface?



Not all meshes are created equal



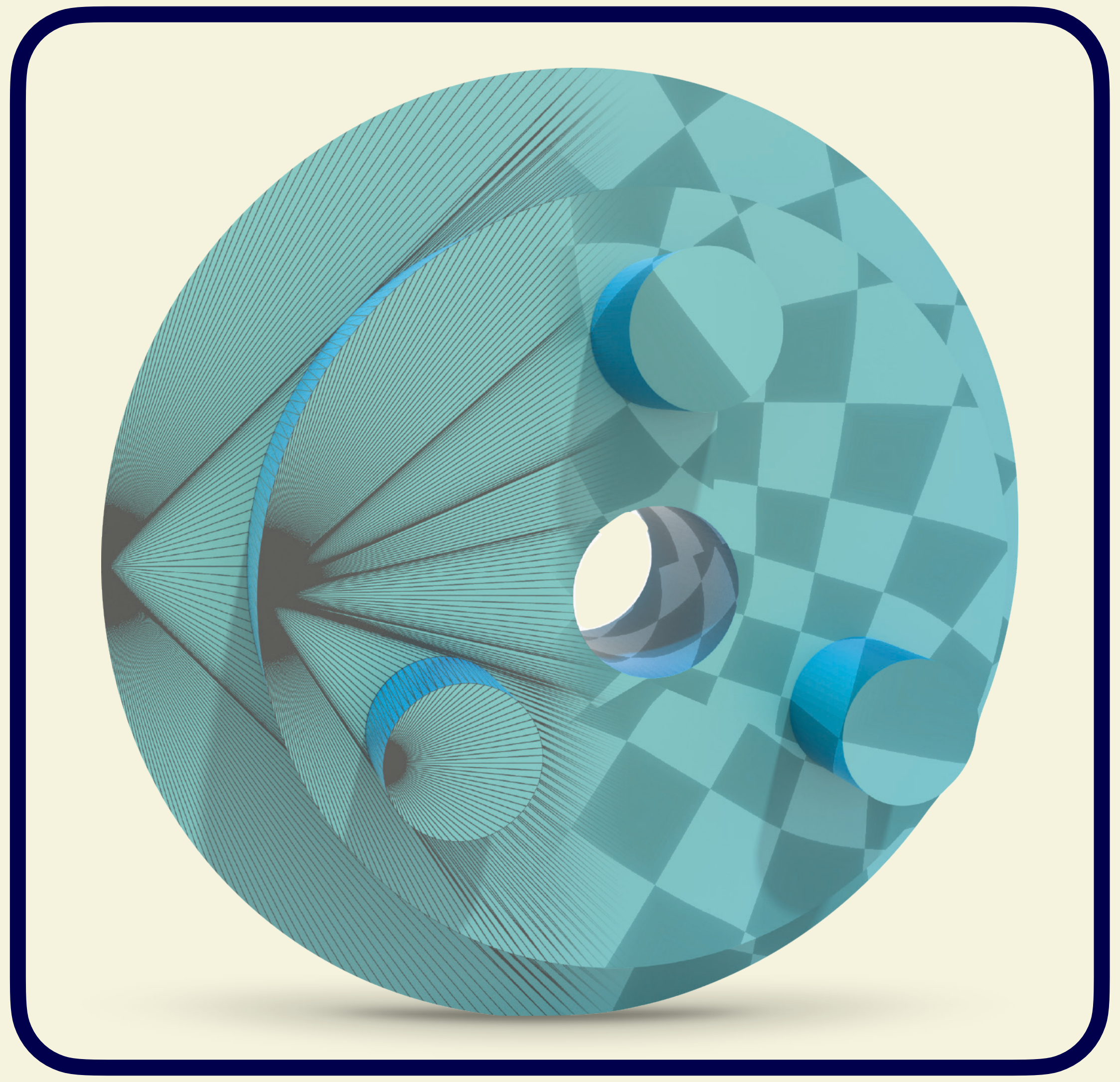


Challenge: Robustness

a hidden opportunity

Outline

I. Surface Parameterization



New algorithm for a ubiquitous subroutine

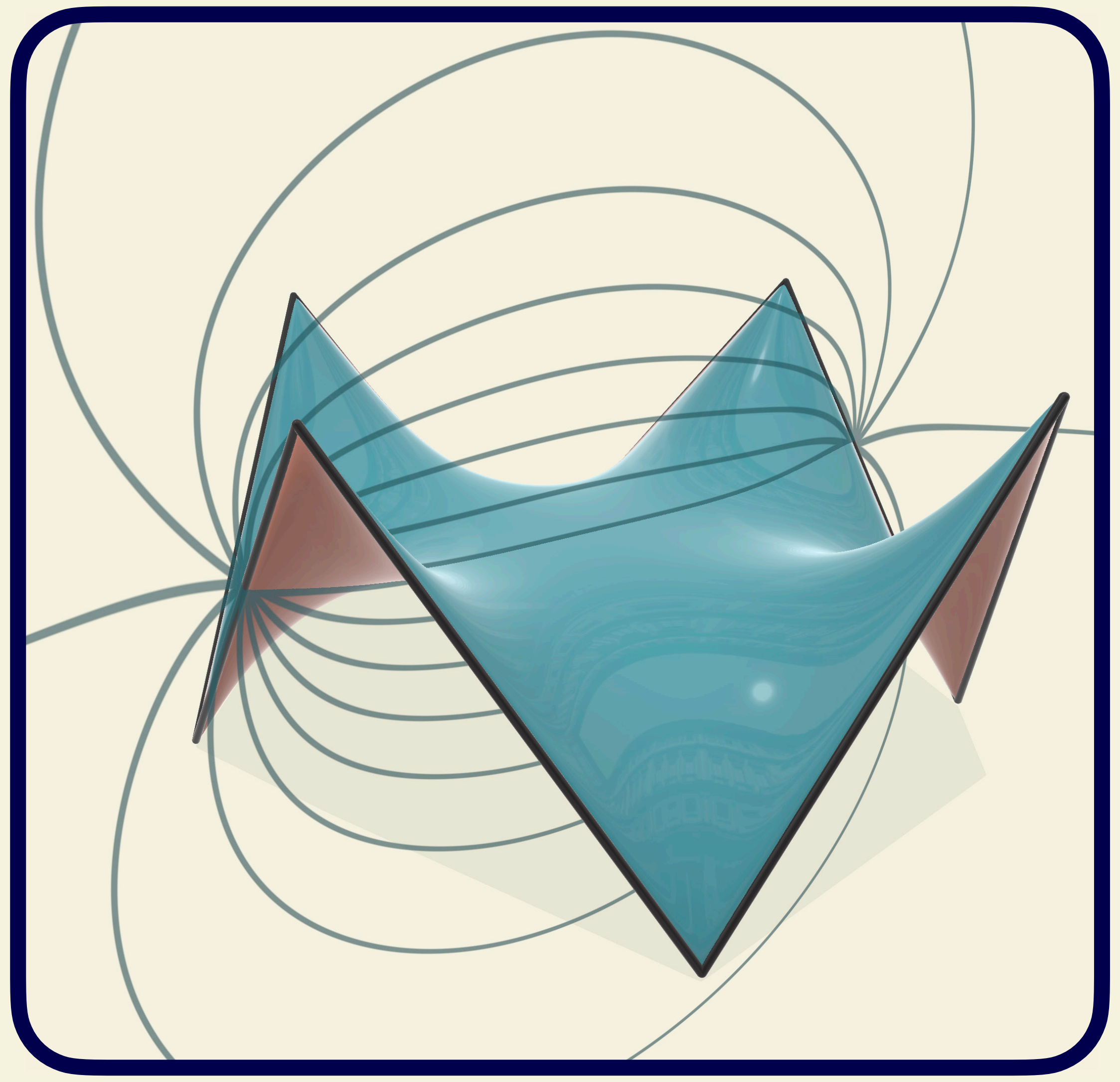
using *hyperbolic polyhedra*

II. Intrinsic Triangulations

General-purpose technique for running classic algorithms more robustly

using *intrinsic geometry*

III. Implicit Surfaces



Efficient algorithm for rendering a new family of surfaces

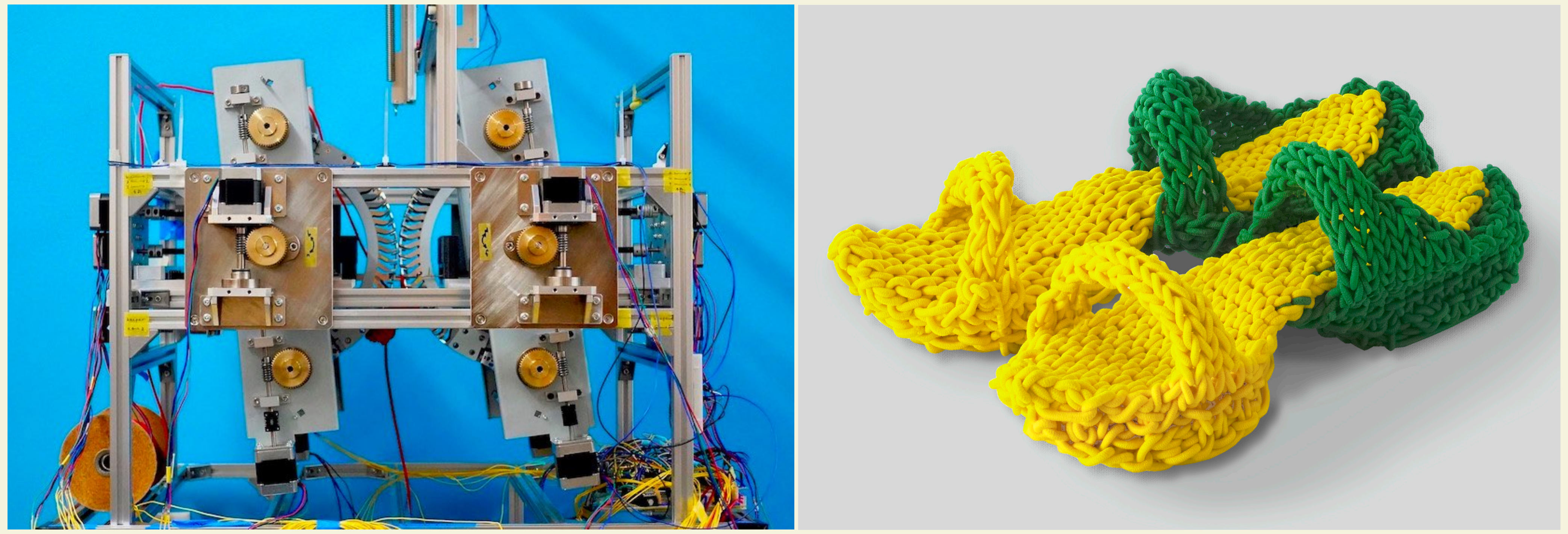
using *harmonic functions*

IV. Future Directions

???

Other interests

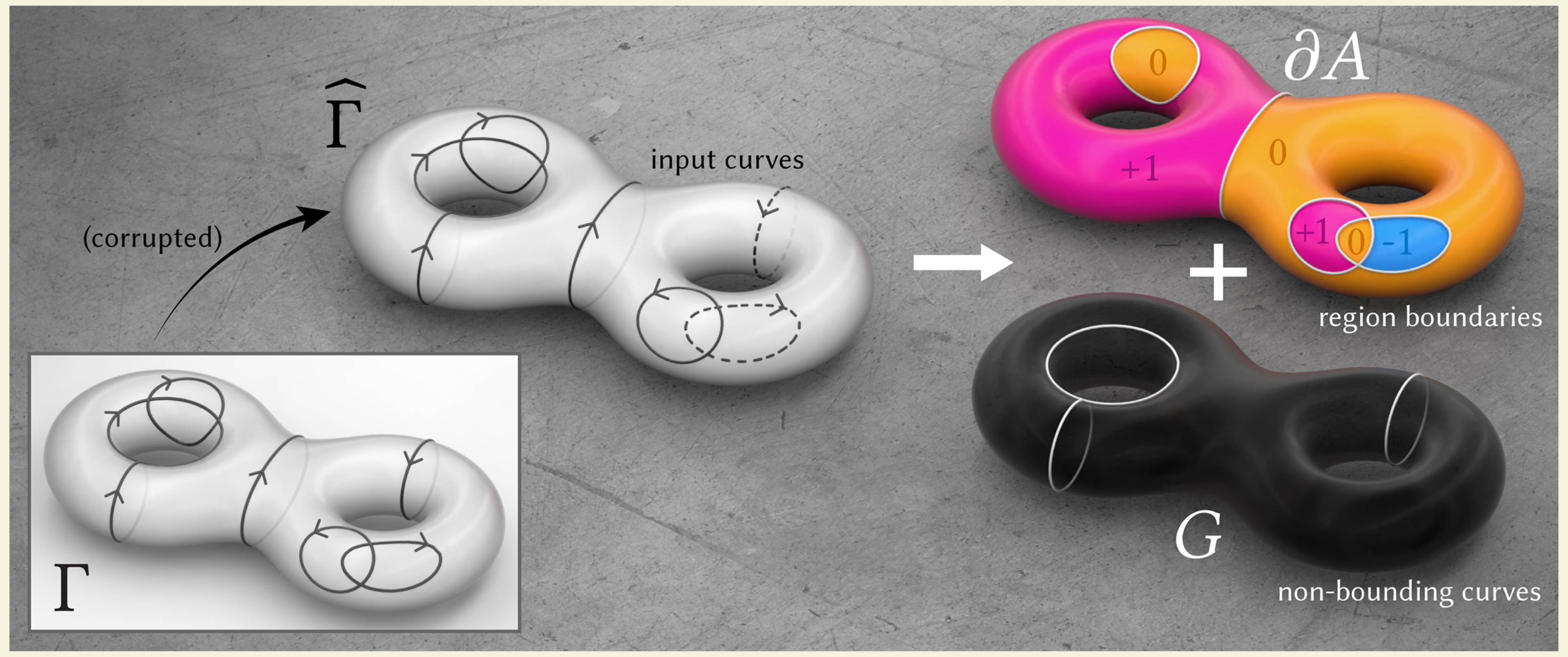
Fabrication



[Hirose, **Gillespie**, Bonilla Fominaya, & McCann. 2024. *Solid Knitting.* ACM Trans. Graph.]

Best Paper (Honorable Mention)

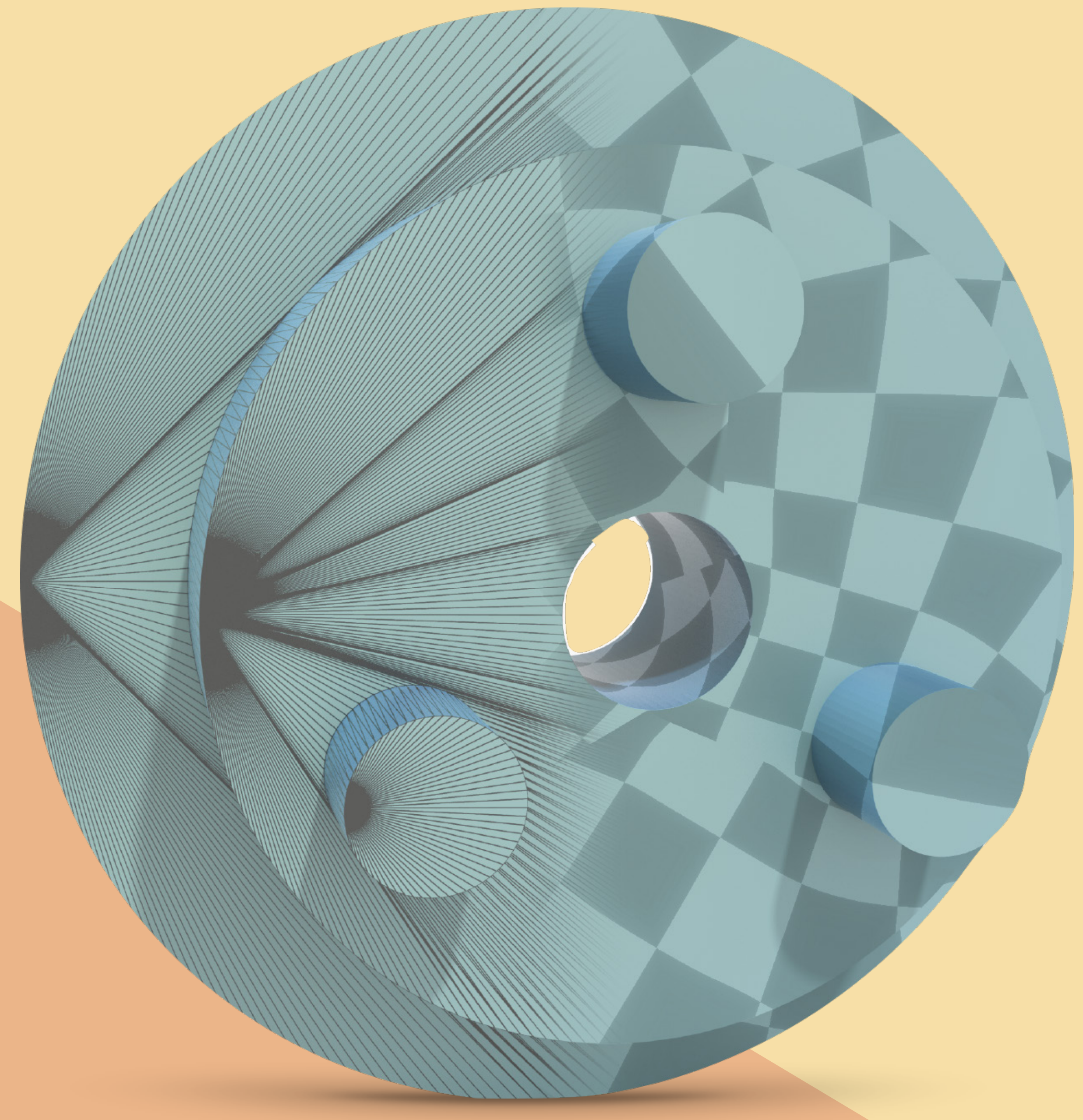
Applied Topology



[Feng, **Gillespie**, & Crane. 2023. *Winding Numbers on Discrete Surfaces.* ACM Trans. Graph.]

Surface Parameterization

[**Gillespie**, Springborn & Crane. 2021. *Discrete Conformal Equivalence of Polyhedral Surfaces.* ACM Trans. Graph.]



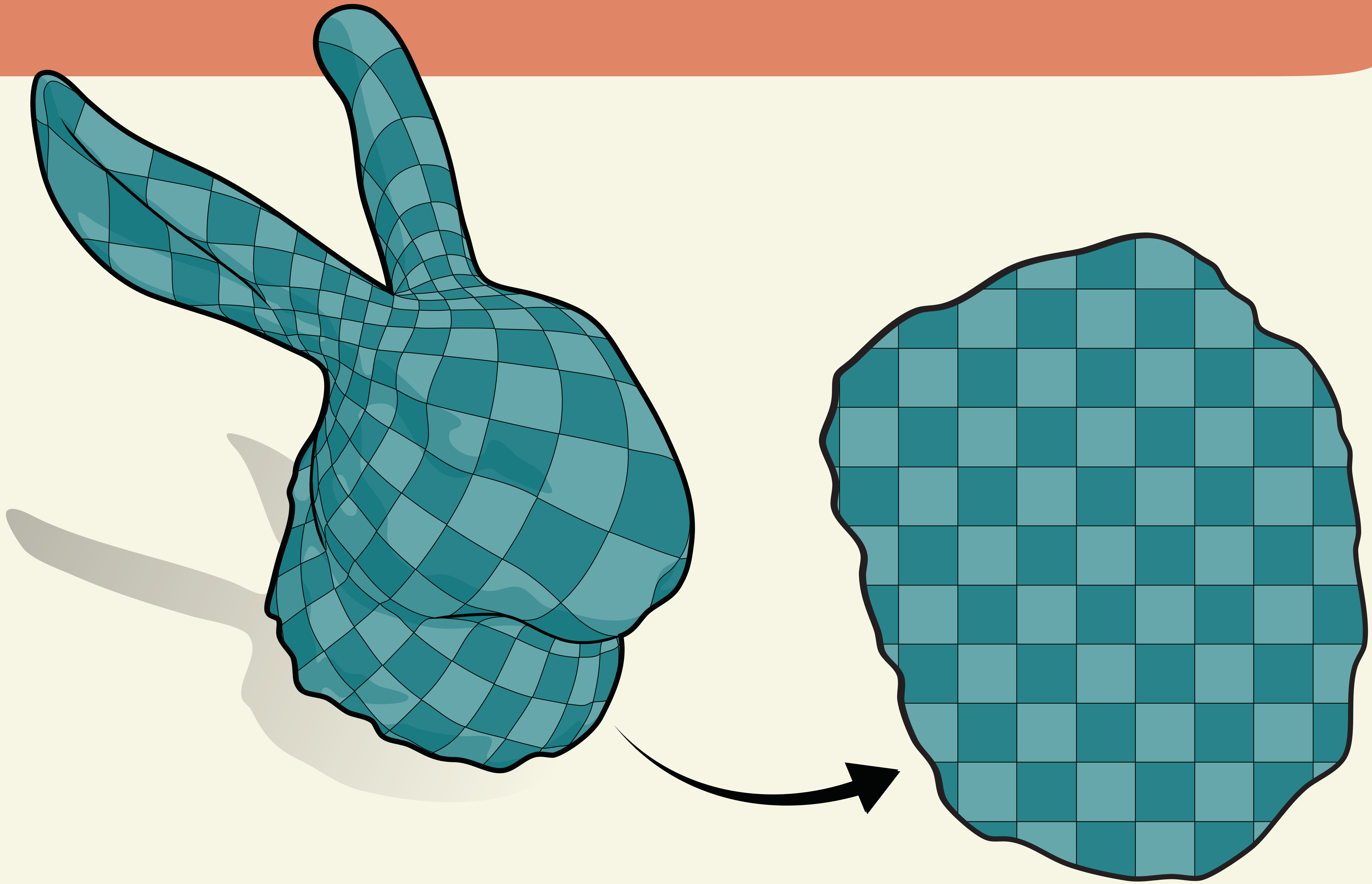
Parameterization

a map from a surface to the plane

[Crane *et al.* 2013]

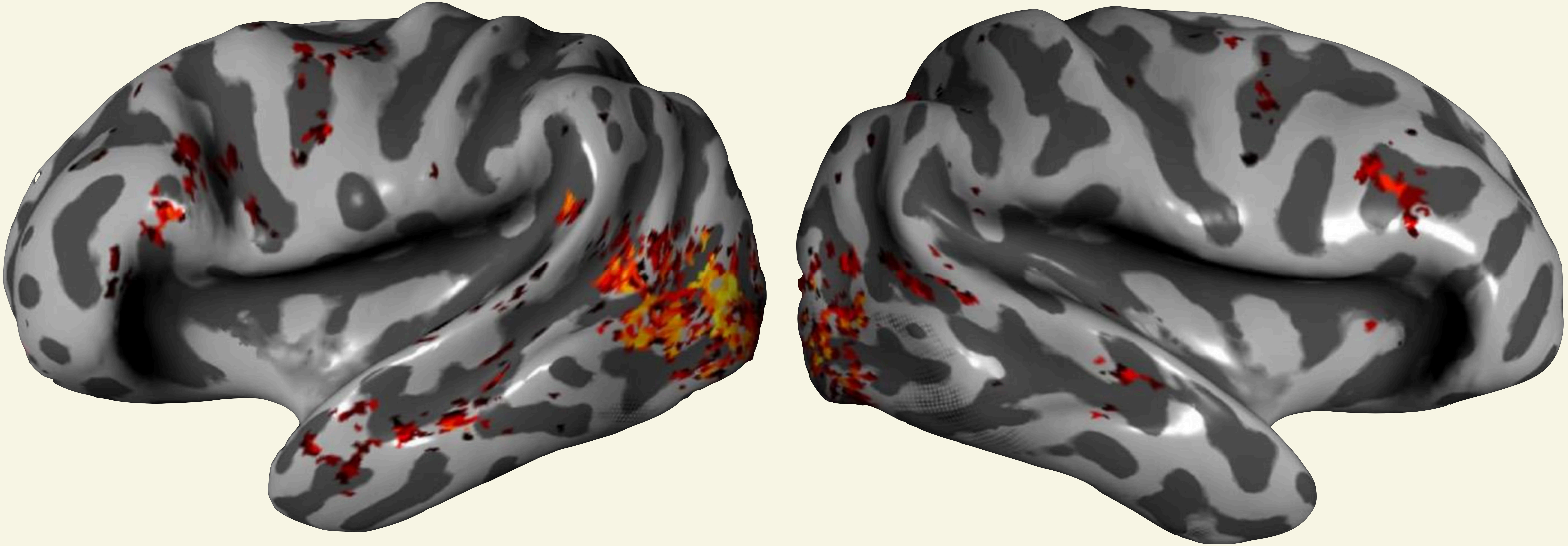
[Kunimune 2018]

Parameterization of general surfaces



Applications of parameterization

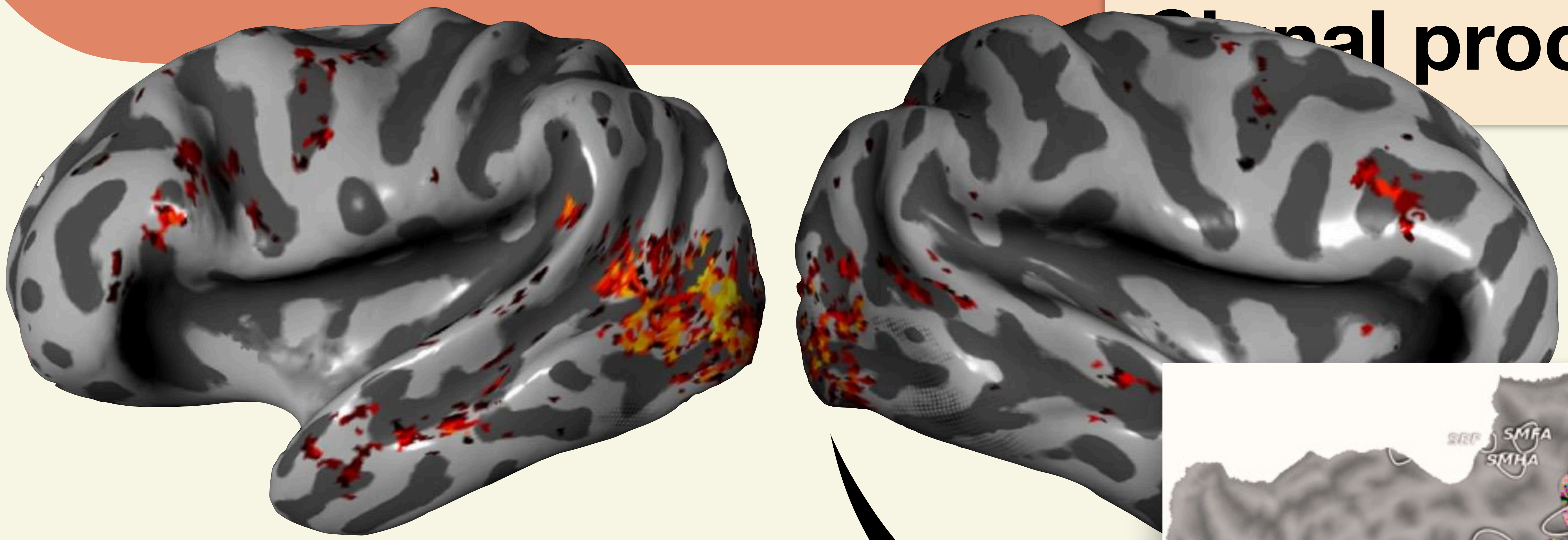
Signal processing

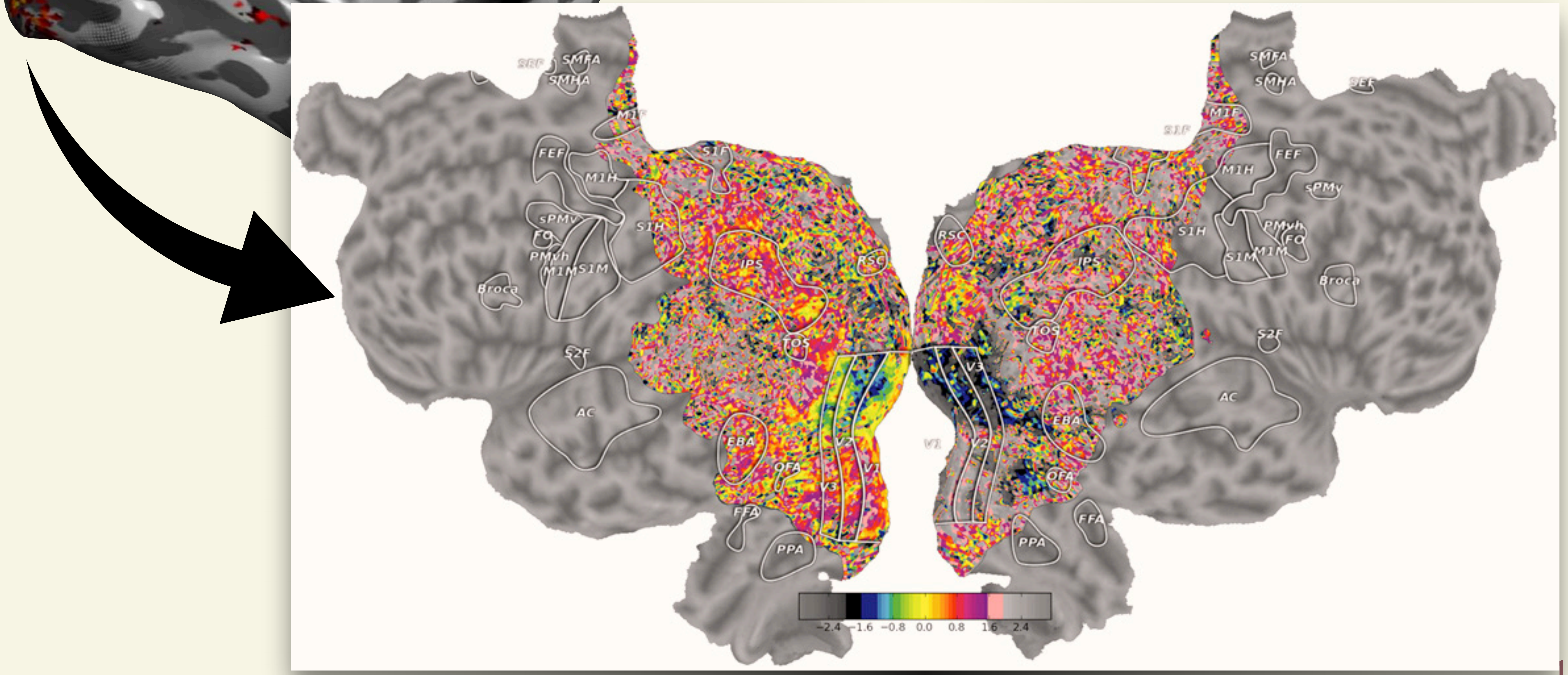


[Gao *et al.* 2015]

Applications of parameterization

al processing

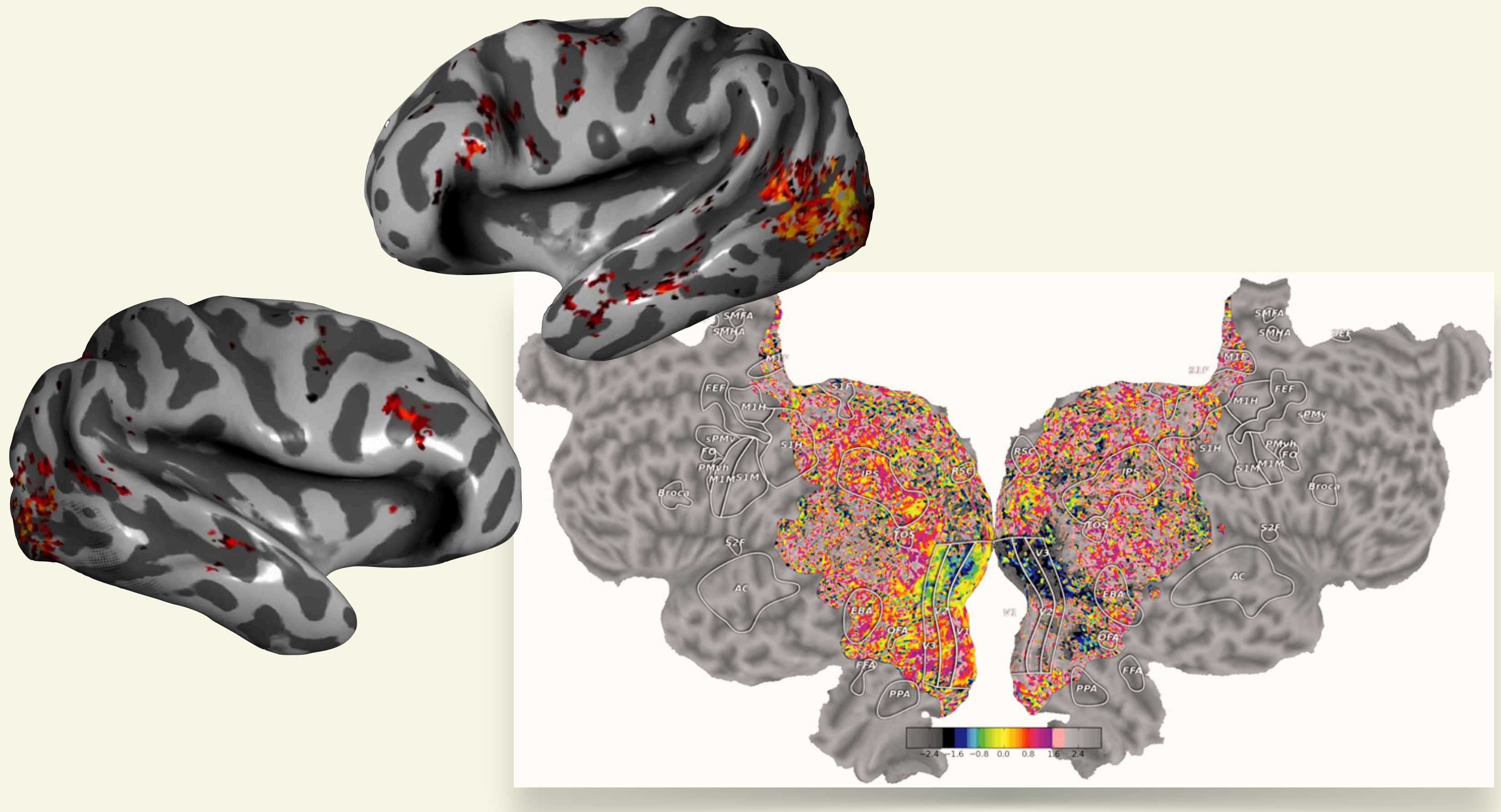




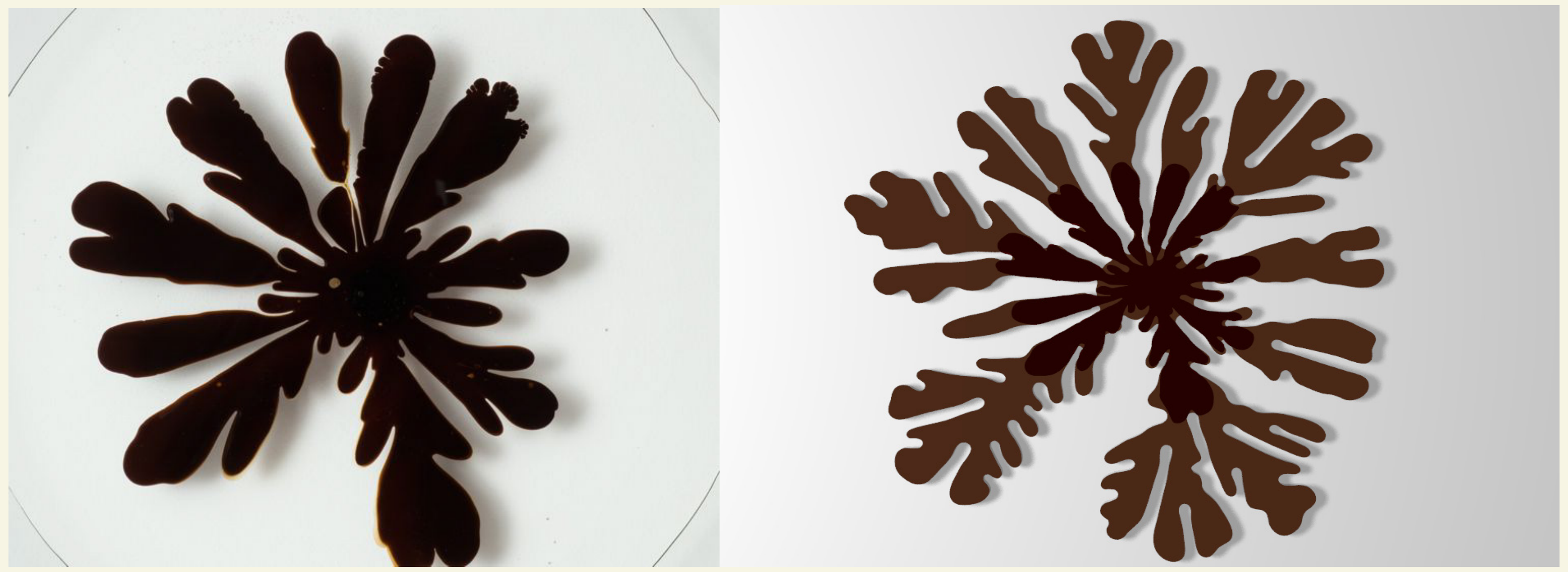
[Gao *et al.* 2015]

Applications of parameterization

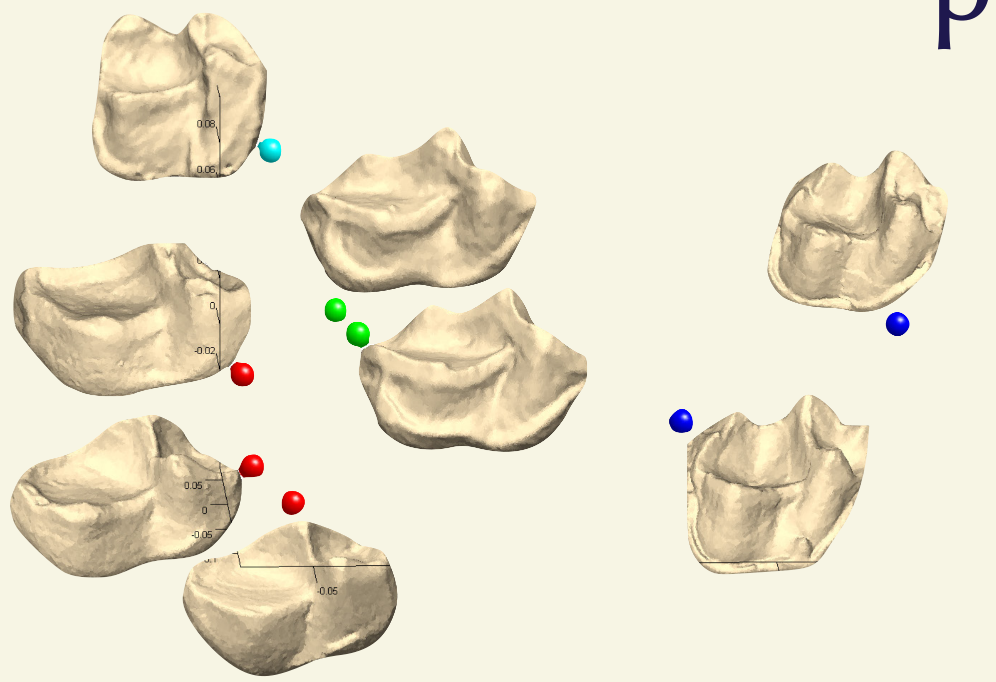
Signal processing



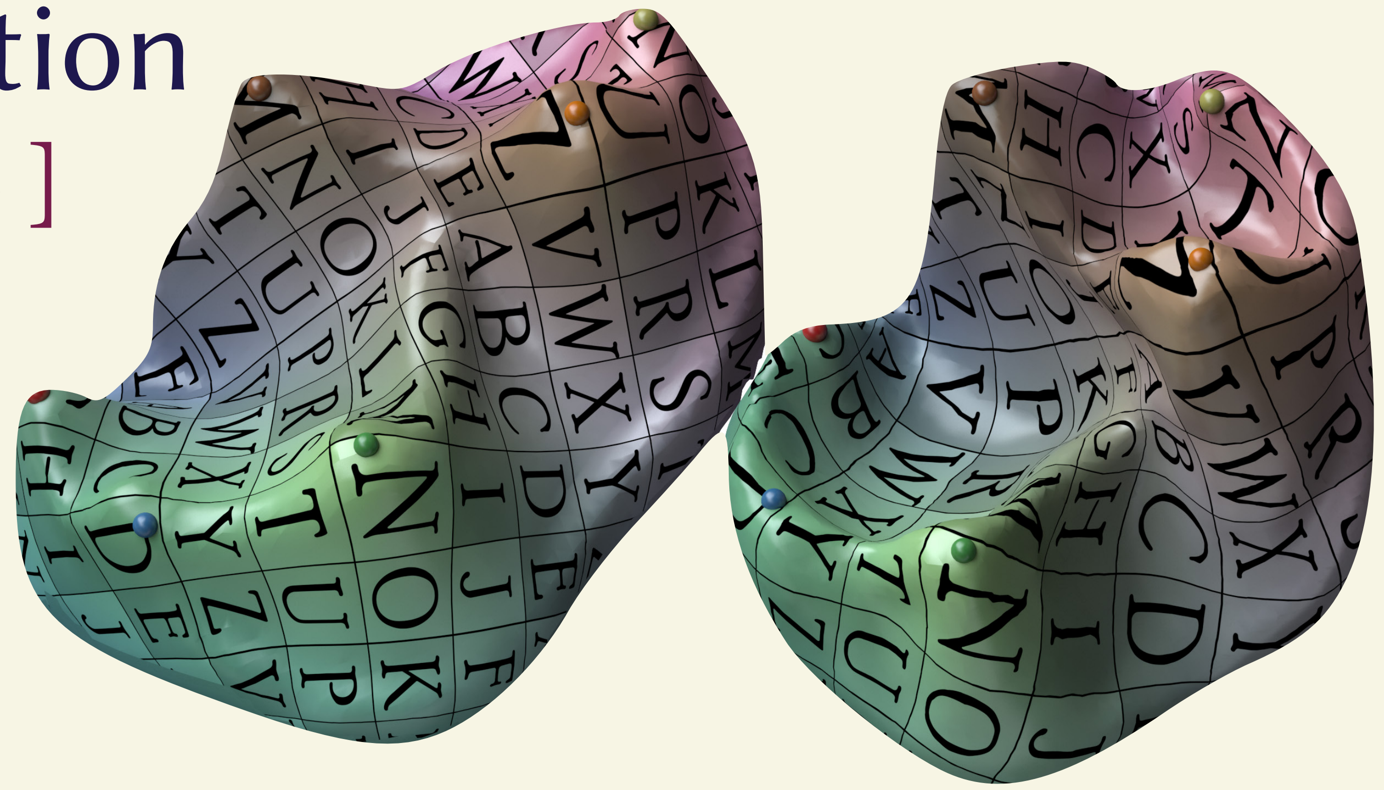
neuroscience
[Gao *et al.* 2015]



physical simulation
[Segall *et al.* 2016]



shape analysis
[Lipman & Daubechies 2011]

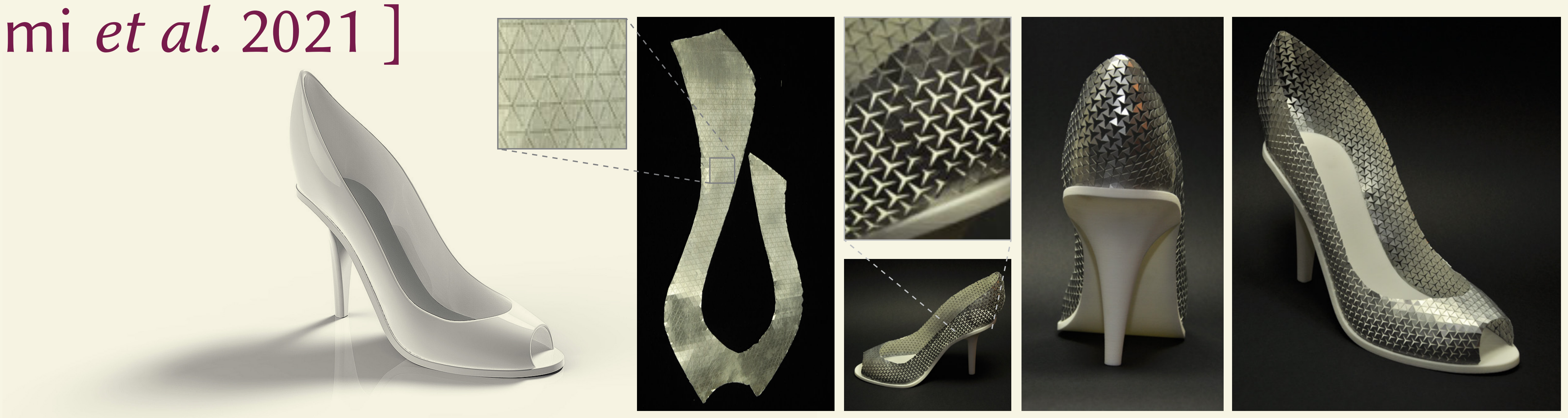


learning on surfaces
[Maron *et al.* 2017]

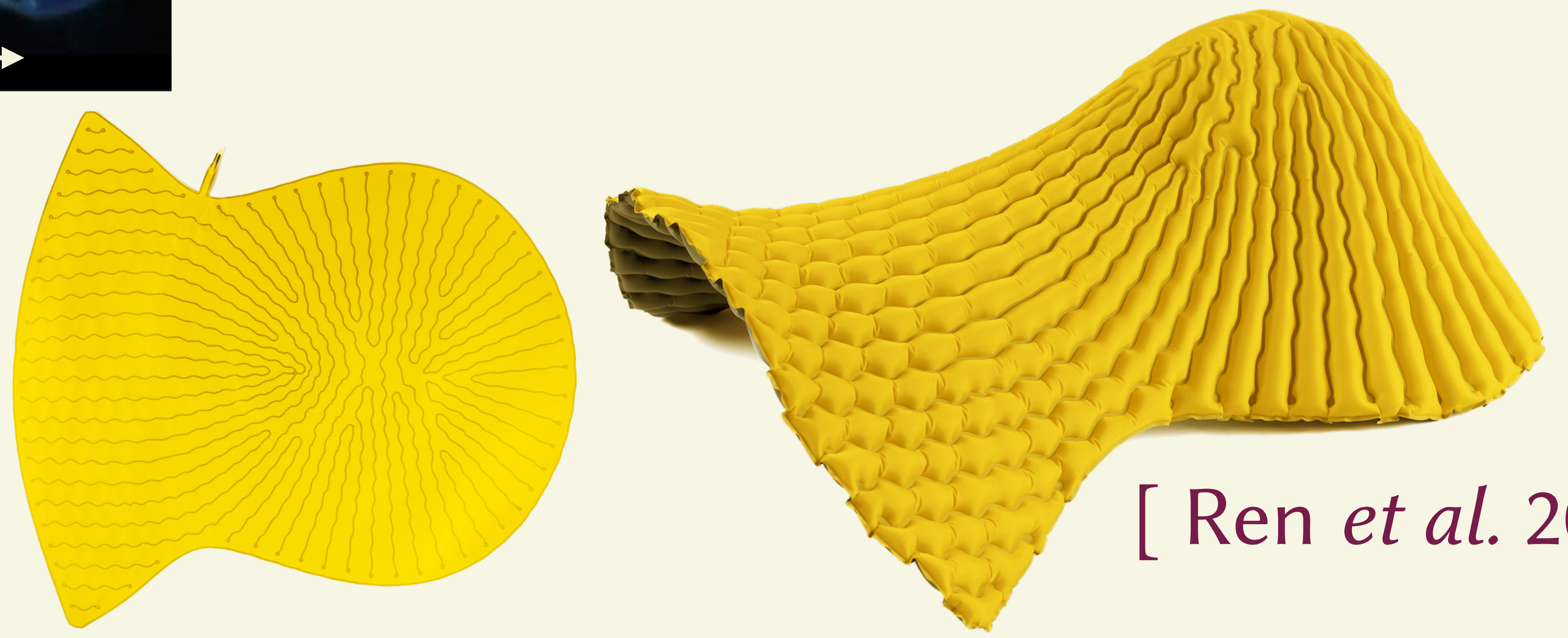
Applications of parameterization

Fabrication

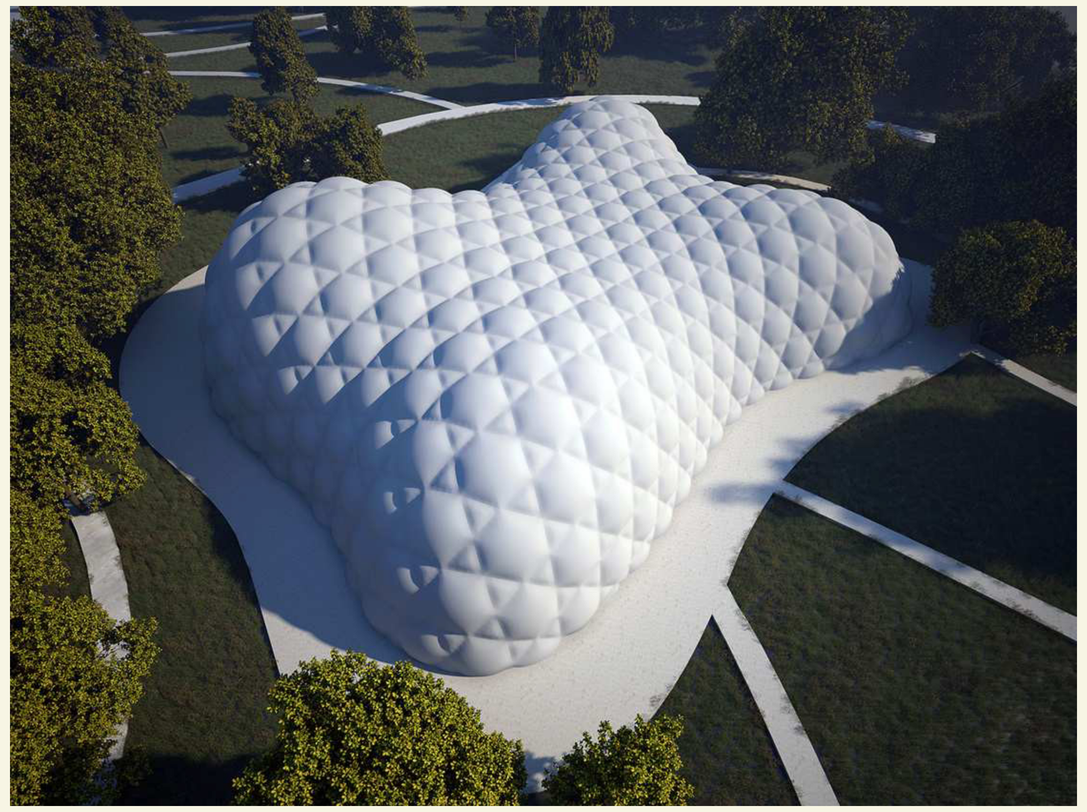
[Nojoomi *et al.* 2021]



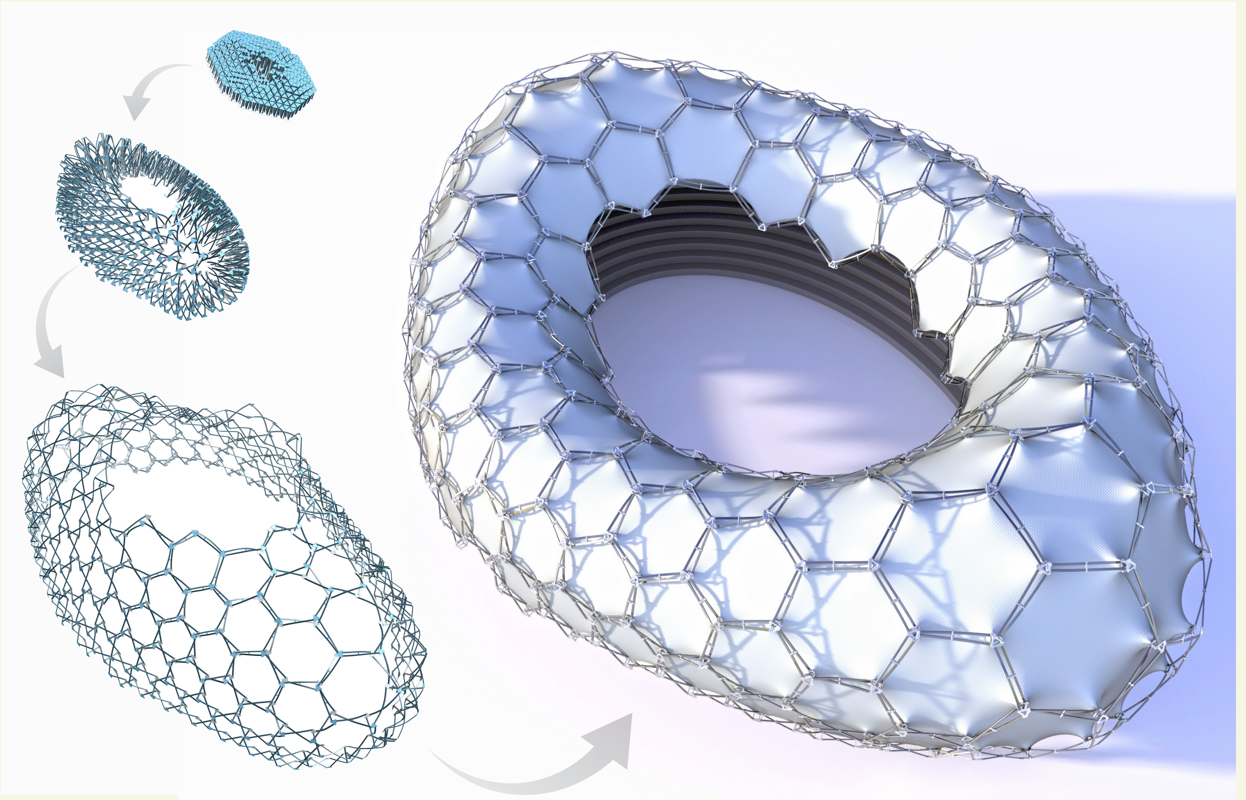
[Konaković *et al.* 2016]



[Ren *et al.* 2024]



[Konaković-Luković *et al.* 2018]



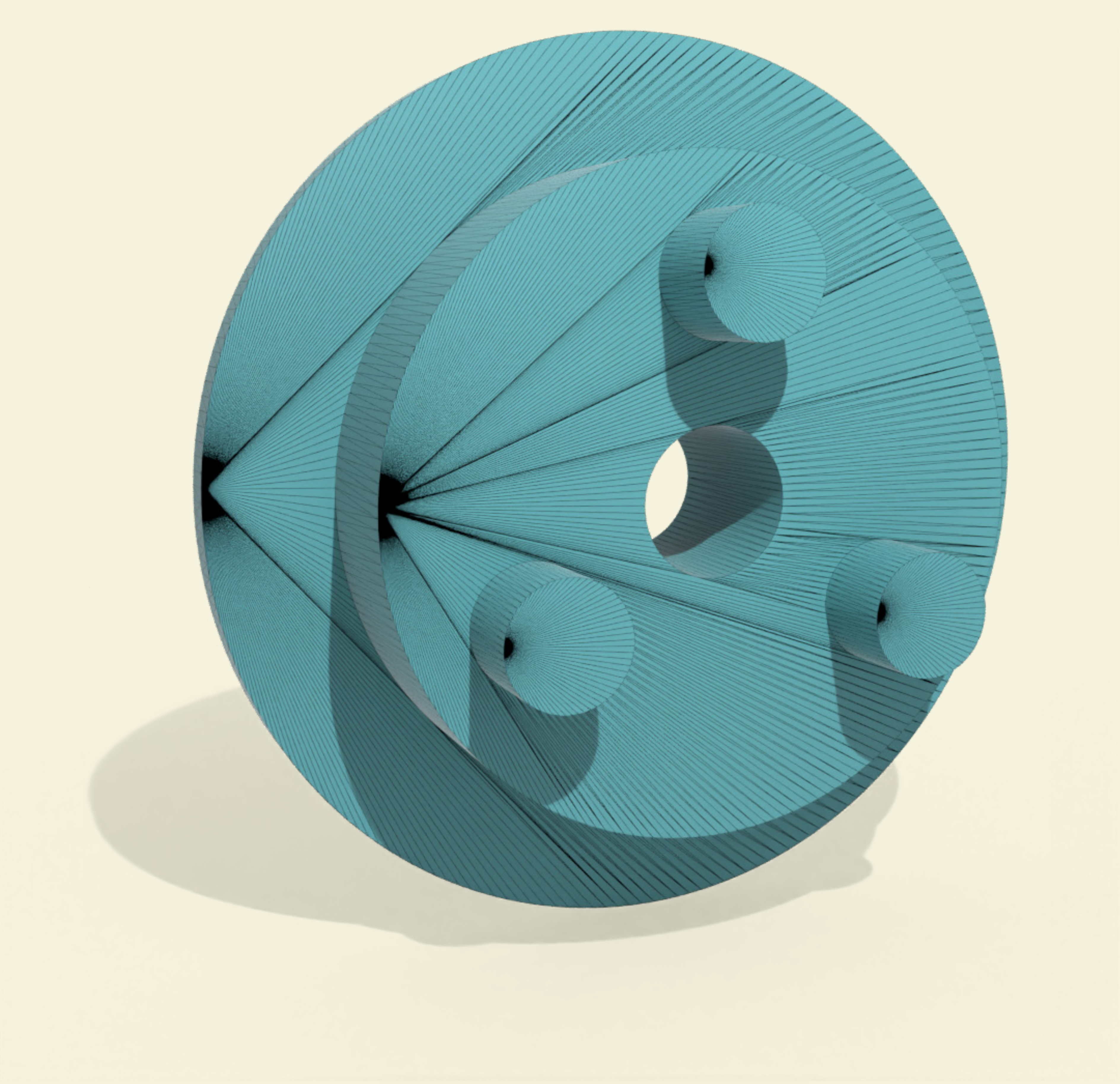
[Ren *et al.* 2022]

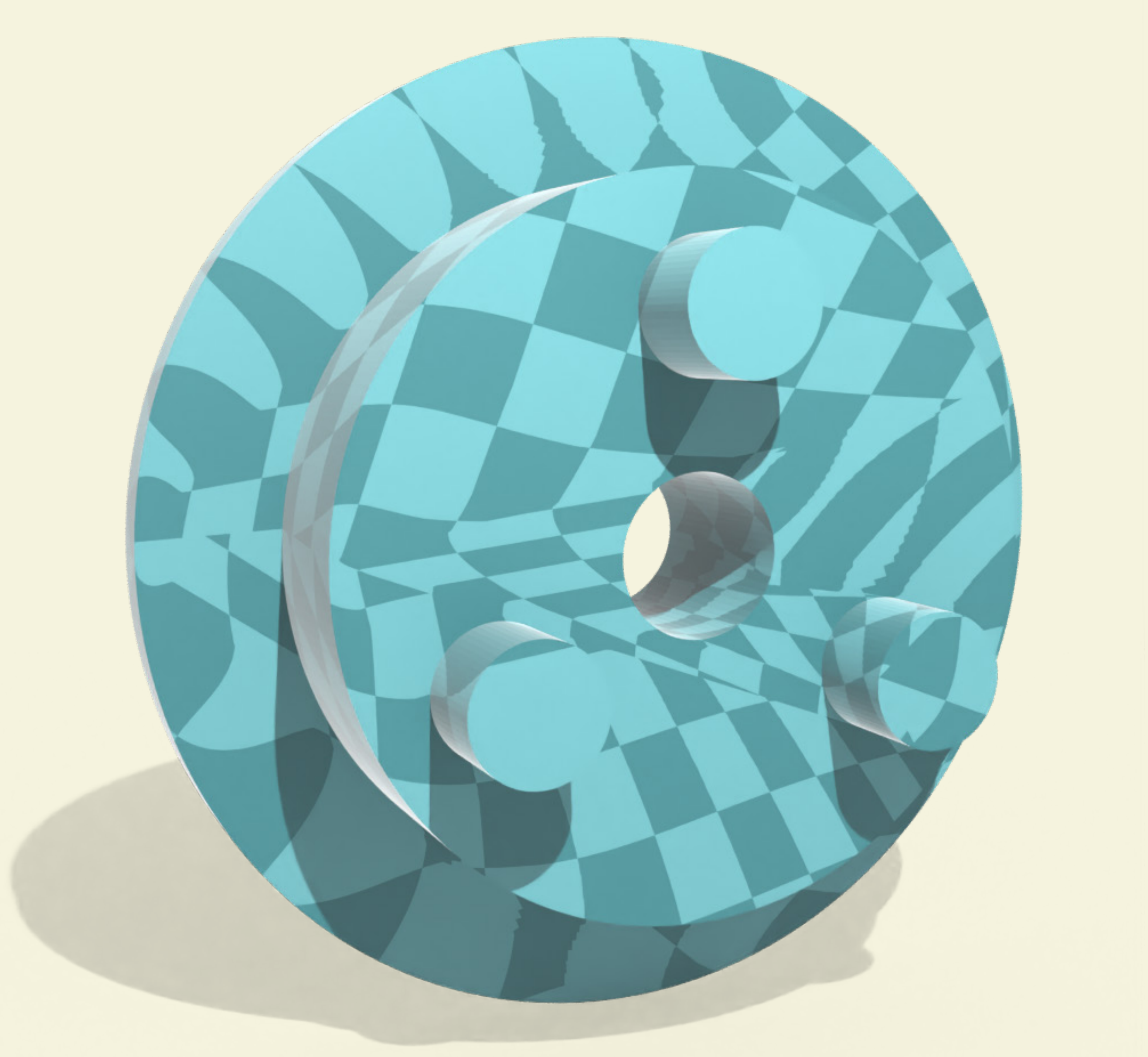
Applications of parameterization

Texturing 3d models

[Timen 2012]

Mesh dependence





[Sawhney & Crane 2017]

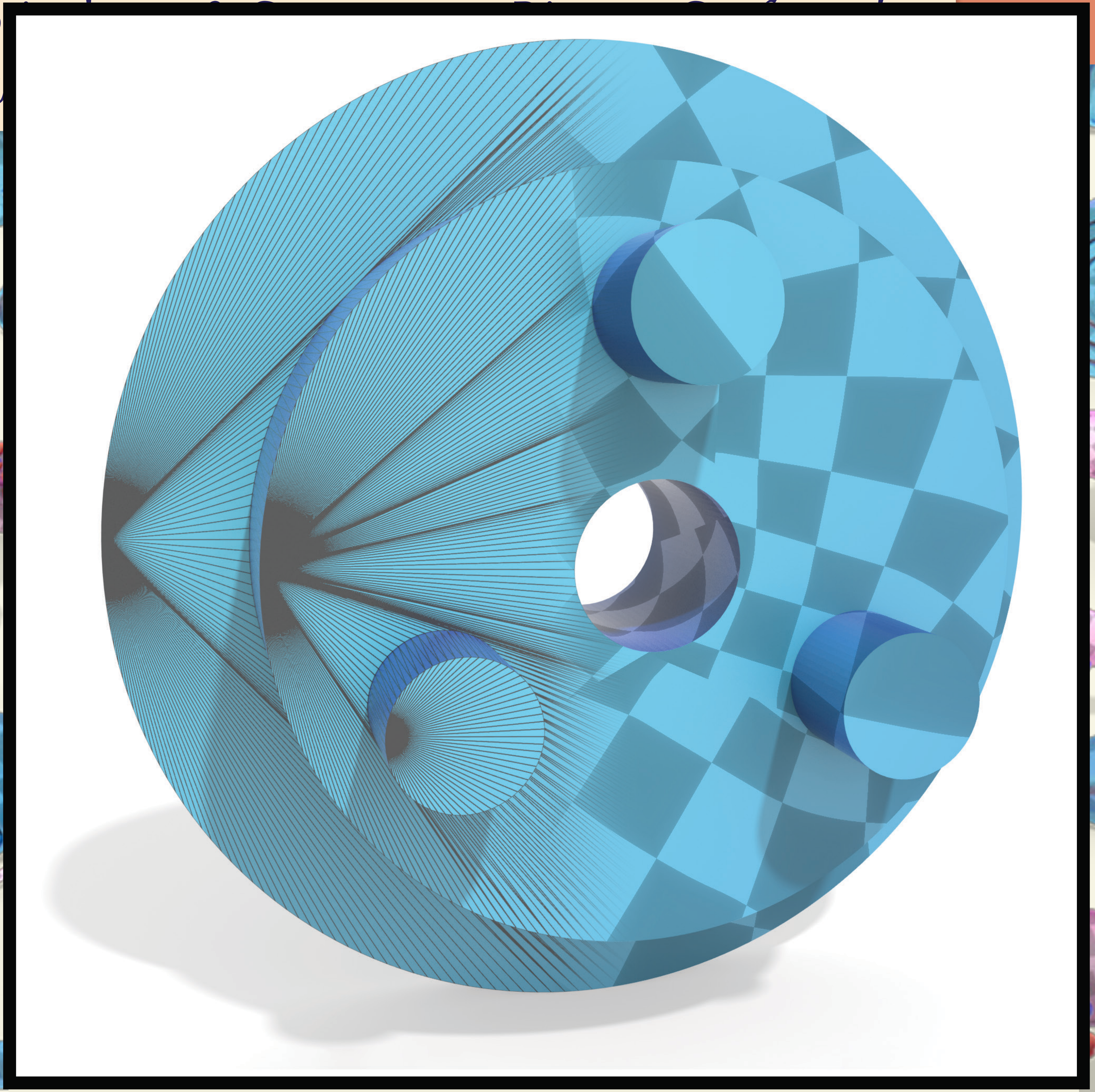
- ✗ high distortion
- ✗ not injective

Robust surface parameterization

[**Gillespie**, Springborn, & Crane. 2021. *Discrete Conformal Equivalence of Polyhedral Surfaces.* ACM Trans. Graph.]

Robust surface parameterization

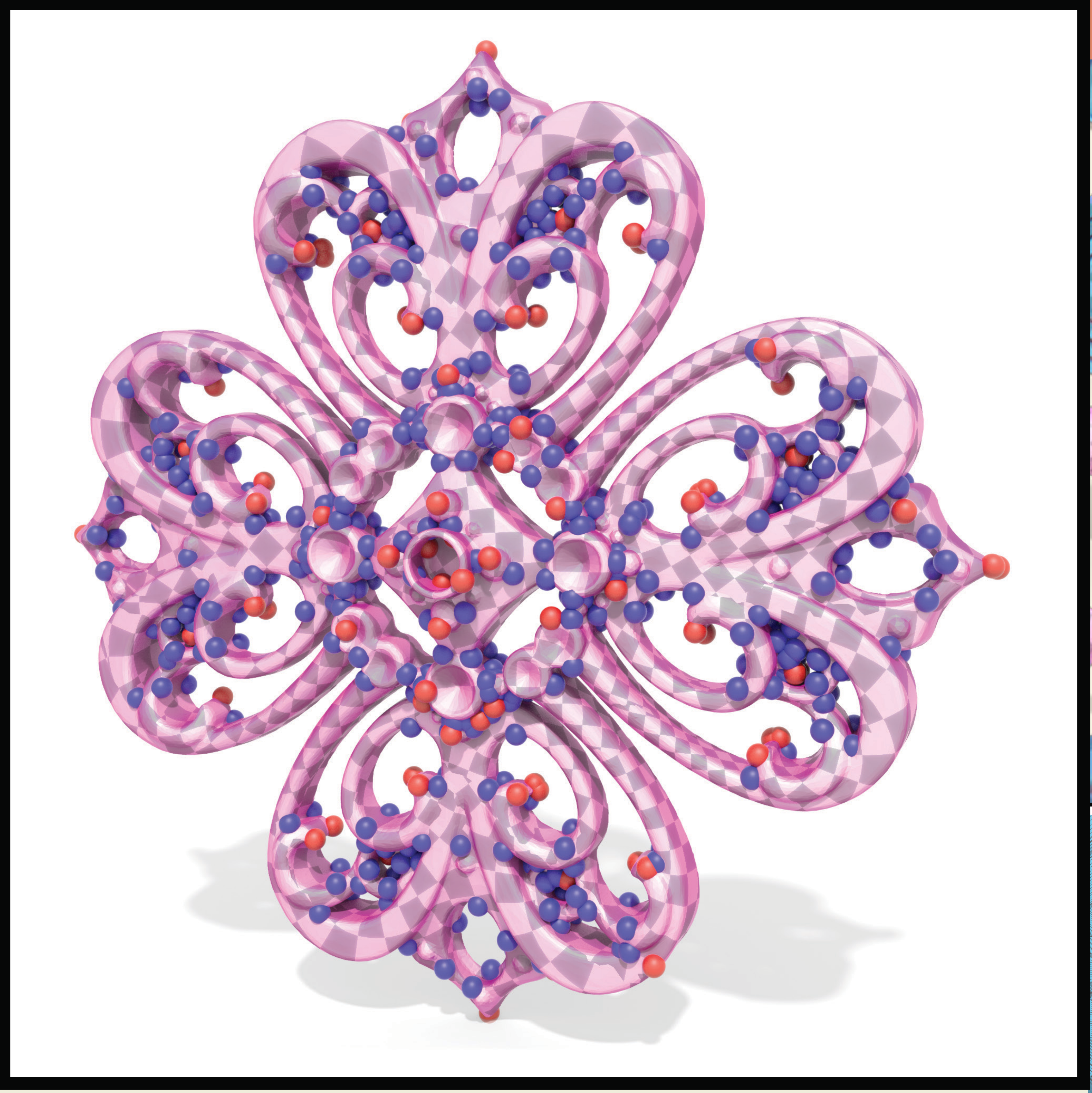
[**Gillespie**, Sp
Equivalence o



challenging input triangulations

Robust surface parameterization

[**Gillespie**, Sp
Equivalence o

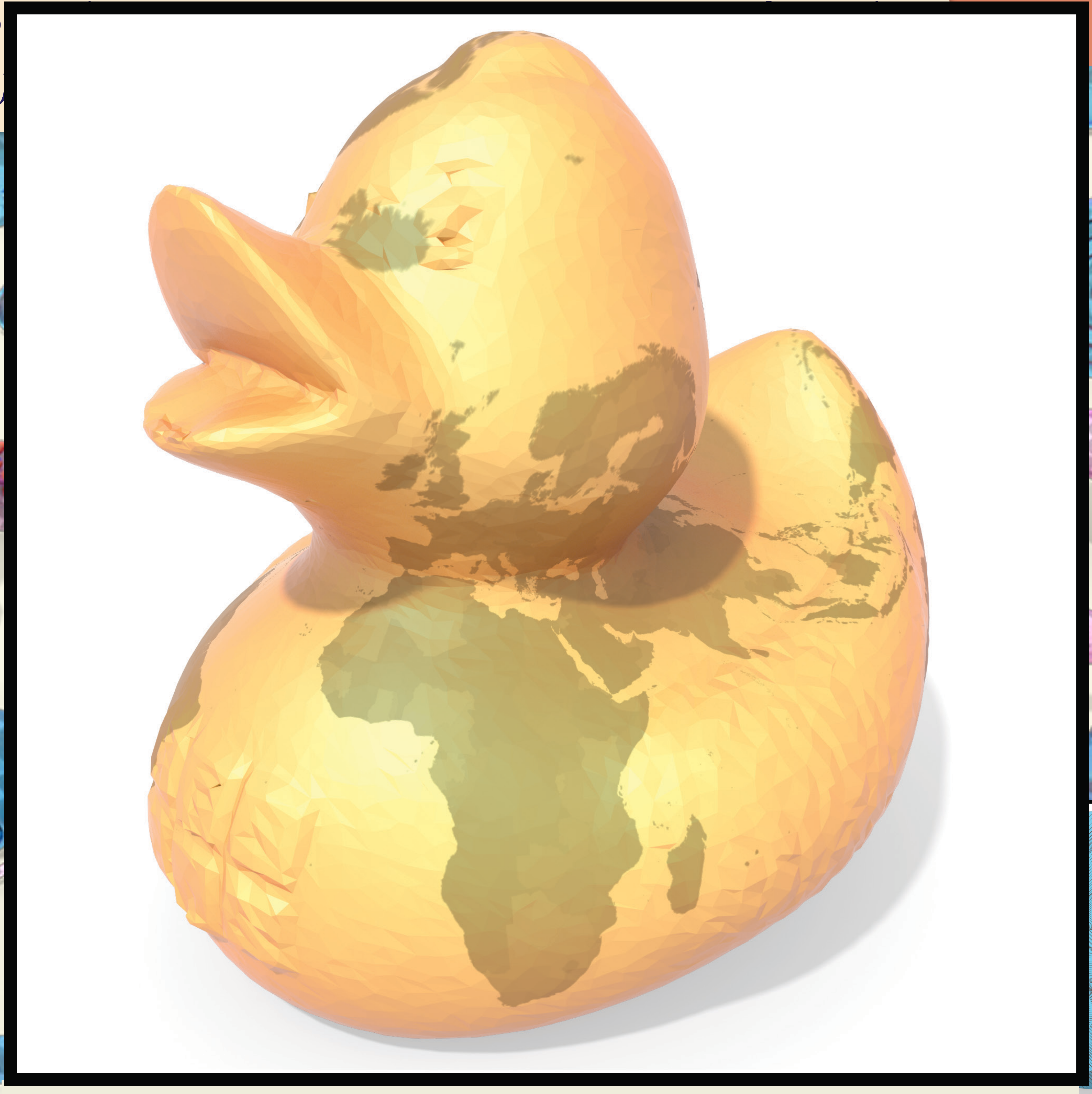


difficult constraints

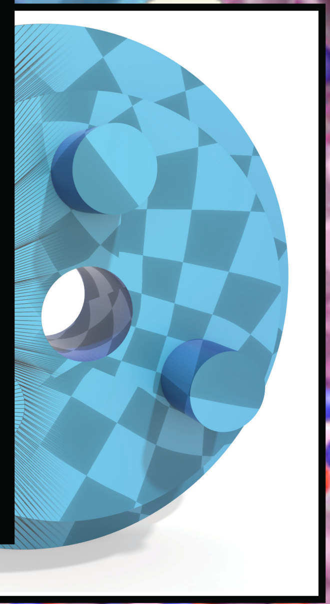
Robust surface parameterization

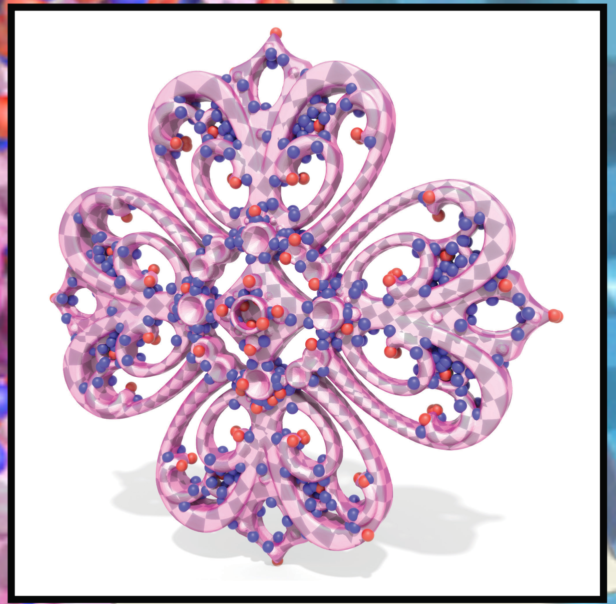
[**Gillespie**, Sp

Equivalence o



maps to the sphere



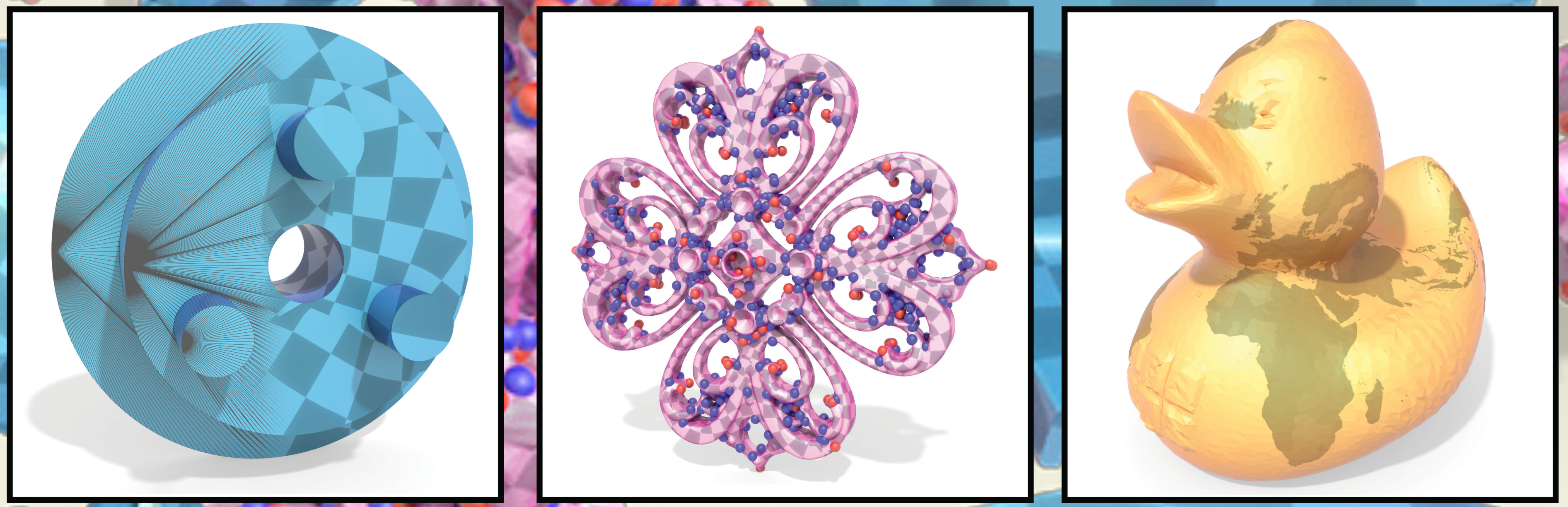


Robust surface parameterization

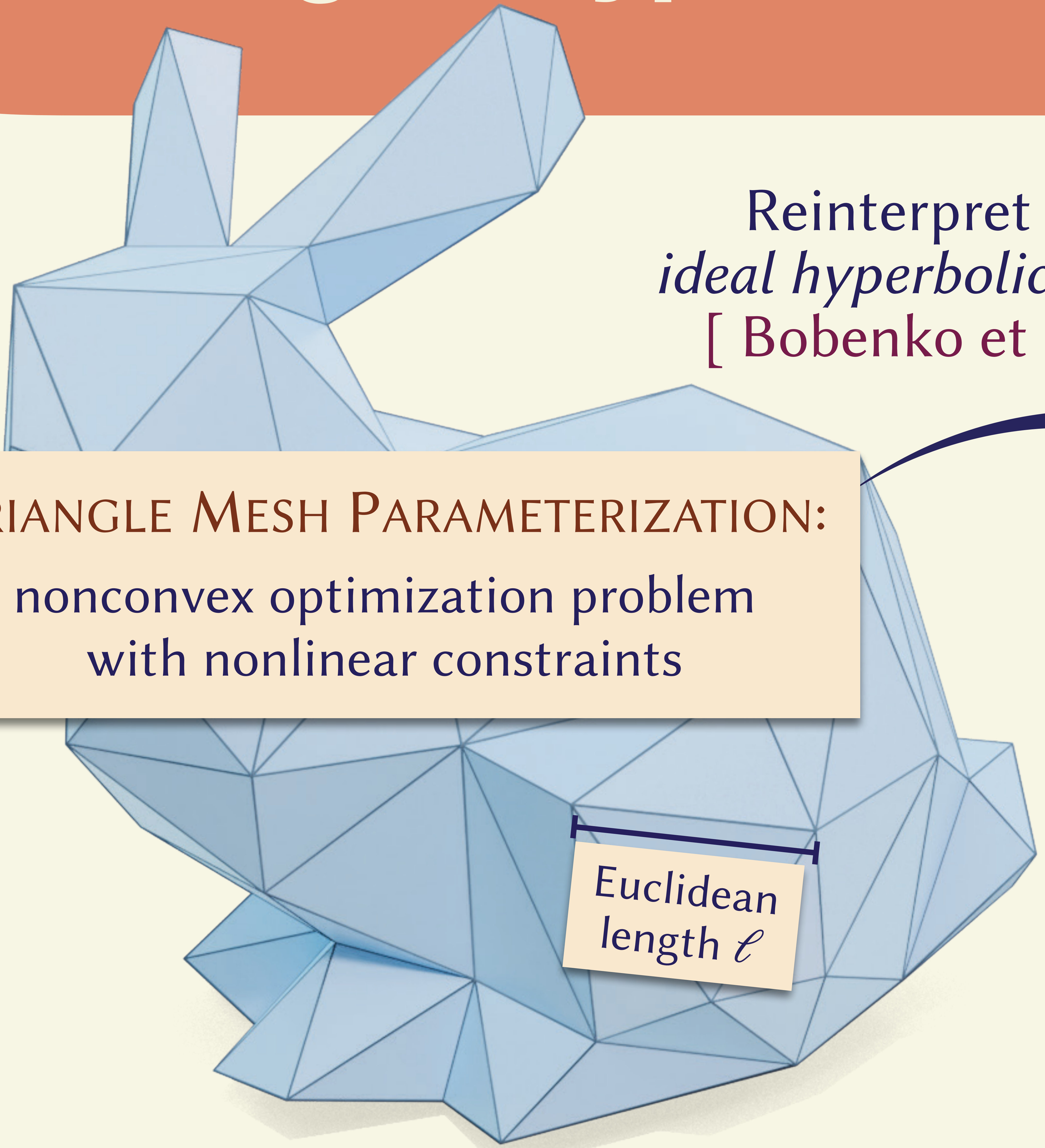
[**Gillespie**, Springborn, & Crane. 2021. *Discrete Conformal Equivalence of Polyhedral Surfaces.* ACM Trans. Graph.]

Correctness guarantee:
parameterizations are locally injective

Quality guarantee:
parameterizations are discrete conformal maps

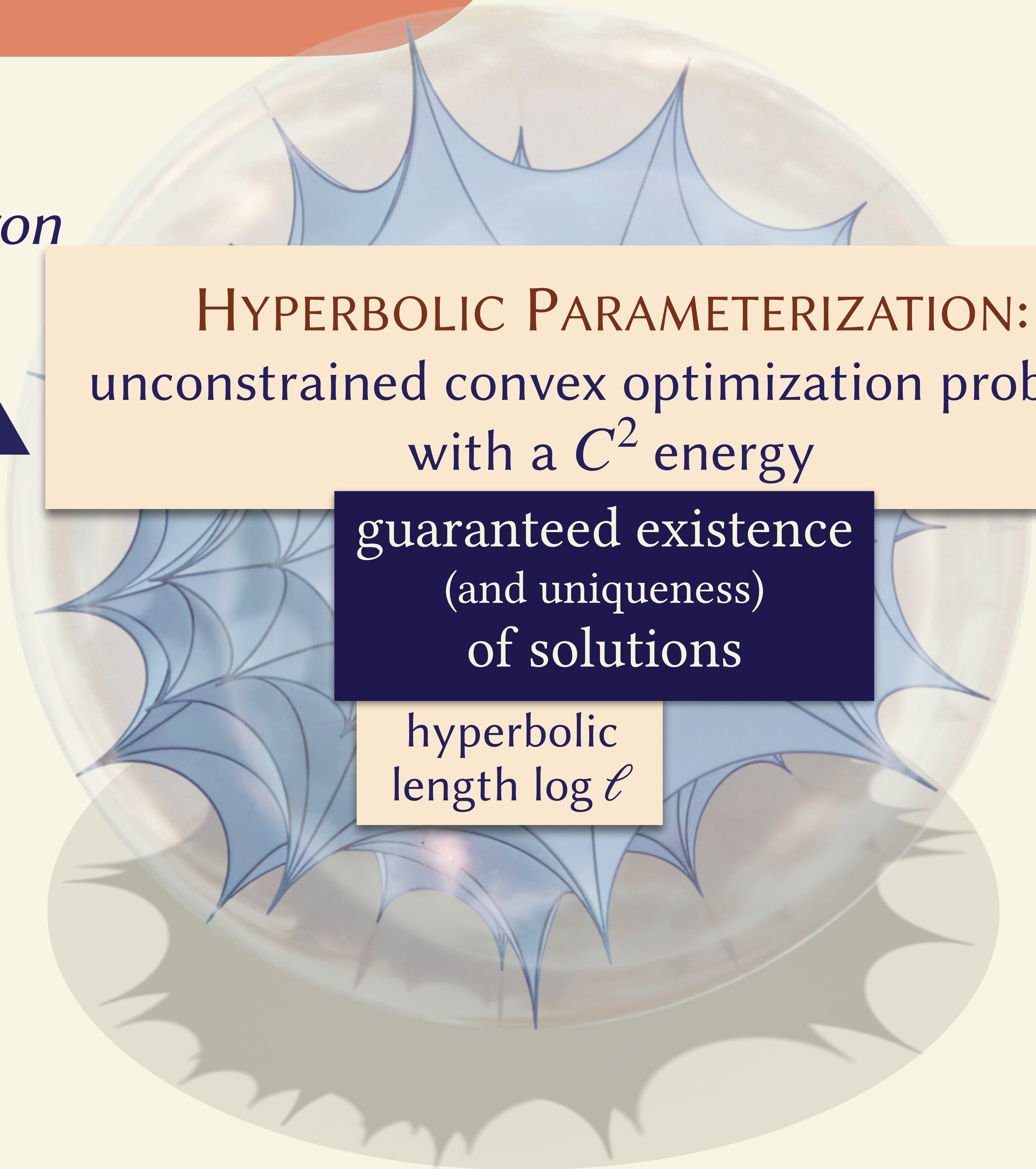


Turning to hyperbolic geometry



TRIANGLE MESH PARAMETERIZATION:
nonconvex optimization problem
with nonlinear constraints

Reinterpret input as
ideal hyperbolic polyhedron
[Bobenko et al. 2010]

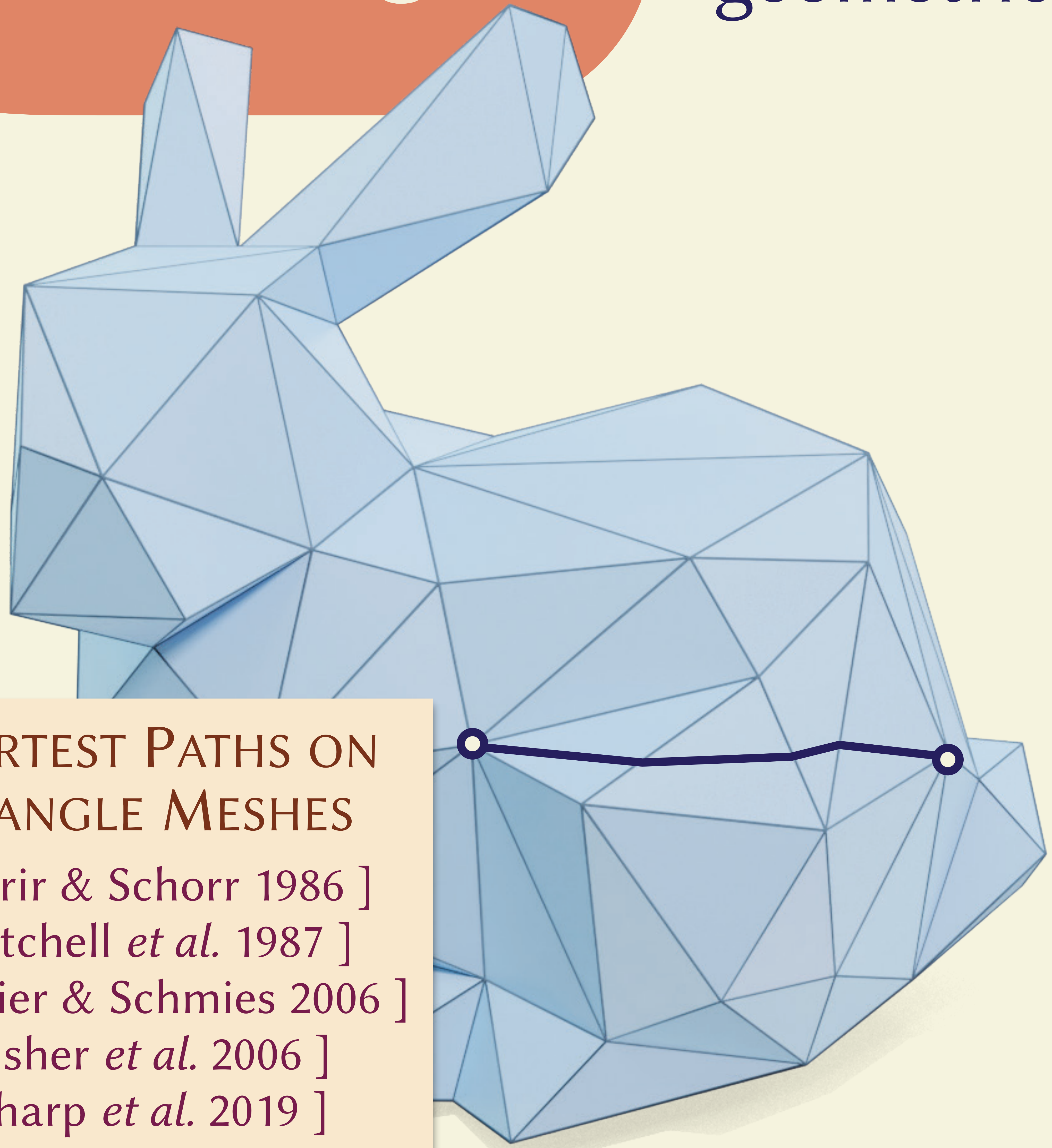


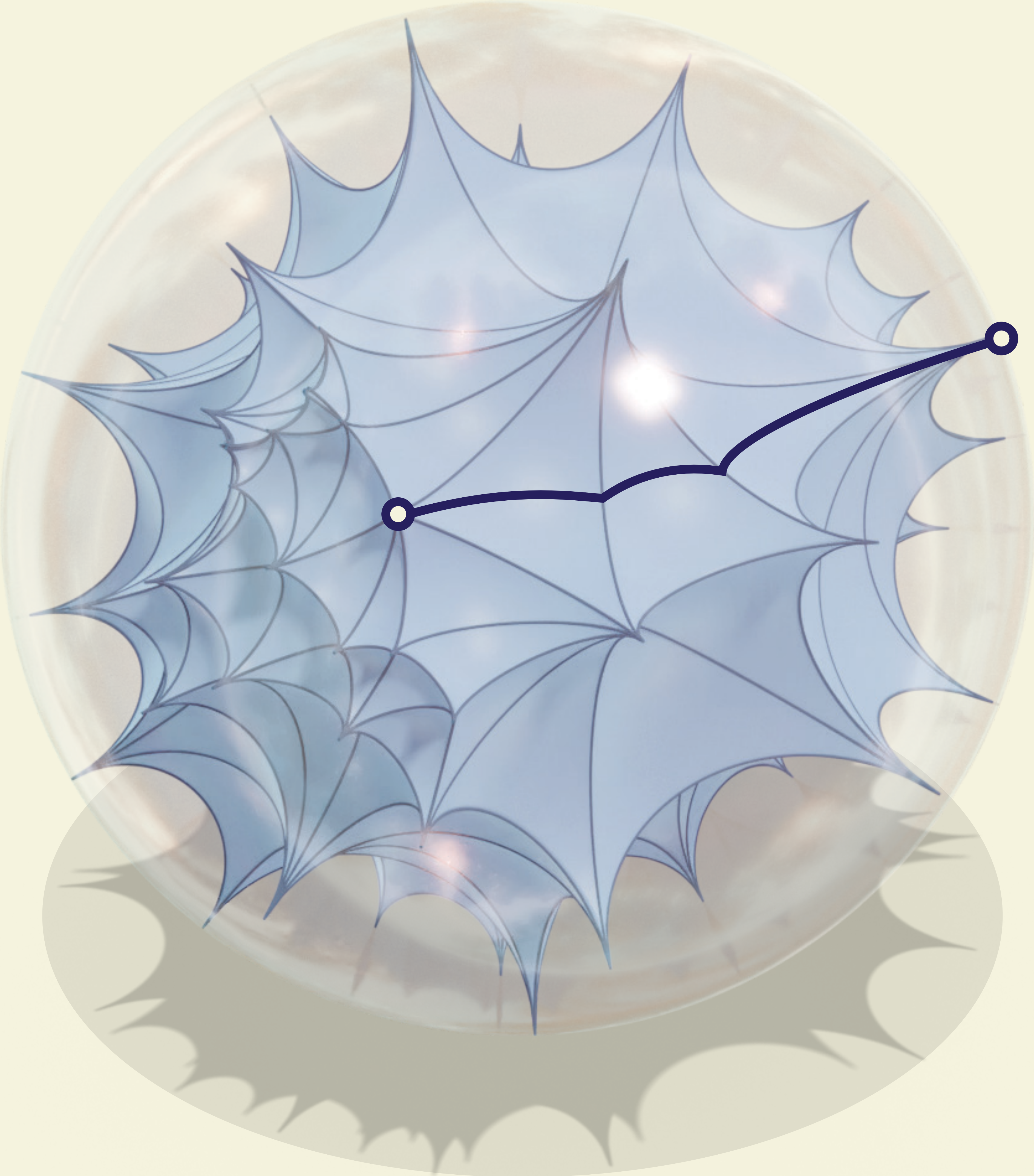
HYPERBOLIC PARAMETERIZATION:
unconstrained convex optimization problem
with a C^2 energy

guaranteed existence
(and uniqueness)
of solutions

Challenge

new algorithms & data structures for geometric calculations on hyperbolic polyhedra



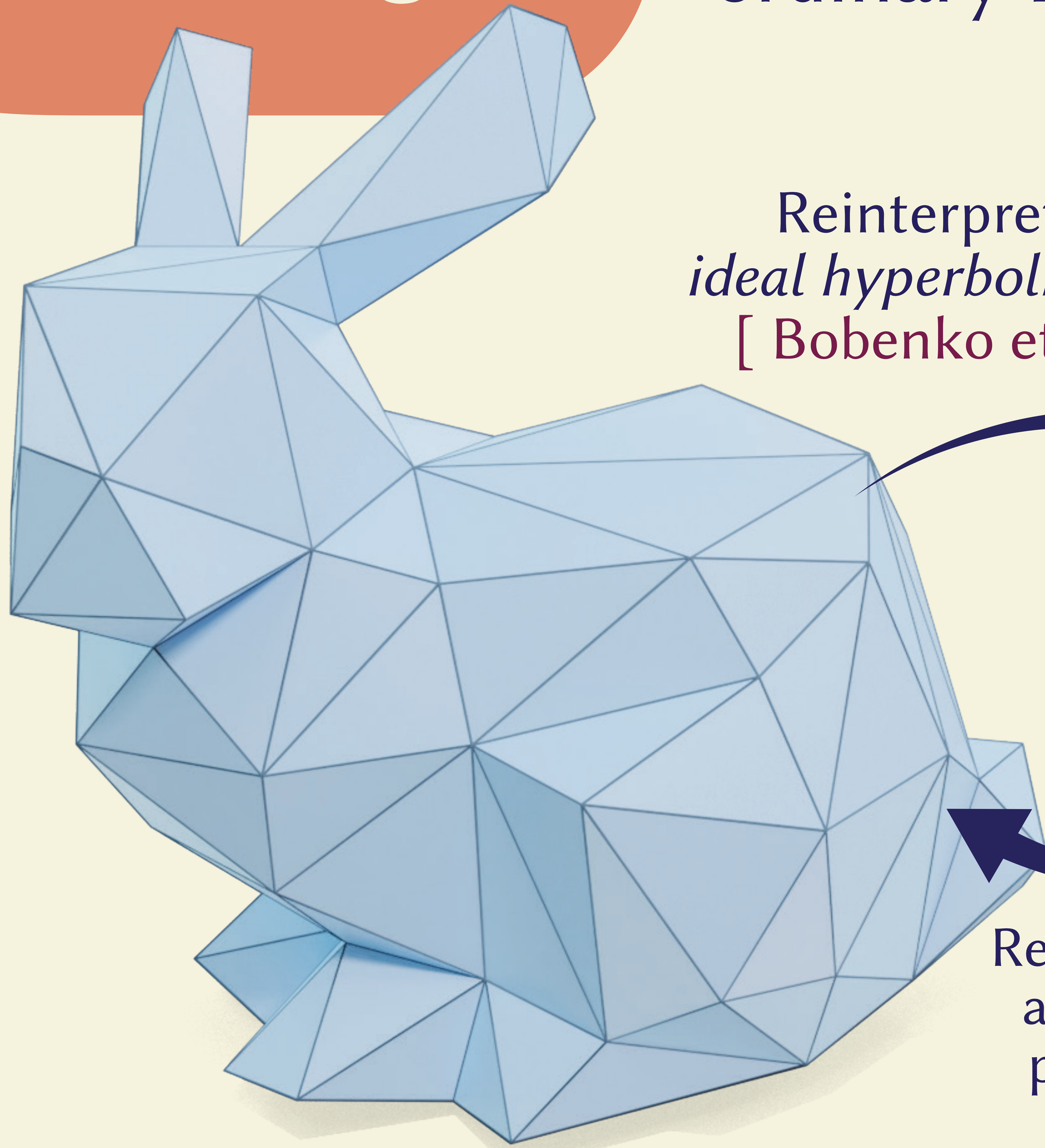


SHORTEST PATHS ON TRIANGLE MESHES

[Sharir & Schorr 1986]
[Mitchell *et al.* 1987]
[Polthier & Schmies 2006]
[Fisher *et al.* 2006]
[Sharp *et al.* 2019]

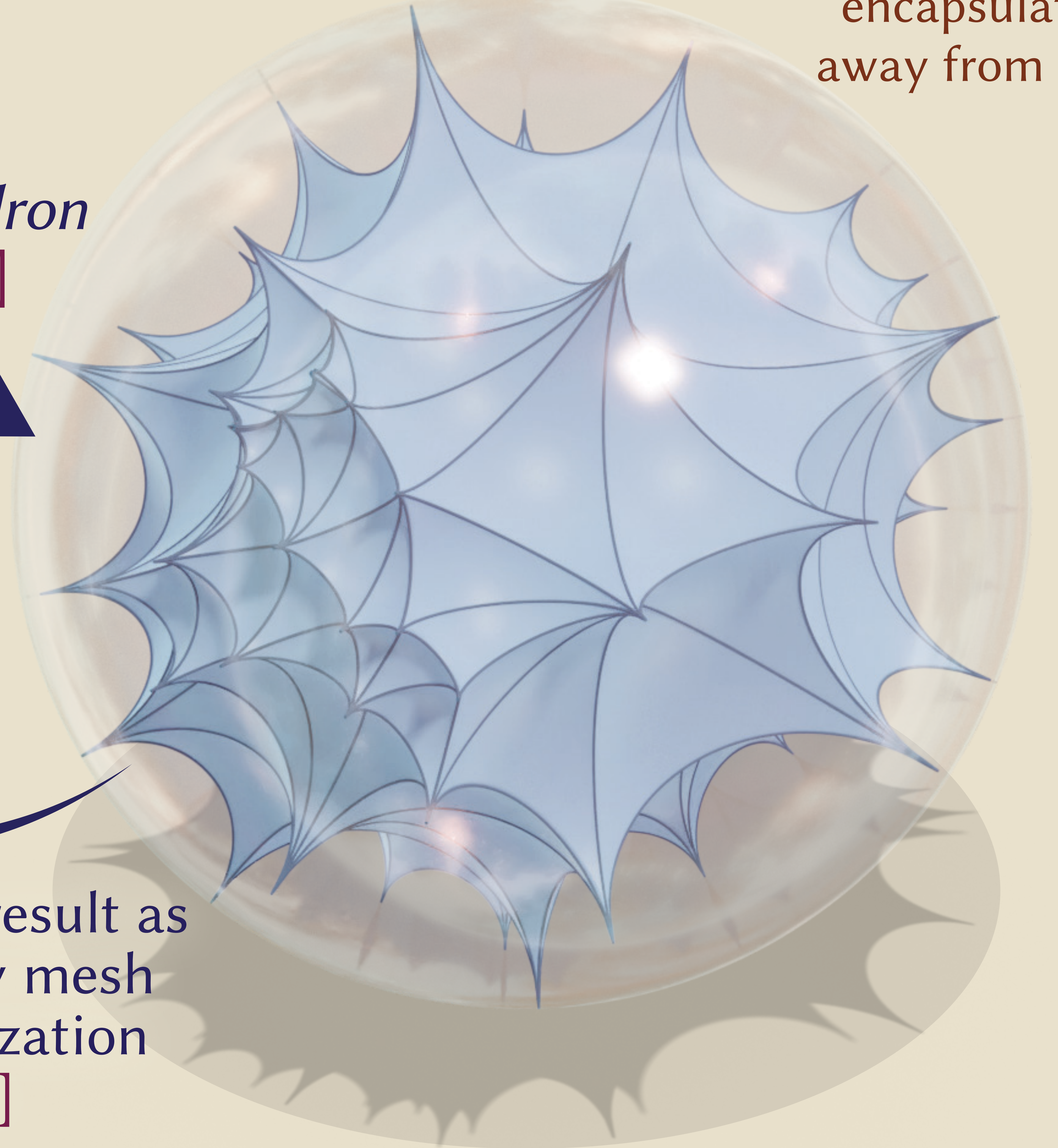
Challenge

convert hyperbolic result back into ordinary Euclidean geometry



Reinterpret input as
ideal hyperbolic polyhedron
[Bobenko et al. 2010]

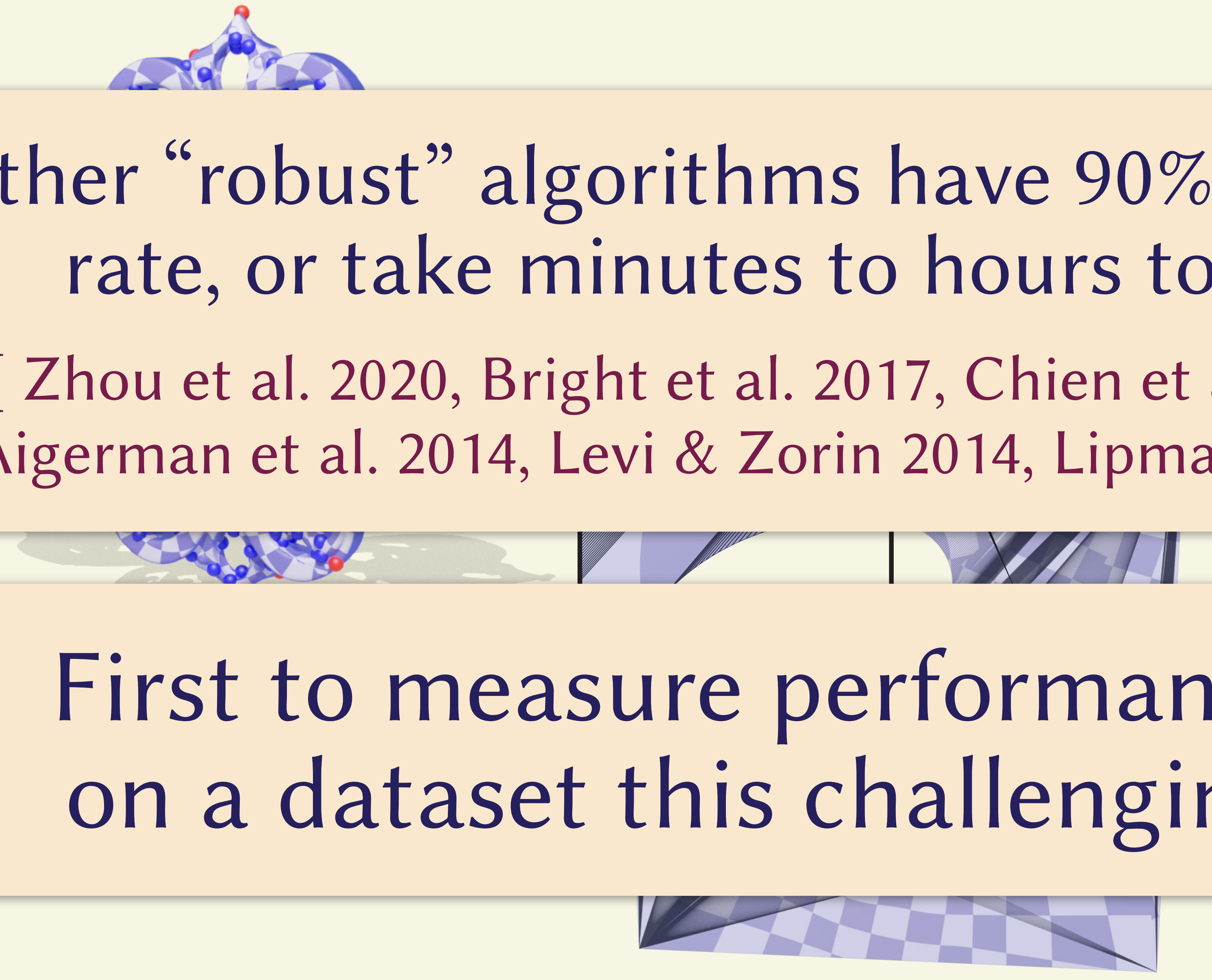
encapsulated
away from user



Reinterpret result as
an ordinary mesh
parameterization
[ours]

Success on difficult datasets

difficult cones



Other “robust” algorithms have 90% success rate, or take minutes to hours to run

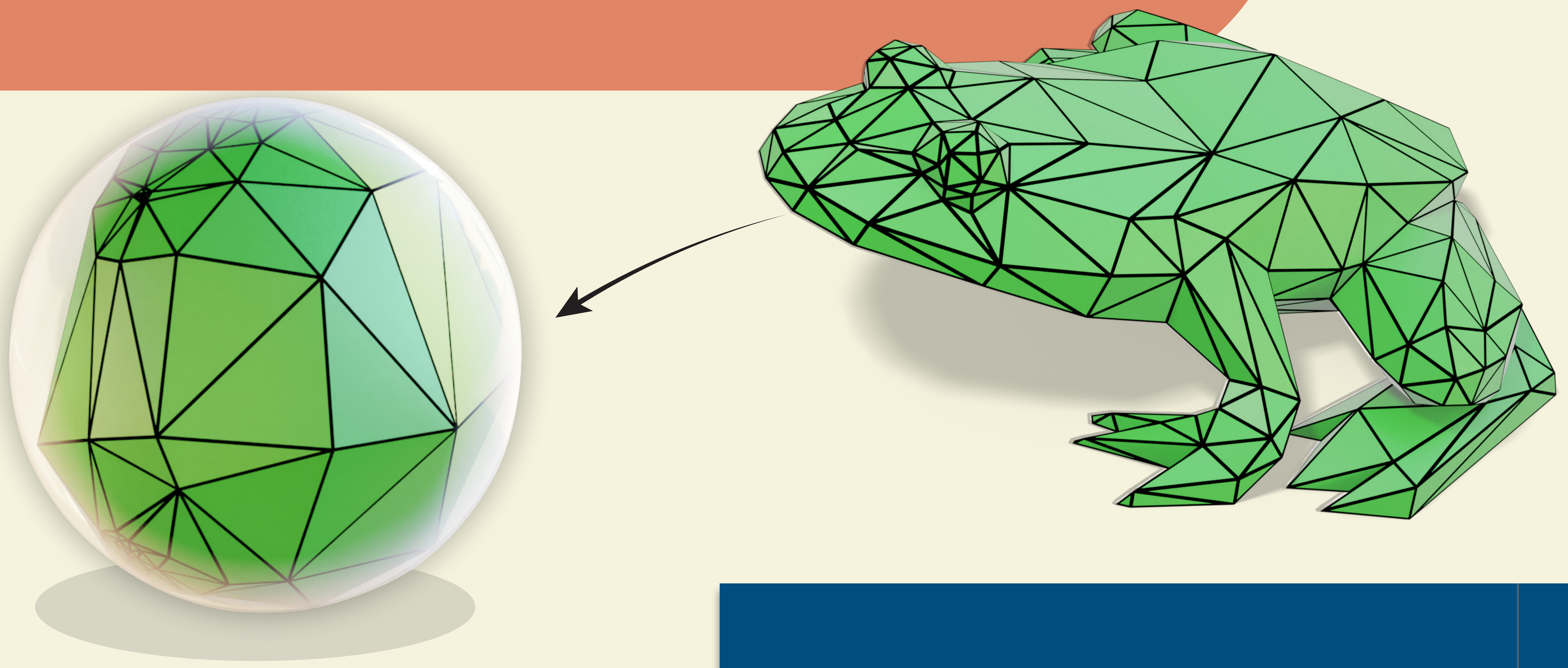
[Zhou et al. 2020, Bright et al. 2017, Chien et al. 2017, Aigerman et al. 2014, Levi & Zorin 2014, Lipman 2012]

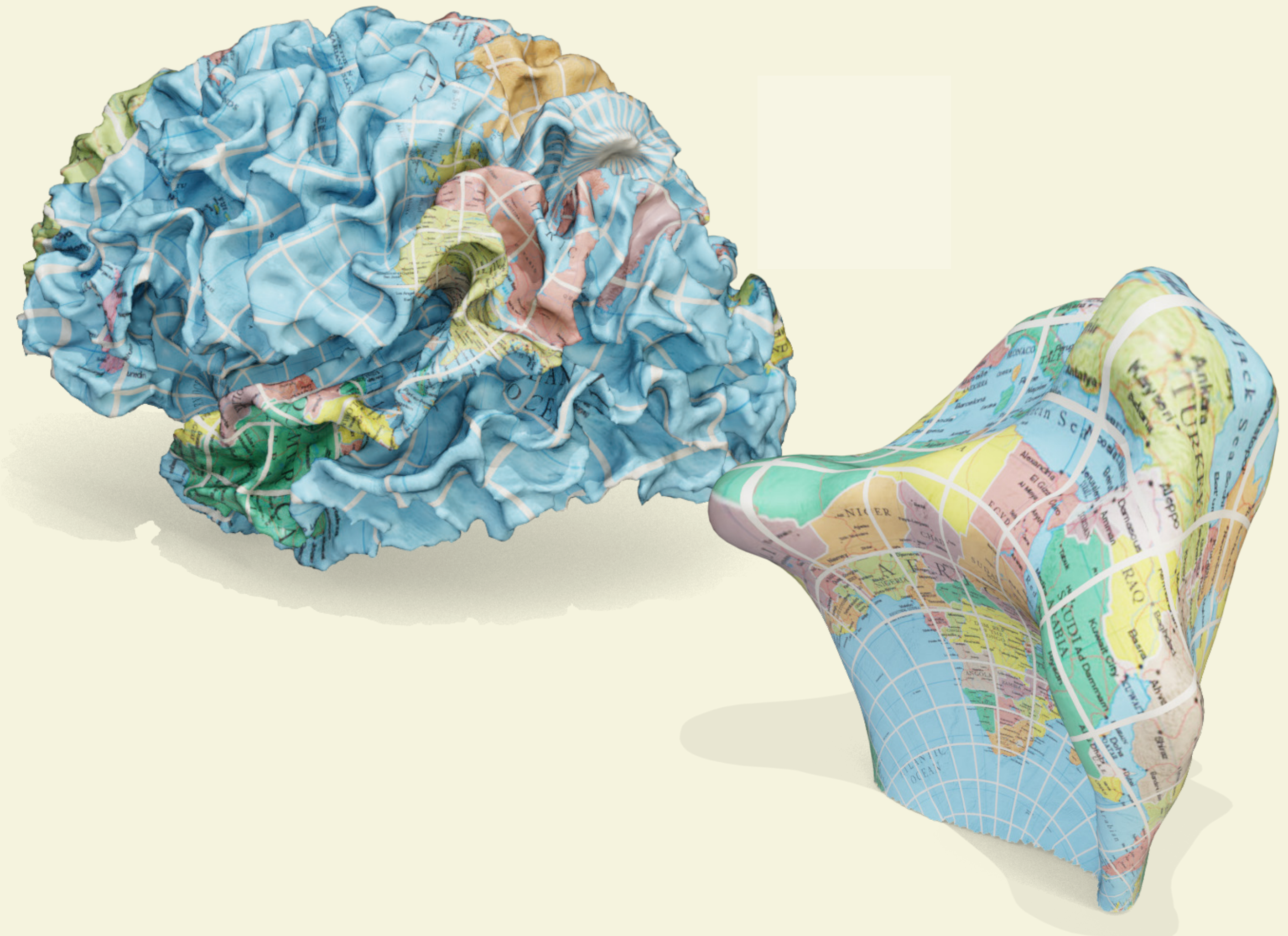
First to measure performance on a dataset this challenging

Dataset	# Models	Success rate	Average time
MPZ Myles *et al.* 2014]	114	100%	8s
Thingi10k [Zhou *et al.* 2016]	32,744	97.7%	57s

Success = produced a locally-injective parameterization

Maps to the sphere





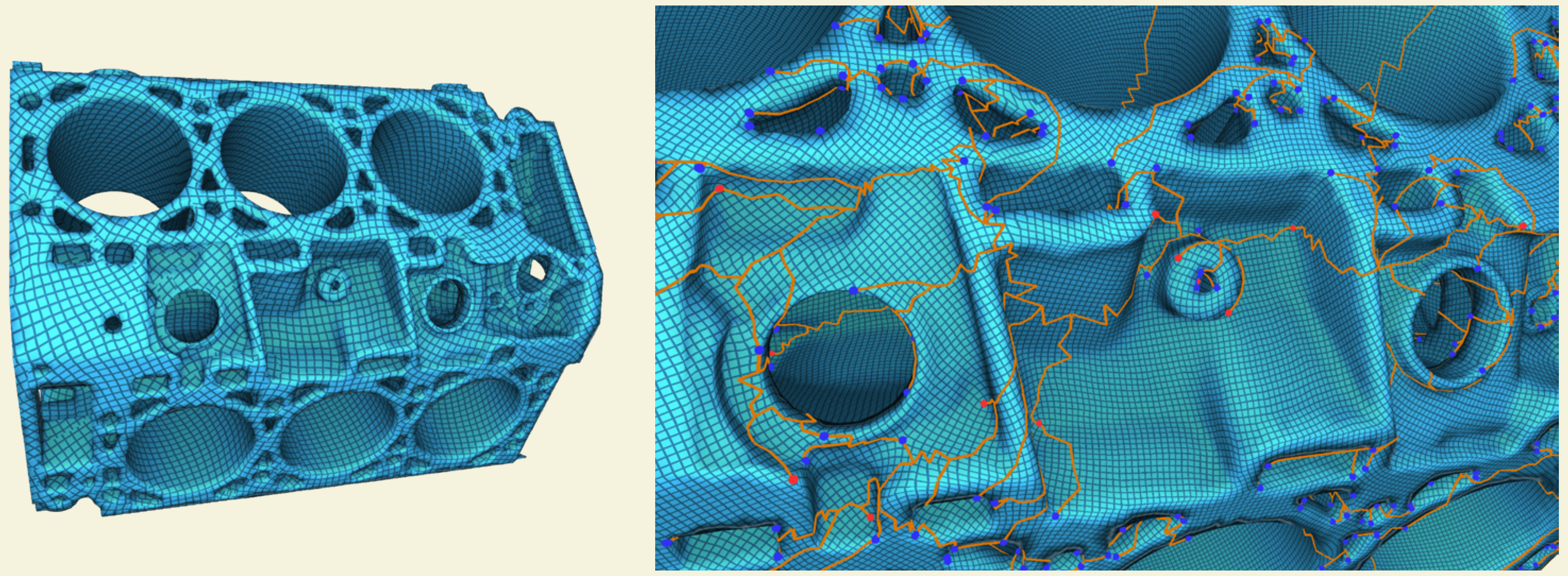
Dataset	# Models	Success rate	Average time
brain scans [Yeo *et al.* 2009]	78	100%	493s
anatomical surfaces [Boyer et al. 2011]	187	100%	15s

Uptake

adopted in a variety of areas

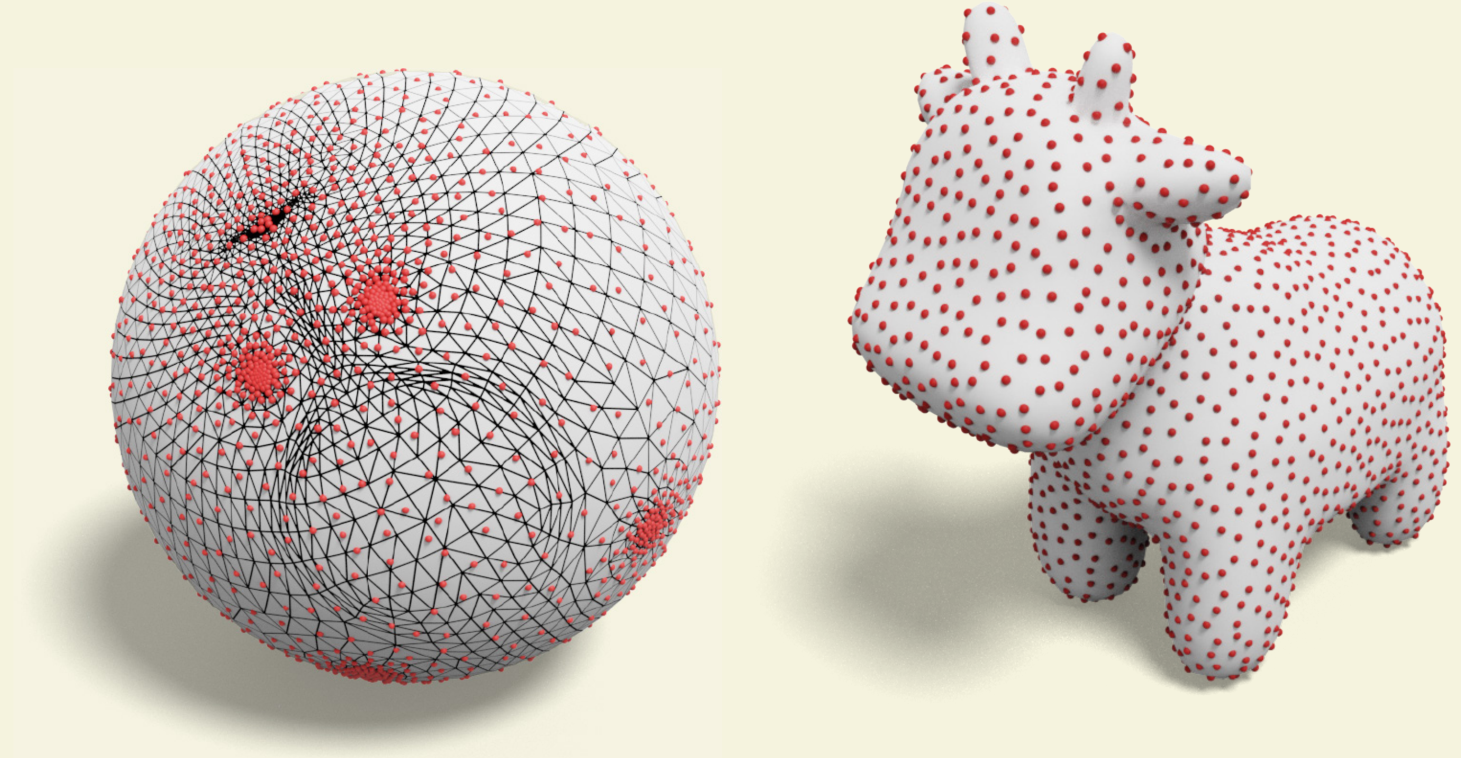
3D printing

[Lenihan *et al.* 2023]



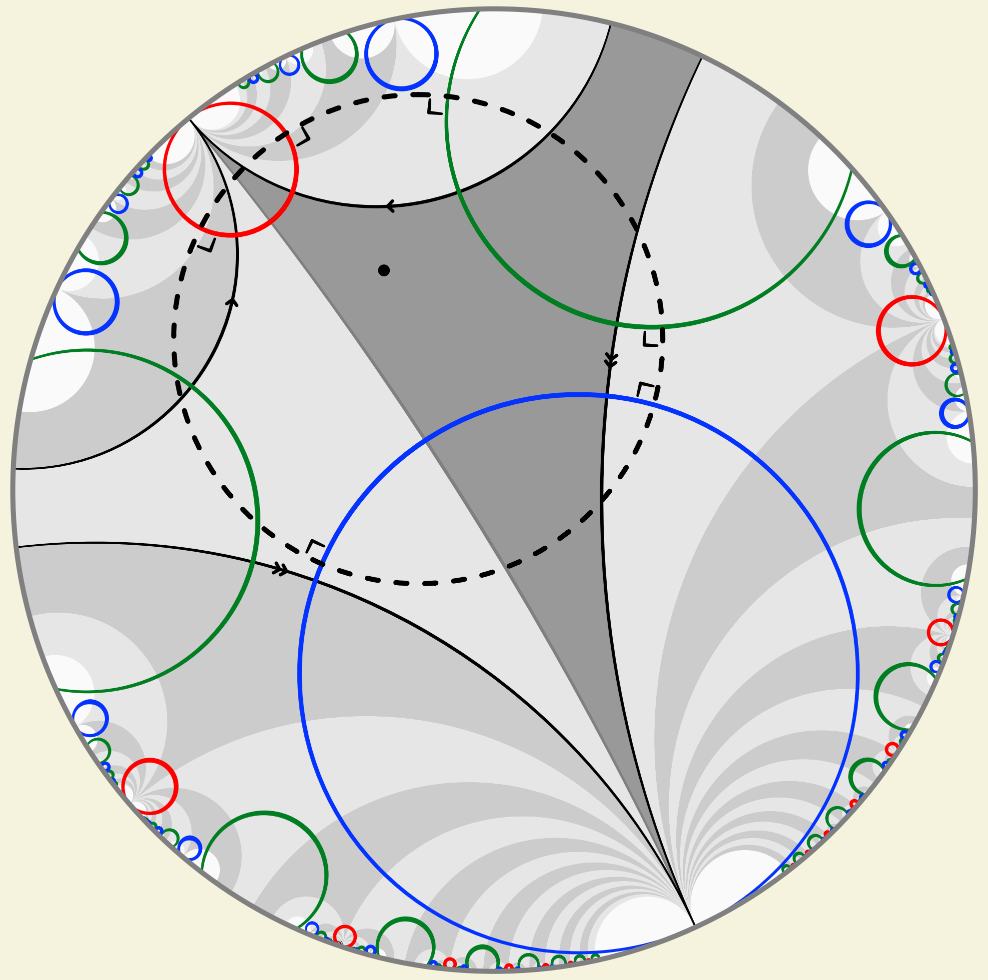
geometry processing

[Capouellez & Zorin 2023, 2024]



probability distributions on surfaces

[Genest *et al.* 2024]



geometric topology

[Bobenko & Lutz 2023]

generative models

(ongoing)

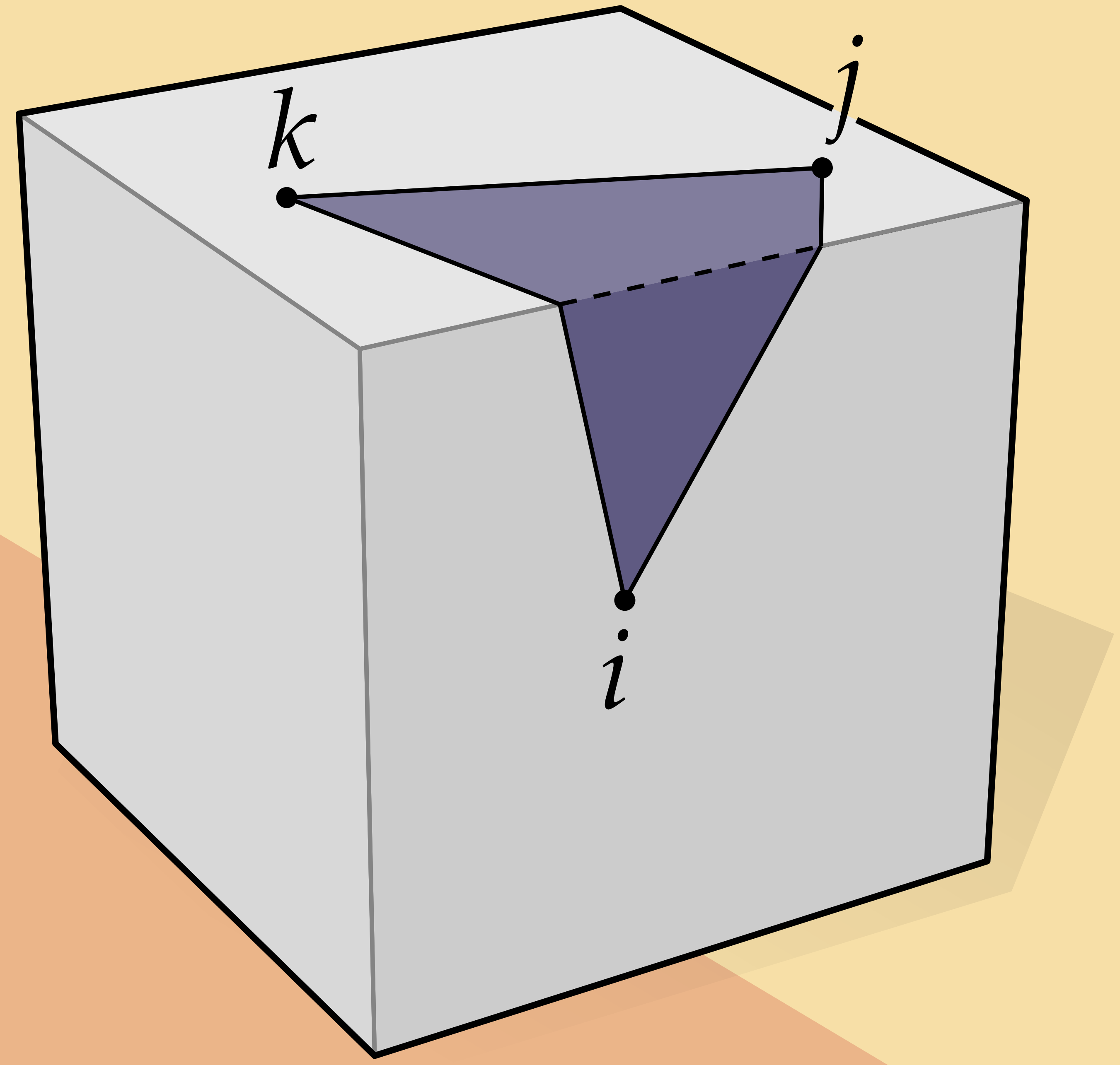
Robust Geometry Processing

[**Gillespie**, Sharp, & Crane. 2021. *Integer Coordinates for Intrinsic Geometry Processing.* ACM Trans. Graph.]

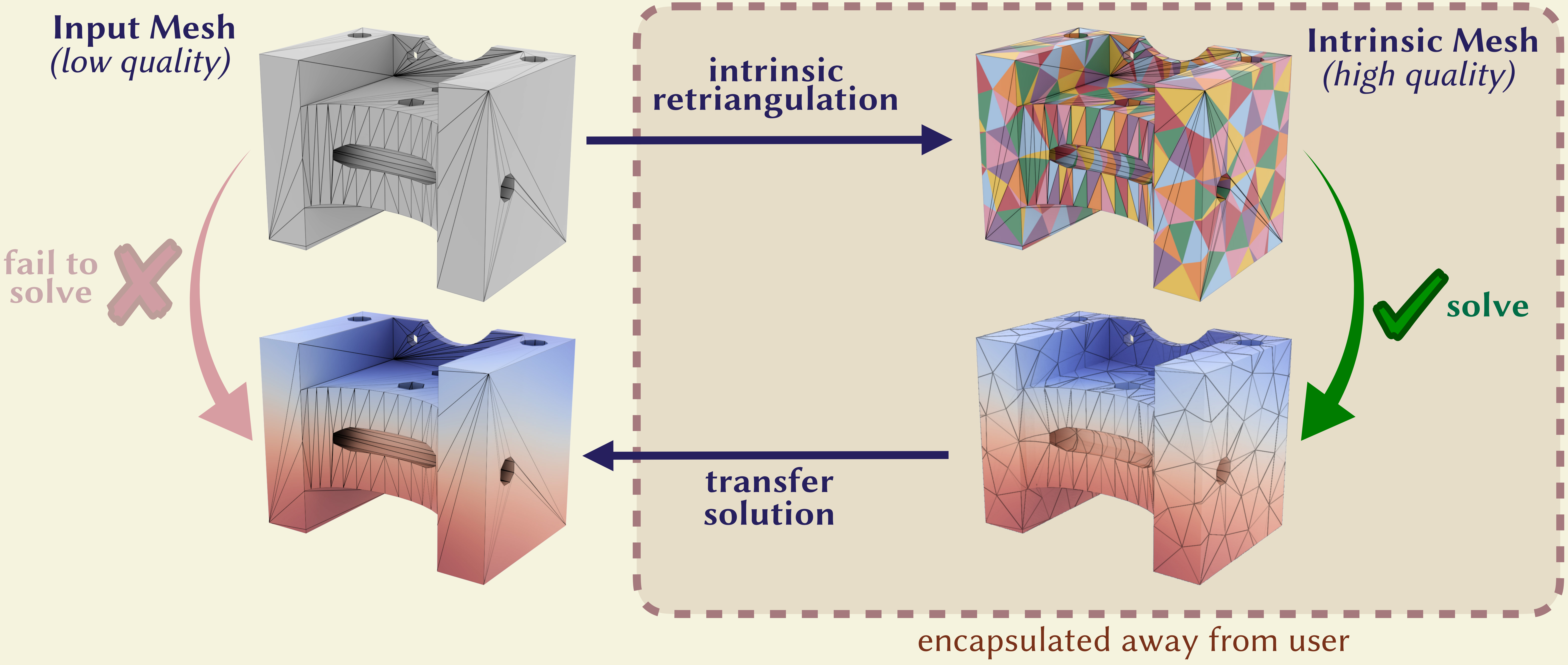
[Liu, **Gillespie**, Chislett, Sharp, Jacobson & Crane. 2023. *Surface Simplification using Intrinsic Error Metrics.* ACM Trans. Graph.]

[Sharp, **Gillespie**, & Crane. *Geometry Processing with Intrinsic Triangulations.* ACM SIGGRAPH 2021 Courses]

[Sharp, **Gillespie**, & Crane. *Geometry Processing with Intrinsic Triangulations.* SIAM IMR 2021 Courses]



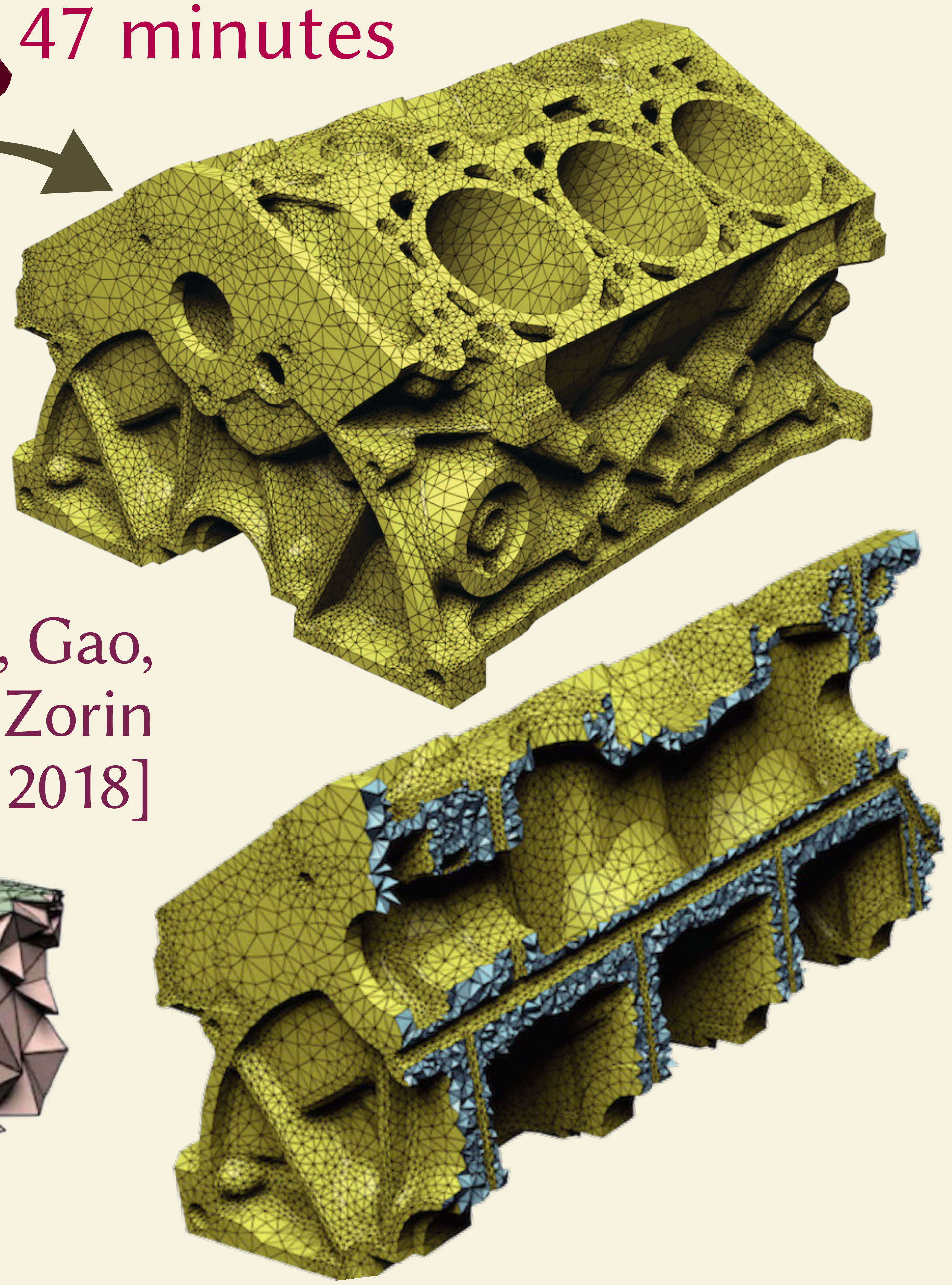
Working with low-quality triangle meshes



Status quo: remeshing

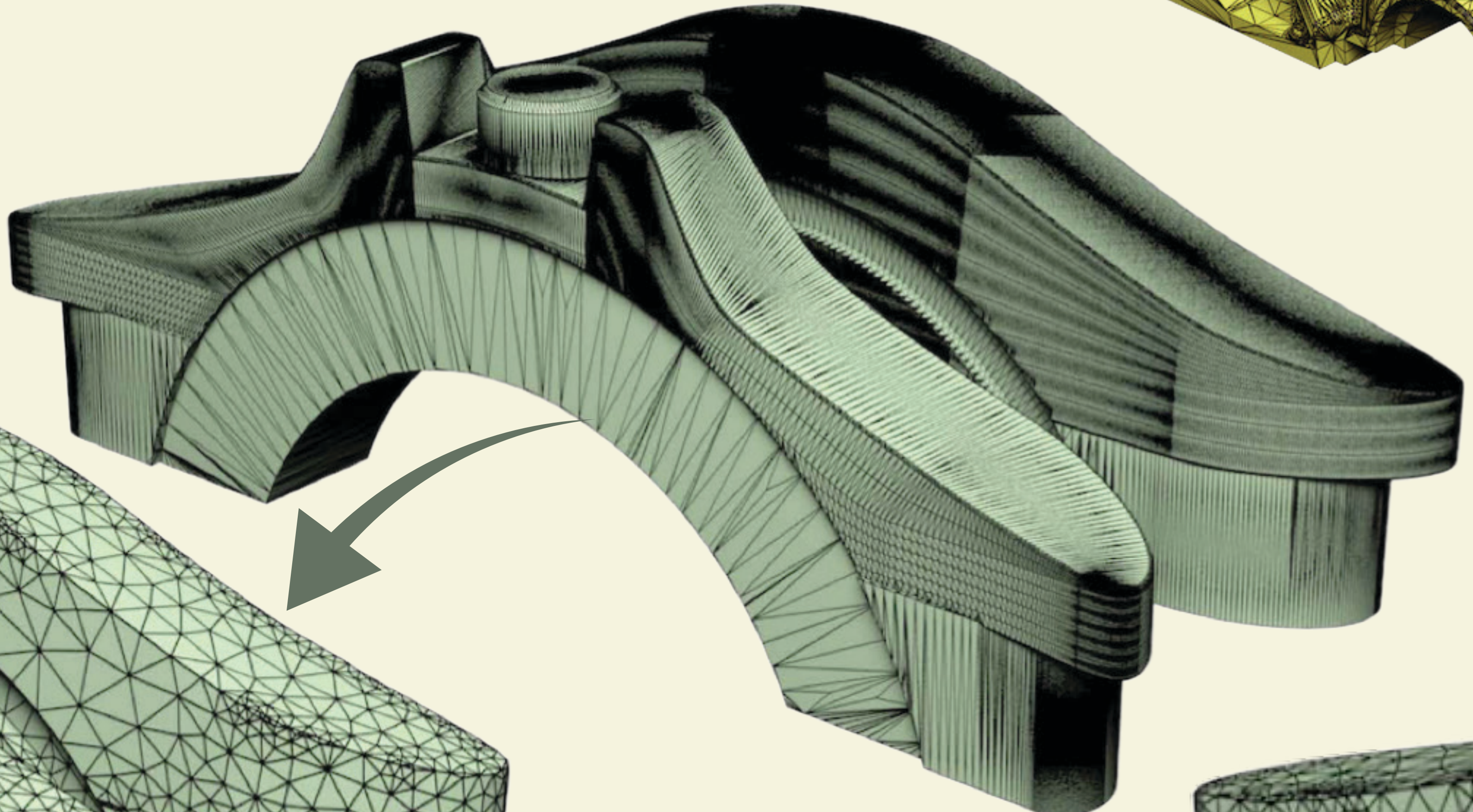
- State-of-the-art is robust but slow
  - Techniques work volumetrically

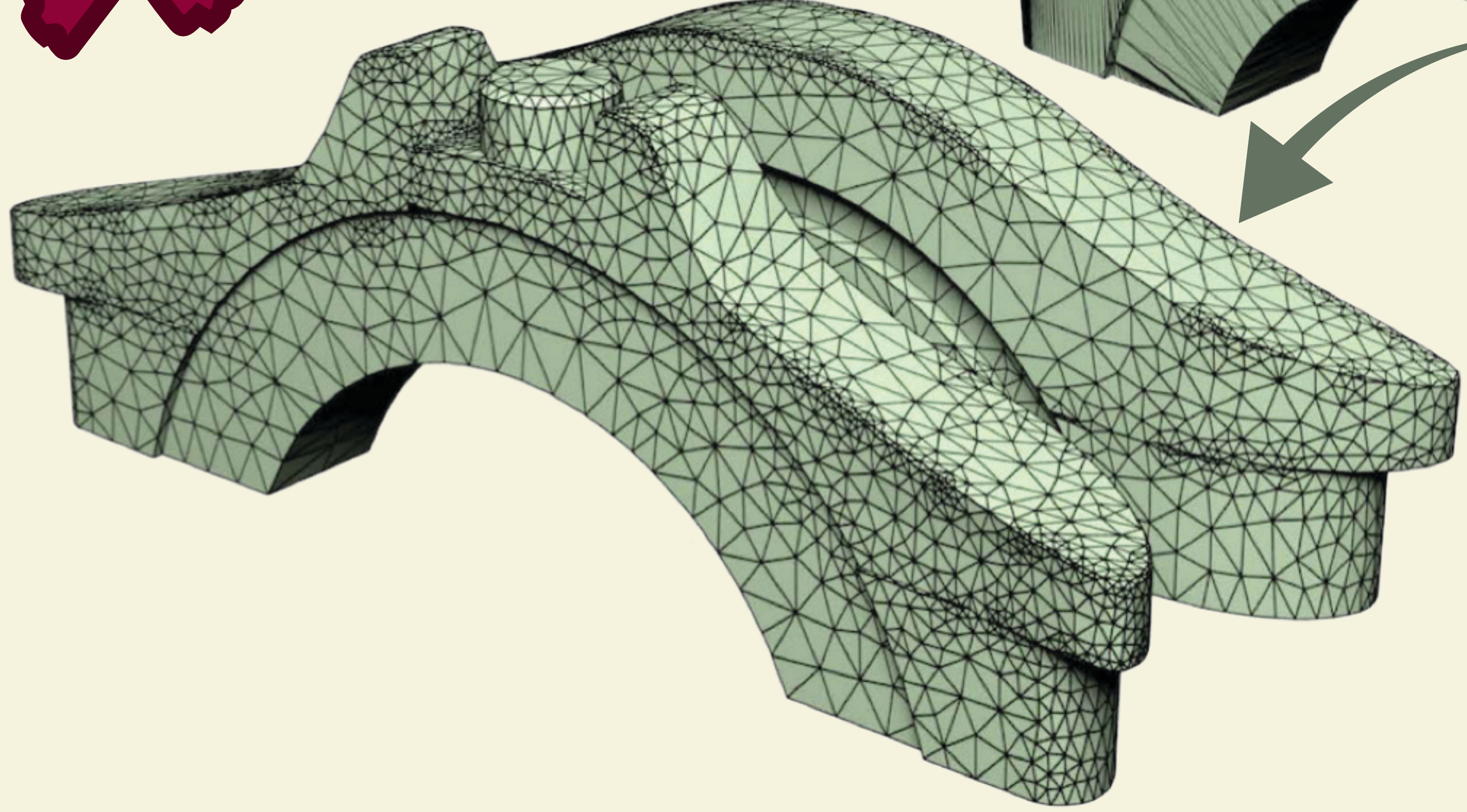
runtime:
47 minutes



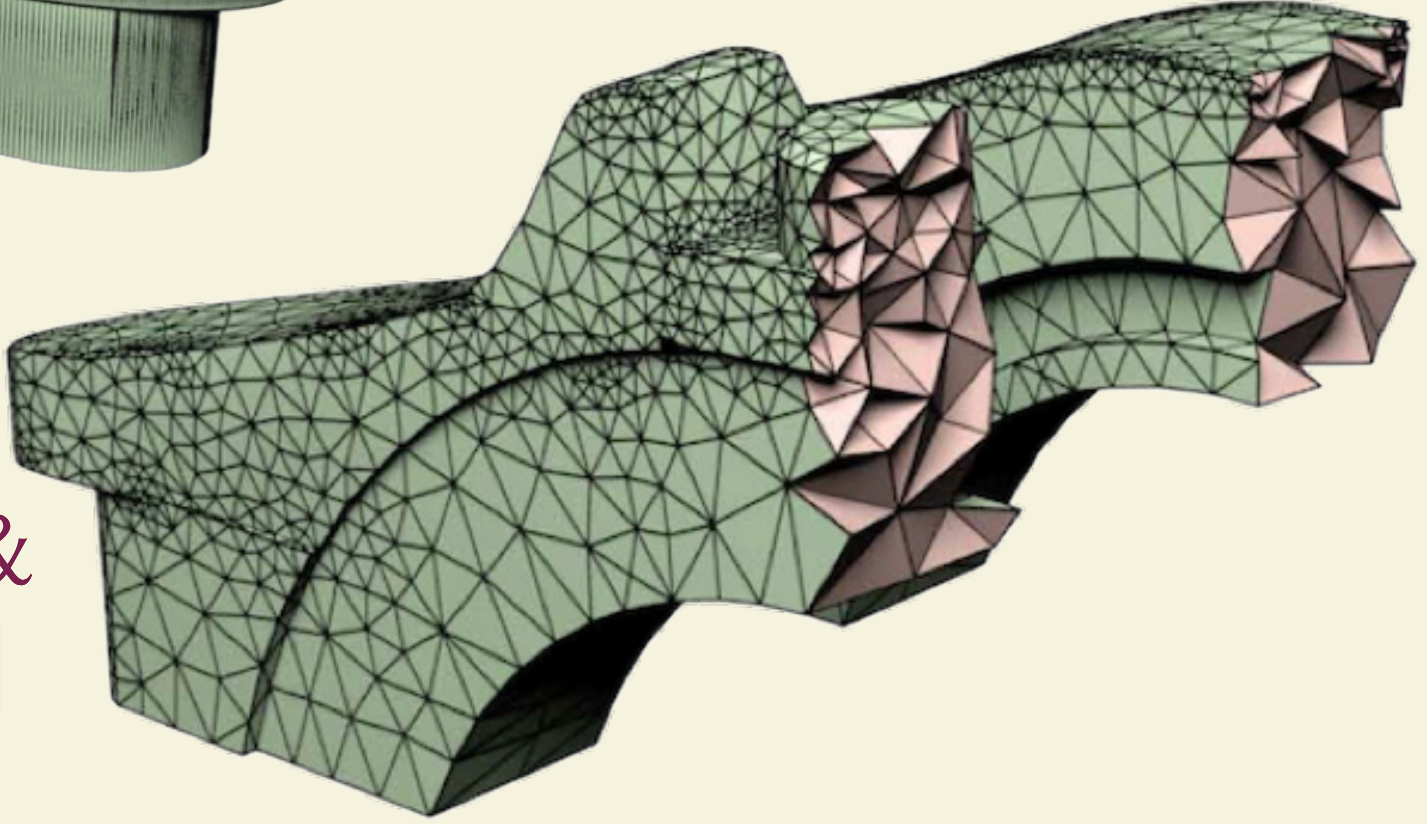
[Hu, Zhou, Gao, Jacobson, Zorin & Panozzo 2018]

runtime:
43 minutes





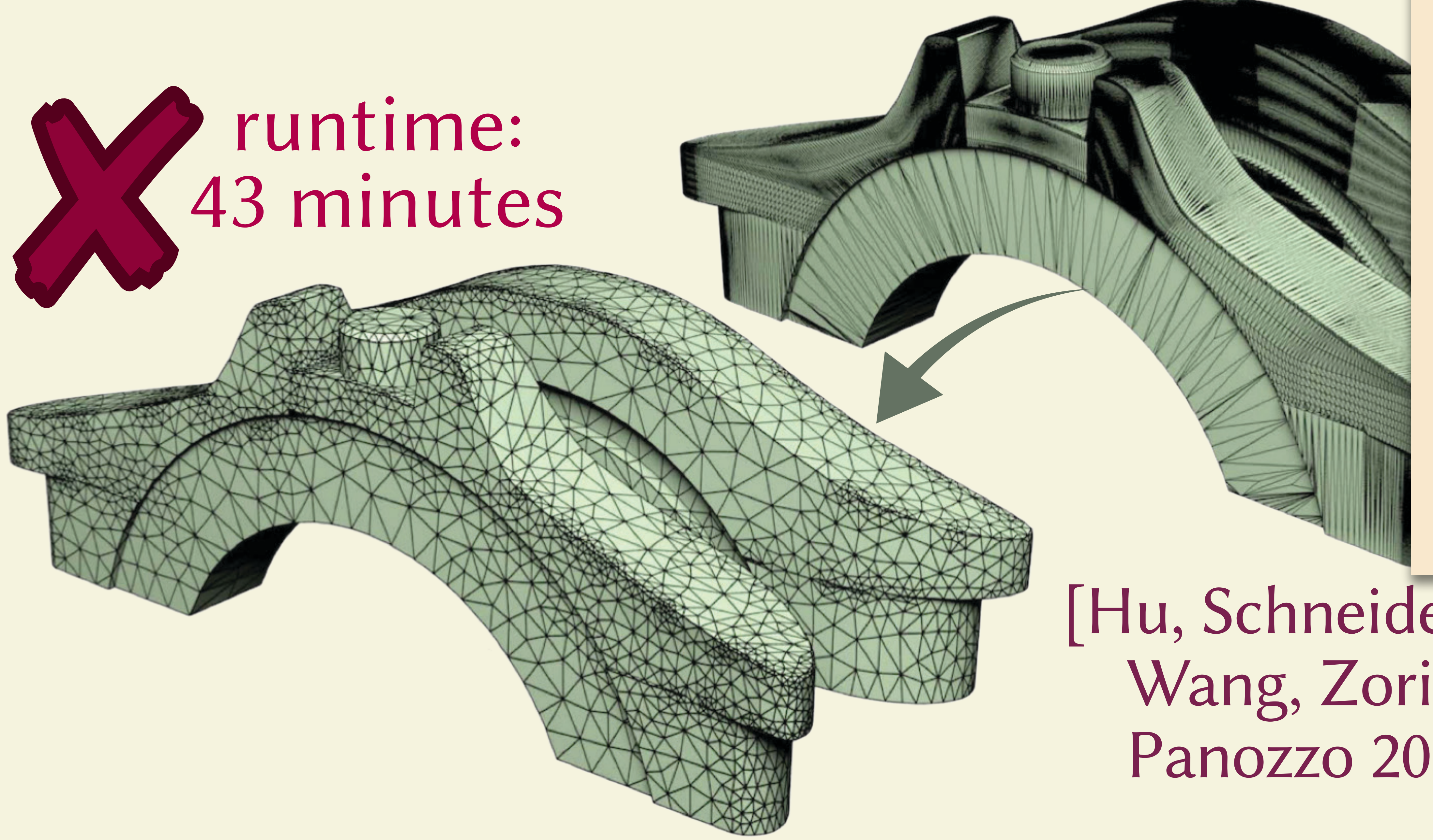
[Hu, Schneider, Wang, Zorin & Panozzo 2020]



Status quo: remeshing

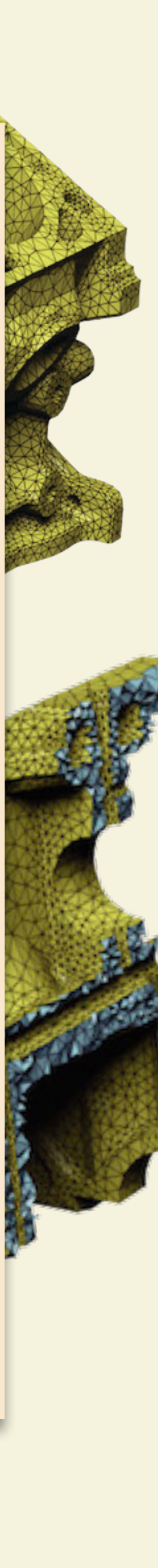
- State-of-the-art is robust but slow
    - Techniques work volumetrically

runtime:
43 minutes



[Hu, Schneider, Wang, Zorin & Panozzo 2020]

runtime:
47 minutes



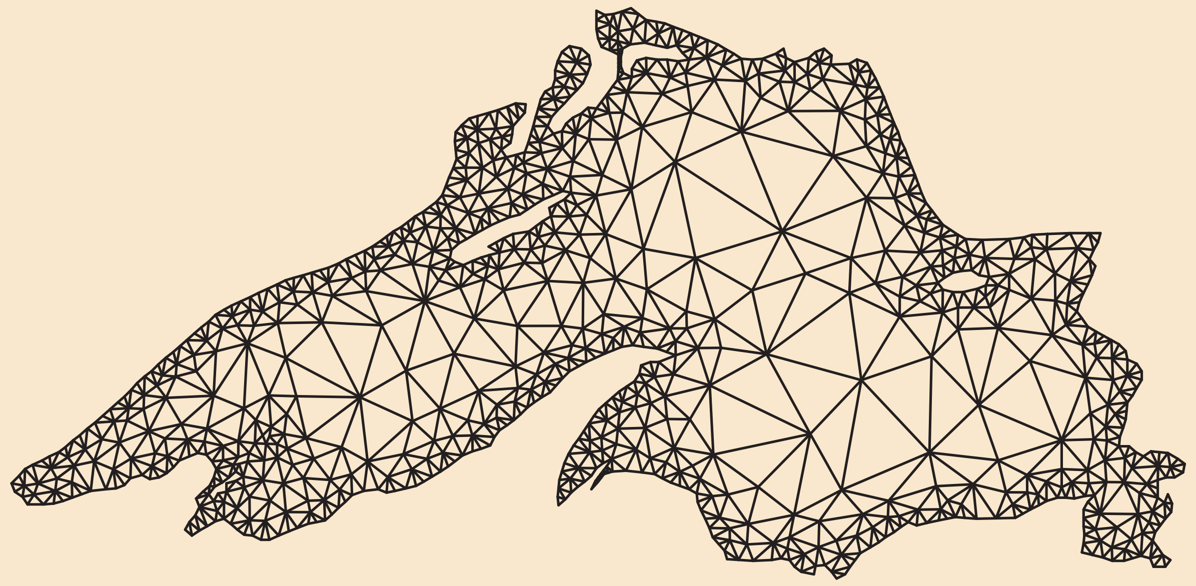
Meshing is much easier in 2D

strong guarantees on quality

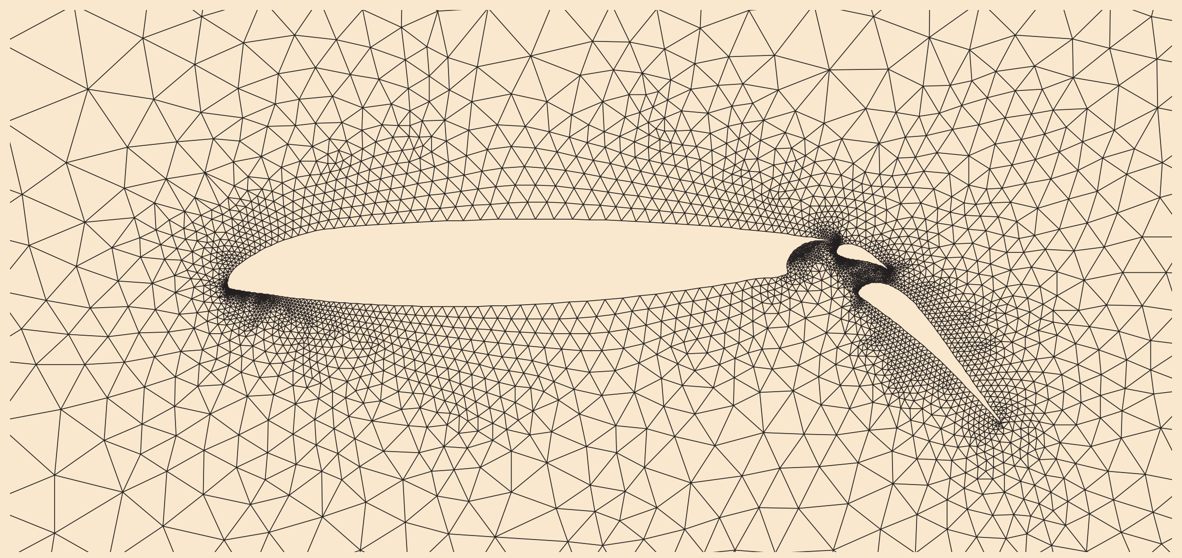
Generate high-quality meshes in milliseconds

via Delaunay refinement [Chew 1993; Shewchuk 1997]

[Shewchuk 1997]

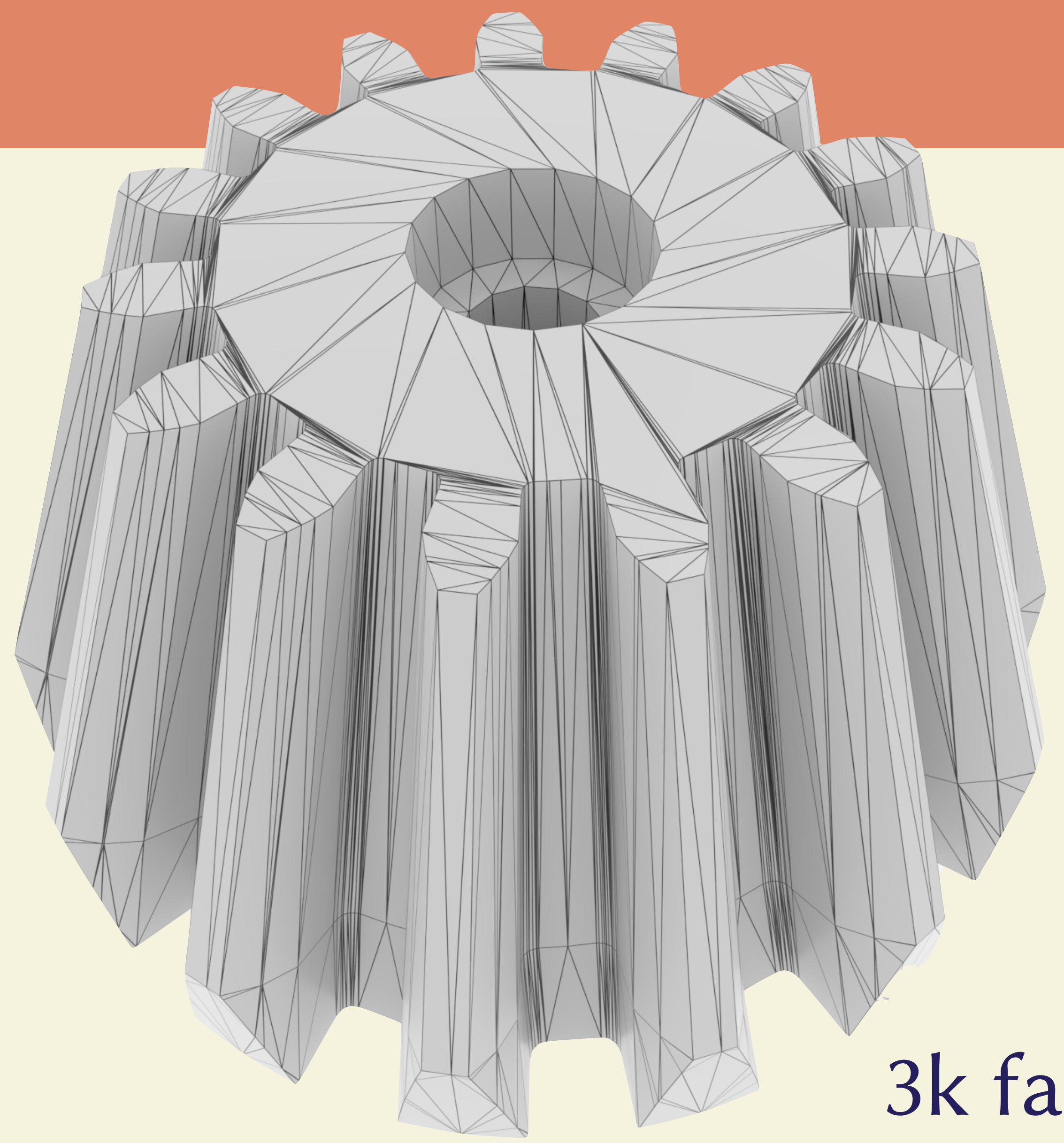


80 milliseconds

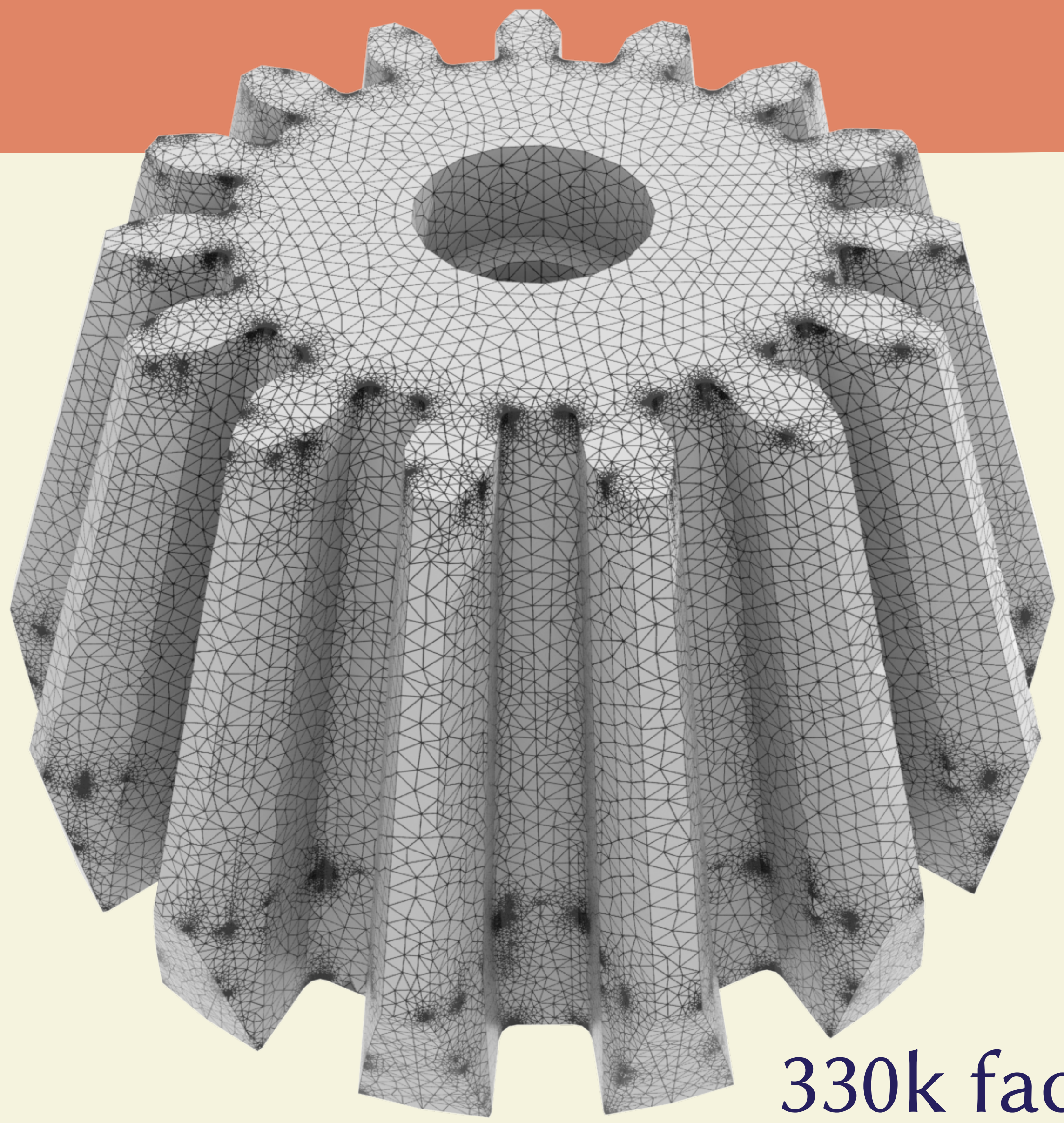


70 milliseconds

Trade offs of extrinsic remeshing



3k faces



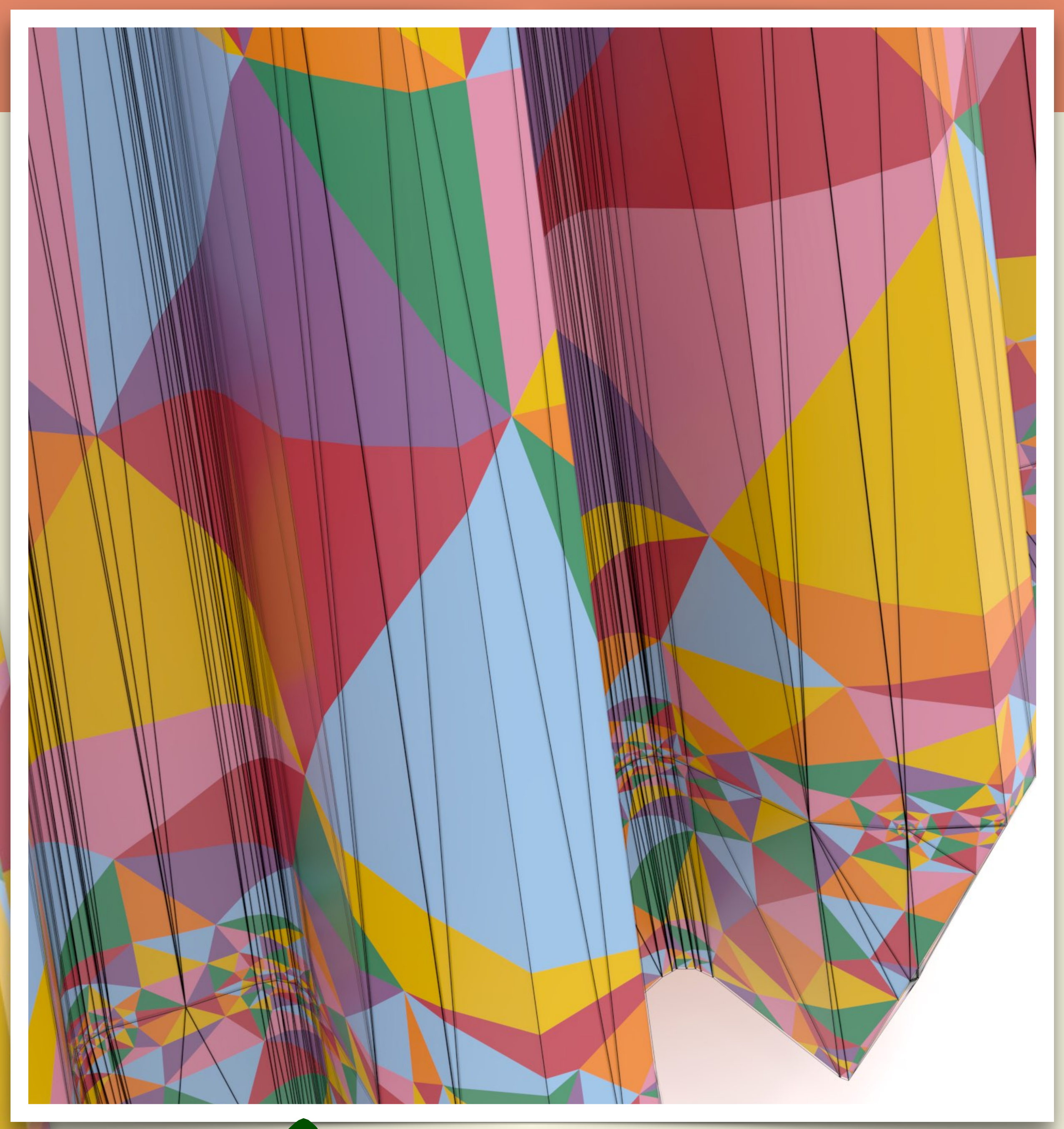
330k faces

3k faces

	3k faces	330k faces	3k faces
triangle quality	✗	✓	✓
mesh size	✓	✗	✓
geometric fidelity	✓	✓	✗

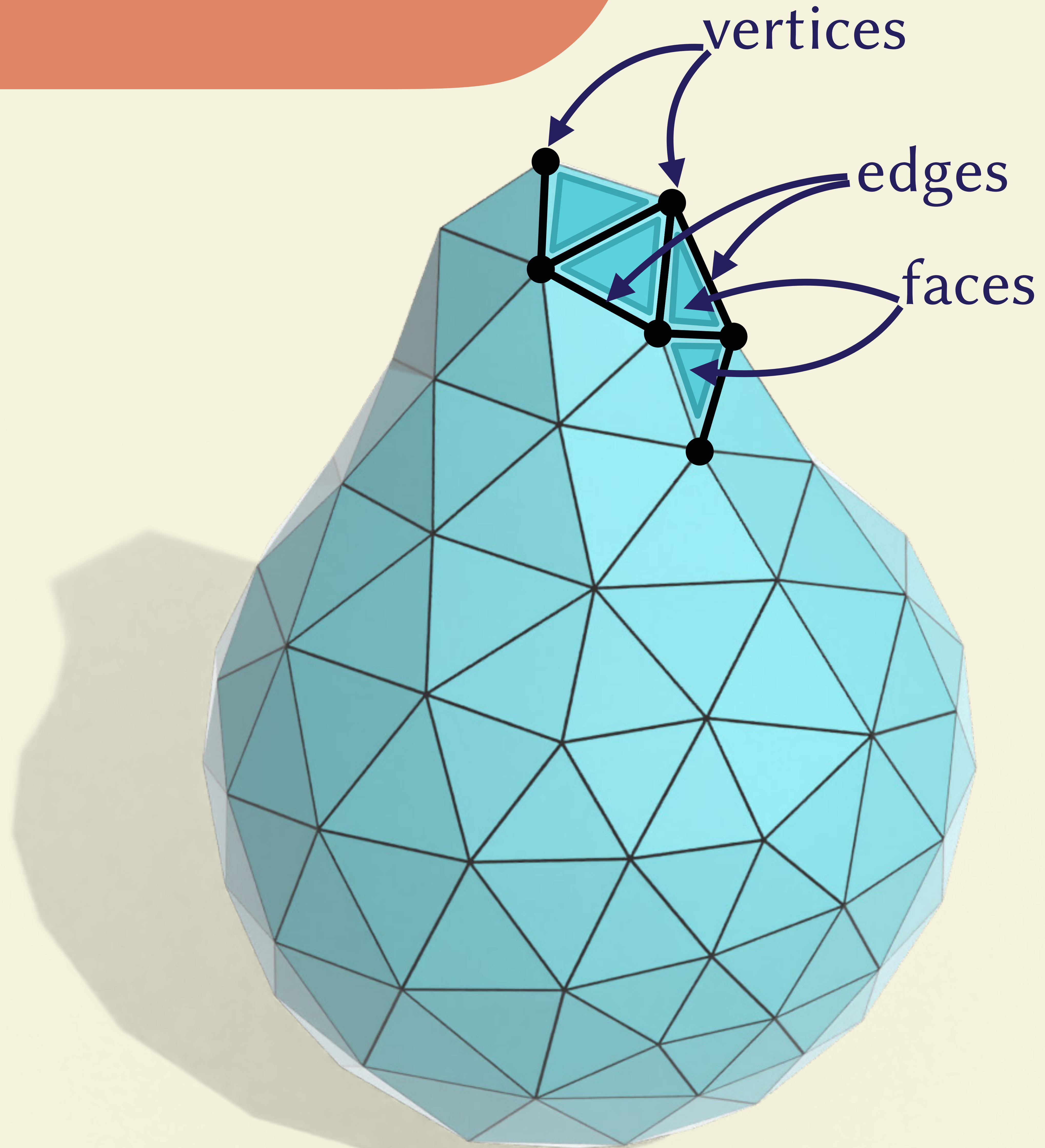
Intrinsic triangulations sidestep the trade off

runs in
milliseconds



- high triangle quality
- exact same geometry
- without too many more triangles

Triangle meshes



DATA STRUCTURE

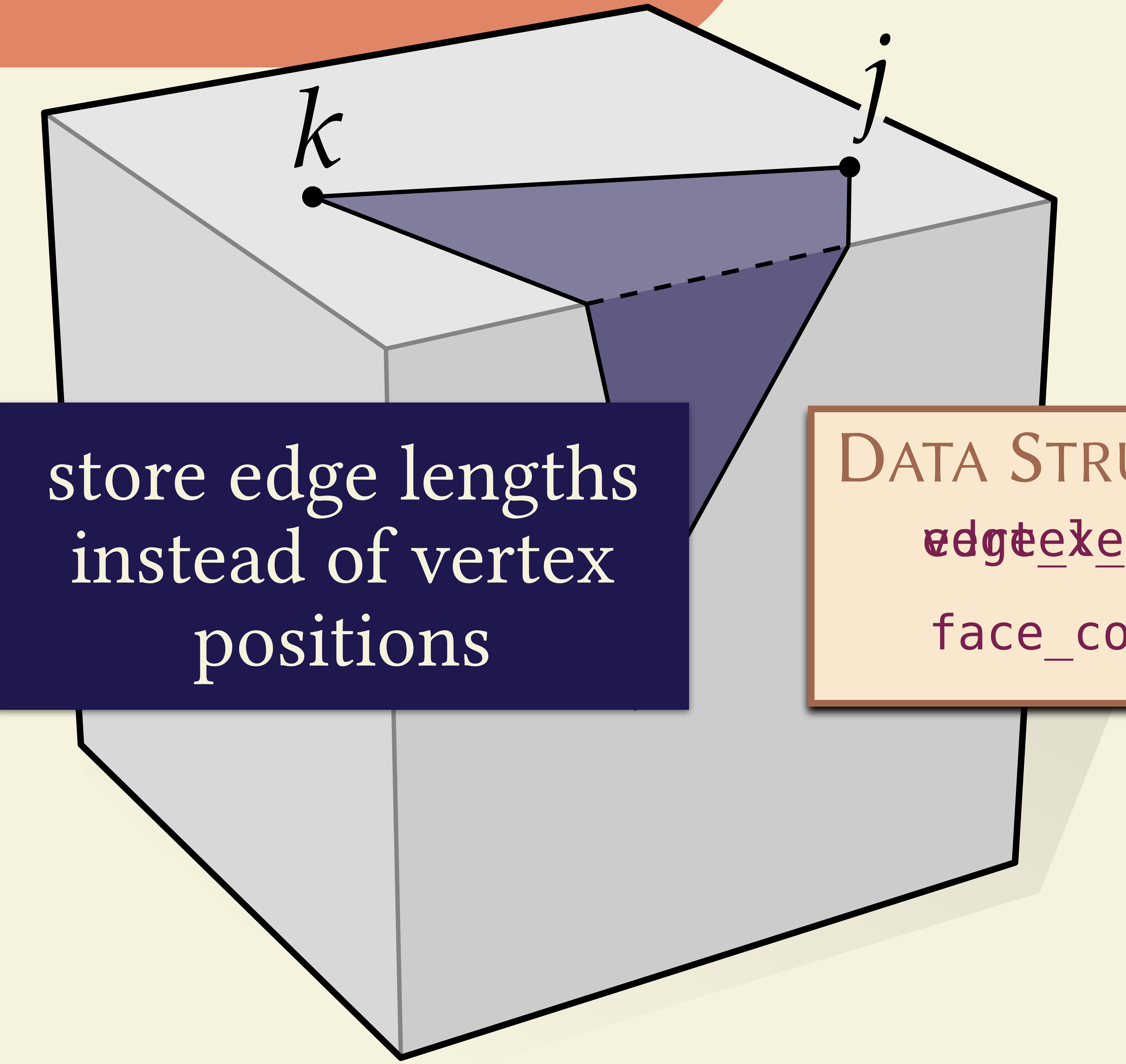
Position Data

```
vertices = {
  ( 1.2, 1.4, 0.7 ),
  ( 1.3,-2.1, 4.7 ),
     . . .
}
```

Connectivity Data

```
faces = {
  ( 0, 1, 2 ),
  ( 2, 1, 3 ),
  ( 5, 8, 9 ),
     . . .
}
```

Intrinsic triangles

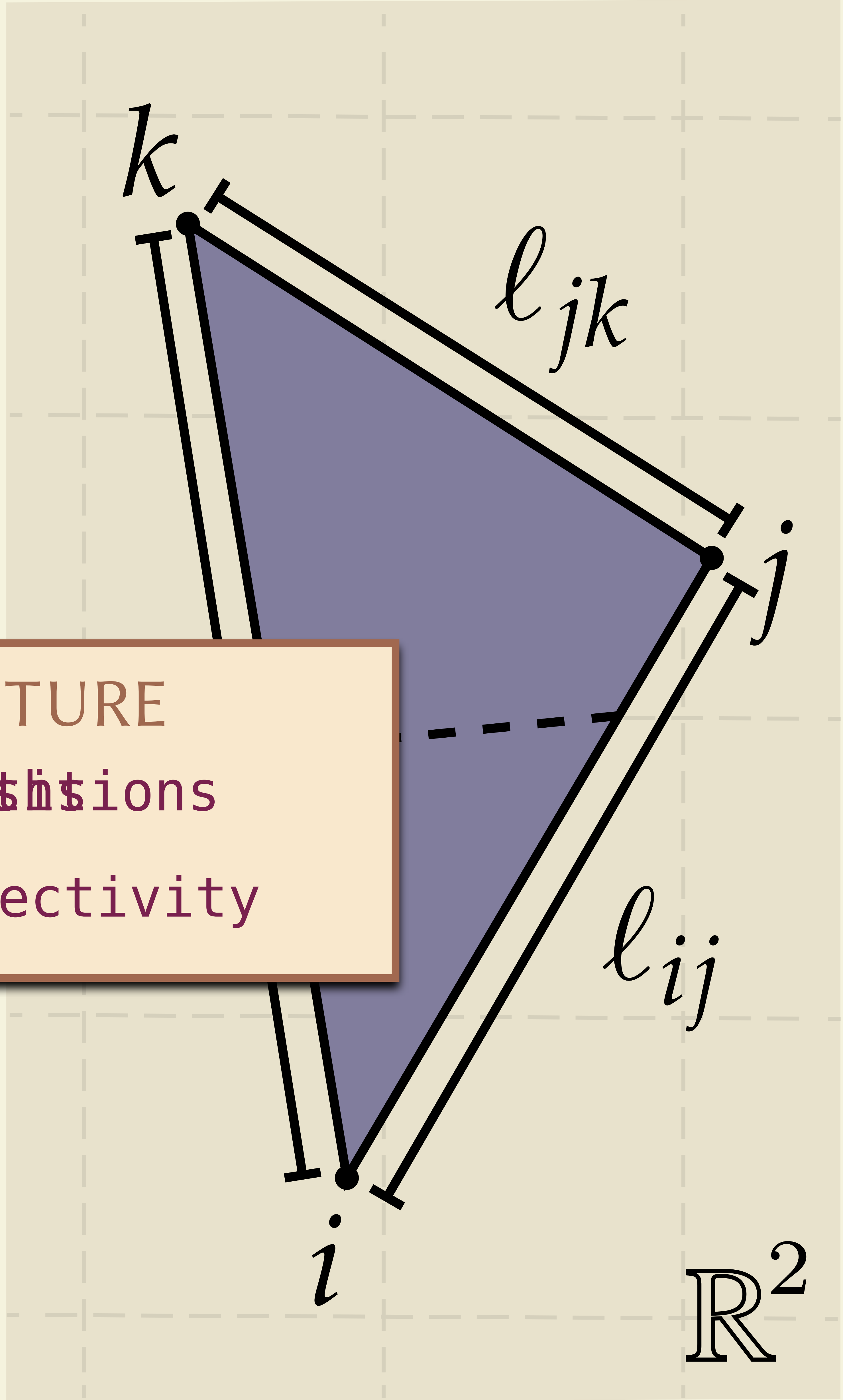


store edge lengths instead of vertex positions

DATA STRUCTURE

`edge_lengths`
`vertex_positions`

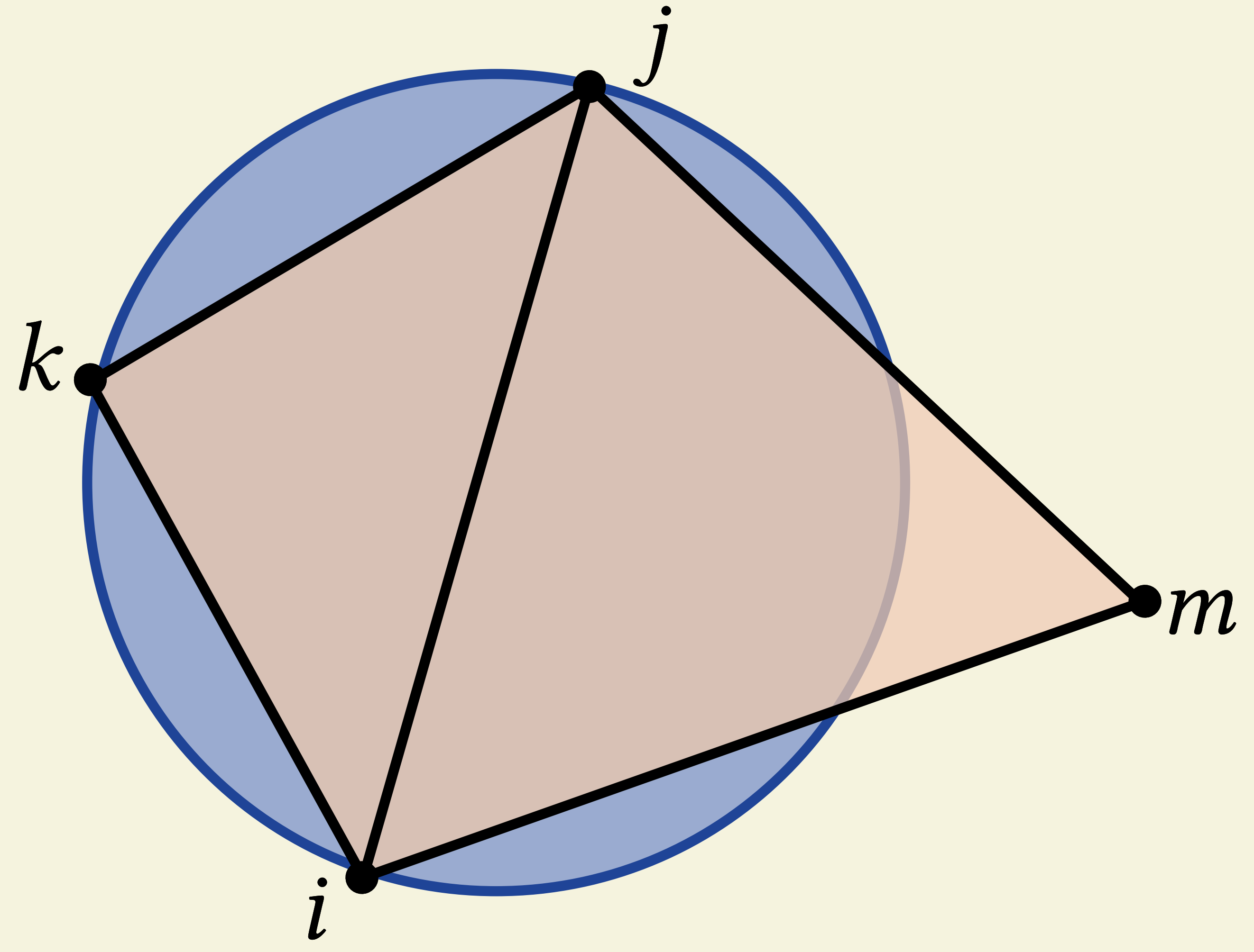
`face_connectivity`

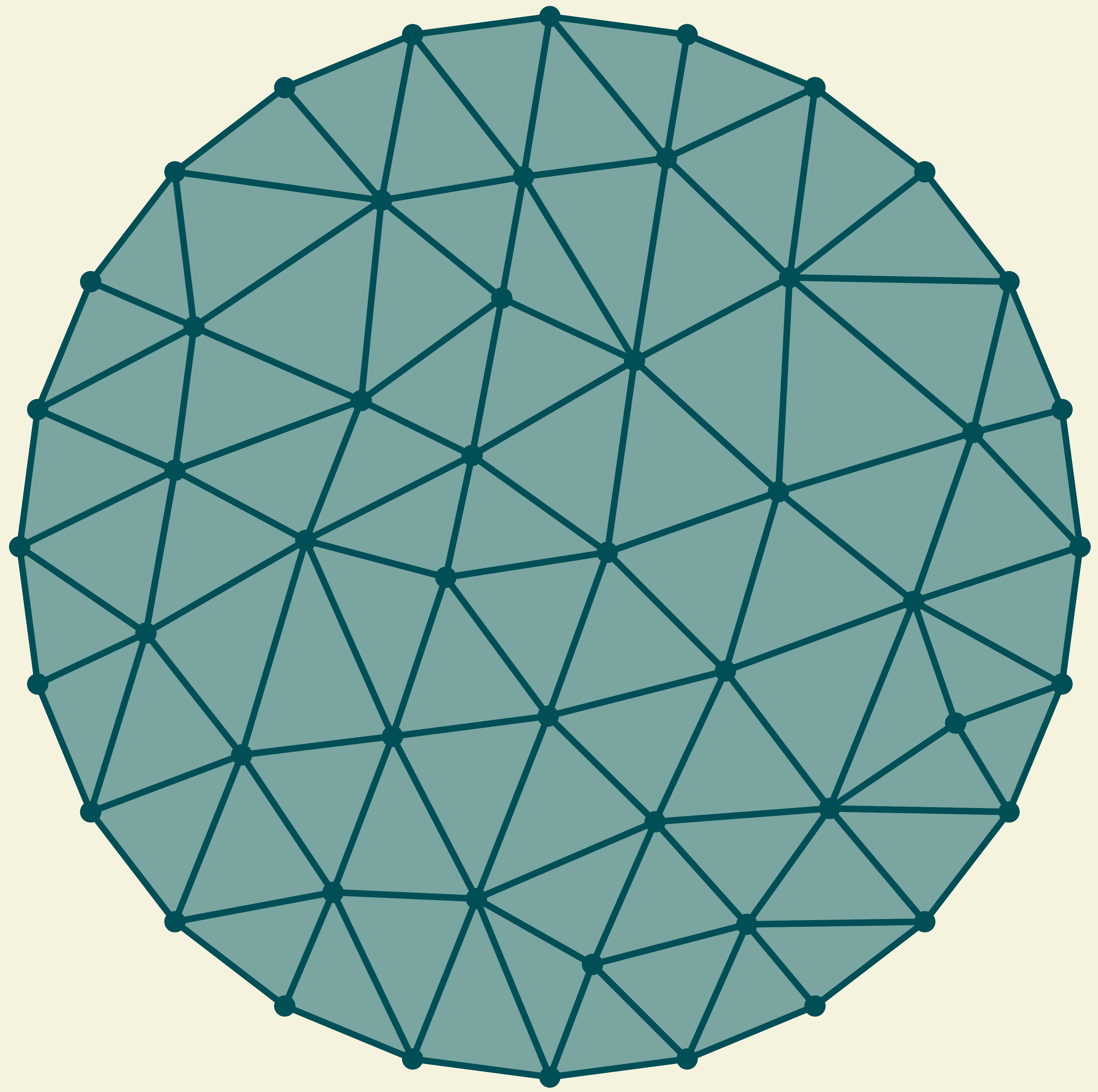


The space of intrinsic triangulations is large

Delaunay triangulations

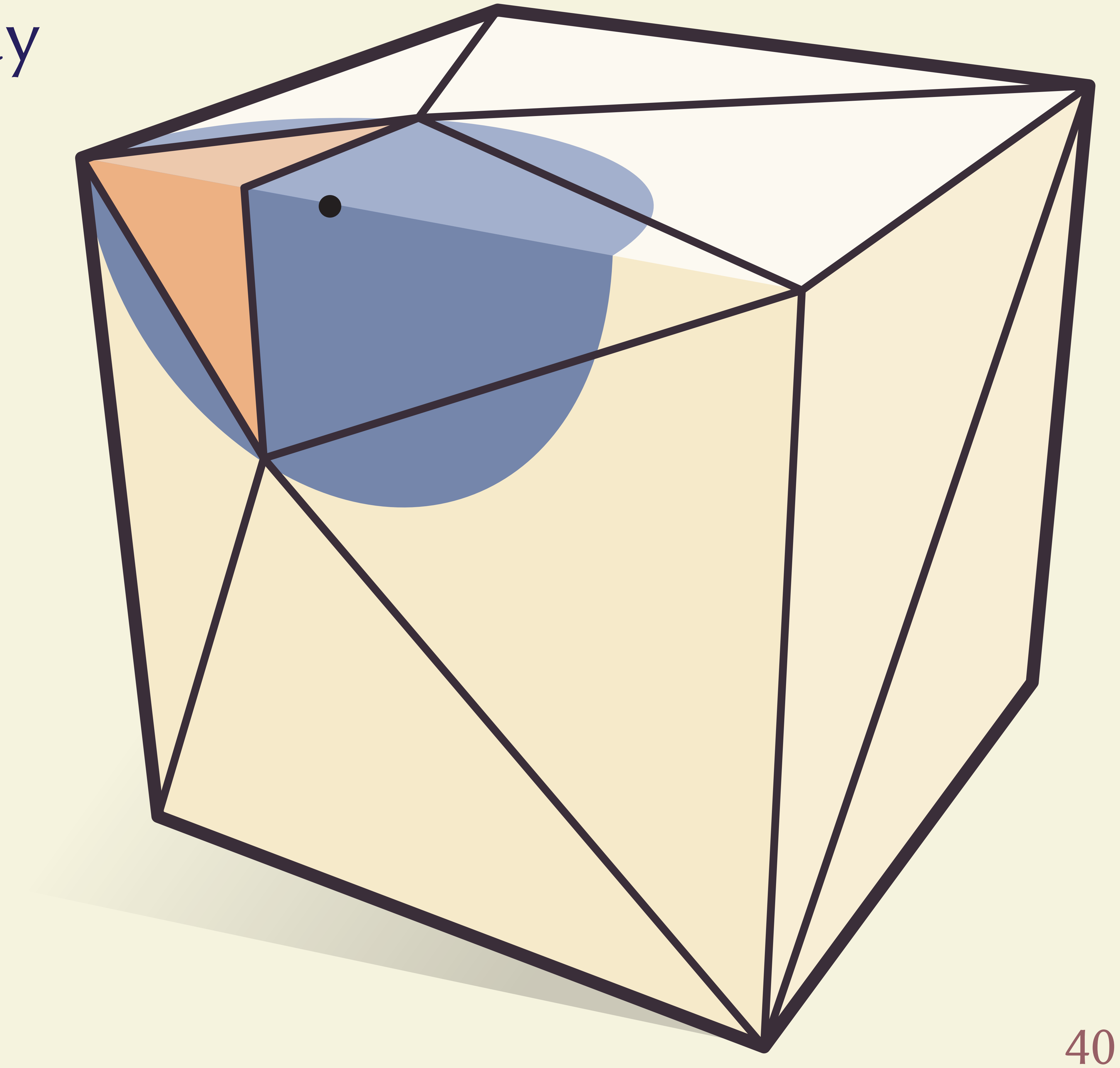
- Countless useful properties:
  - Maximize angles lexicographically, minimize spectrum lexicographically, smoothest interpolation, positive cotan weights…
- Key to successful 2D remeshing algorithms
- Characterized by *empty circumcircle condition*





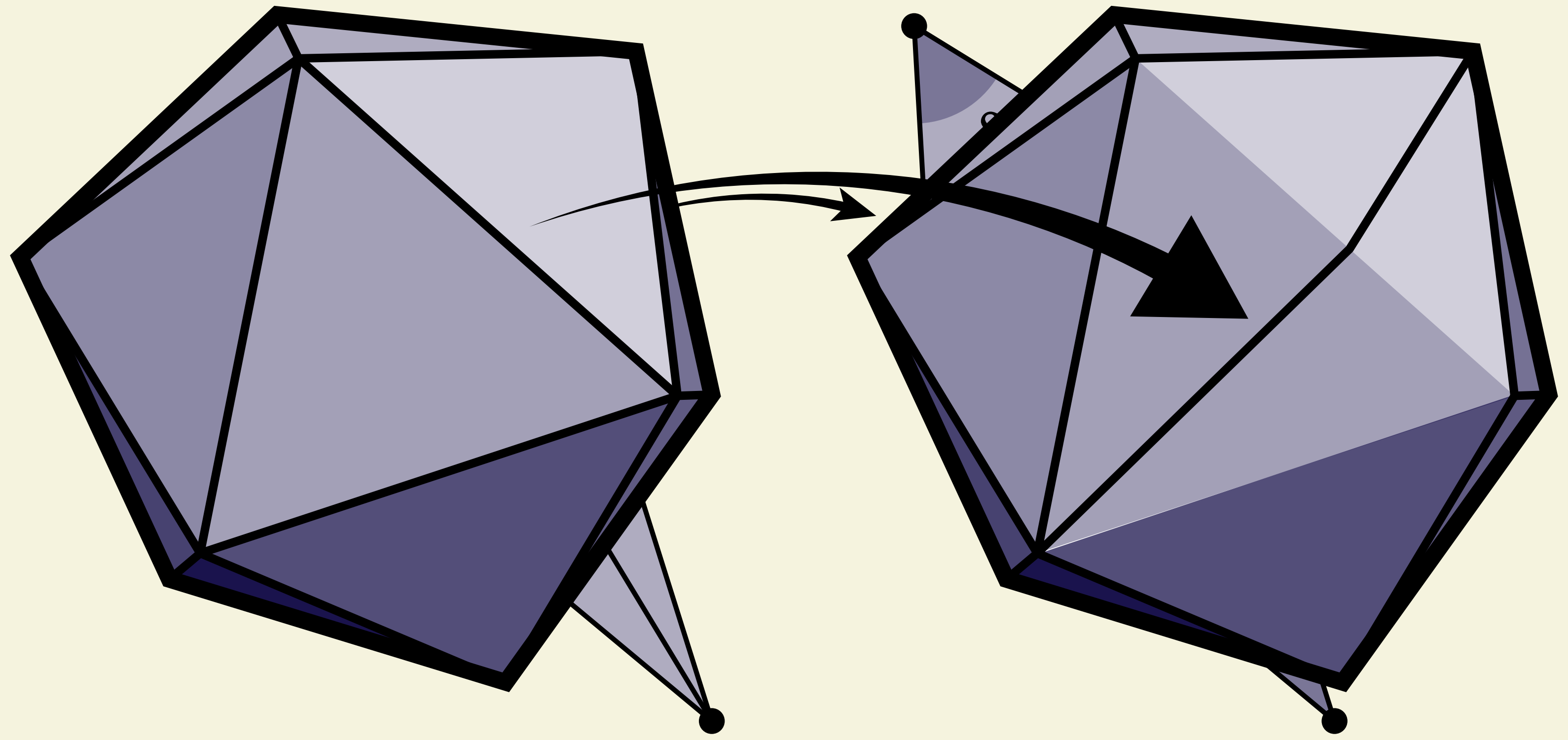
Intrinsic Delaunay triangulations

- [Masur & Smillie 1991, Rivin 1994, Indermitte *et al.* 2001, Bobenko & Springborn 2007]: intrinsic Delaunay
    - Maintain useful mathematical properties.
      [Sharp, **Gillespie** & Crane 2021]



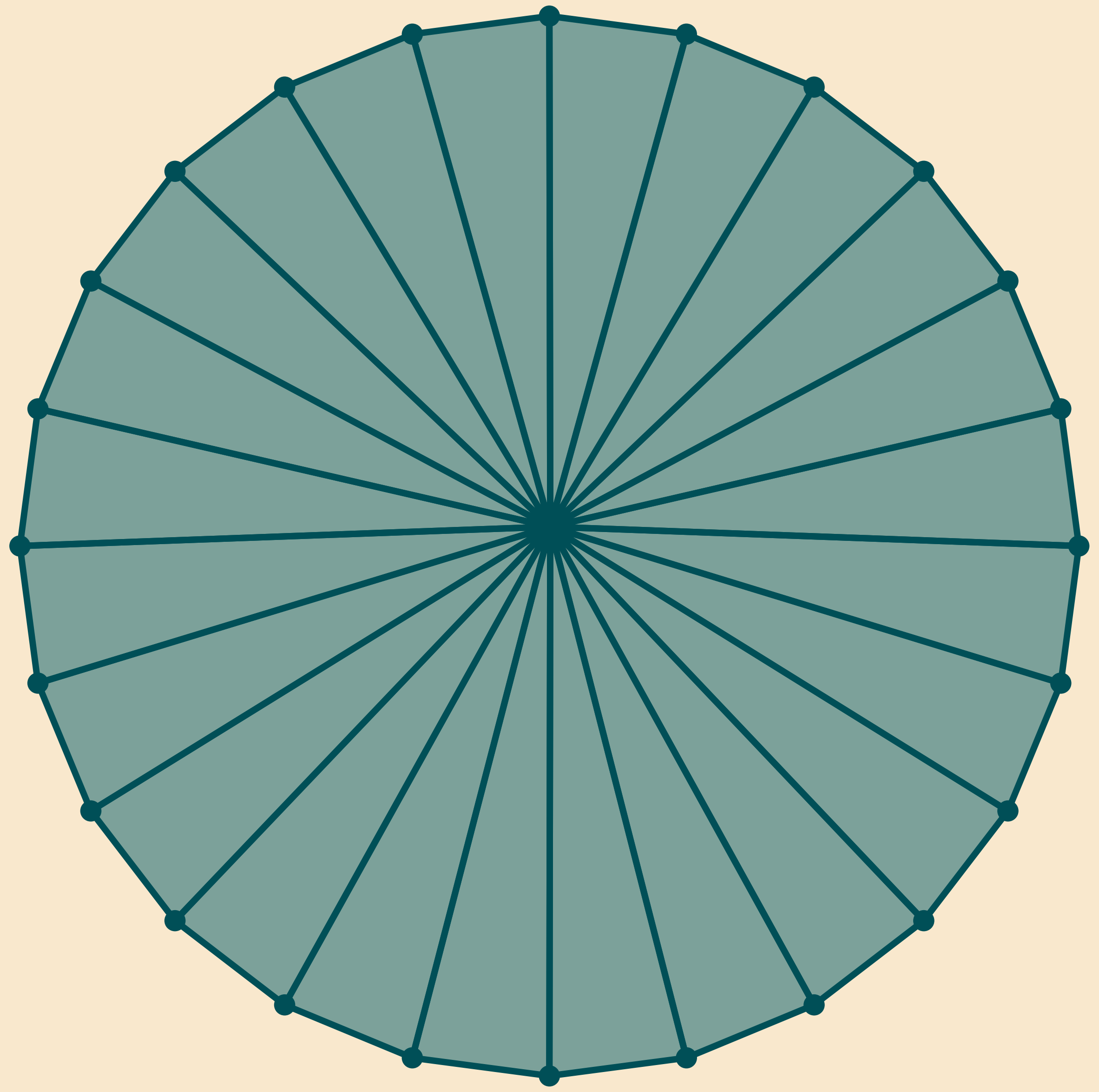
Intrinsic Delaunay triangulations

- [Masur & Smillie 1991, Rivin 1994, Indermitte *et al.* 2001, Bobenko & Springborn 2007]: intrinsic Delaunay
    - Maintain useful mathematical properties. [Sharp, **Gillespie** & Crane 2021]
- Compute by a simple algorithm:
    - Flip any non-Delaunay edge until none remain



Faster than reading the mesh from disk

Problem: cannot guarantee high triangle quality

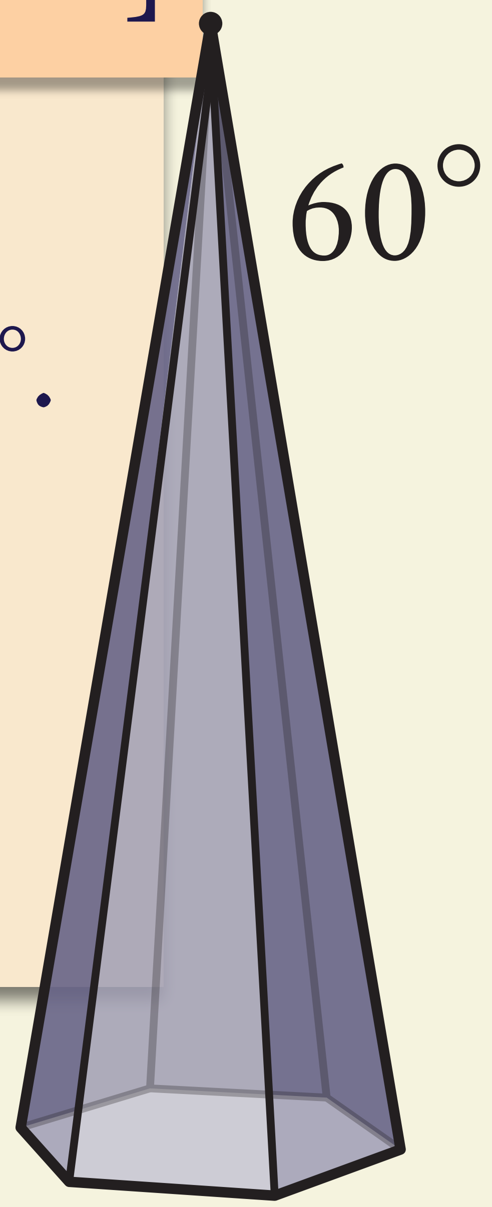


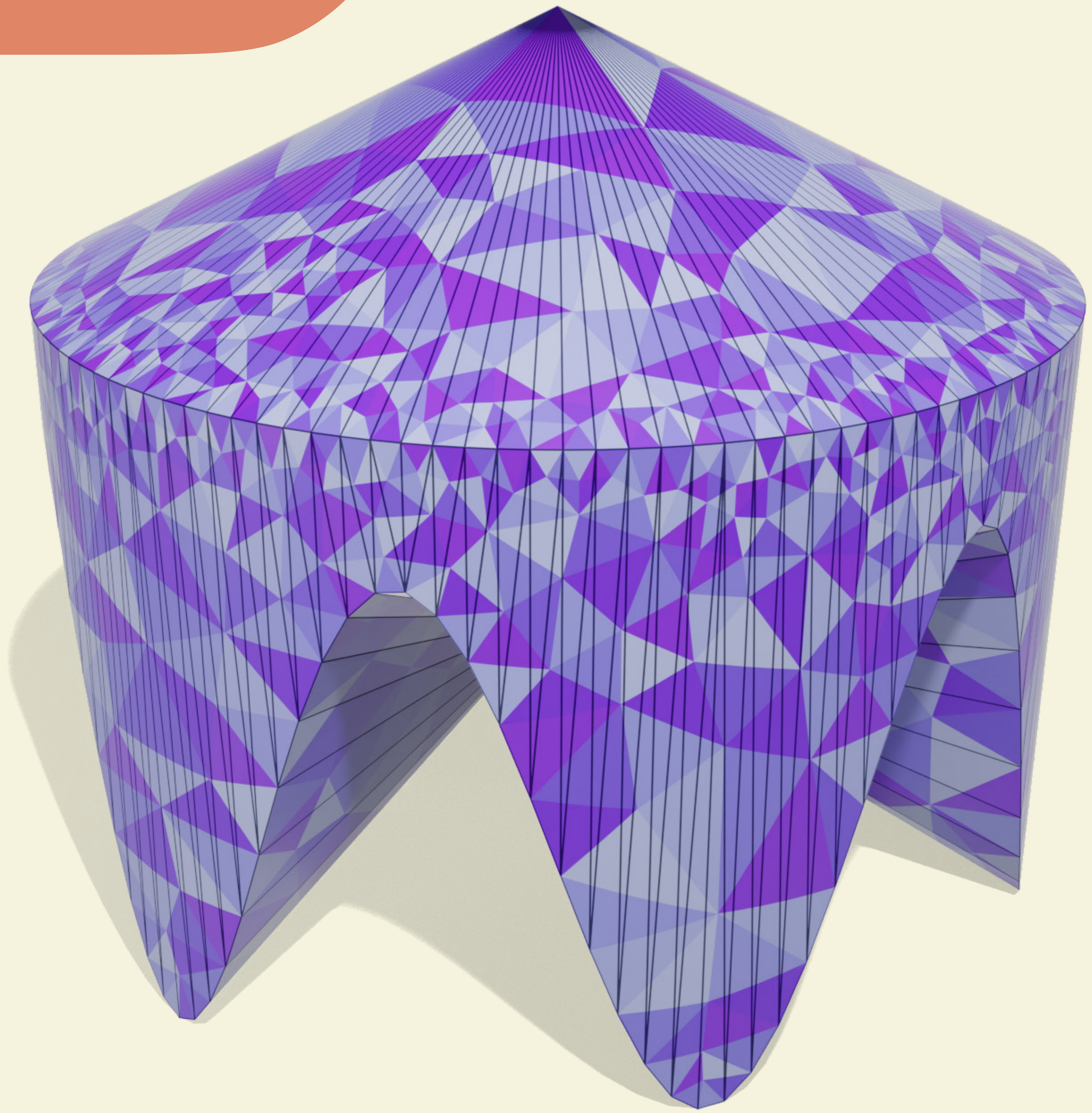
low quality Delaunay triangulation

Intrinsic Delaunay refinement

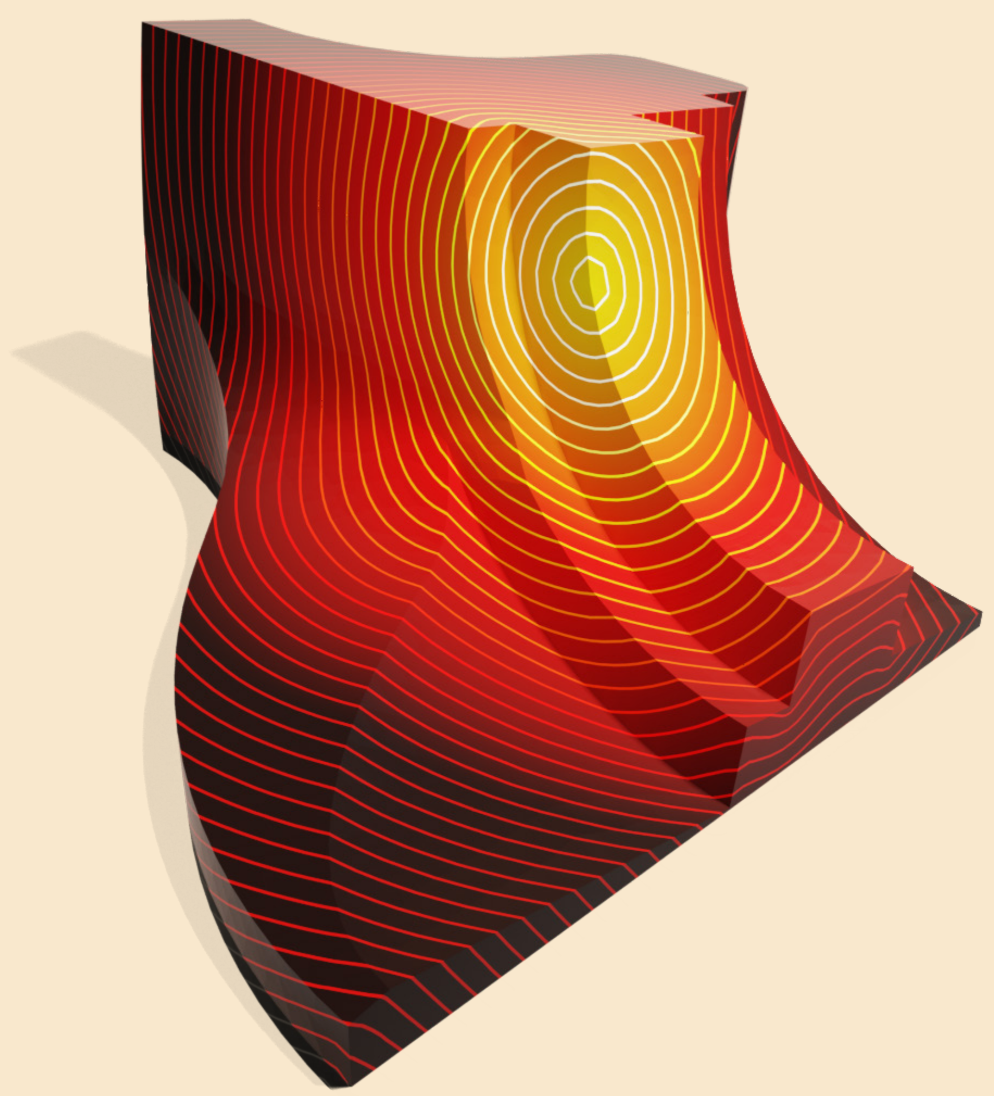
Theorem [**Gillespie**, Sharp & Crane 2021]

Let M be a mesh without boundary whose cone angles are all at least 60°. Then intrinsic Delaunay refinement produces a Delaunay mesh with triangle corner angles at least 30°



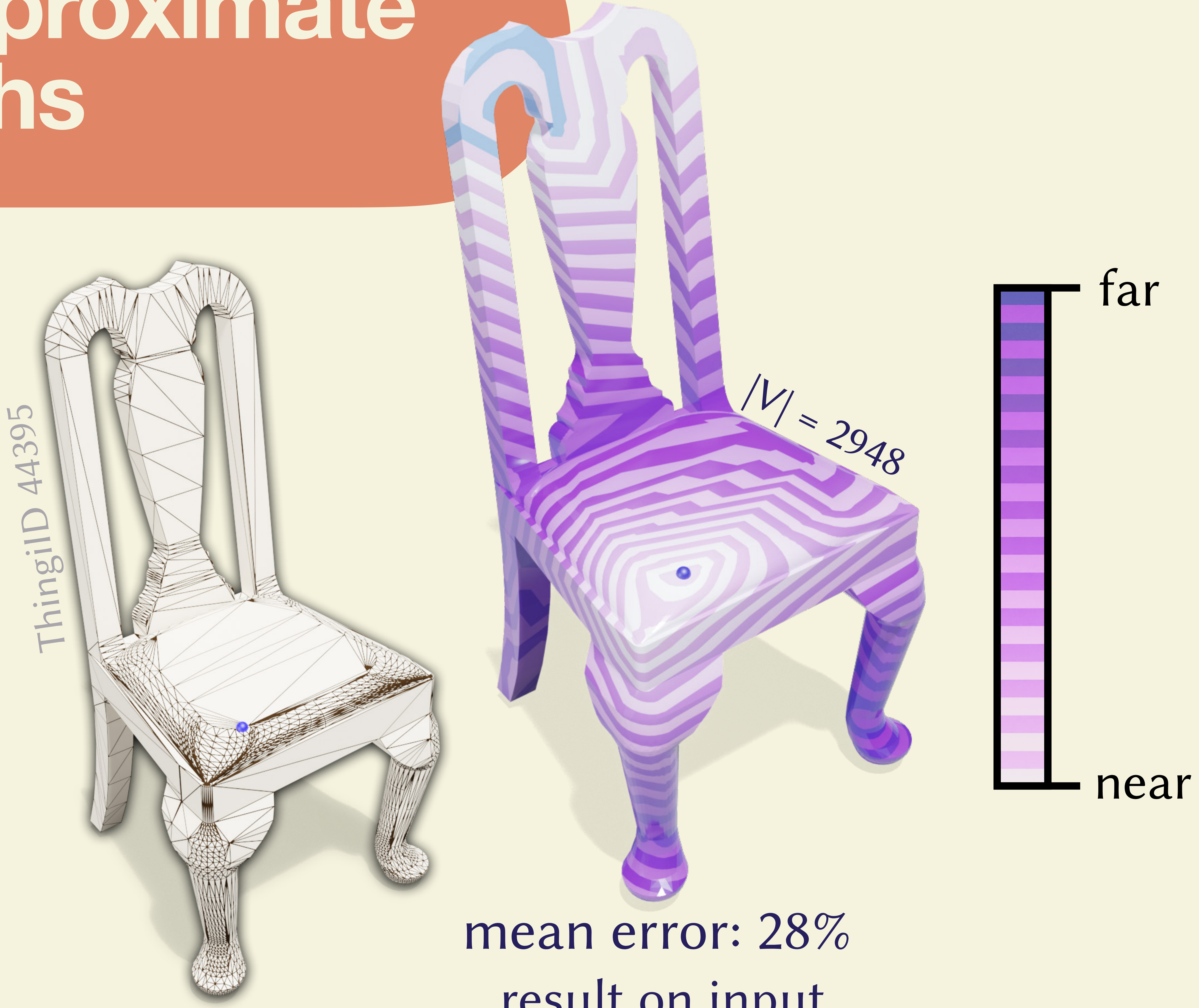


Example: Approximate Shortest Paths



heat method for surface distances

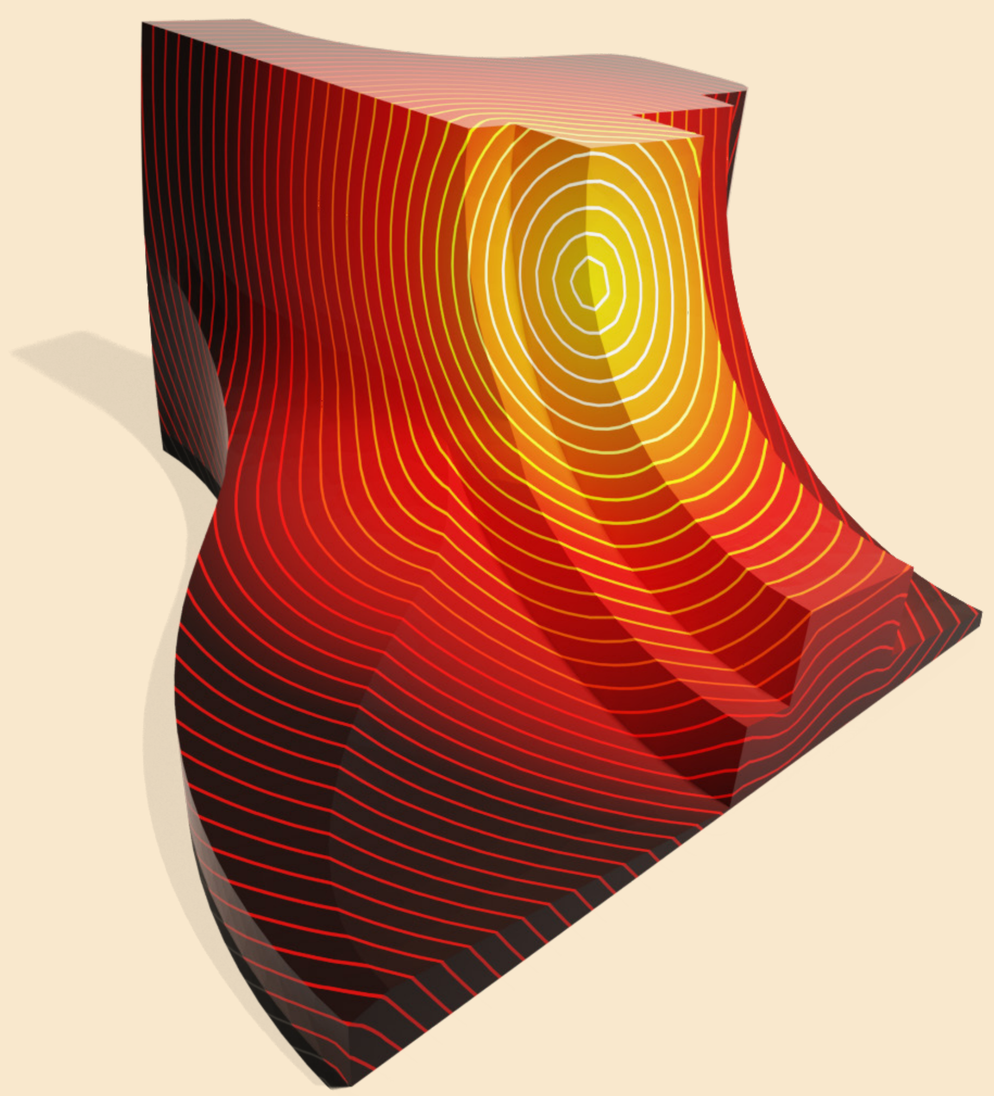
[Crane, Weischedel & Wardetzky 2013]



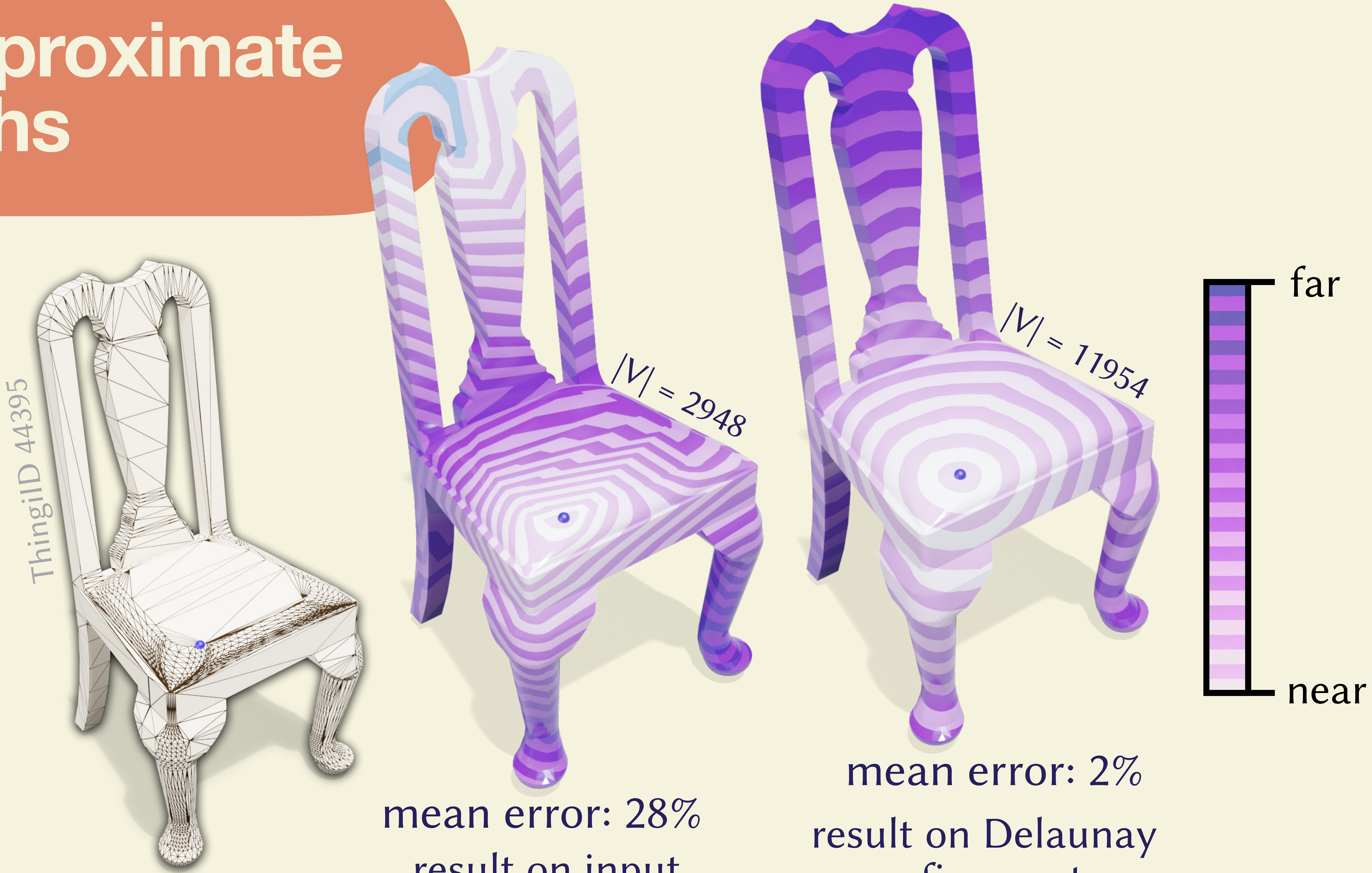
mean error: 28%

result on input mesh

Example: Approximate Shortest Paths



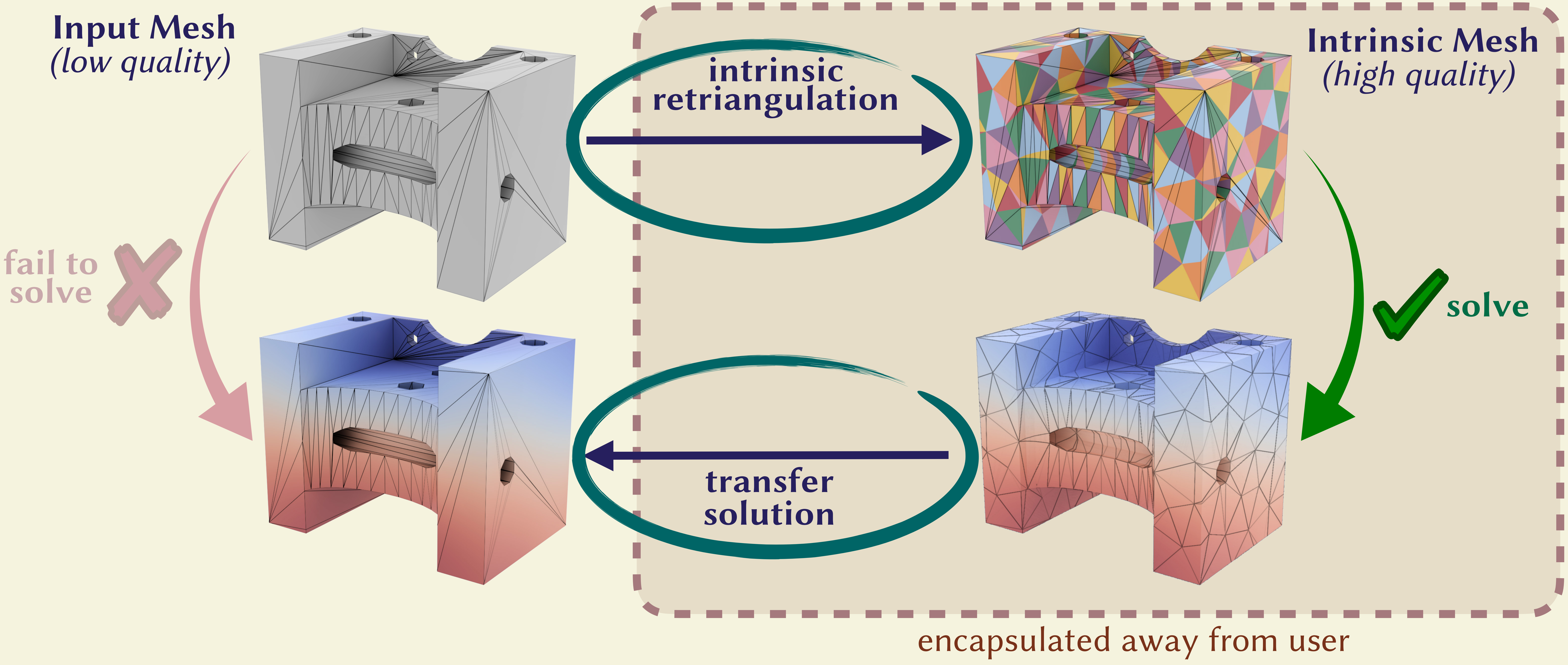
heat method for
surface distances
[Crane, Weischedel &
Wardetzky 2013]



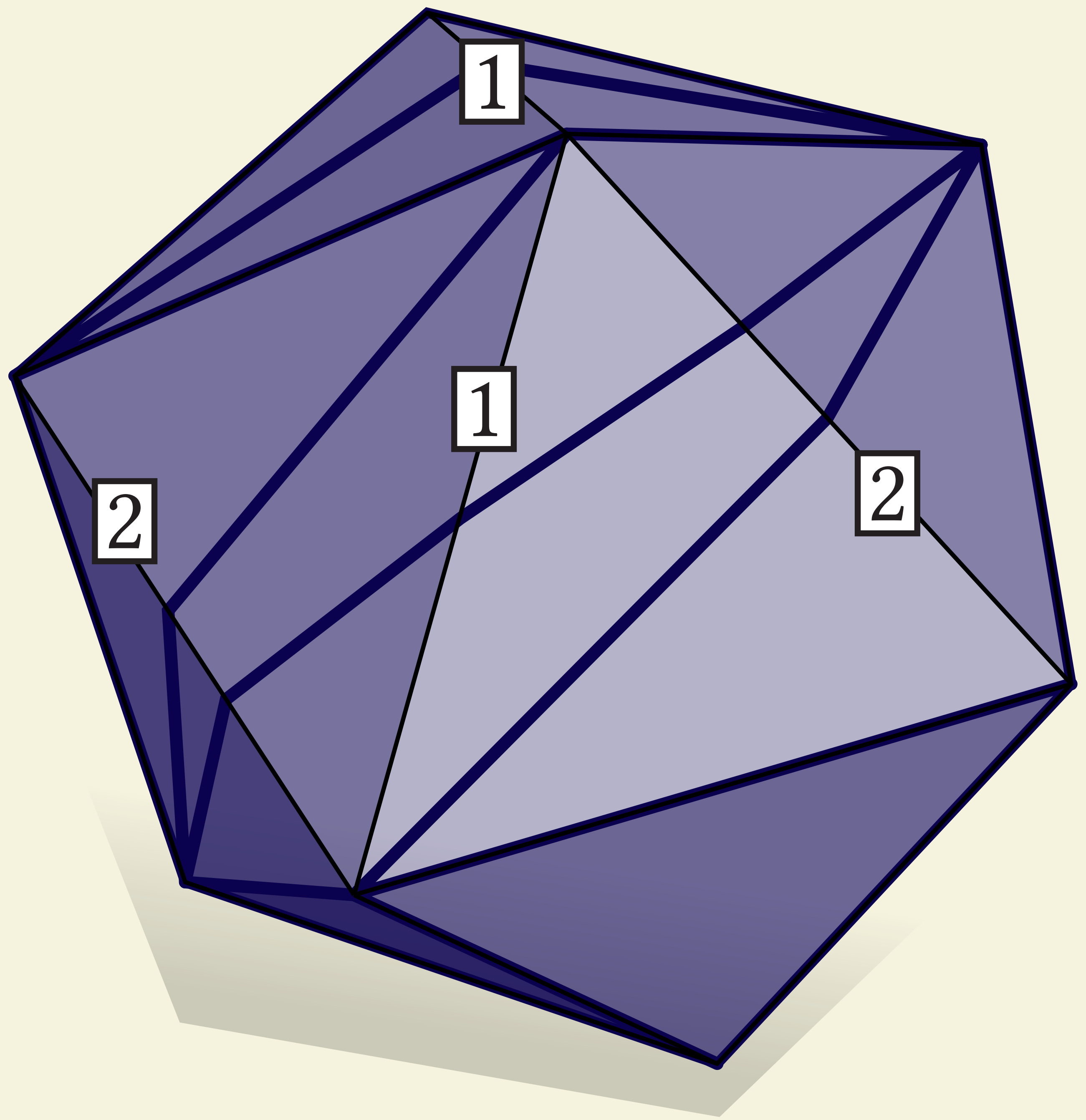
mean error: 28%
result on input mesh

mean error: 2%
result on Delaunay refinement

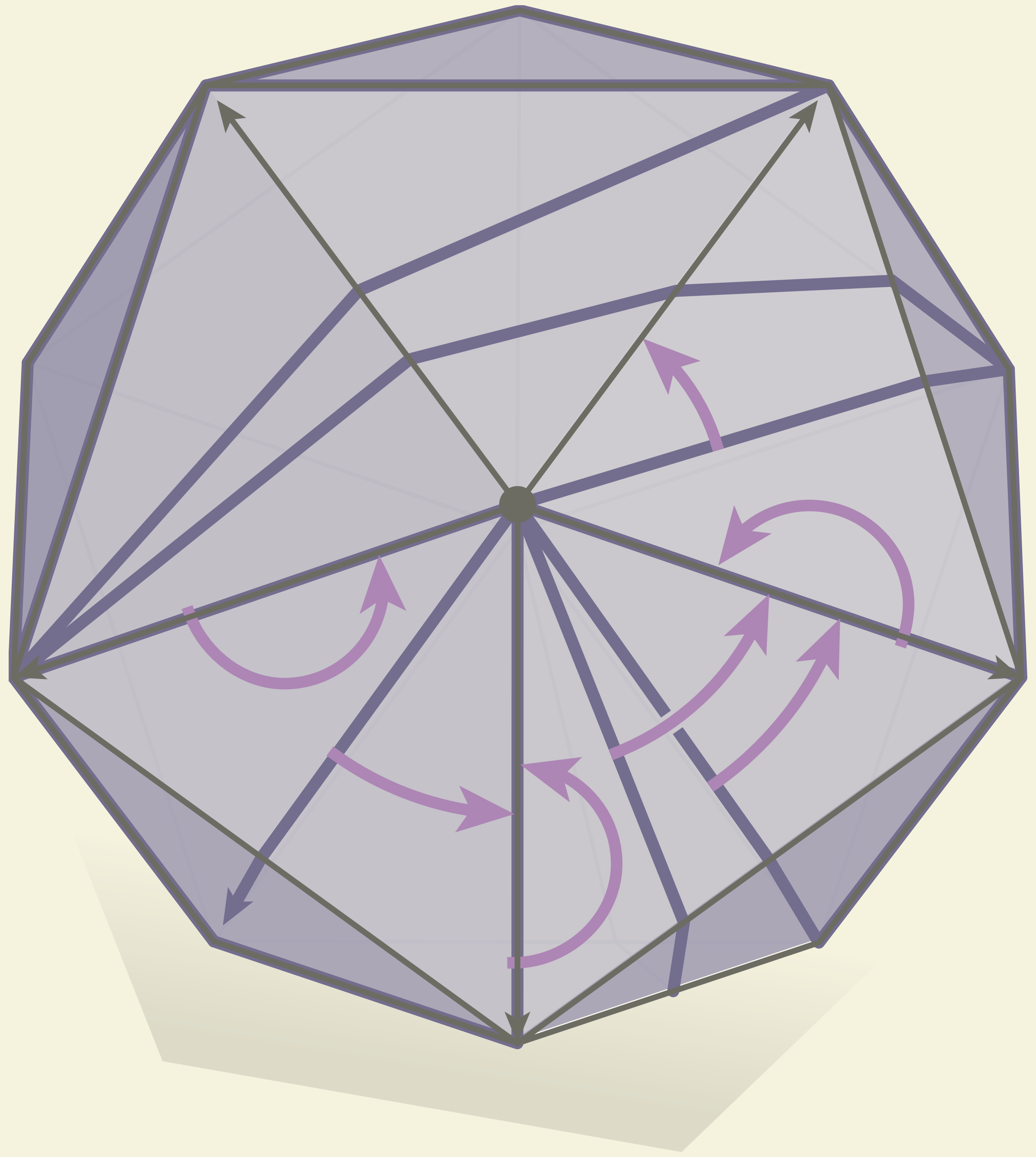
Tracking Correspondence



Correspondence data structure



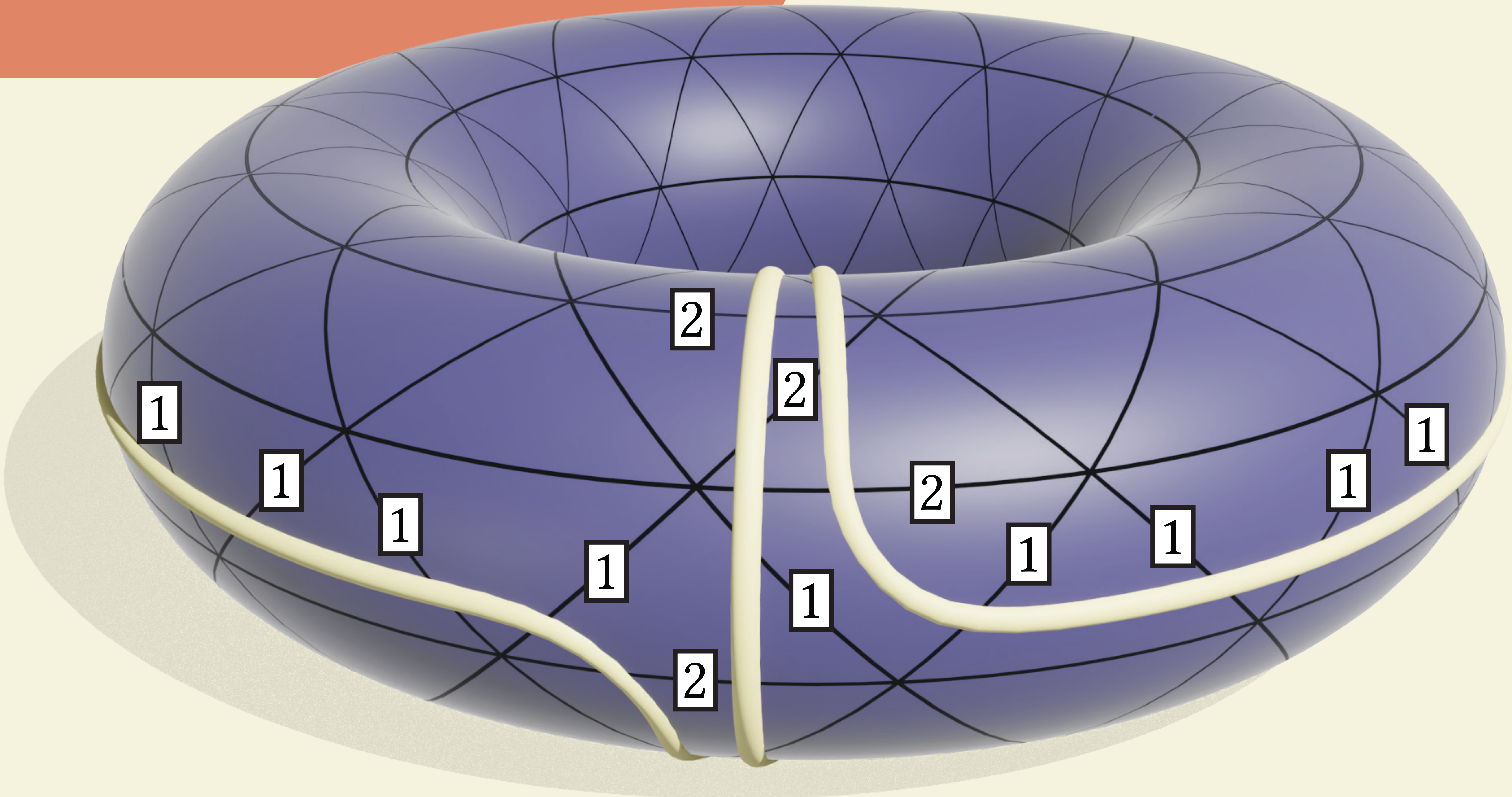
normal coordinates



roundabouts

(concretely, just 3 integers per edge)

Normal coordinates

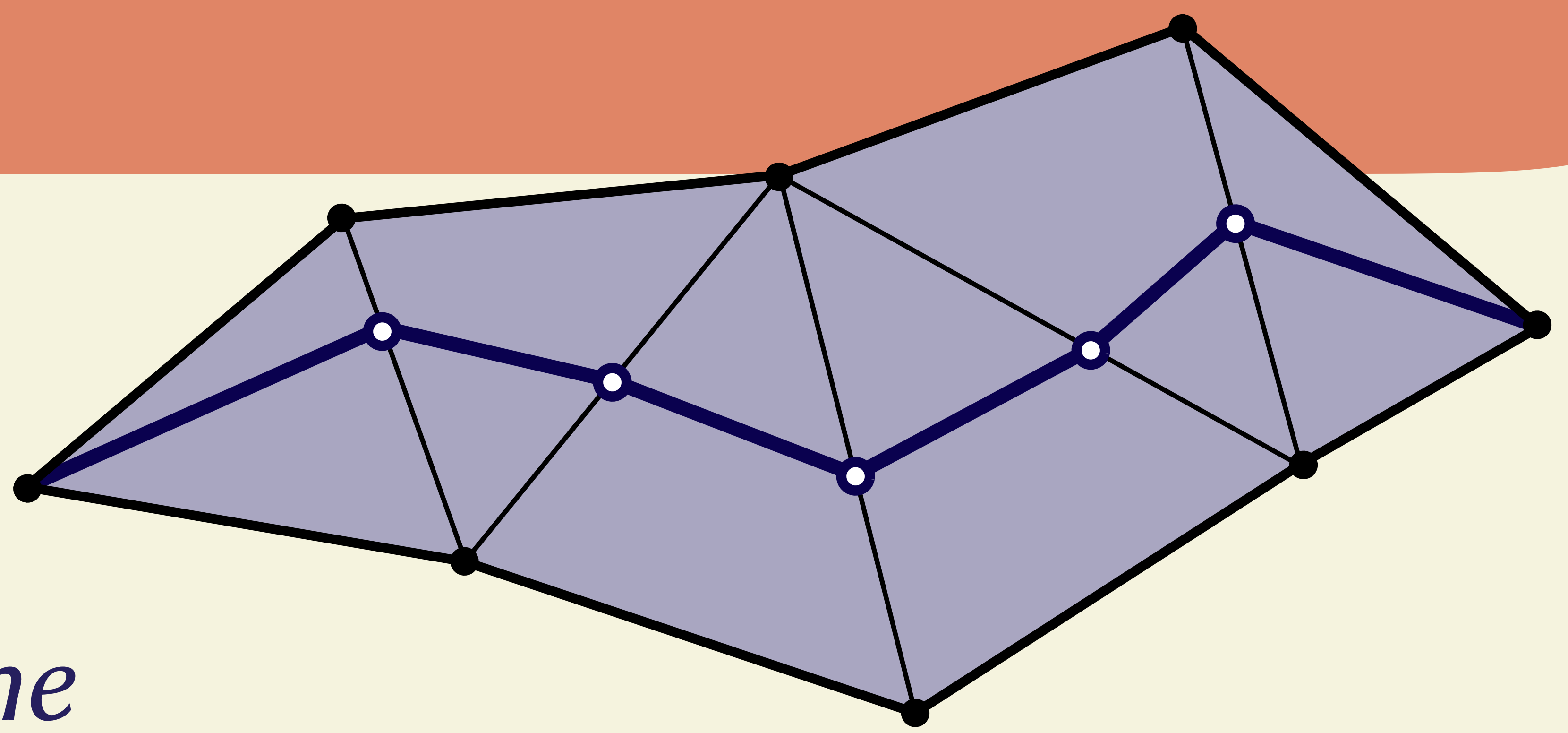


Foundations: [Kneser 1929; Haken 1961]

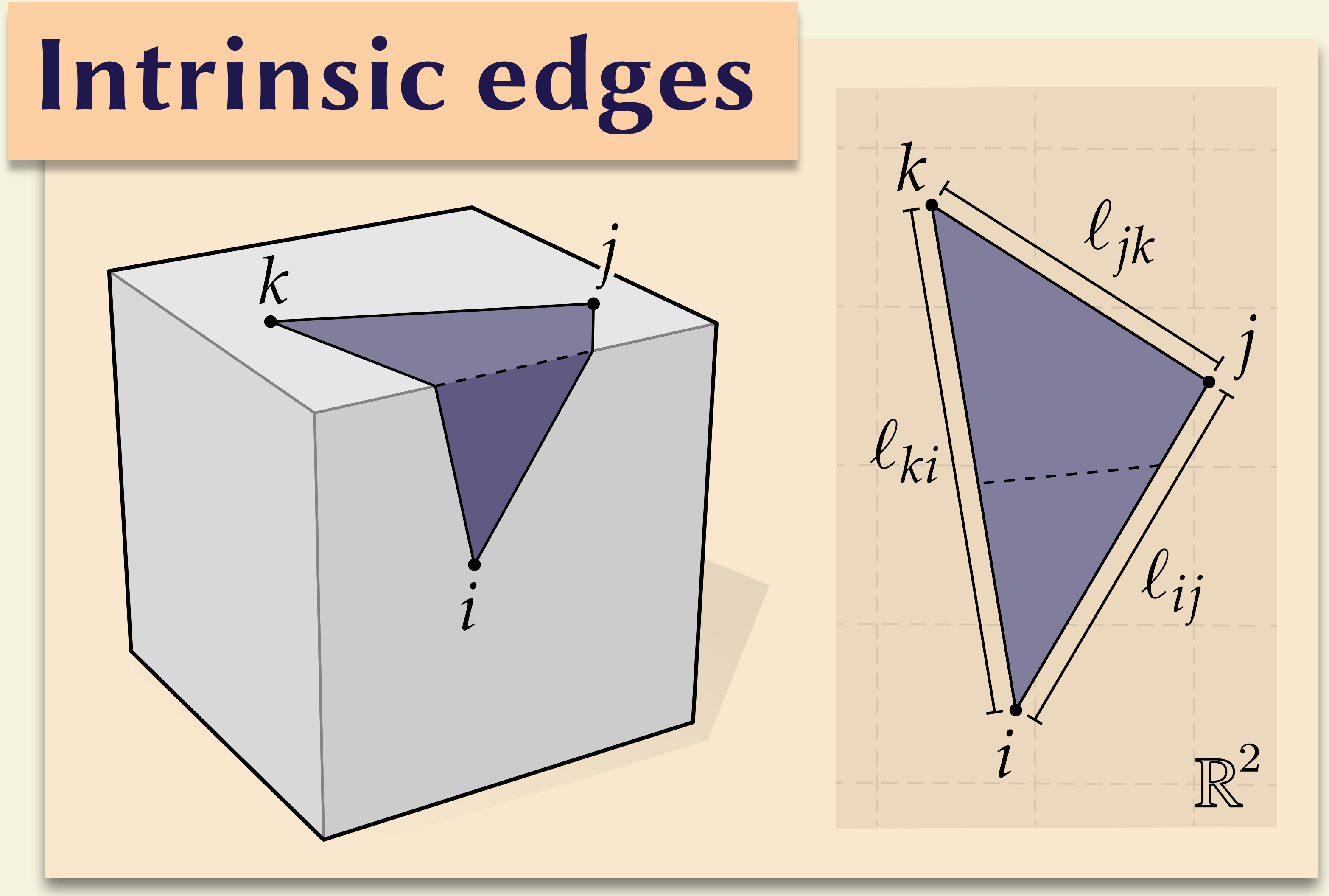
Computational Topology: [Schaefer+ 2008; Erickson & Nayyeri 2013]

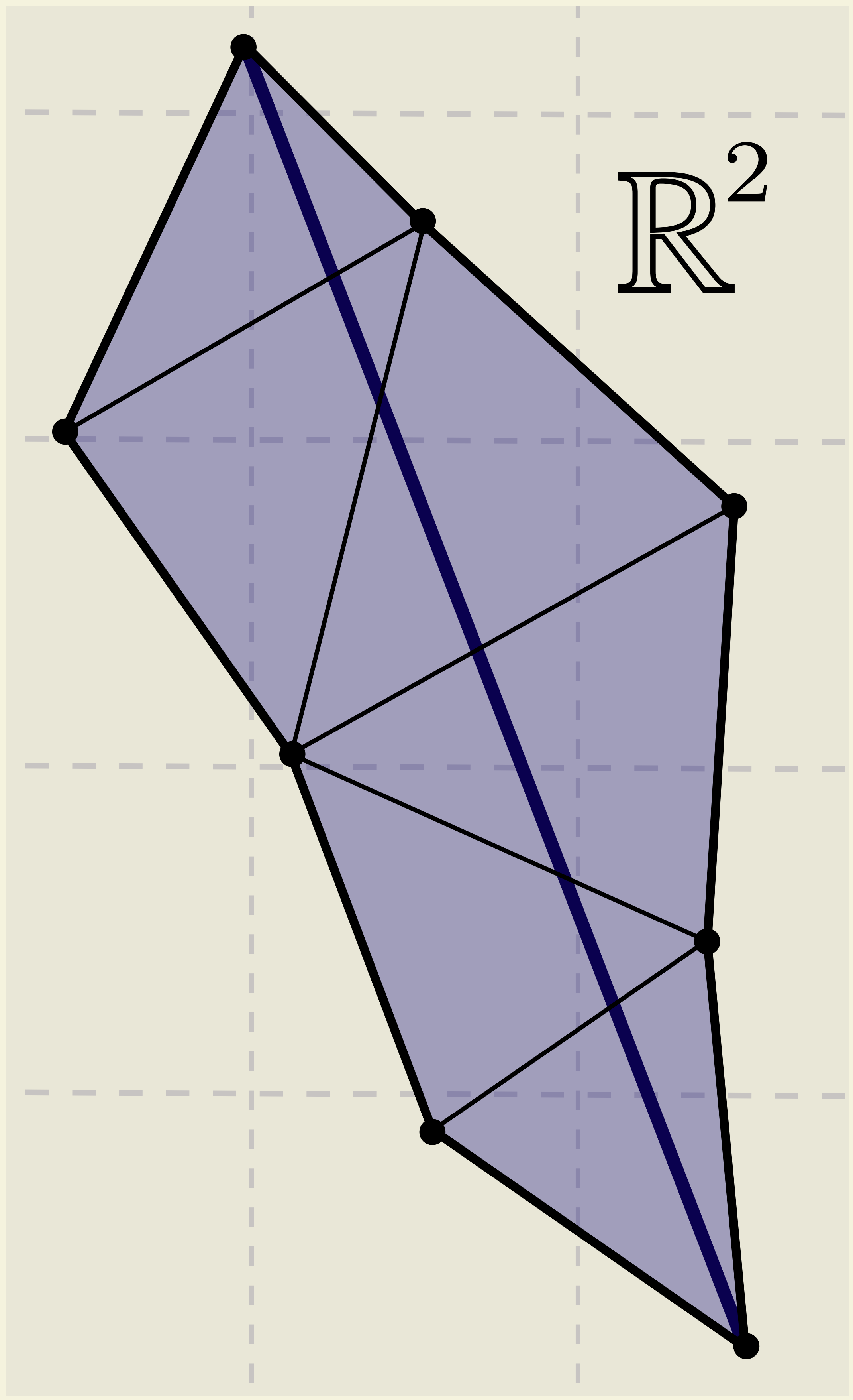
Geometry Processing: [Hass & Trnkova 2020]

Finding the exact curve geometry



- Ordinarily: sequence of triangles
- True curve unfolds to a *straight line*
    - Lay out in plane for exact curve
- Normal coordinates determine edges exactly



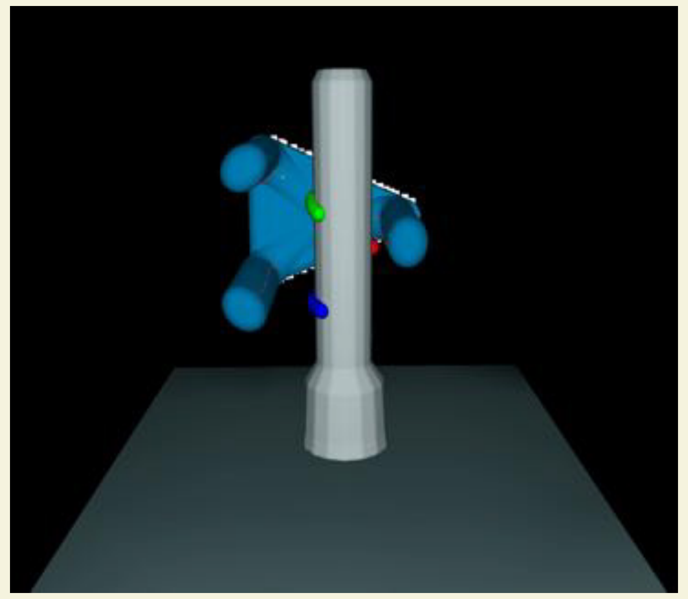


Representing an intrinsic triangulation

Intrinsic Delaunay refinement — validation

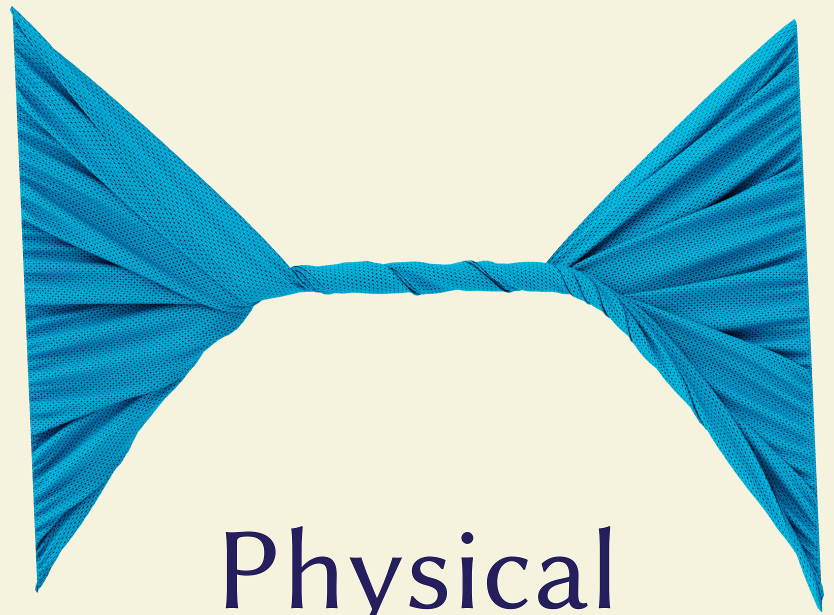
- Compute refinements for Thingi10k dataset [Zhou & Jacobson 2016]
  - 7696 manifold meshes
- 100% success rate
  - [Sharp *et al.* 2019] succeed on only 69.1% of meshes

Usage

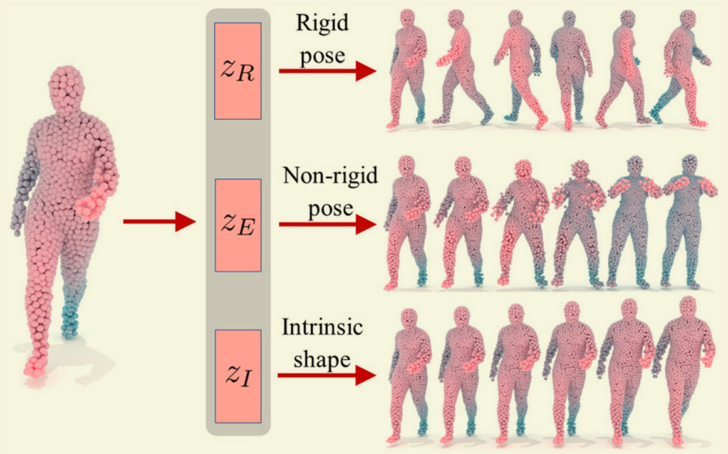


Robotics
[Lakshmipathy *et al.* 2024]

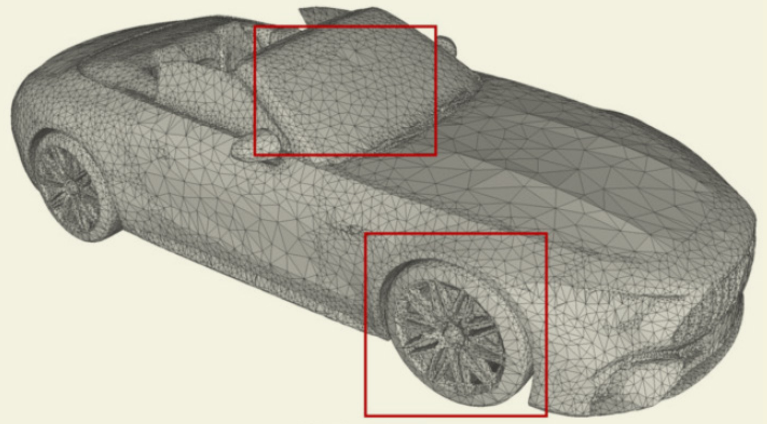
Cell Modeling
[Numerow *et al.* 2024]



Physical Simulation
[Zhang *et al.* 2023]

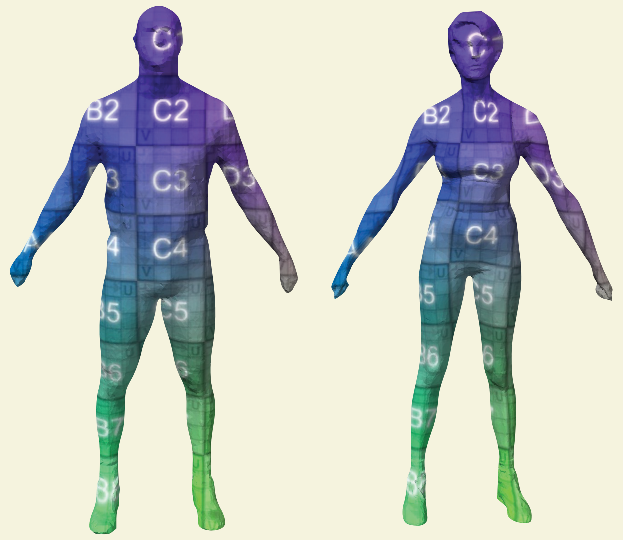


Geometric ML
[Aumentado-Armstrong *et al.* 2023]

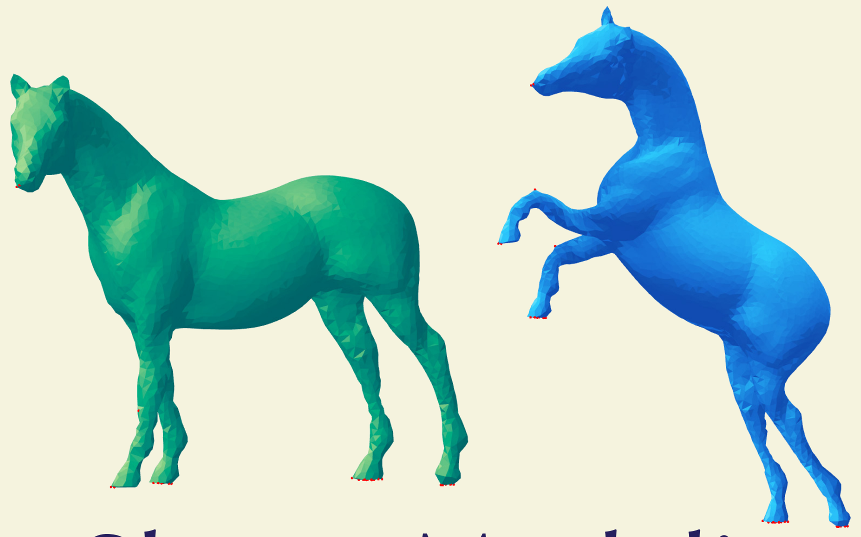


Extrinsic Remeshing
[Dai *et al.* 2024]

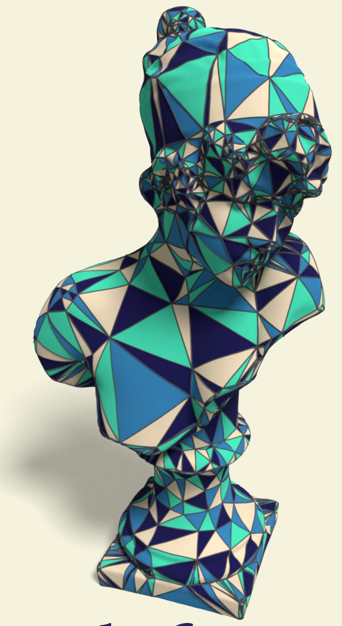
Parameterization
[Fargion & Weber 2022]
[Wang *et al.* 2022]
[Akalin *et al.* 2024]



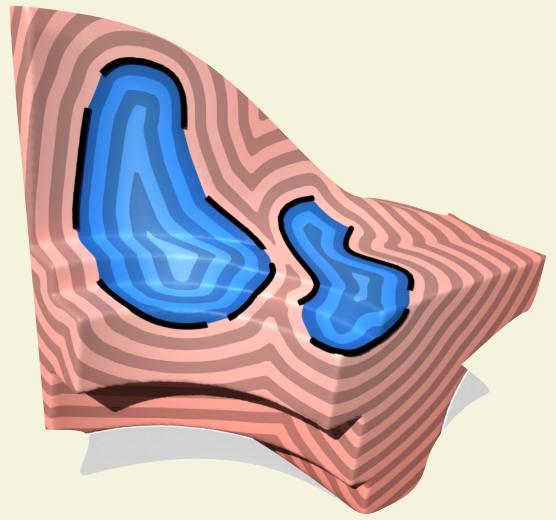
Surface Correspondence
[Takayama 2022]



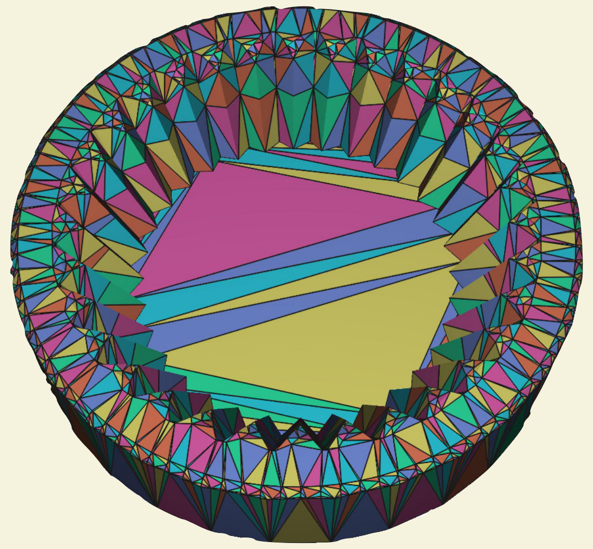
Shape Modeling
[Finnendahl *et al.* 2023]



Simplification
[Shoemaker *et al.* 2023]



Surface Distances
[Feng & Crane 2024]

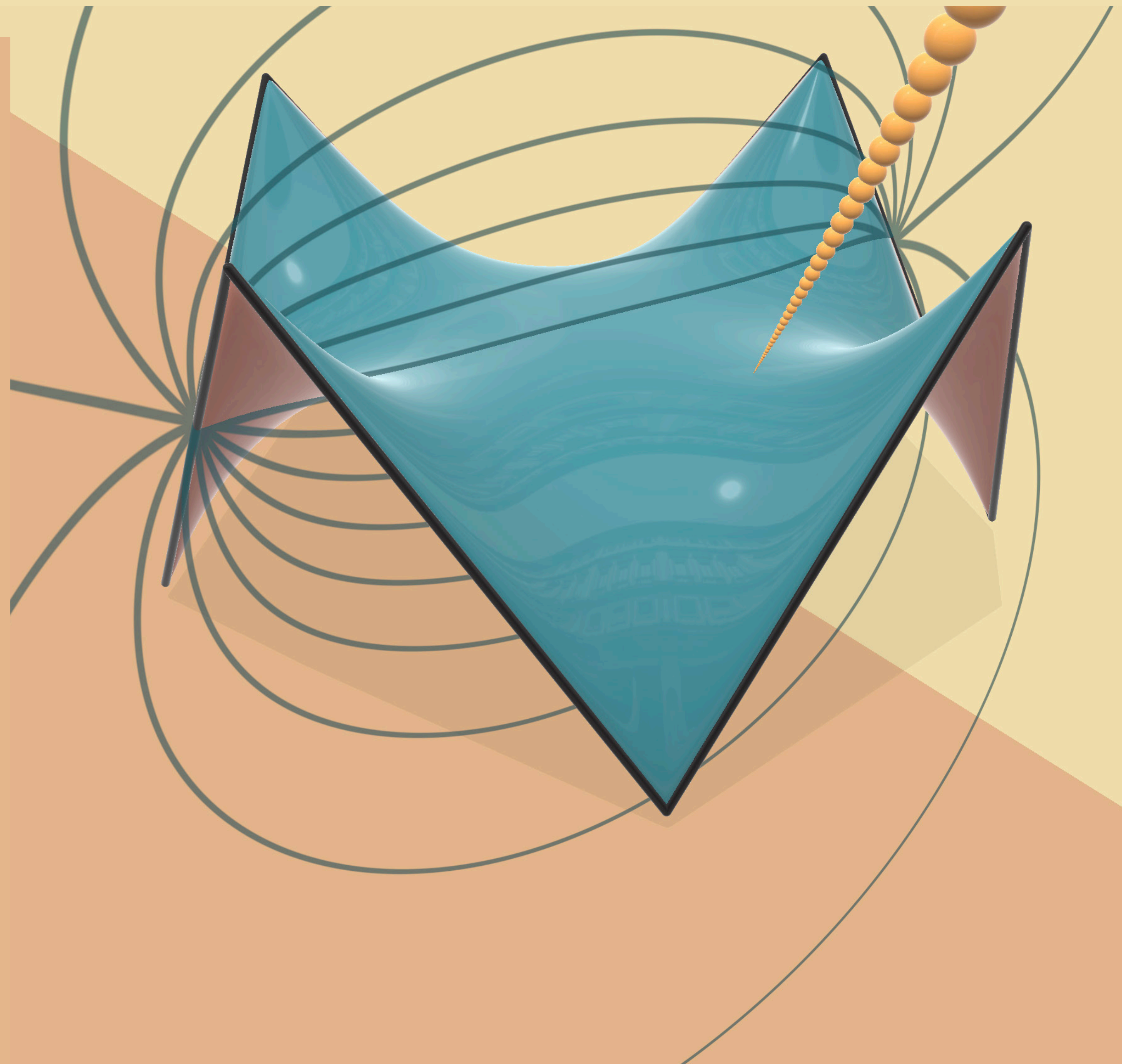


Rendering
[Celes 2025]

A New Family of Implicit Surfaces

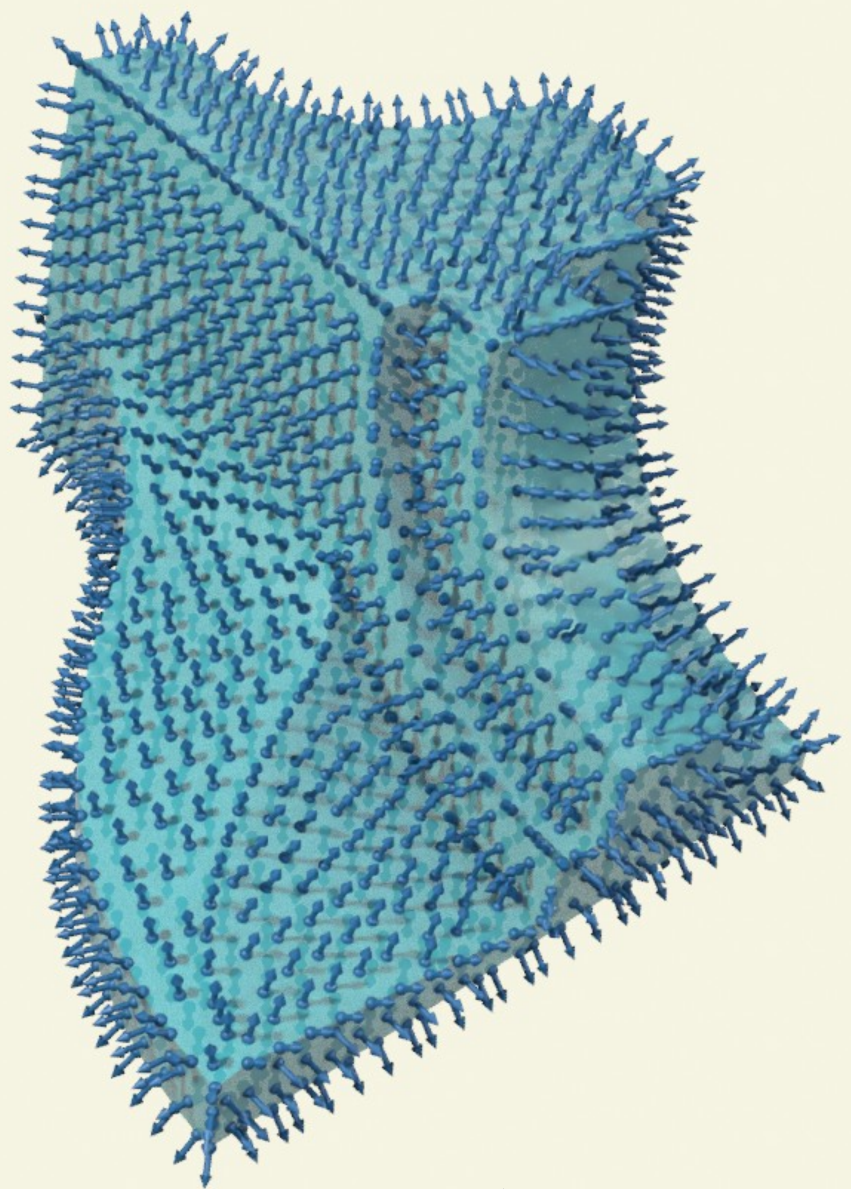
[**Gillespie**, Yang, Botch & Crane. 2024.
Ray Tracing Harmonic Functions. ACM Trans. Graph.]

Best Paper (Honorable Mention)

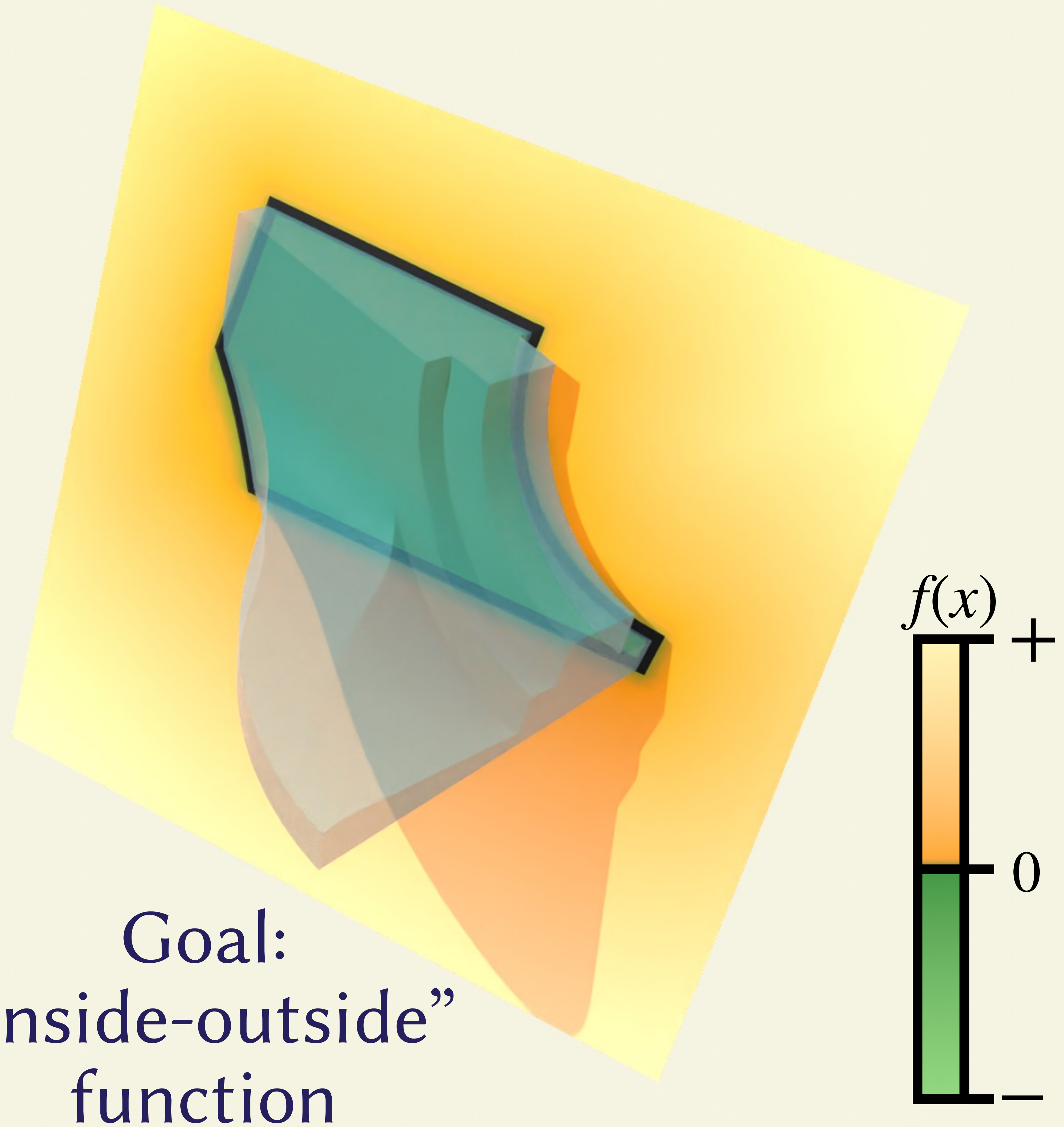


Motivation: Poisson reconstruction

[Kazhdan *et al.* 2006]



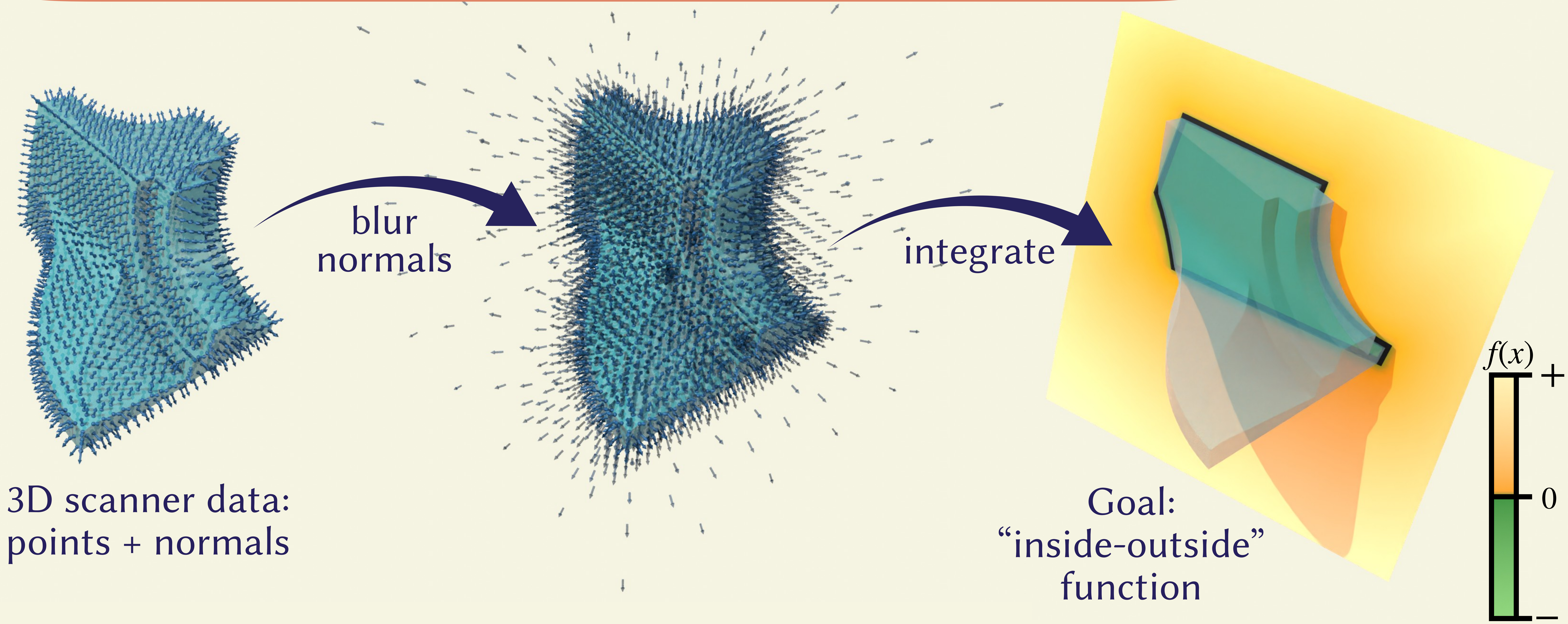
3D scanner data:
points + normals



Goal:
“inside-outside”
function

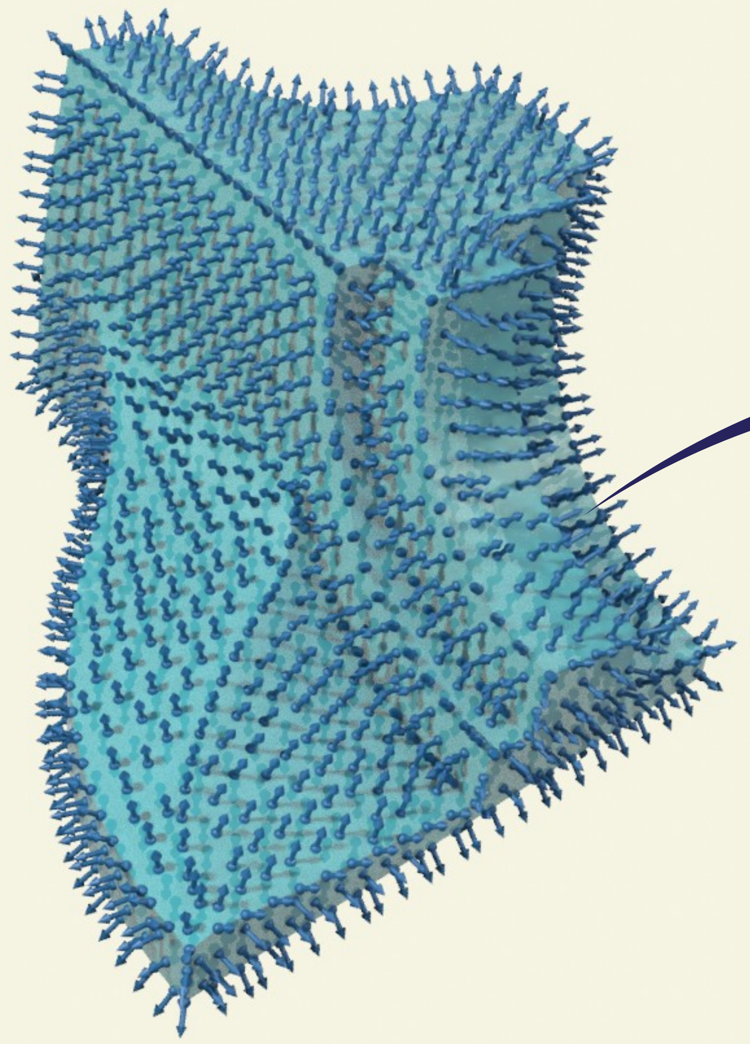
Motivation: Poisson reconstruction

[Kazhdan *et al.* 2006]



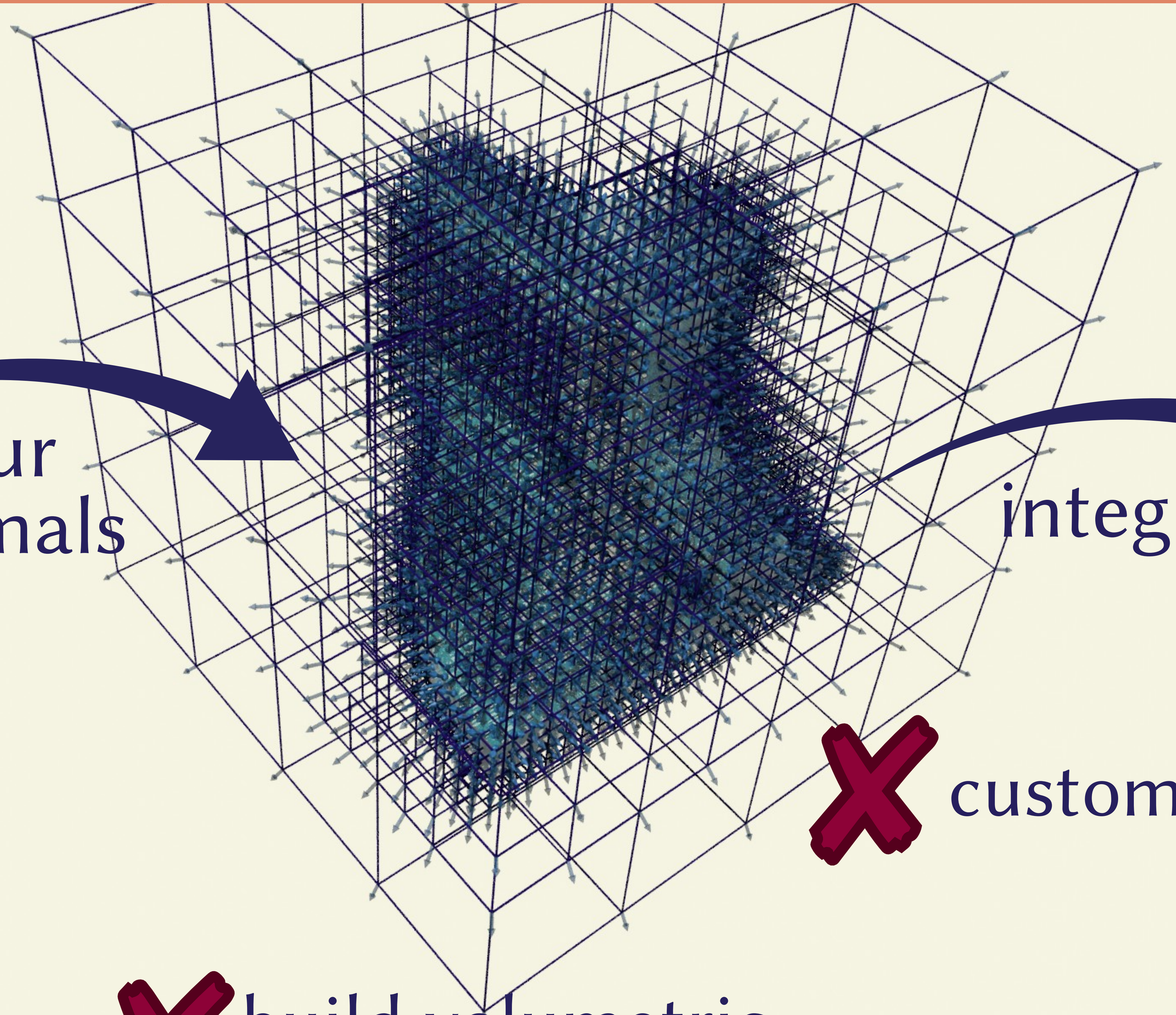
Motivation: Poisson reconstruction

[Kazhdan *et al.* 2006]



3D scanner data:
points + normals

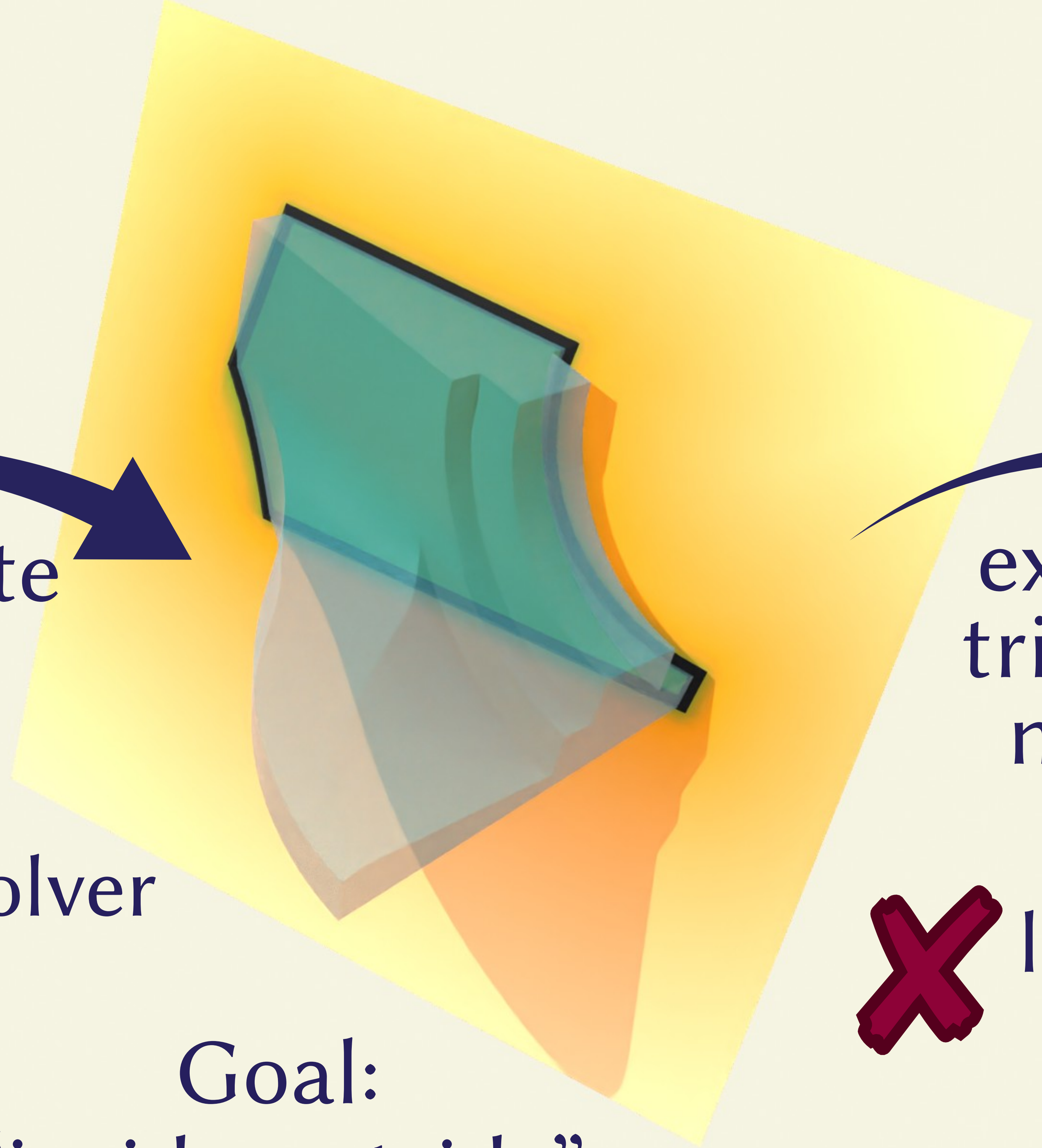
blur
normals



build volumetric
mesh

integrate

custom solver



Goal:
“inside-outside”
function

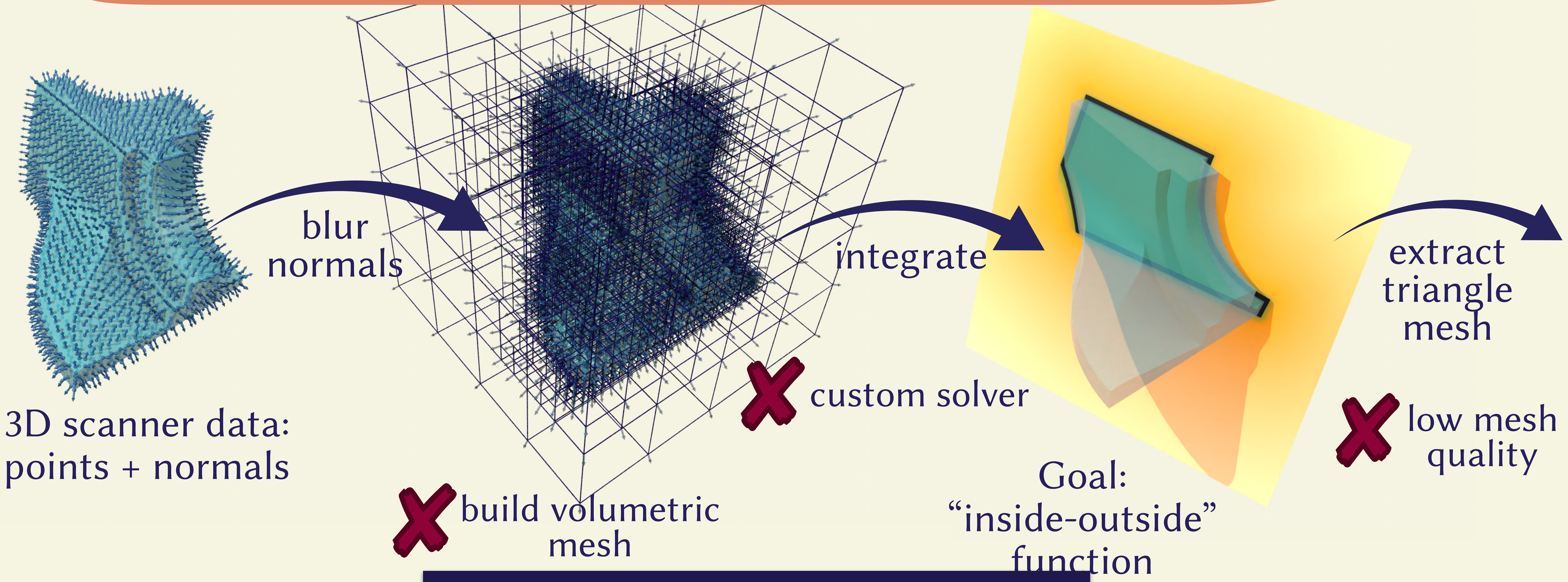
extract
triangle
mesh

low mesh
quality

in practice, very complicated

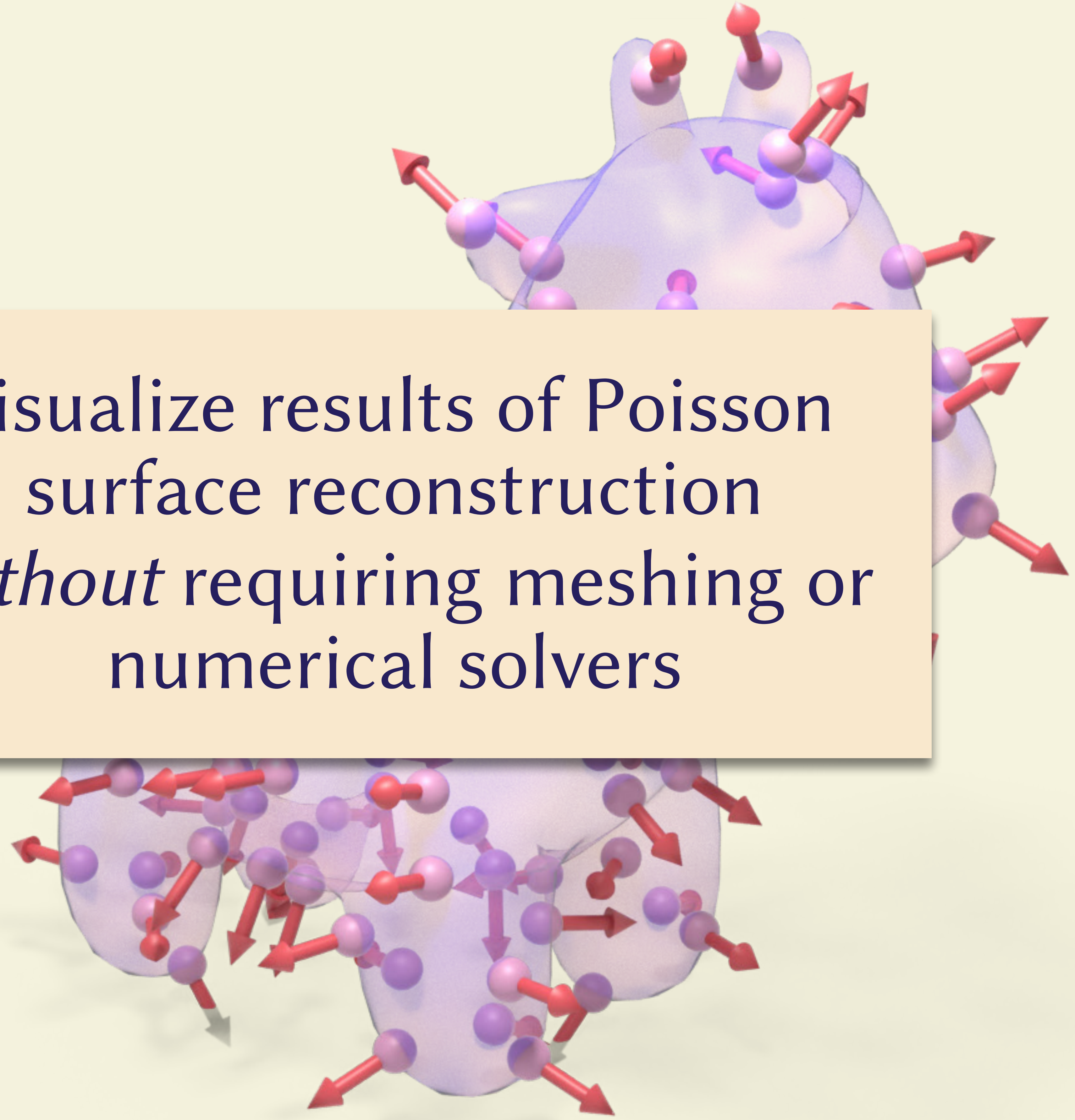
Motivation: Poisson reconstruction

[Kazhdan *et al.* 2006]



in practice, very complicated

Skipping straight to the answer



visualize results of Poisson surface reconstruction *without* requiring meshing or numerical solvers

(directly ray traced)

Harmonic functions

special kind of function

heat transfer

electrostatics

$$\boxed{\Delta f := \sum_i \frac{\partial^2 f}{\partial x_i^2} = 0}$$

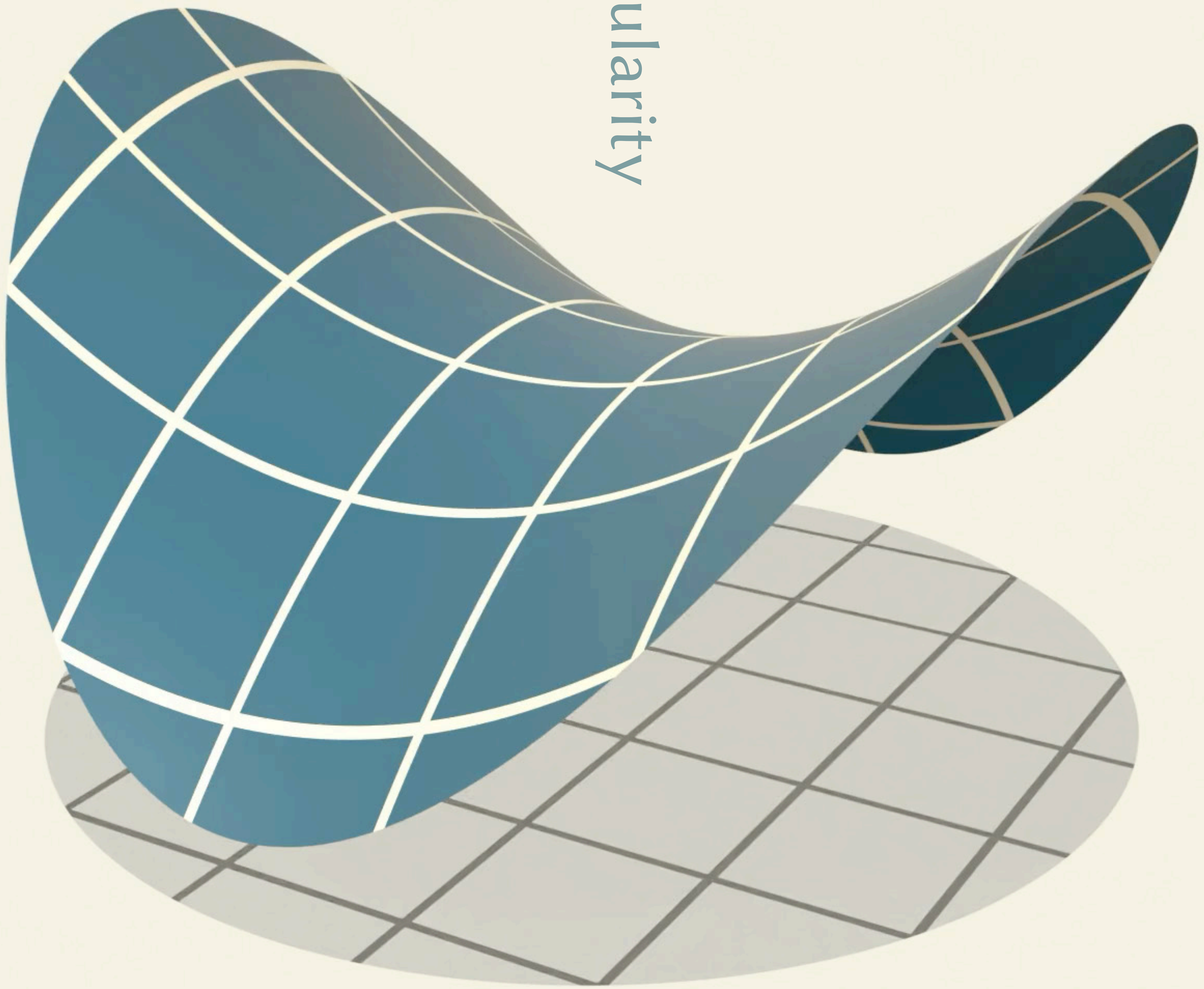
gravitation

complex analysis

well-studied

Harmonic functions

$$\boxed{\Delta f := \sum_i \frac{\partial^2 f}{\partial x_i^2} = 0}$$

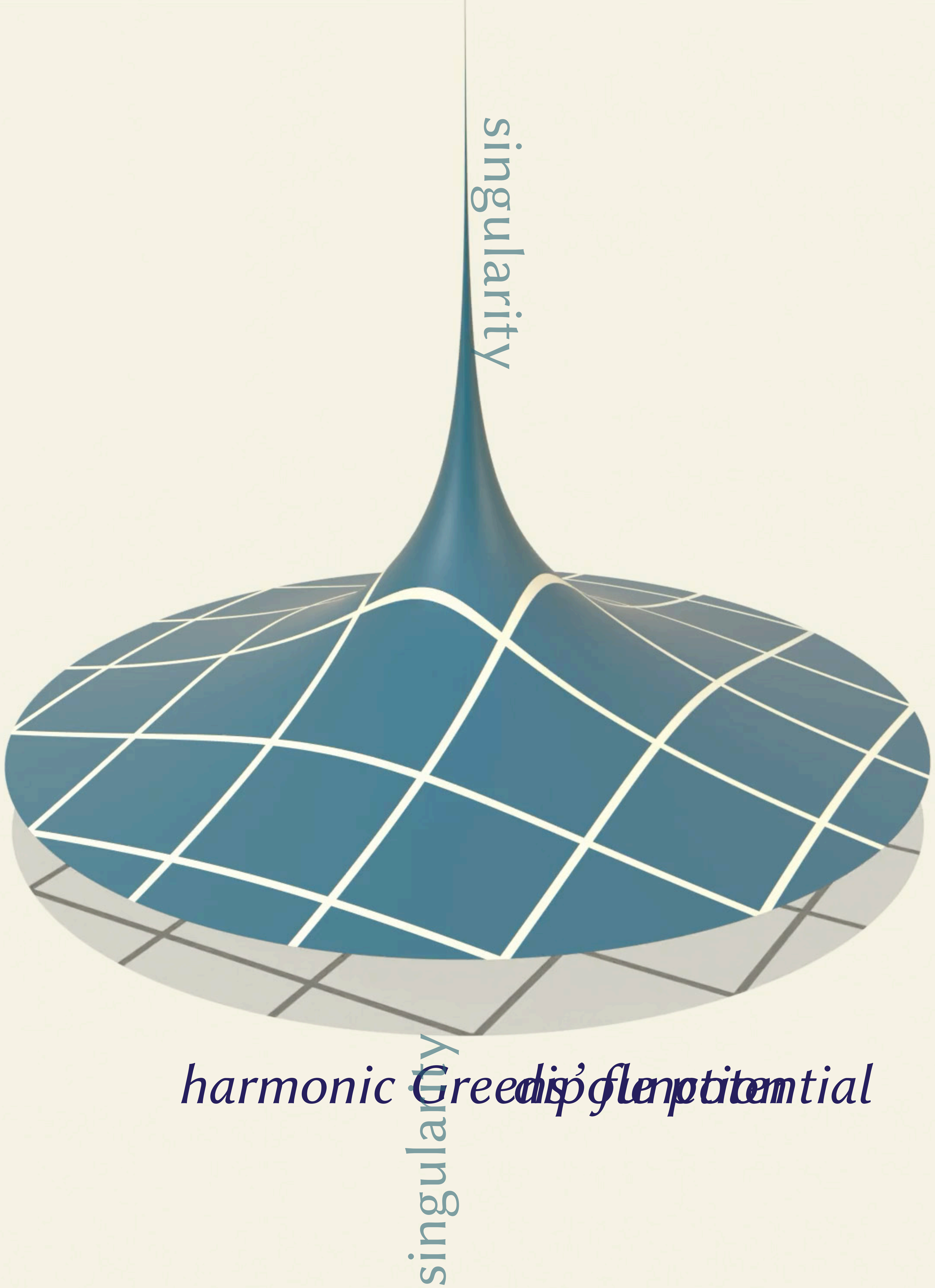


harmonic Greens function

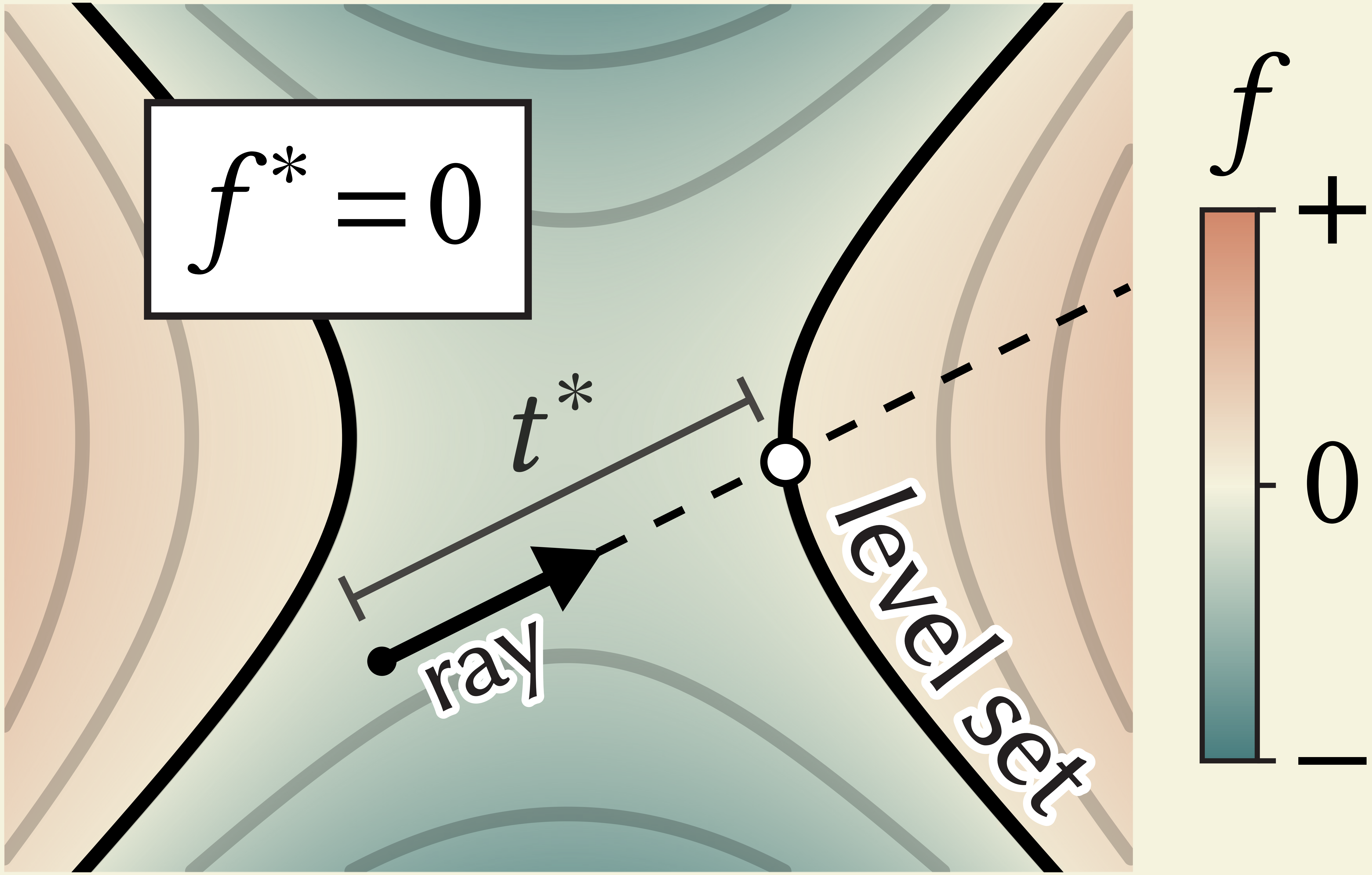
$f(x, y) = x^2 - y^2$

Harmonic functions

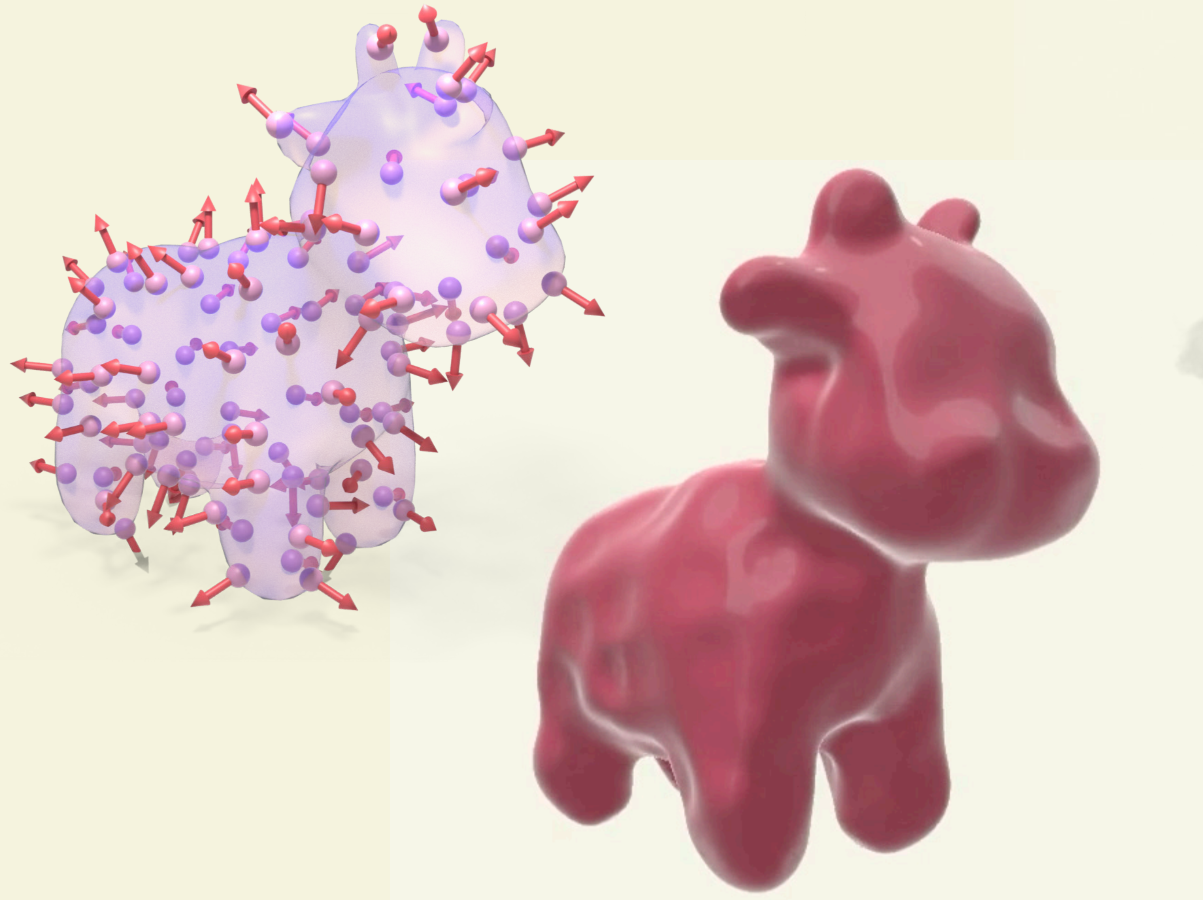
$$\boxed{\Delta f := \sum_i \frac{\partial^2 f}{\partial x_i^2} = 0}$$



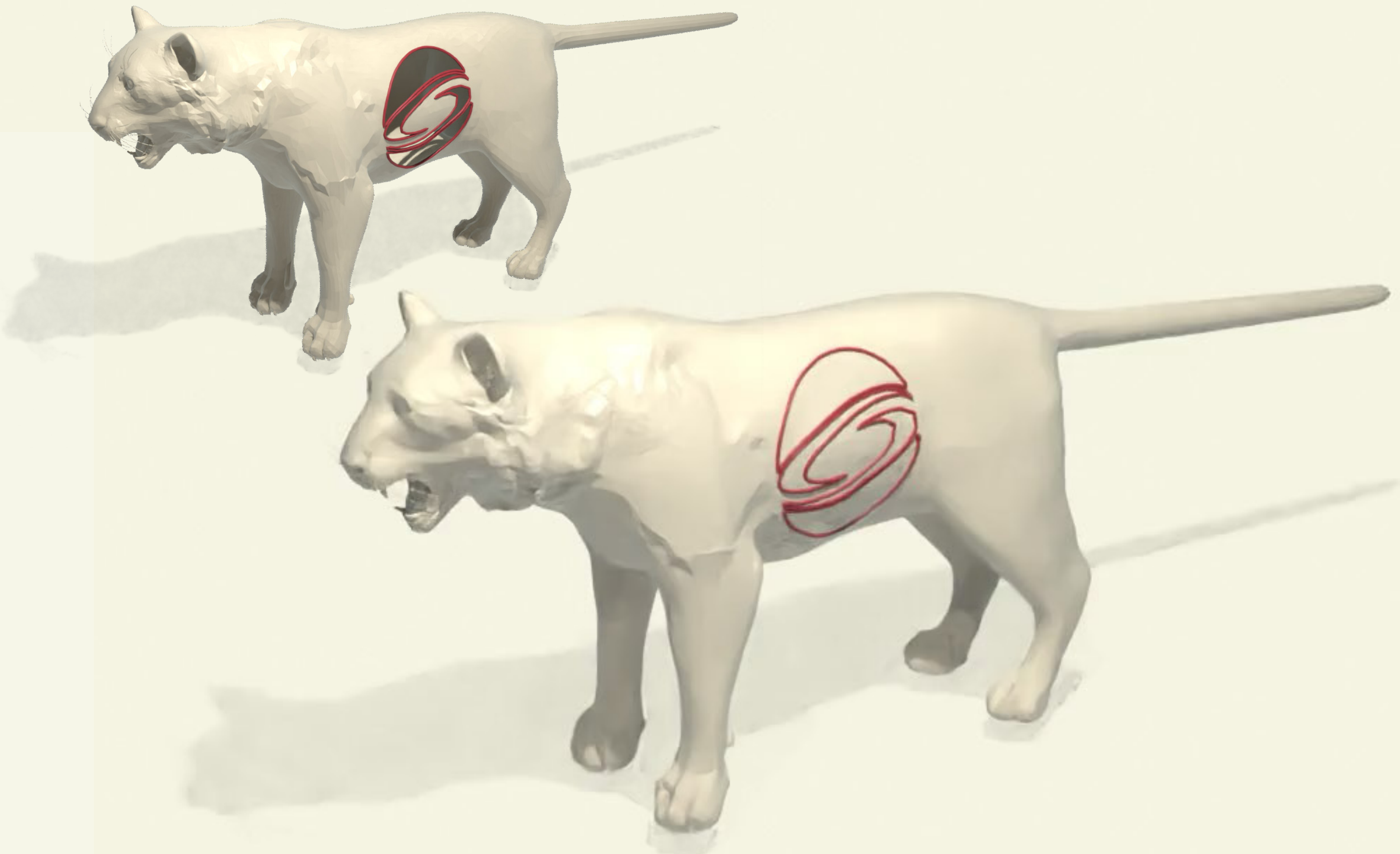
Intersecting a ray with a level set



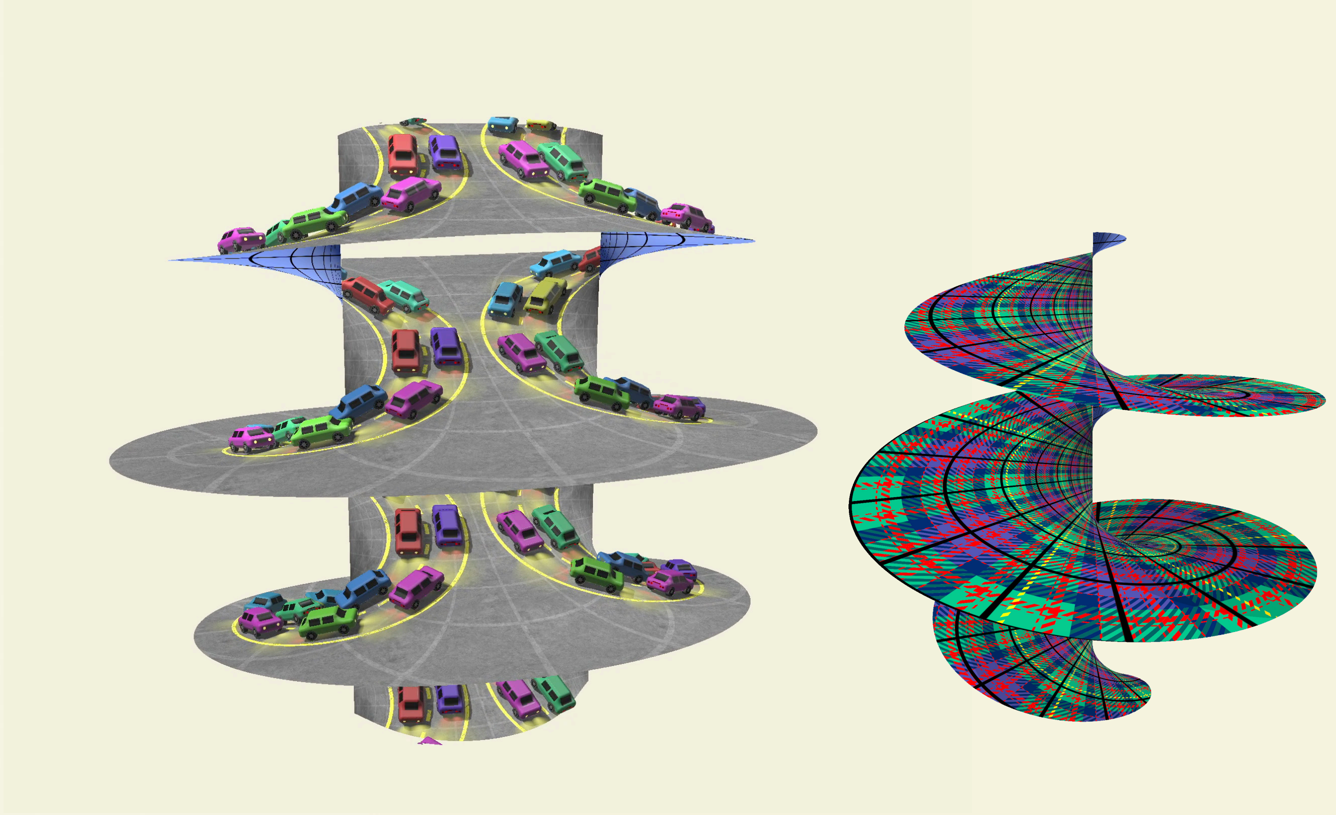
Level sets of harmonic functions show up everywhere



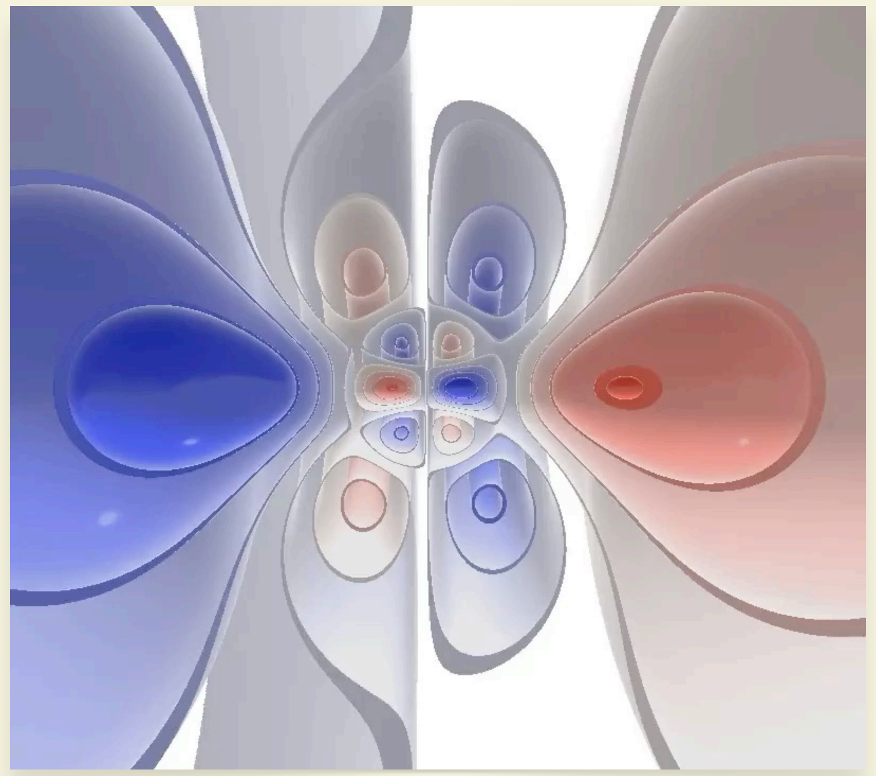
Poisson surface reconstruction
[Kazhdan *et al.* 2006]



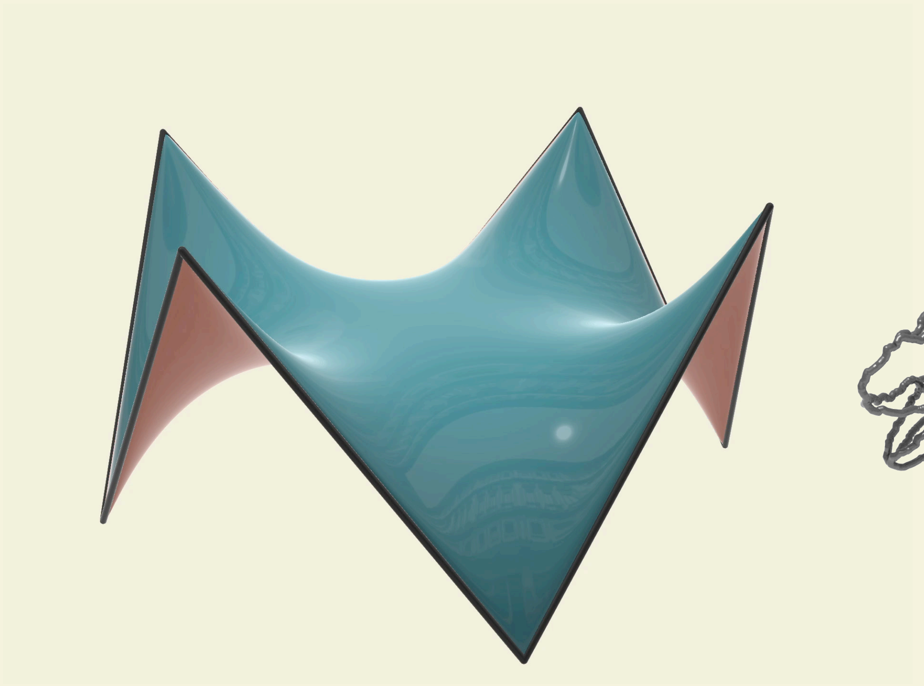
generalized winding numbers
[Jacobson *et al.* 2013]



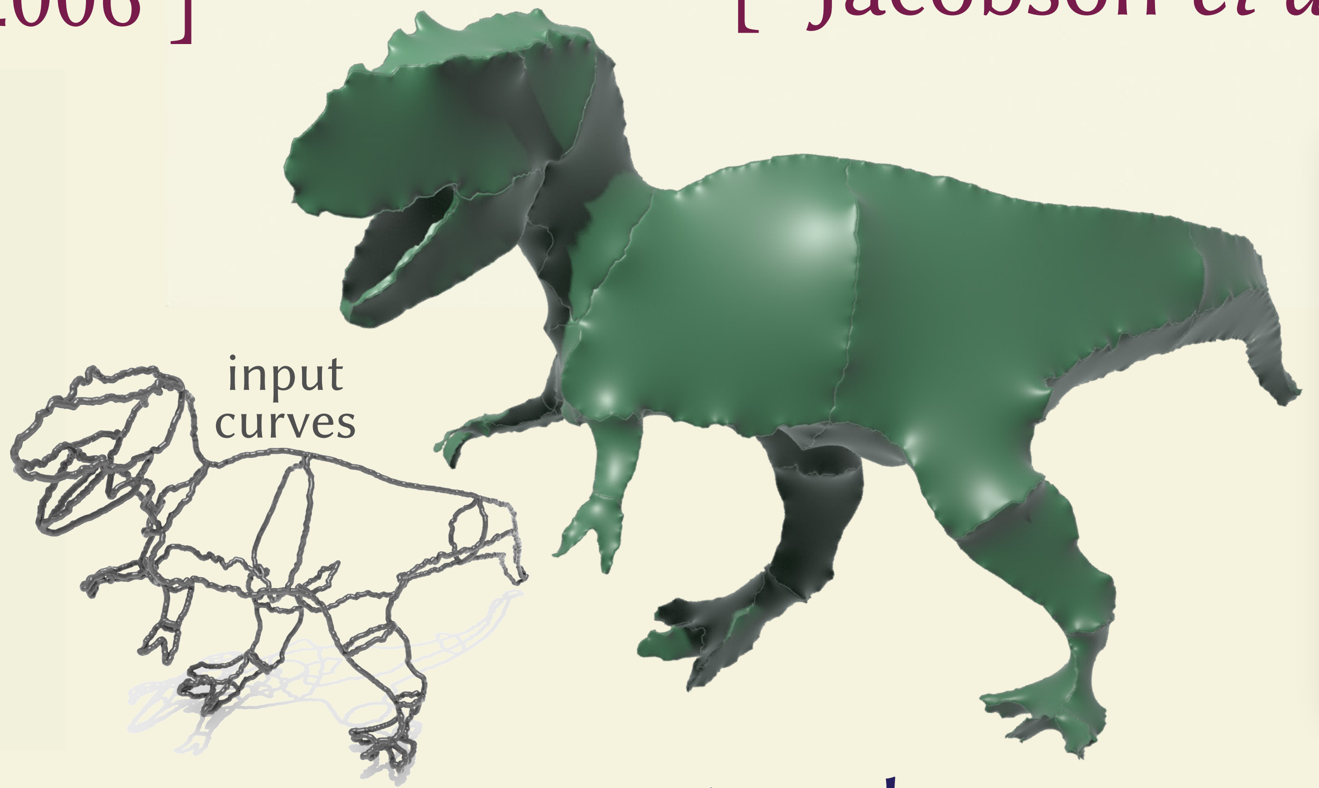
Riemann surfaces
[Riemann 1851]



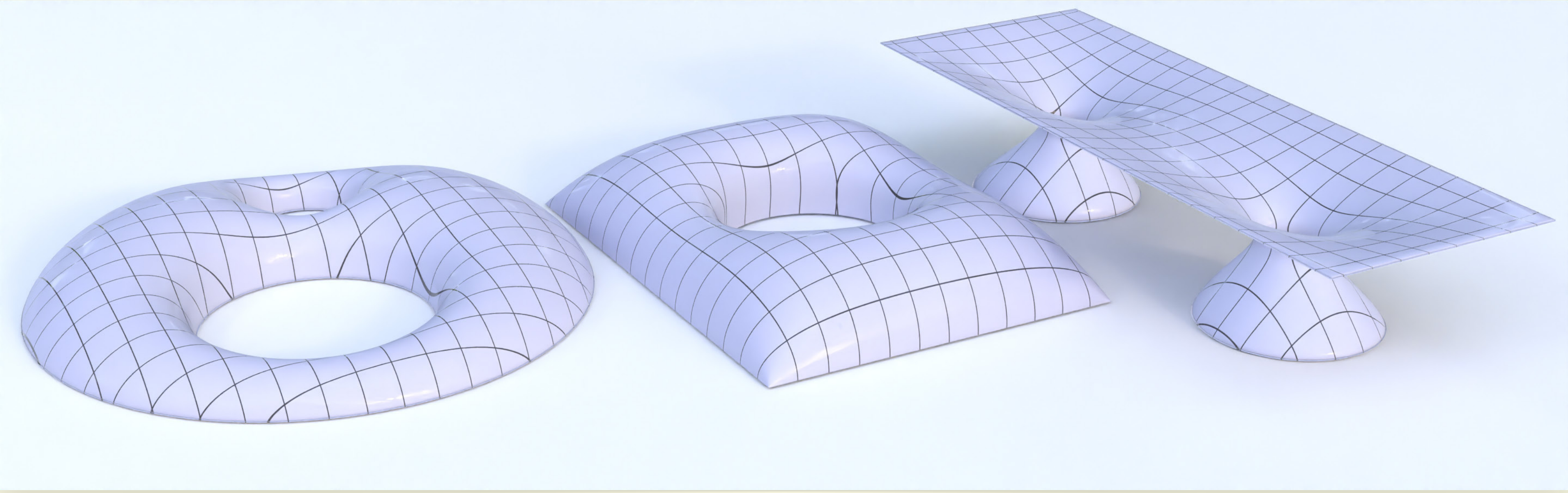
hyperspherical harmonics
[Fock 1935]



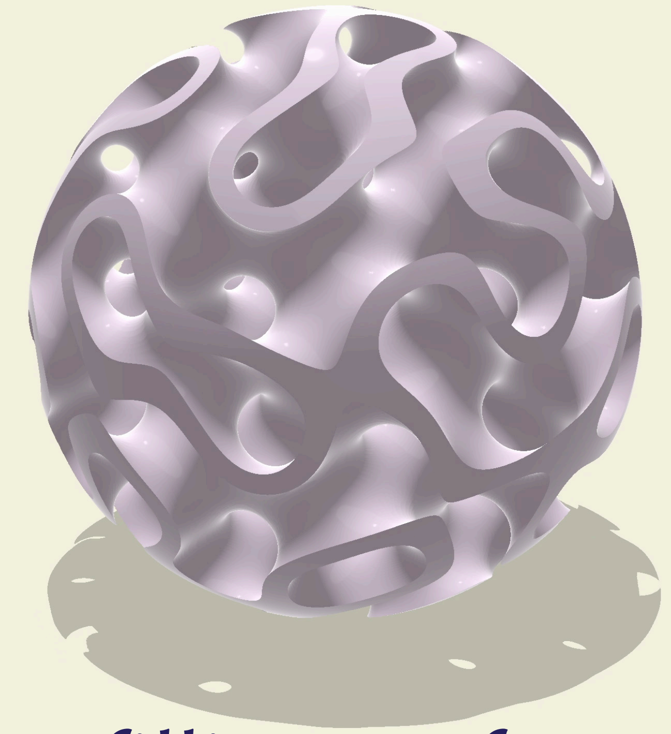
nonplanar polygons
[Maxwell 1873]



curve networks
[de Goes *et al.* 2011]



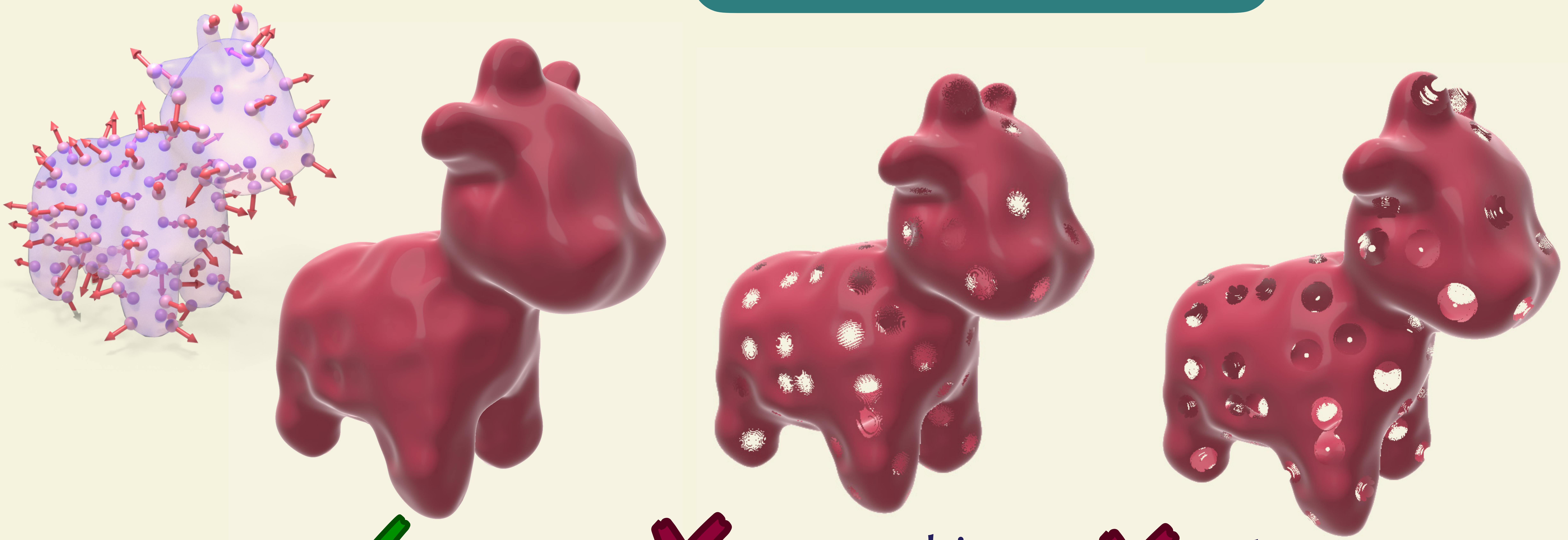
shell structures in architectural geometry
[Adiels *et al.* 2022]



space-filling surfaces for
digital fabrication

… but, they're hard to render with existing techniques

may have singularities



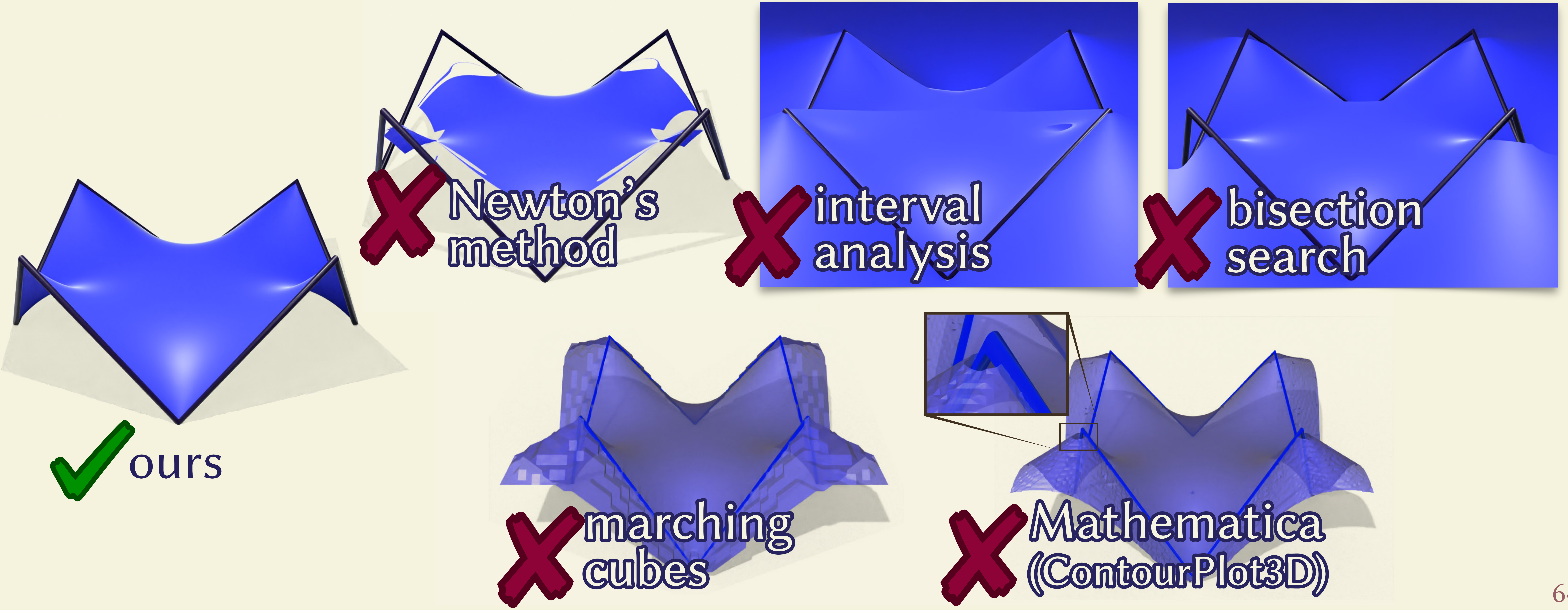
✓ ours

✗ ray marching
(with fixed step size)

✗ sphere tracing
(with purported Lipschitz constant)

… but, they're hard to render with existing techniques

may have boundaries



Sphere tracing

[Hart 1996]

compute intersections for *signed distance functions (SDFs)*

Sphere tracing

[Hart 1996]

compute intersections for *signed distance functions (SDFs)*

Sphere tracing: beyond SDFs

[Hart 1996]

- Easy to generalize to *Lipschitz* functions:

  (essentially, $|\nabla f| \leq L$)

- Important fact:

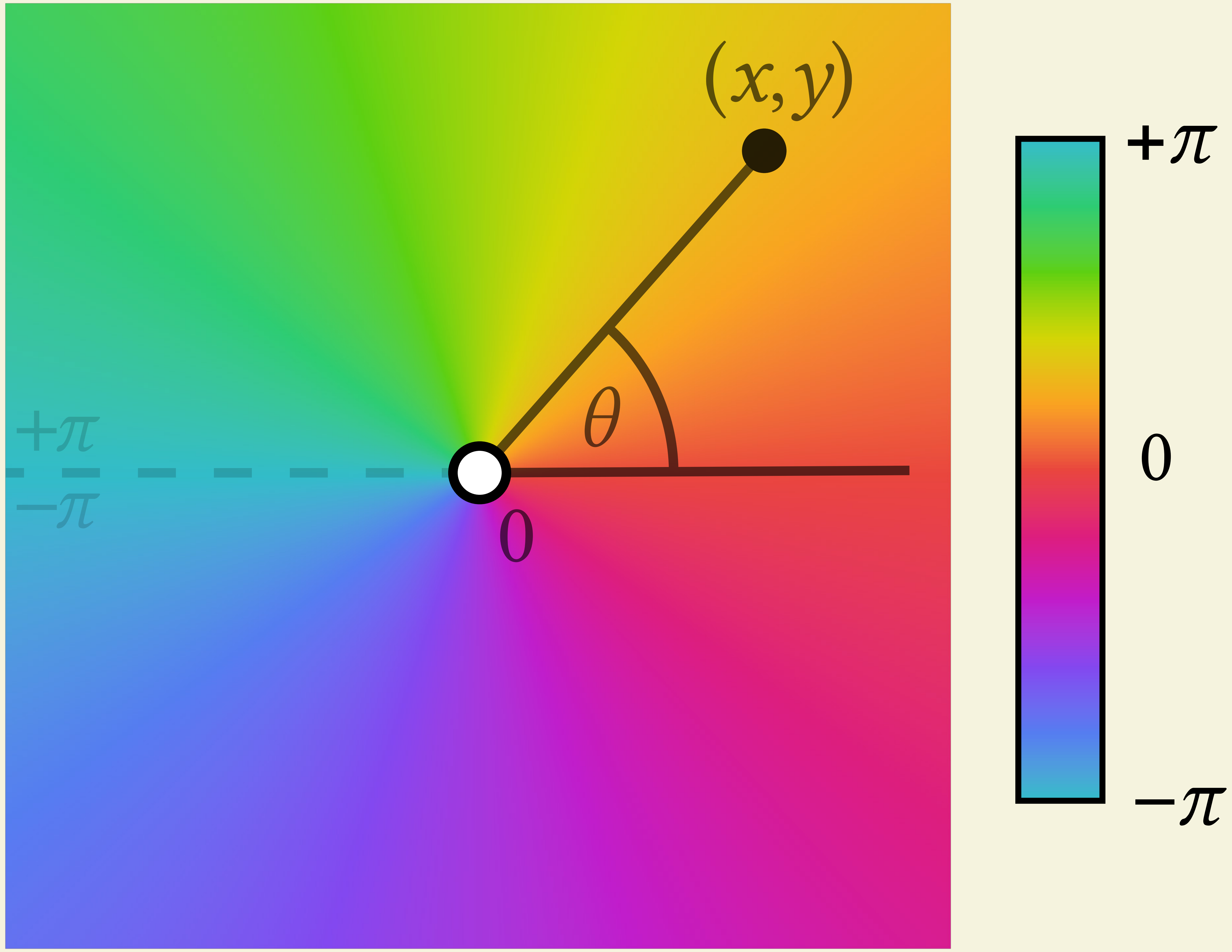
  $|f(x) - f(y)| \leq L|x - y|$

- provides a conservative bound on distance

[Inigo Quilez 2015]

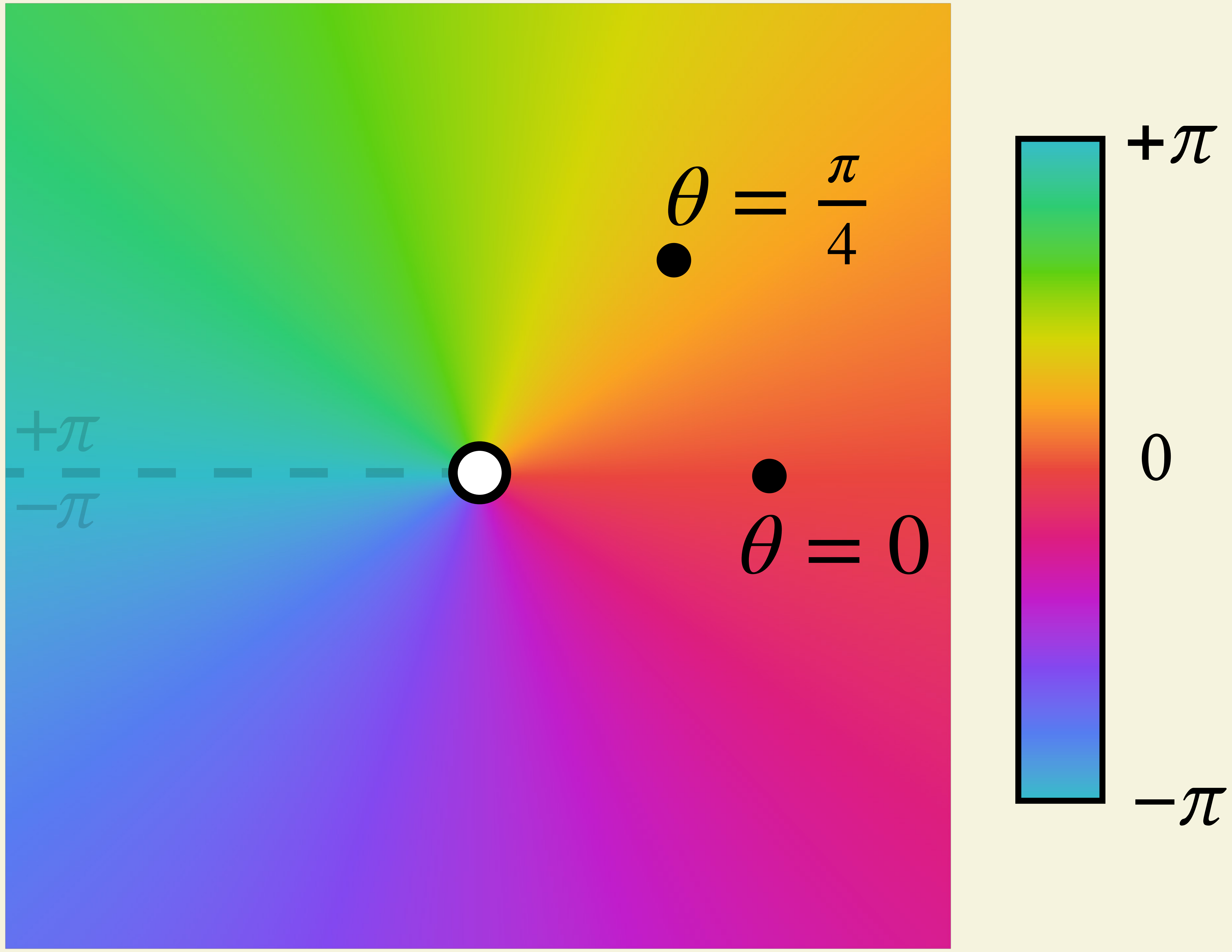
Problem: many harmonic functions are not Lipschitz

$$\theta(x, y) = \mathrm{atan2}(y, x)$$



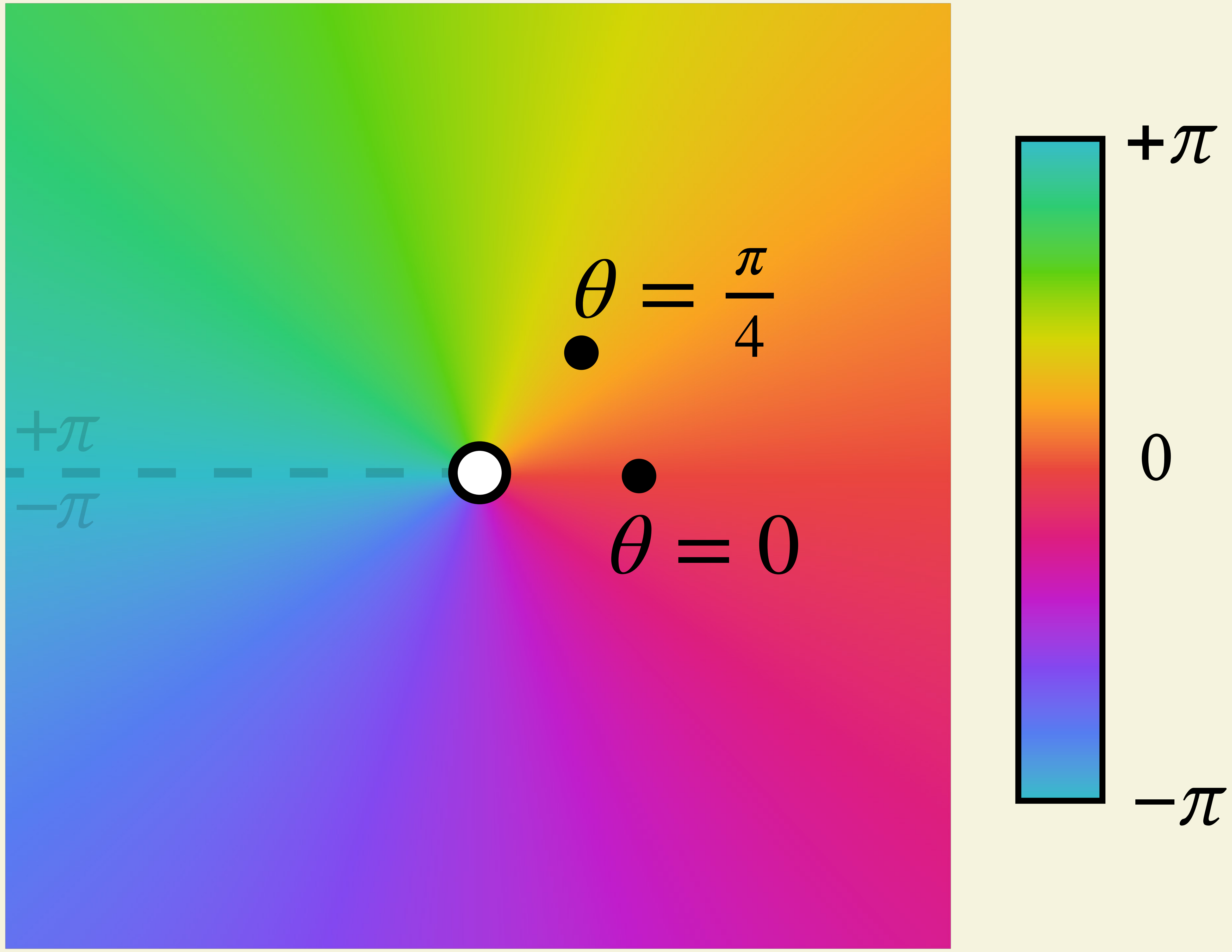
Problem: many harmonic functions are not Lipschitz

$$\theta(x, y) = \mathrm{atan2}(y, x)$$



Problem: many harmonic functions are not Lipschitz

$$\theta(x, y) = \mathrm{atan2}(y, x)$$

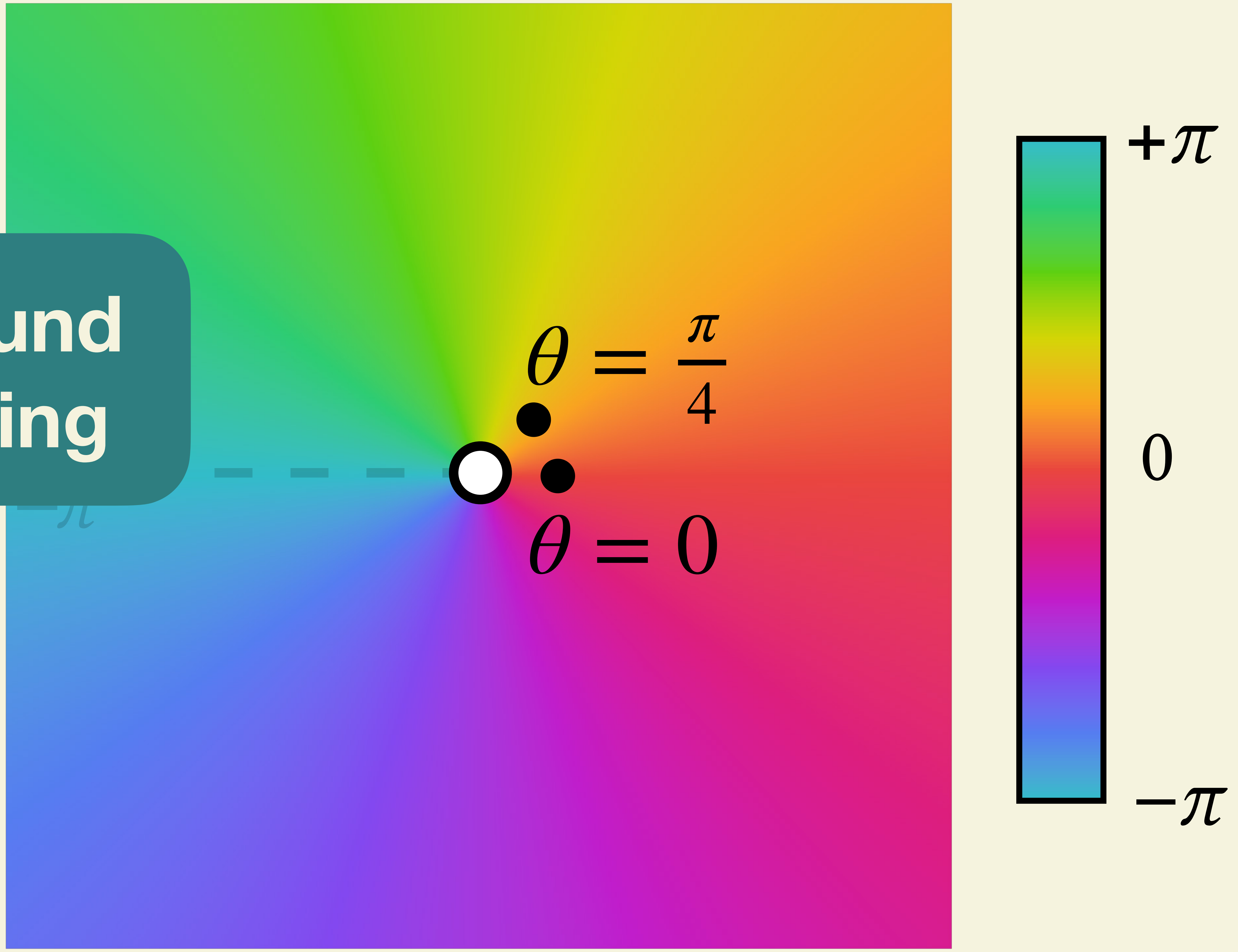


Problem: many harmonic functions are not Lipschitz

$$\theta(x, y) = \mathrm{atan2}(y, x)$$

No matter how close points get, function values never get closer

no distance bound for sphere tracing



Main idea: get distance bounds from Harnack's inequality

Let f be a positive harmonic function on a ball:

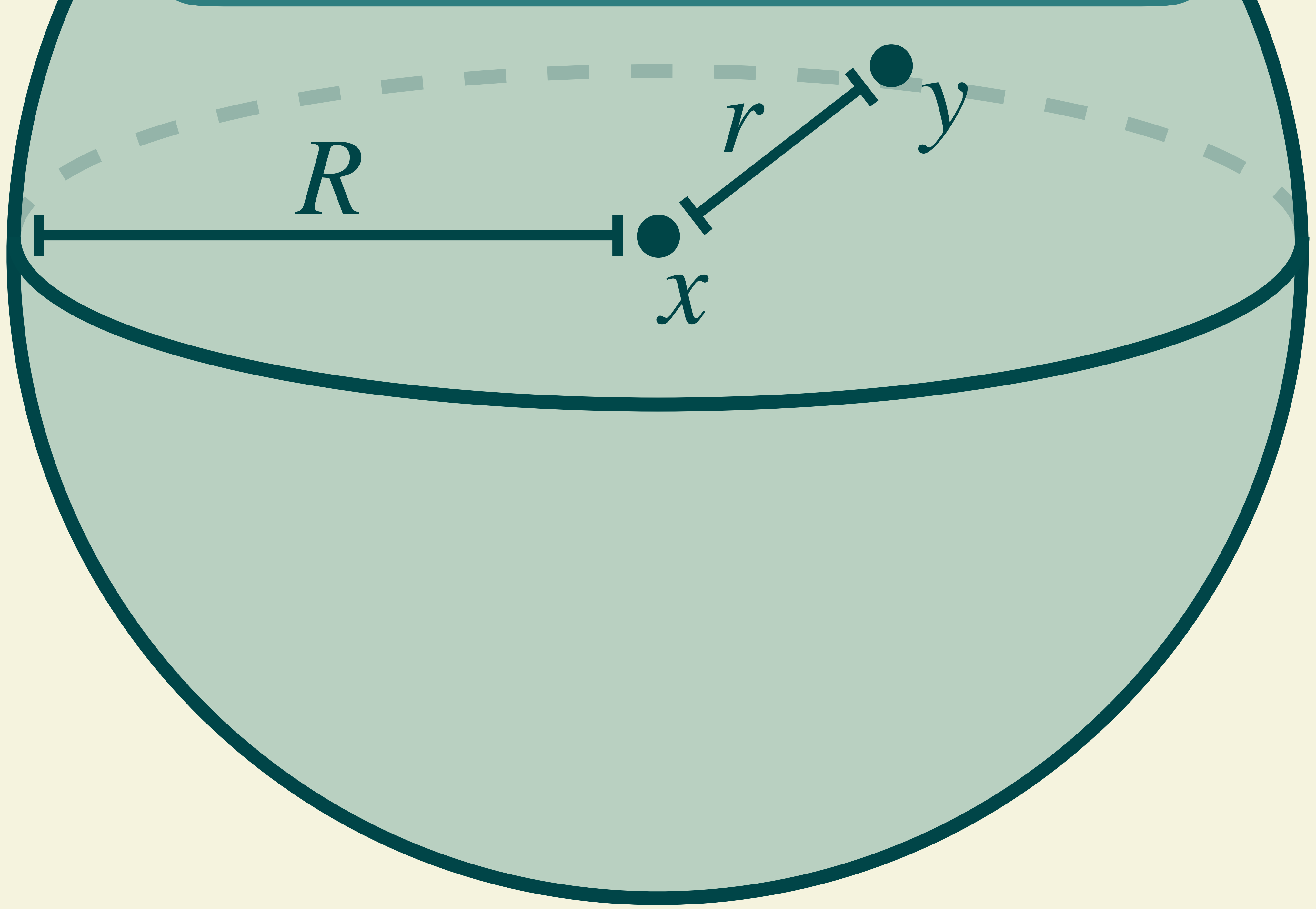
$$\underbrace{\frac{1 - r/R}{(1 + r/R)^2} f(x)}_{\text{lower bound}} \leq f(y) \leq \underbrace{\frac{1 + r/R}{(1 - r/R)^2} f(x)}_{\text{upper bound}}$$

always safe to take step of size

$$\frac{R}{2}\left|a + 2 - \sqrt{a^2 + 8a}\right|,$$

where $a = \dfrac{f(x)}{f^*}$

We can use the fact that f is harmonic to obtain a distance bound



Main idea: get distance bounds from Harnack's inequality

Let f be a positive harmonic function on a ball:

What if f is not positive?

Just add a constant to make it positive on the ball

lower bound

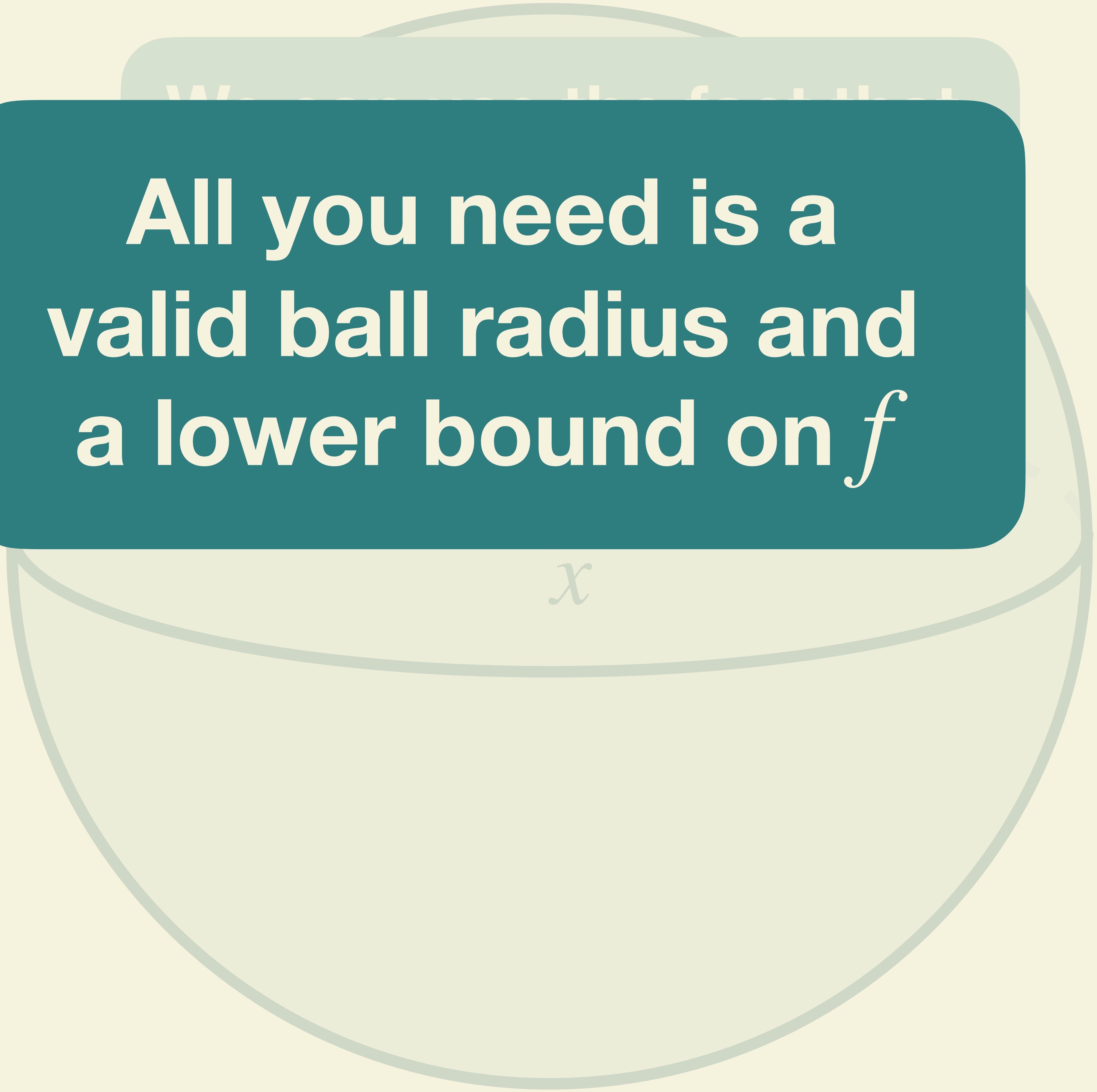
upper bound

All you need is a valid ball radius and a lower bound on f

always safe to take step of size

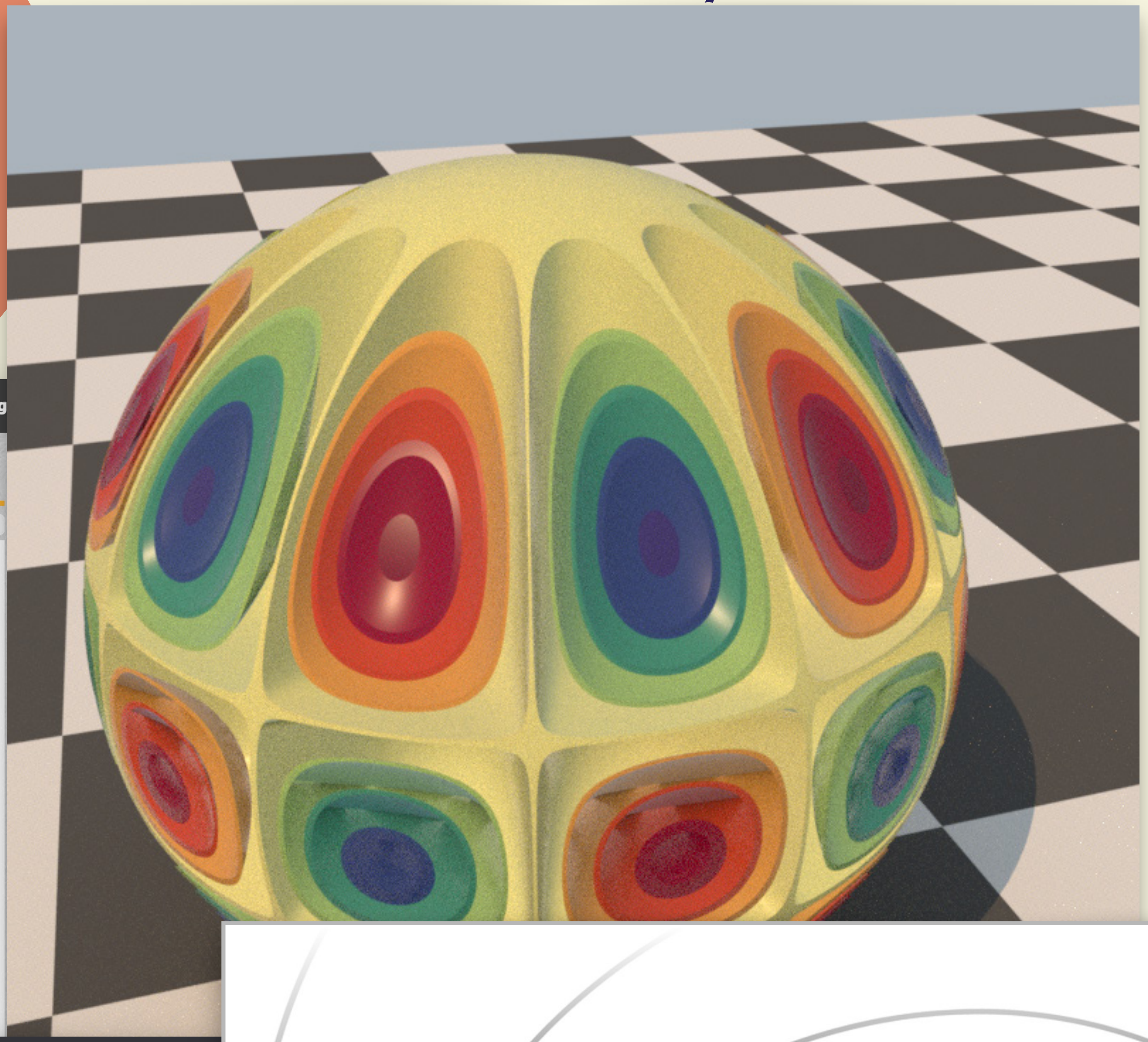
$$r = \frac{R}{2}\left|a + 2 - \sqrt{a^2 + 8a}\right|,$$

where $a = \frac{f(x)}{f^*}$

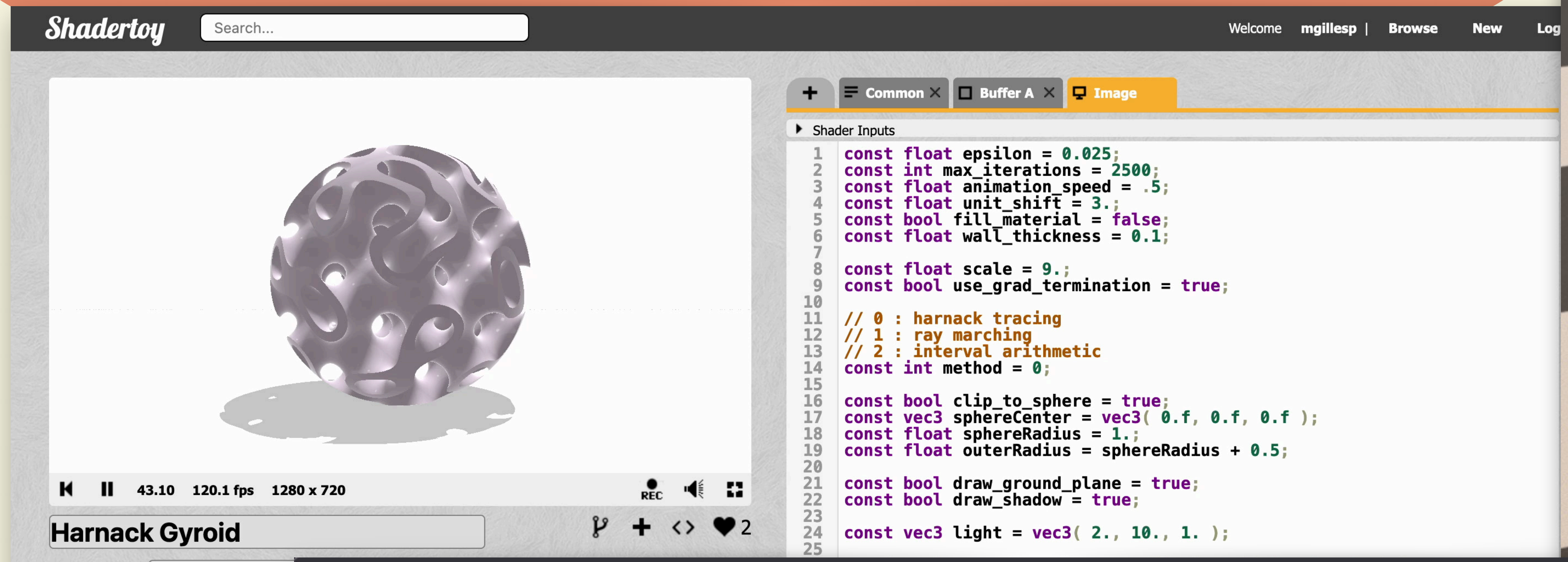


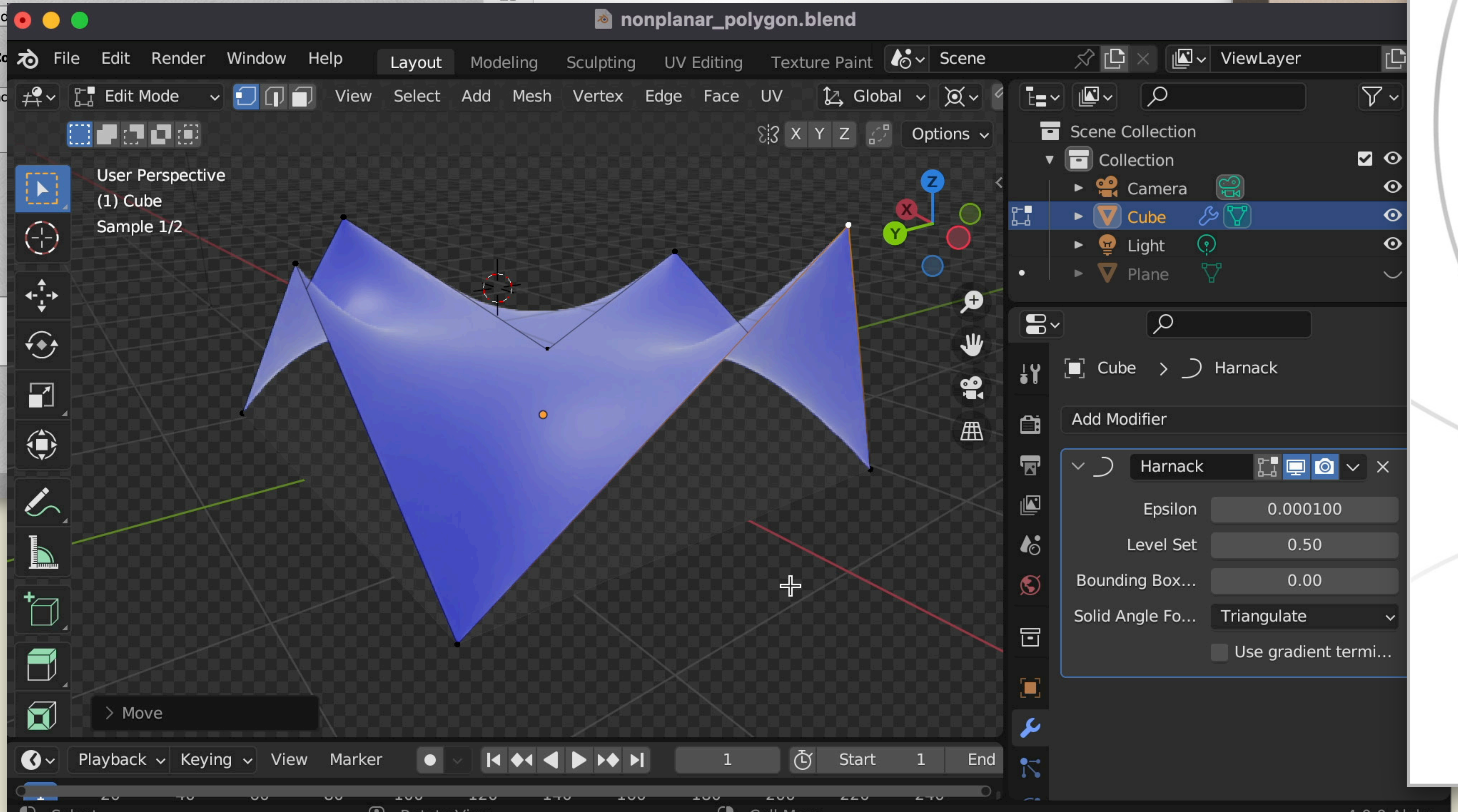
Simple to implement

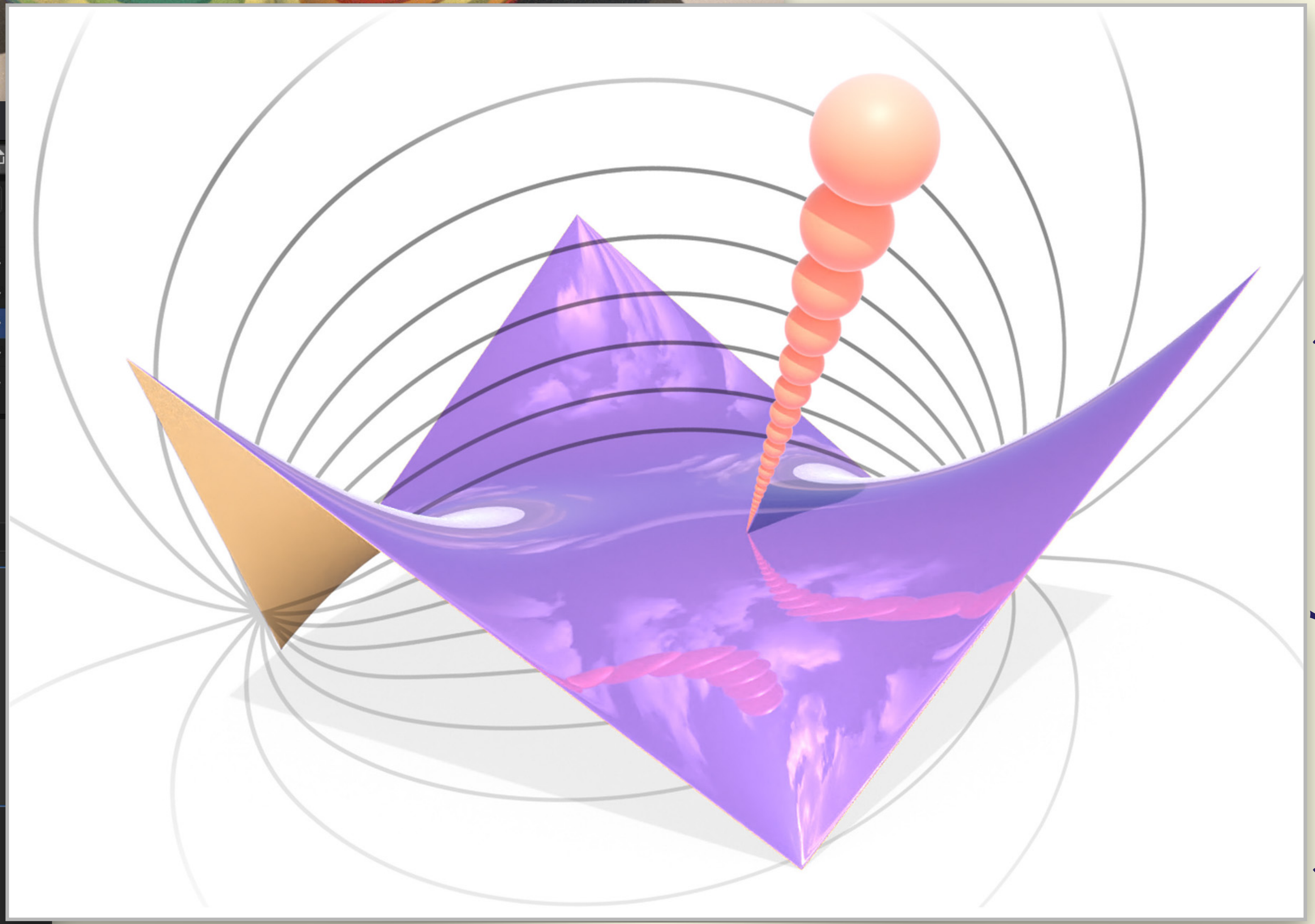
PBRT (CPU ray tracer)



ShaderToy (WebGL shaders)



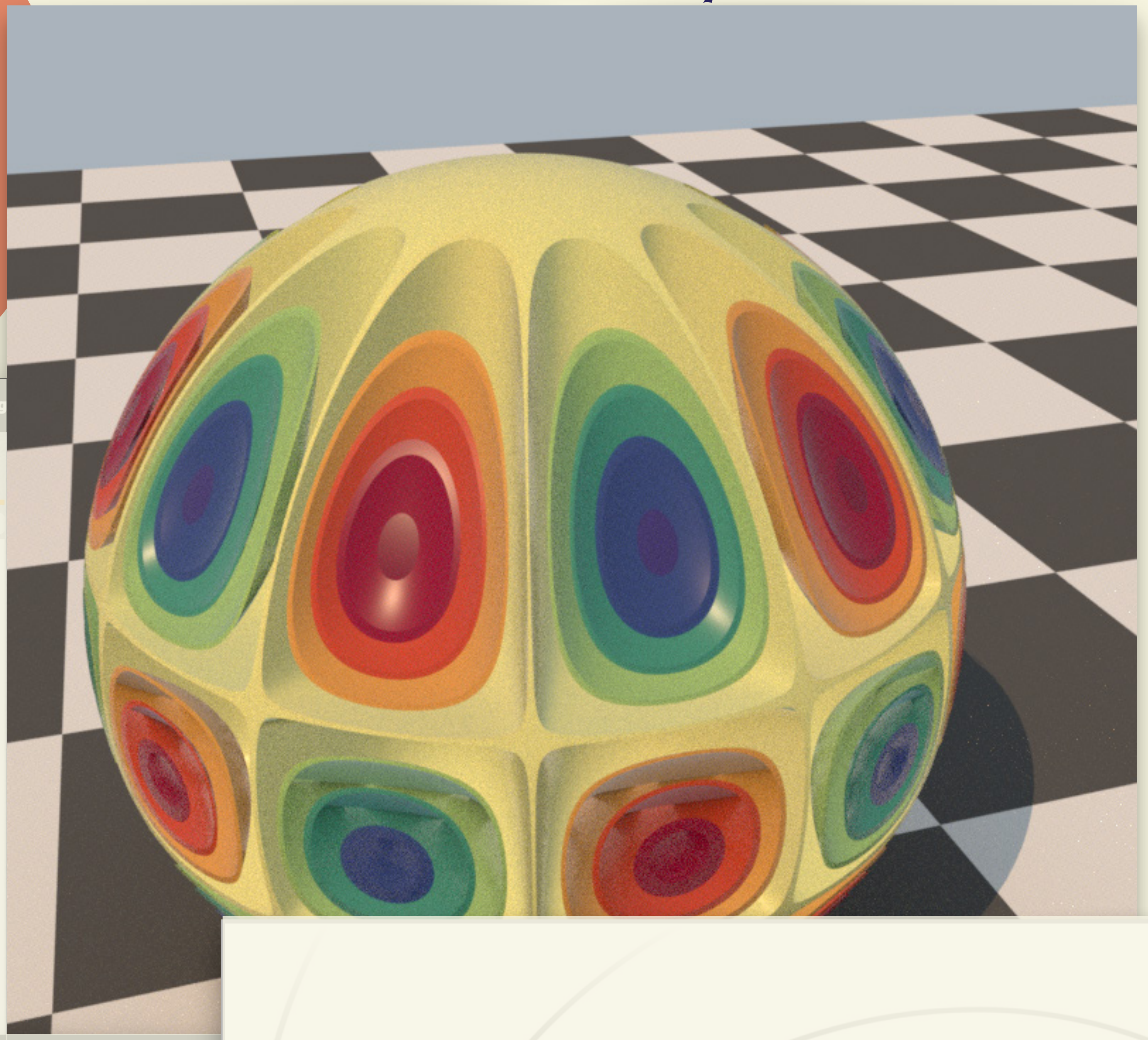


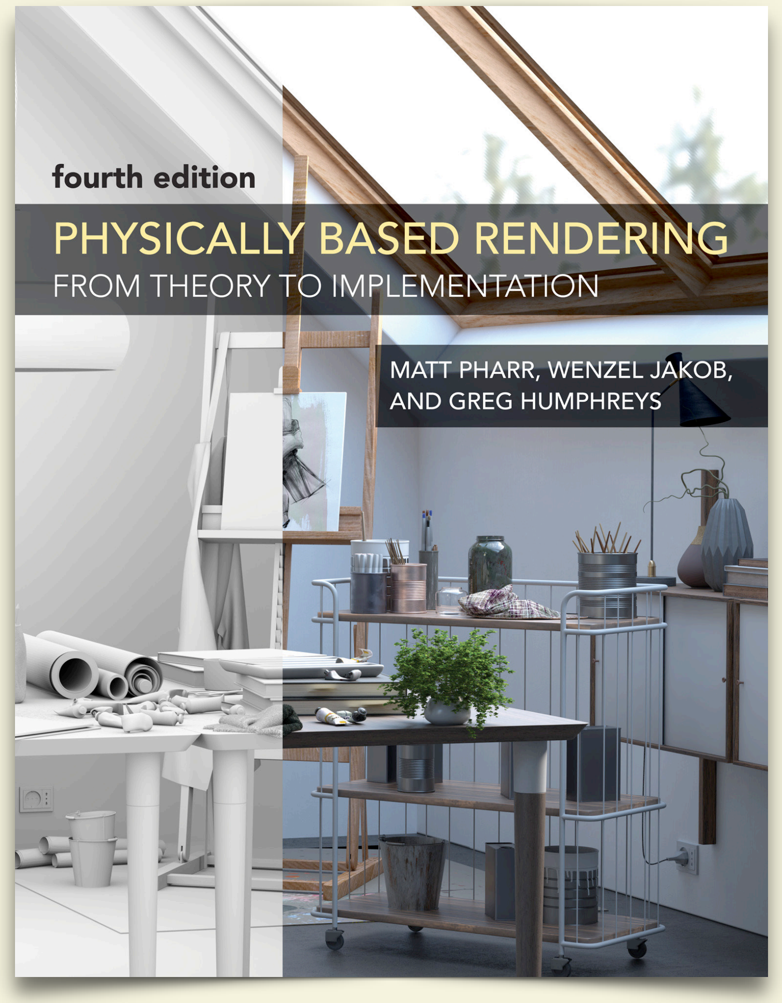


Blender (CPU ray tracer)

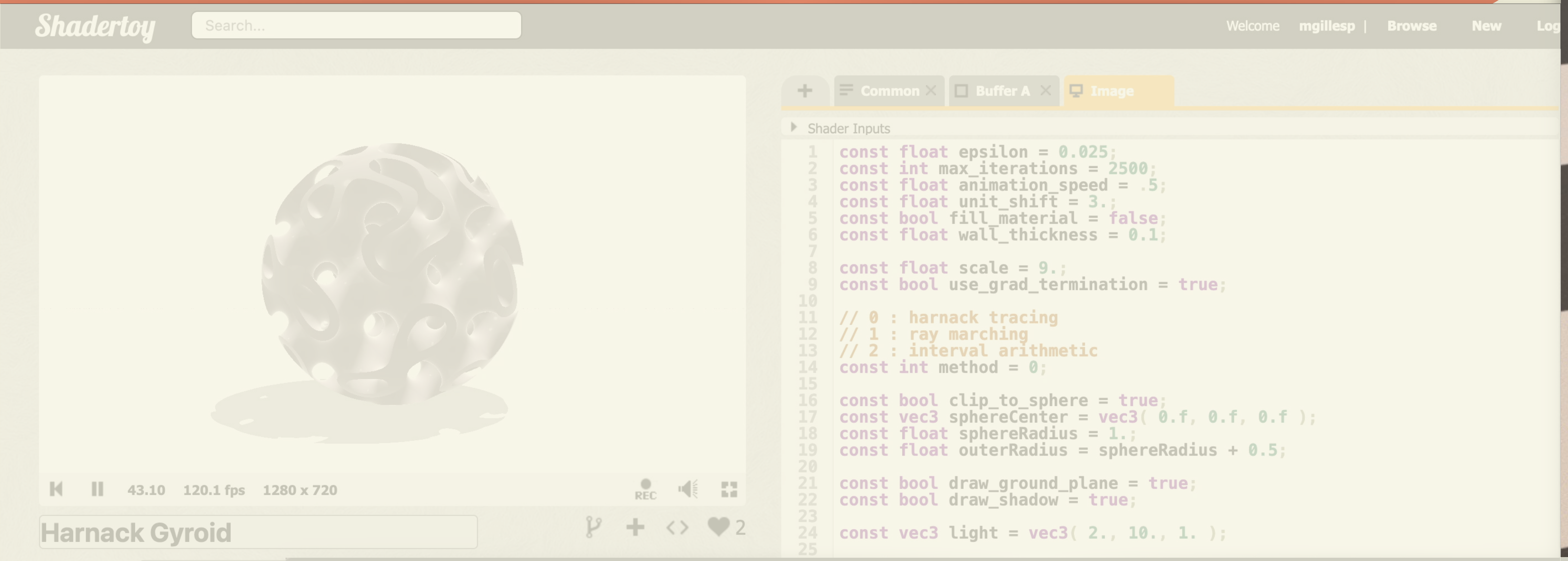
Simple to implement

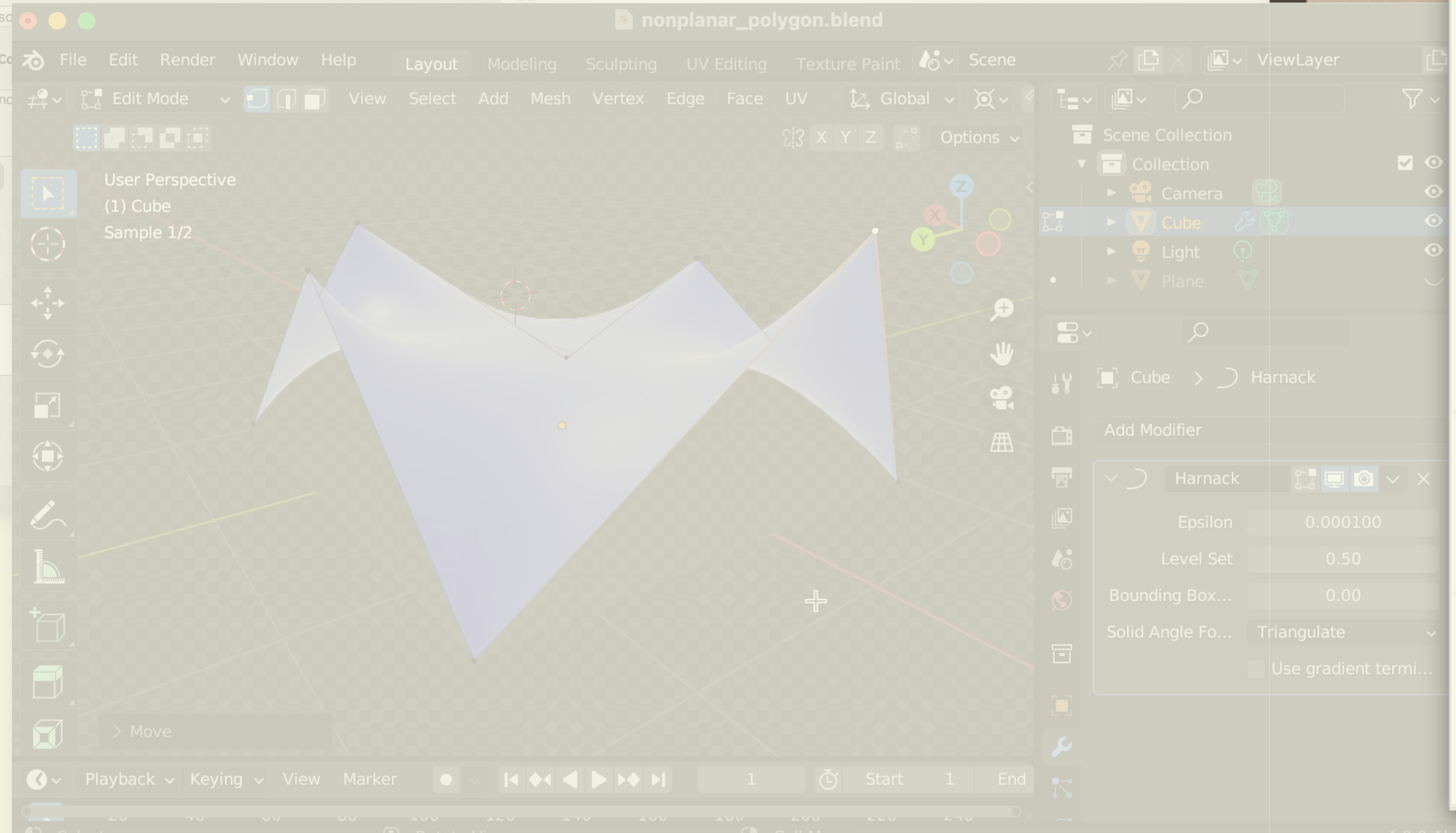
PBRT (CPU ray tracer)





ShaderToy (WebGL shaders)

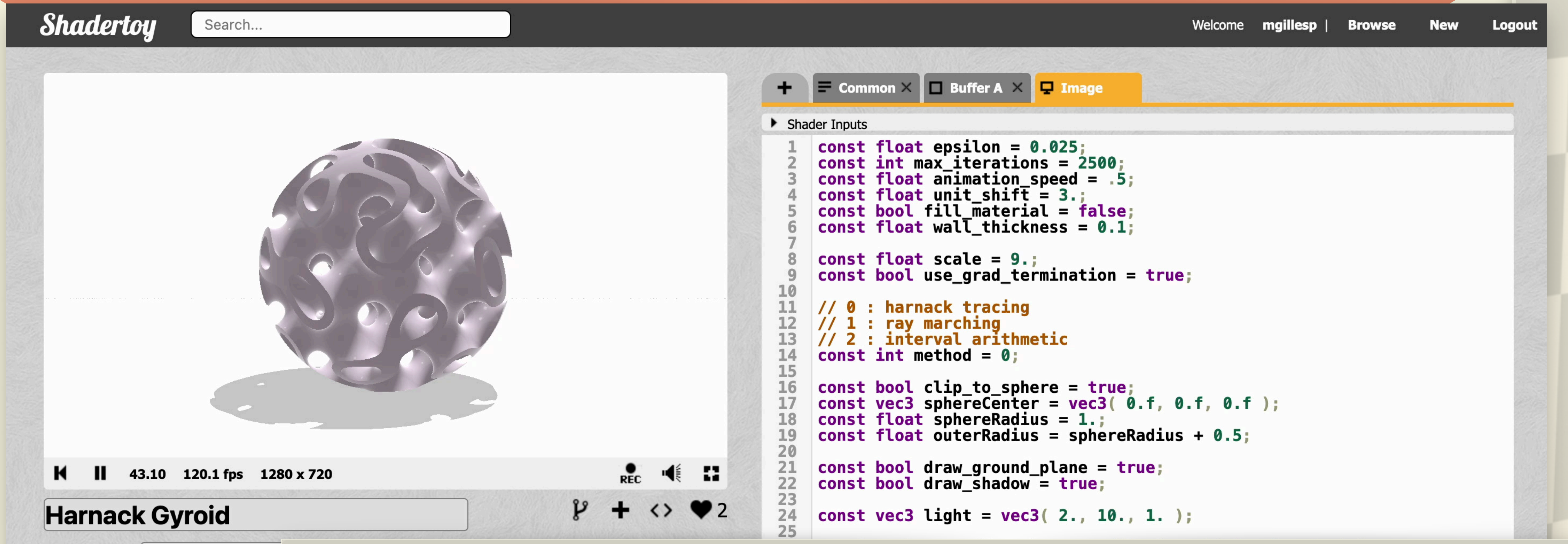


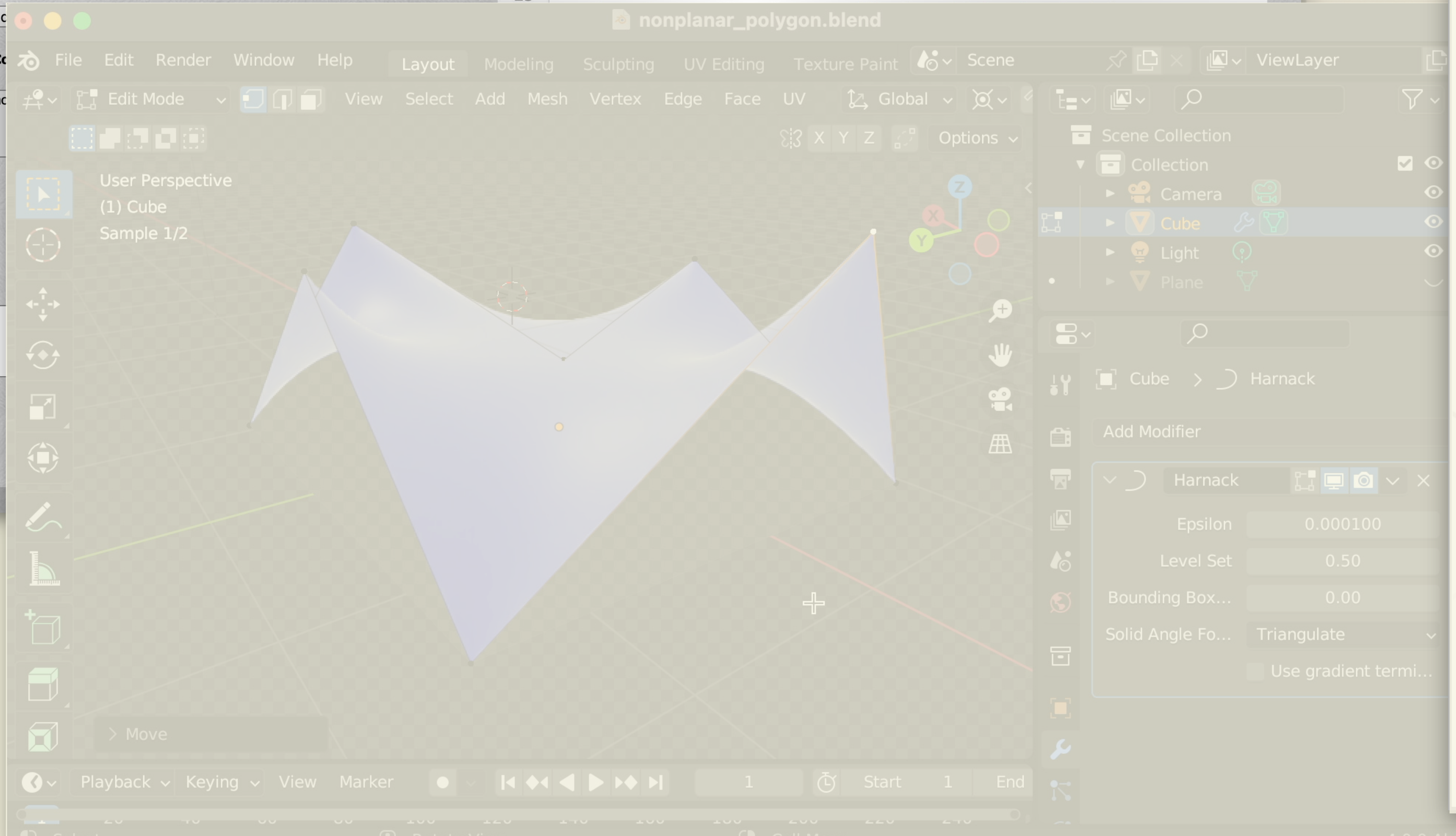


Blender (CPU ray tracer)

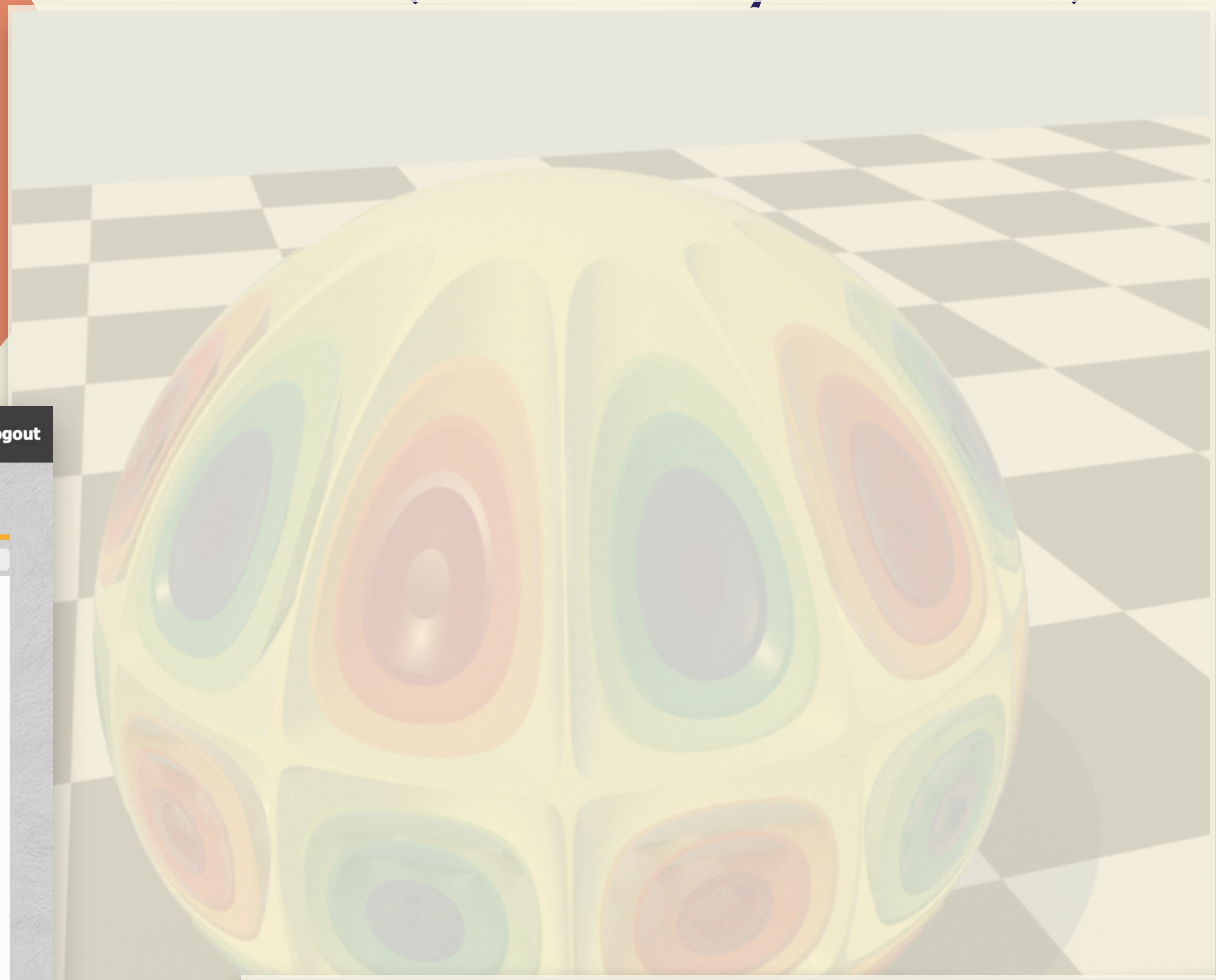
Simple to implement

ShaderToy (WebGL shaders)





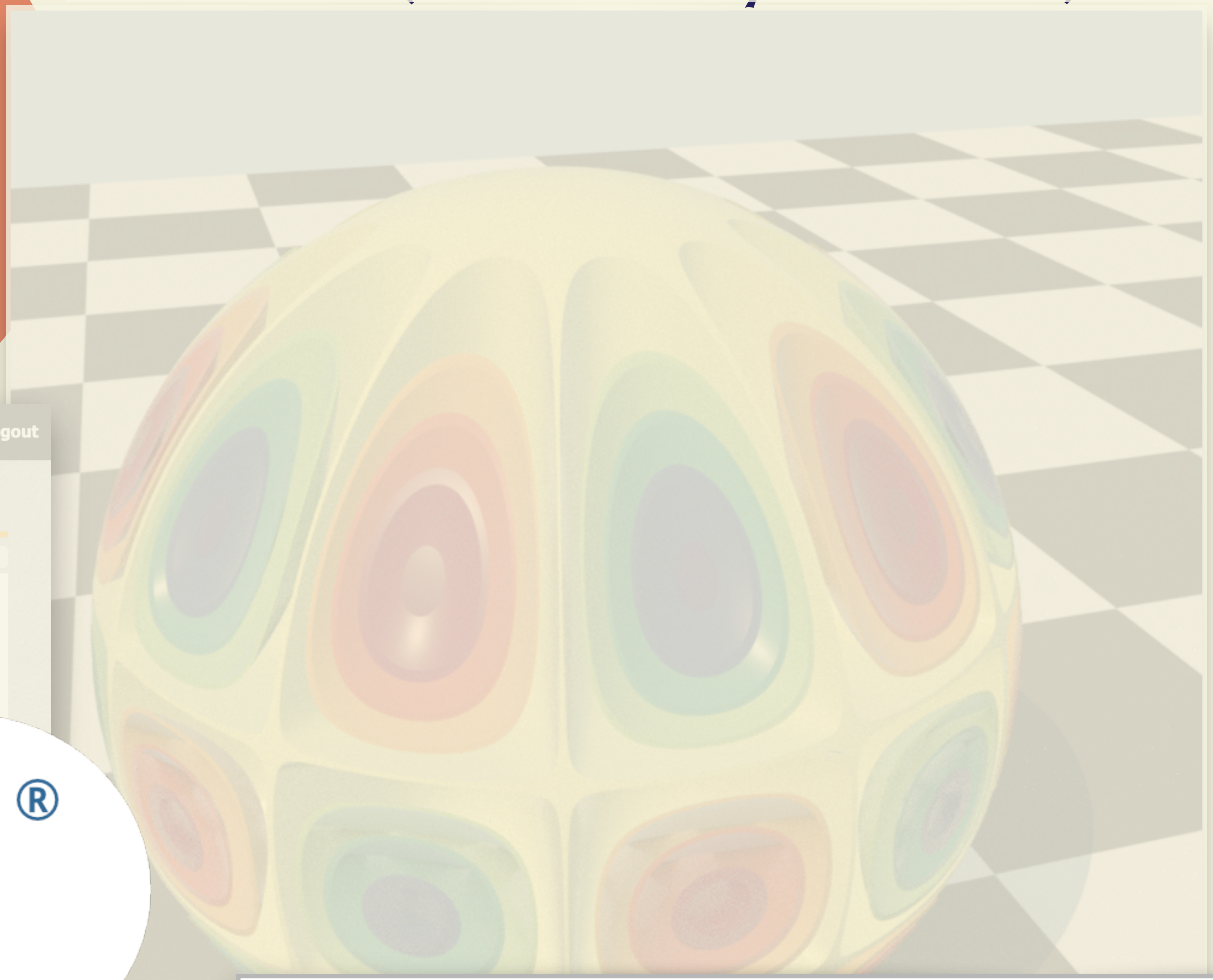
PBRT (CPU ray tracer)



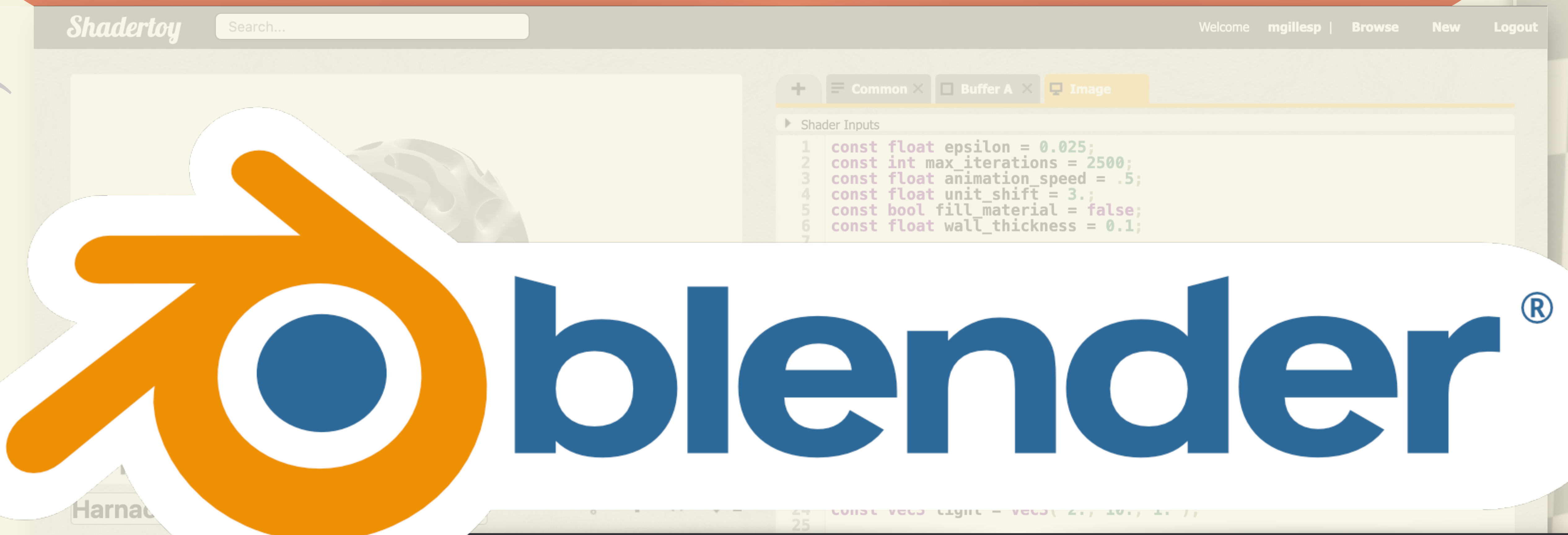
Blender (CPU ray tracer)

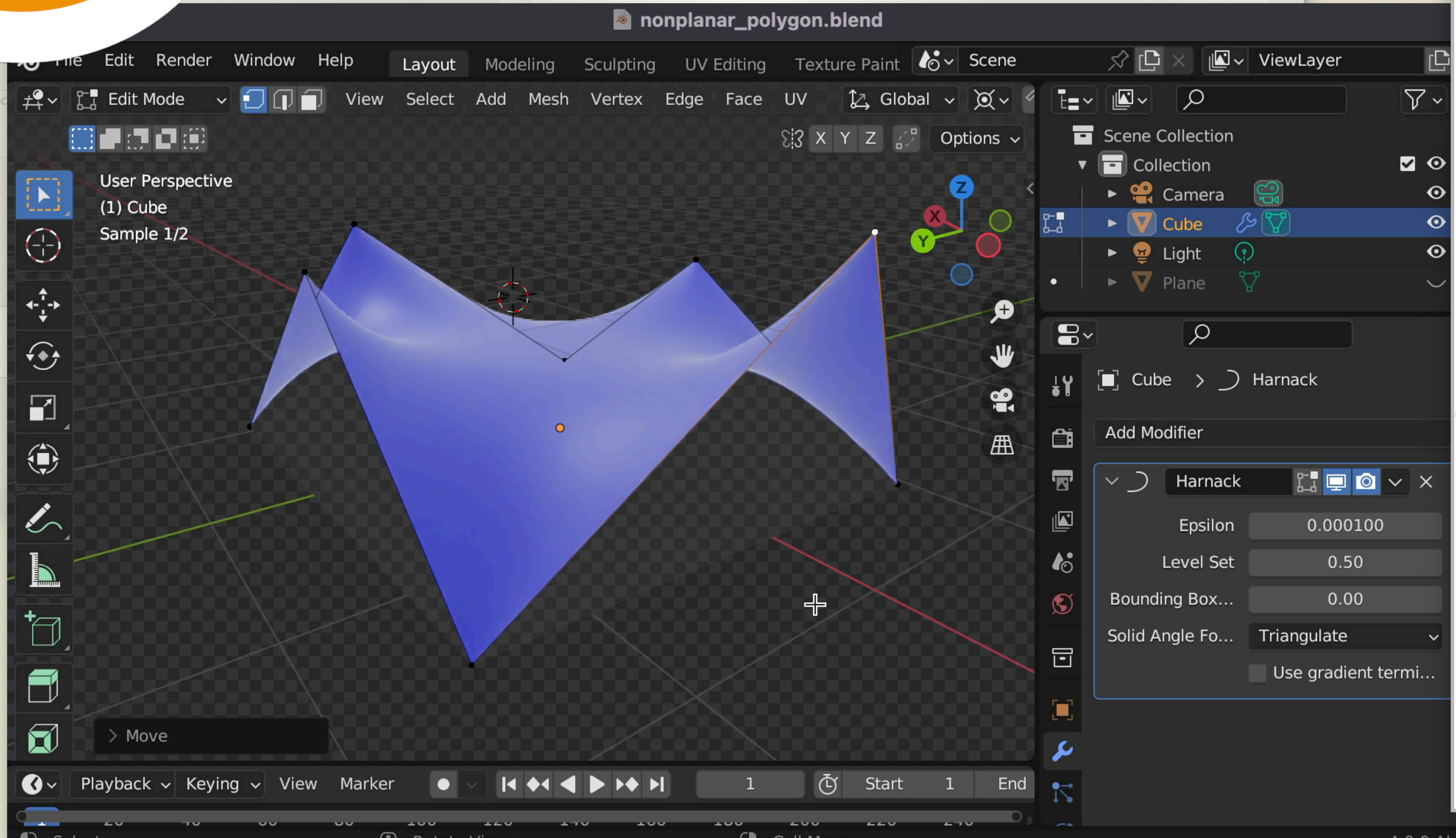
Simple to implement

PBRT (CPU ray tracer)

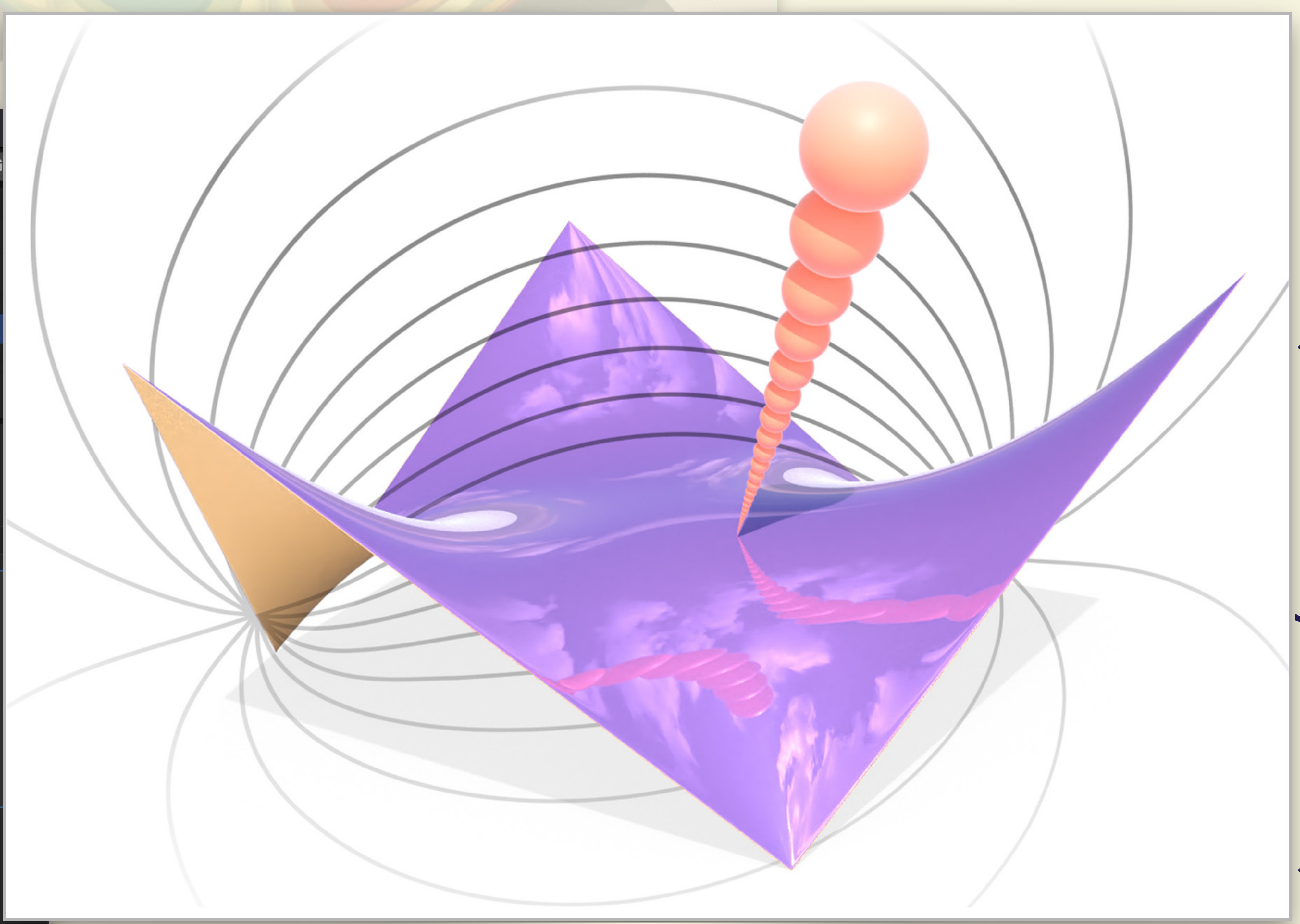


ShaderToy (WebGL shaders)





Blender (CPU ray tracer)



Generalized winding number

[Jacobson et al. 2013]

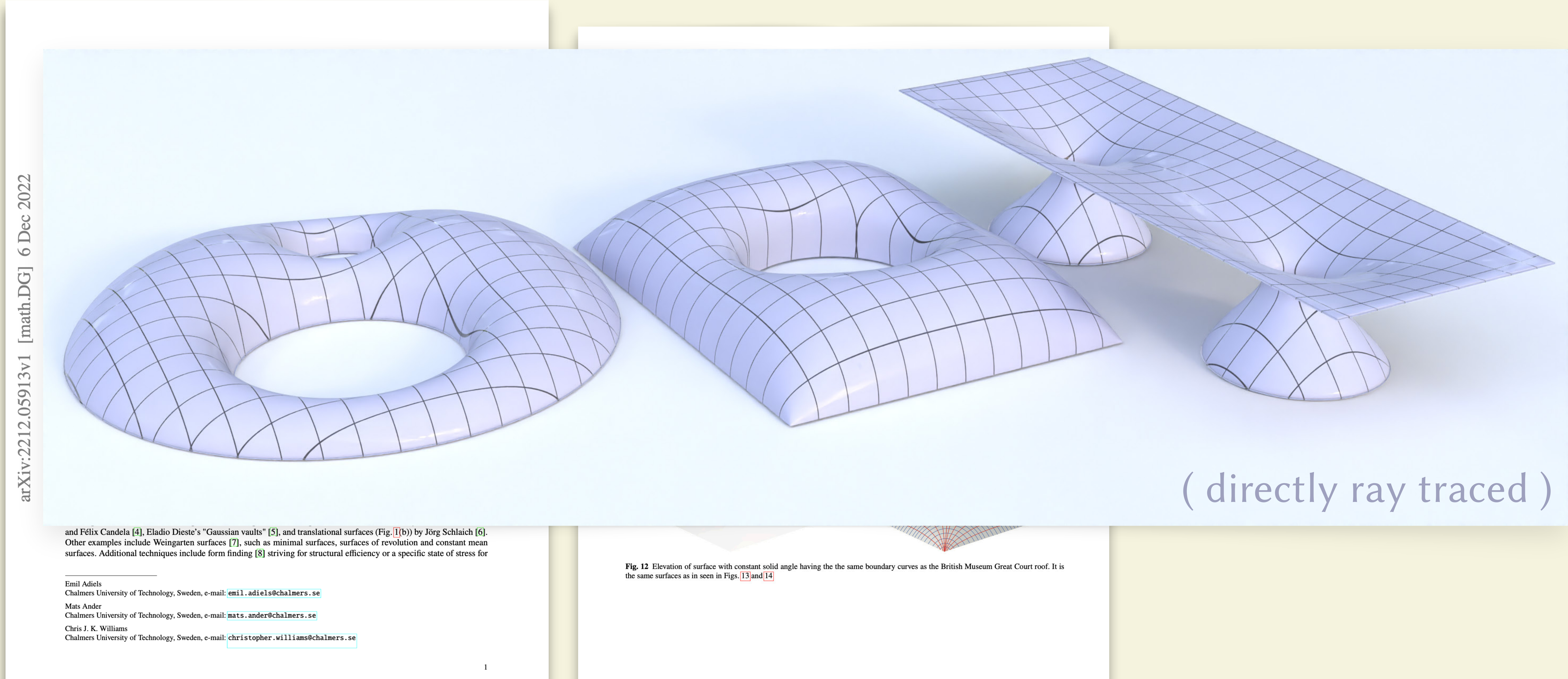
a.k.a. *signed solid angle*

input mesh

repaired mesh
(directly ray traced)

Architectural grid shells

[Adiels et al. 2022]



arXiv:2212.05913v1 [math.DG] 6 Dec 2022

and Félix Candela [4], Eladio Dieste's "Gaussian vaults" [5], and translational surfaces (Fig. 1(b)) by Jörg Schlaich [6]. Other examples include Weingarten surfaces [7], such as minimal surfaces, surfaces of revolution and constant mean surfaces. Additional techniques include form finding [8] striving for structural efficiency or a specific state of stress for

Emil Adiels
Chalmers University of Technology, Sweden, e-mail: emil.adiels@chalmers.se

Mats Ander
Chalmers University of Technology, Sweden, e-mail: mats.ander@chalmers.se

Chris J. K. Williams
Chalmers University of Technology, Sweden, e-mail: christopher.williams@chalmers.se

1

Fig. 12 Elevation of surface with constant solid angle having the the same boundary curves as the British Museum Great Court roof. It is the same surfaces as in seen in Figs. 13 and 14

The gyroid

not a harmonic function in 3D

… but is a *slice* of a harmonic function in 4D

[Diegel 2021]

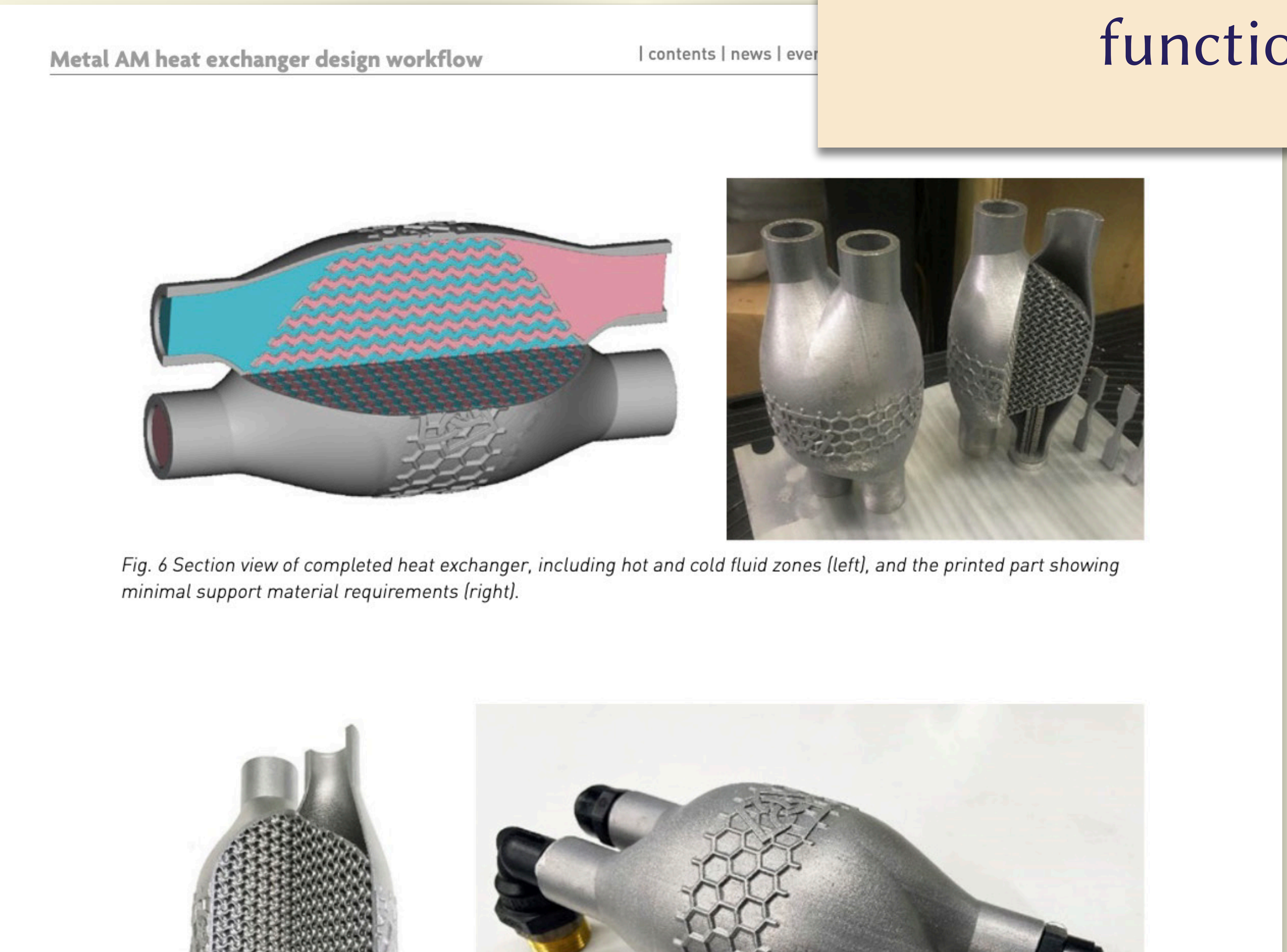
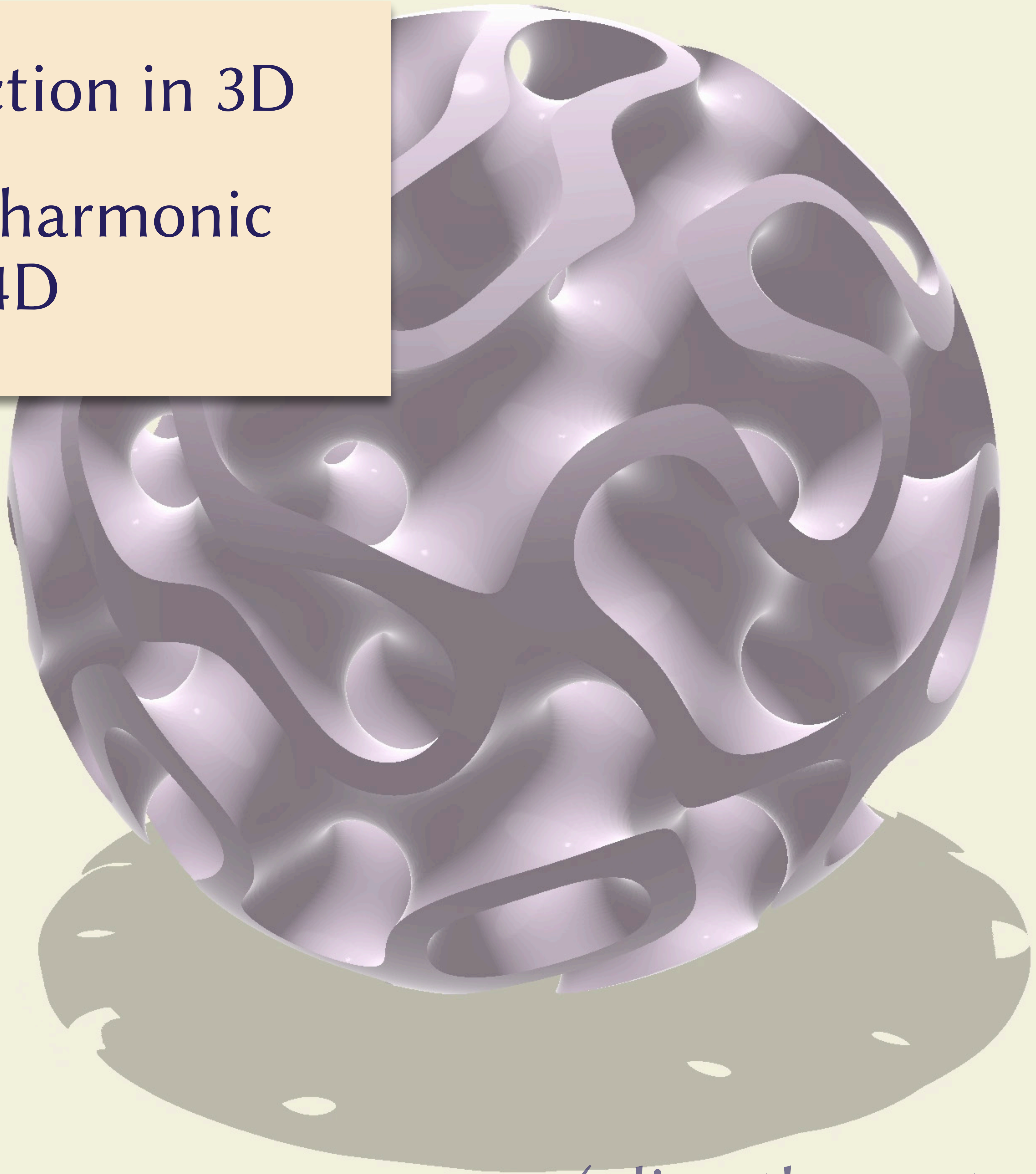

Metal AM heat exchanger design workflow | contents | news | ever

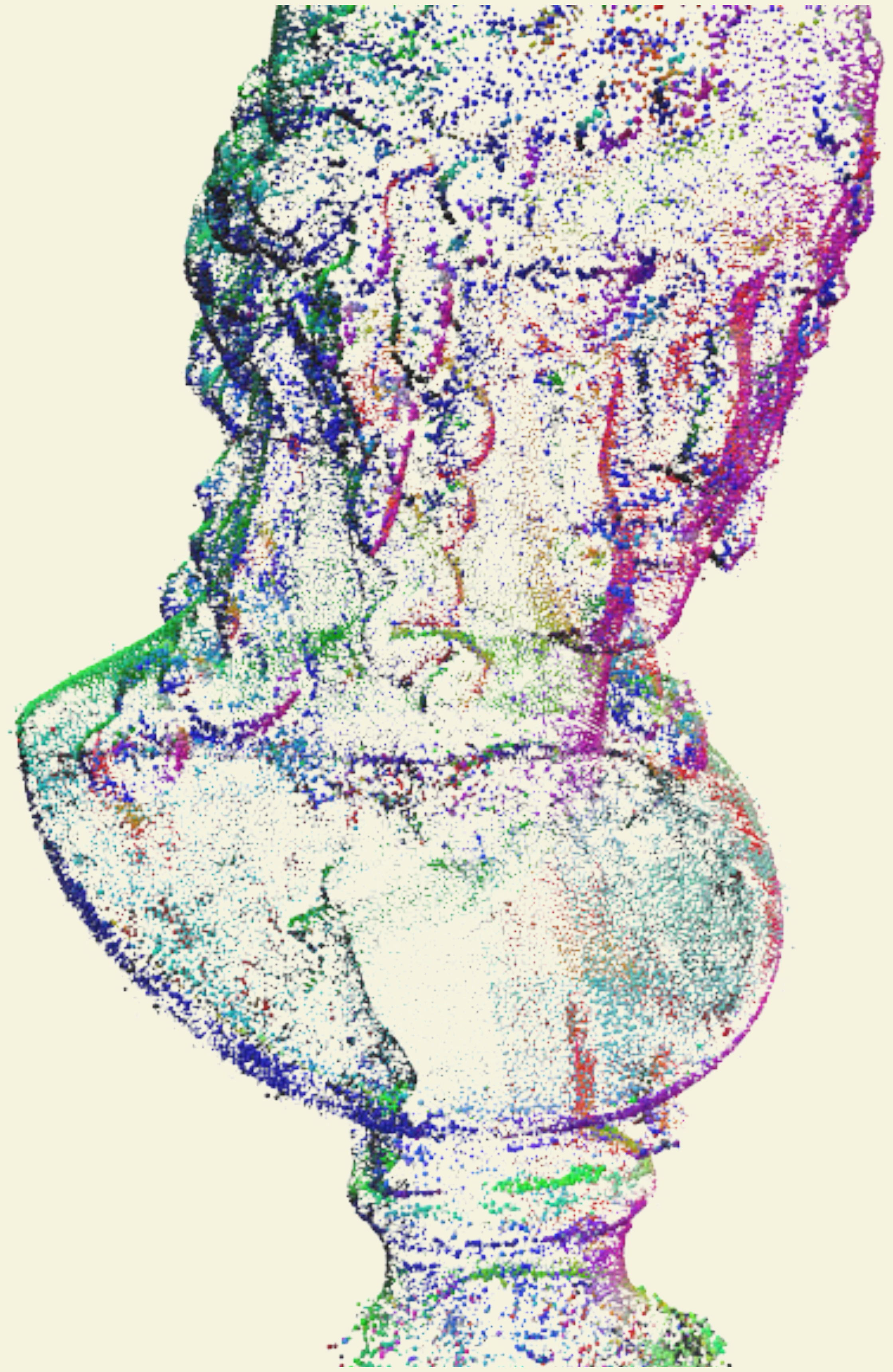
Fig. 6 Section view of completed heat exchanger, including hot and cold fluid zones (left), and the printed part showing minimal support material requirements (right).

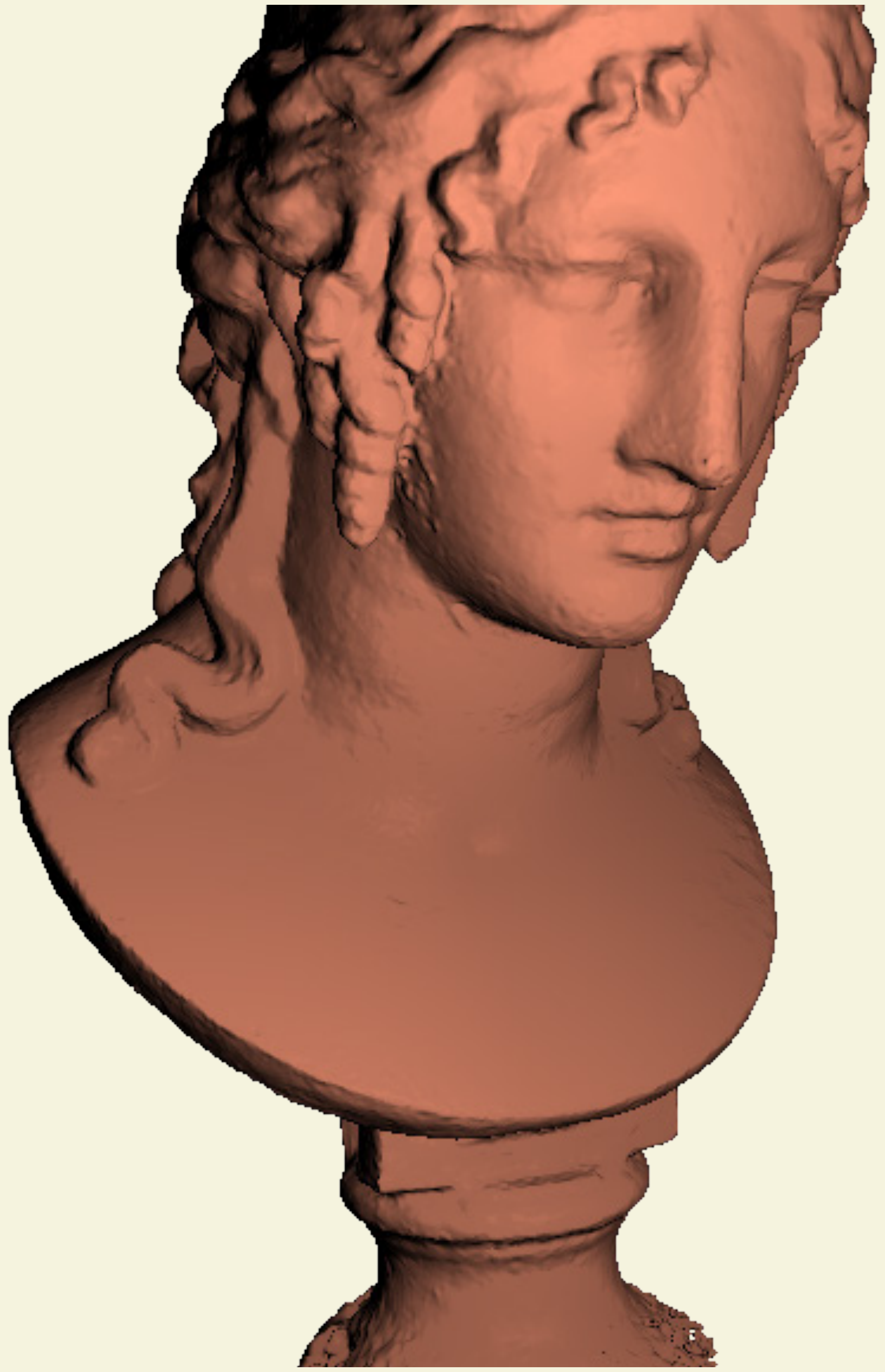


(directly ray traced)

Differentiable rendering

[Chen *et al.* 2024] incorporated Harnack tracing into their differentiable 3D reconstruction algorithm





Future Directions

Robustness by default

Analogy: numerical linear algebra

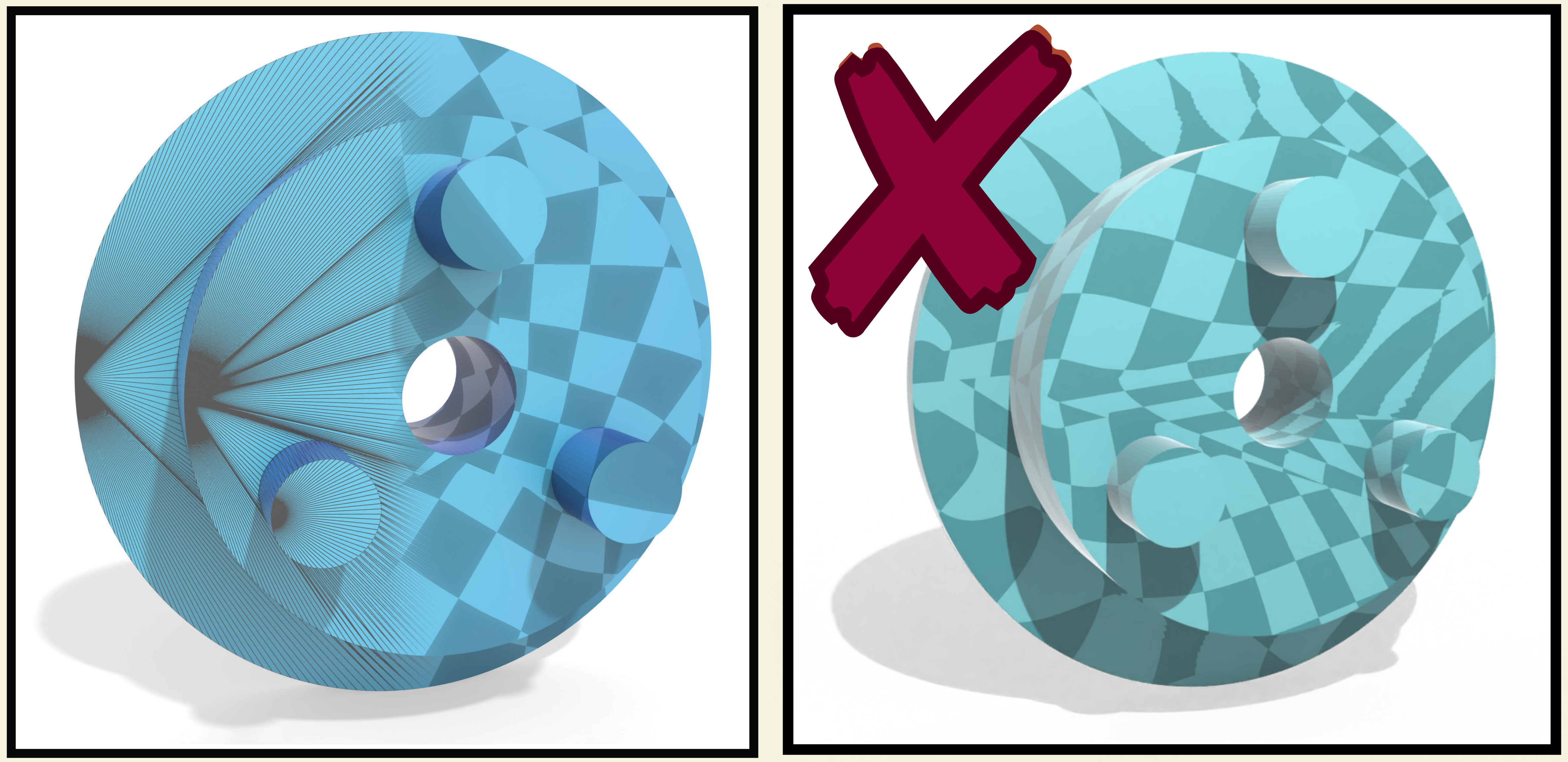
`A \ b`

solve $Ax = b$

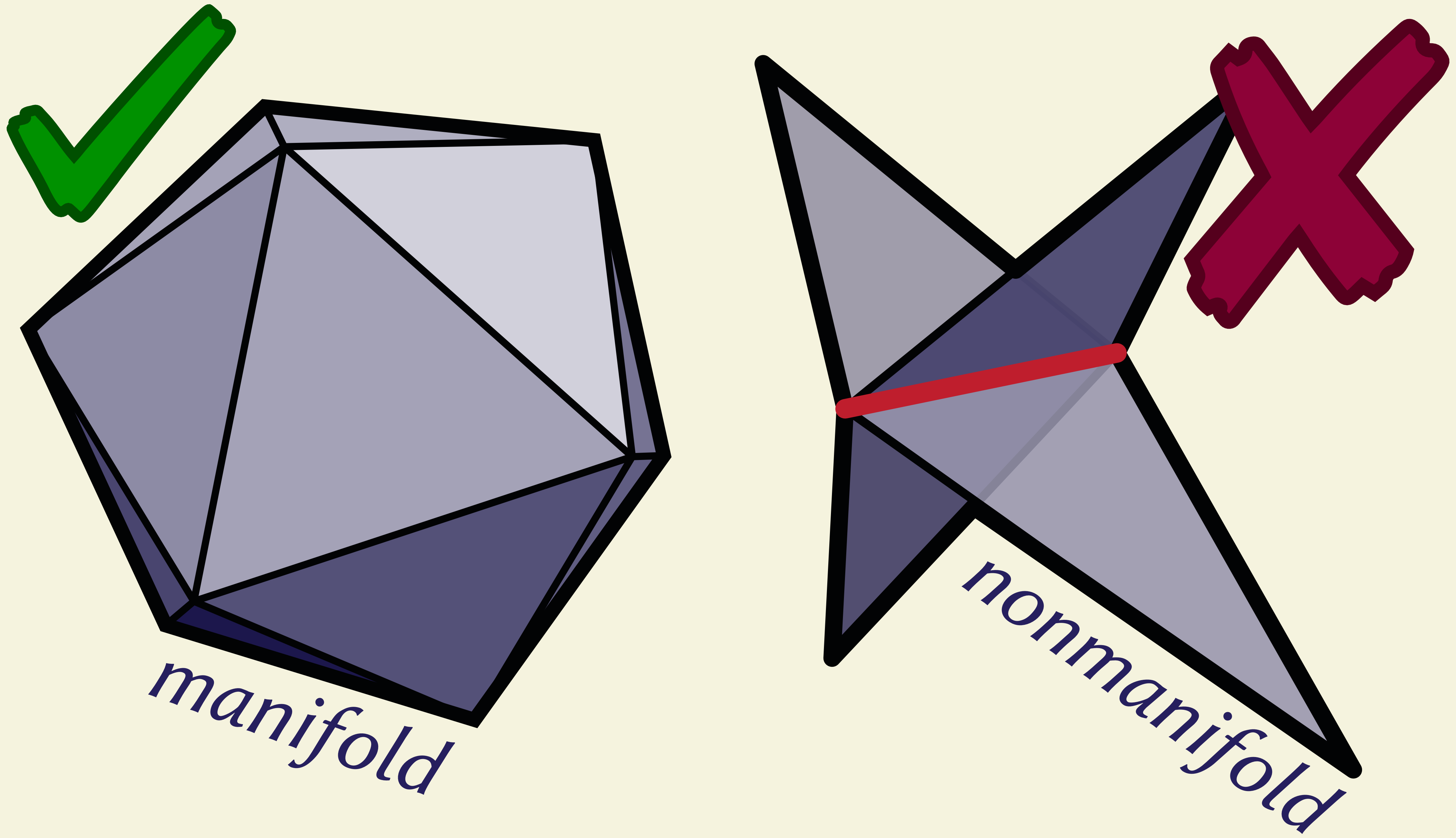
- Cholesky factorization
- LU decomposition
- QR decomposition
- triangular solver
- Hessenberg solver
- LDLT decomposition

Topological robustness

Today's talk:
geometric robustness

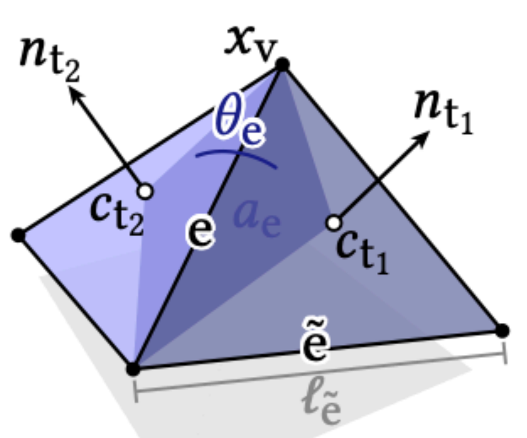


The missing ingredient:
topological robustness



Everyone assumes the input is manifold

Notation. In the smooth case, we consider a surface M, whose geometry is given by an embedding $\mathbf{x}: M \to \mathbb{R}^3$. We use $\mathbf{n}$ to denote the corresponding unit normals, and dA for the area element of the embedded surface. In the discrete case, we have a triangle mesh $M = (V, E, T)$ with manifold connectivity. Its geometry is given by vertex coordinates $x_v \in \mathbb{R}^3$ for each $v \in V$.



... the area, unit ... $e \in E$ we use ... $E_{int} \subset E$, with

Assume a 2-manifold triangle mesh $M =$ (... intrinsic geometry of M by $\mathcal{S} = (\mathcal{T}, \mathbf{L})$ w...

6.3. Limitations

KerGen assumes that each edge in the input mesh is shared by ex-

We represent surfaces as ... piecewise linear curves w... essential steps of our algo... distance field and finding ... surface. Both problems h... geometry representation, ... rithms with publicly avail... work: For our implement... *Heat Method)* [CWW13] ... this choice is not unique, ... as well (refer to surveys ... ment for any method is b... given oriented topological 2-ma...

but in the real world, many models
are not manifold

, and resp., supersolutions). ... M. A function $u: V \to \mathbb{R}$ is ... tion) of $\frac{\partial u}{\partial t} + H(x, q, u) = 0$ ... d every point $x \in V$ such ... , minimum) at x, we have

... the evolving Stokes ... expressed in tangent-normal splitting with a simple, coordinate-free differential-geometric formulation. This ... irectly leads to a straightforward discrete model and a numerical scheme to ... ng-standing problem in its full geometric generality. ... the theoretical framework is not limited, the current minimal implementation is ... closed manifolds and employs first-order discretization. Future work could ... boundary conditions and higher-order discretizations. For applications, we expect the minimal system presented here to serve as a foundation for integrating more complex models for specific biophysical problems. These could include additional global volumetric/areal constraints, heterogeneity in material properties, in-plane anisotropy, and surface activity, as exemplified in Zhu *et al.* (2022, 2024).

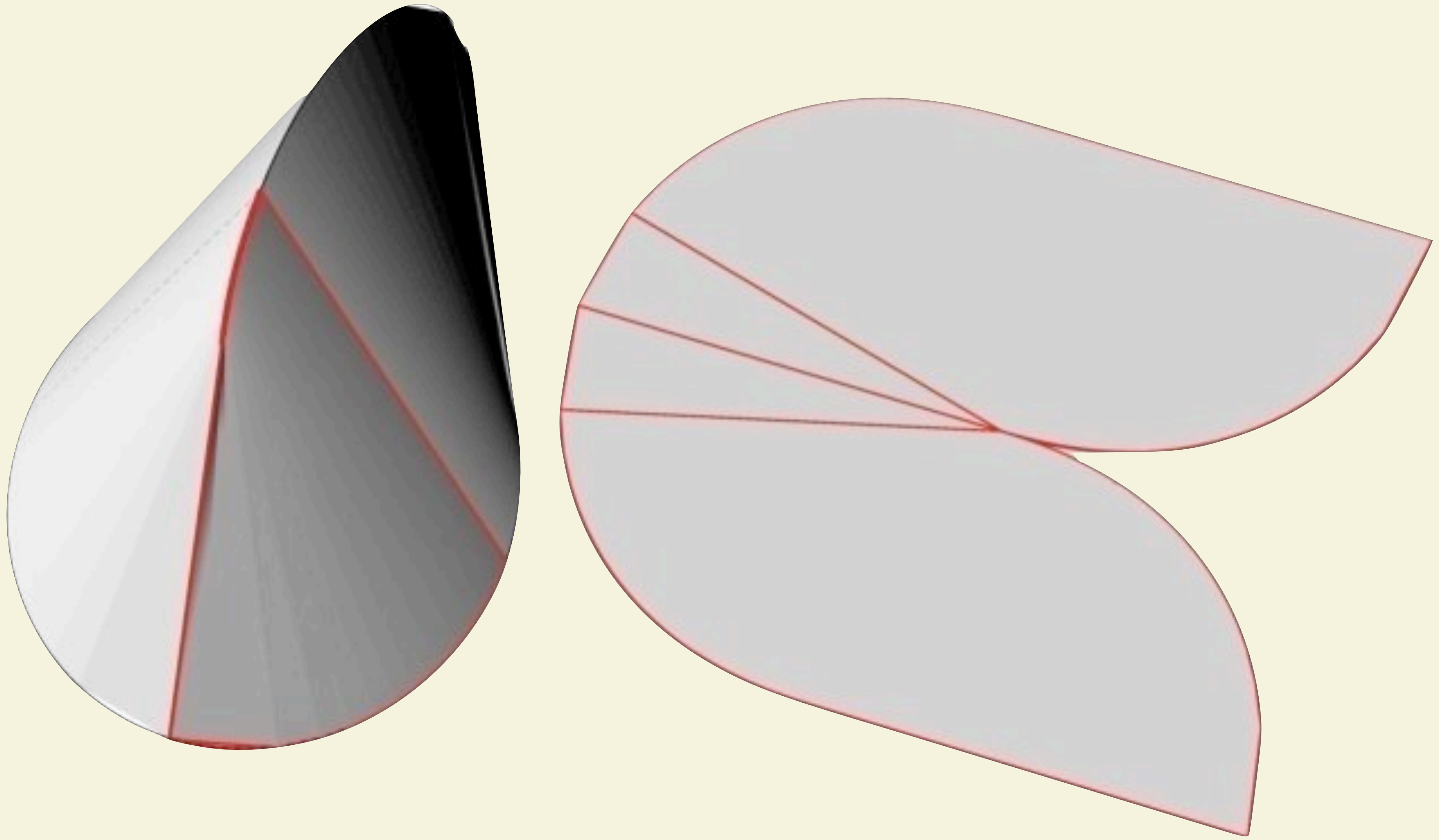
3.1 Discrete manifold and choice of variable space

In this section, our goal is to establish notations for representing a 2D manifold using a discrete mesh and navigating its local topology. Our focus here lies in developing a common language for tensor calculus on a discrete manifold, which forms the basis for discretizing the system of equations.

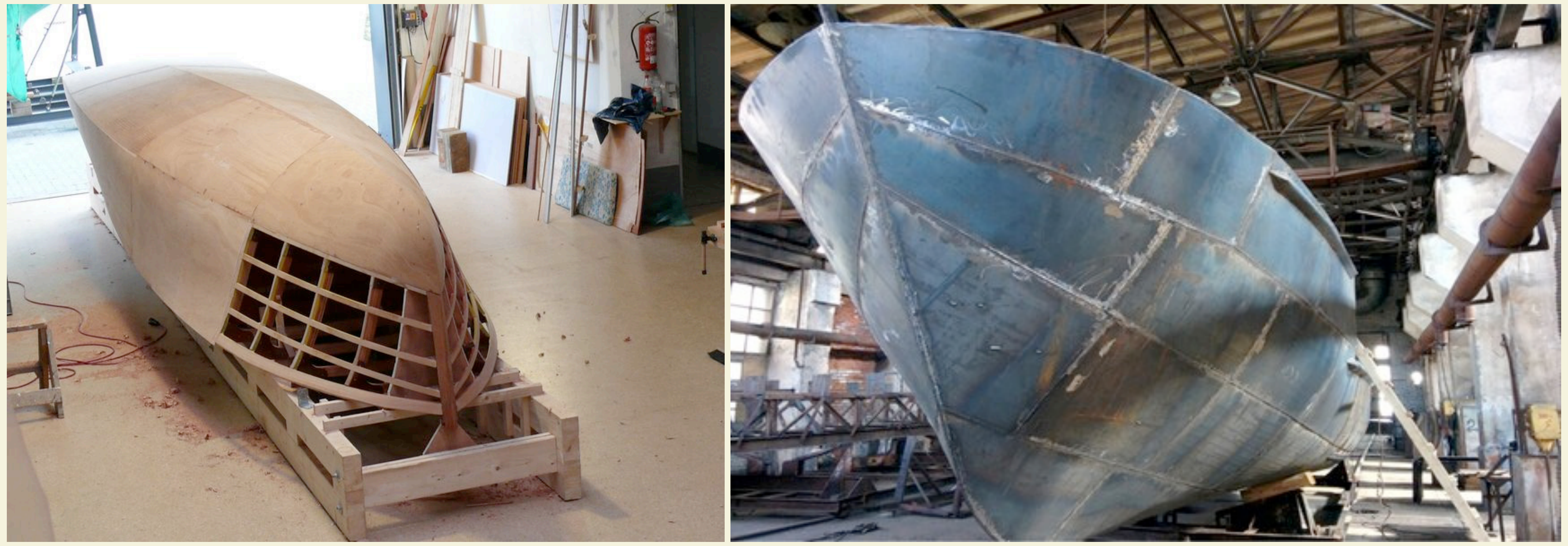
... can be found in Table 1. In general, the runti... on the same order as those of Stein et al. [... that our solver consistently converged for m... meshes with reasonable triangles.

Fabrication

developable surfaces



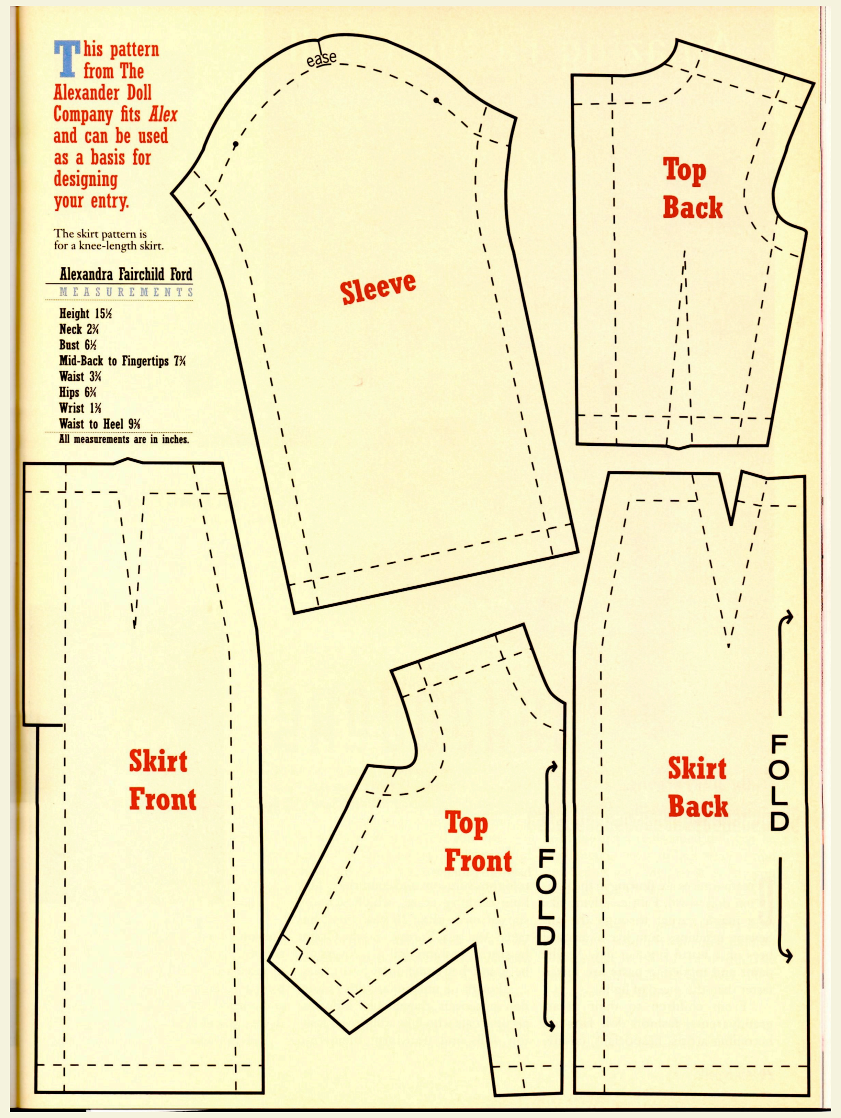
laid out in the plane without stretching



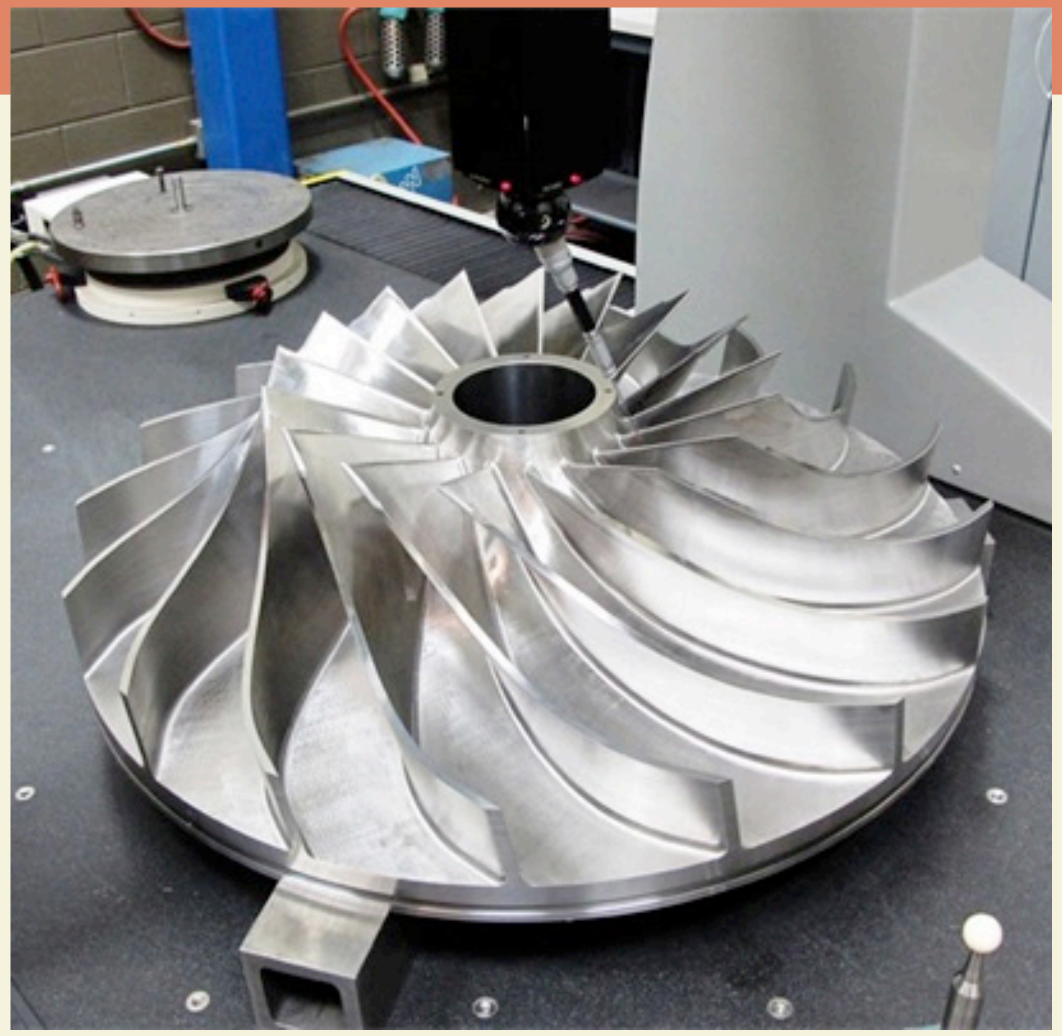
easy to make from plywood or sheet metal

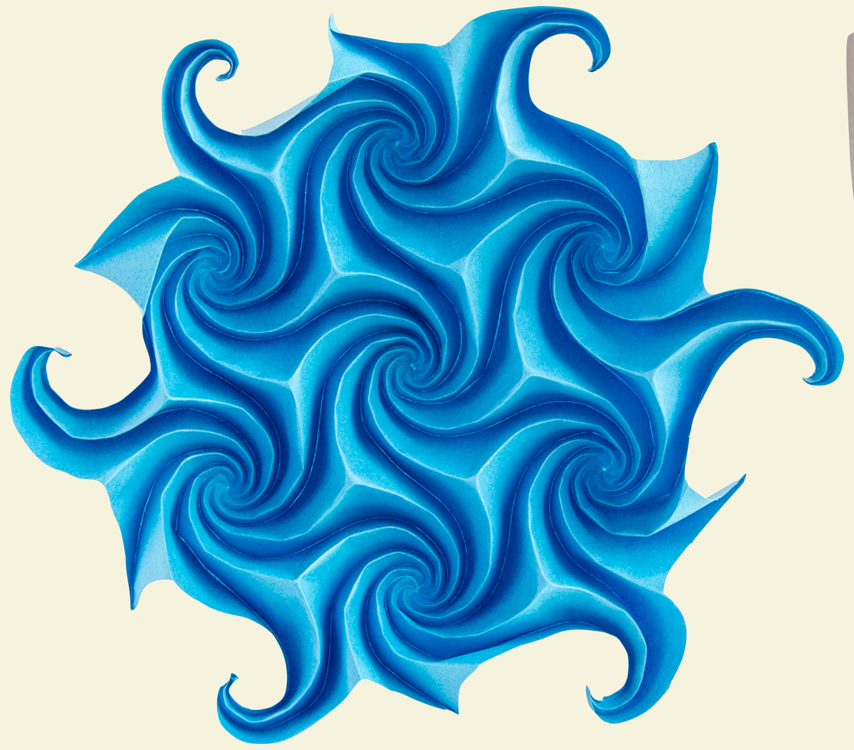
Geometric design tools

developable surfaces



textile design

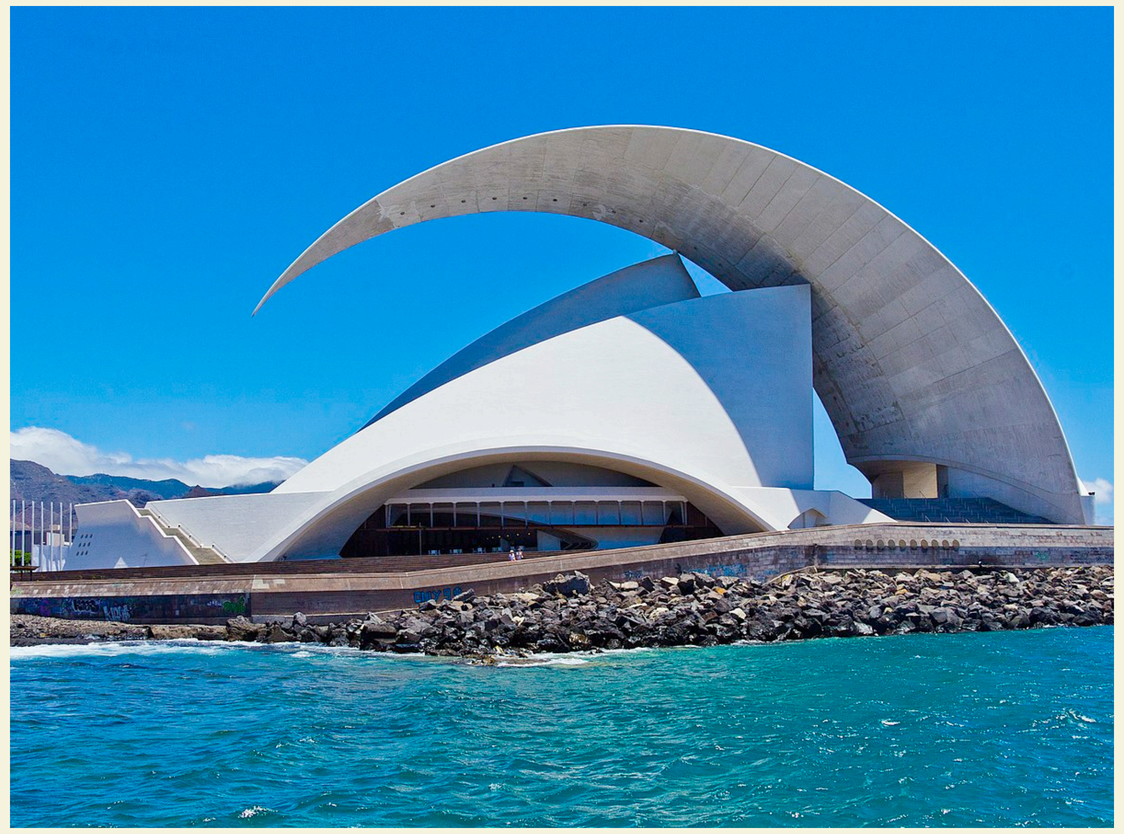

flank milling



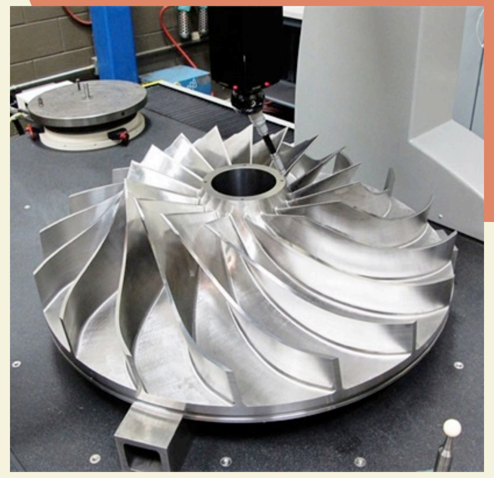
curved-crease origami

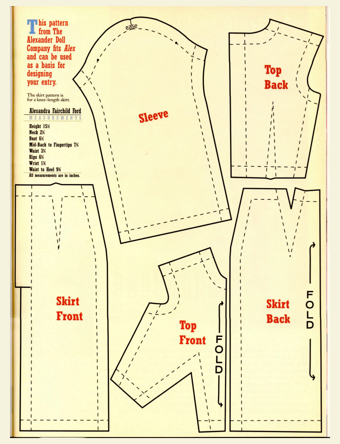


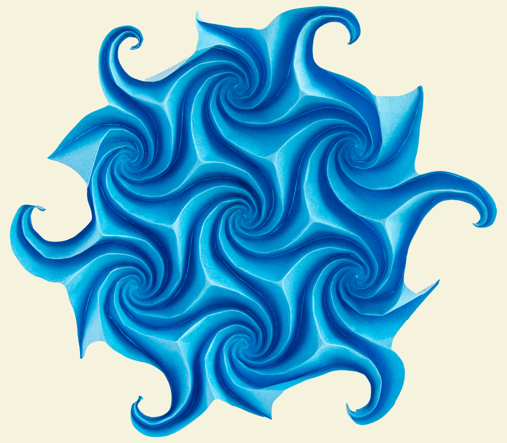
easy to make from plywood or sheet metal

Fabrication



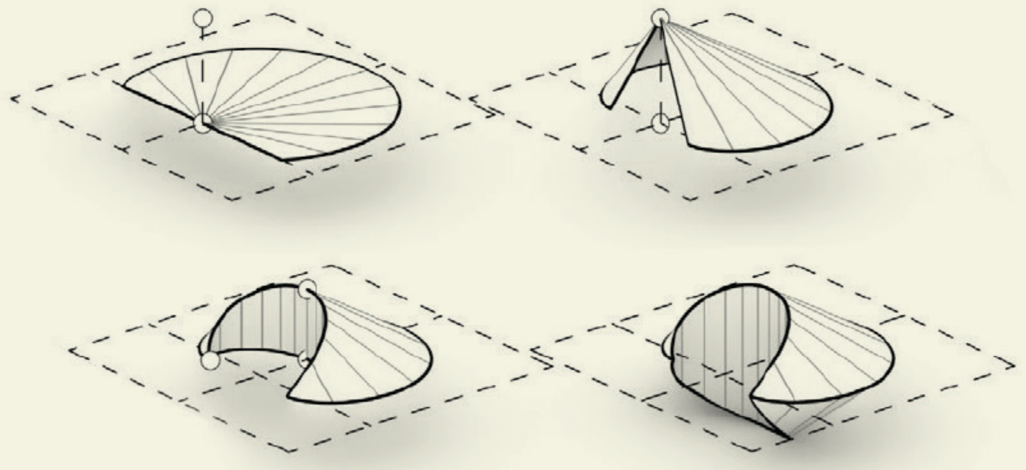




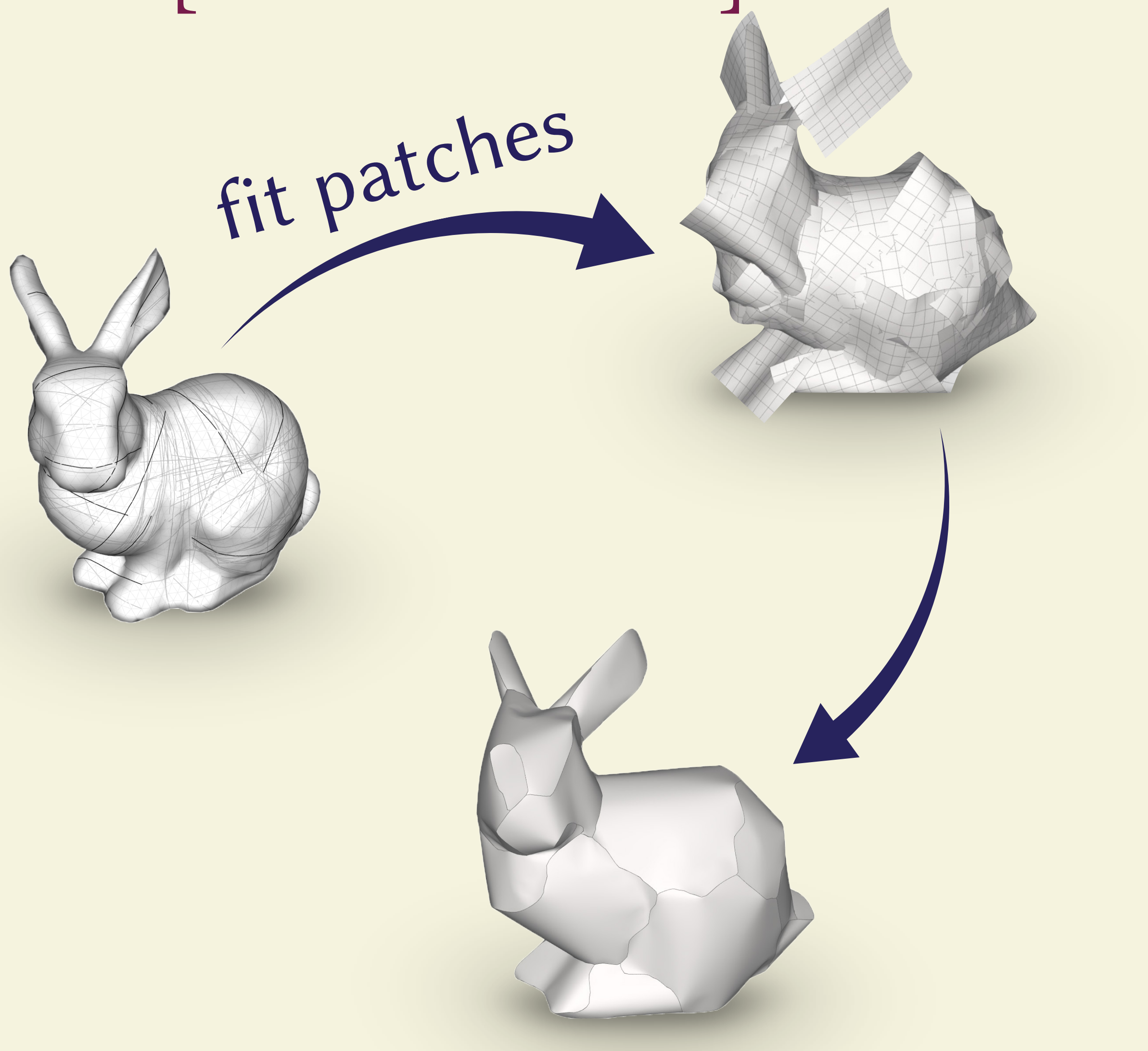
developable surfaces

hard to design

1. Build from closed-form pieces
[Munidlova *et al.* 2021]



2. Complex non-convex optimization
[Ion *et al.* 2020]

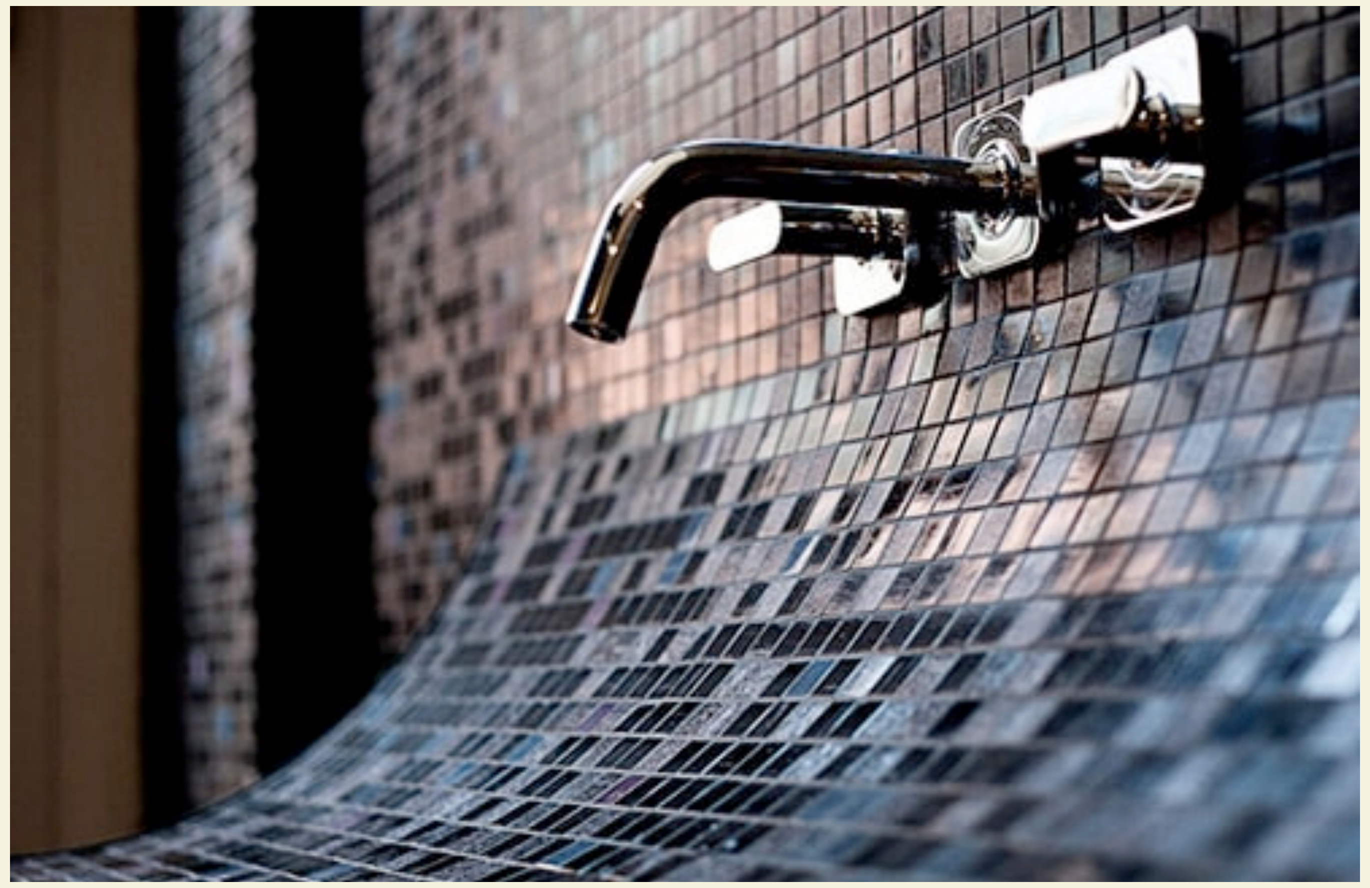


Work intrinsically:

developable surface design becomes a parameterization problem

Fabrication

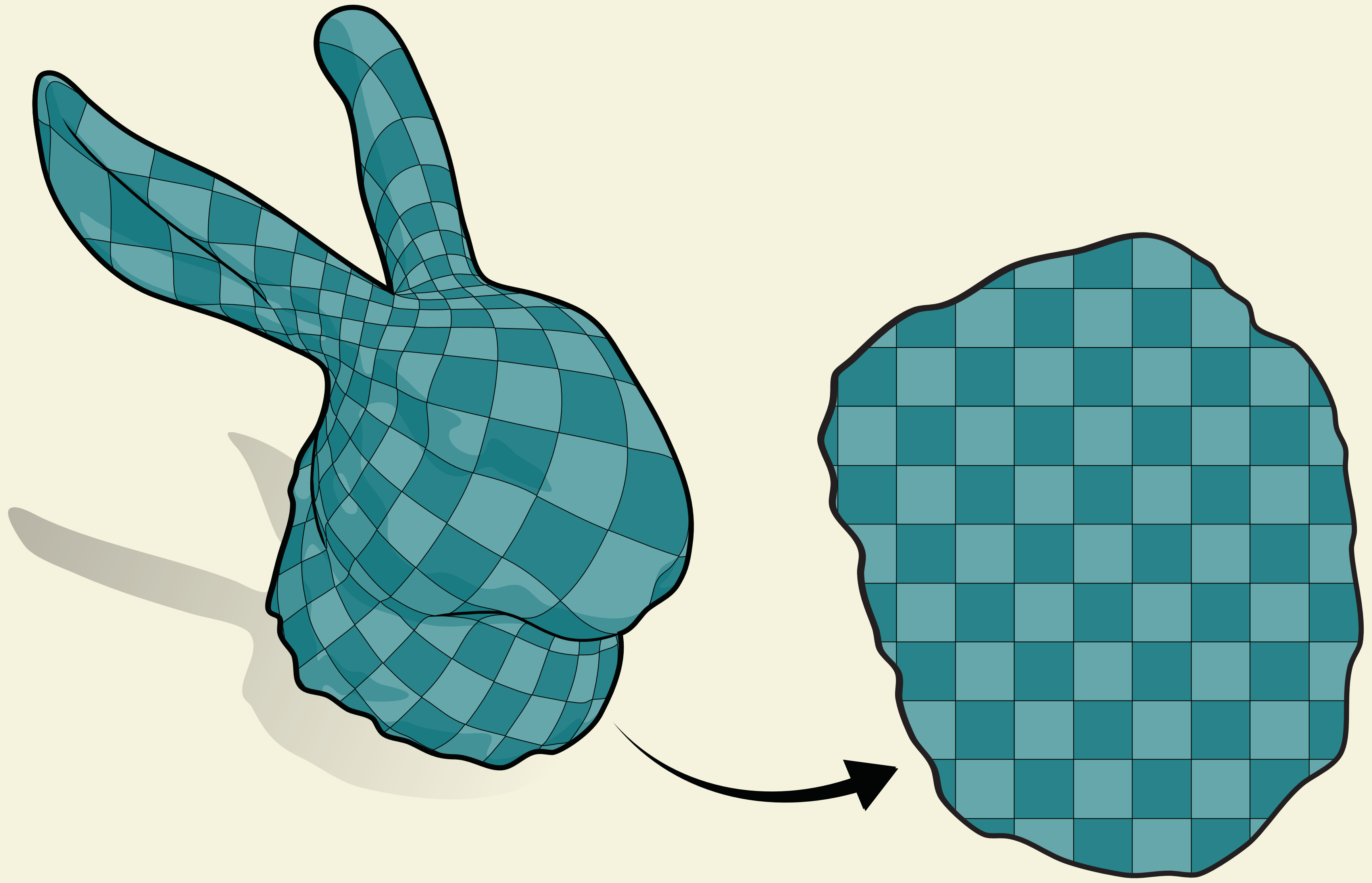
polygonal tilings



Fabrication

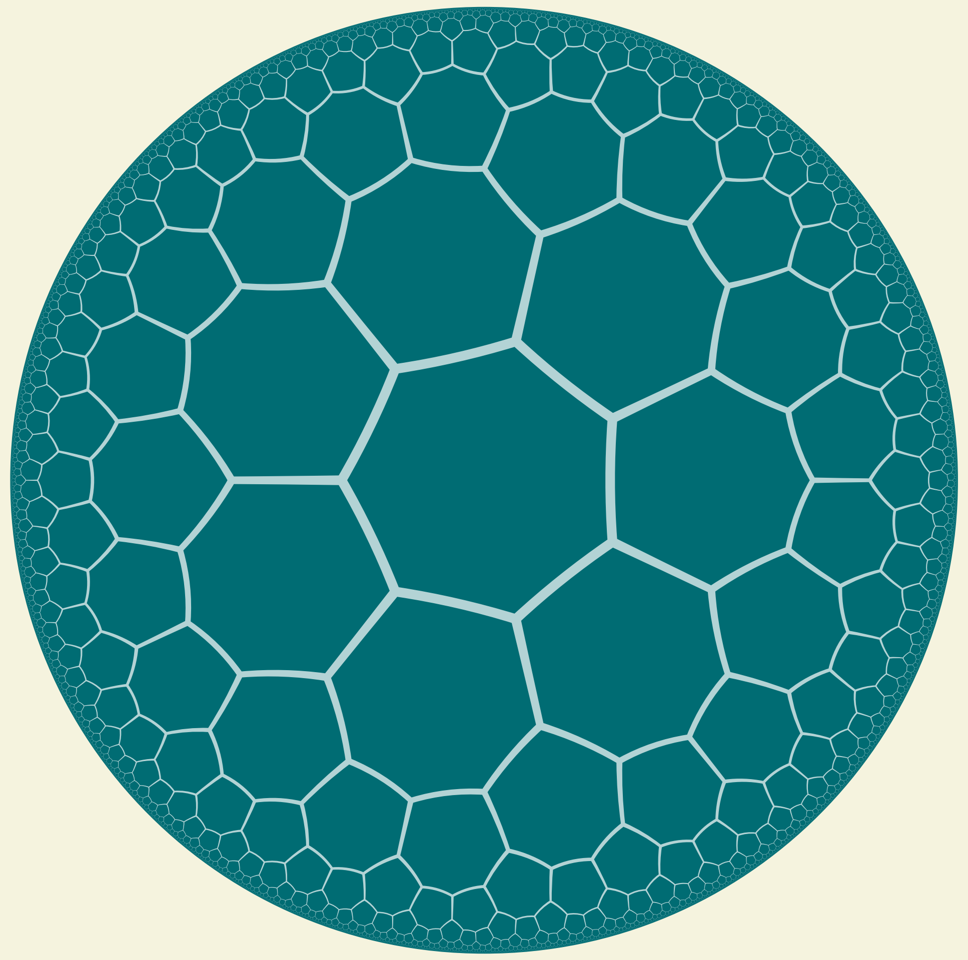
polygonal tilings

standard approach:
parameterization to plane

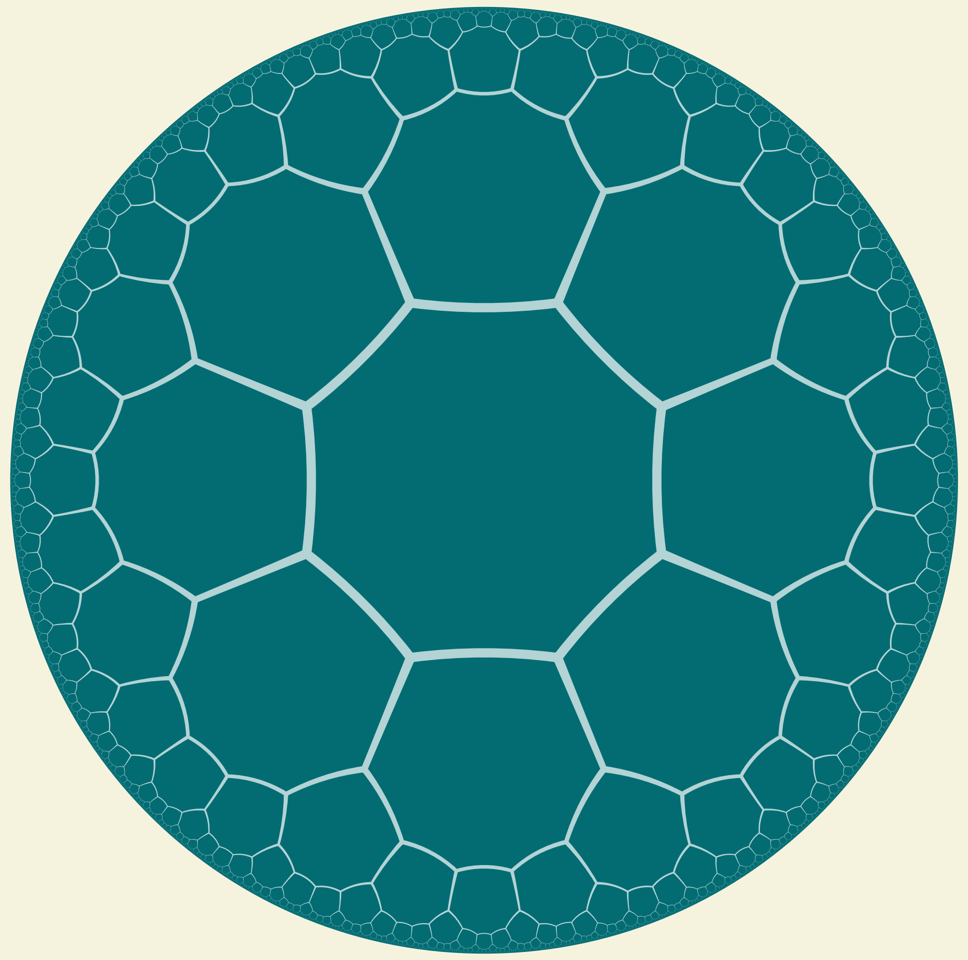


limited to regular
quadrilaterals & triangles

new approach:
parameterization to hyperbolic plane



tiling by
regular heptagons

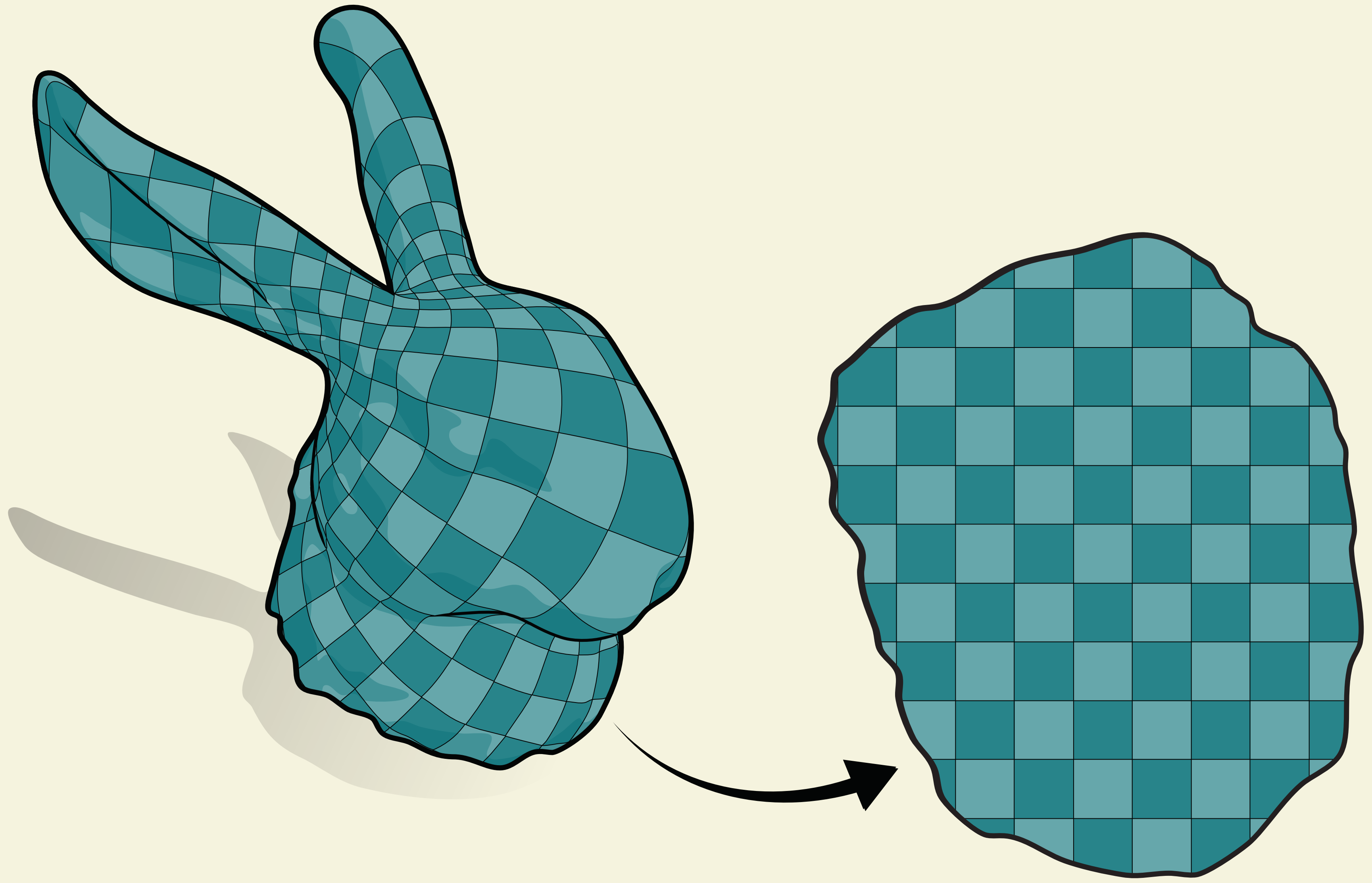


tiling by
regular octagons

Fabrication

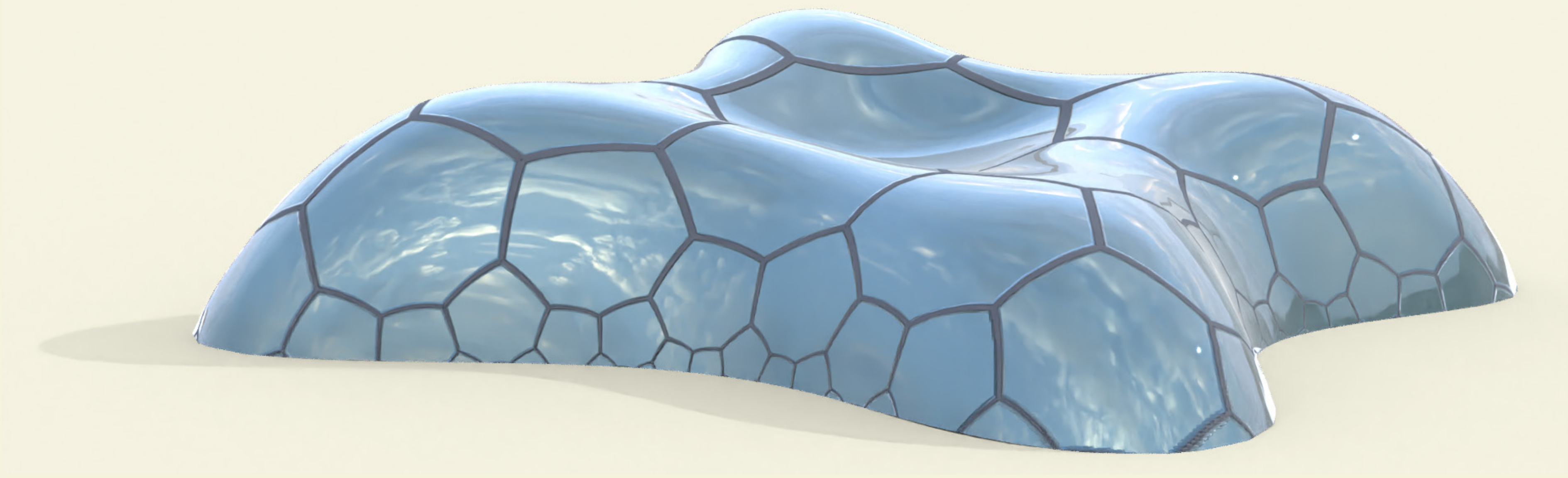
polygonal tilings

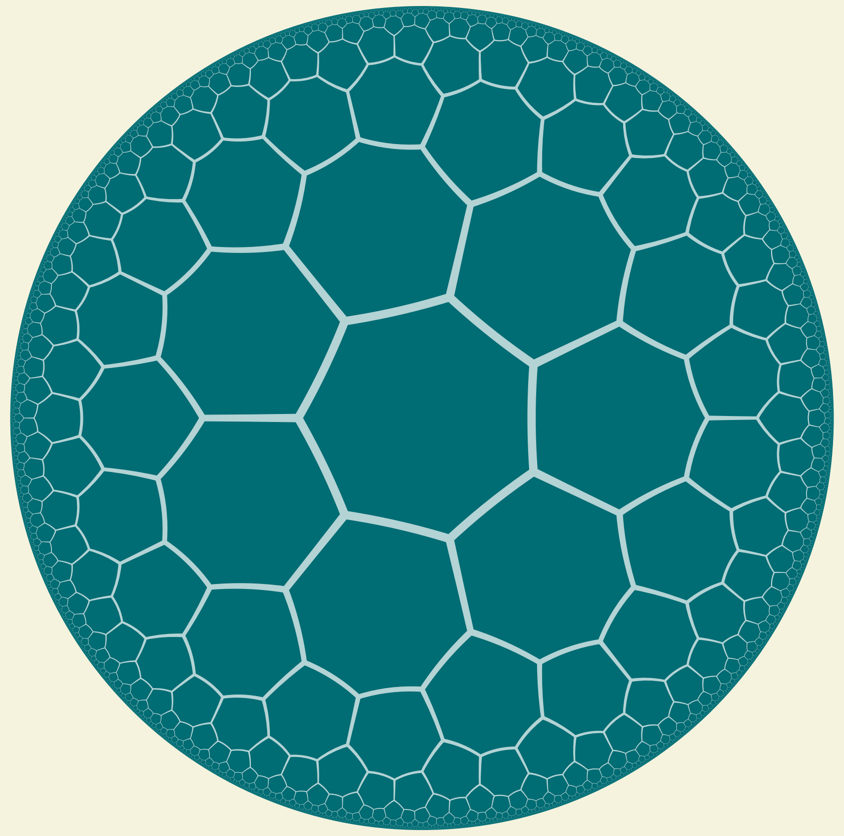
standard approach:
parameterization to plane



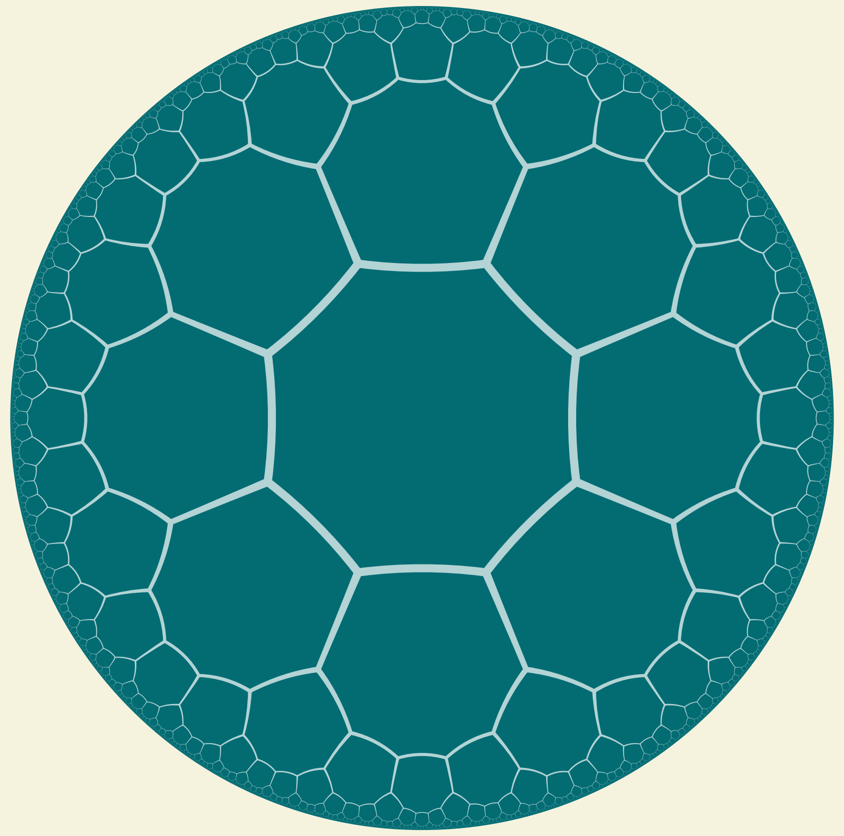
limited to regular
quadrilaterals & triangles

new approach:
parameterization to hyperbolic plane





tiling by
regular heptagons

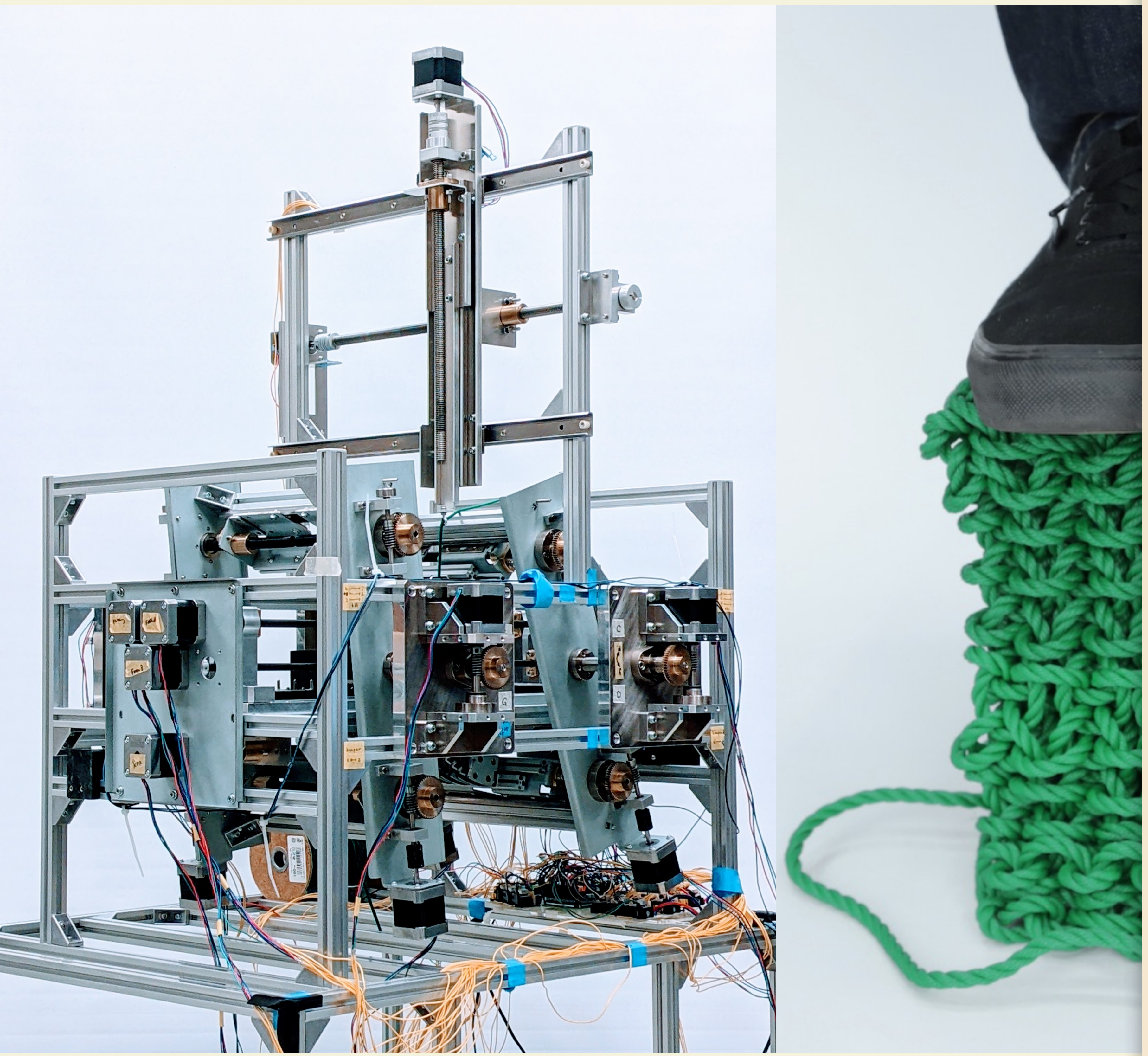


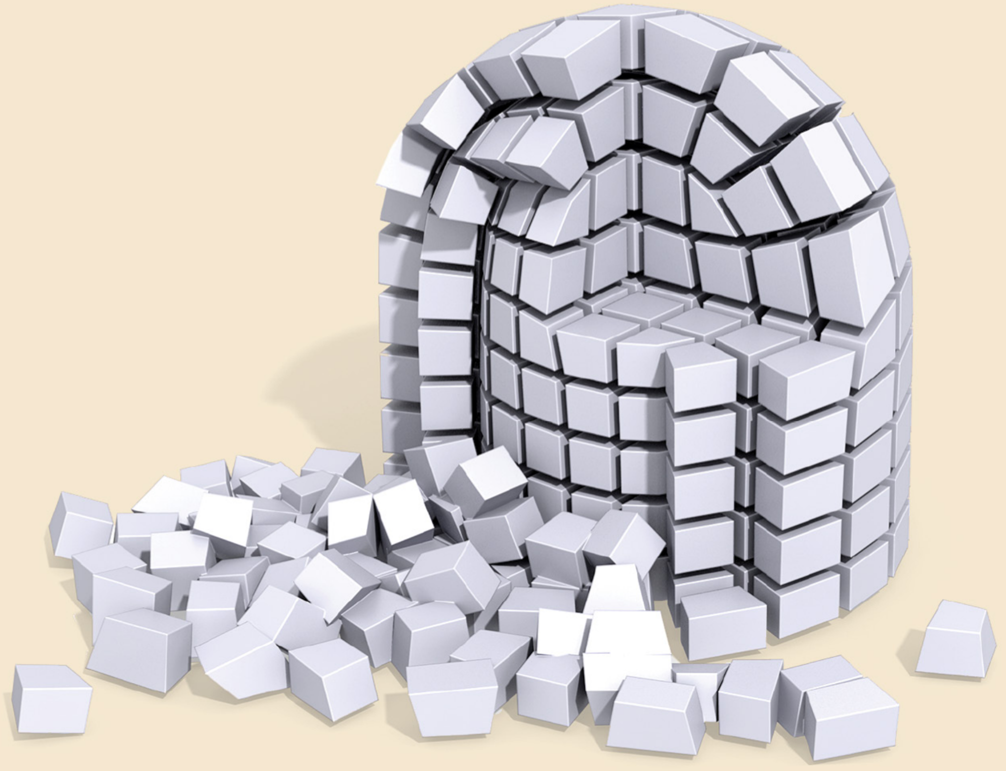
tiling by
regular octagons

Fabrication

solid knitting

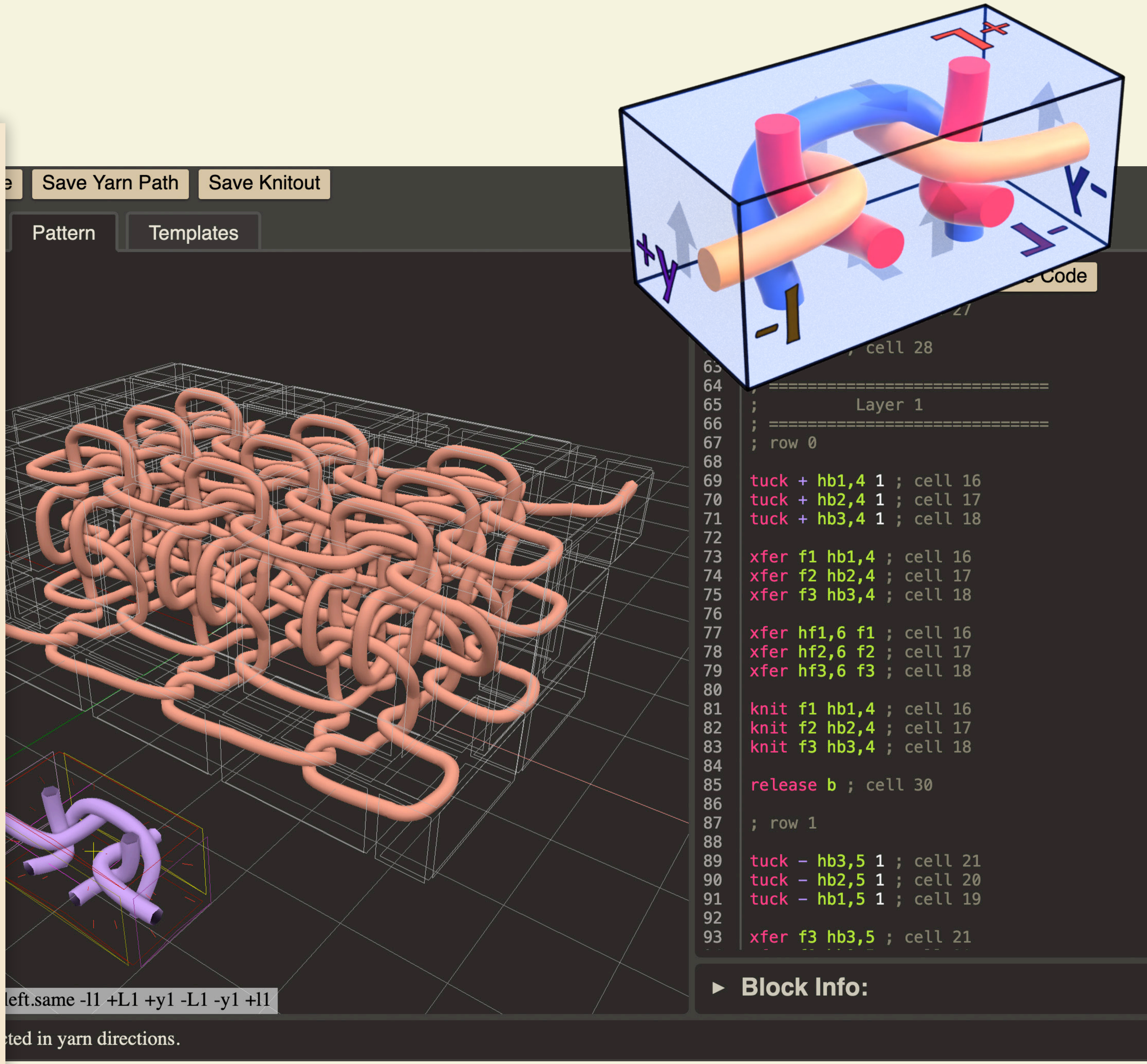
[Hirose, **Gillespie**, Bonilla Fominaya, & McCann. 2024. *Solid Knitting.* ACM Trans. Graph.]





Hexahedral Meshing

[Pietroni et al. 2023]
[Desobry et al. 2021]
[Fang et al. 2021]
[Palmer et al. 2020]

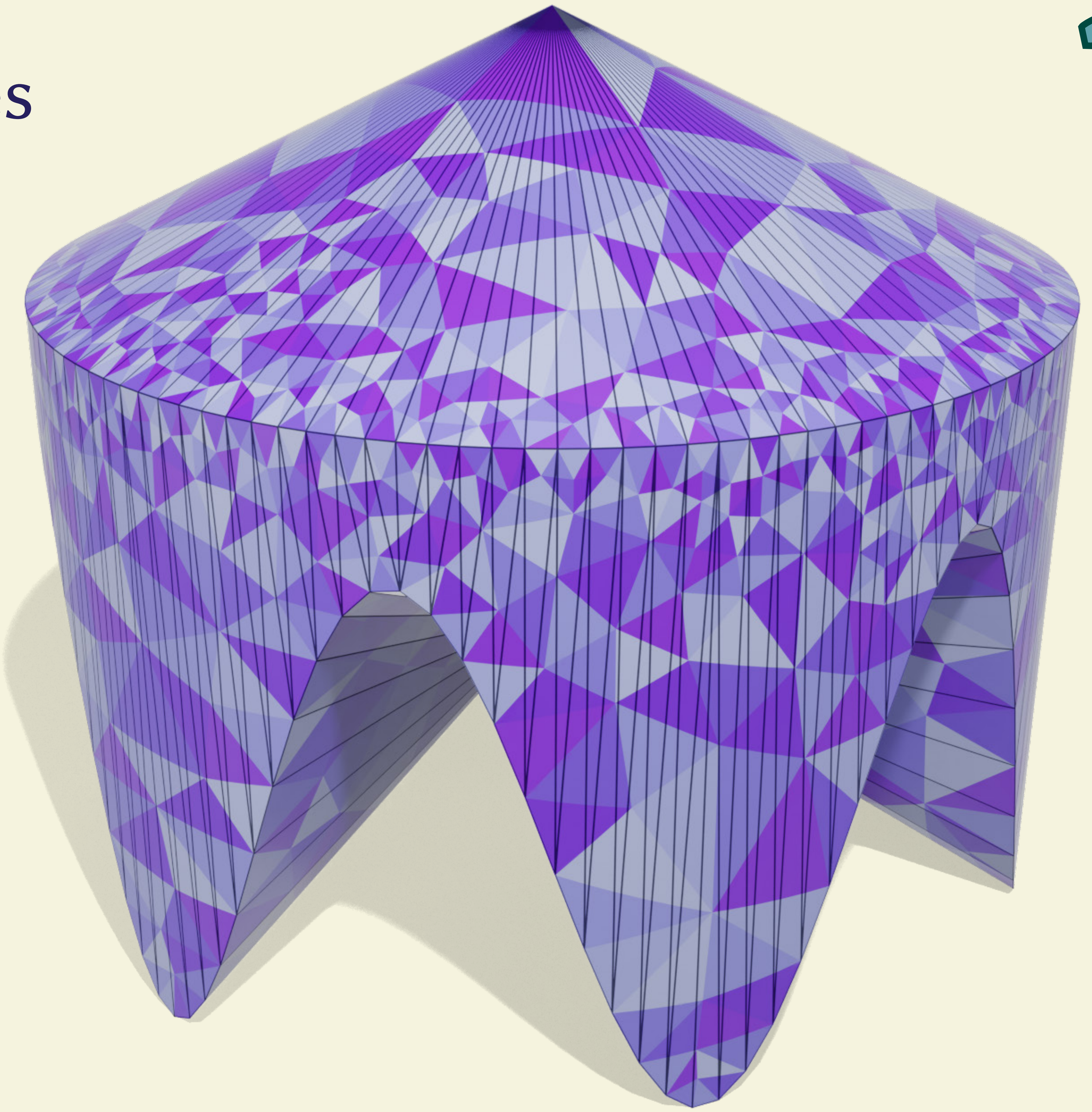


Stronger guarantees for intrinsic retriangulation

✓ quality guarantees

? efficiency guarantees

meet quality bounds using as few triangles as possible



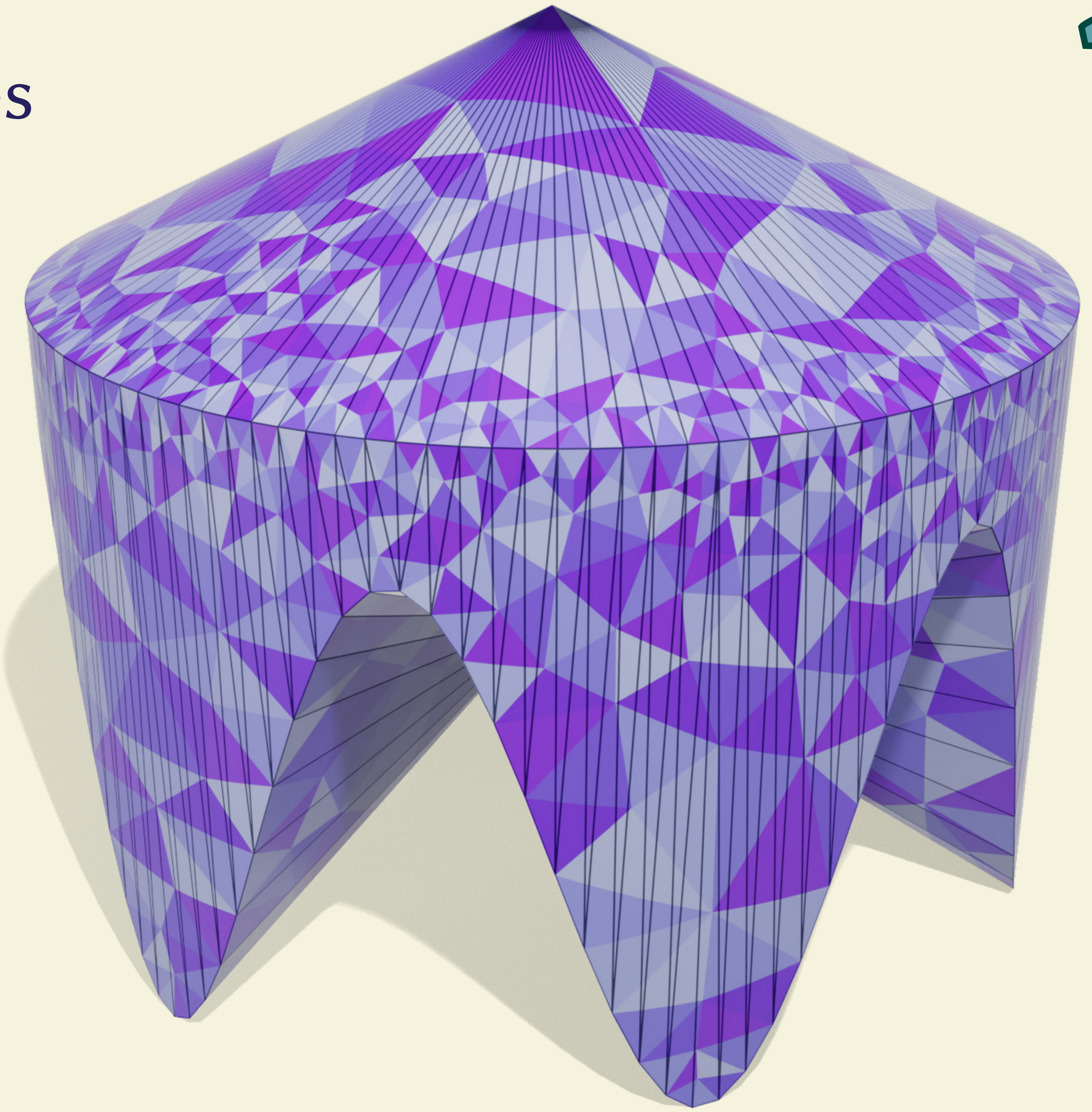
intrinsic Delaunay refinement

Stronger guarantees for intrinsic retriangulation

✓ quality guarantees

? efficiency guarantees

meet quality bounds using as few triangles as possible



intrinsic Delaunay refinement

Other Open Questions

asymptotically-optimal Delaunay triangulation
[Guibas & Stolfi 1985]

spatial distribution of inserted vertices
[Shewchuck 1997]

exact predicates
[Devillers & Pion 2003]

Thank you to all of my great collaborators

Keenan
Crane

James
McCann

Yuichi
Hirose

Angelica
Bonilla Fominaya

Mario
Botsch

Denise
Yang

Nicole
Feng

Nicholas
Sharp

Hsueh-Ti
Derek Liu

Benjamin
Chislett

Alec
Jacobson

Boris
Springborn

Mathieu
Desbrun

Theo
Braune

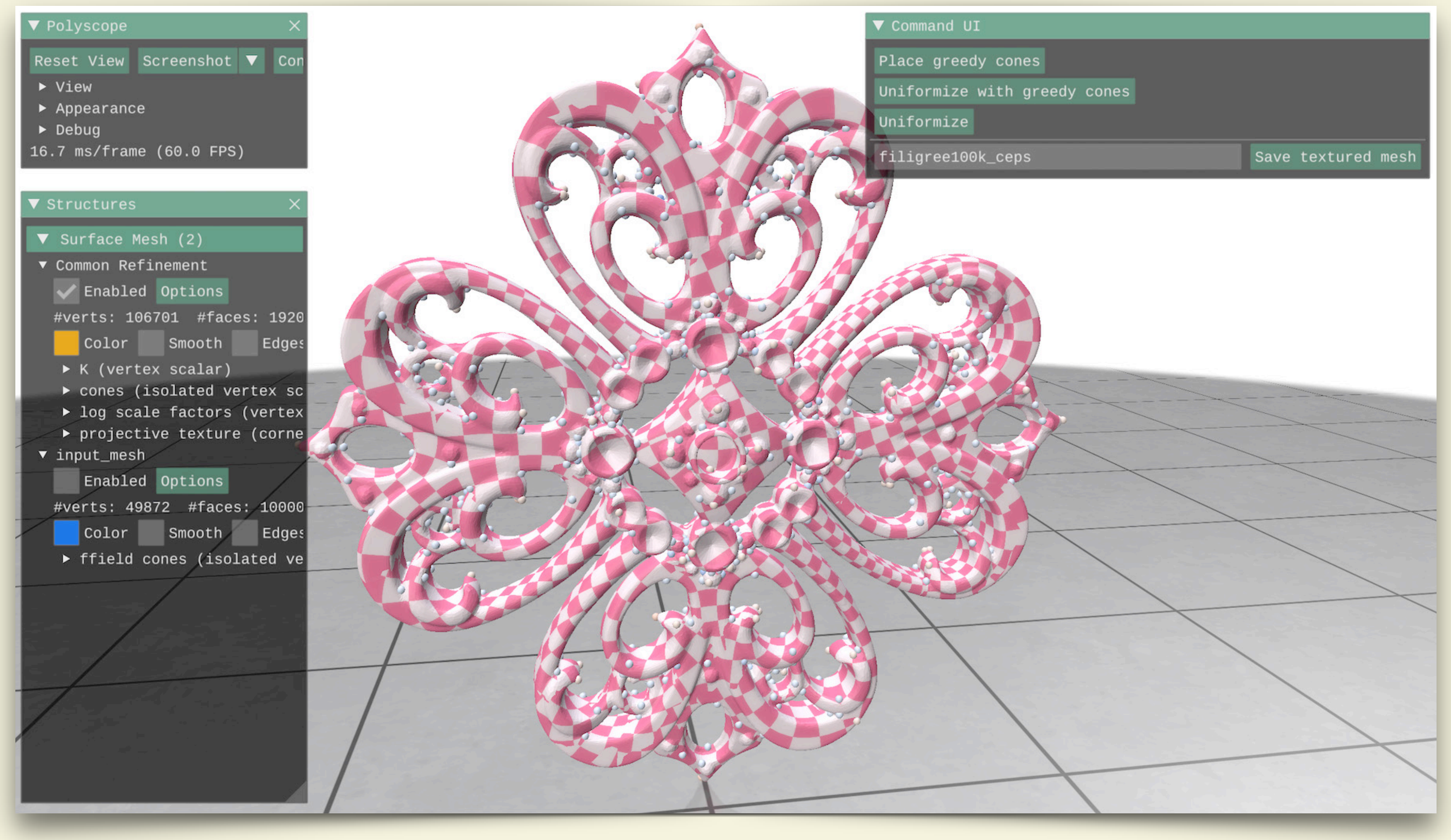
Yiying
Tong

Fernando
de Goes

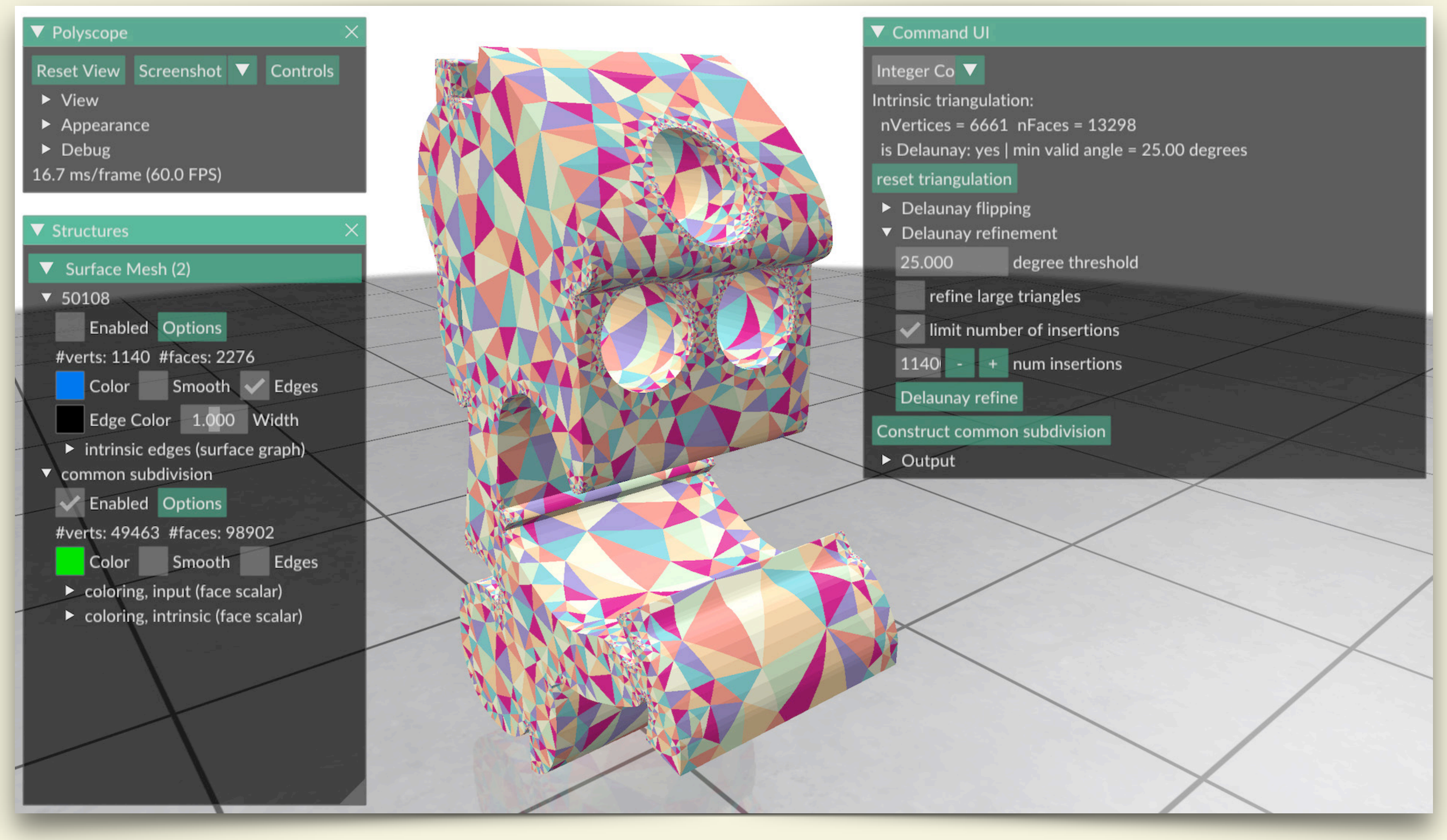
Try it yourself

markjgillespie.com

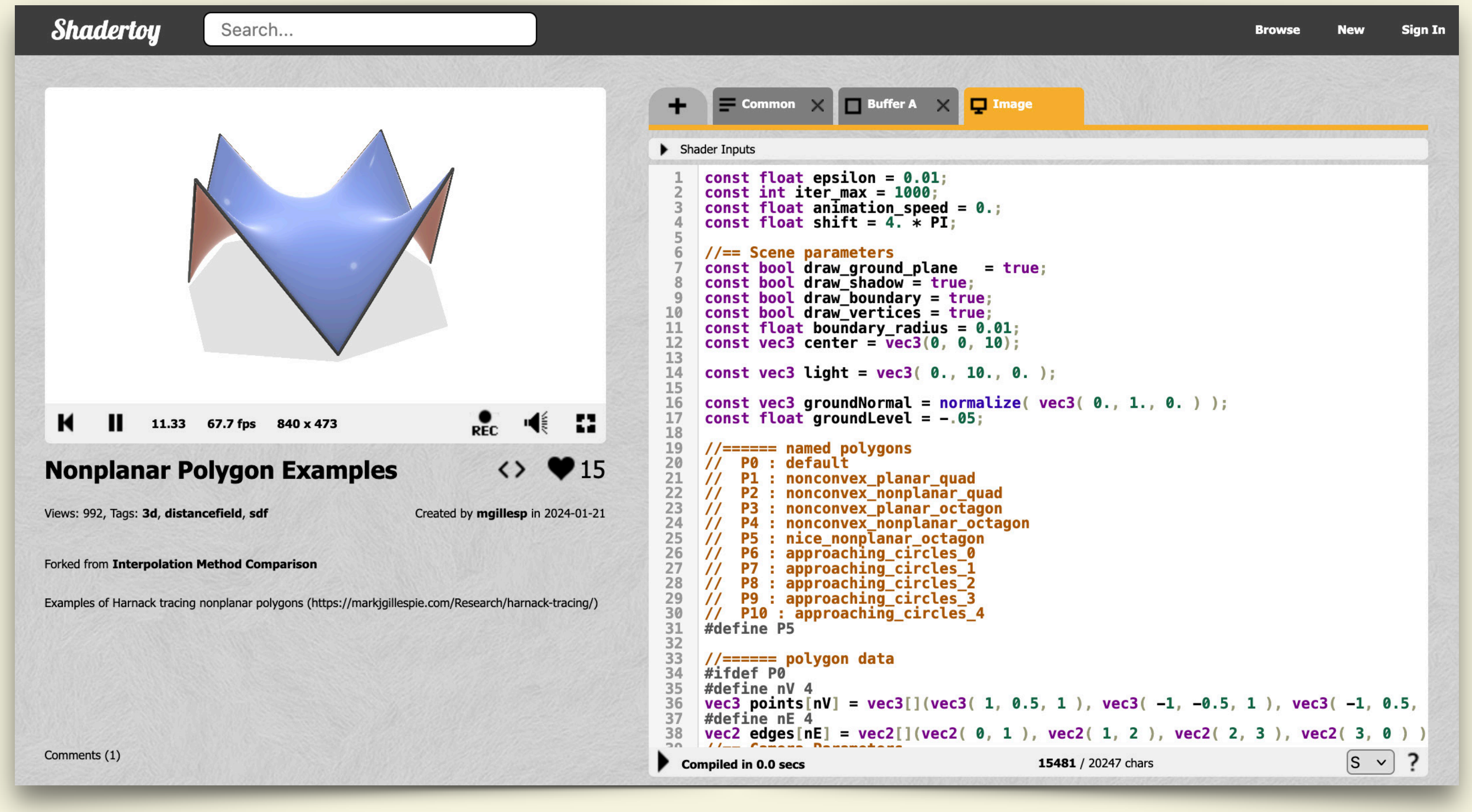
Open source implementations available at:



https://github.com/markGillespie/CEPS



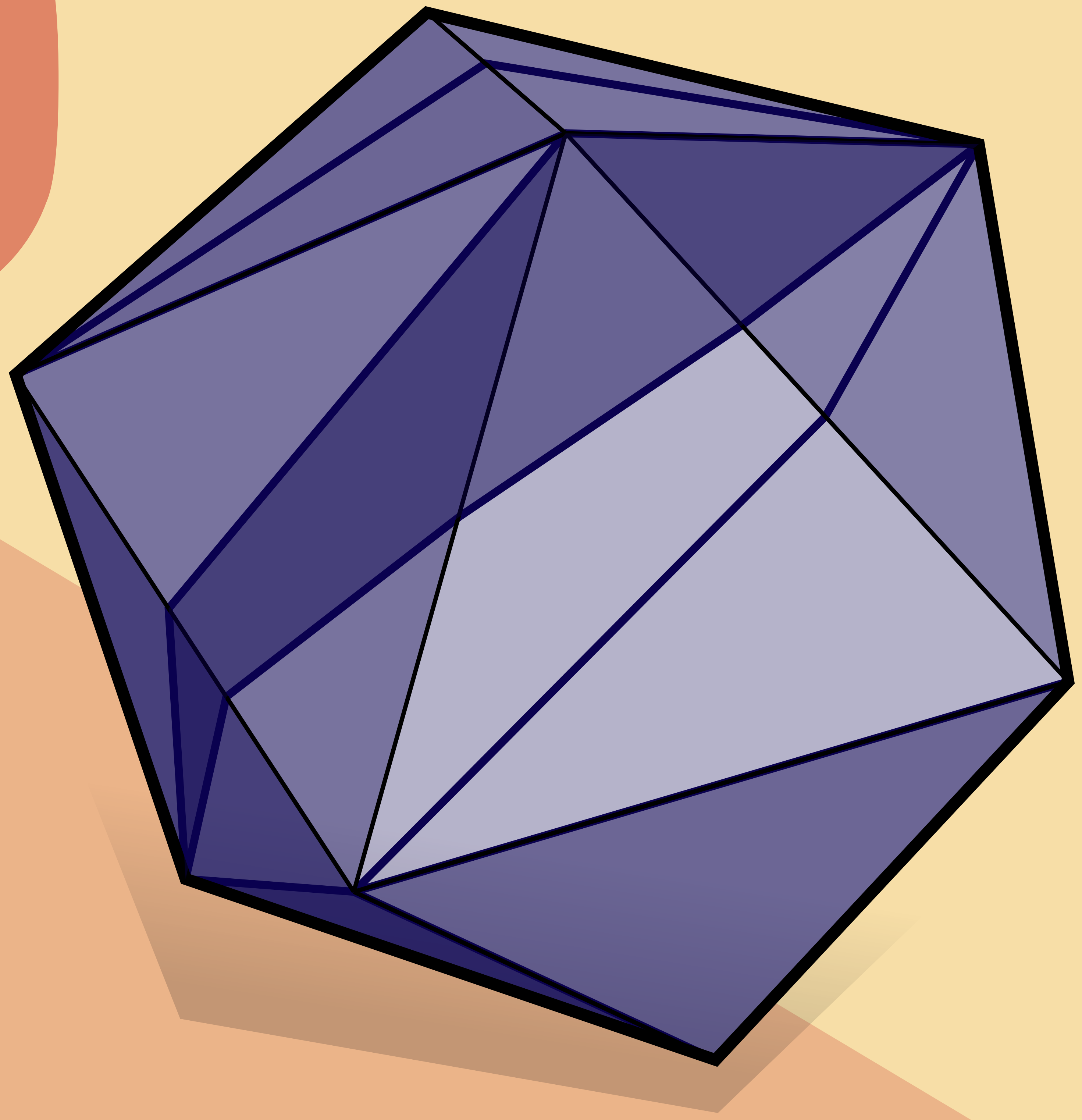
https://github.com/MarkGillespie/intrinsic-triangulations-demo



https://www.shadertoy.com/user/mgillesp

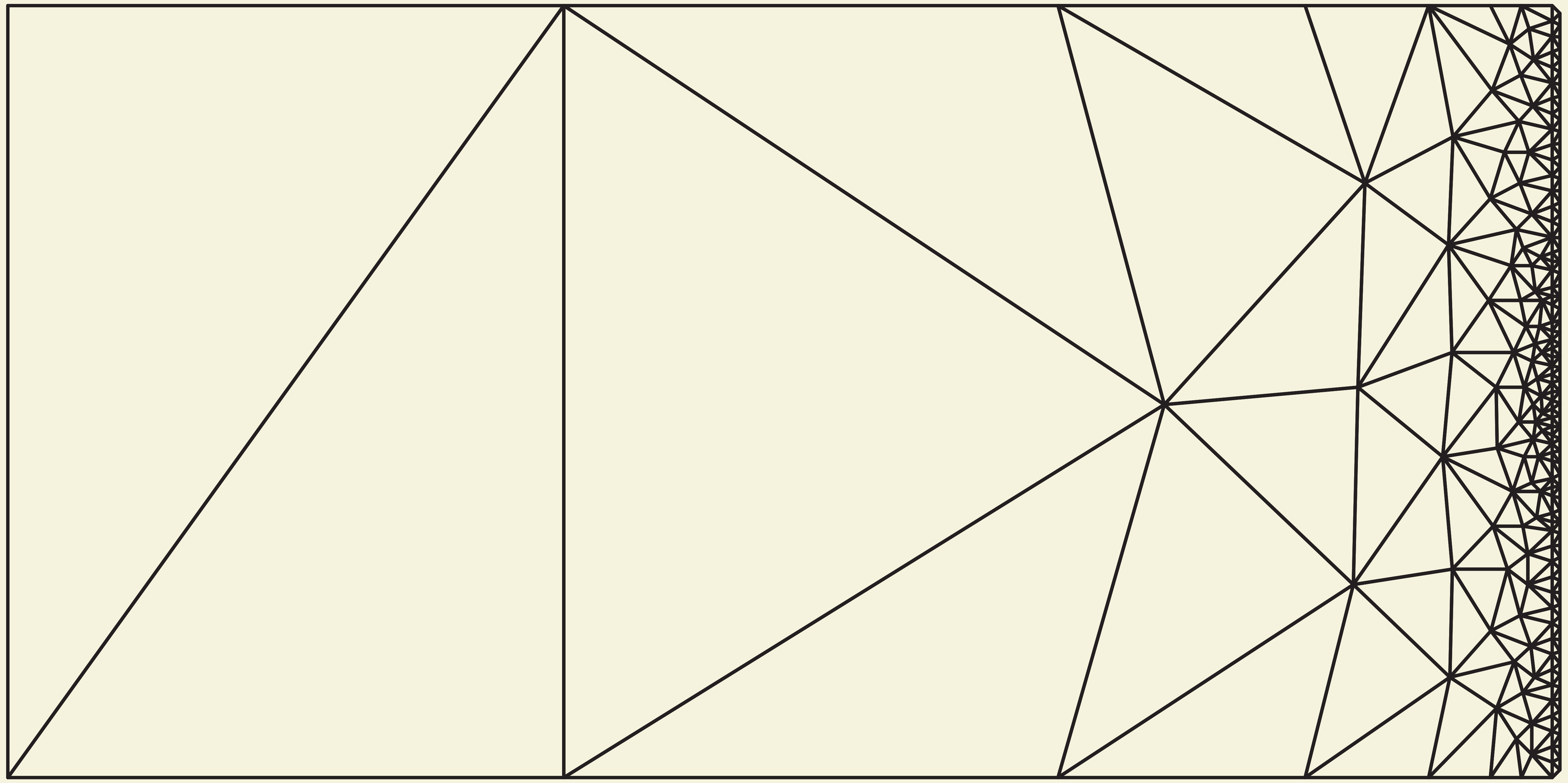
mark.gillespie81@gmail.com
markjgillespie.com

Thanks for listening

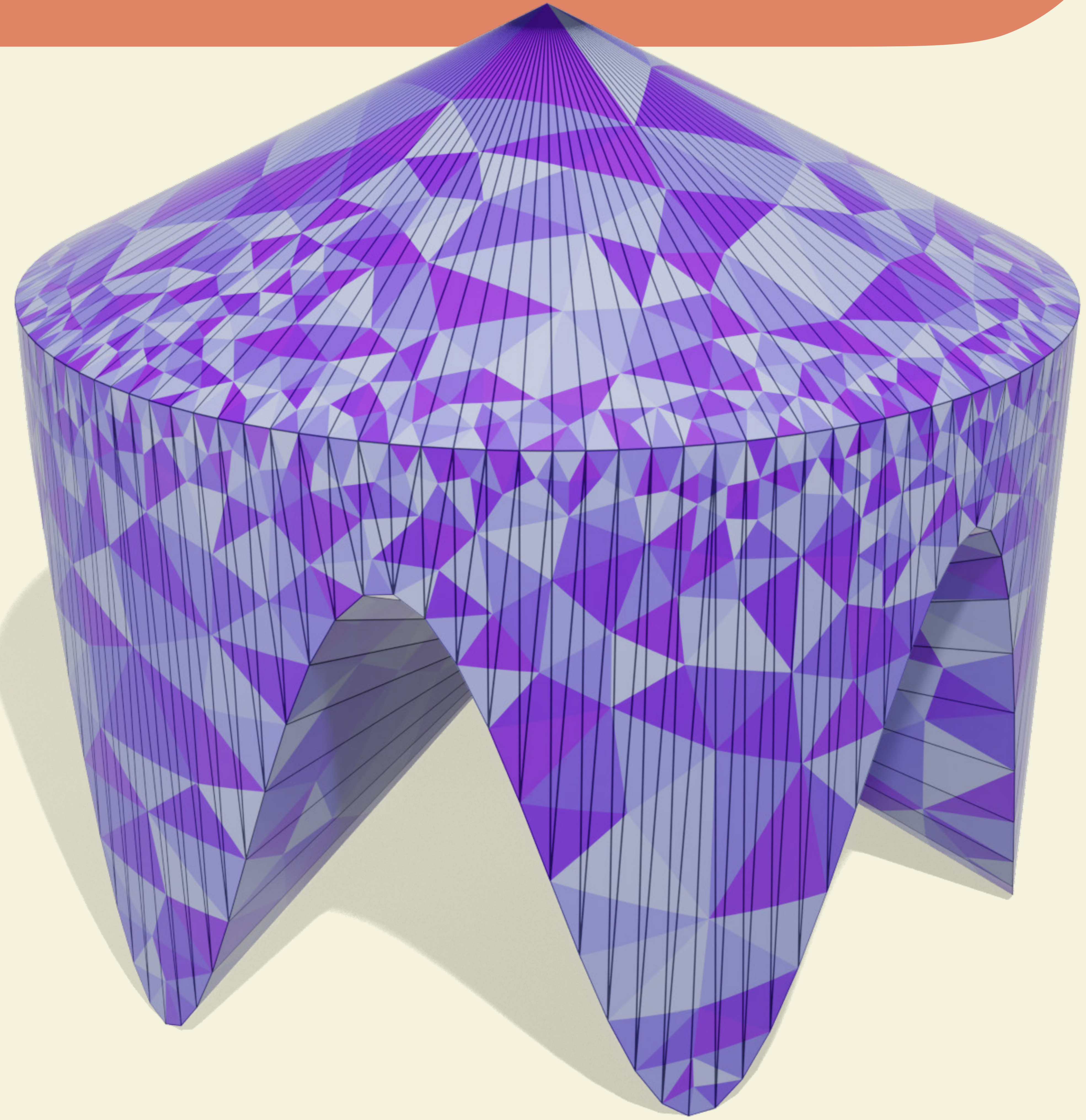


Supplemental Slides

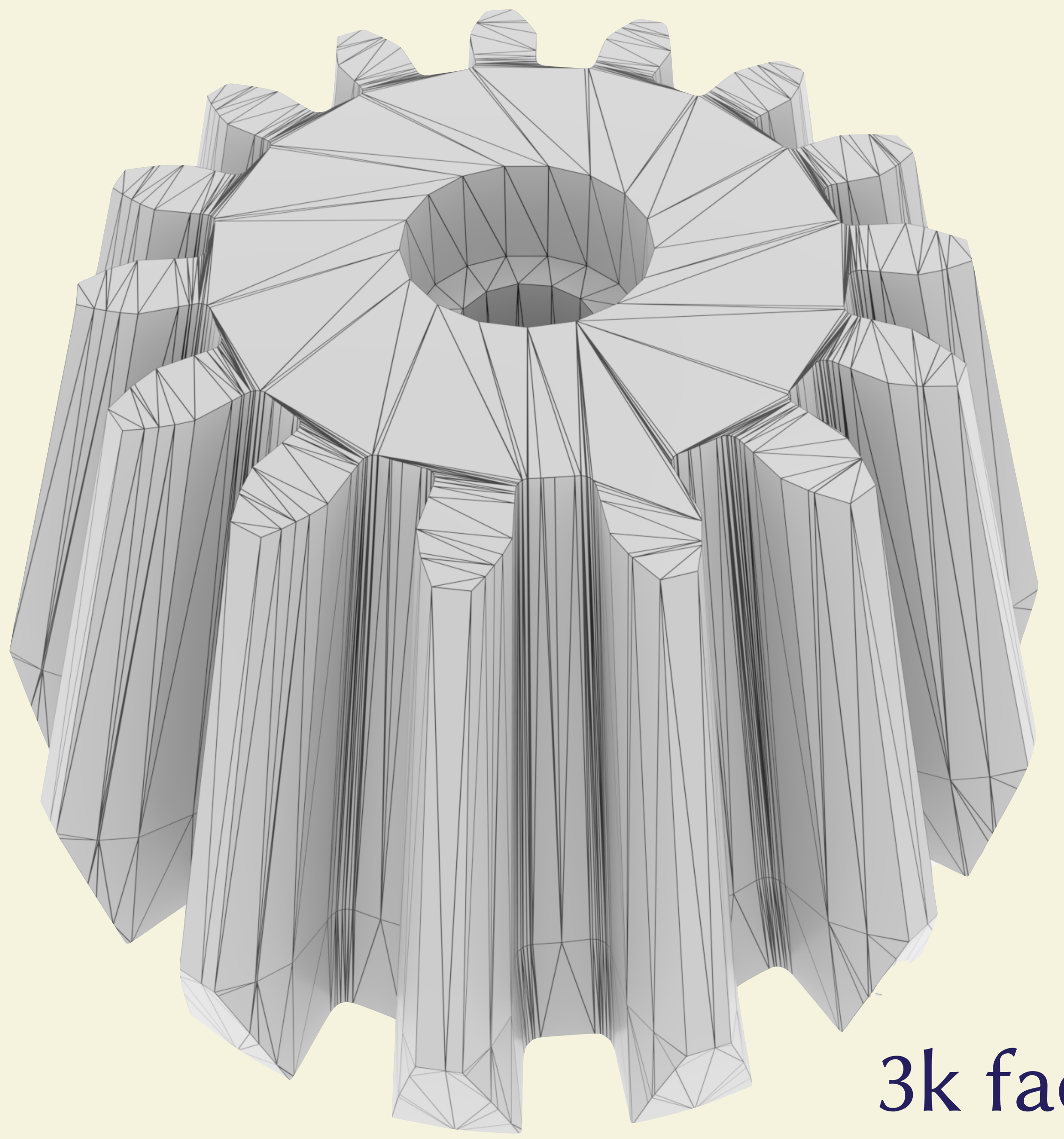
Intrinsic Delaunay refinement size grading



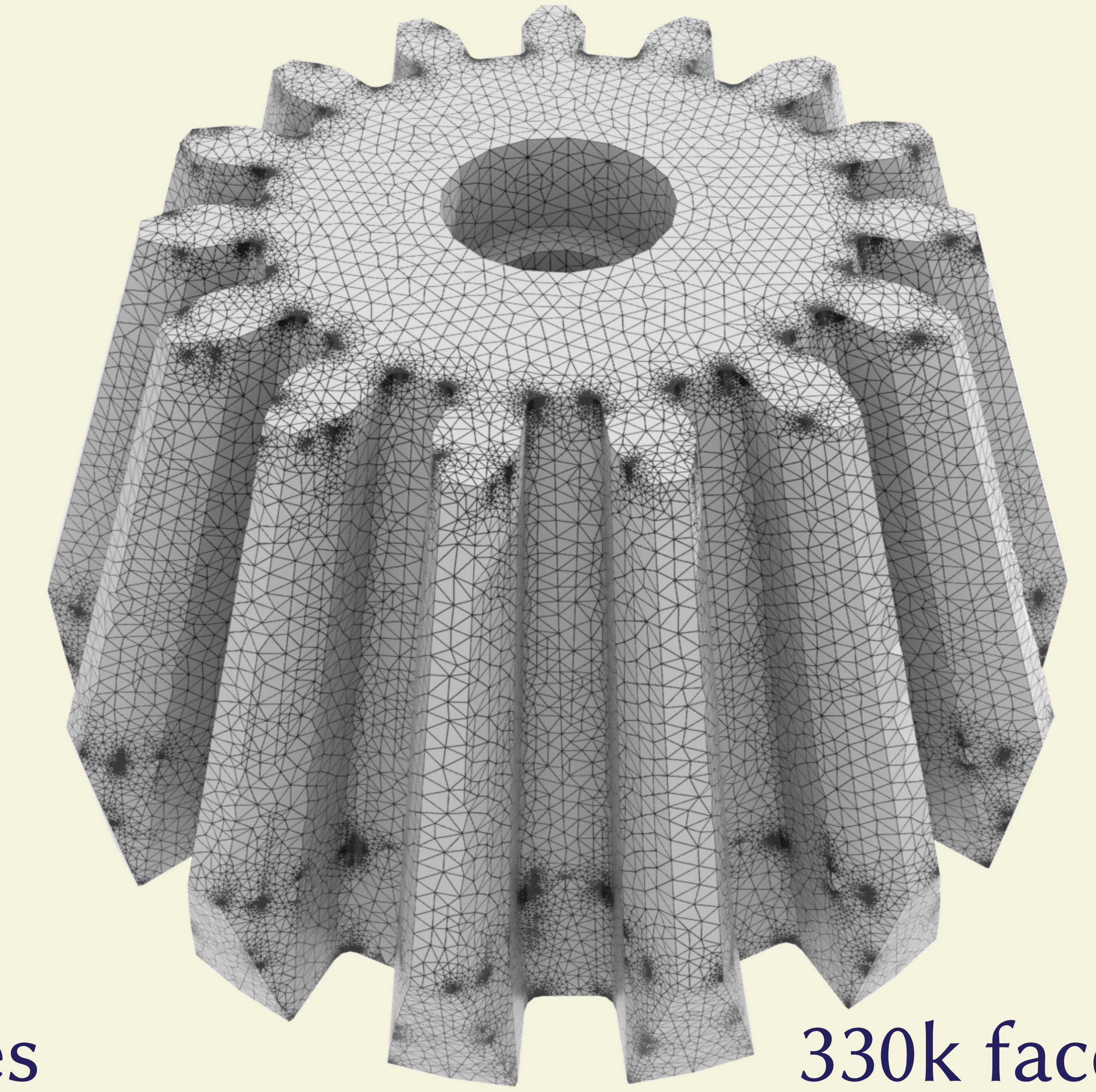
[Shewchuk 1997]



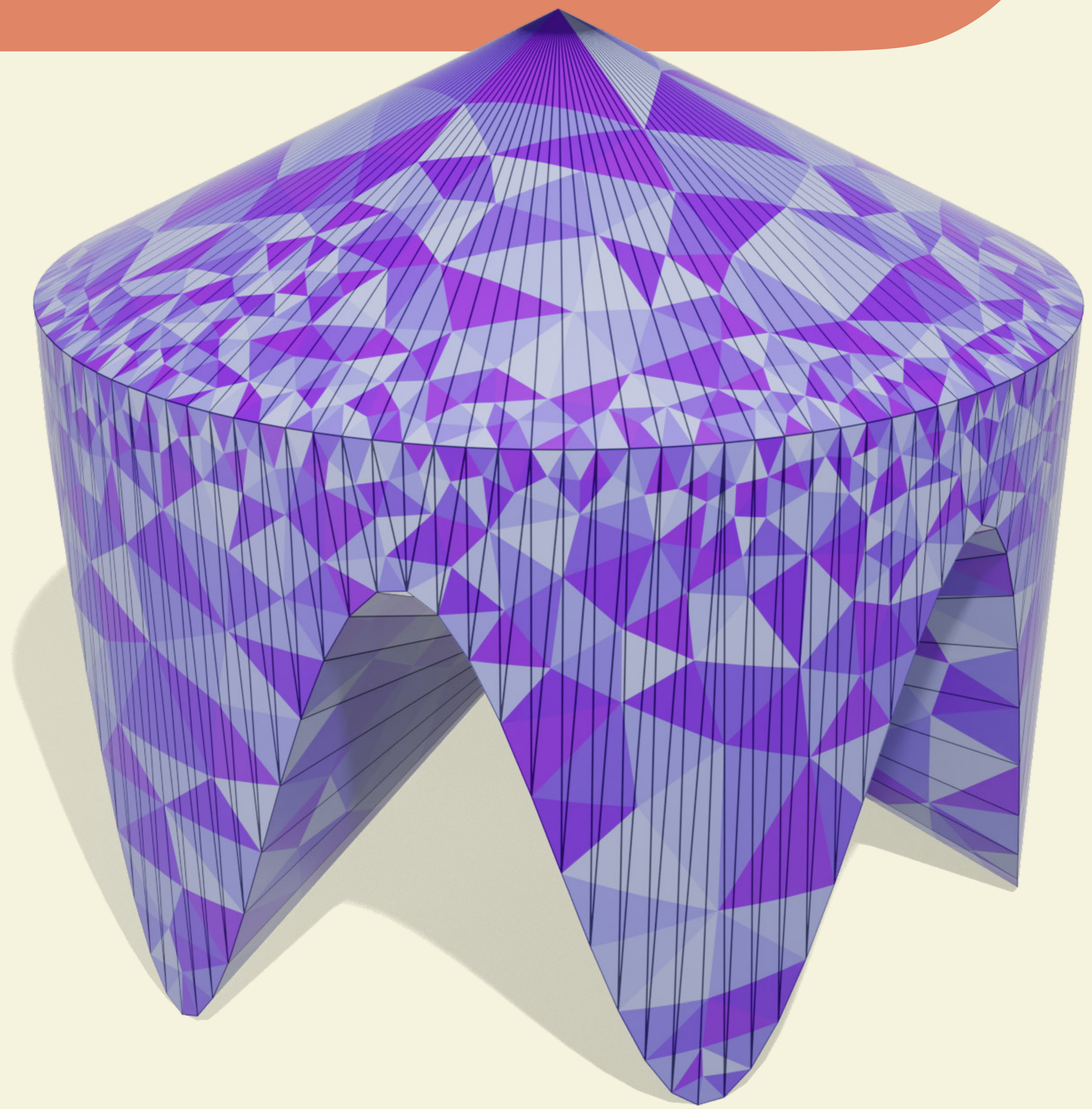
Intrinsic Delaunay refinement size grading



3k faces



330k faces



Bad basis functions

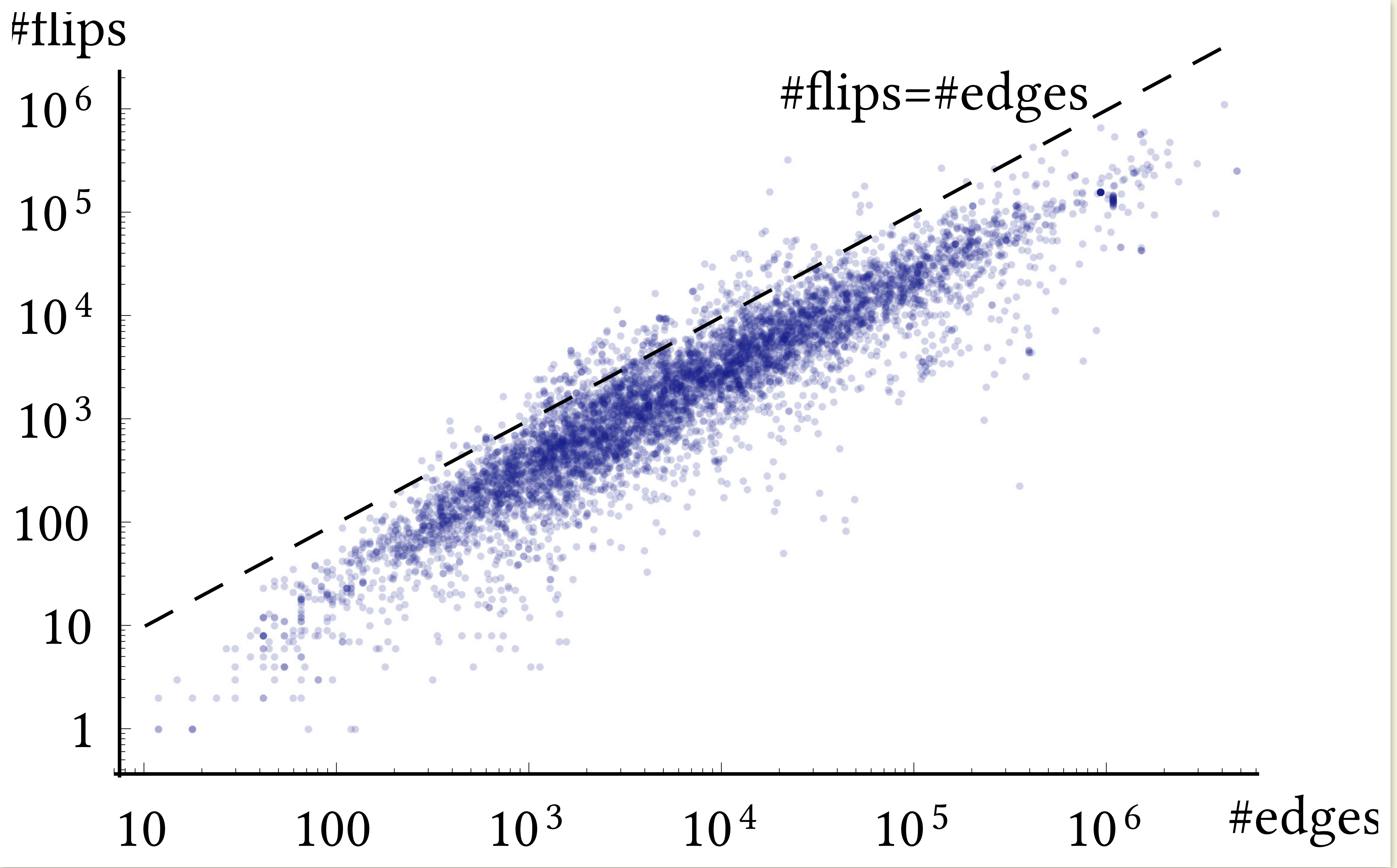
Input basis function

Intrinsic basis function

[Sharp, Soliman & Crane 2019]

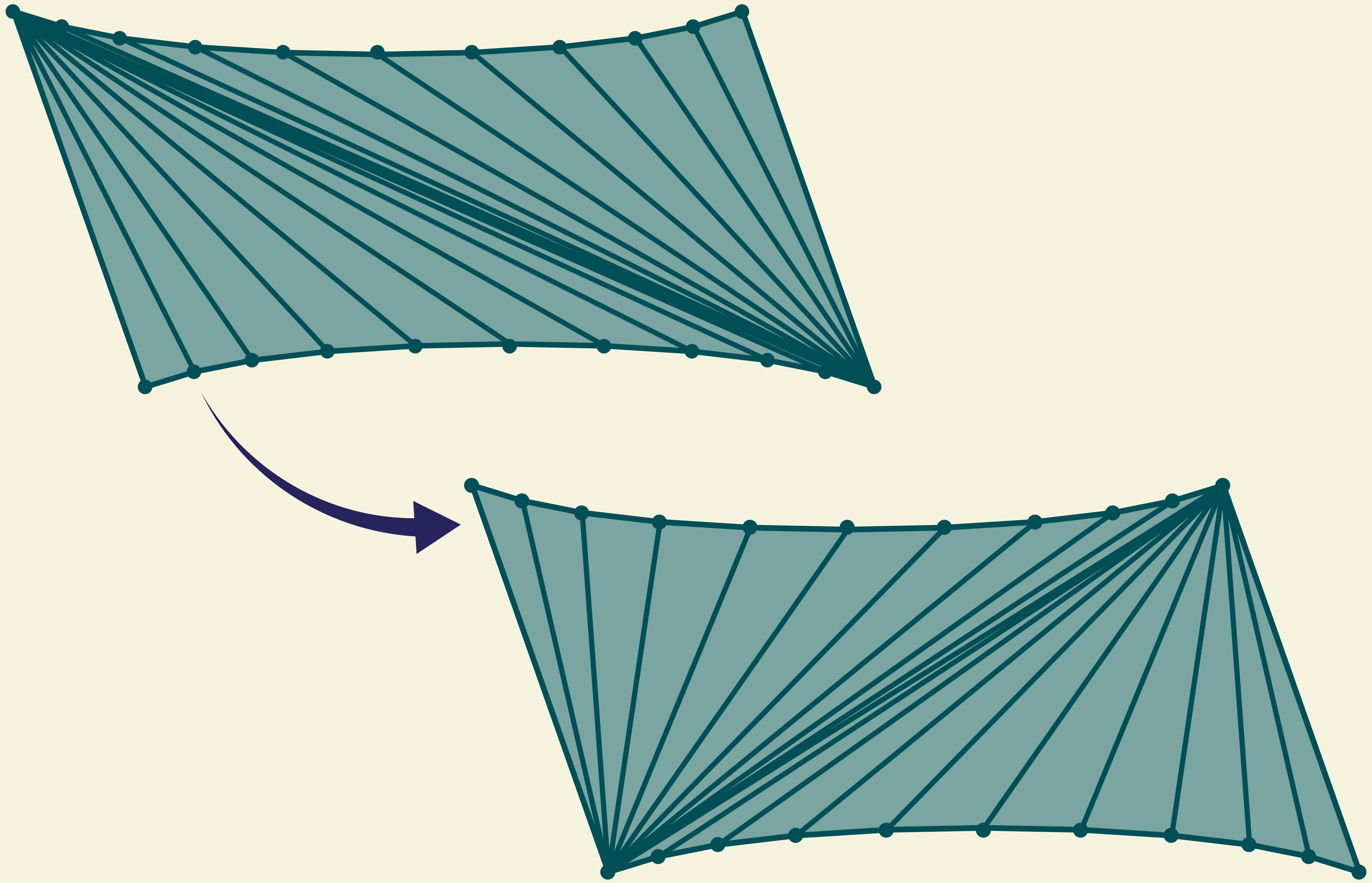
Intrinsic Delaunay flip complexity

Empirically, usually linear time



[Sharp, Soliman & Crane 2019]

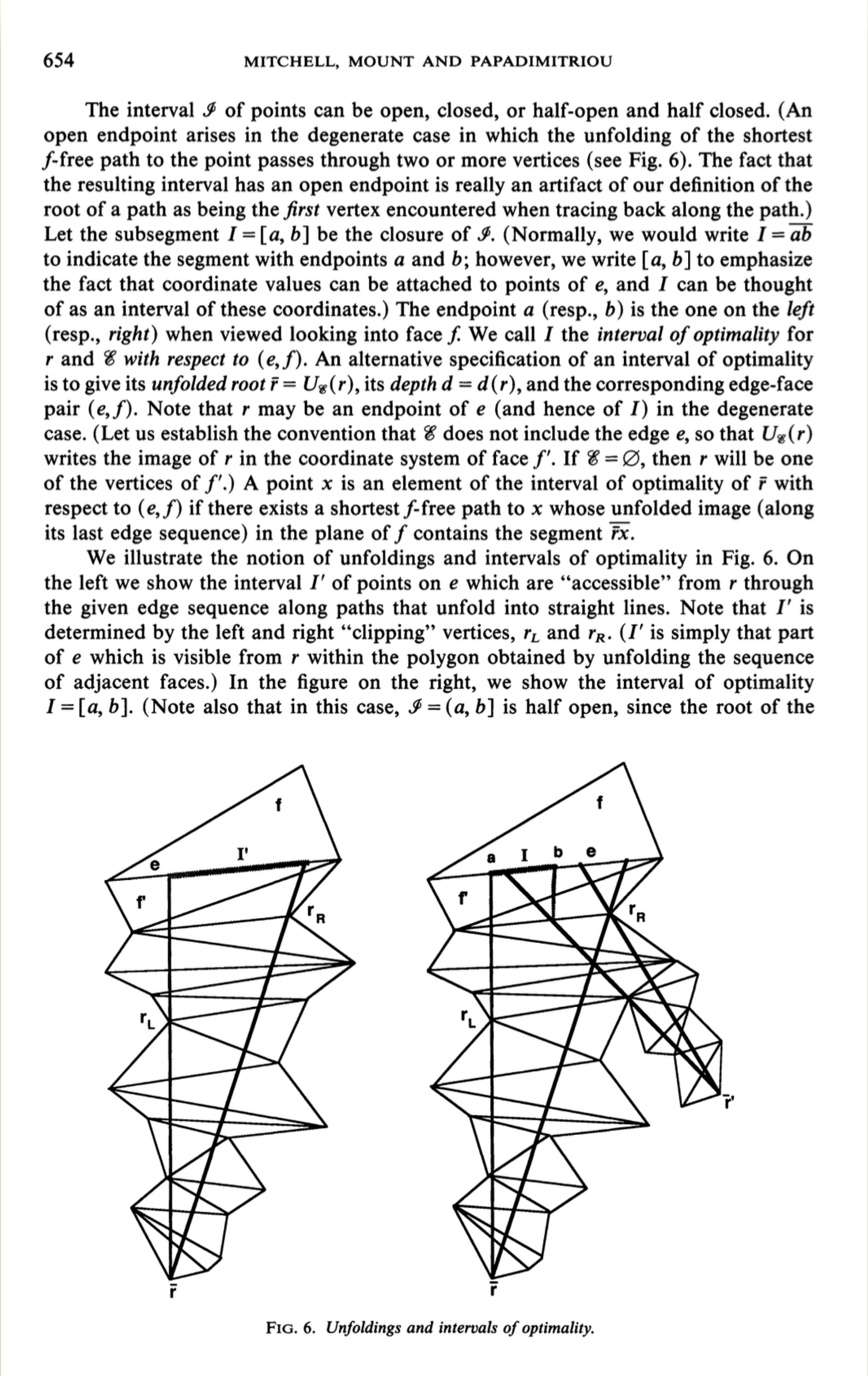
$\Omega(n^2)$ worst-case examples in the plane



Exact polyhedral geodesic distance

[Mitchell, Mount & Papadimitriou 1987]

“continuous Dijkstra”



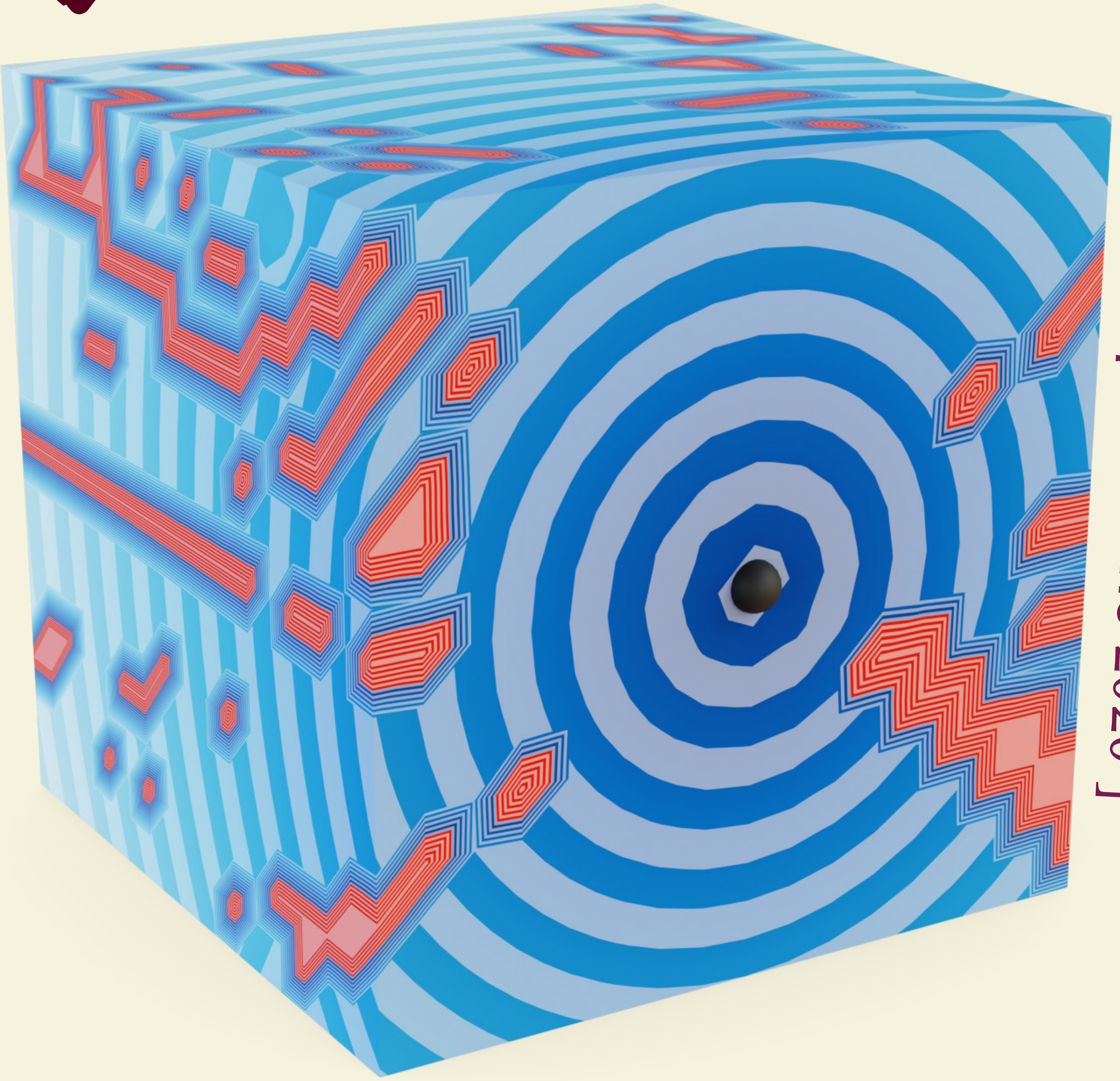
654 MITCHELL, MOUNT AND PAPADIMITRIOU

The interval $\mathscr{I}$ of points can be open, closed, or half-open and half closed. (An open endpoint arises in the degenerate case in which the unfolding of the shortest f-free path to the point passes through two or more vertices (see Fig. 6). The fact that the resulting interval has an open endpoint is really an artifact of our definition of the root of a path as being the *first* vertex encountered when tracing back along the path.) Let the subsegment $I=[a, b]$ be the closure of $\mathscr{I}$. (Normally, we would write $I=\overline{ab}$ to indicate the segment with endpoints a and b; however, we write $[a, b]$ to emphasize the fact that coordinate values can be attached to points of e, and I can be thought of as an interval of these coordinates.) The endpoint a (resp., b) is the one on the *left* (resp., *right*) when viewed looking into face f. We call I the *interval of optimality* for r and $\mathscr{E}$ *with respect to* (e, f). An alternative specification of an interval of optimality is to give its *unfolded root* $\bar{r} = U_{\mathscr{E}}(r)$, its *depth* $d = d(r)$, and the corresponding edge-face pair (e, f). Note that r may be an endpoint of e (and hence of I) in the degenerate case. (Let us establish the convention that $\mathscr{E}$ does not include the edge e, so that $U_{\mathscr{E}}(r)$ writes the image of r in the coordinate system of face f'. If $\mathscr{E}=\varnothing$, then r will be one of the vertices of f'.) A point x is an element of the interval of optimality of $\bar{r}$ with respect to (e, f) if there exists a shortest f-free path to x whose unfolded image (along its last edge sequence) in the plane of f contains the segment $\overline{\bar{r}x}$.

We illustrate the notion of unfoldings and intervals of optimality in Fig. 6. On the left we show the interval I' of points on e which are “accessible” from r through the given edge sequence along paths that unfold into straight lines. Note that I' is determined by the left and right “clipping” vertices, r_L and r_R. (I' is simply that part of e which is visible from r within the polygon obtained by unfolding the sequence of adjacent faces.) In the figure on the right, we show the interval of optimality $I=[a, b]$. (Note also that in this case, $\mathscr{I}=(a, b]$ is half open, since the root of the

FIG. 6. *Unfoldings and intervals of optimality.*

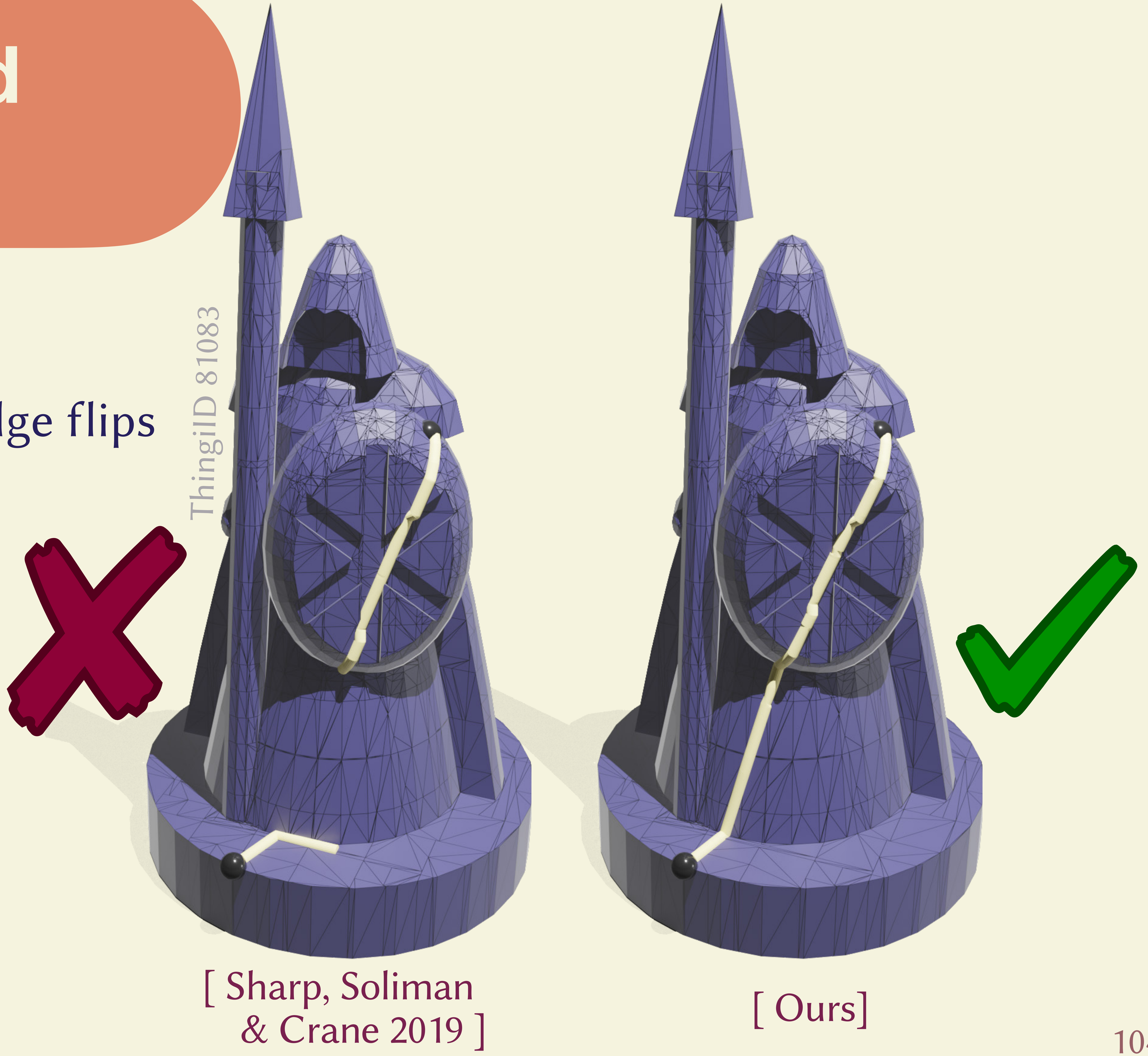
floating point error



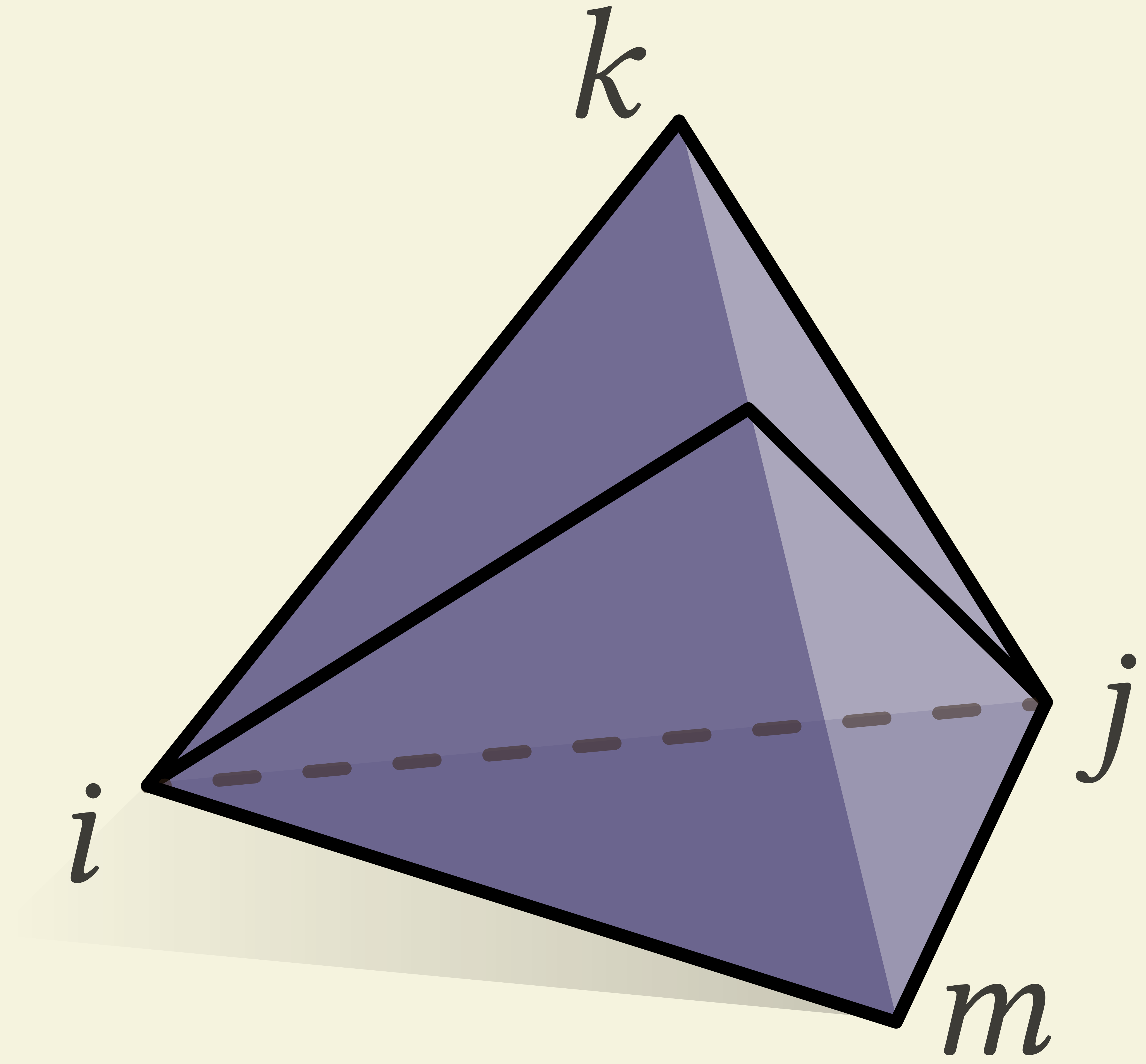
[Sharp & Crane 2020]

Example: Flip-Based Geodesic Paths

- FlipOut [Sharp & Crane 2020]:
  - computes geodesic paths via edge flips



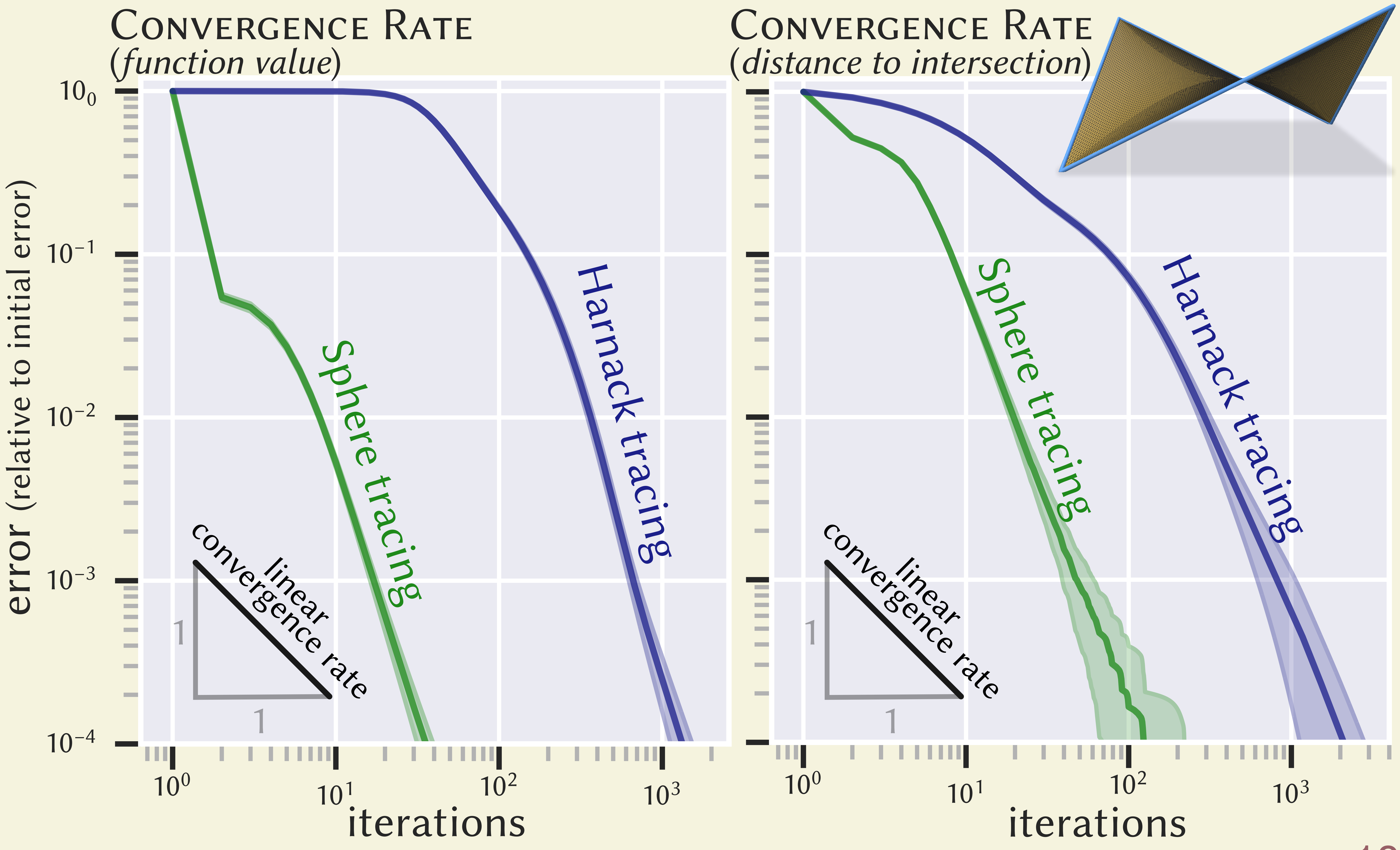
Normal coordinates are not enough to encode correspondence



Harnack tracing convergence

Theorem [**Gillespie**, Yang, Botch & Crane 2024]

Suppose the radius $R(x)$ and shift $c(x)$ are compatible, in the sense that $f(x) - c(x) > 0$ on the ball $B_{R(x_0)}(x_0)$ for all x_0, and that $R(x) > 0$. Then Harnack tracing converges linearly to the first intersection of $r(t)$ with $\{x \ : \ f(x) = 0\}$, so long as an intersection exists.



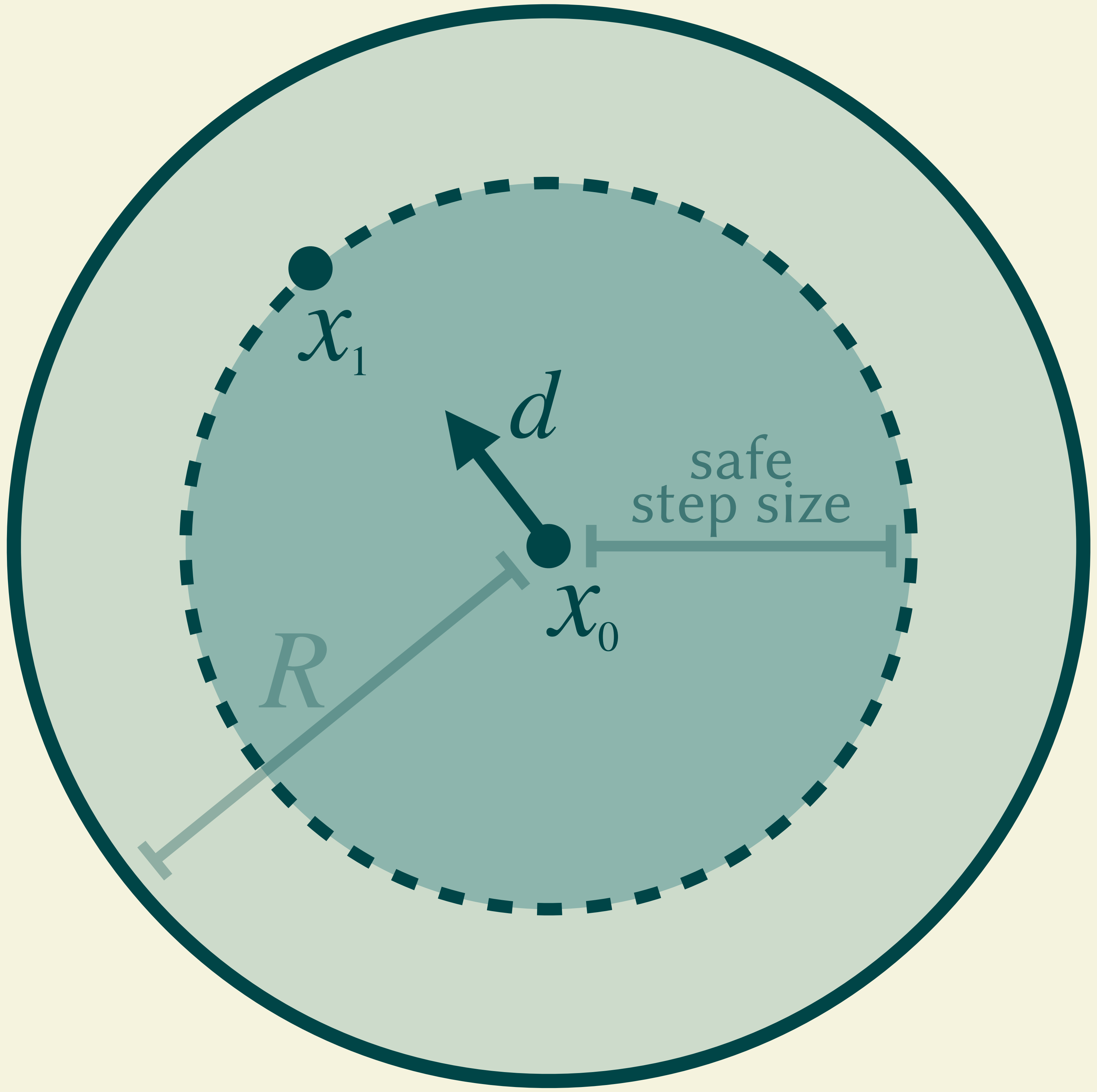
Algorithm sketch

Harnack Tracing

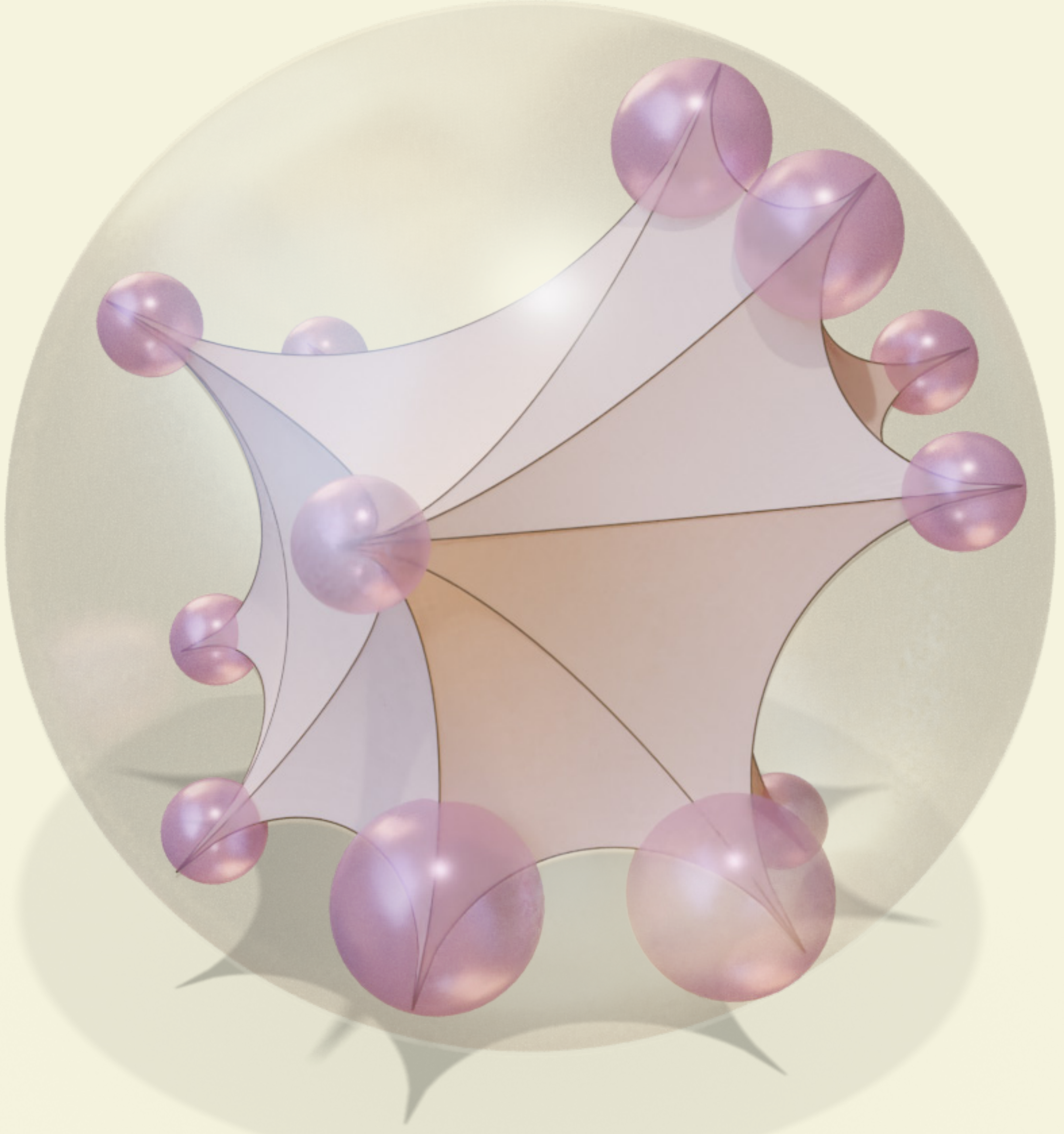
Starting from point x_0 in direction d:

- Pick ball radius
- Shift f to be positive on ball
- Calculate safe step size
- Take safe step in ray direction

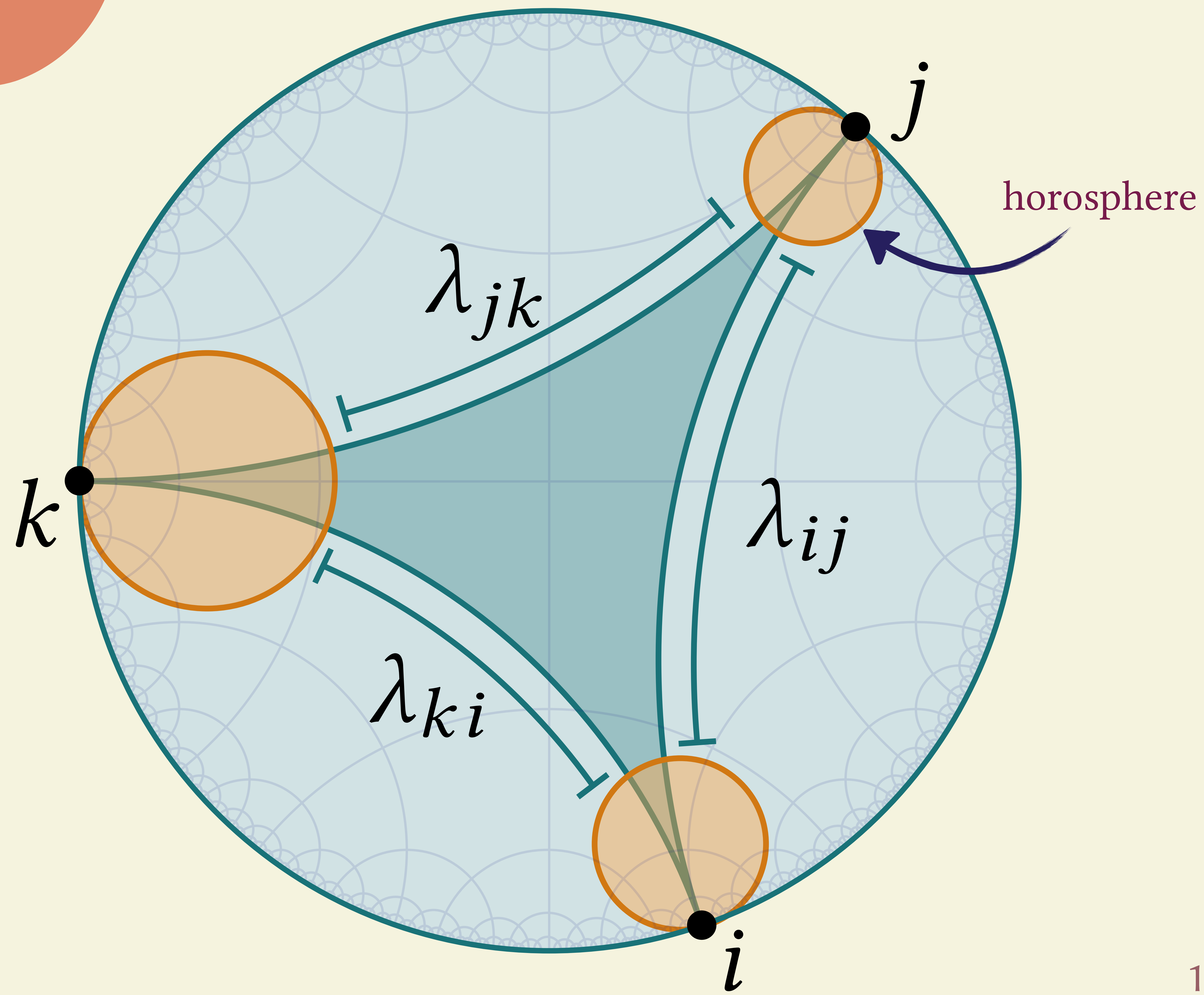
Repeat until f is sufficiently close to f^*



The Poincare disk

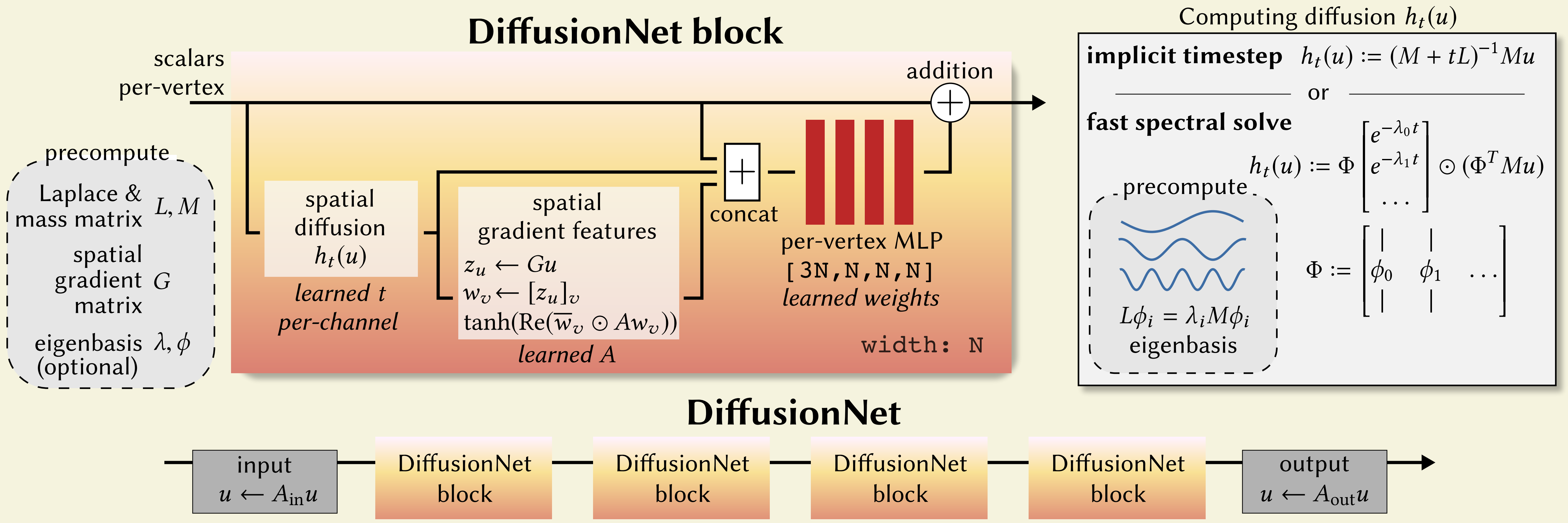


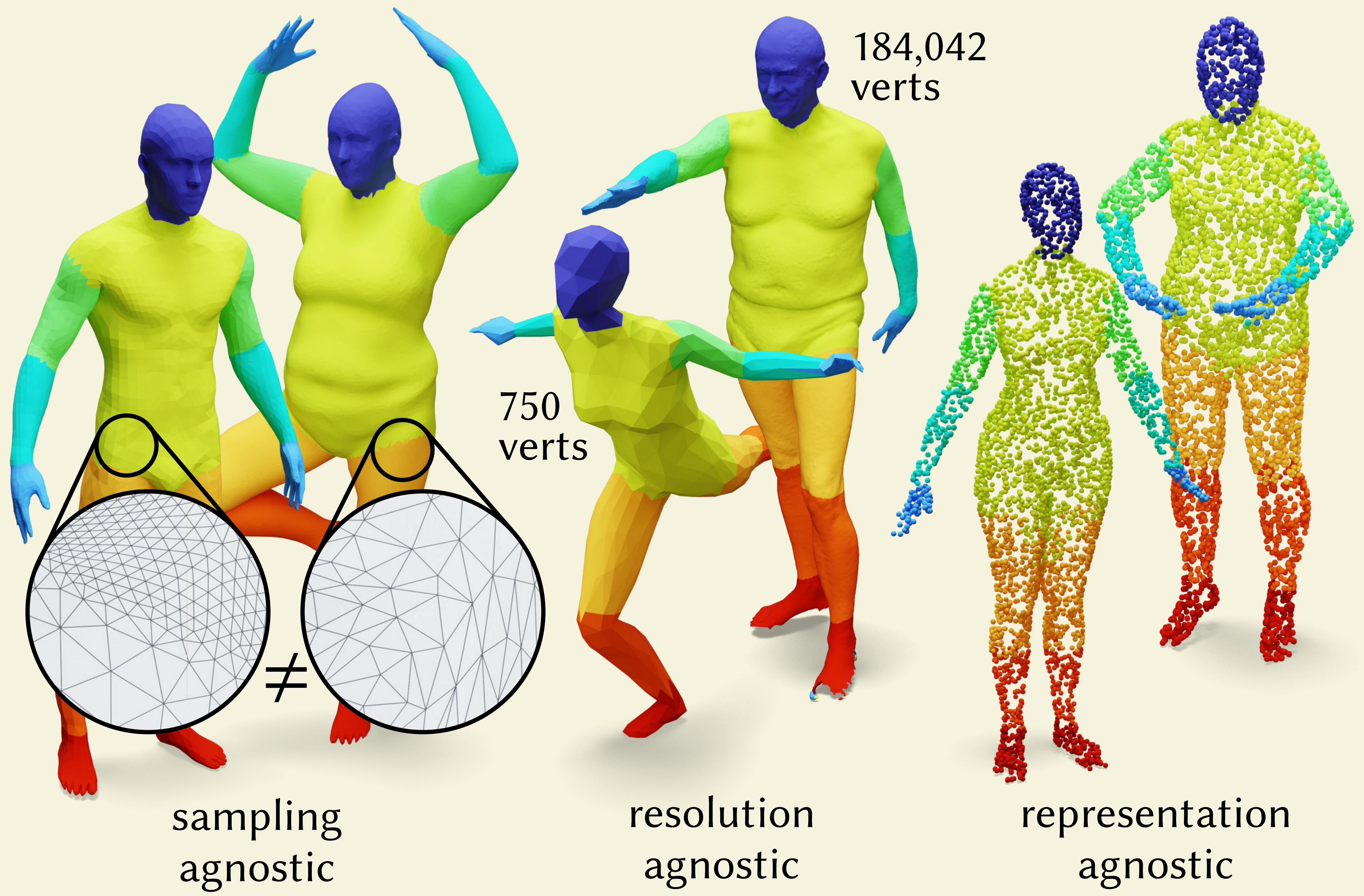
ideal hyperbolic polyhedron



Intrinsic triangulations for learning on surfaces

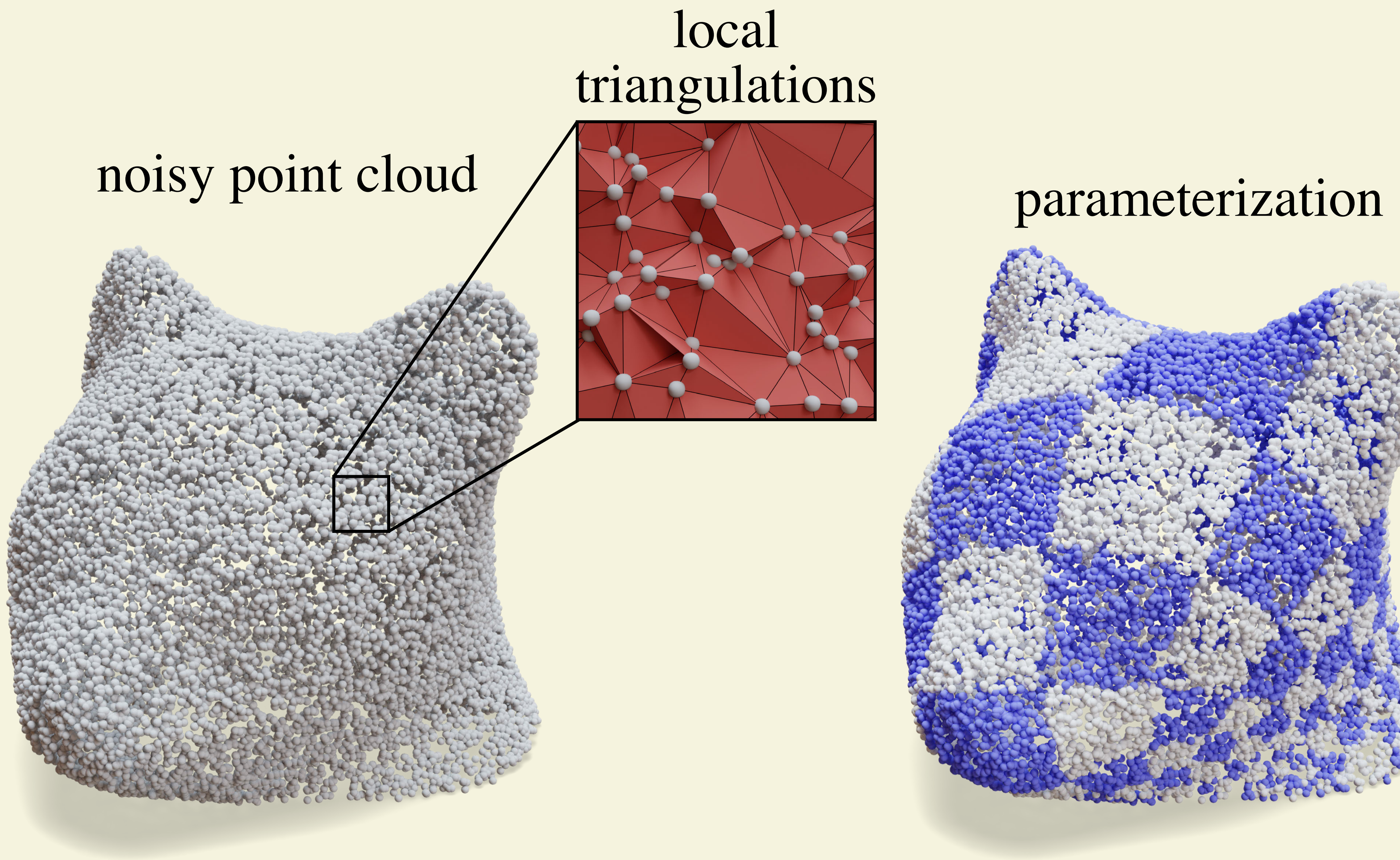
[Sharp *et al.* 2022]

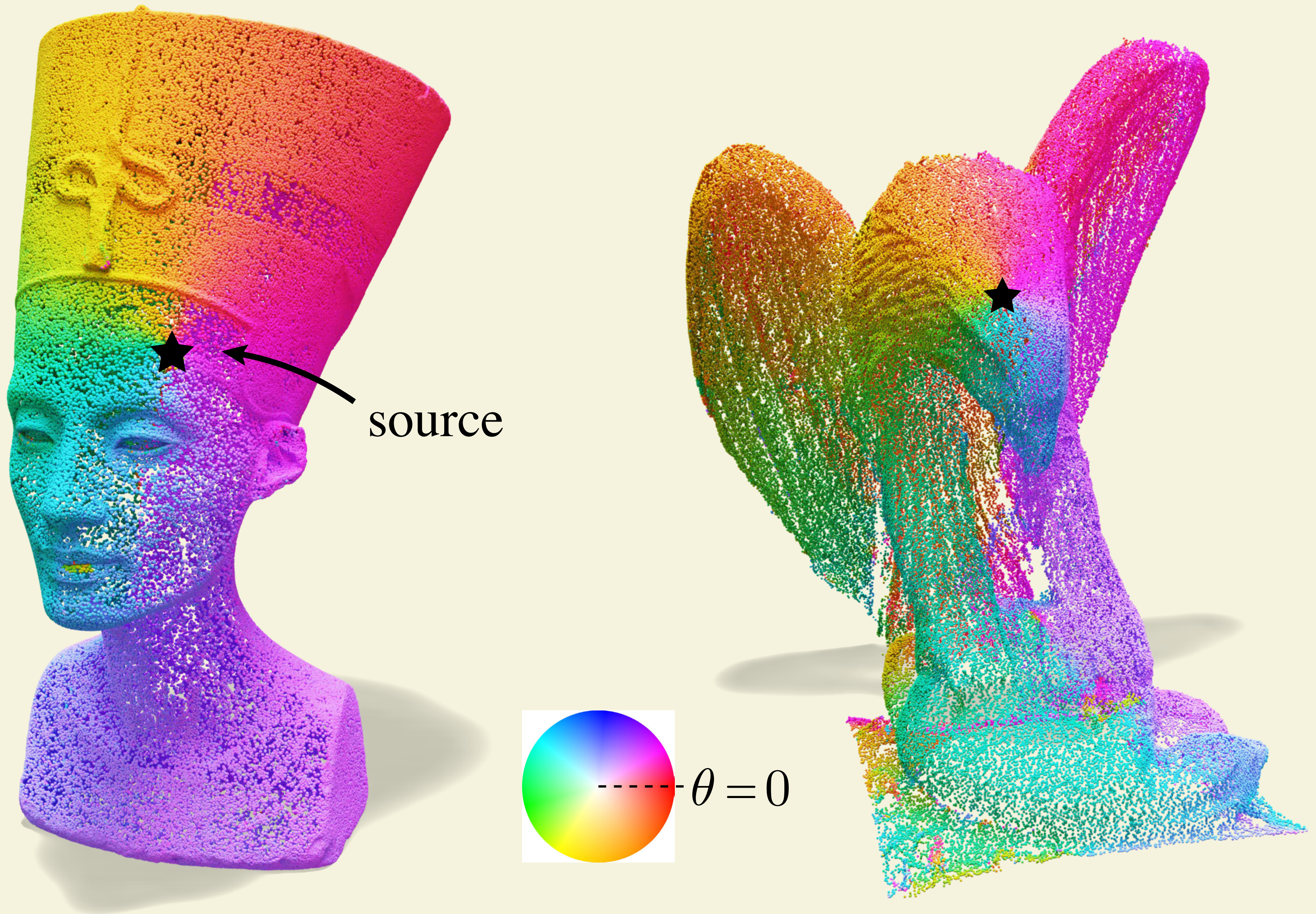




Point cloud processing

[Sharp & Crane 2020]



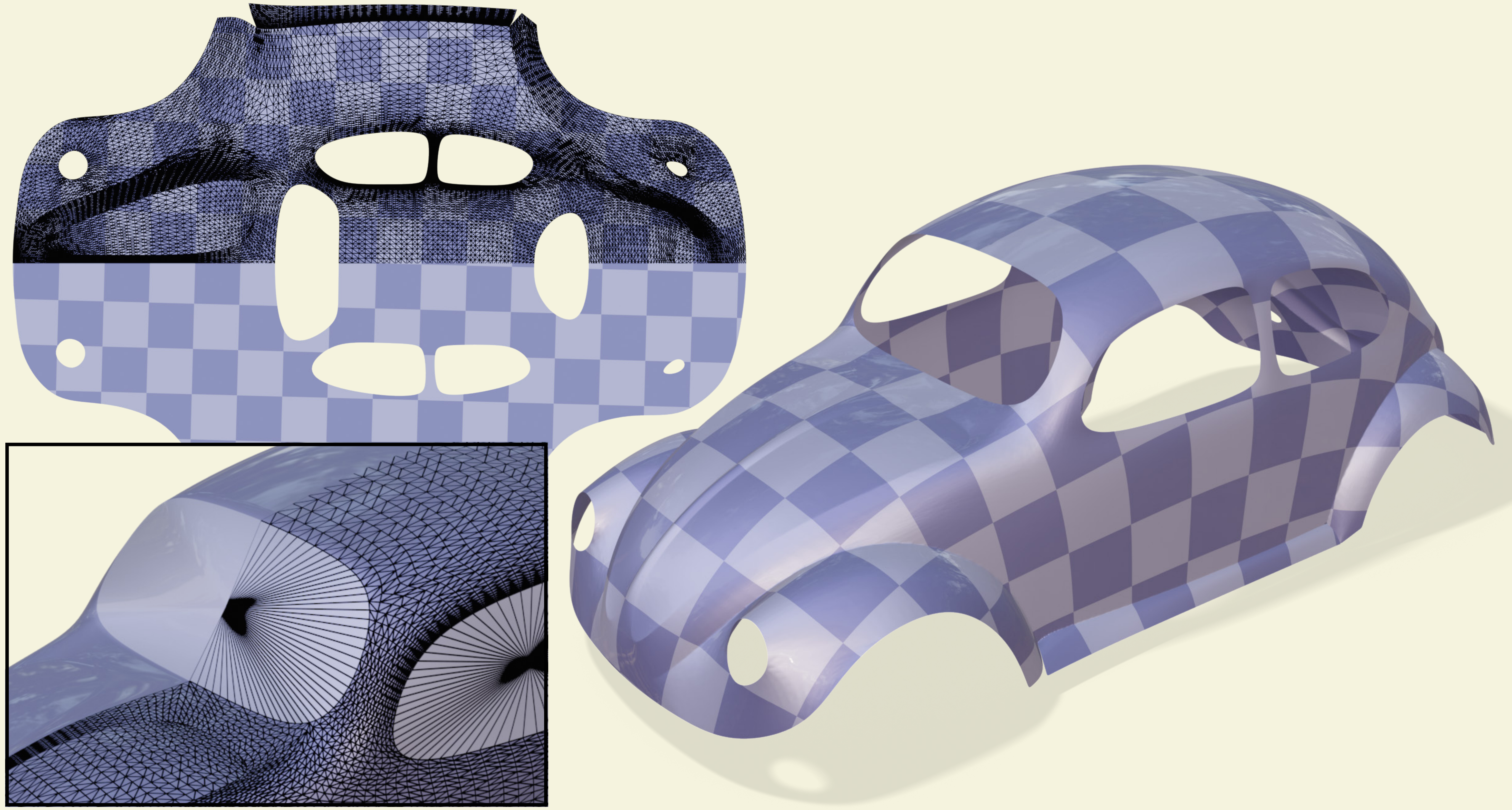


Ptolemy flips improve performance

Boundary conditions

Multiply-connected domains

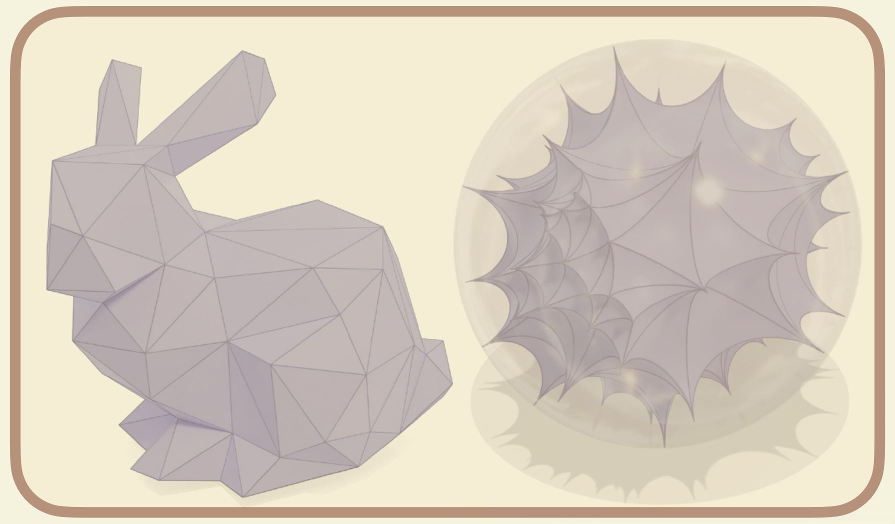
No hole filling

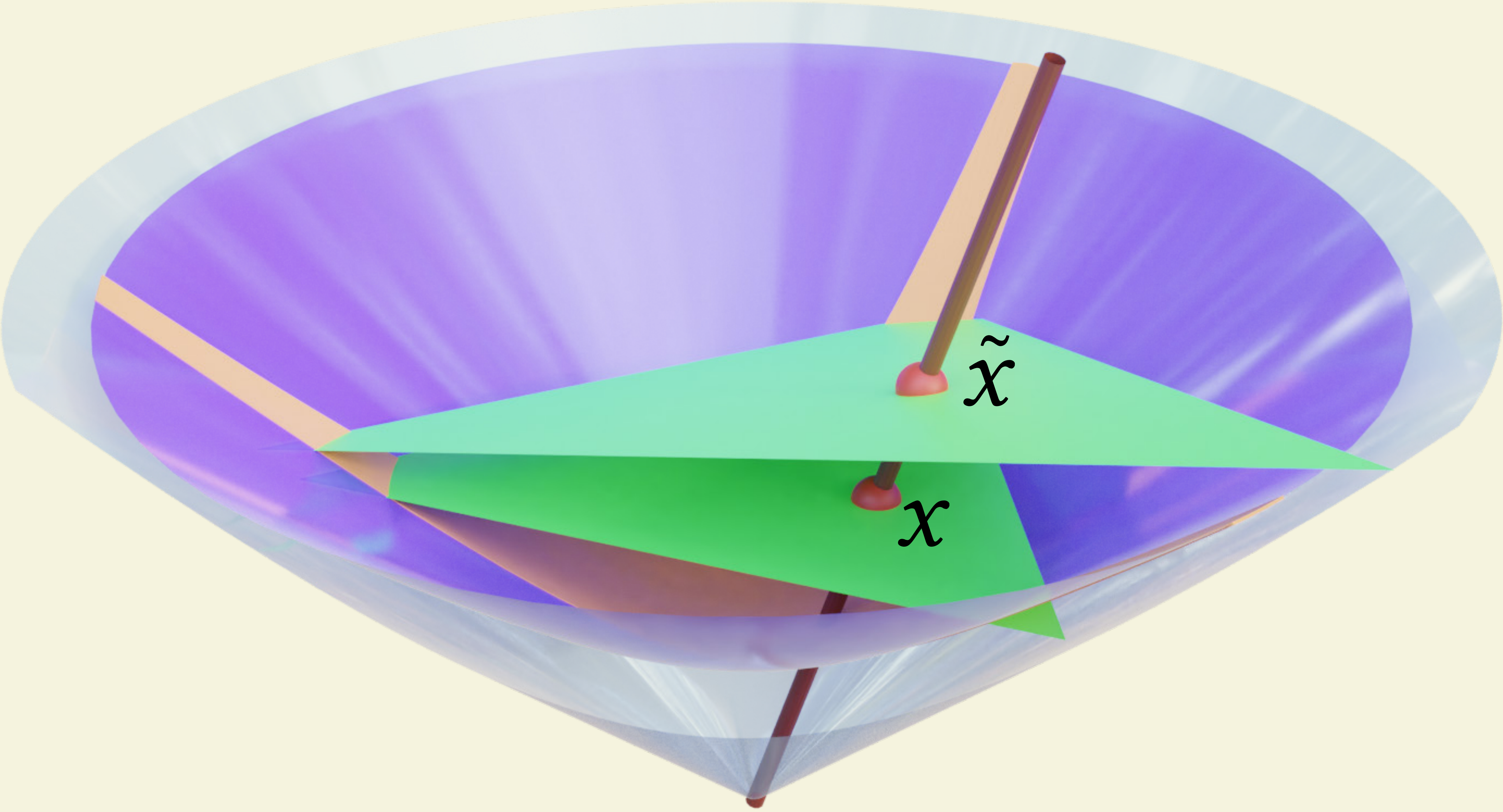


Hole filling

[Hefetz+2019]

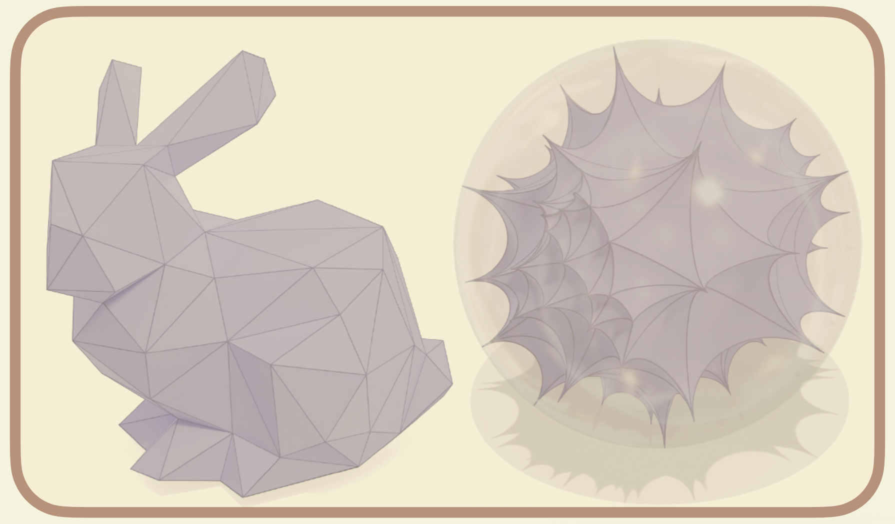
Interpolation in the hyperboloid model

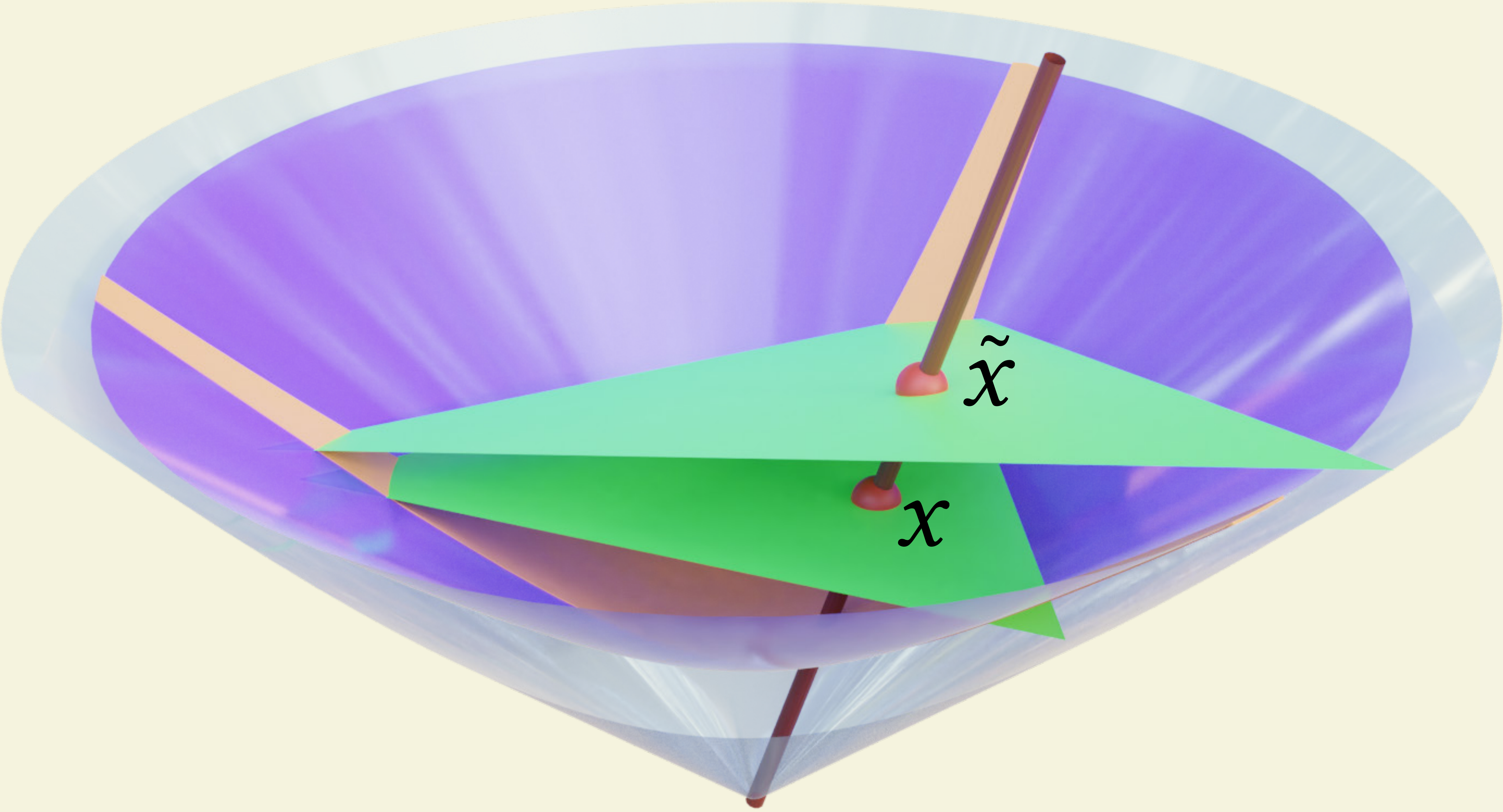




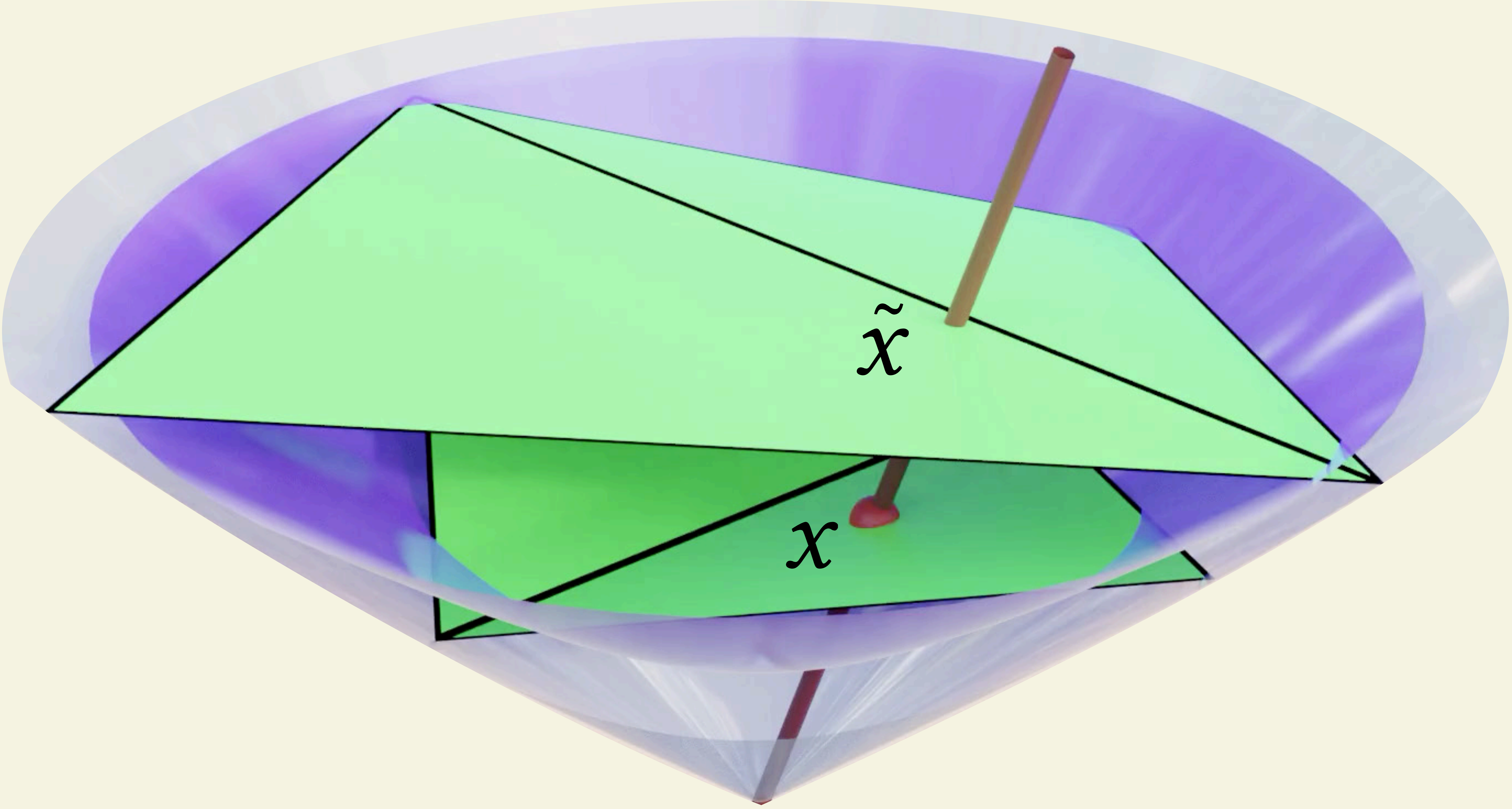
fixed triangulation

Interpolation in the hyperboloid model



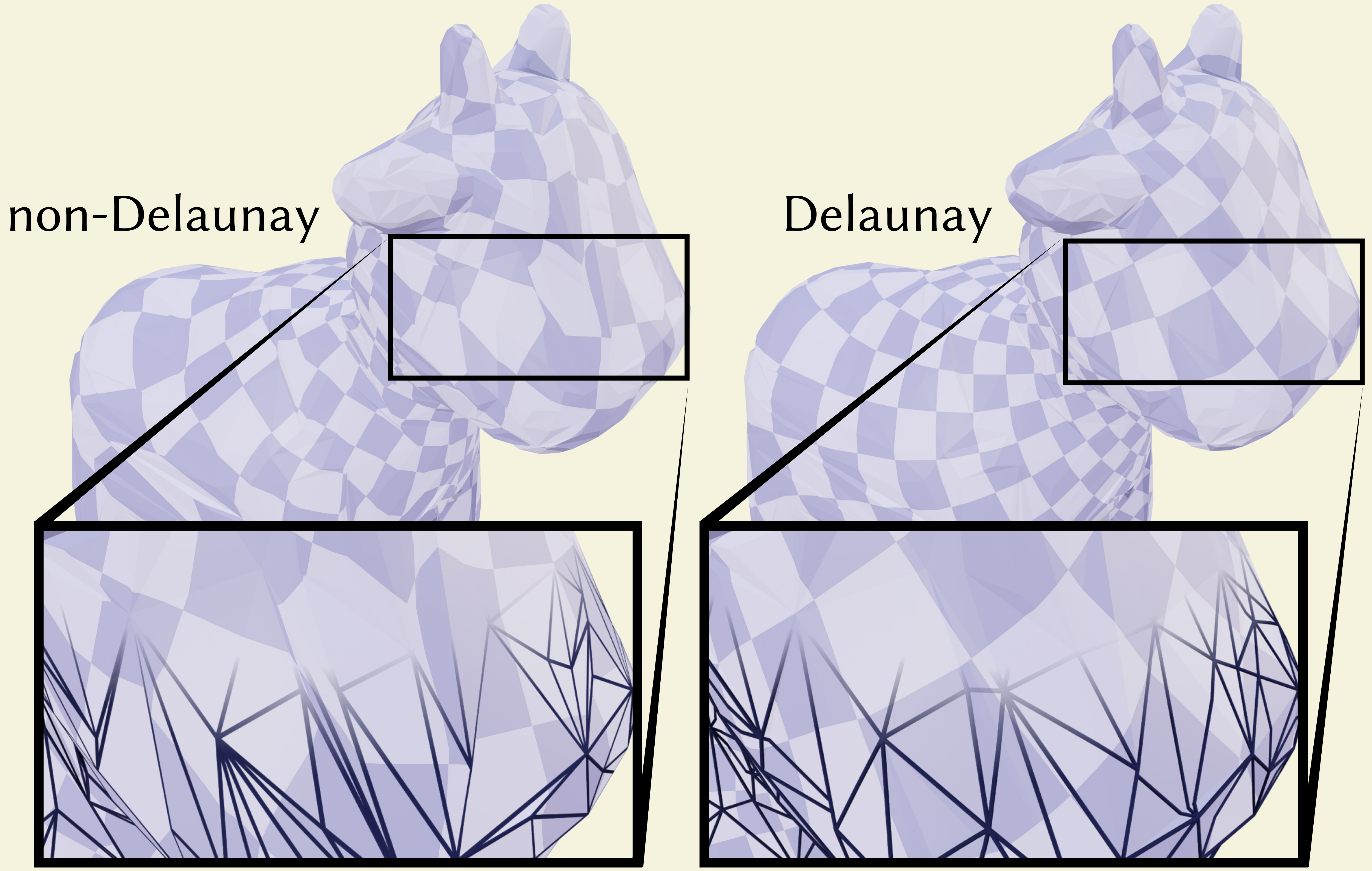


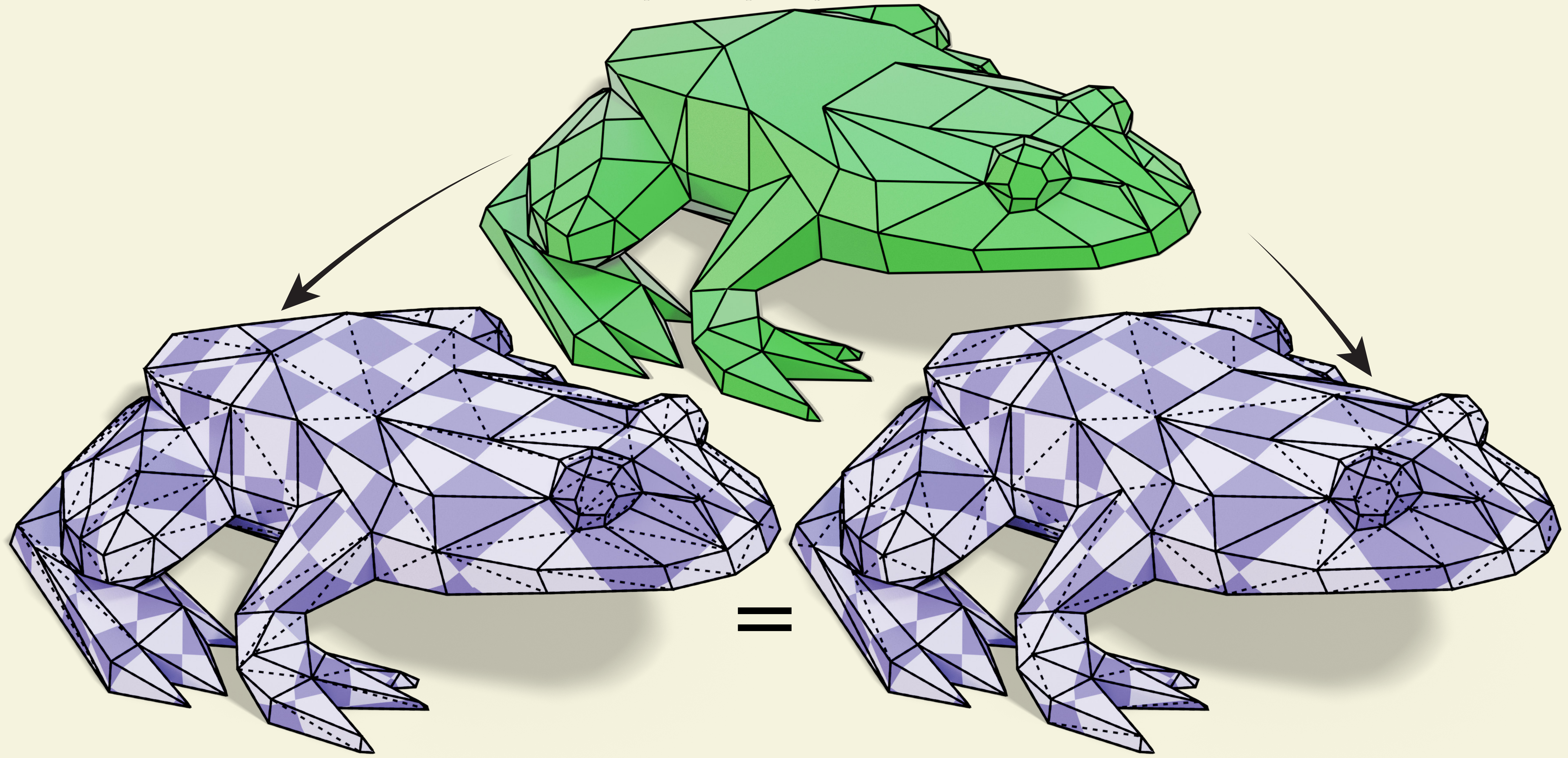
fixed triangulation



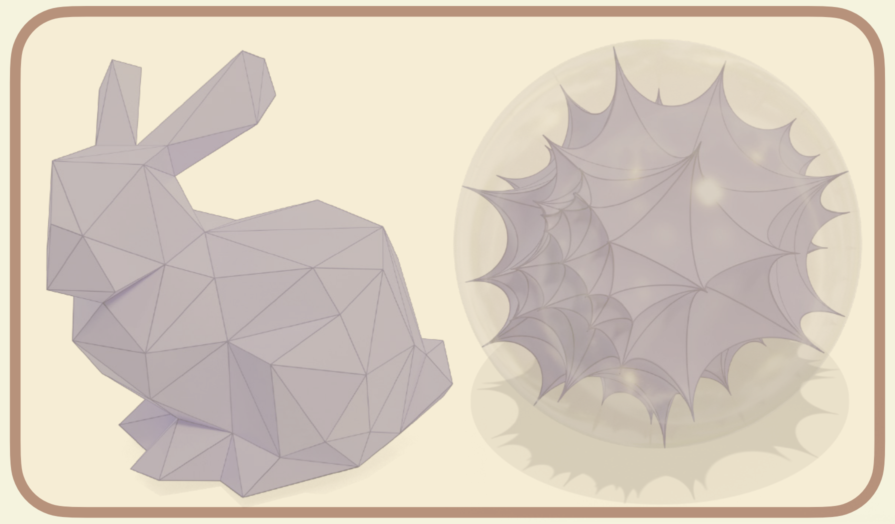
variable triangulation

Starting from Delaunay

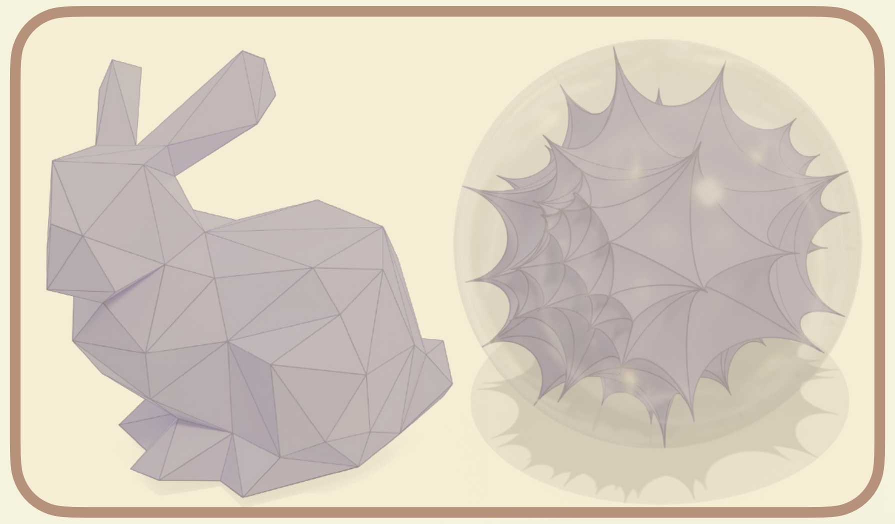


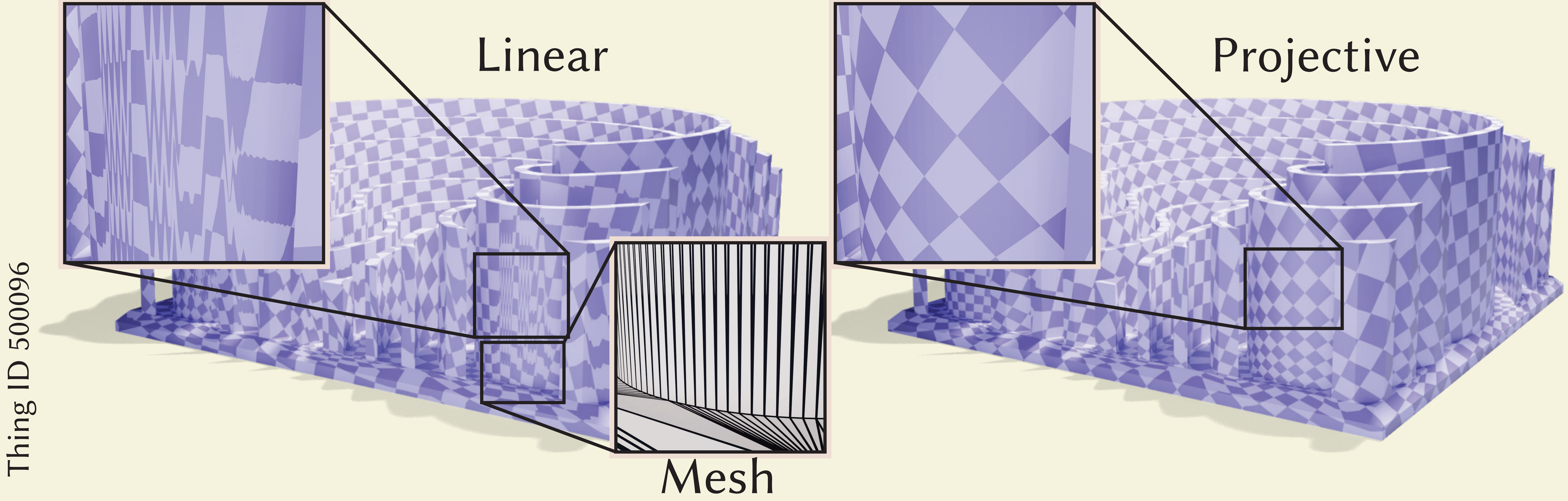


Final algorithm

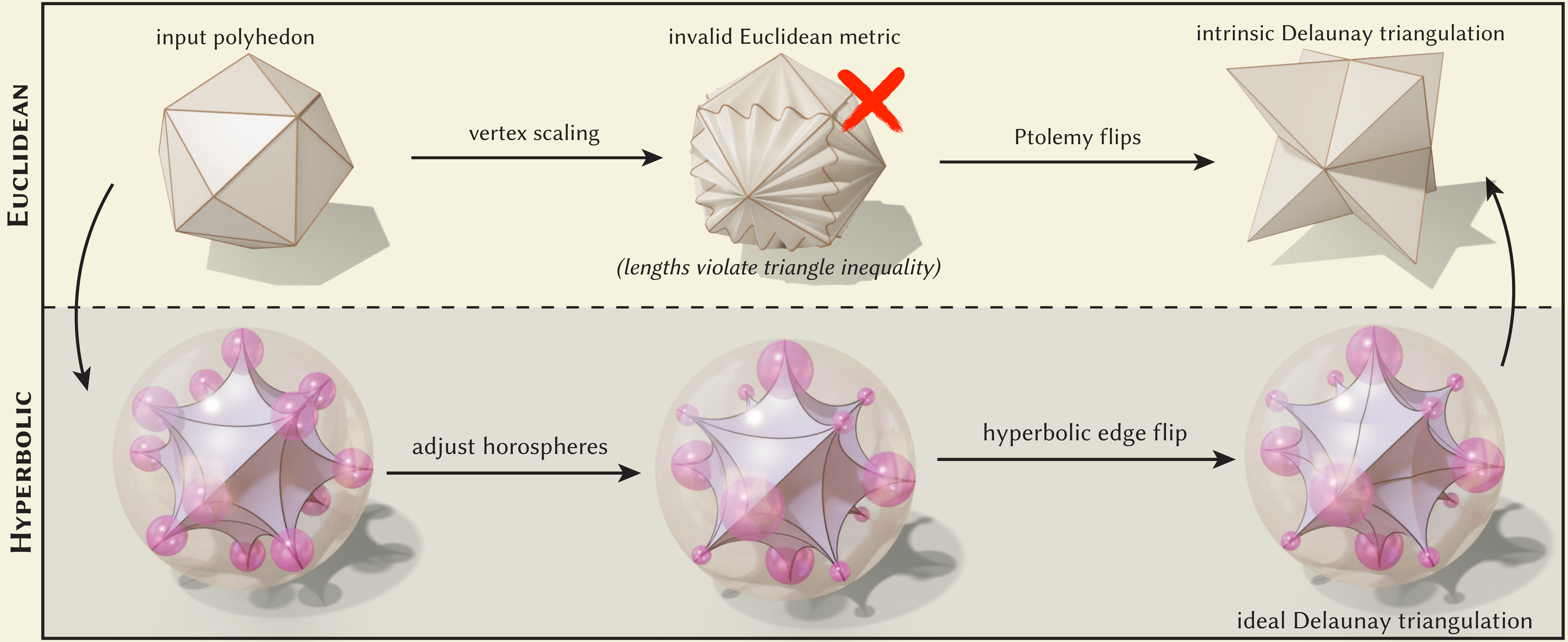


Projective interpolation



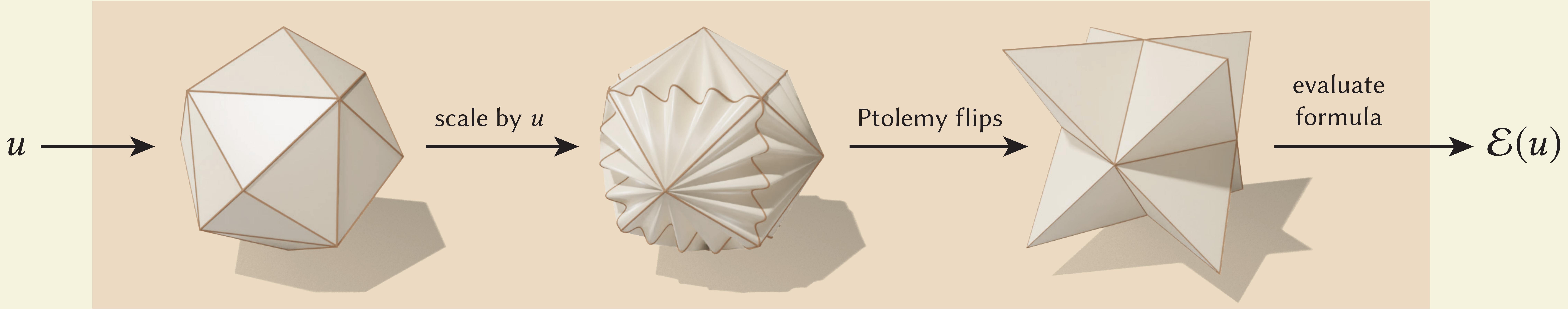


Discrete conformal equivalence across triangulations



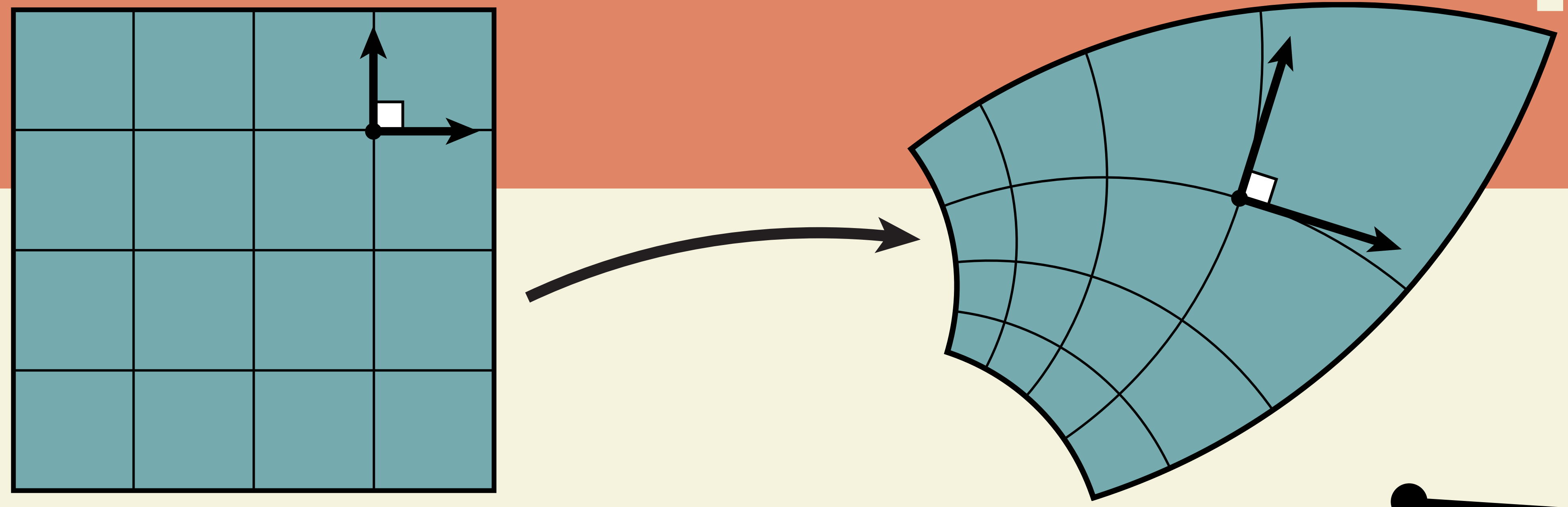
Optimization with Ptolemy Flips

- Express energy and derivatives in terms of edge lengths [Springborn 2019]



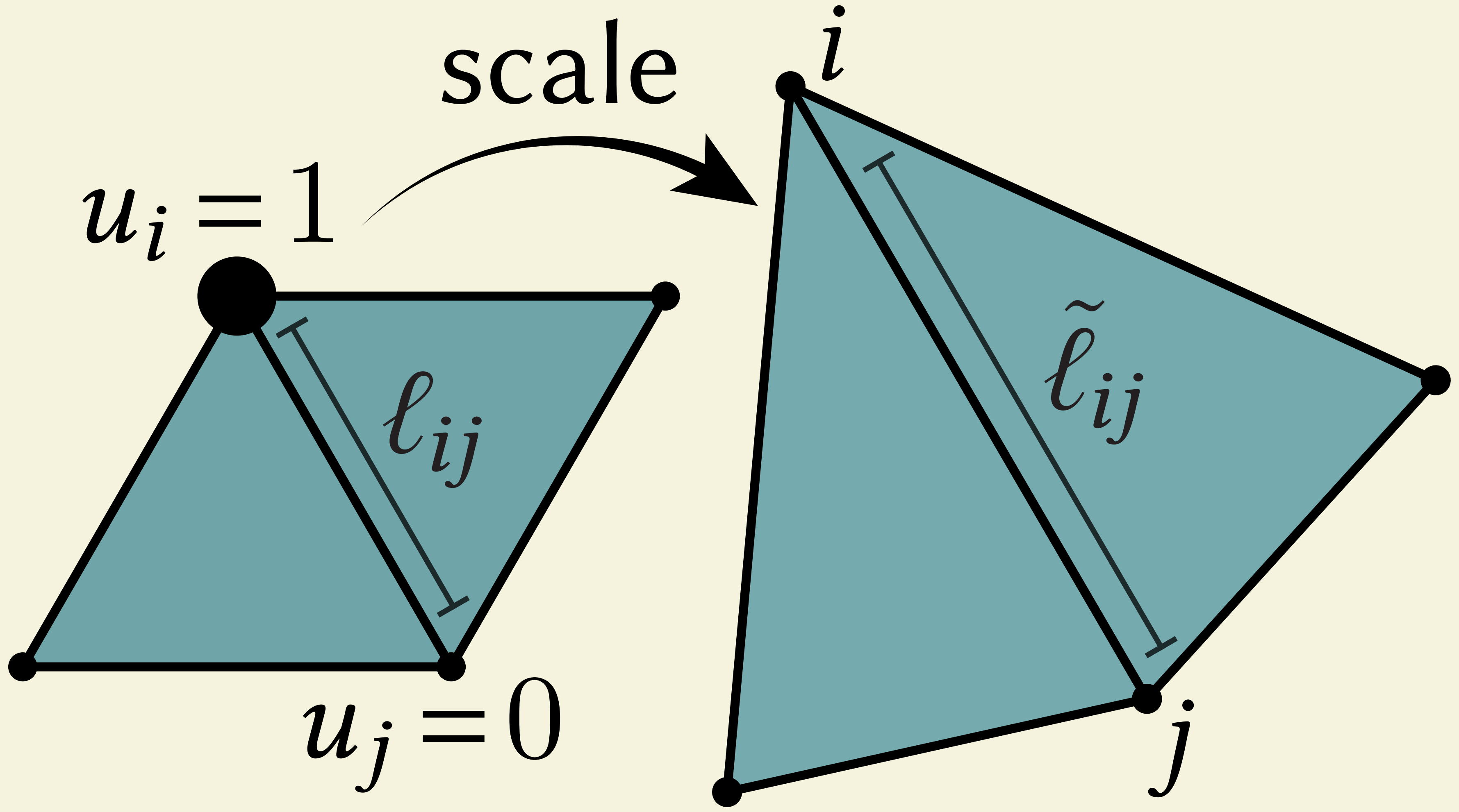
- Hand to any optimization algorithm

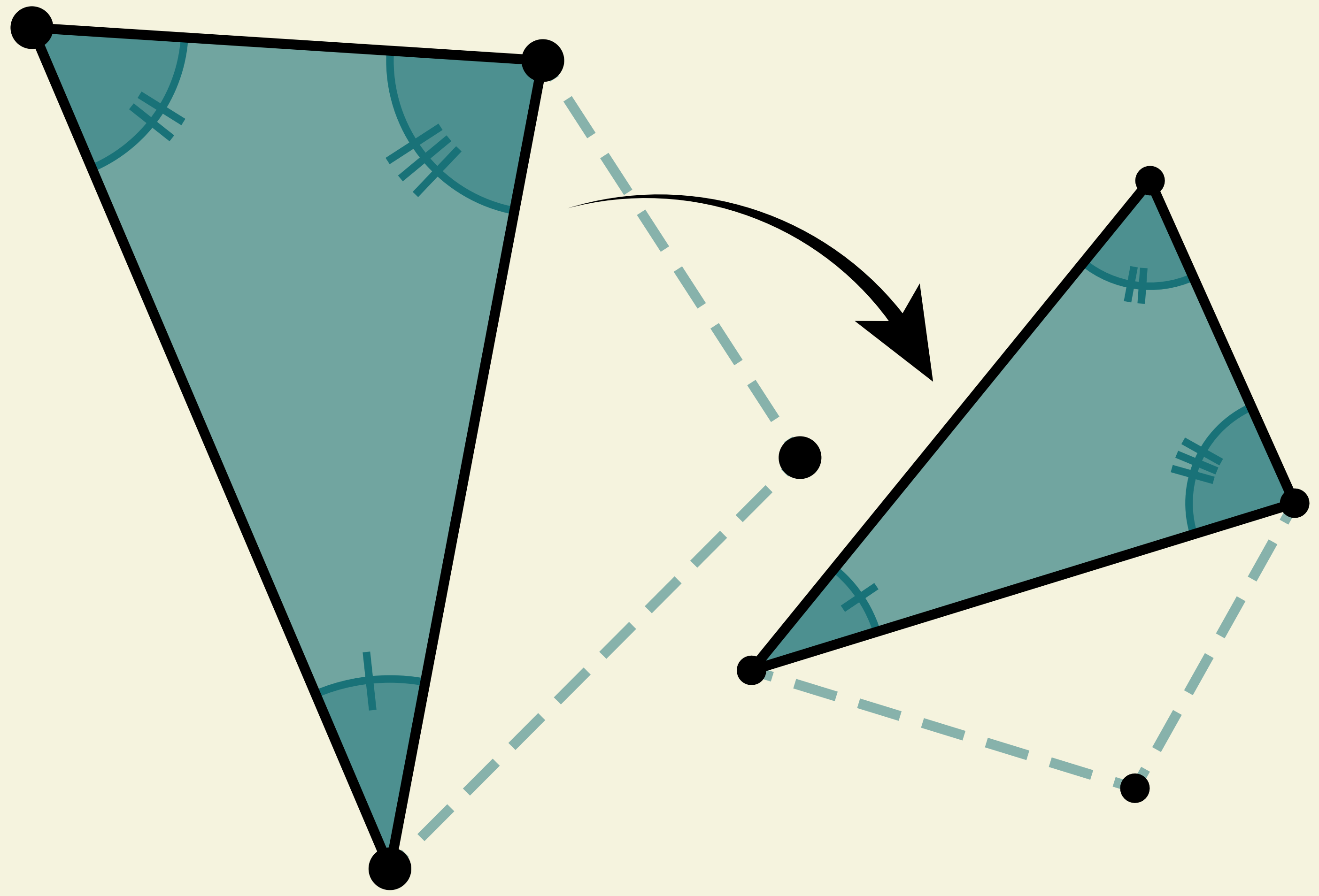
What is a discrete conformal map?



- Why not preserve angles?
  - Too strict
- Positive metric scaling
  - $\tilde{g}_p = e^{2u(p)} g_p$
- Vertex scaling
  - $u : V \to \mathbb{R}$
  - $\tilde{\ell}_{ij} = e^{(u_i+u_j)/2} \ell_{ij}$
  - Just flexible enough

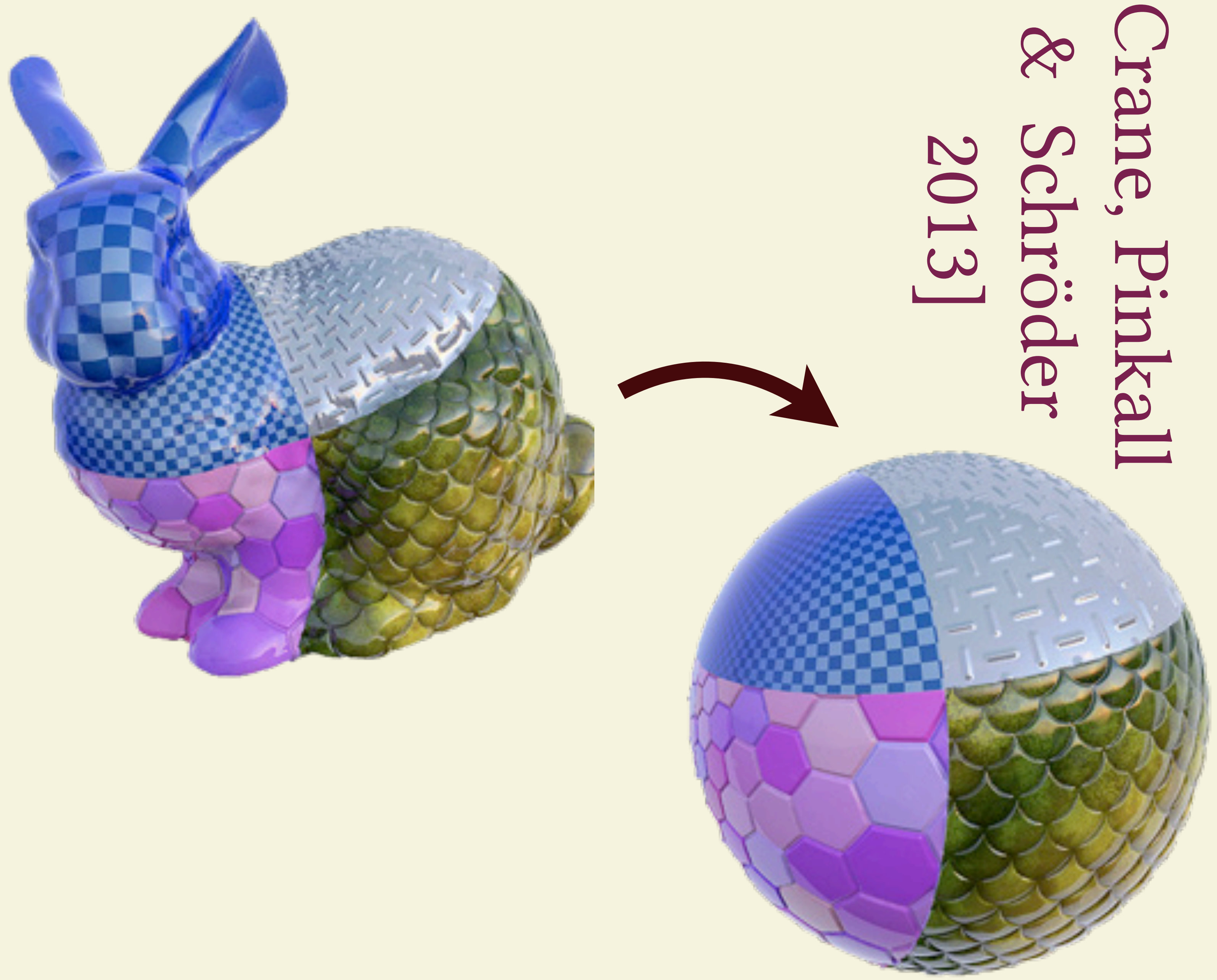
[Bobenko, Pinkall & Springborn 2011]



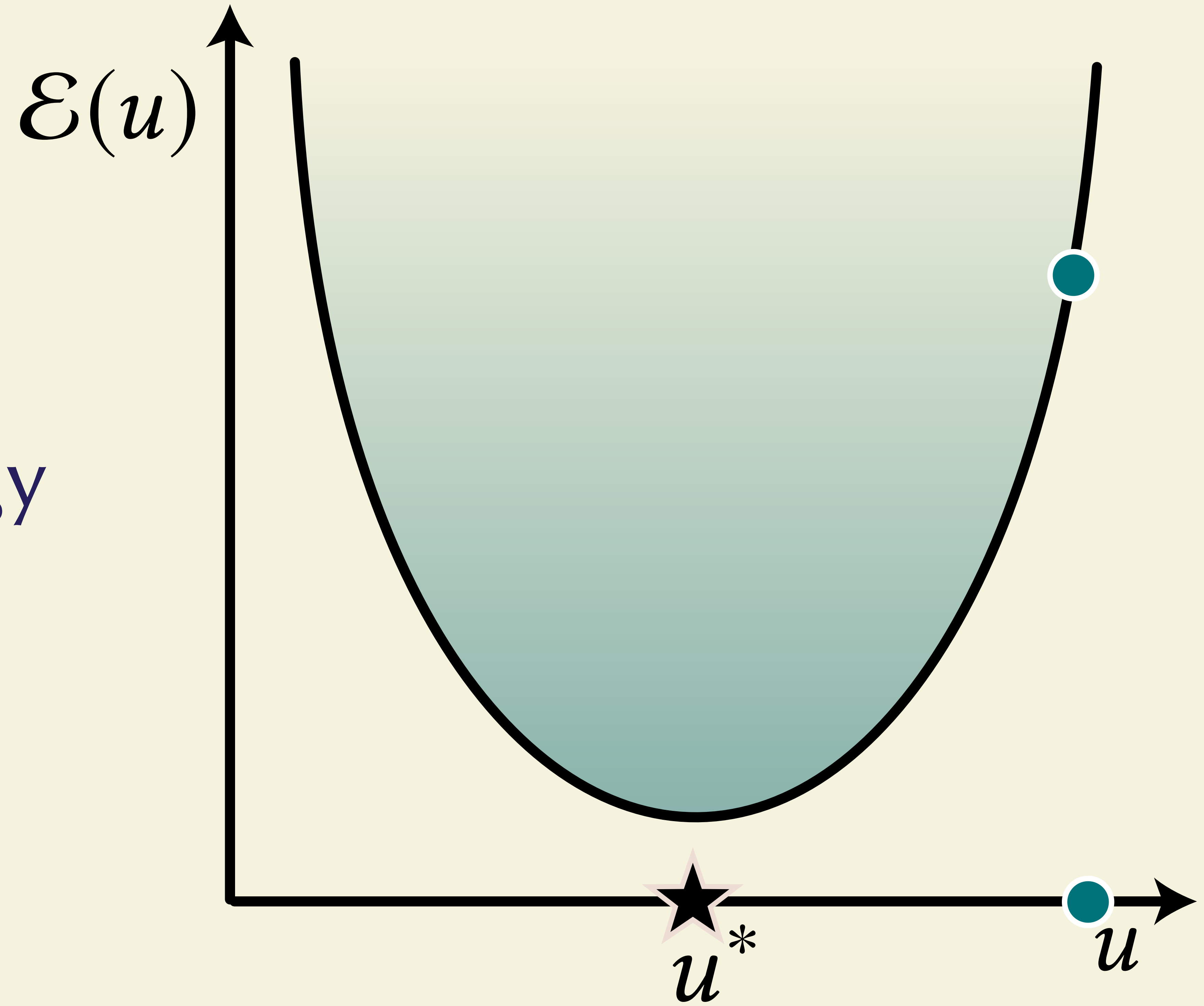


Discrete Uniformization

- Discrete uniformization [Gu, Luo, Sun & Wu 2018; Springborn 2019]
    - *Essentially any vertex curvatures satisfying Gauss-Bonnet can be realized by some vertex scaling*
- [Luo 2004]: follow flow
- [Springborn, Schröder & Pinkall 2008]: minimize energy

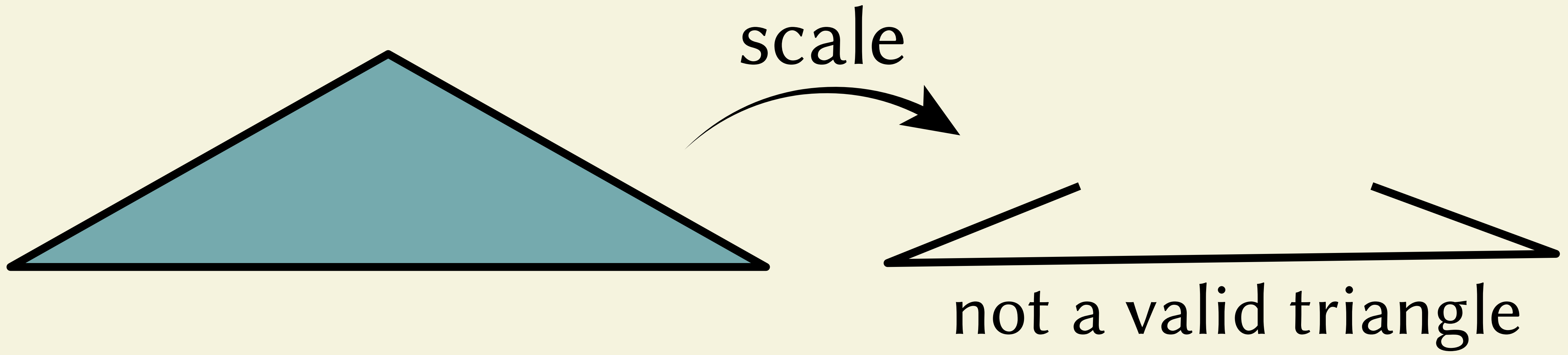


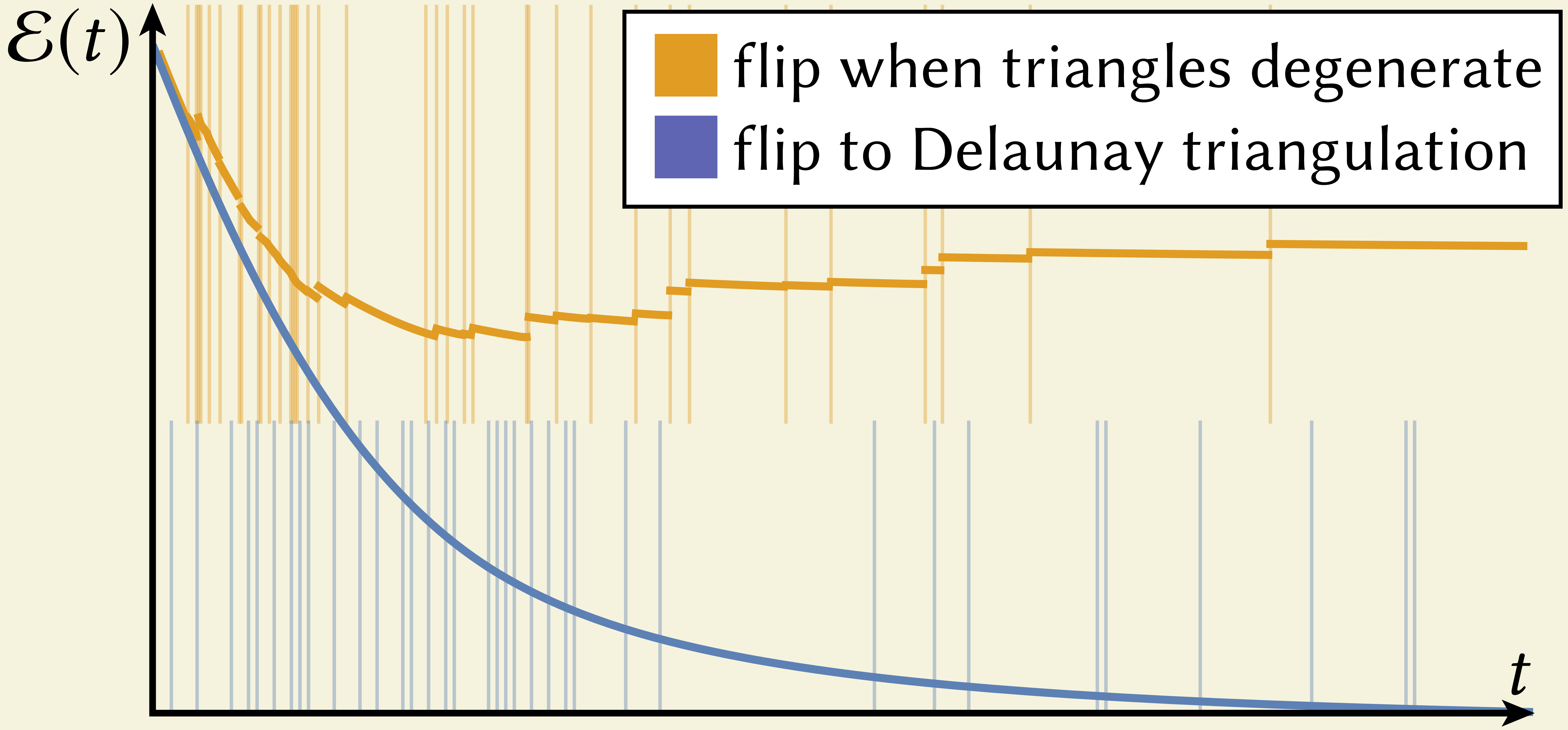
[Crane, Pinkall & Schröder 2013]



Challenges with discrete uniformization

- **Discrete uniformization doesn't always work on a fixed mesh because triangles can degenerate**
- Idea: flip edges when triangles break
  - Problem: energy discontinuous at flips (vertical lines)
- [Gu, Luo, Sun & Wu 2018]: maintain Delaunay
  - Problem: stop to flip





Riemann surfaces

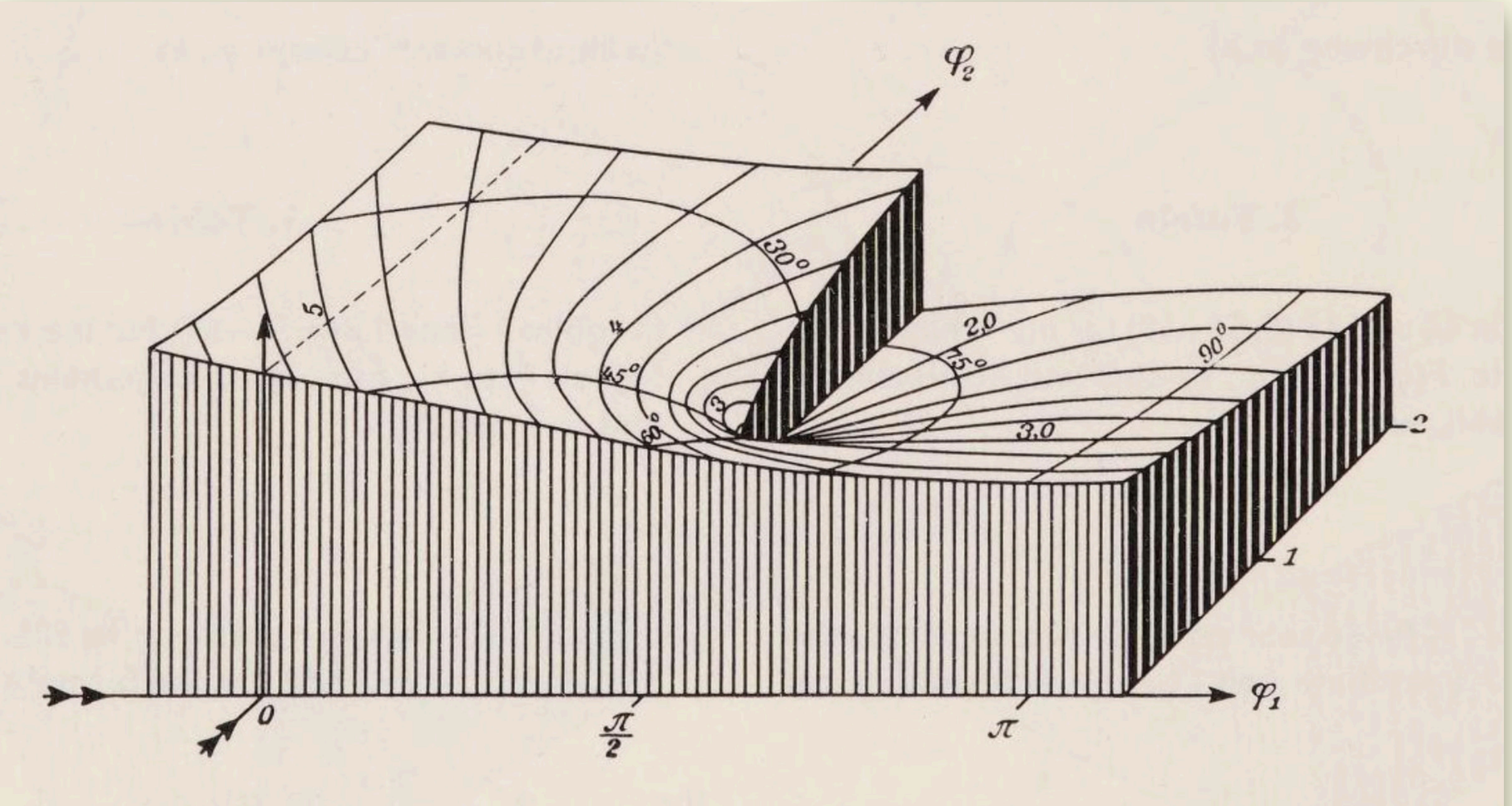
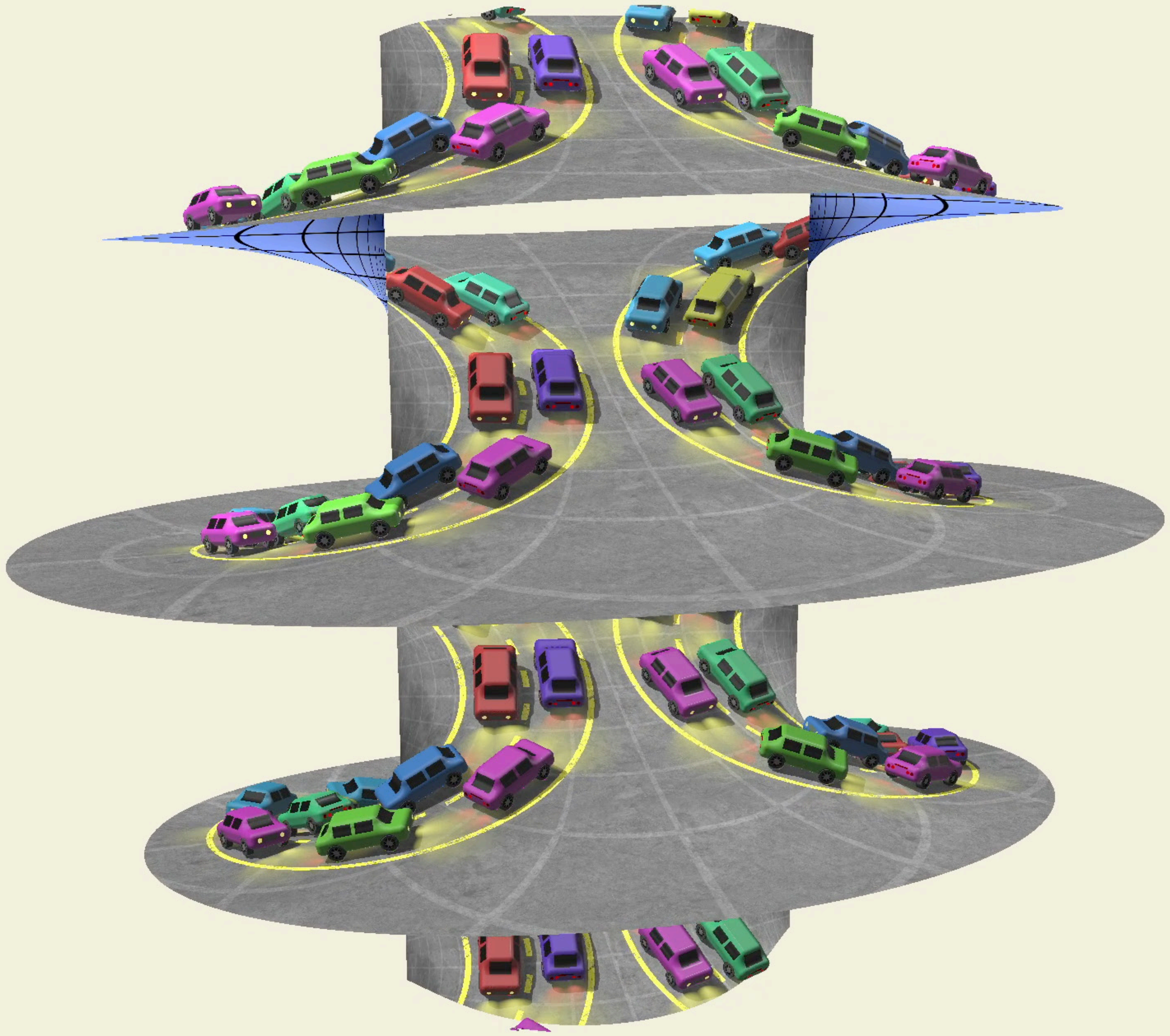


Fig. 24 Relief des 2. Zweiges der Funktion $F(\varphi, k)$ mit $k = 0{,}8$. $(\varphi = \varphi_1 + i\varphi_2)$

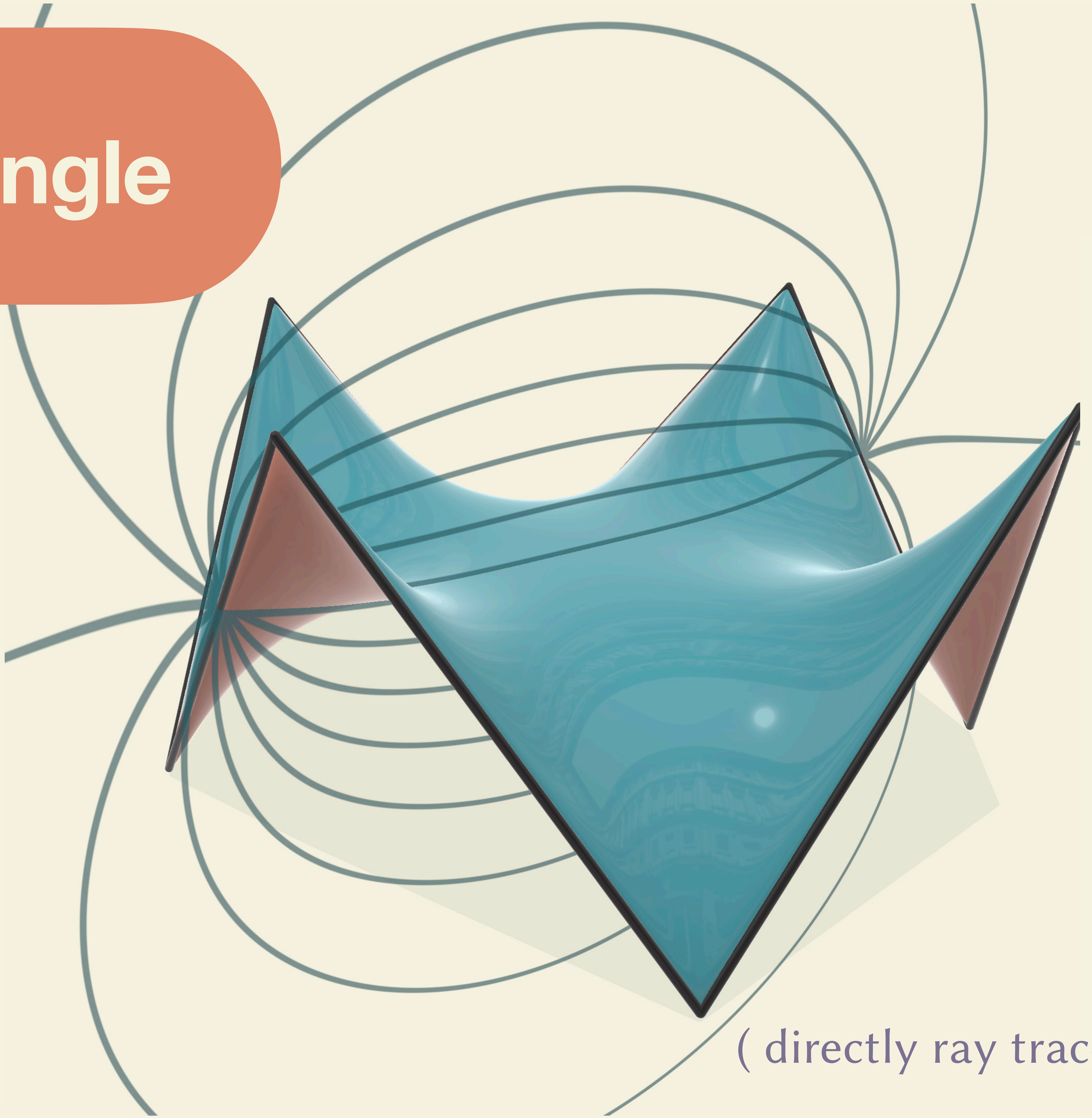
Fig. 24 Relief of the 2 nd branch of the function $F(\varphi, k)$ with $k = 0{,}8$. $(\varphi = \varphi_1 + i\varphi_2)$

[Jahnke, Emde & Lösch 1960]



(directly ray traced)

Signed solid angle



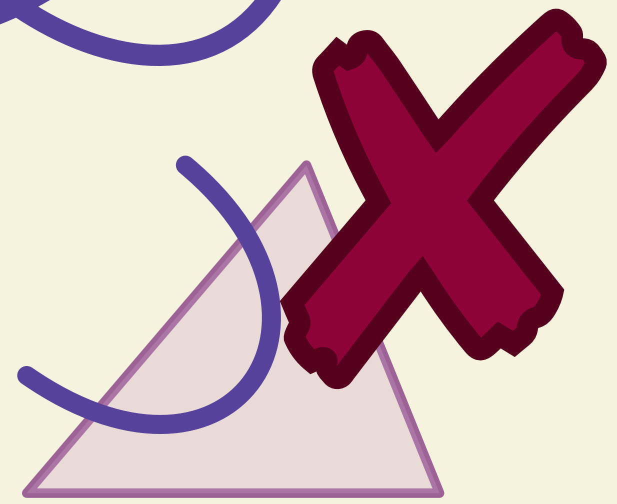
(directly ray traced)

Encoding a curve with normal coordinates

- Just count intersections

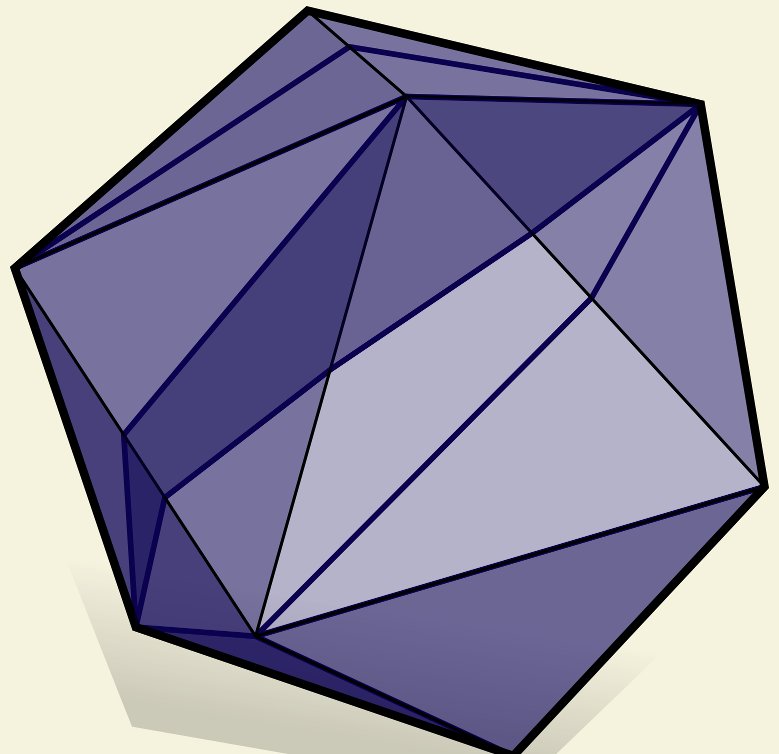
Rules

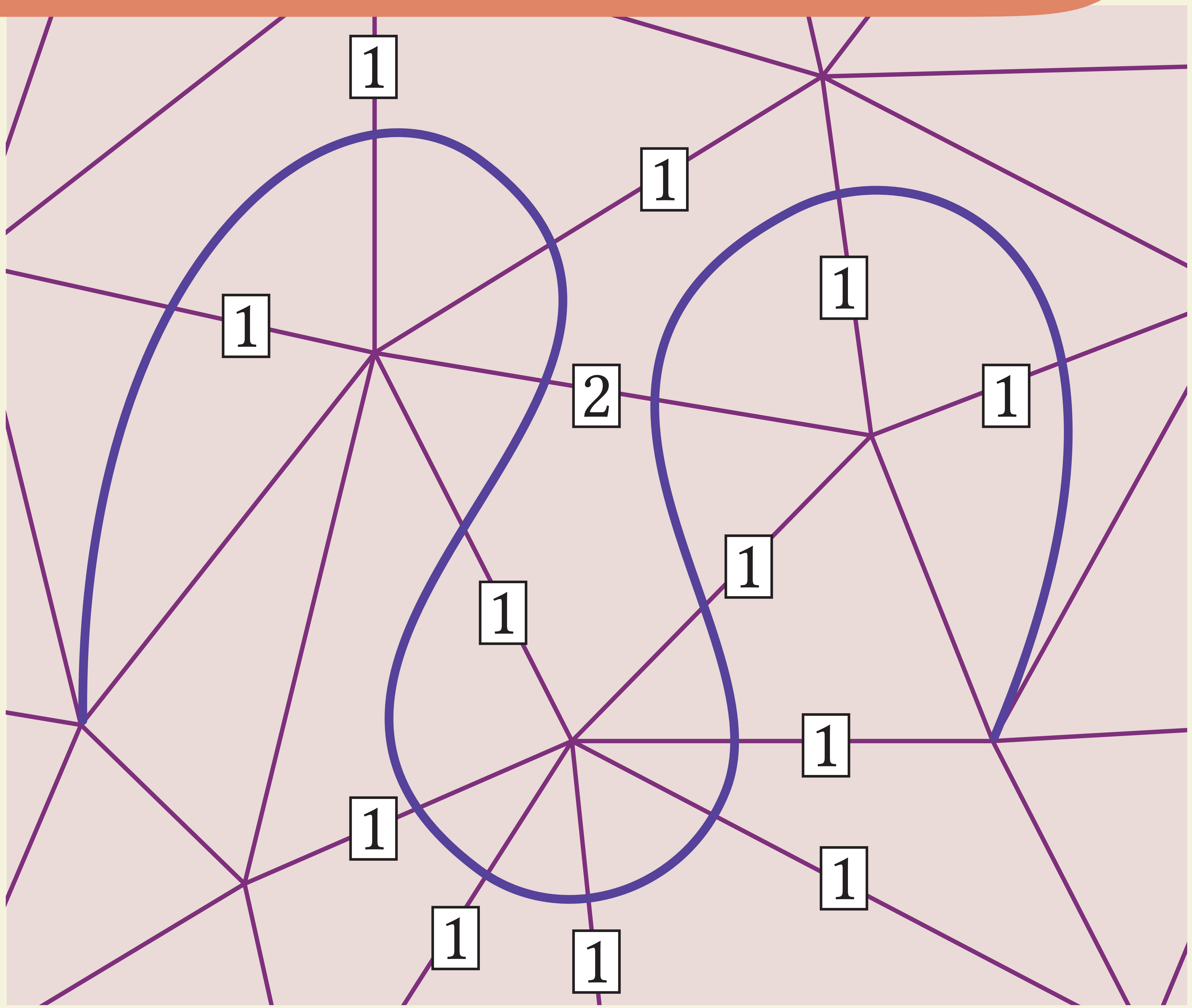
1. No self-crossings
2. No U-turns



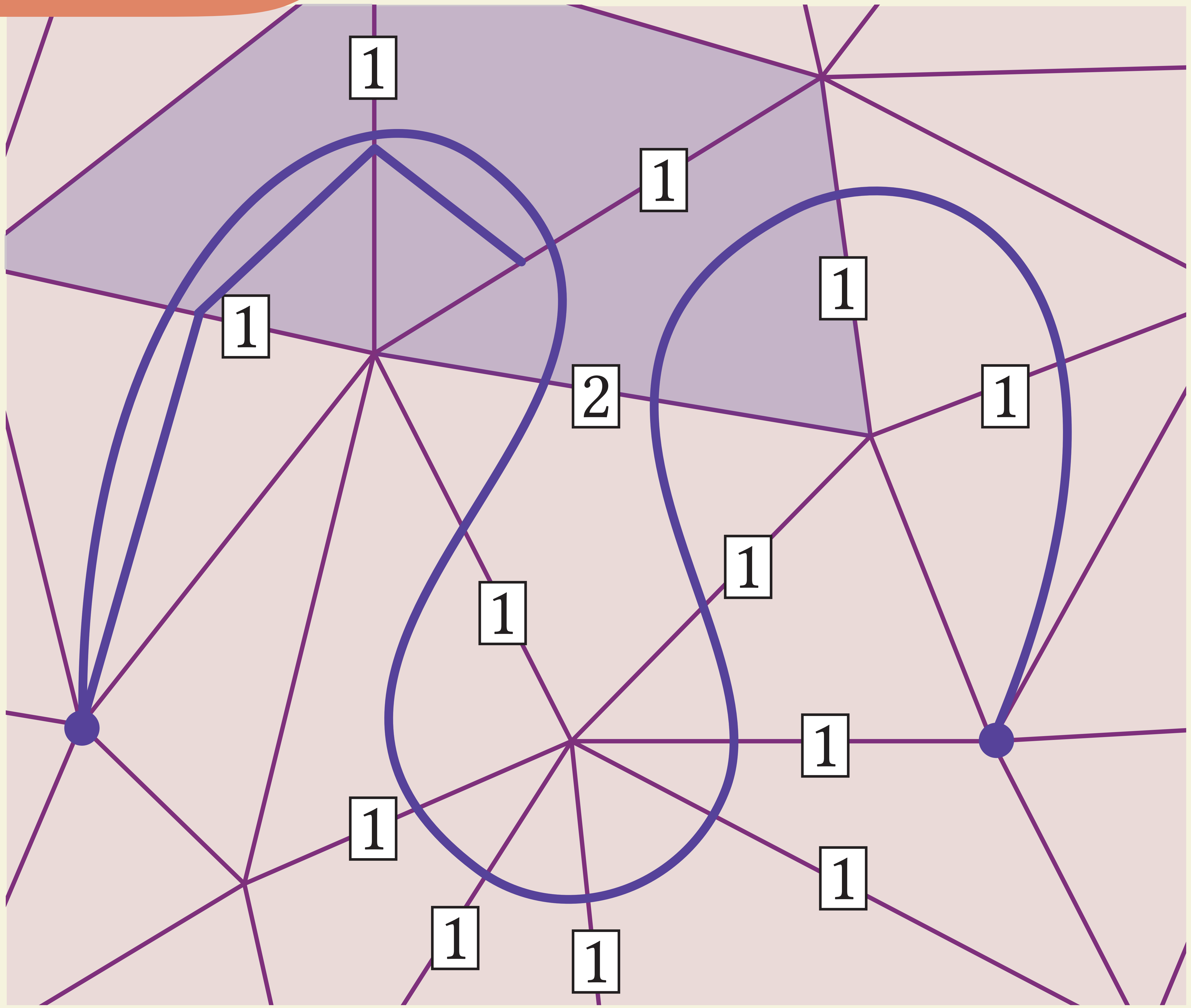
(also curves may only start or end at vertices of the triangulation)

automatically satisfied for our triangulations

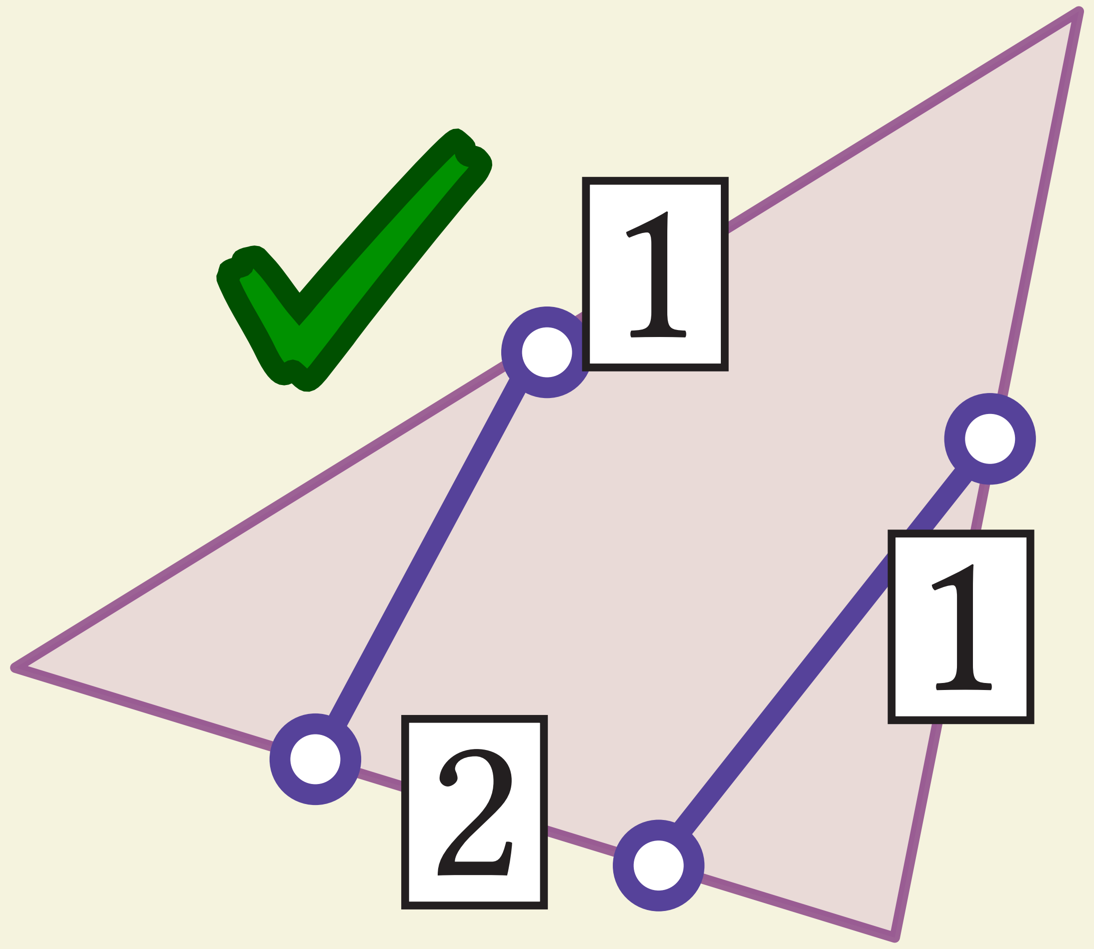


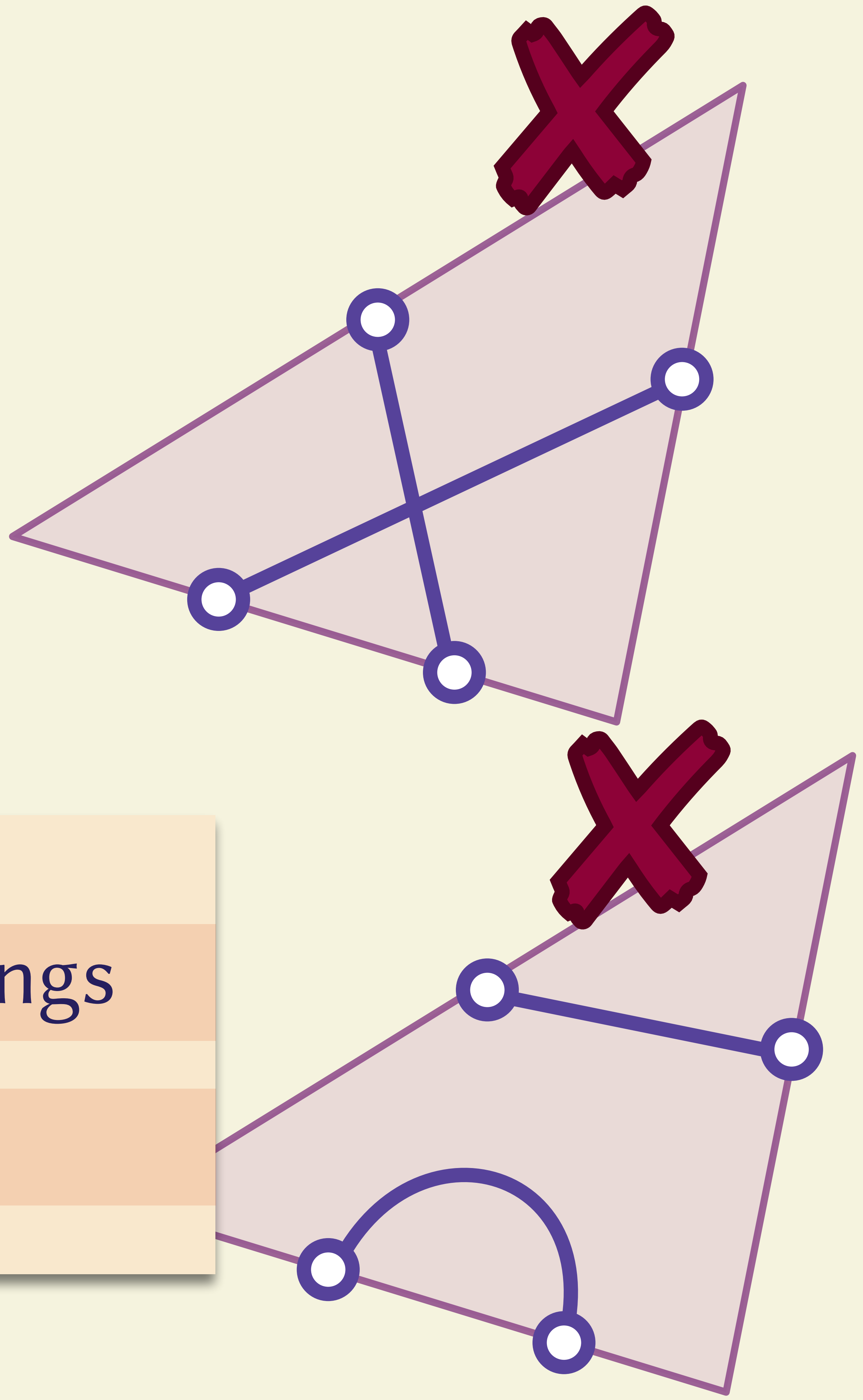


Reconstructing the curve



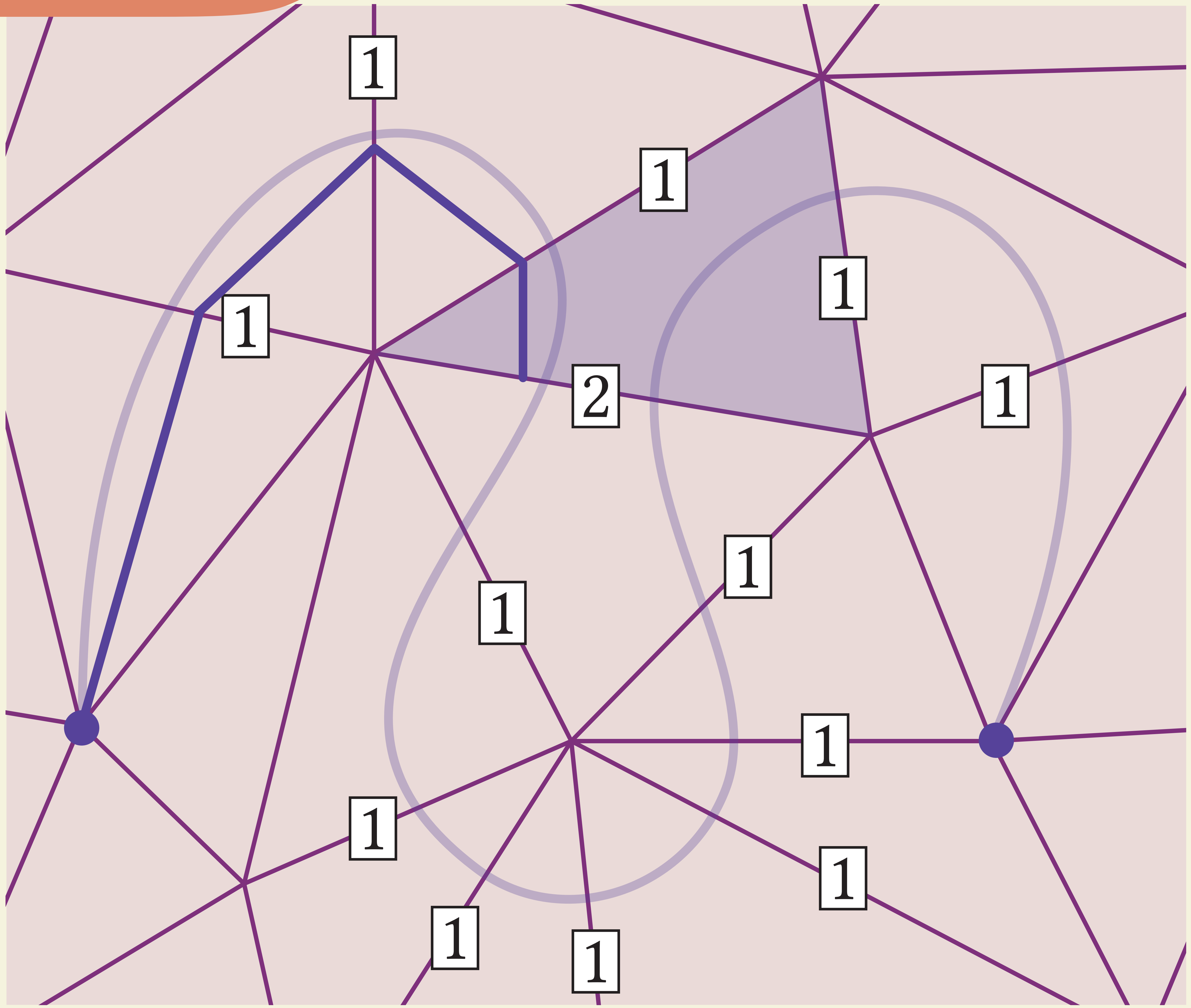
Reconstructing the curve



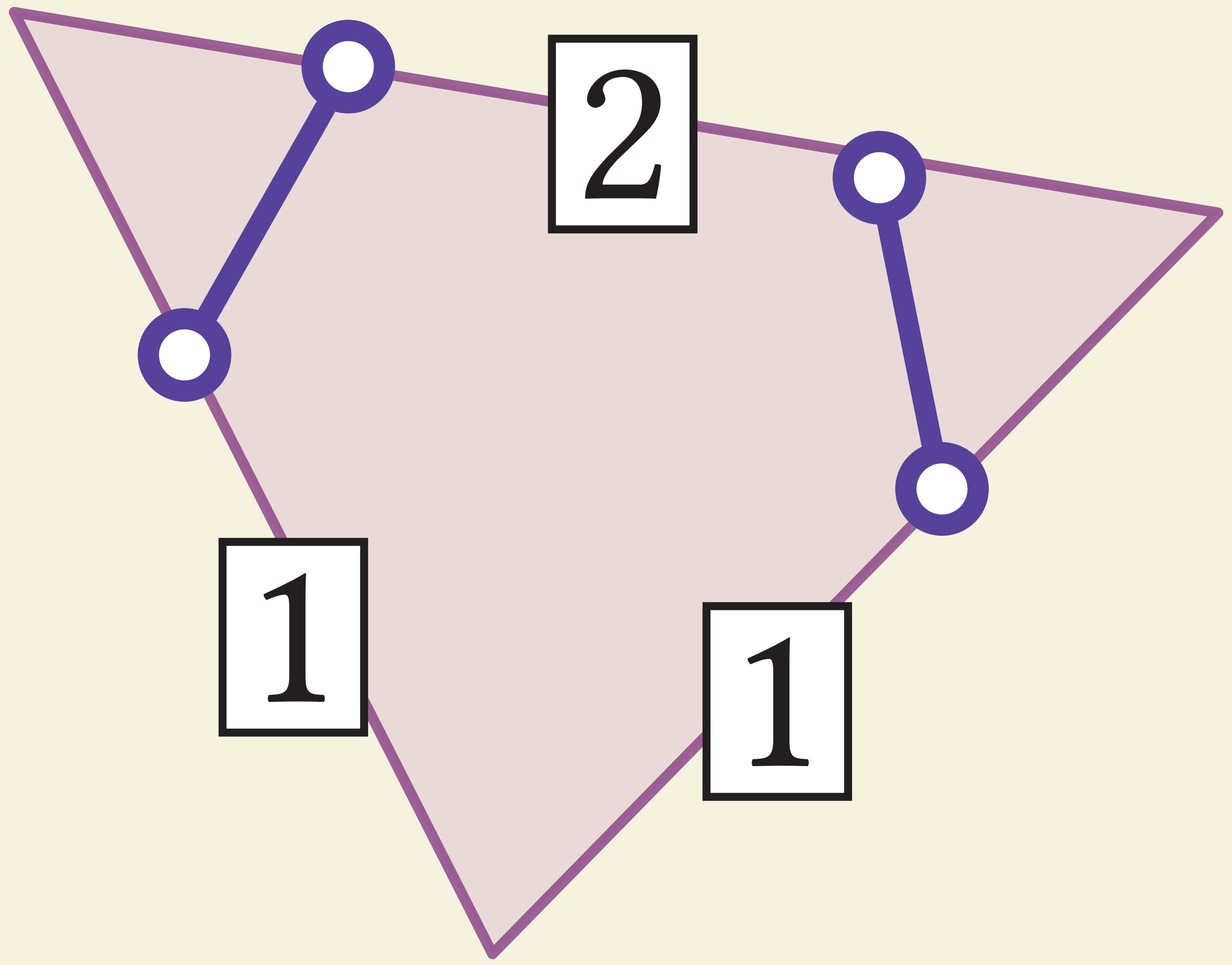


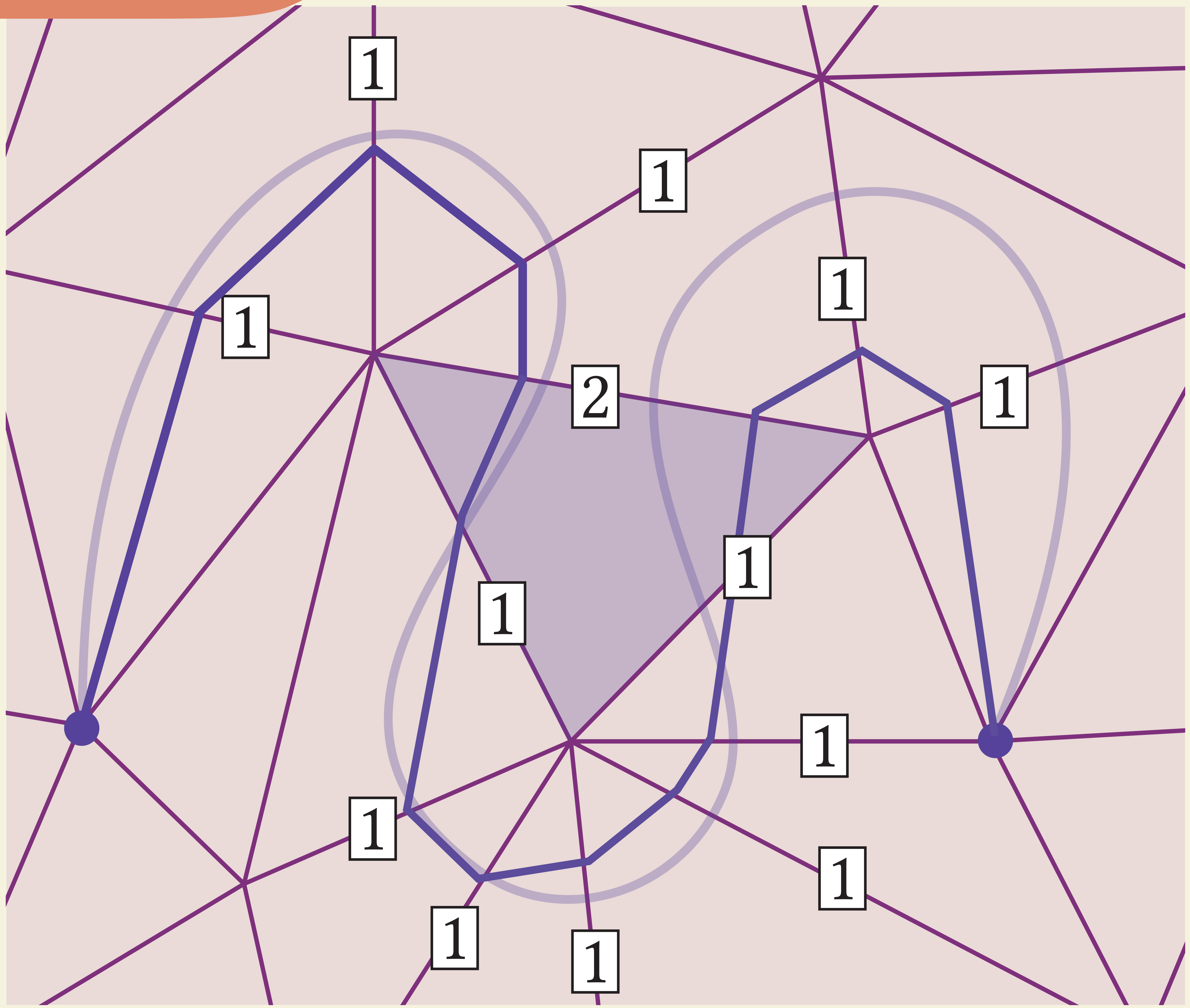
Rules

1. No self-crossings
2. No U-turns

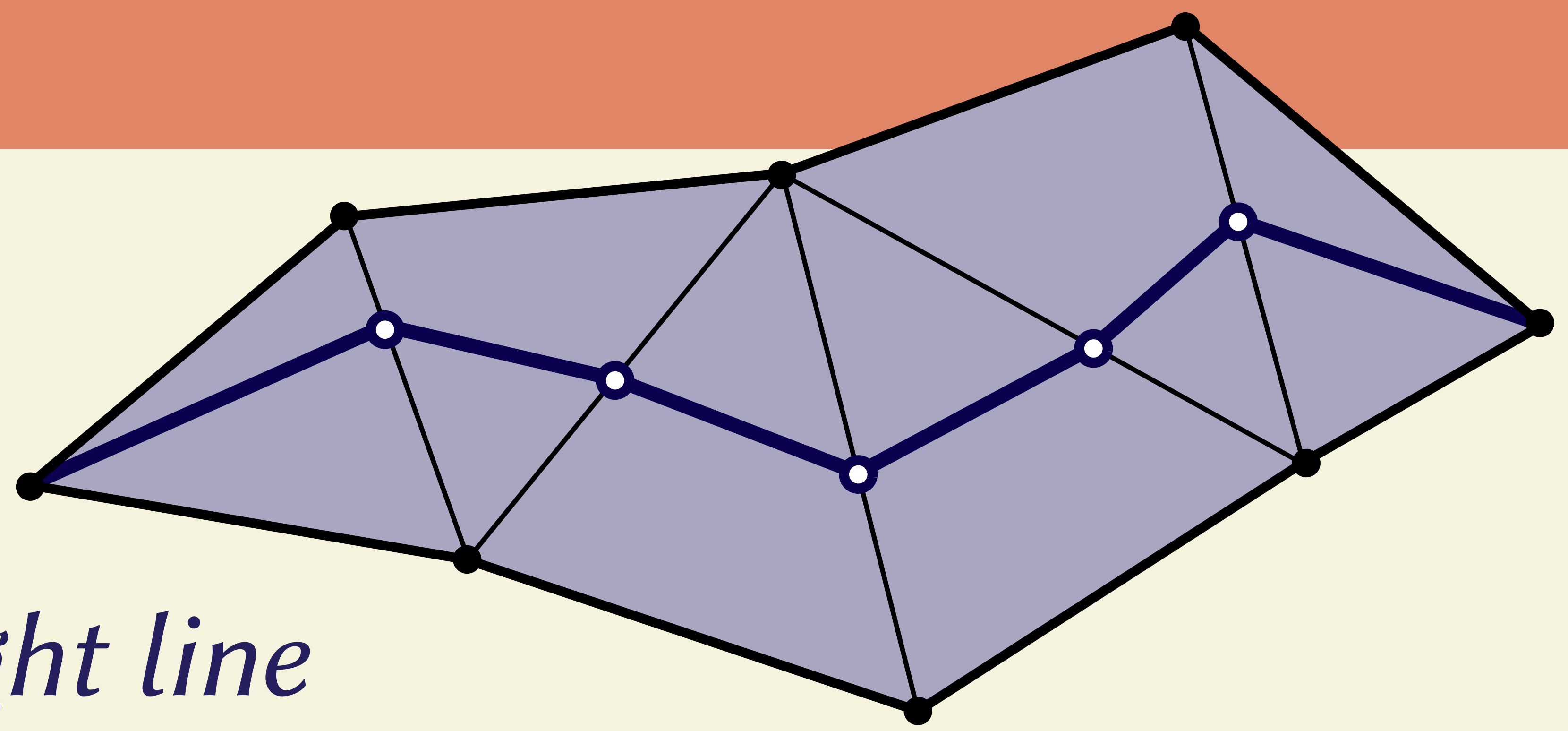


Reconstructing the curve

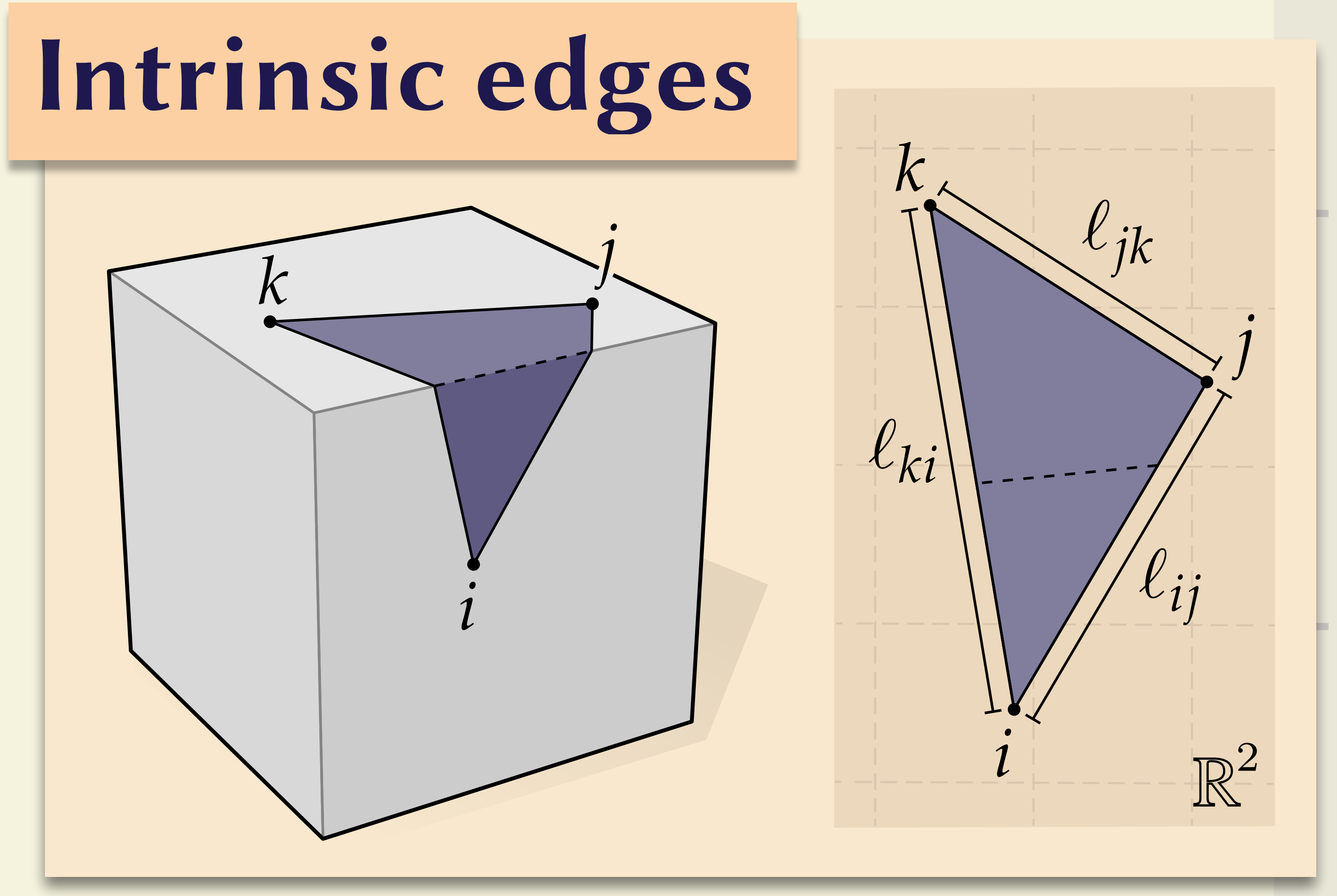


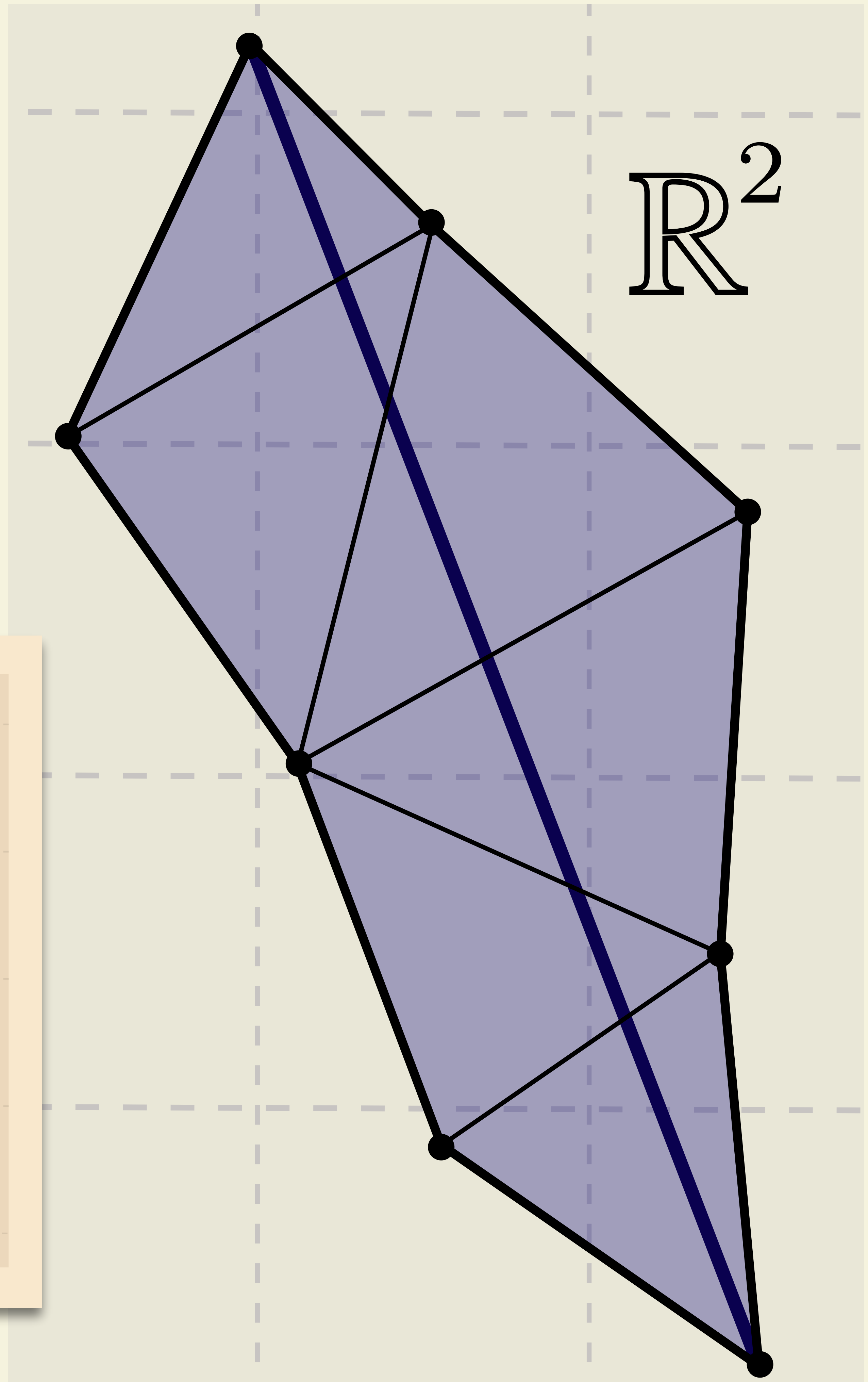


Finding the exact curve geometry



- So far: sequence of triangles
- True curve unfolds to a *straight line*
  - Lay out in plane for exact curve
- Normal coordinates determine edges exactly





Collections of Curves

- *e.g.* edges of a triangulation
- Could store multiple sets of normal coordinates
  - Expensive ✗
- Instead, just store one set of normal coordinates

Store just one integer per edge