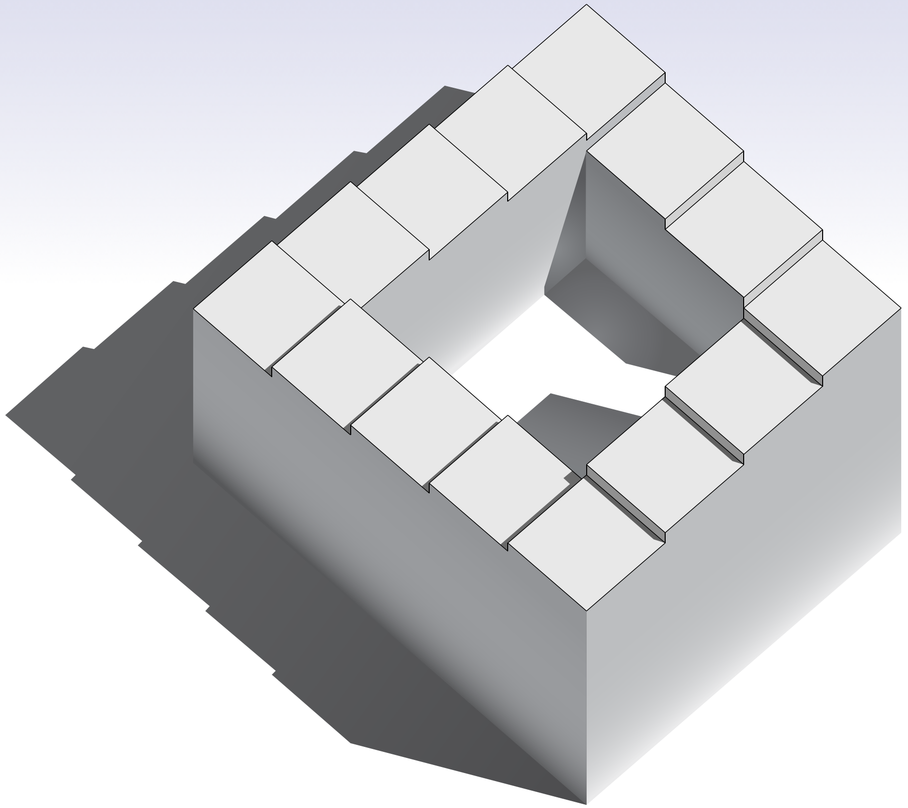


# Discrete Differential Geometry: An Applied Introduction

**Keenan Crane • CMU 15-458/858B • Fall 2017**

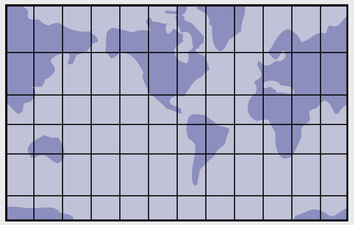
These slides are a work in progress and ***there will be errors***. As always, *your brain* is the best tool for determining whether statements are actually true! Please do not hesitate to stop and ask for clarifications/corrections, or send email to `kmcrane@cs.cmu.edu`.

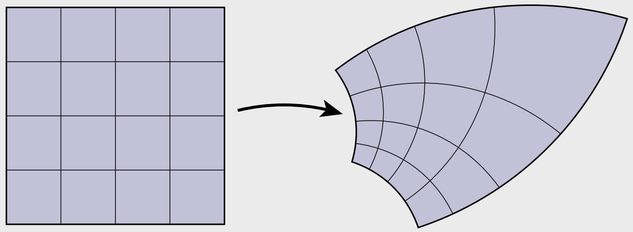
# Lecture 12:
# Conformal Geometry

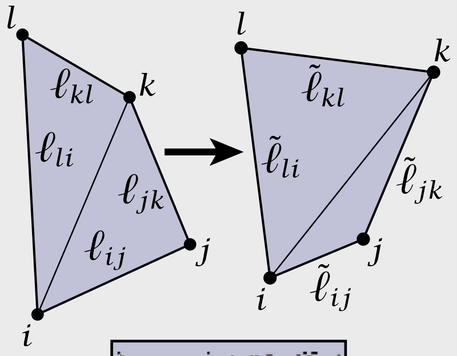
## Discrete Differential Geometry:
## An Applied Introduction

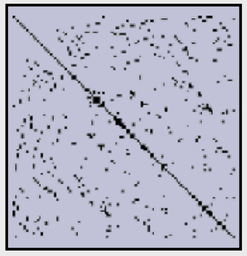
**Keenan Crane • CMU 15-458/858B • Fall 2017**

# *Outline*







**PART I: OVERVIEW**

---

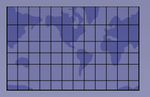
**PART II: SMOOTH THEORY**

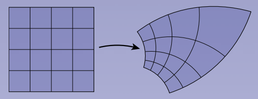
---

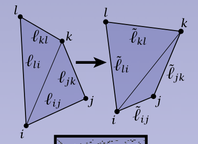
**PART III: DISCRETIZATION**

---

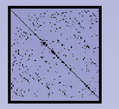
**PART IV: ALGORITHMS**


PART I: OVERVIEW

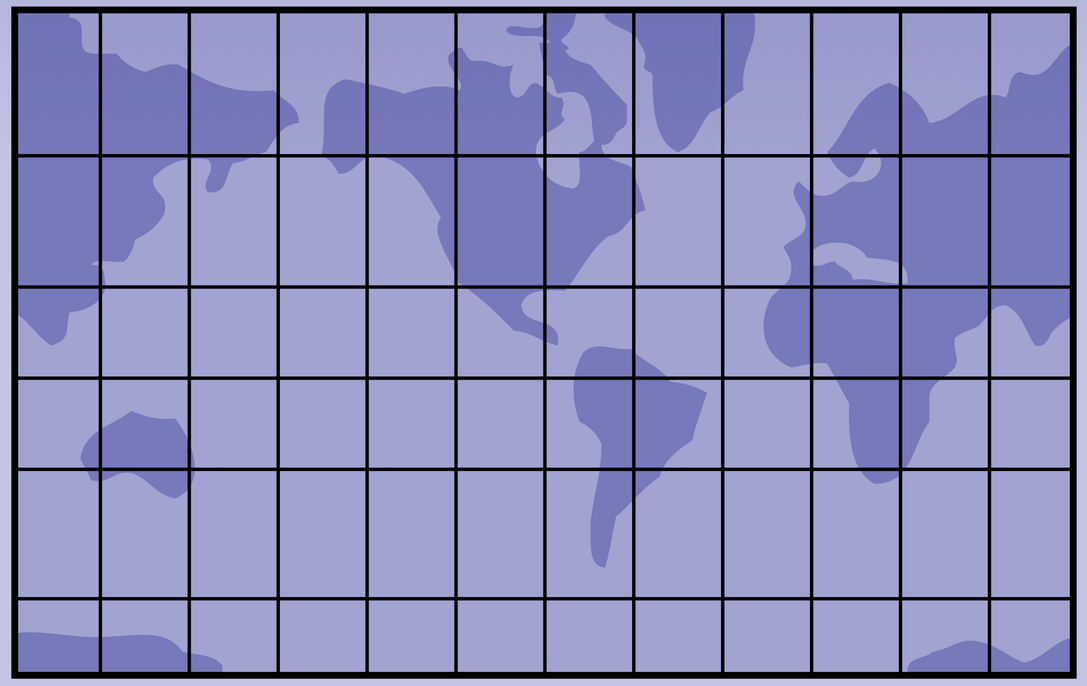

PART II: SMOOTH THEORY



PART III: DISCRETIZATION


PART IV: ALGORITHMS


# PART I: OVERVIEW



# CONFORMAL GEOMETRY PROCESSING


**Keenan Crane • CMU 15-458/858B • Fall 2017**

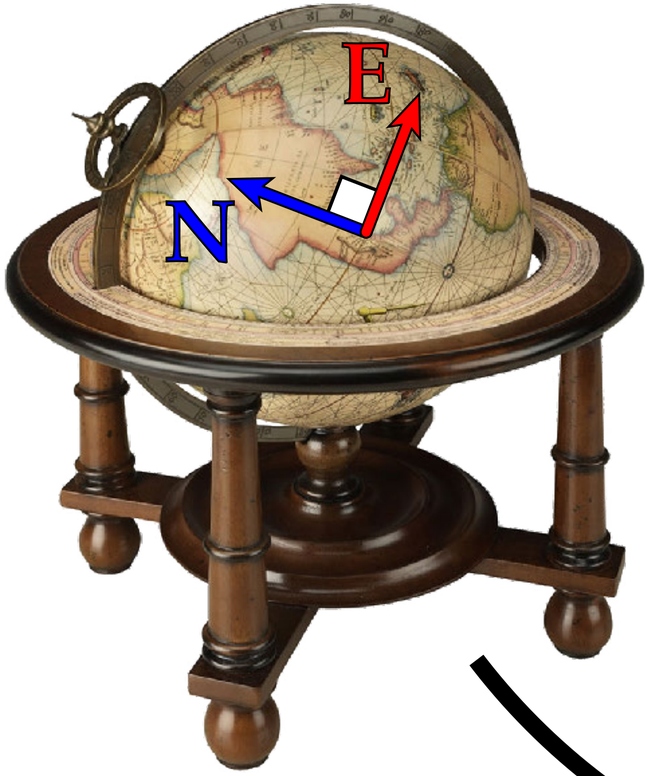
# *Motivation: Mapmaking Problem*

- How do you make a flat map of the round globe?
- Hard to do!  Like trying to flatten an orange peel…

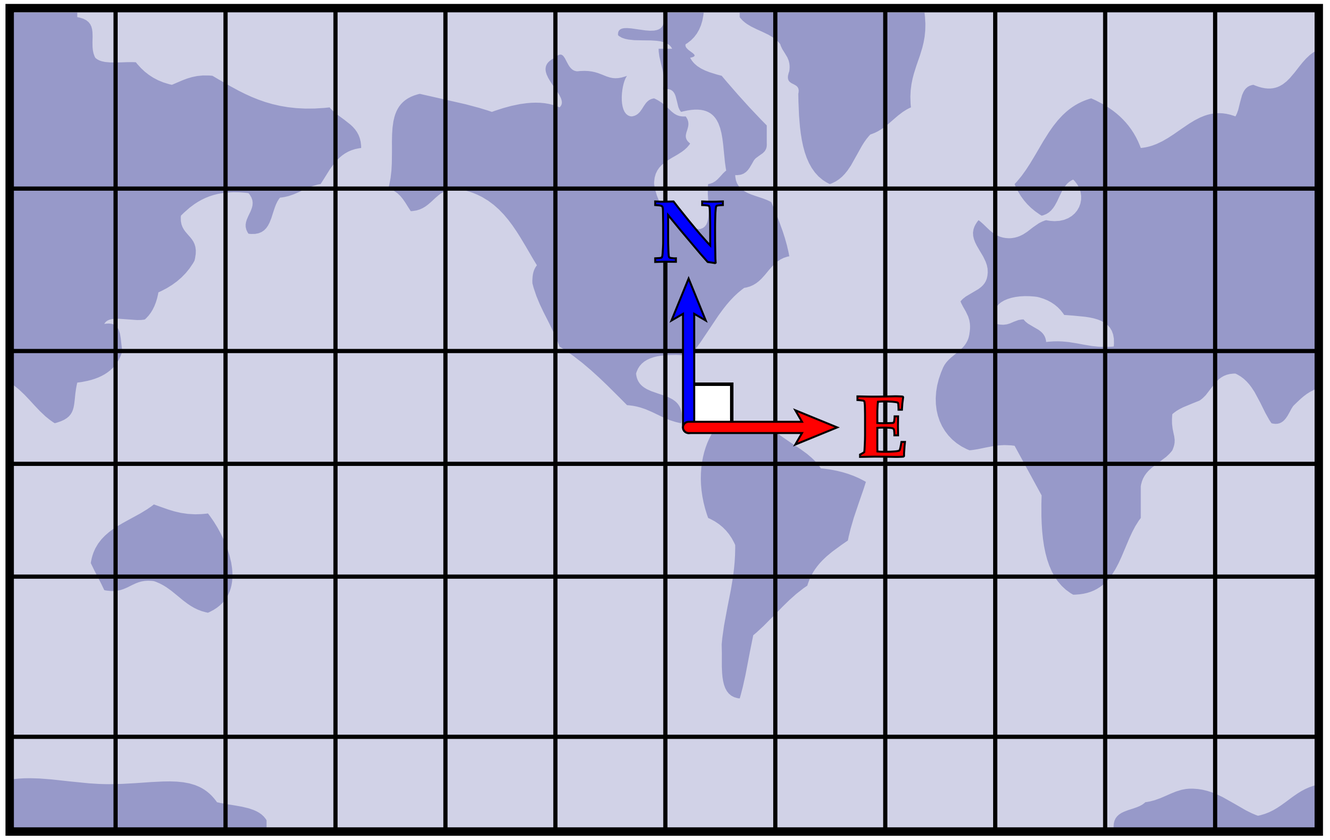
*Impossible* without some kind of distortion and/or cutting.

# *Conformal Mapmaking*

- Amazing fact: can always make a map that exactly preserves **angles**.



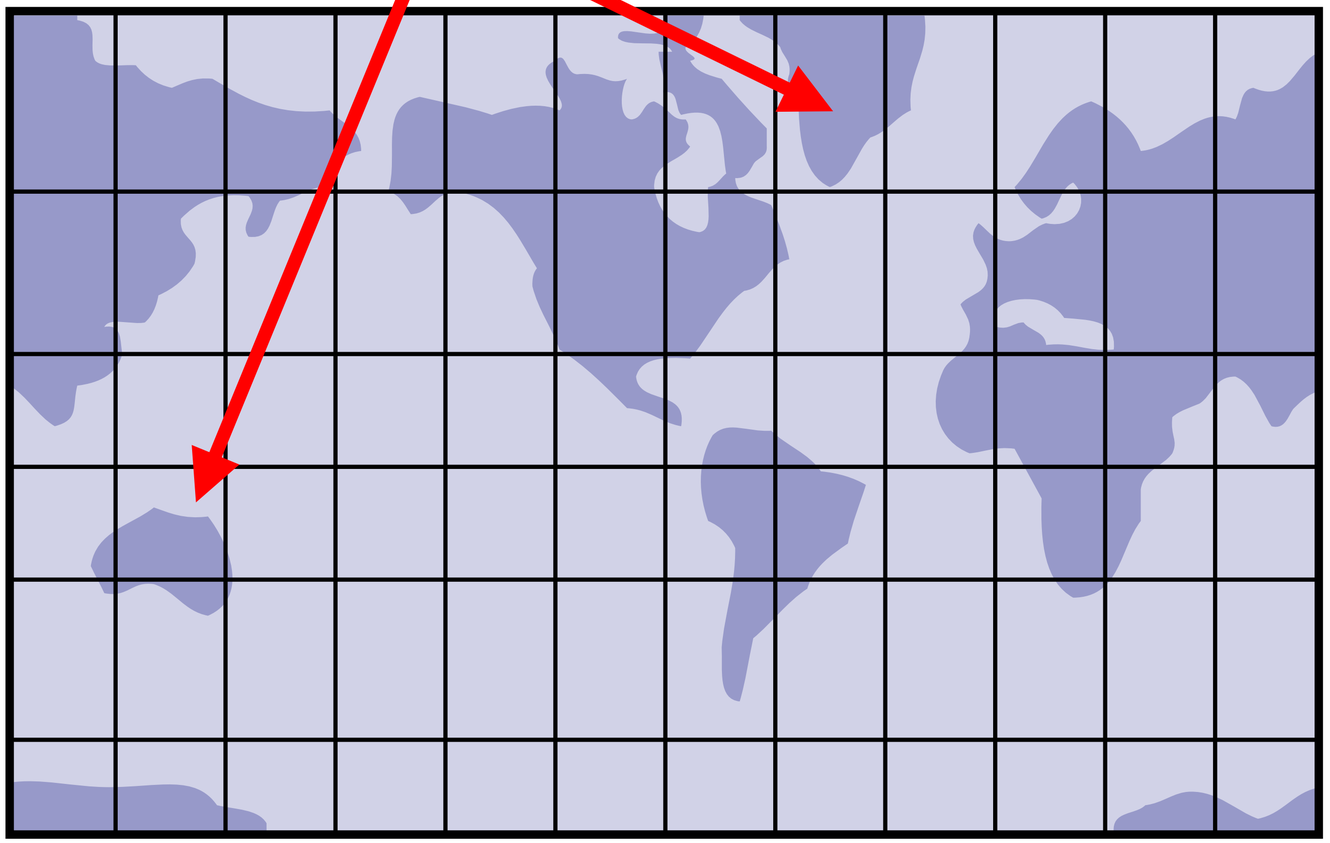

*(Very useful for navigation!)*



# *Conformal Mapmaking*

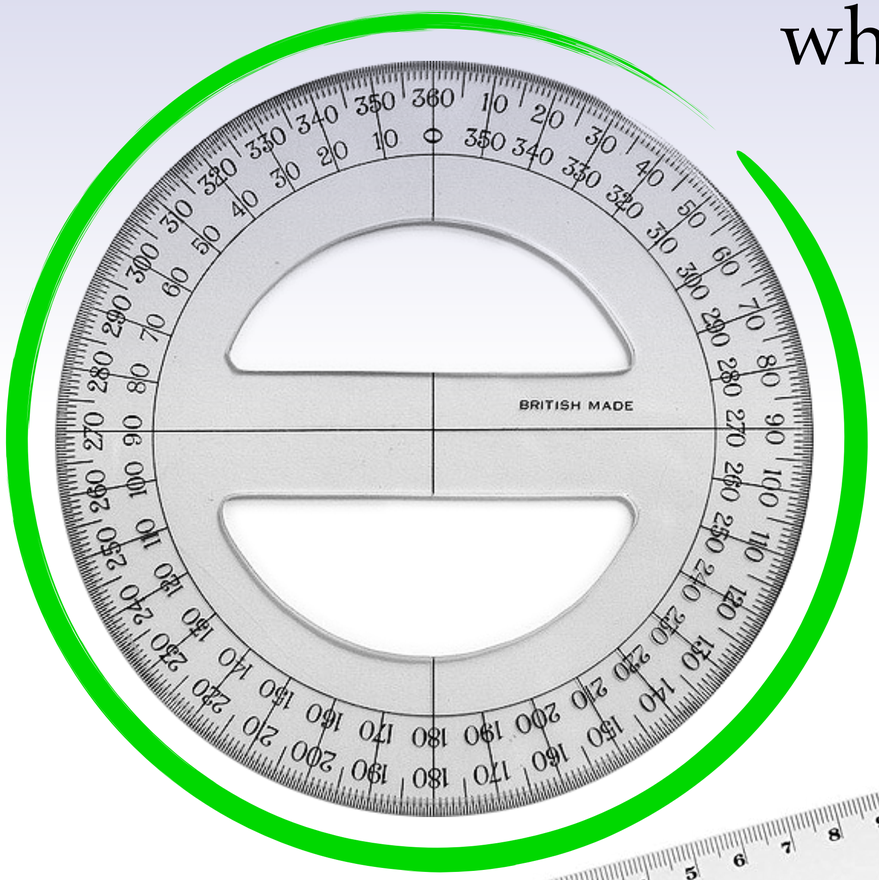
- However, **areas** may be badly distorted…

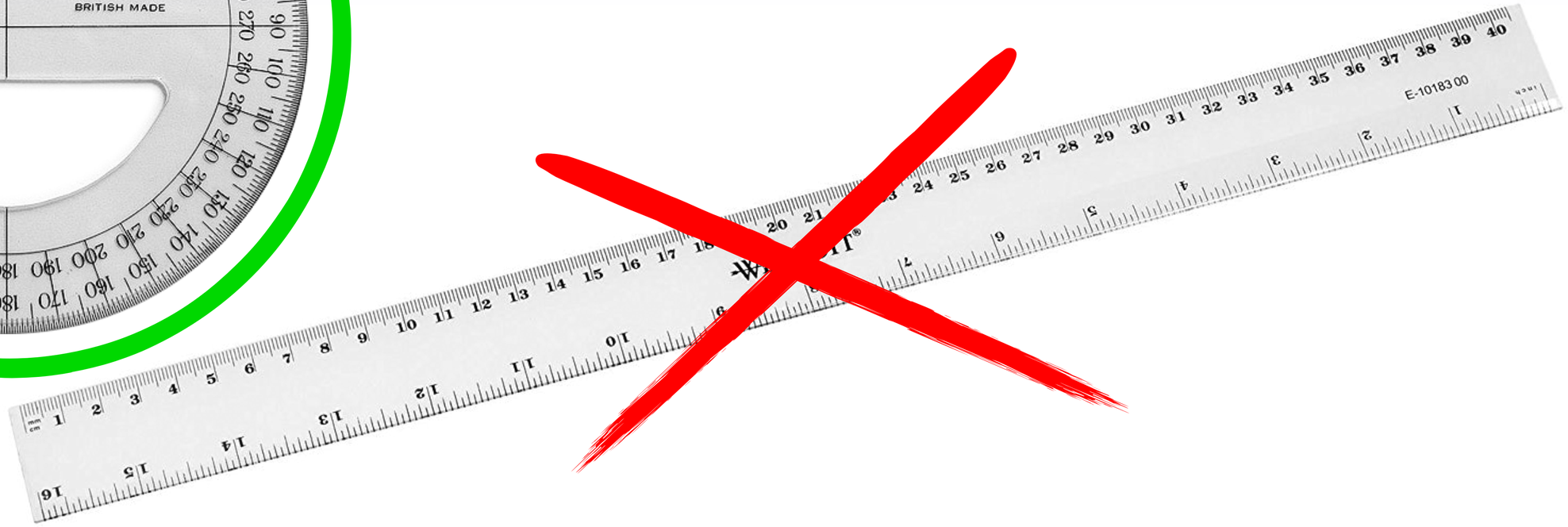
*(Greenland is not bigger than Australia!)*



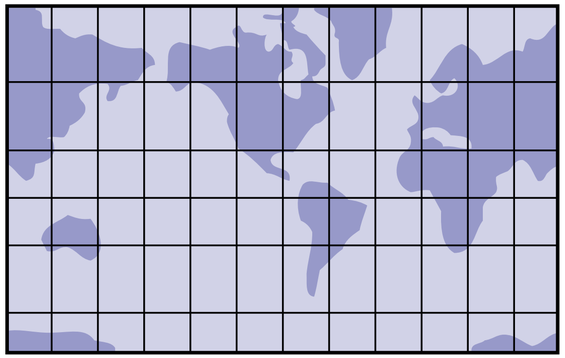
# *Conformal Geometry*

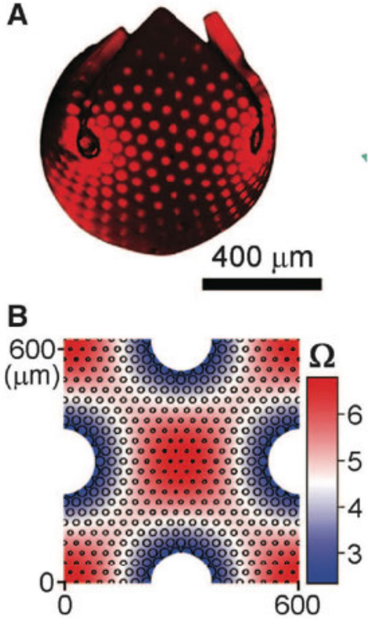
More broadly, *conformal geometry* is the study of shape when one can measure only **angle** (not length).

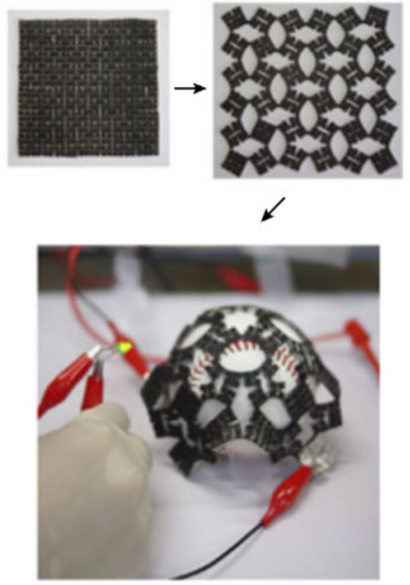


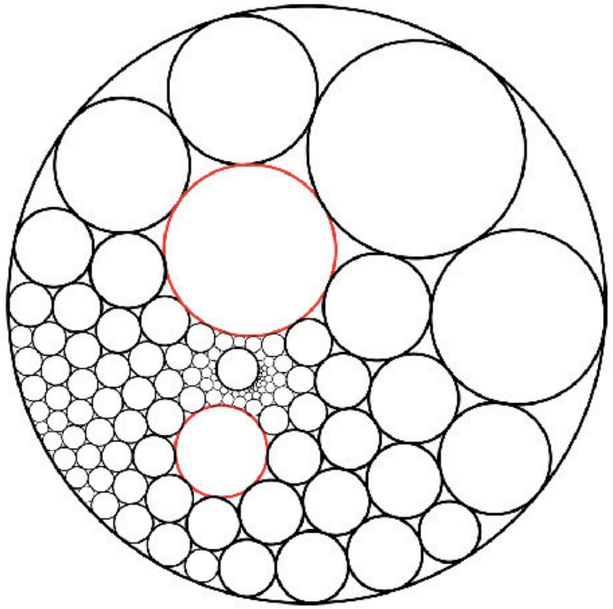



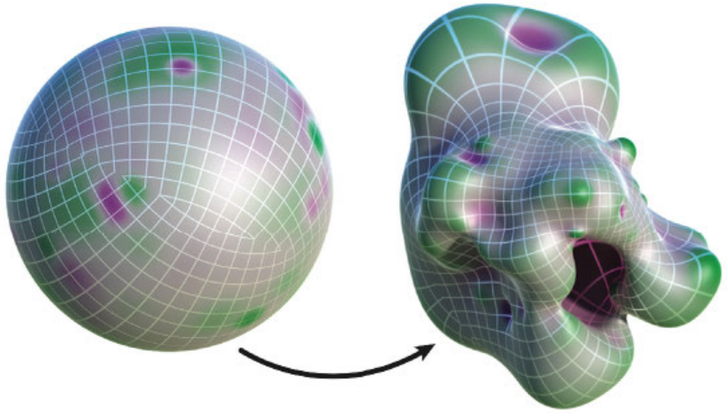
# *Conformal Geometry—Visualized*

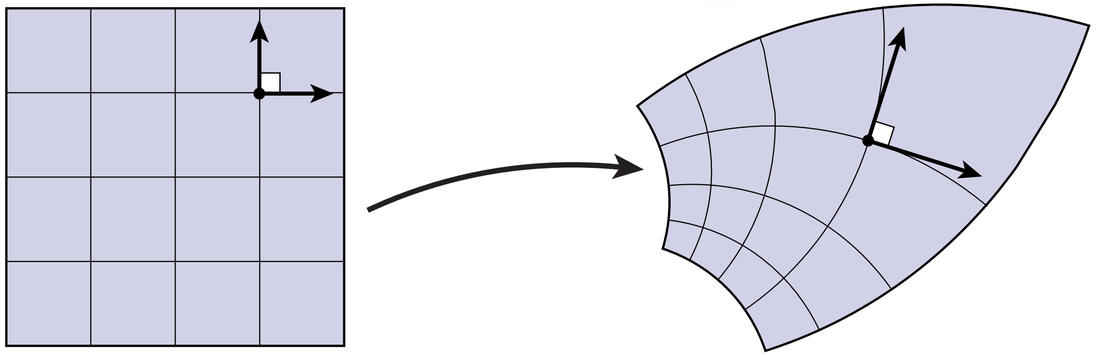


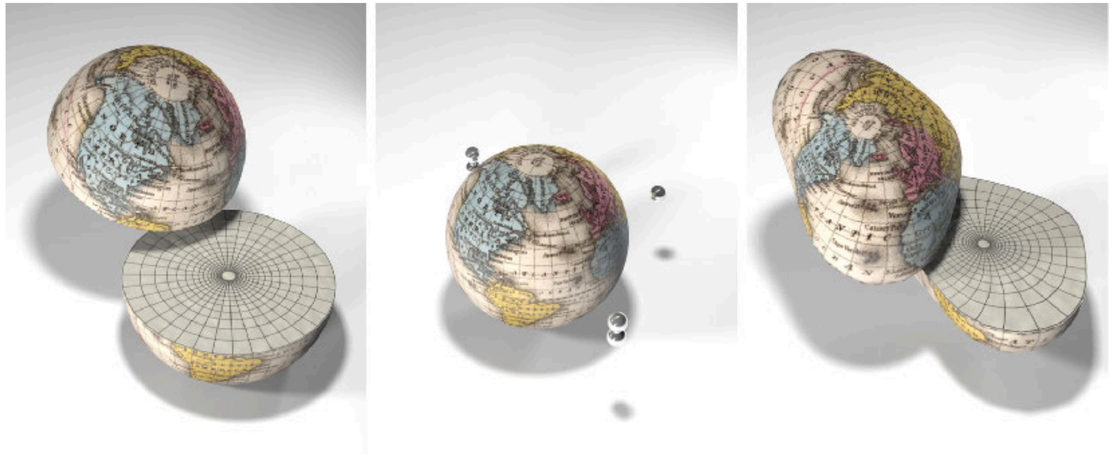




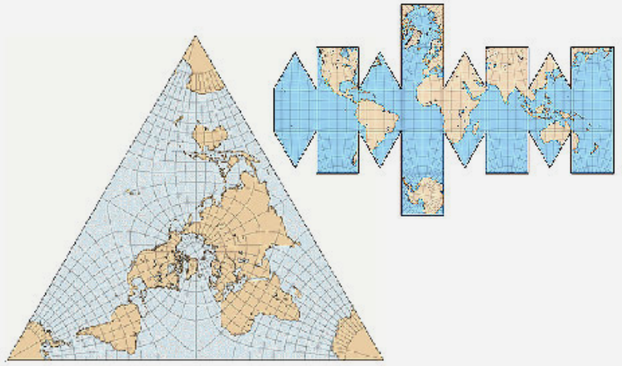




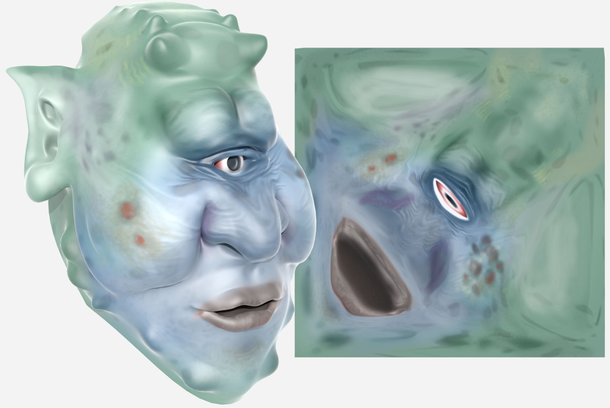


# *Applications of Conformal Geometry Processing*

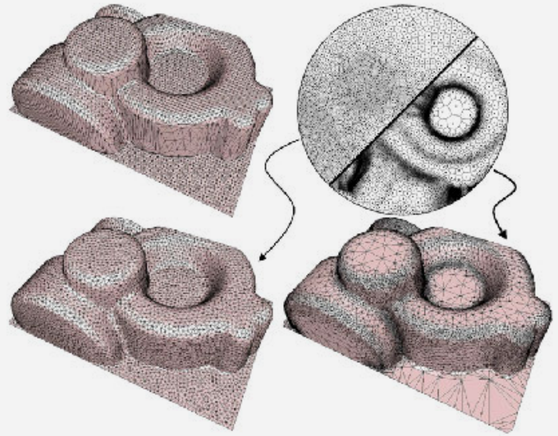
Basic building block for *many* applications…



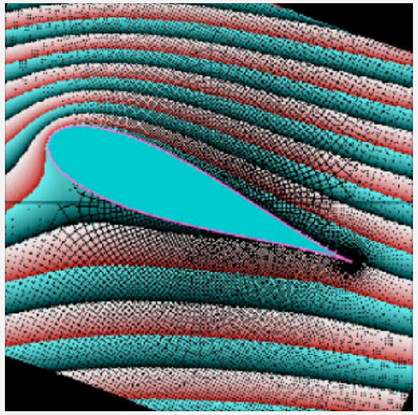
CARTOGRAPHY



TEXTURE MAPPING

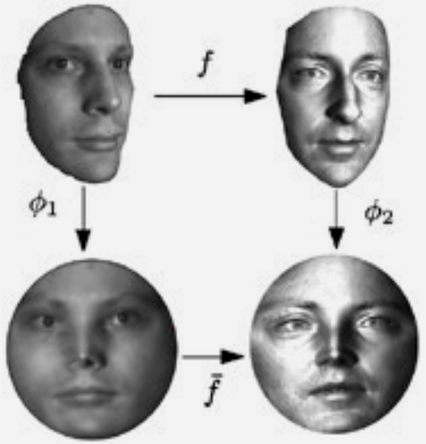


REMESHING

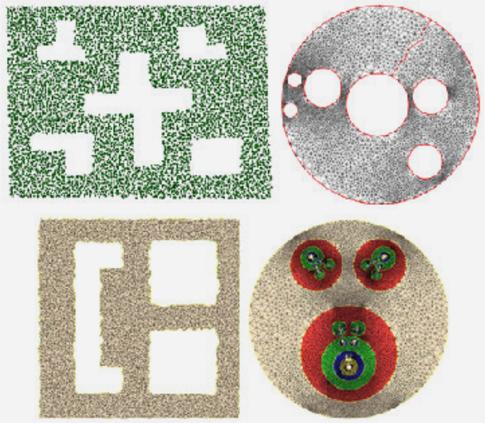


SIMULATION

3D FABRICATION



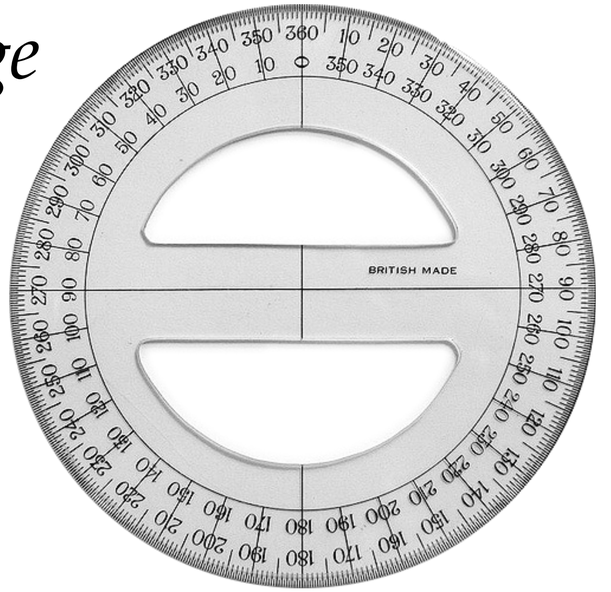

SHAPE ANALYSIS



SENSOR NETWORKS

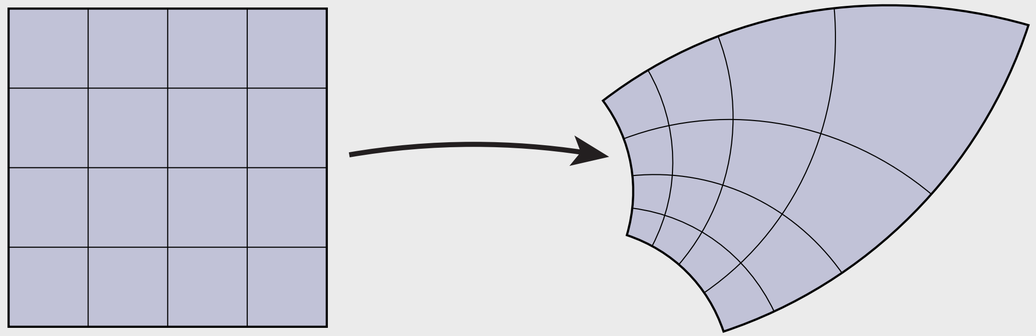
# *Why **Conformal**?*

- Why so much interest in maps that preserve *angle*?
  - **QUALITY**: *Every conformal map is already "really nice"*
  - **SIMPLICITY**: *Makes "pen and paper" analysis easier*
  - **EFFICIENCY**: *Often yields computationally easy problems*
  - **GUARANTEES**: *Well understood, lots of theorems/knowledge*

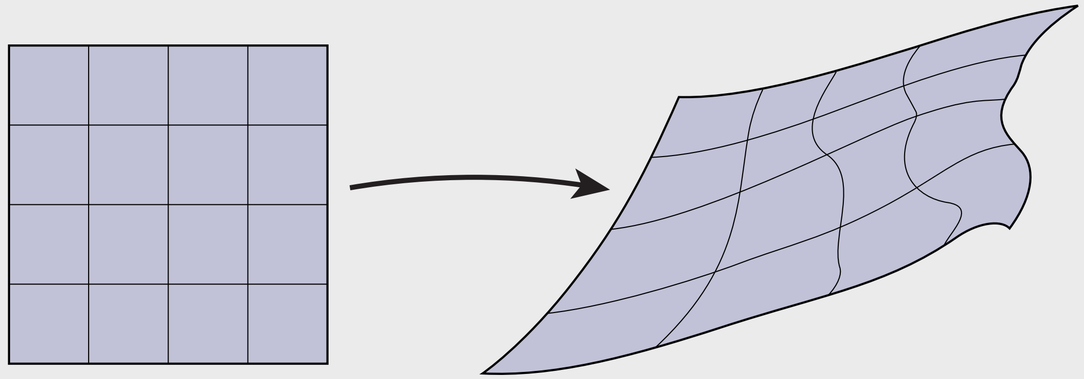


# *Conformal Maps are "Really Nice"*

- Angle preservation already provides a lot of regularity
- E.g., every conformal map has infinitely many derivatives ($C^\infty$)
- Scale distortion is smoothly distributed (harmonic)



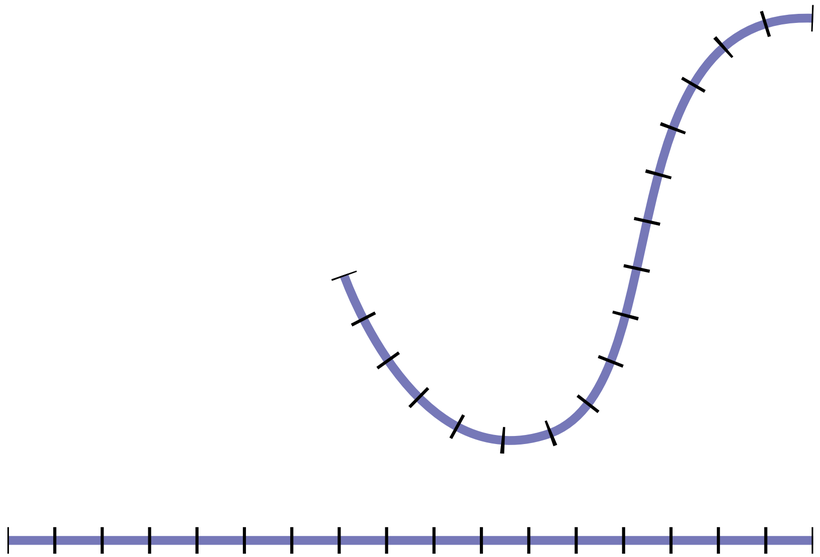
CONFORMAL

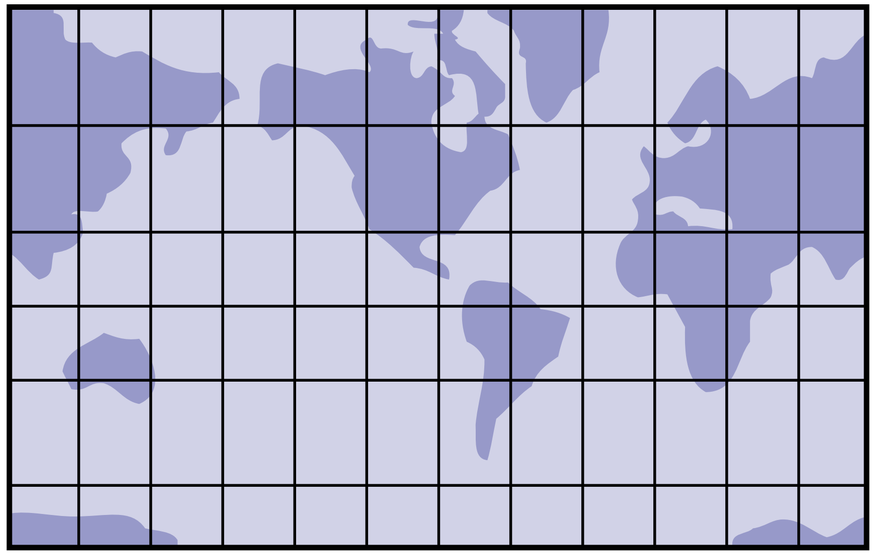


NOT CONFORMAL

# *Conformal Coordinates Make Life Easy*

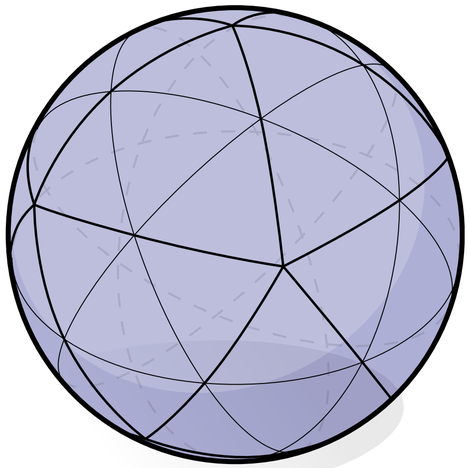
- Makes life easy "on pen and paper"
  - **Curves:** life greatly simplified by assuming *arc-length* parameterization
  - **Surfaces:** "arc-length" (isometric) not usually possible
    - conformal coordinates are "next best thing" (and always possible!)
    - only have to keep track of scale (rather than arbitrary Jacobian)



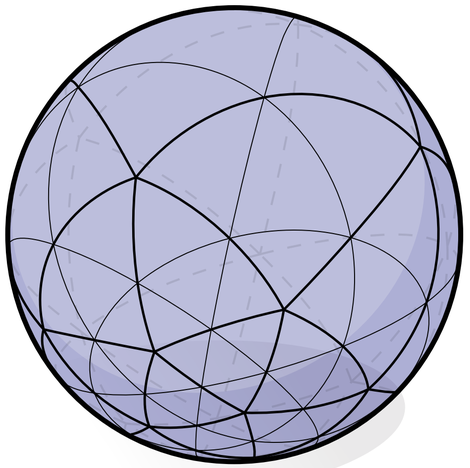


# *Aside: Isn't Area-Preservation "Just as Good?"*

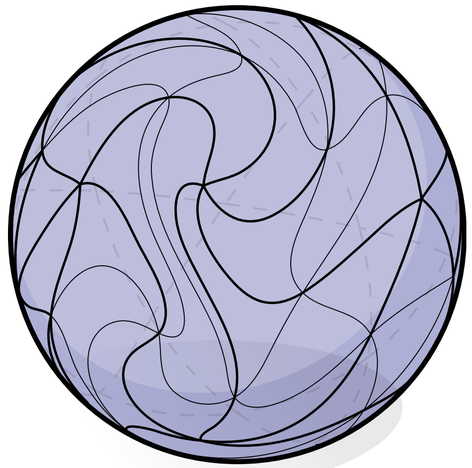
- **Q:** What's so special about *angle*? Why not preserve, say, *area* instead?
- **A:** Area-preservation alone can produce maps that are ***nasty!***
  - Don't even have to be smooth; *huge* space of possibilities.
- E.g., any motion of an incompressible fluid (e.g., swirling water):



ORIGINAL



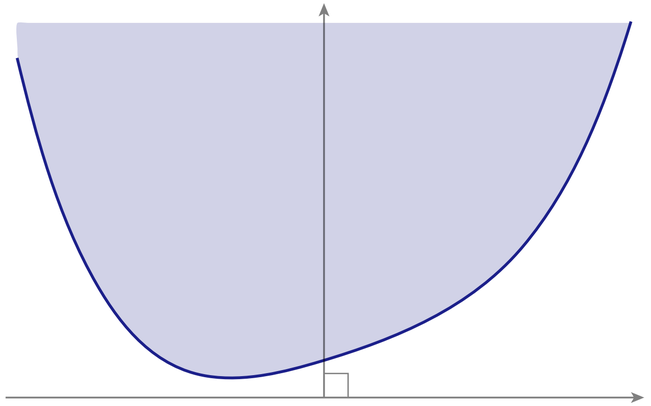
ANGLE PRESERVING



AREA PRESERVING

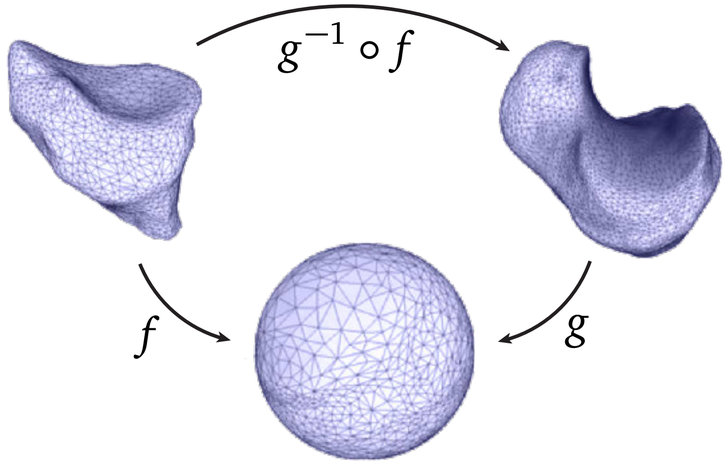
# *Computing Conformal Maps is Efficient*

- Algorithms boil down to efficient, scalable computation
  - sparse linear systems / sparse eigenvalue problems
  - convex optimization problems
- Compare to more elaborate mapping problems
  - *bounded distortion*, *locally injective*, etc.
  - entail more difficult problems (e.g., *SOCP*)
- Much broader domain of applicability
  - real time vs. “just once”



# *Conformal Maps Help Provide Guarantees*

- Established topic*
  - lots of existing theorems, analysis
  - connects to standard problems (e.g., Laplace)
  - makes it easier to provide guarantees (max principle, Delaunay, *etc.*)
- *Uniformization theorem* provides (nearly) canonical maps

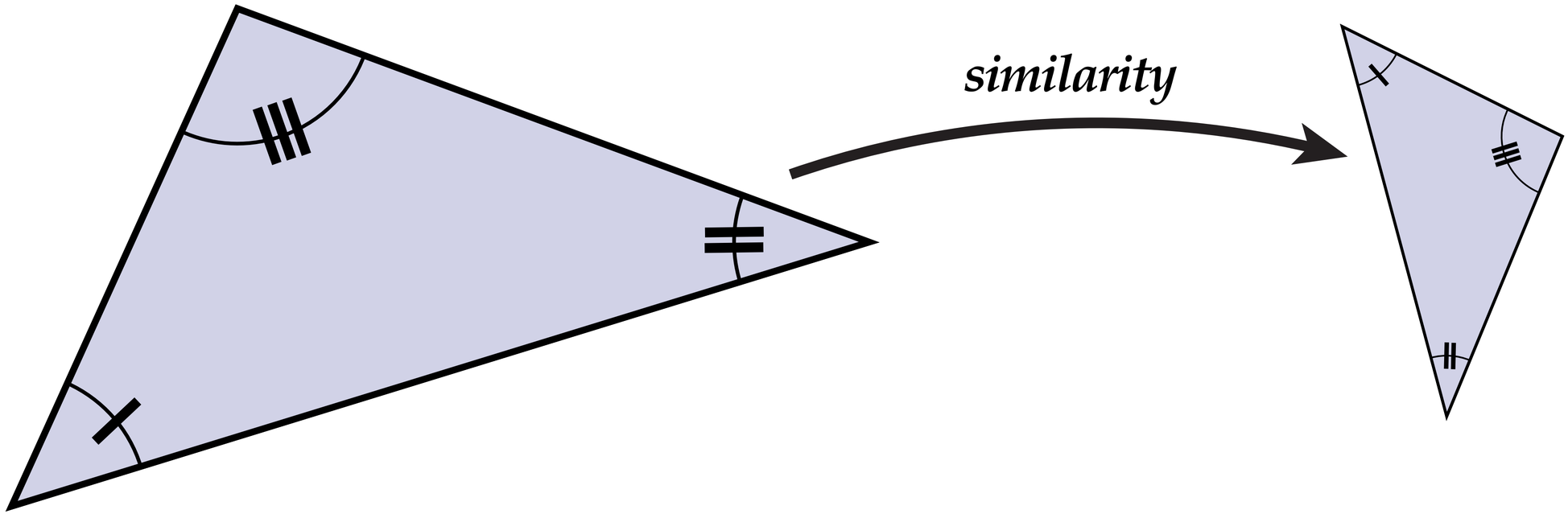



**Also makes it harder to do something truly new in conformal geometry processing…!*

# *Discrete Conformal Maps?*

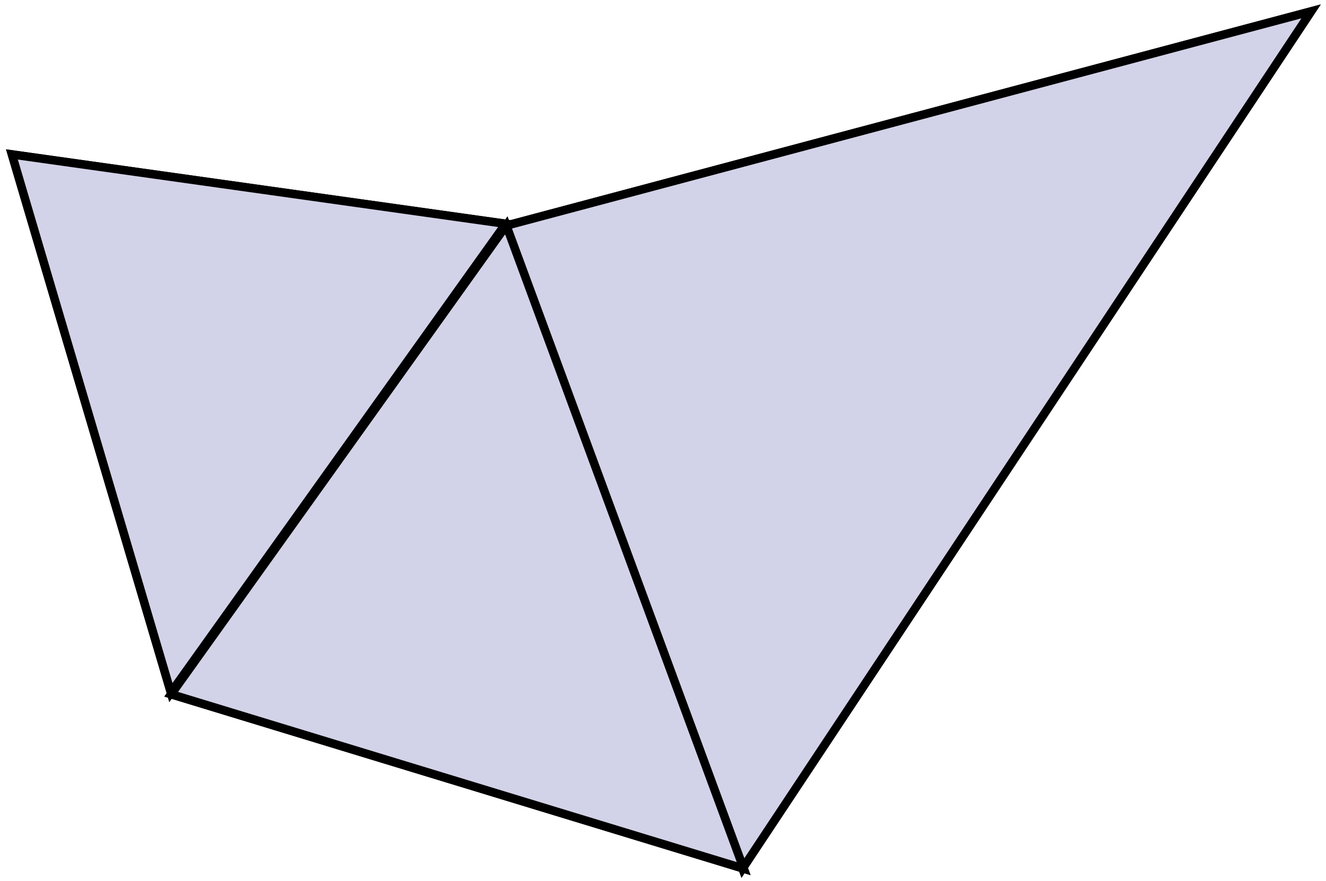
To compute conformal maps, we need some finite "discretization."

**First attempt:** preserve corner angles in a triangle mesh:



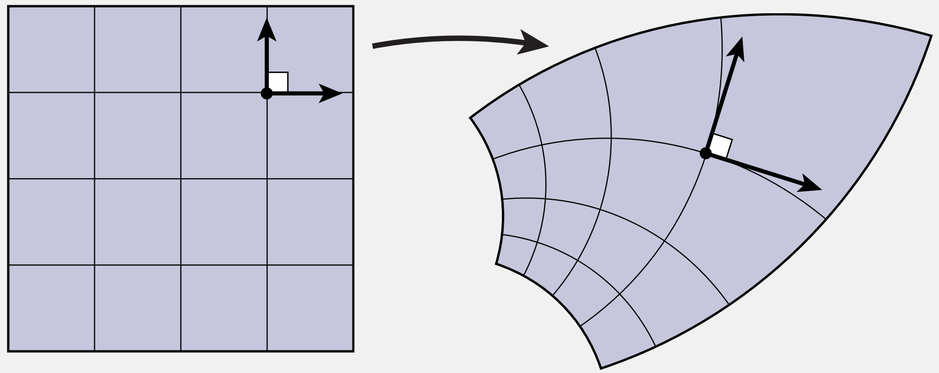
# *Rigidity of Angle Preservation*

**Problem:** one triangle determines the entire map! (Too “rigid”)

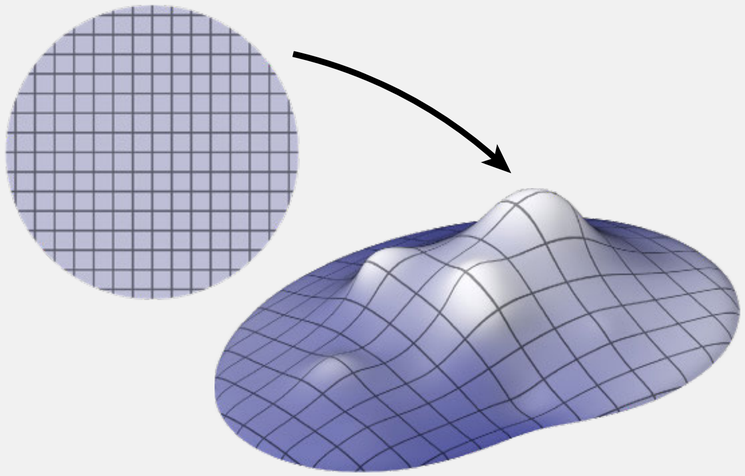


Need a different way of thinking…

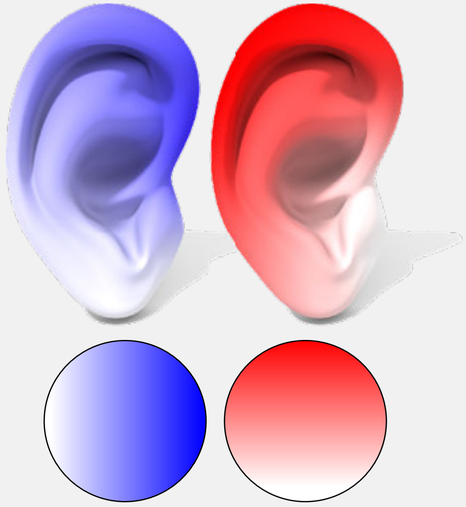
# *(Some) Characterizations of Conformal Maps*



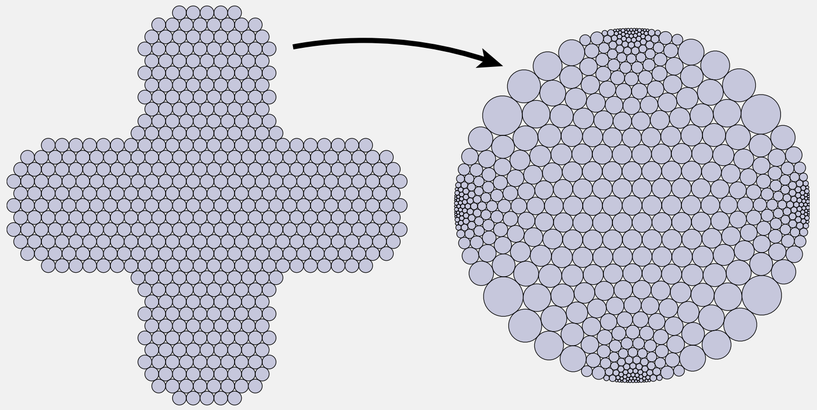
**angle preservation**



**metric rescaling**



**conjugate harmonic functions**



**preservation of circles**

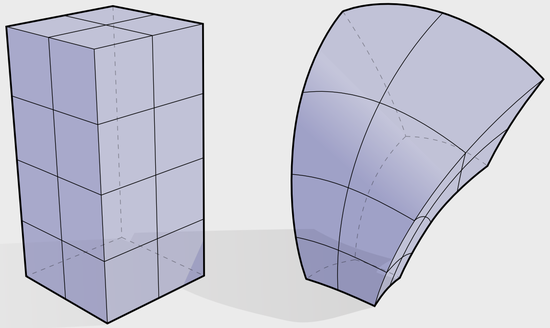
**critical points of Dirichlet energy**

# *(Some) Conformal Geometry Algorithms*

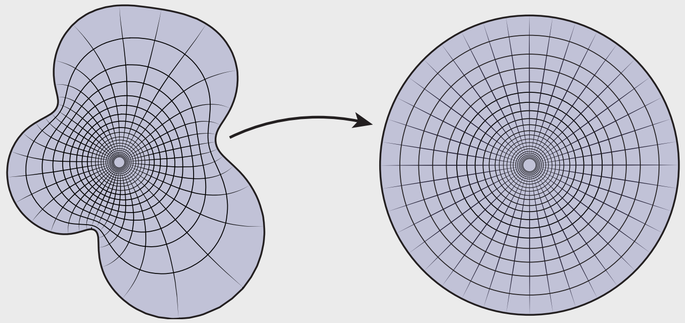
| CHARACTERIZATION | ALGORITHMS |
|---|---|
| **Cauchy-Riemann** | *least square conformal maps (LSCM)* |
| **Dirichlet energy** | *discrete conformal parameterization (DCP)*<br>*genus zero surface conformal mapping (GZ)* |
| **angle preservation** | *angle based flattening (ABF)* |
| **circle preservation** | *circle packing*<br>*circle patterns (CP)* |
| **metric rescaling** | *conformal prescription with metric scaling (CPMS)*<br>*conformal equivalence of triangle meshes (CETM)* |
| **conjugate harmonic** | *boundary first flattening (BFF)* |

# *Some Key Ideas in Conformal Surface Geometry*

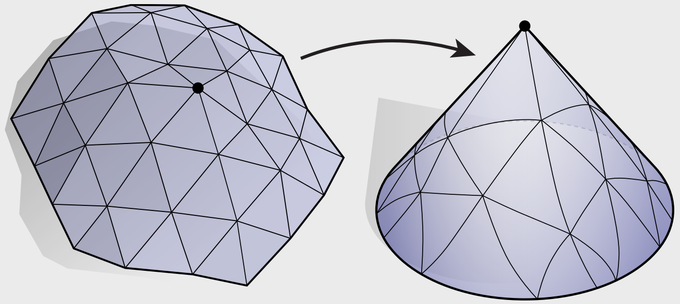
## Möbius Transformations / Stereographic Projection



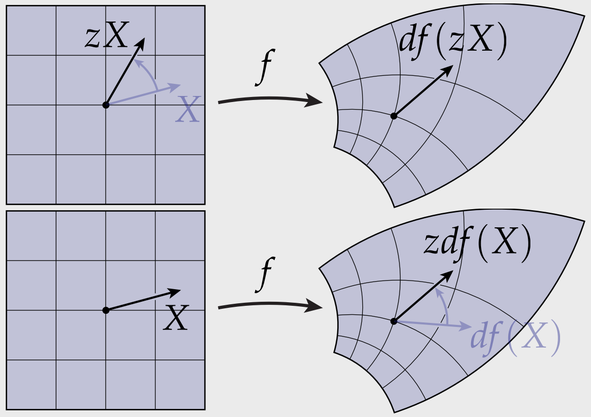
## Riemann Mapping / Uniformization



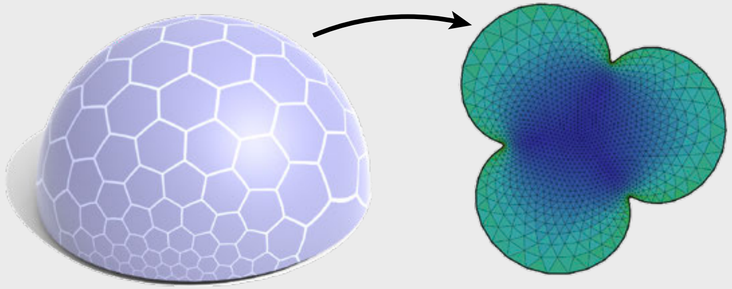
## Cone Singularities



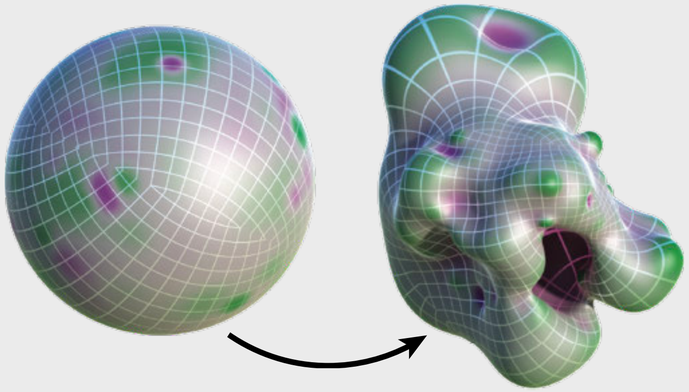
## Cauchy-Riemann Equation

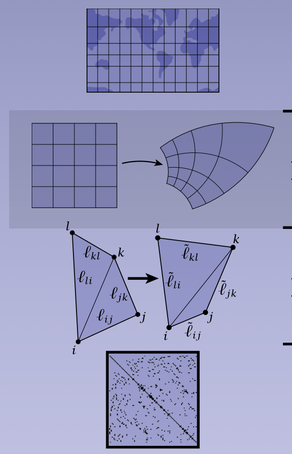



## Ricci Flow / Cherrier Formula



## Dirac Equation







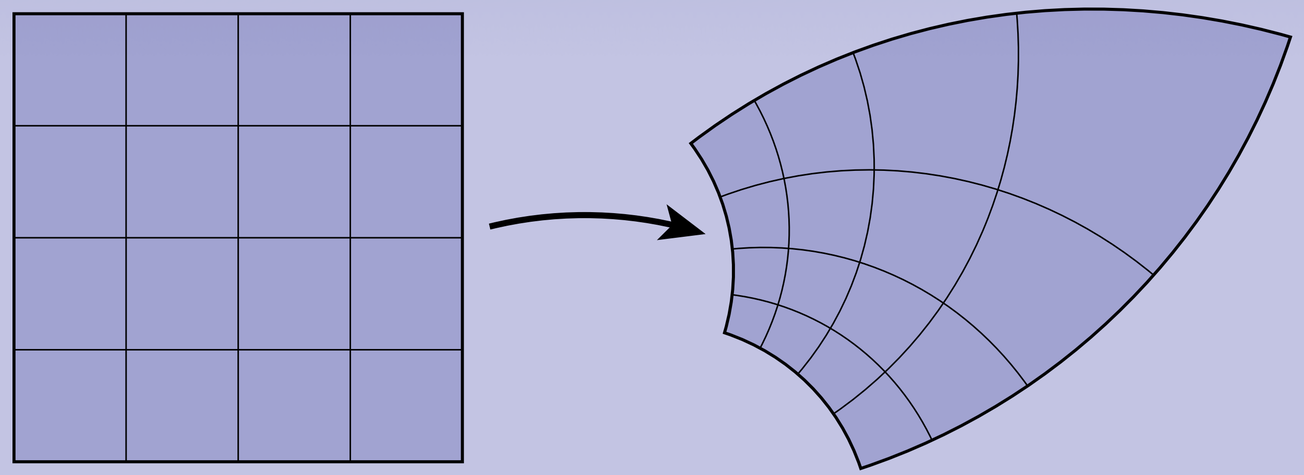
PART I: OVERVIEW

PART II: SMOOTH THEORY

PART III: DISCRETIZATION

PART IV: ALGORITHMS

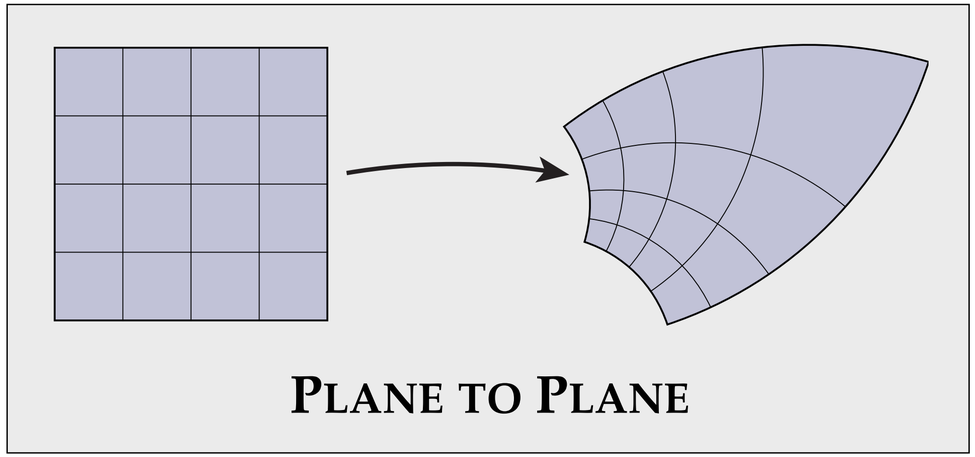

# PART II: SMOOTH THEORY



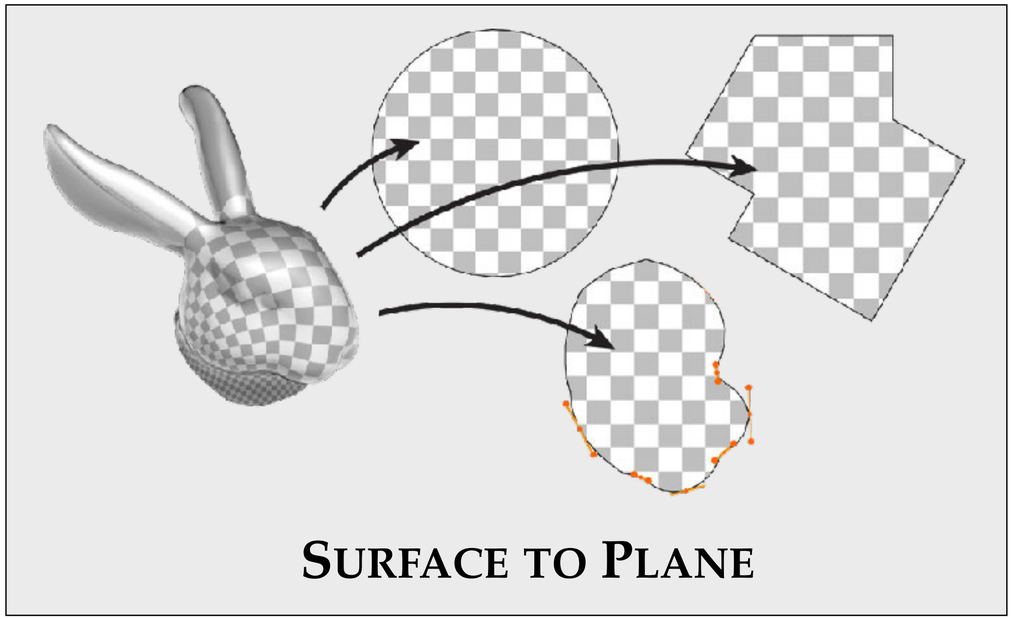
# CONFORMAL GEOMETRY PROCESSING


**Keenan Crane • CMU 15-458/858B • Fall 2017**

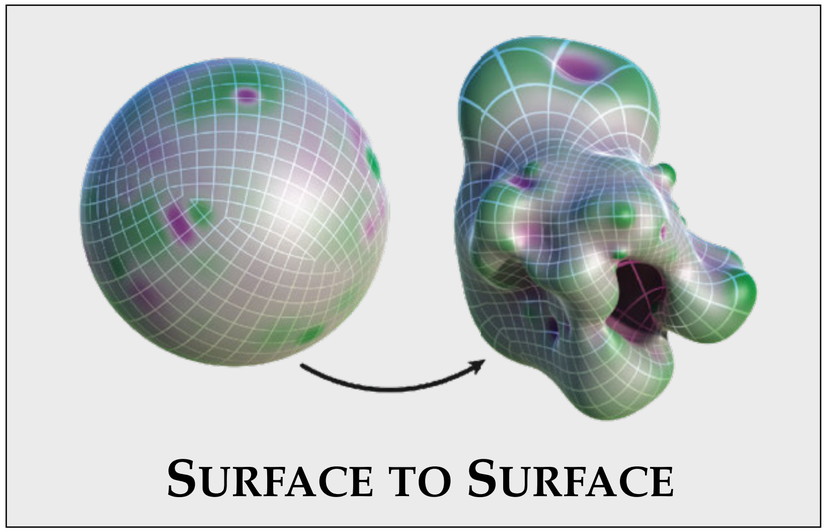
# *Conformal Maps of Surfaces*



PLANE TO PLANE



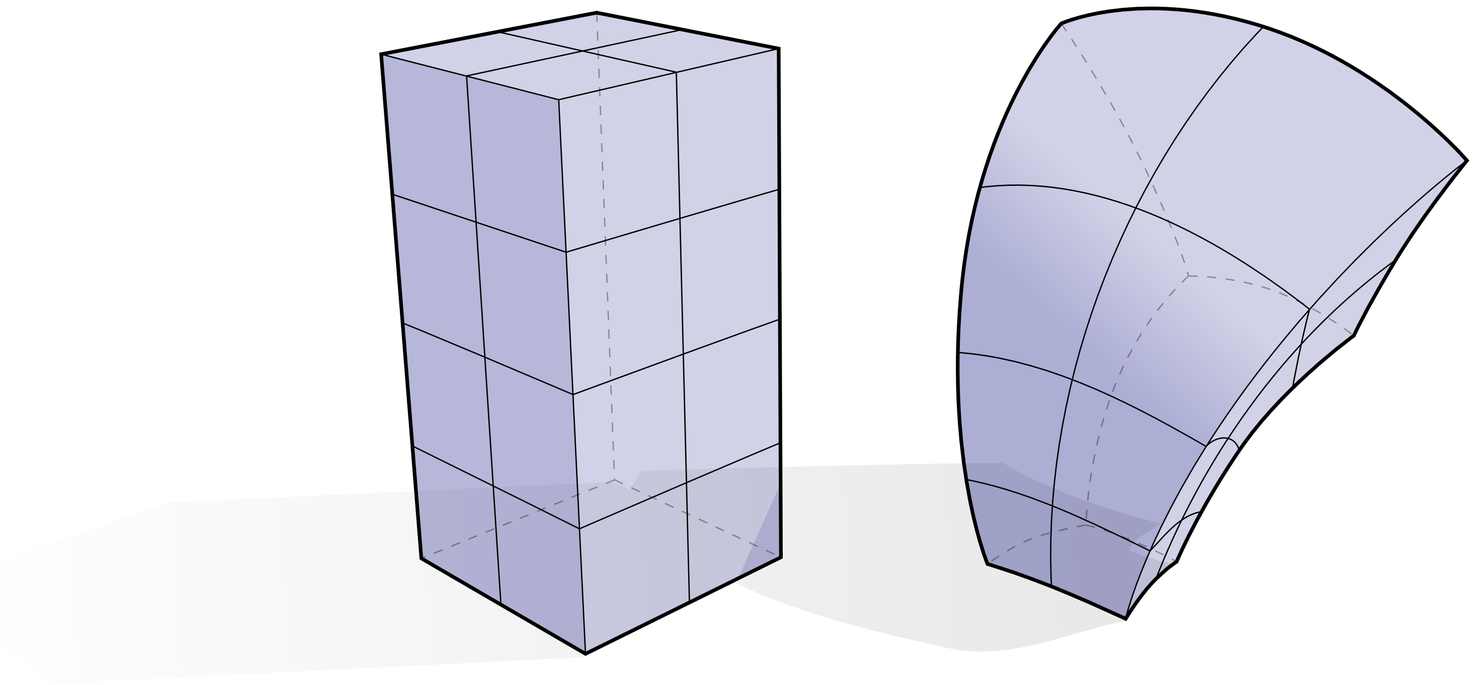
SURFACE TO PLANE



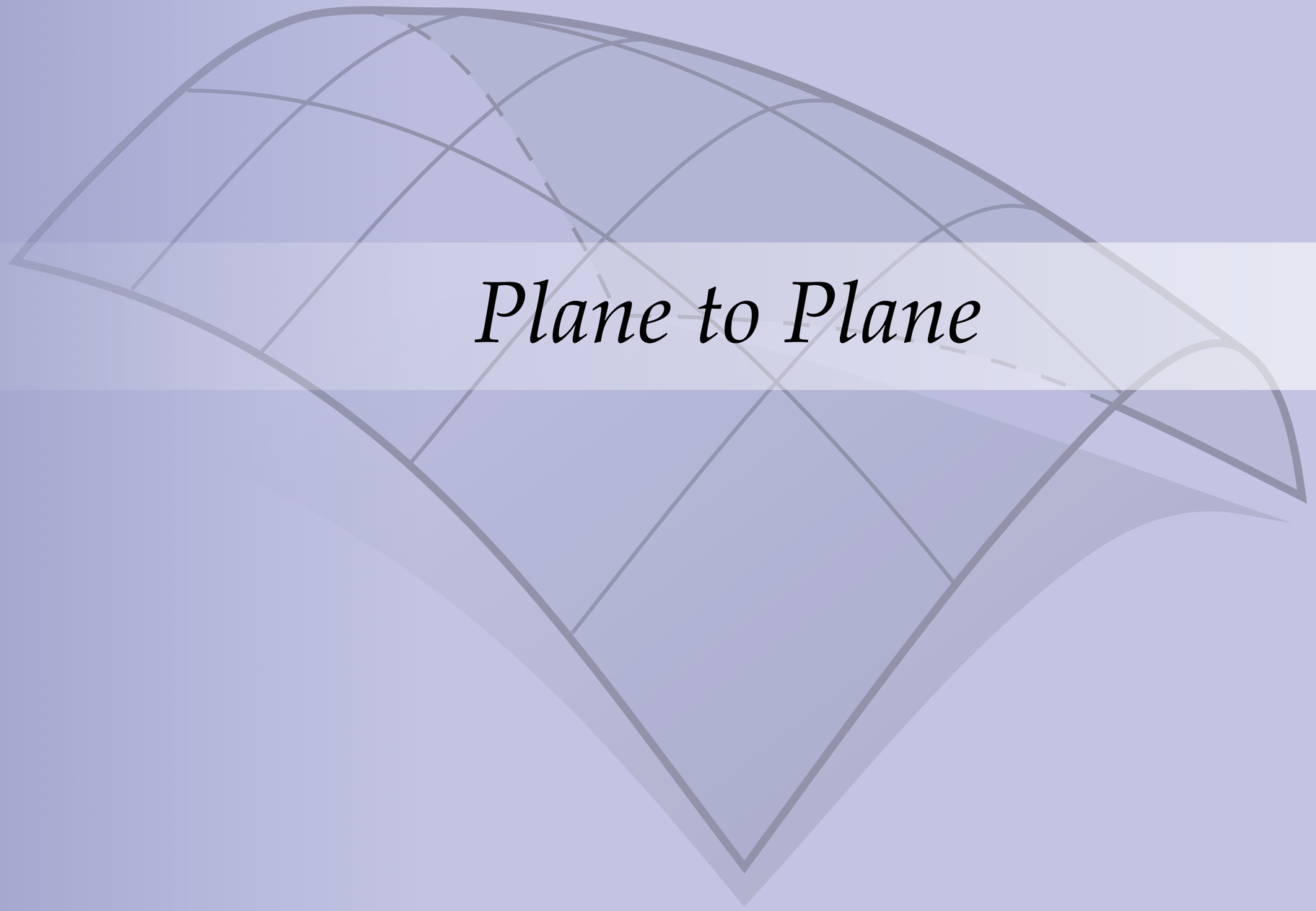
SURFACE TO SURFACE

# *Why Not Higher Dimensions?*

**Theorem** (Liouville)**.** For $n \geq 3$, the only angle-preserving maps from $\mathbb{R}^n$ to itself (or from a region of $\mathbb{R}^n$ to $\mathbb{R}^n$) are Möbius transformations.



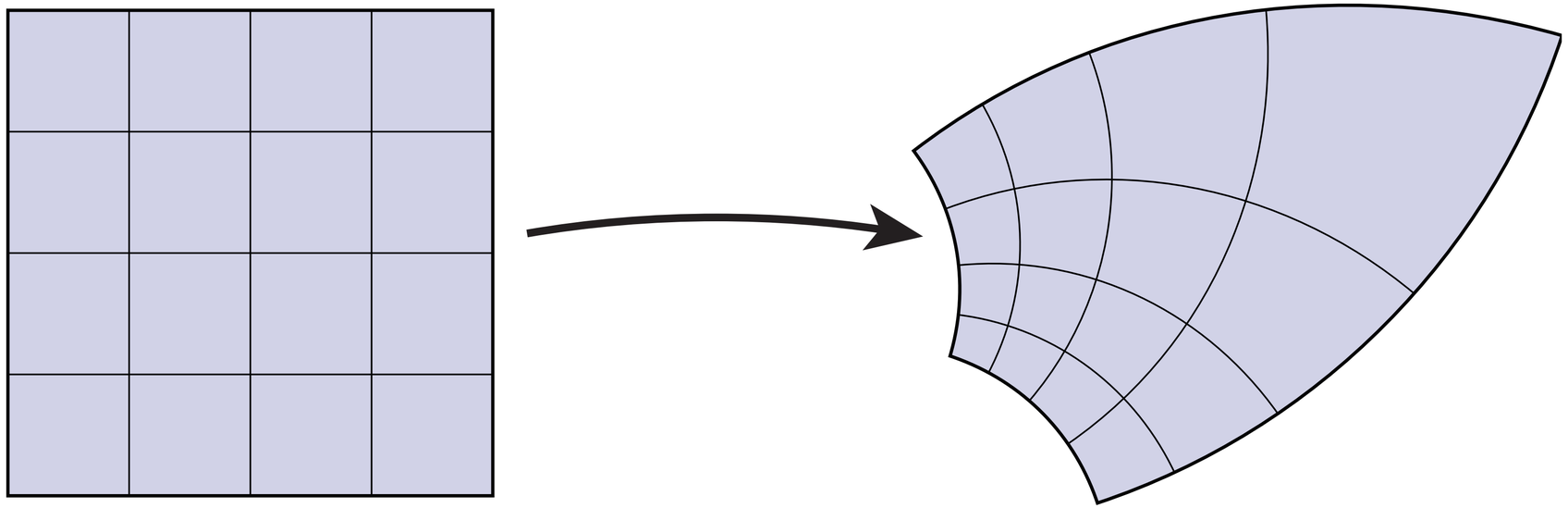
**Key idea:** conformal maps of volumes are *very* rigid.



# *Plane to Plane*

# *Plane to Plane*

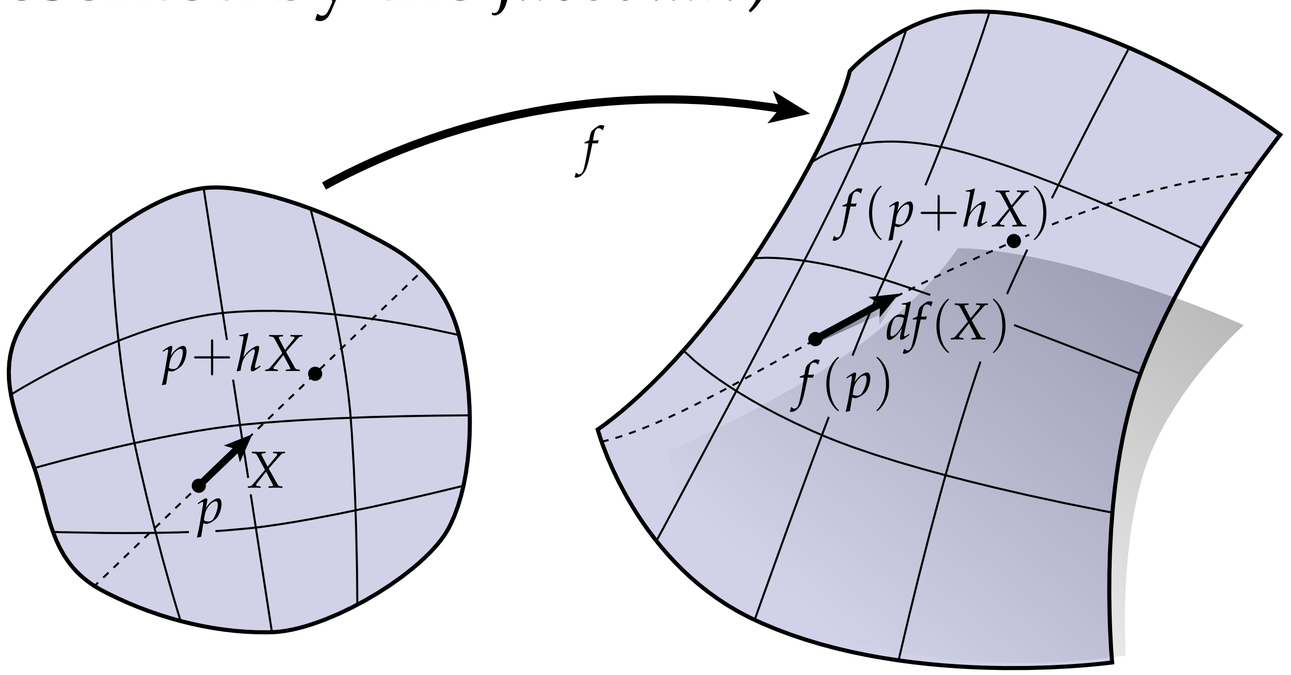
- Most basic case: conformal maps from region of 2D plane to 2D plane.
- Basic topic of complex analysis
- Fundamental equation: *Cauchy-Riemann*
- *Many* ideas we will omit (e.g., power series / analytic point of view)



# *Differential of a Map*

- Basic idea we'll need to understand: *differential* of a map
- Describes how to *"push forward"* vectors under a differentiable map
- (In coordinates, differential is represented by the *Jacobian*)

$$df(X) = \lim_{h \to 0} \frac{f(p + hX) - f(p)}{h}$$

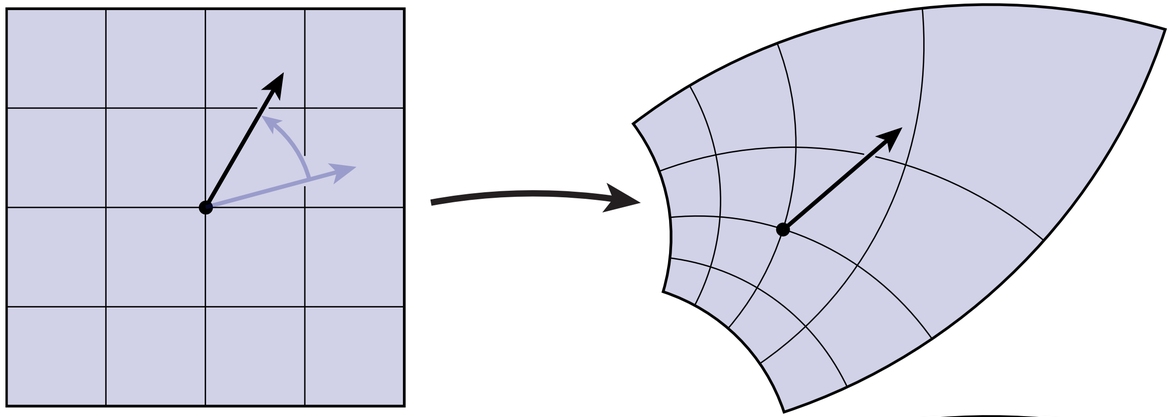



**Intuition:** "how do vectors get stretched out?"

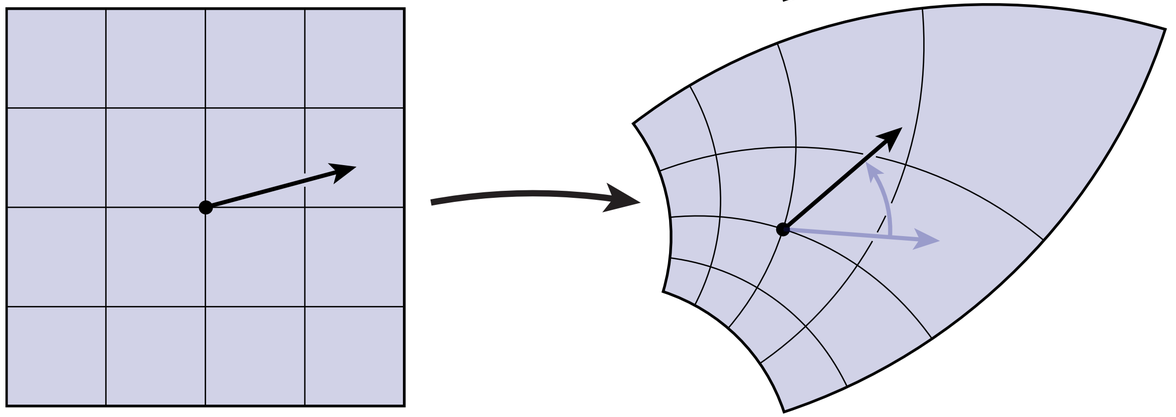
# *Conformal Map*

- A map is conformal if two operations are equivalent:

1. rotate, then push forward vector



2. push forward vector, then rotate

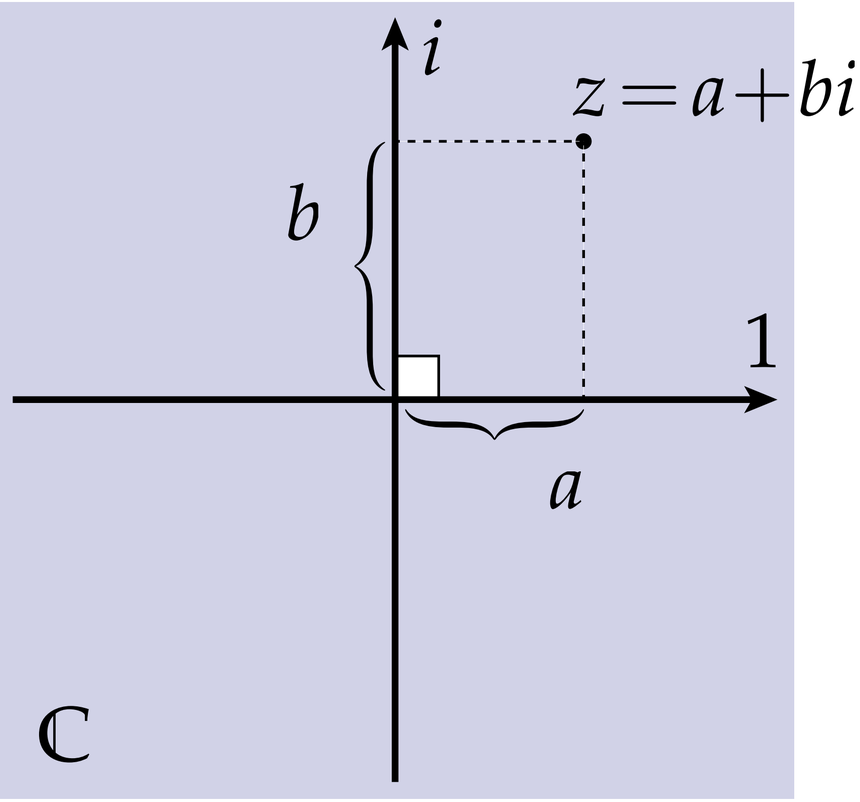


(How can we write this condition more explicitly?)

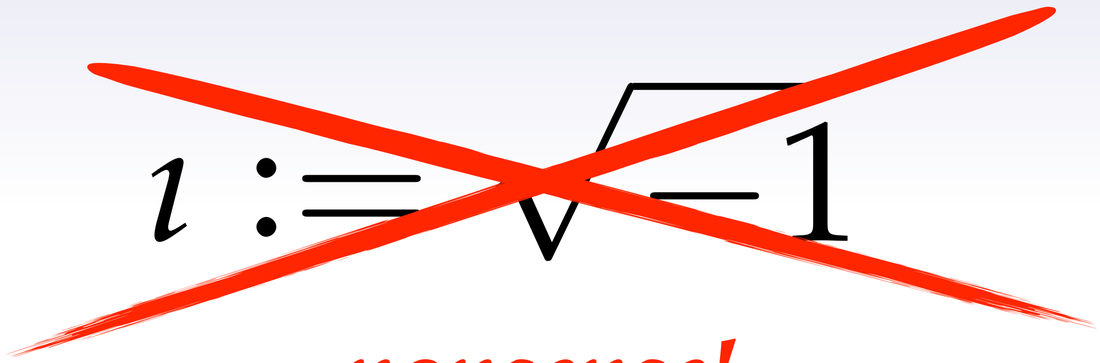
# *Complex Numbers*

- Not much different from the usual Euclidean plane
- Additional operations make it easy to express **scaling & rotation**
- Extremely natural for conformal geometry

- Two basis directions: 1 and $i$
- Points expressed as $z = a+bi$



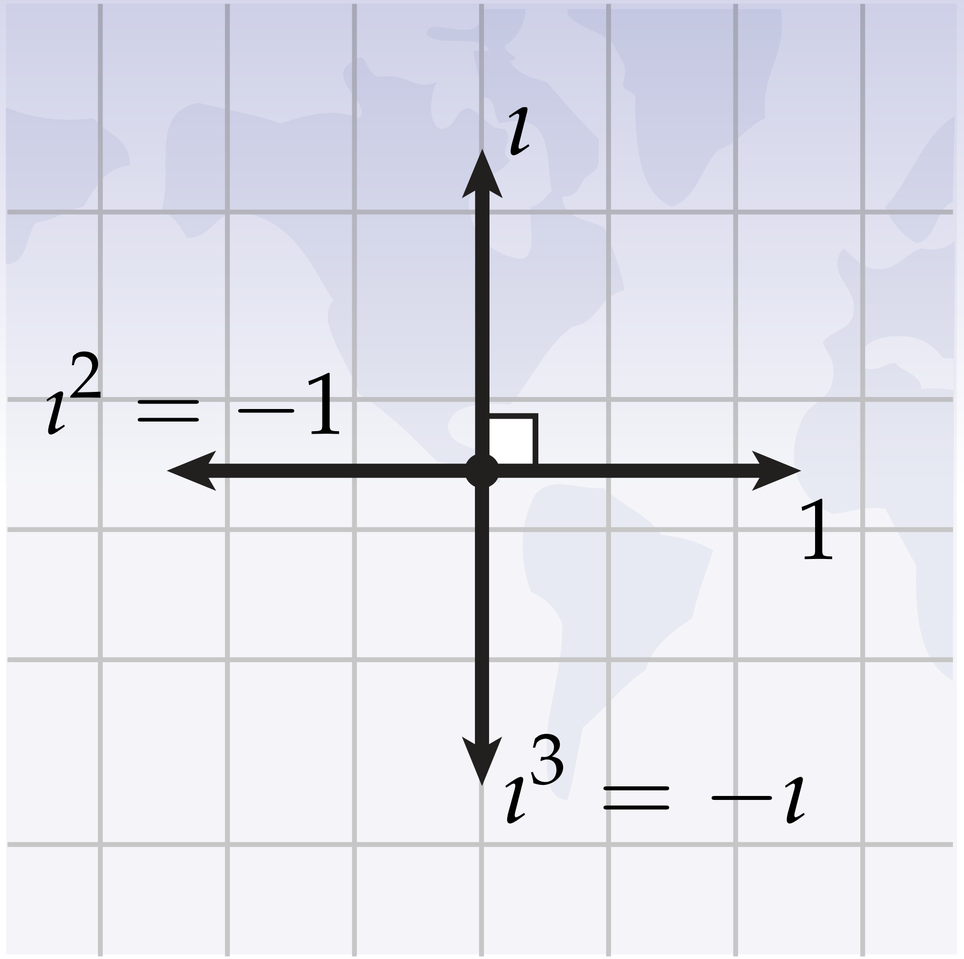
# *Complex Numbers*




***nonsense!***

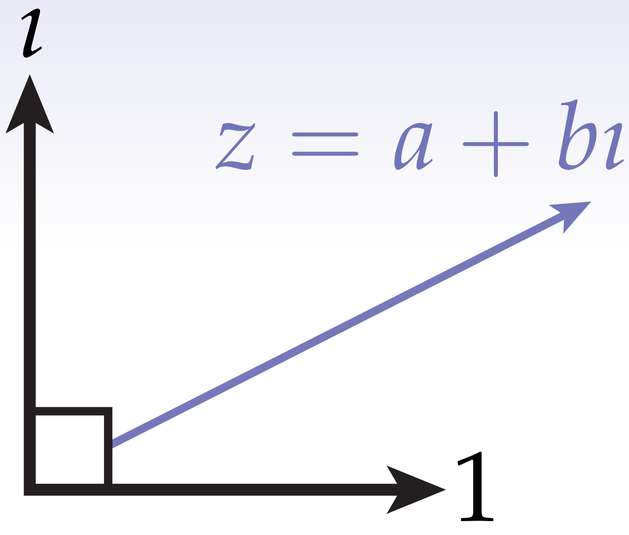
**More importantly:** obscures geometric meaning.

# *Imaginary Unit—Geometric Description*

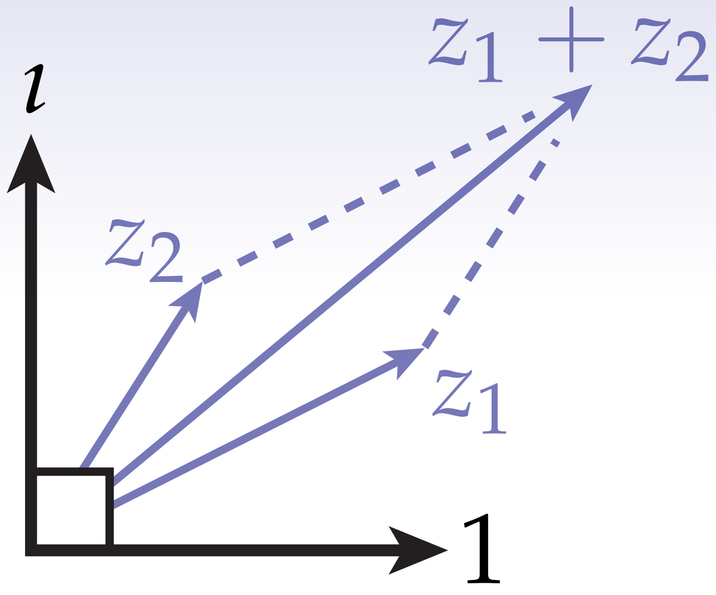



Symbol $\imath$ denotes *quarter-turn* in the *counter-clockwise* direction.

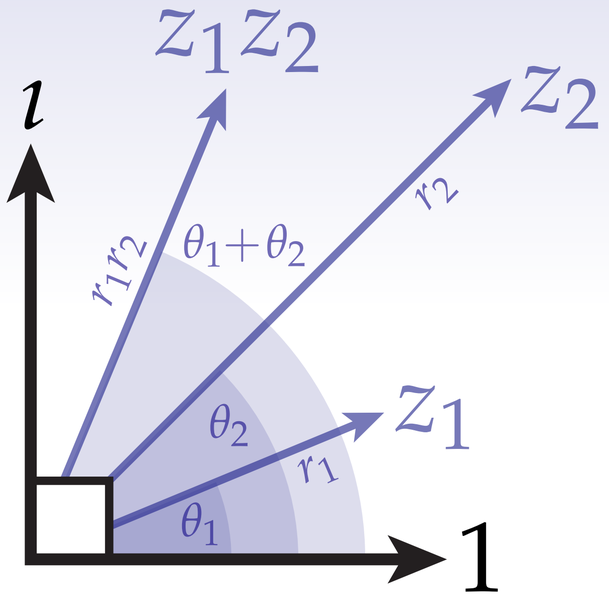
# *Complex Arithmetic—Visualized*




*rectangular coordinates*




*addition*




*multiplication*

# *Complex Product*

- Usual definition:
- Complex product distributes over addition. Hence,

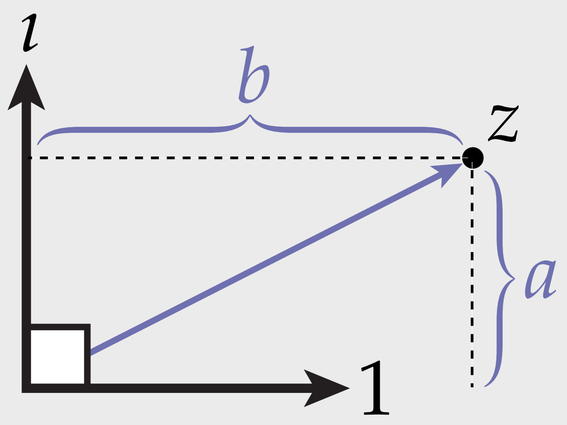
$$\begin{aligned} z_1 &:= a + b\imath \\ z_2 &:= c + d\imath \end{aligned}$$

$$\begin{aligned} z_1 z_2 &= (a + b\imath)(c + d\imath) \\ &= (a + b\imath)c + (a + b\imath)d\imath \\ &= ac + bc\imath + ad\imath + bd\imath^2 \\ &= \boxed{(ac - bd) + (ad + bc)\imath} \end{aligned}$$

Ok, terrific… but what does it mean *geometrically?*

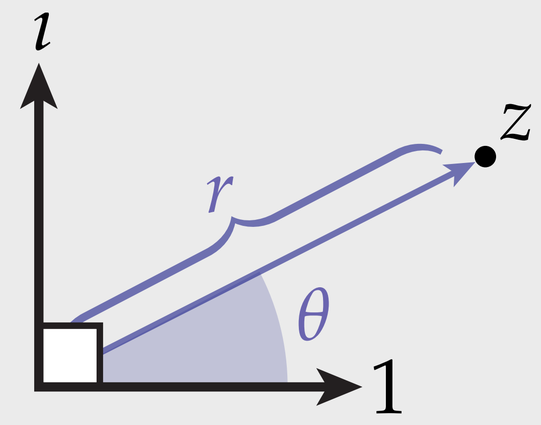
# *Rectangular vs. Polar Coordinates*

## RECTANGULAR




$$z = a + b\imath$$

## POLAR




$$z = r(\cos\theta + \imath\sin\theta) = re^{\imath\theta}$$

## EULER'S IDENTITY

$$\boxed{e^{\imath\theta} = \cos\theta + \imath\sin\theta}$$

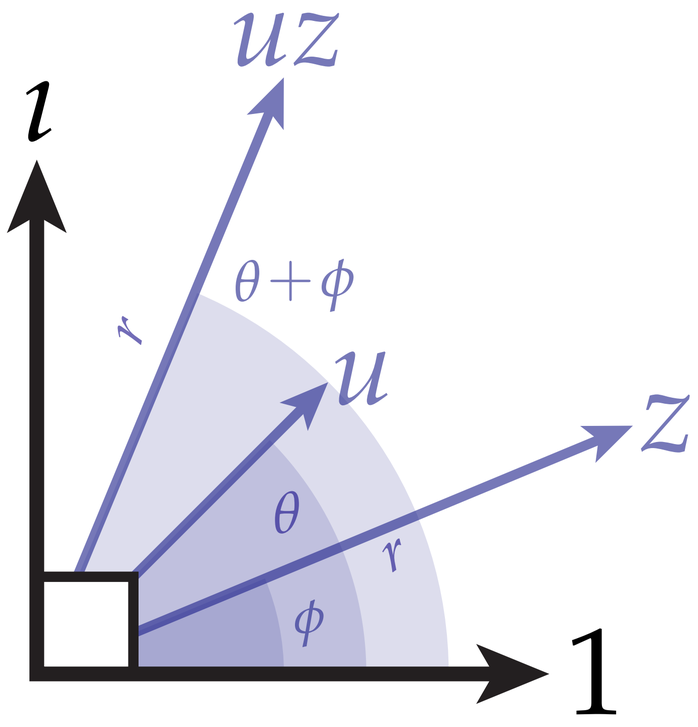
*(In practice: just convenient shorthand!)*

# *Rotations with Complex Numbers*

- How can we express rotation?
- Let $u$ be any *unit* complex number: $u = e^{\imath\theta}$
- Then for any point $z = re^{\imath\phi}$ we have

$$uz = (e^{\imath\theta})(re^{\imath\phi}) = re^{\imath(\theta+\phi)}$$

(same radius, new angle)

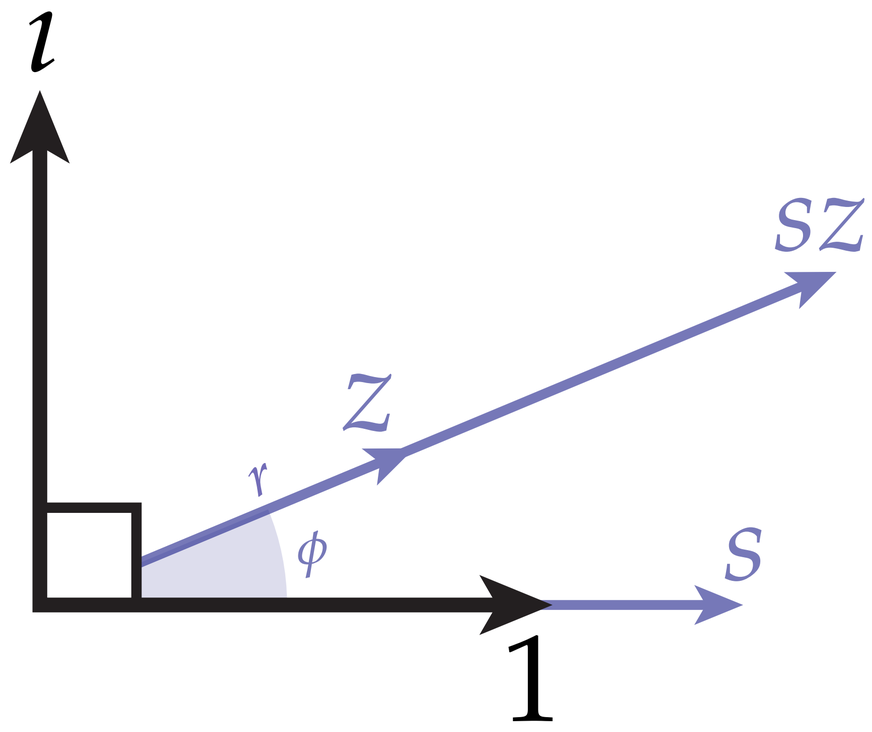


# *Scaling with Complex Numbers*

- How can we express scaling?
- Let $s$ be any *real* complex number: $s = a + 0\imath$
- Then for any point $z = re^{\imath\phi}$ we have

$$sz = (a + 0\imath)(re^{\imath\phi}) = are^{\imath\phi}$$

(same angle, new radius)



# *Complex Product—Polar Form*

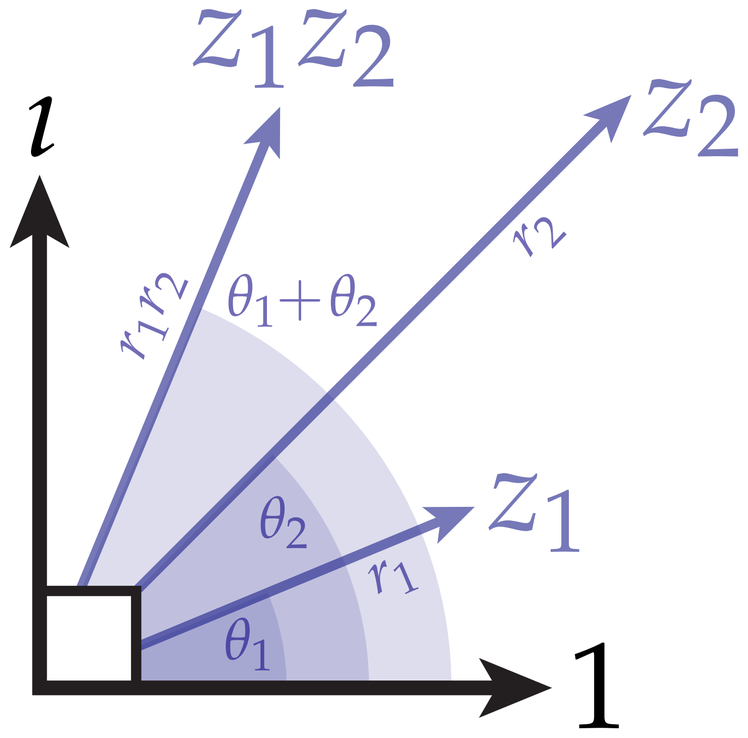
More generally, consider *any* two complex numbers:

$$\begin{aligned} z_1 &:= r_1 e^{\imath\theta_1} \\ z_2 &:= r_2 e^{\imath\theta_2} \end{aligned}$$

We can express their product as

$$z_1 z_2 = (r_1 e^{\imath\theta_1})(r_2 e^{\imath\theta_2}) = r_1 r_2 e^{\imath(\theta_1+\theta_2)}$$

- New angle is *sum* of angles
- New radius is *product* of radii

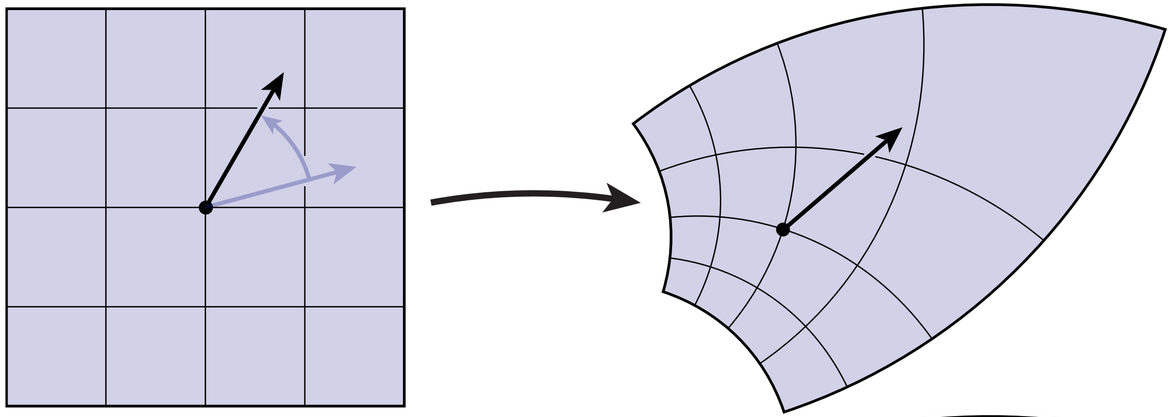



*(Now forget the algebra and remember the* ***geometry!****)*

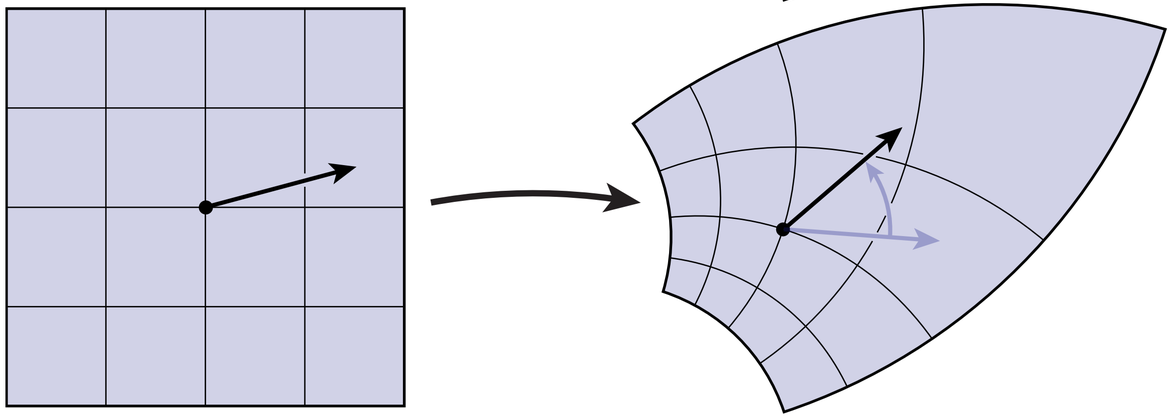
# *Conformal Map, Revisited*

- A map is conformal if two operations are equivalent:

1. rotate, then push forward vector



2. push forward vector, then rotate



**(How can we write this condition more explicitly?)**

# *Conformal Map, Revisited*

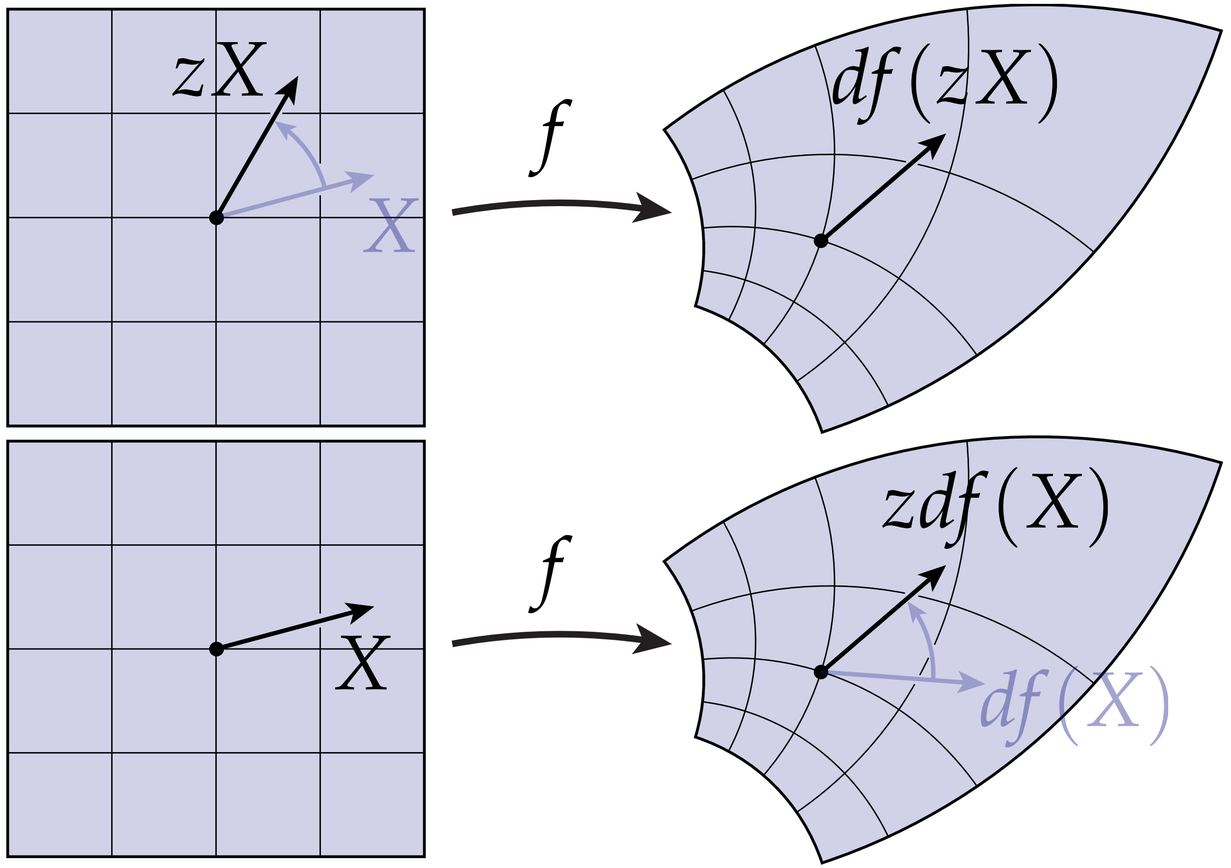
Consider a map $f : \mathbb{C} \to \mathbb{C}$

Then $f$ is conformal as long as

$$df(zX) = zdf(X)$$

for all tangent vectors $X$ and all complex numbers $z$.

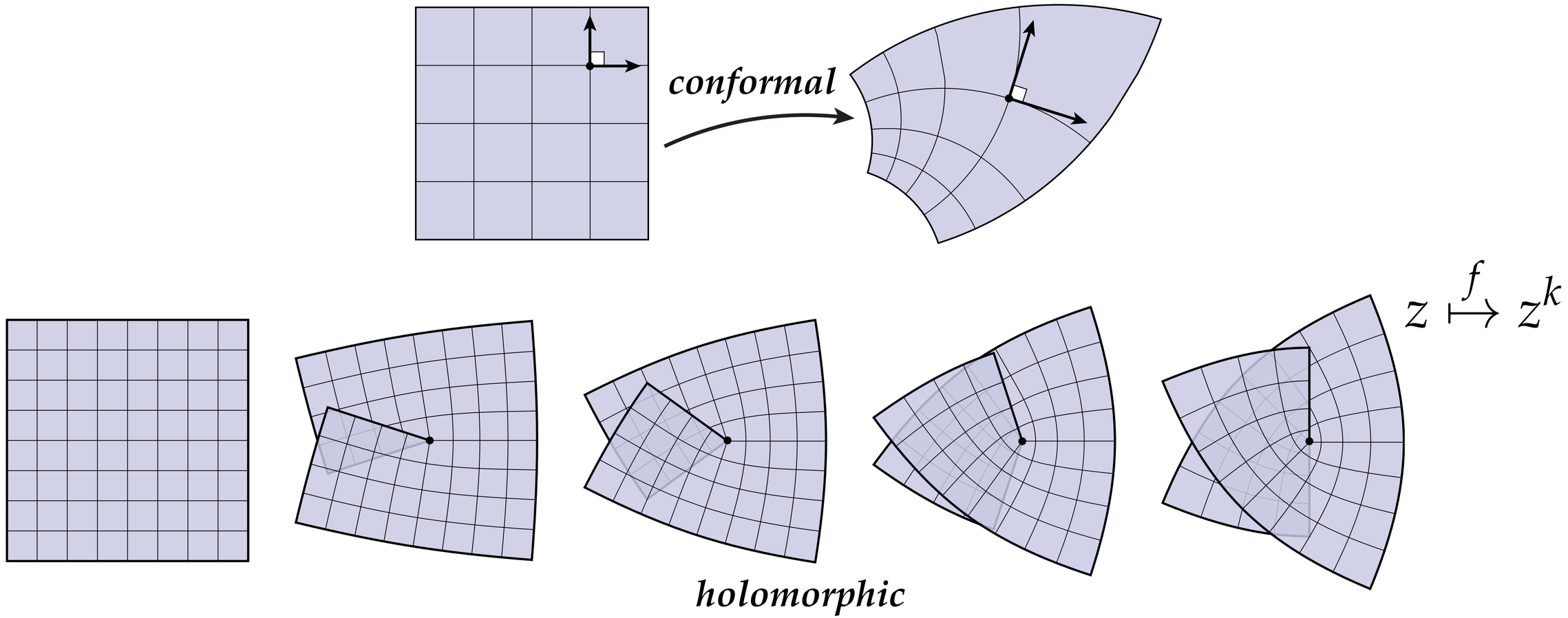
I.e., if it doesn't matter whether you rotate/scale *before* or *after* applying the map.




($df$ is "complex linear")

# *Holomorphic vs. Conformal*

- Important linguistic distinction: a *conformal map* is a holomorphic map that is “nondegenerate”, i.e., the differential is never zero.



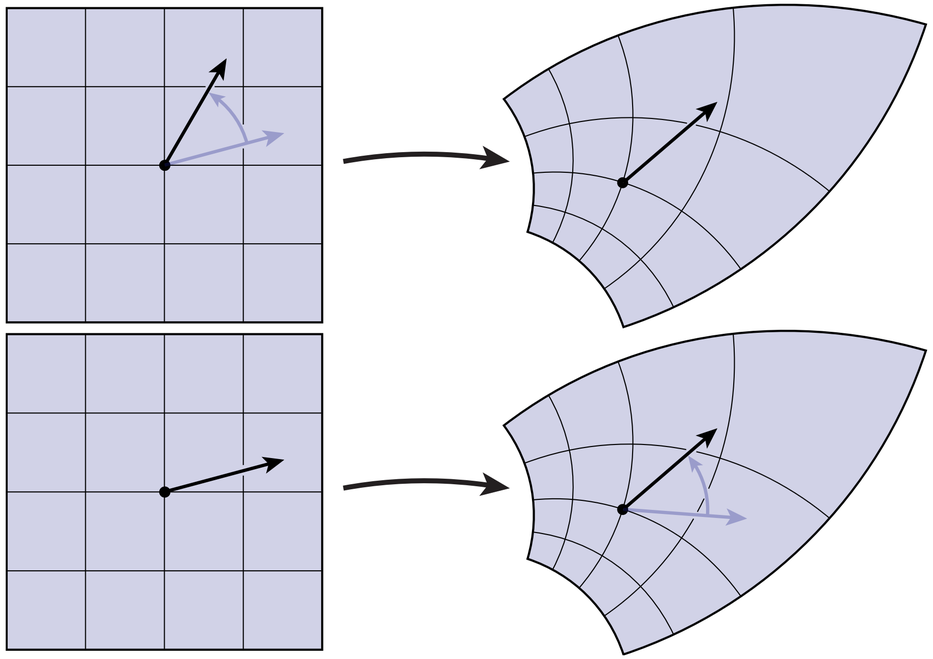
# *Cauchy-Riemann Equation*

Several equivalent ways of writing *Cauchy-Riemann equation*:

$$df(zX) = z\,df(X)$$

$$df(\imath X) = \imath\, df(X)$$

$$\star df = \imath\, df$$



$$\left.\begin{aligned} \frac{\partial f_1}{\partial x} &= \frac{\partial f_2}{\partial y} \\ \frac{\partial f_1}{\partial y} &= -\frac{\partial f_2}{\partial x} \end{aligned}\right\}$$

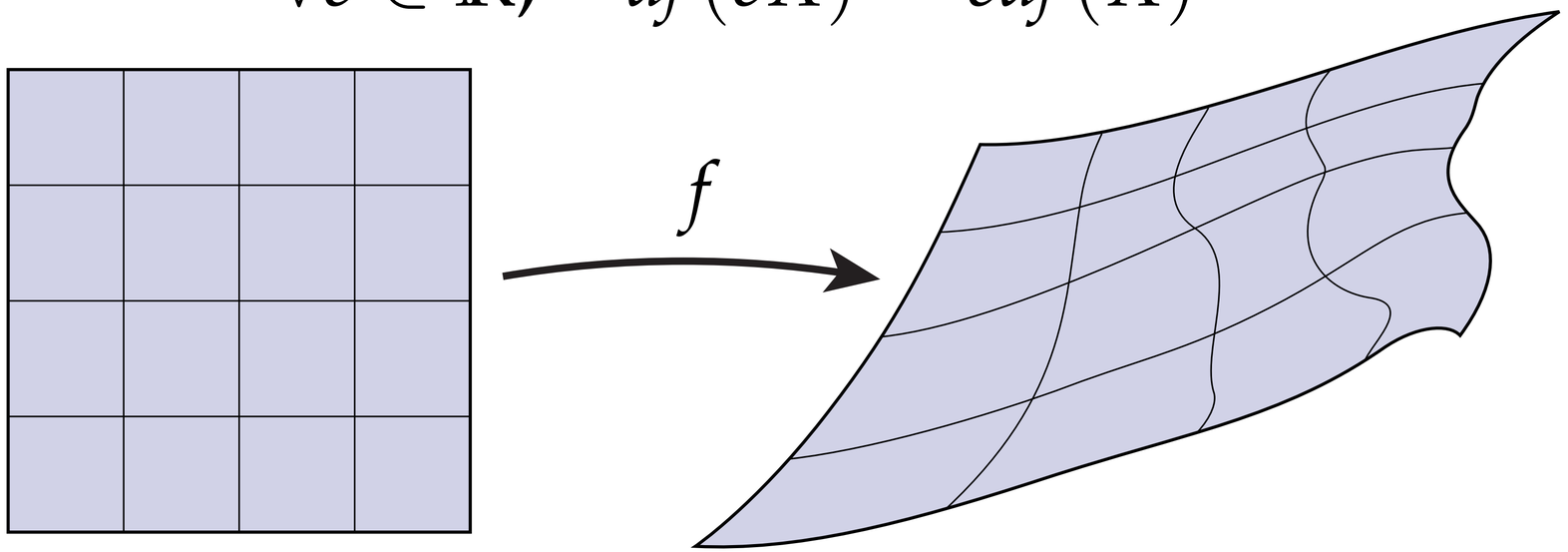
$$\bar{\partial} f = 0$$

All express the same **geometric** idea!

# *Aside: Real vs. Complex Linearity*

What if we just ask for *real* linearity?

$$\forall c \in \mathbb{R}, \quad df(cX) = cdf(X)$$




No angle preservation.

In fact, maps can be arbitrarily “ugly”. Why?

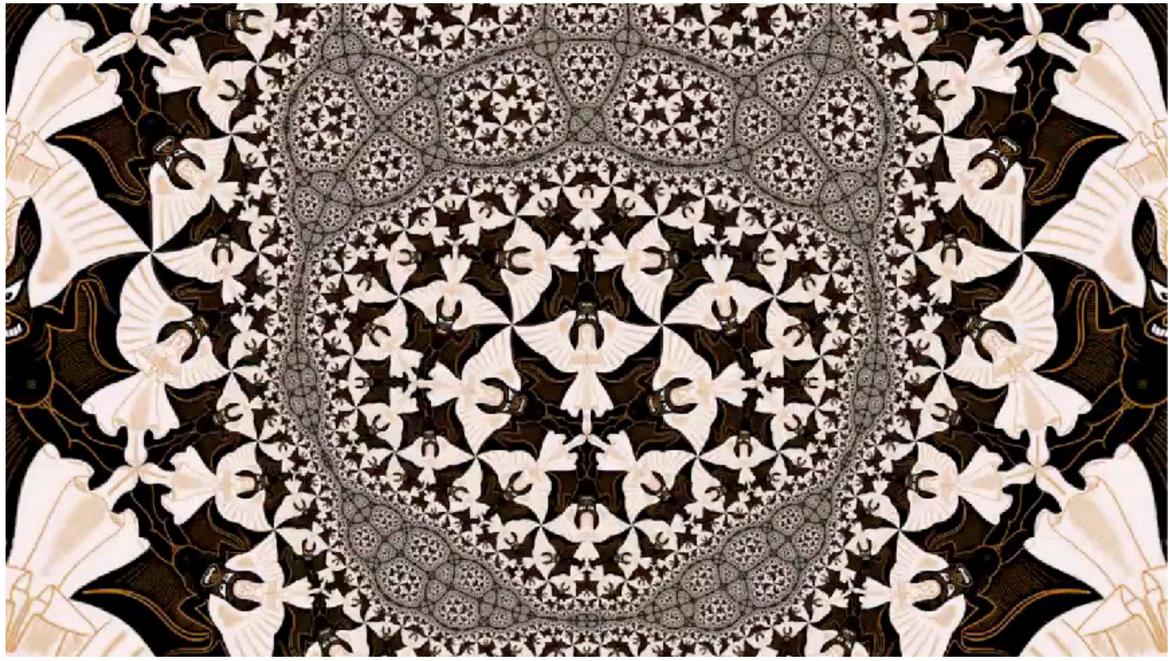
Because *any* differentiable $f$ trivially satisfies this property!

# *Example—Möbius Transformations (2D)*

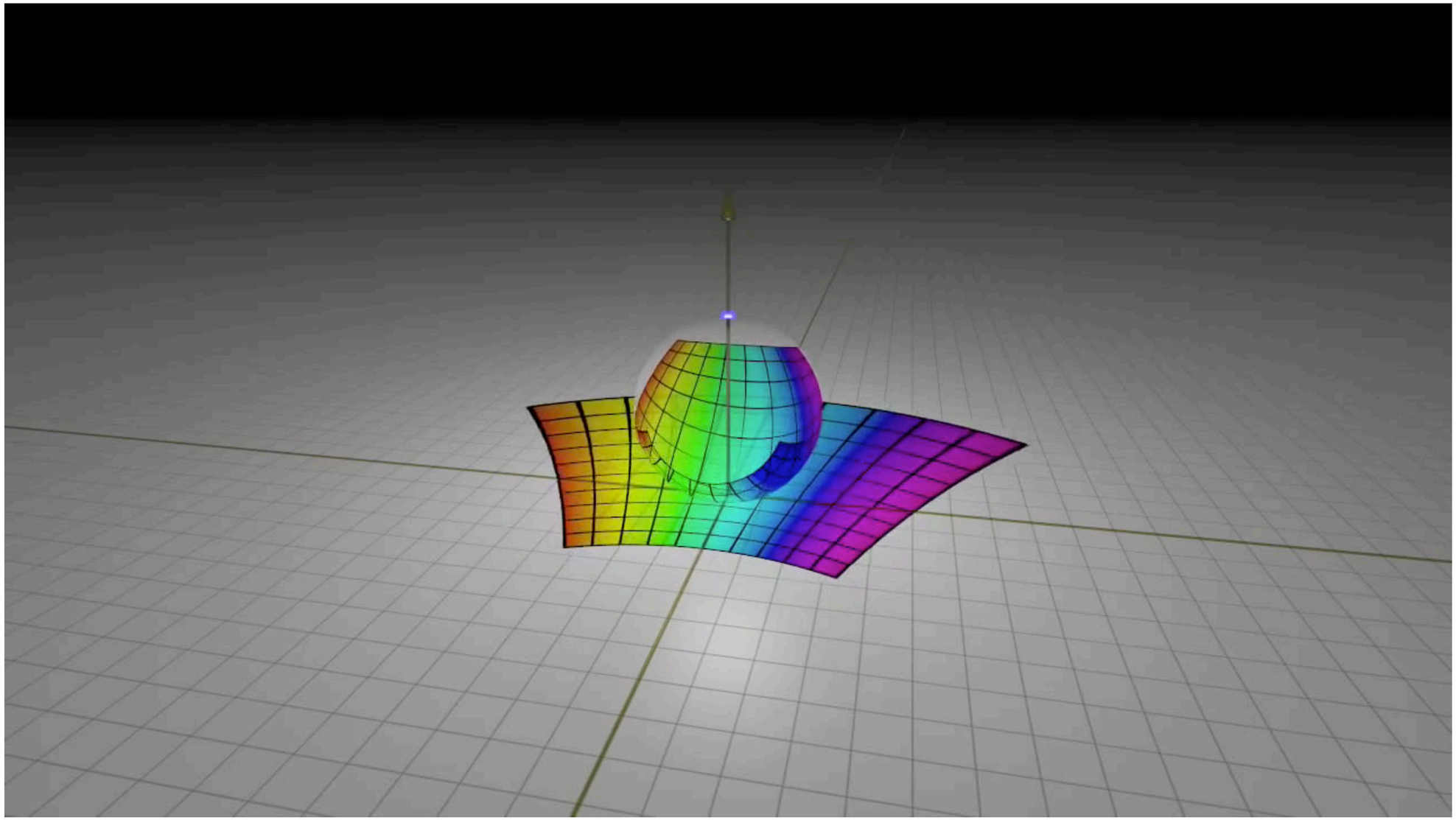
**Definition.** In 2D, a Möbius transformation is an orientation-preserving map taking circles to circles or lines, and lines to lines or circles. Algebraically, any Möbius transformation can be expressed as a map of the form

$$z \mapsto \frac{az + b}{cz + d}$$

for complex constants $ad \neq bc$.



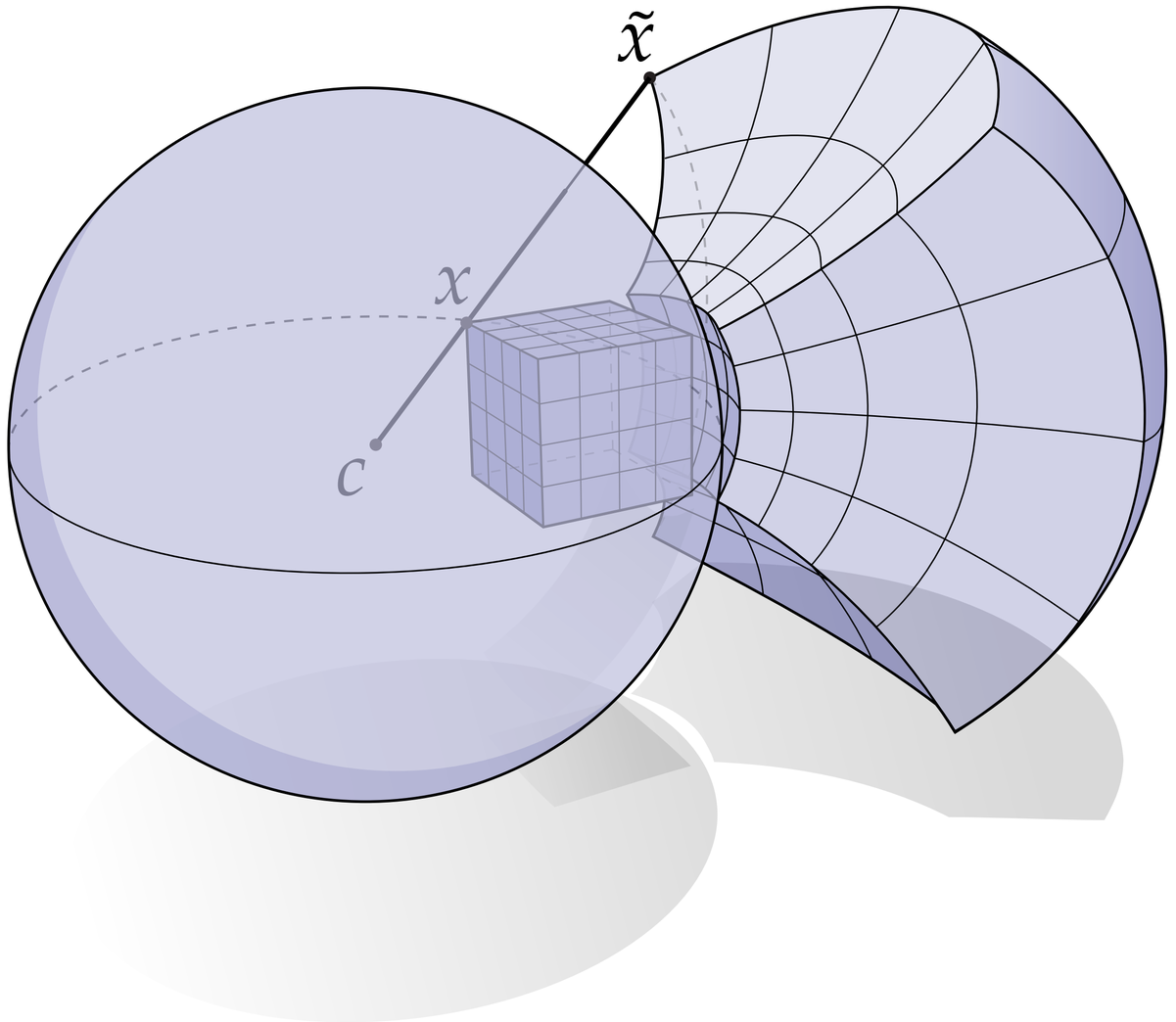
# *Möbius Transformations "Revealed"*



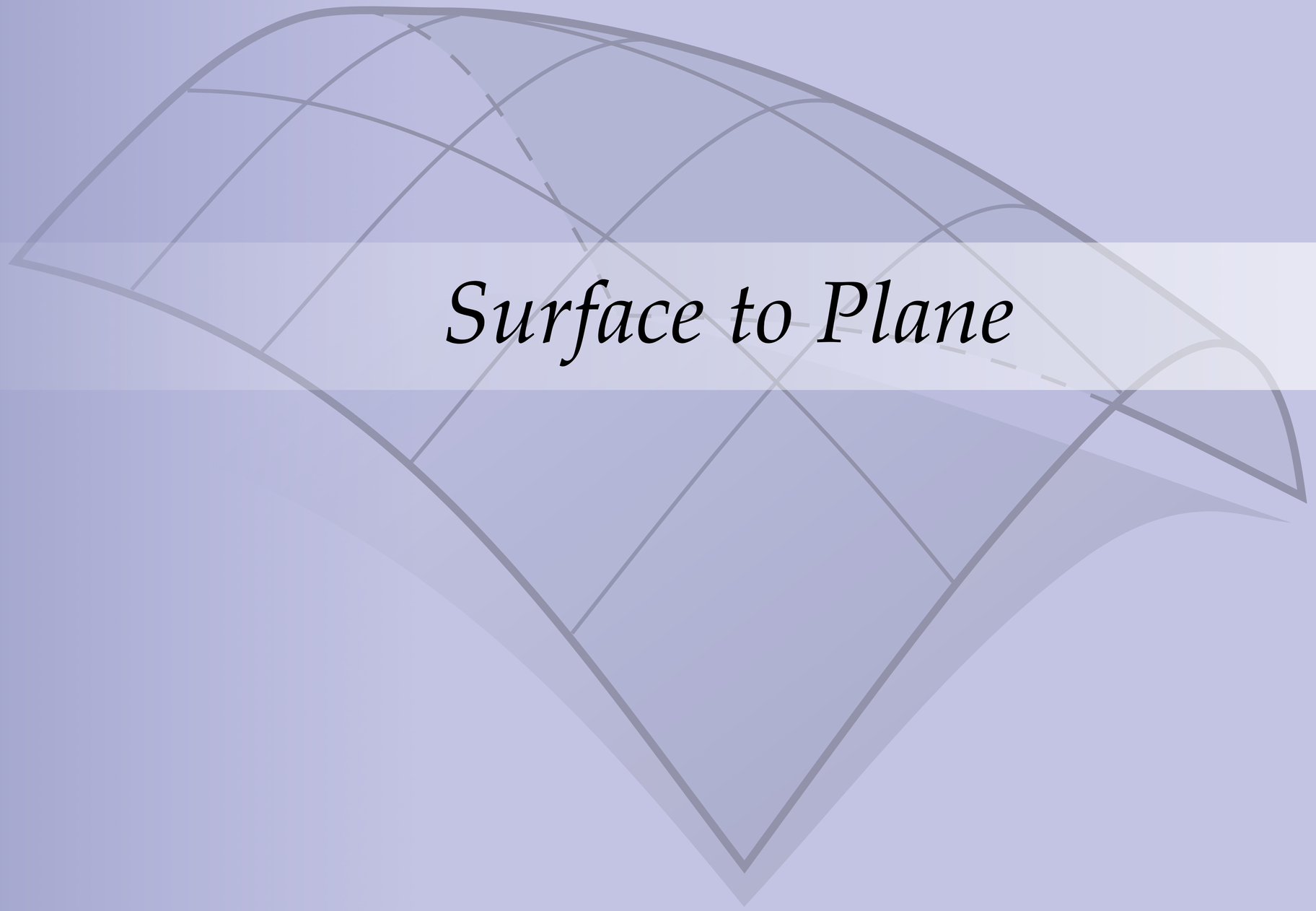
(Douglas Arnold and Jonathan Rogness) https://www.ima.umn.edu/~arnold/moebius/

# *Sphere Inversion (nD)*

$$x \mapsto \frac{x - c}{|x - c|^2}$$



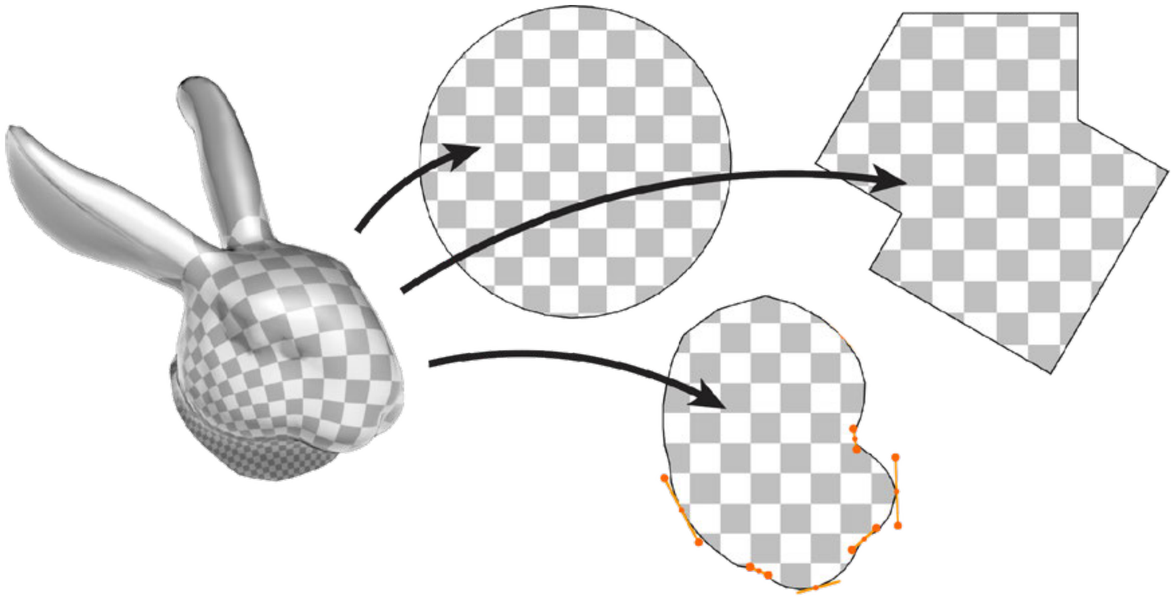

(**Note:** Reverses orientation—*anticonformal* rather than conformal)

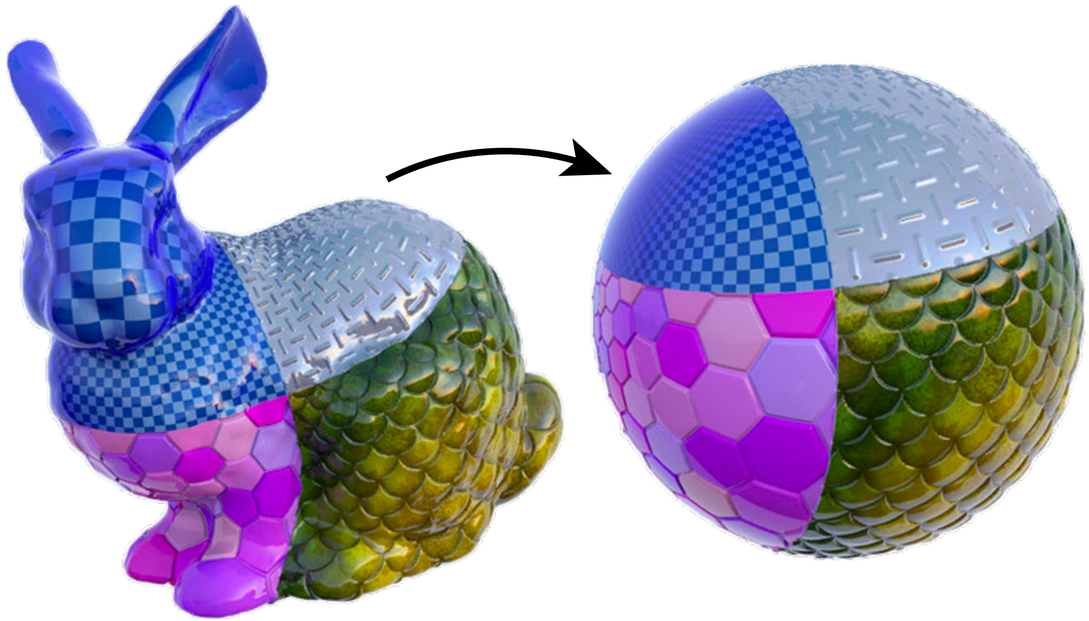


# *Surface to Plane*

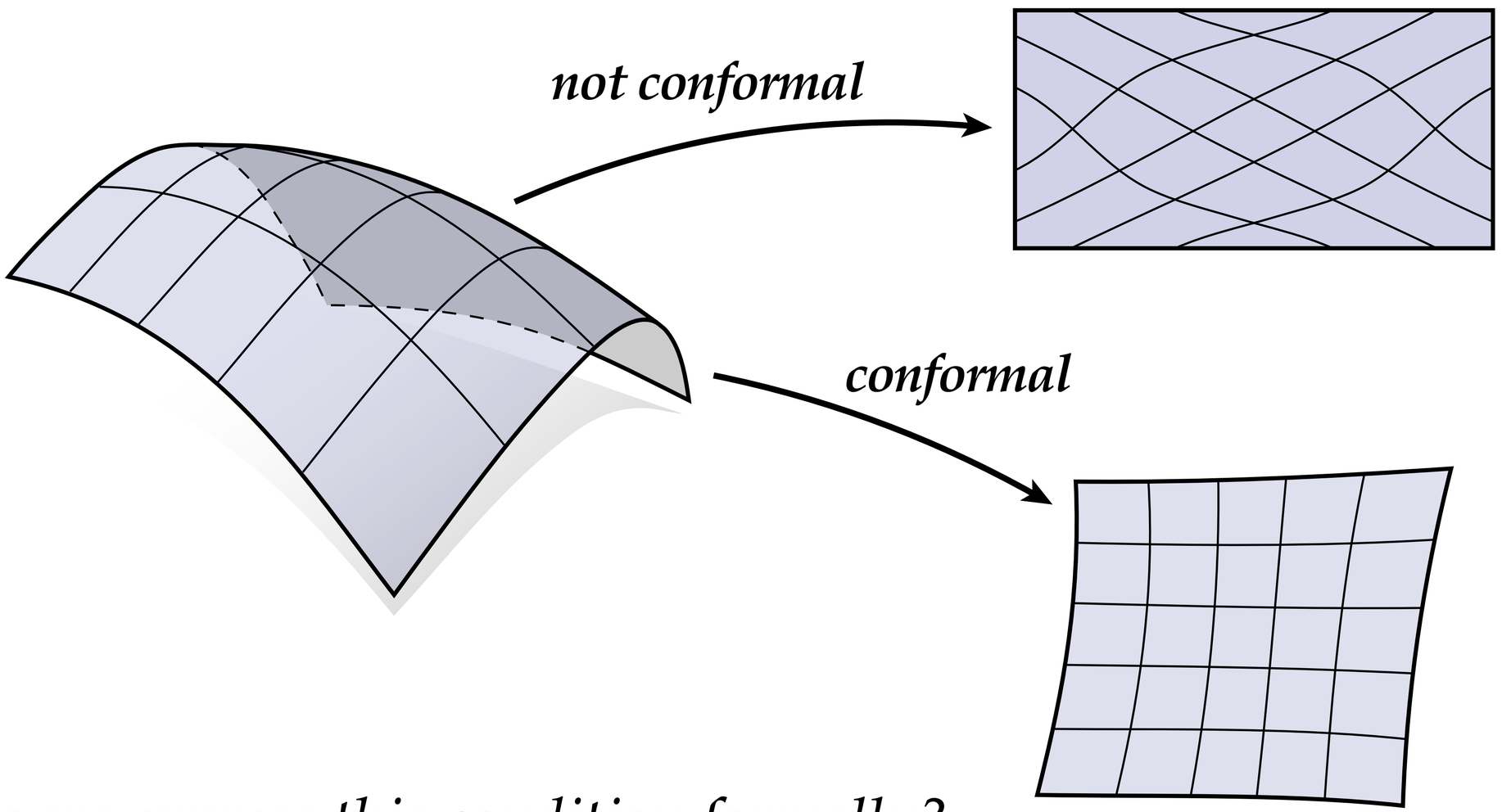
# *Surface to Plane*

- Map curved surface to 2D plane (“conformal flattening”)
- Surface does not necessarily sit in 3D
- Slight generalization: target curvature is constant but *nonzero* (*e.g.*, sphere)
- Many different equations: Cauchy-Riemann, Yamabe, …





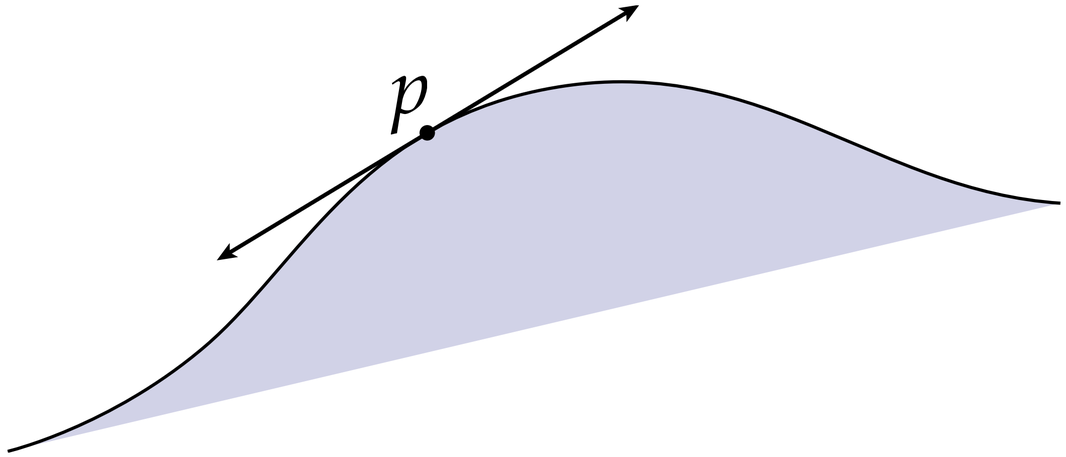
# *Conformal Maps on Surfaces—Visualized*

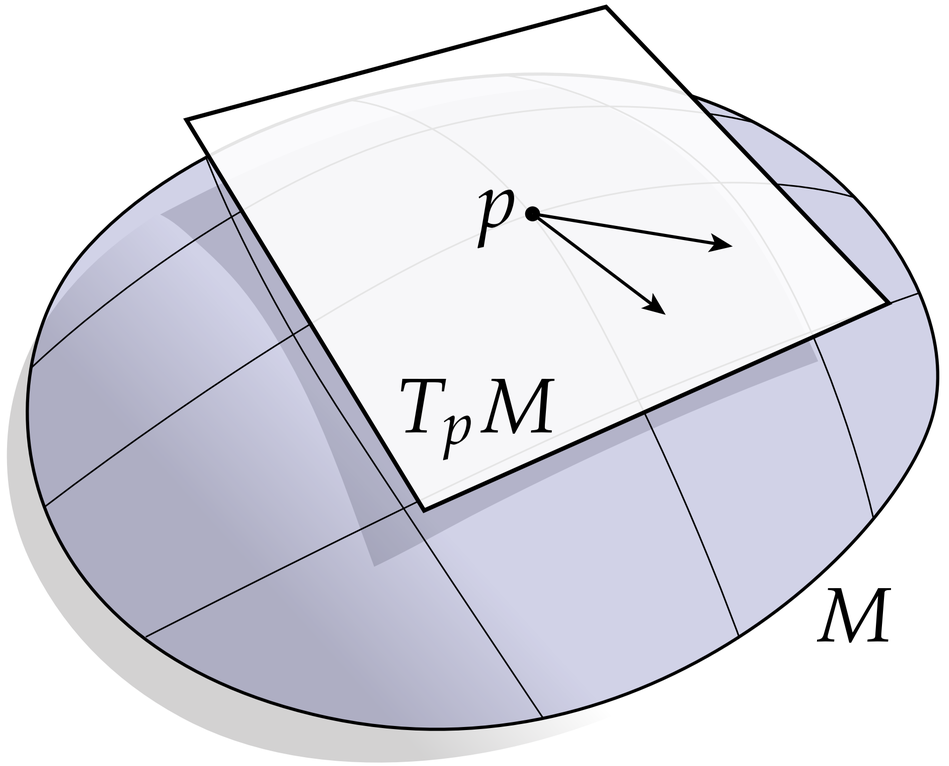



*How do we express this condition formally?*

# *Tangent Plane*

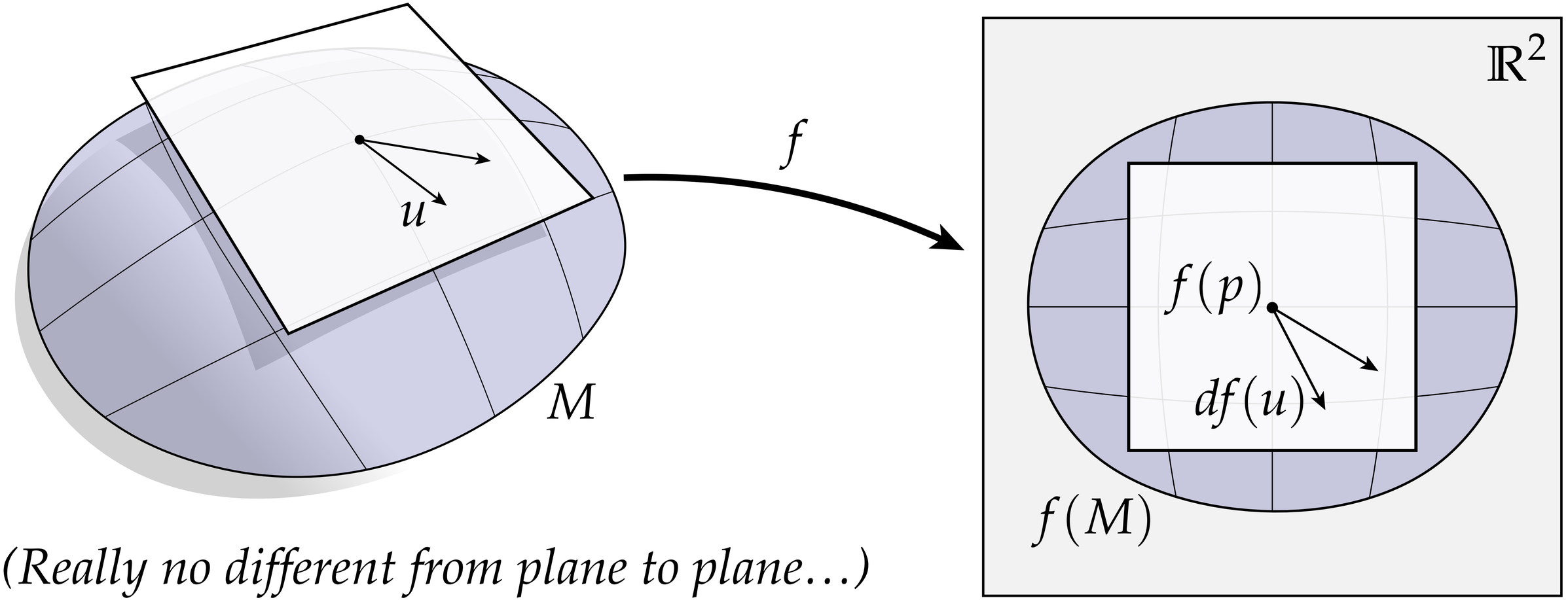
- *Tangent vectors* are those that "graze" the surface
- *Tangent plane* is all the tangent vectors at a given point



# *Differential of a Map from Surface to Plane*

- Consider a map taking each point of a surface to a point in the plane
- *Differential* says how tangent vectors get "stretched out" under this map



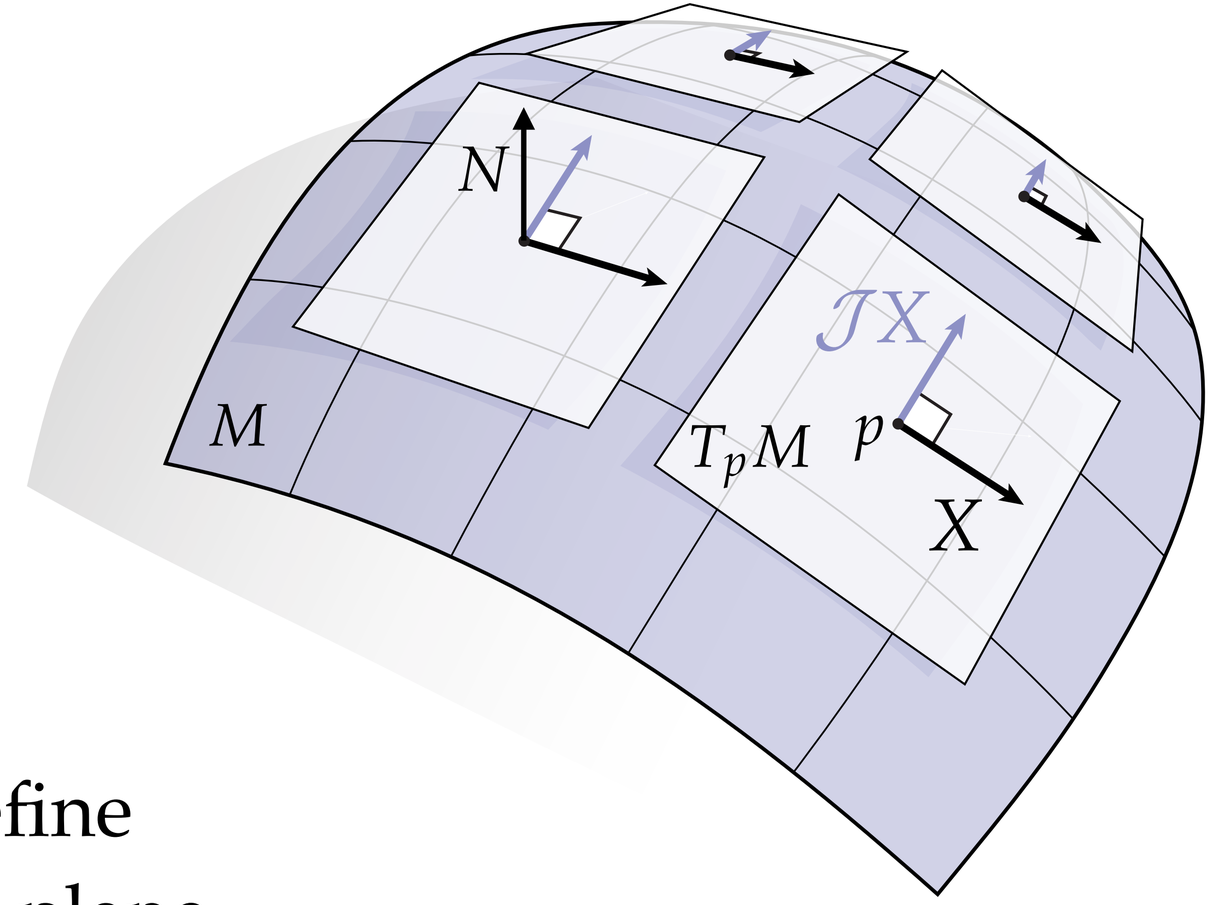

*(Really no different from plane to plane…)*

# *Complex Structure*

- *Complex structure J* rotates vectors in each tangent plane by 90 degrees
- Analogous to complex unit $i$
- E.g., $\mathcal{J} \circ \mathcal{J} = -\mathrm{id}$
- For a surface in $\mathbb{R}^3$:

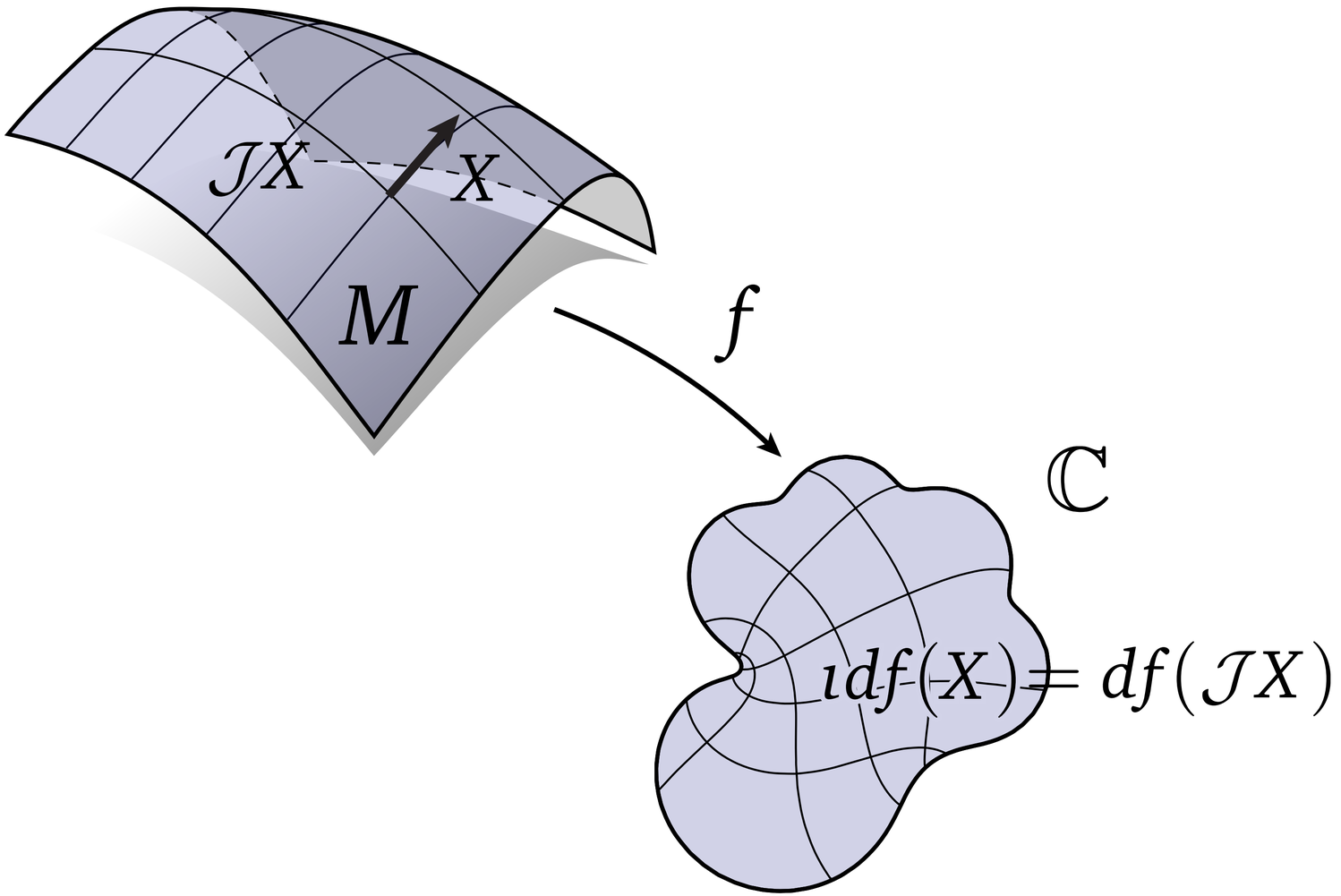
$$\mathcal{J}X = N \times X$$

(where $N$ is unit normal)




**Motivation:** will enable us to define conformal maps from surface to plane.

# *Holomorphic Maps from a Surface to the Plane*



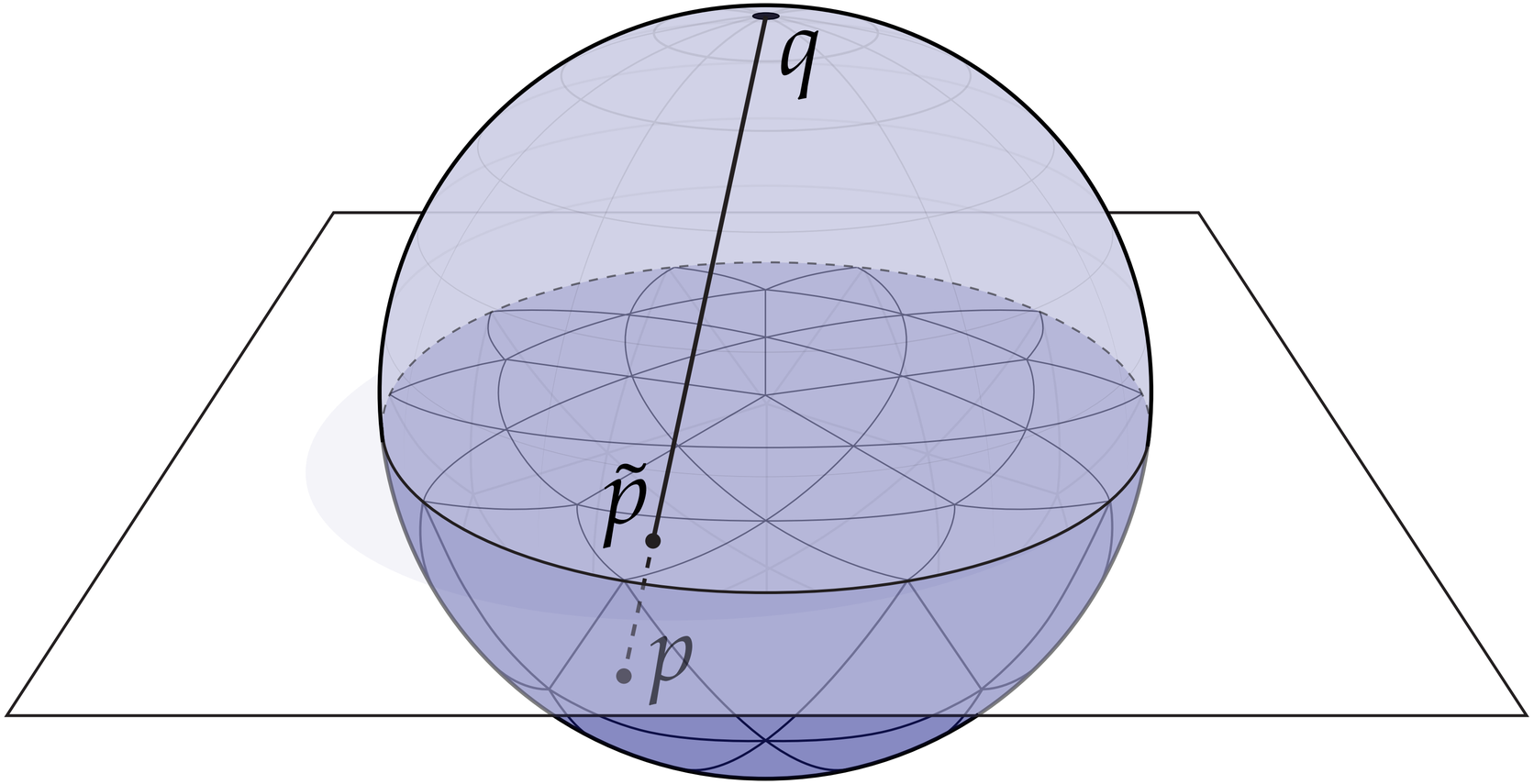

**Plane to plane:**

$$df(\imath X) = \imath df(X)$$

**Surface to plane:**

$$df(\mathcal{J}X) = \imath df(X)$$

# *Example—Stereographic Projection*

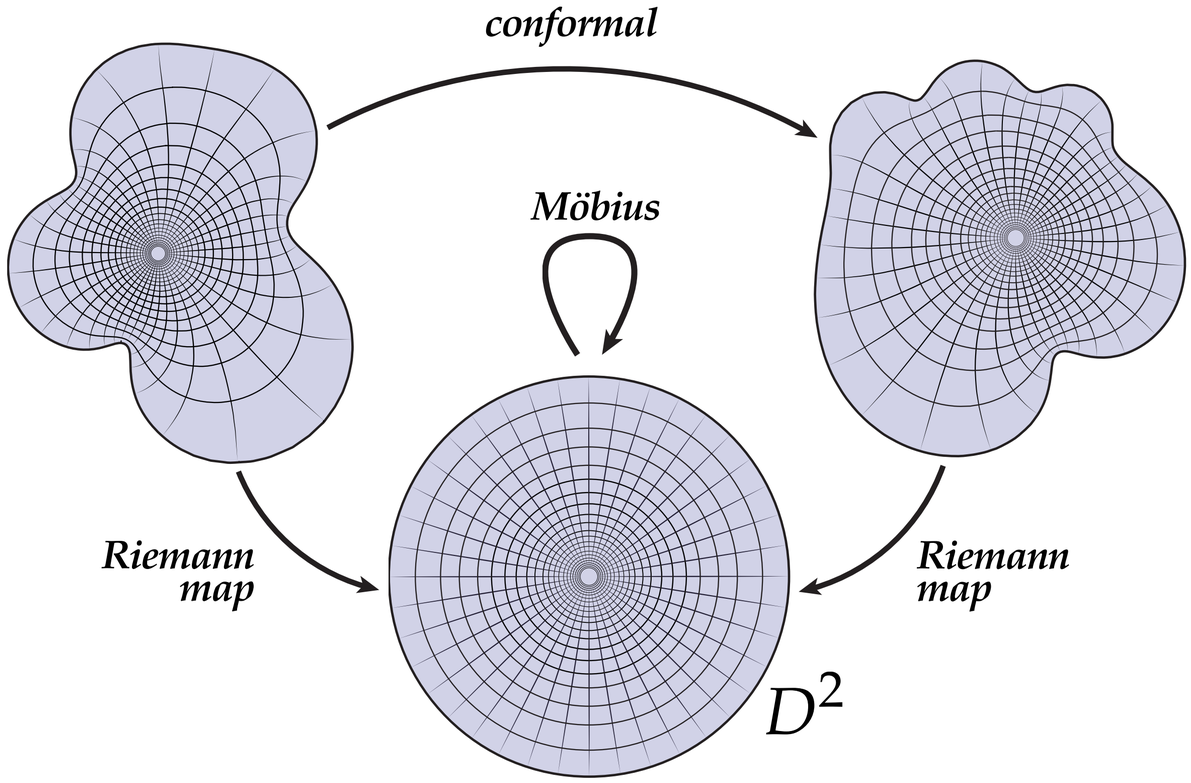



**How?** Don't memorize some formula—*derive it yourself!*

E.g., What's the equation for a sphere? What's the equation for a ray?

# *Riemann Mapping Theorem*

**Theorem** (Riemann)**.** Any nonempty simply-connected open proper subset of $\mathbb{C}$ can be conformally mapped to the unit open disk $D^2 := \{z \in \mathbb{C} : |z| < 1\}$.




**Fact.** The only conformal maps from $D^2$ to $D^2$ are Möbius transformations of the form

$$z \mapsto e^{\imath\theta} \frac{z - a}{1 - \bar{a}z}$$

where $a \in D^2$ and $\theta \in S^1$ (three degrees of freedom: inversion center and rotation).

# *Riemannian Metric*

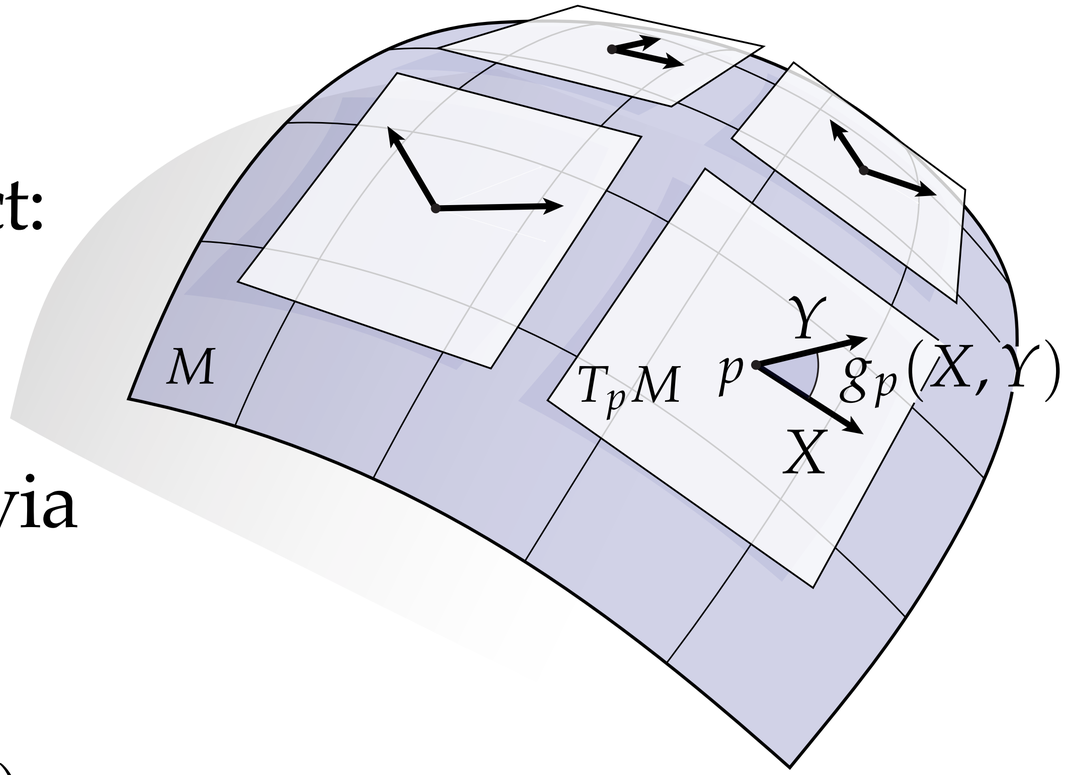
- Can also understand conformal maps in terms of *Riemannian metric*
- Riemannian metric $g$ is simply inner product in each tangent space
- Allows us to measure length, angle, etc.
- E.g., Euclidean metric is just dot product:

$$g_{\mathbb{R}^n}(X, Y) := \sum_i X_i Y_i$$

- In general, length and angle recovered via

$$|X| := \sqrt{g(X, X)}$$

$$\angle(X, Y) := \arccos(g(X, Y)/|X||Y|)$$

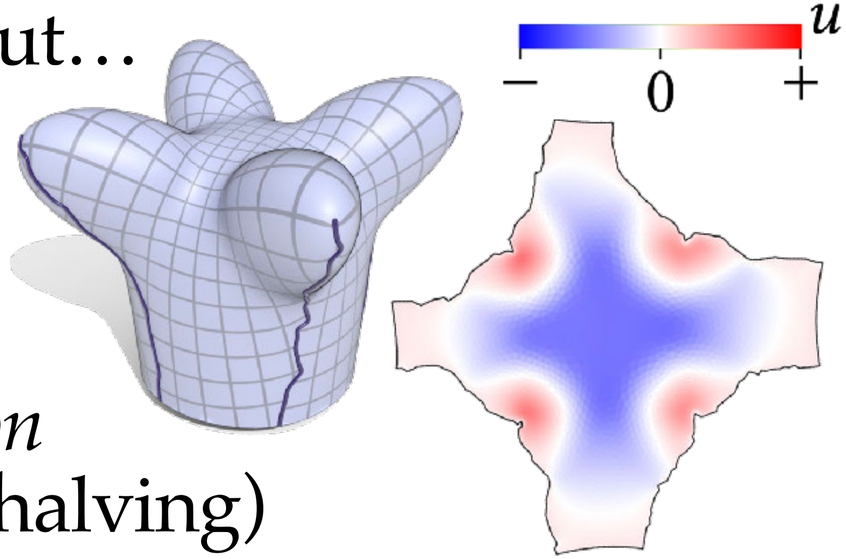


# *Conformally Equivalent Metrics*

- Two metrics are *conformally equivalent* if they are related by a positive **conformal scale factor** at each point $p$:

$$\boxed{\tilde{g}_p = e^{2u(p)} g_p} \qquad u : M \to \mathbb{R}$$

- Why write scaling as $e^{2u}$? Initially mysterious, but…
  - ensures scaling is always *positive*
  - factor $e^u$ gives length scaling
  - more natural way of talking about *area distortion* (e.g., doubling in scale "costs" just as much as halving)

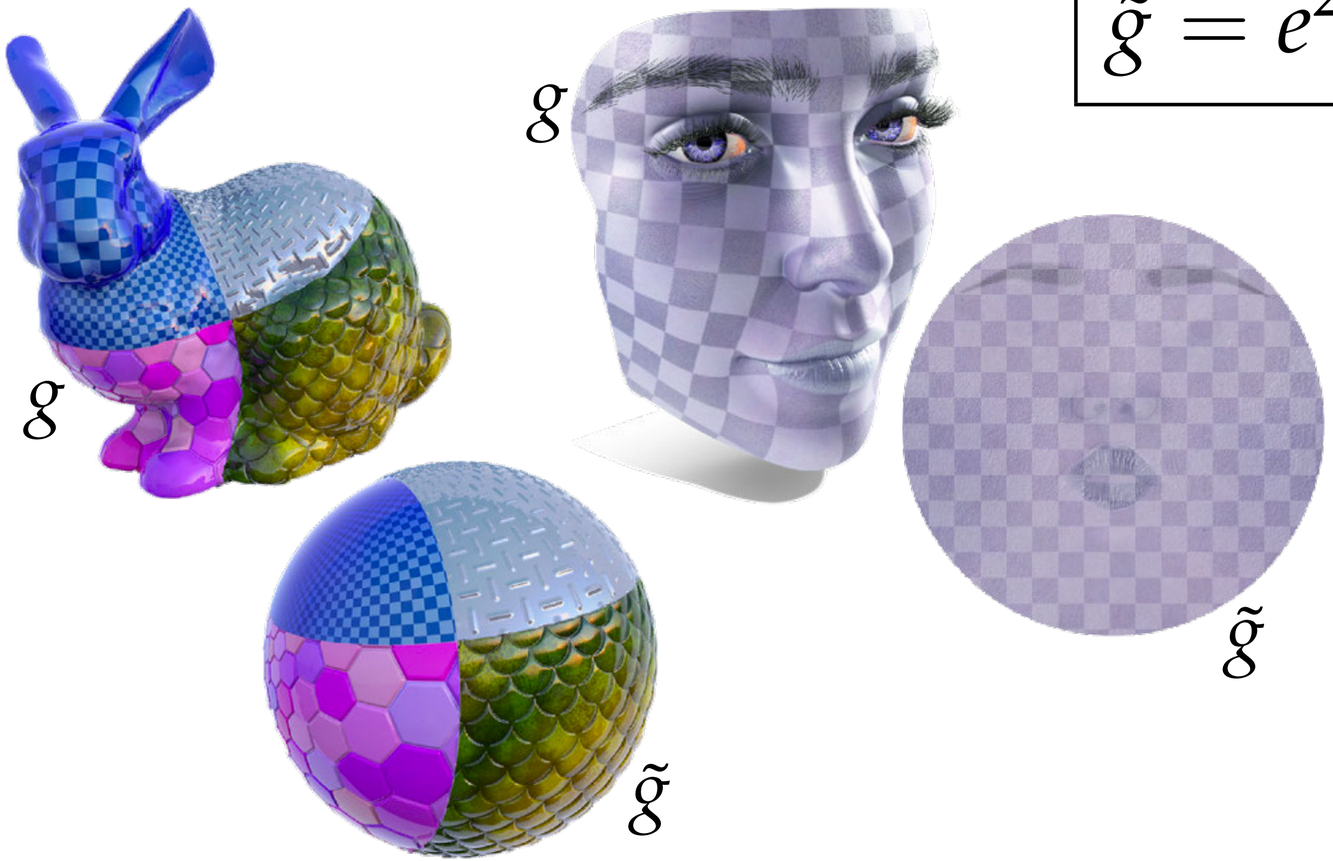


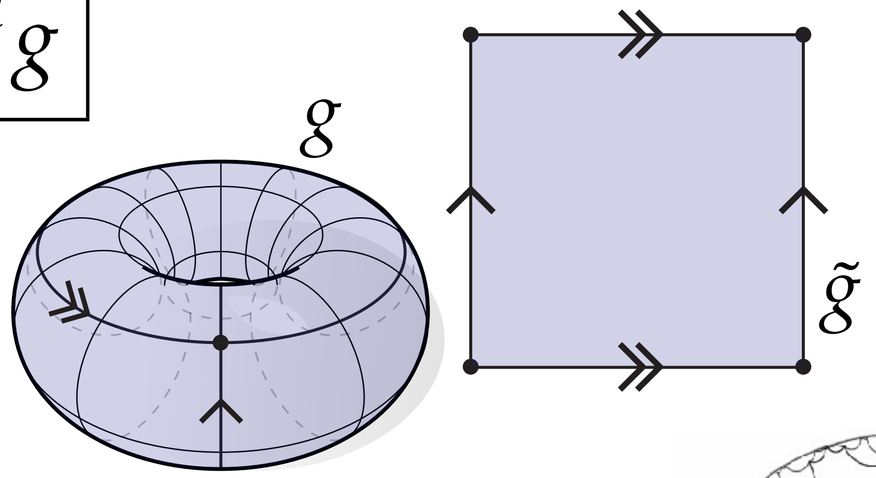

**Q:** Does this transformation preserve *angles*?

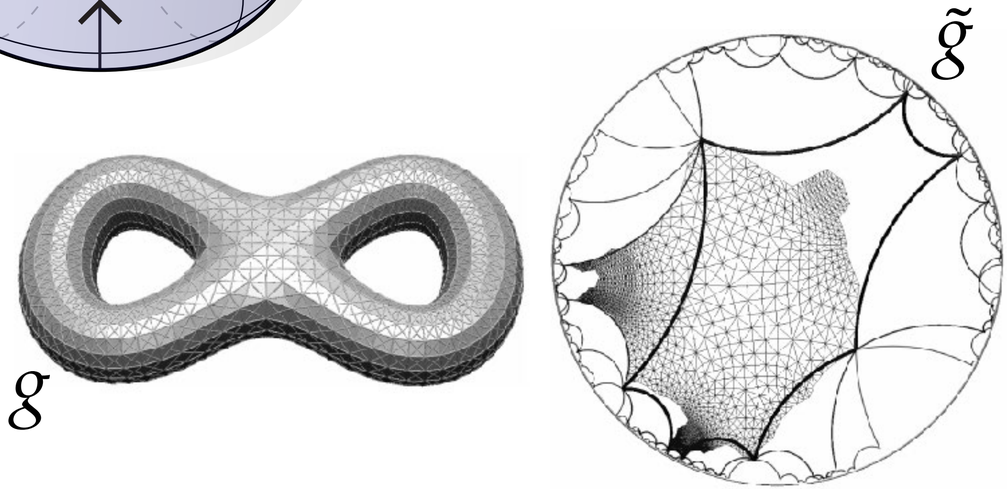
# *Uniformization Theorem*

- Roughly speaking, Riemannian metric on any surface is conformally equivalent to one with *constant curvature* (flat, spherical, hyperbolic).

$$\tilde{g} = e^{2u} g$$

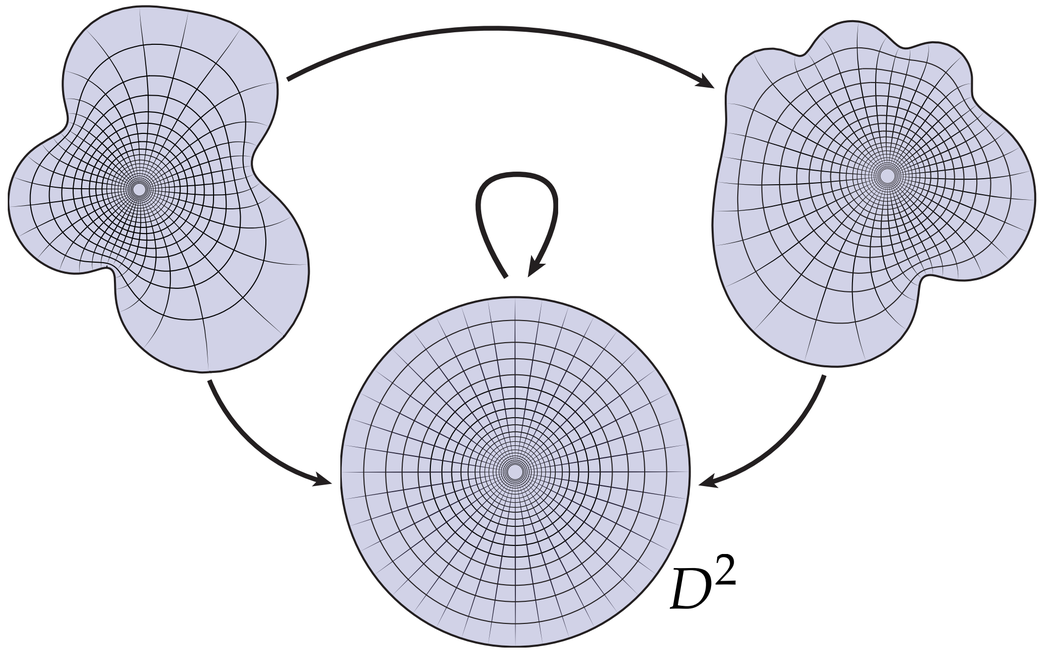


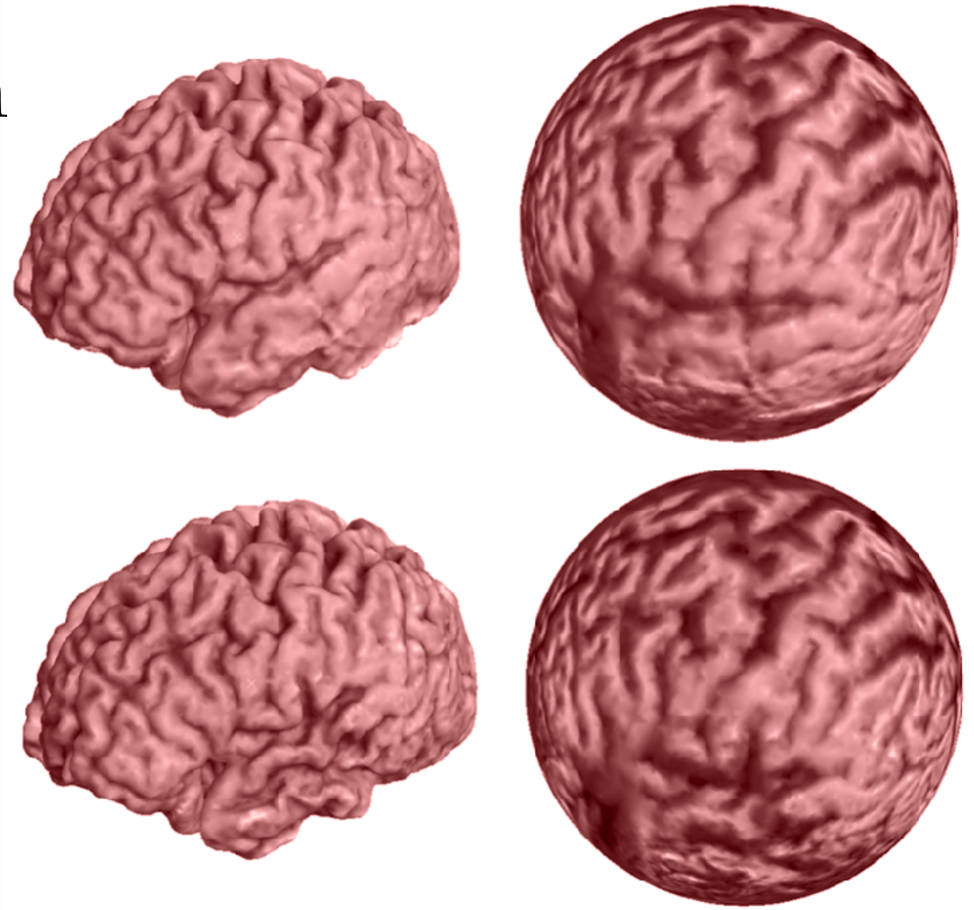



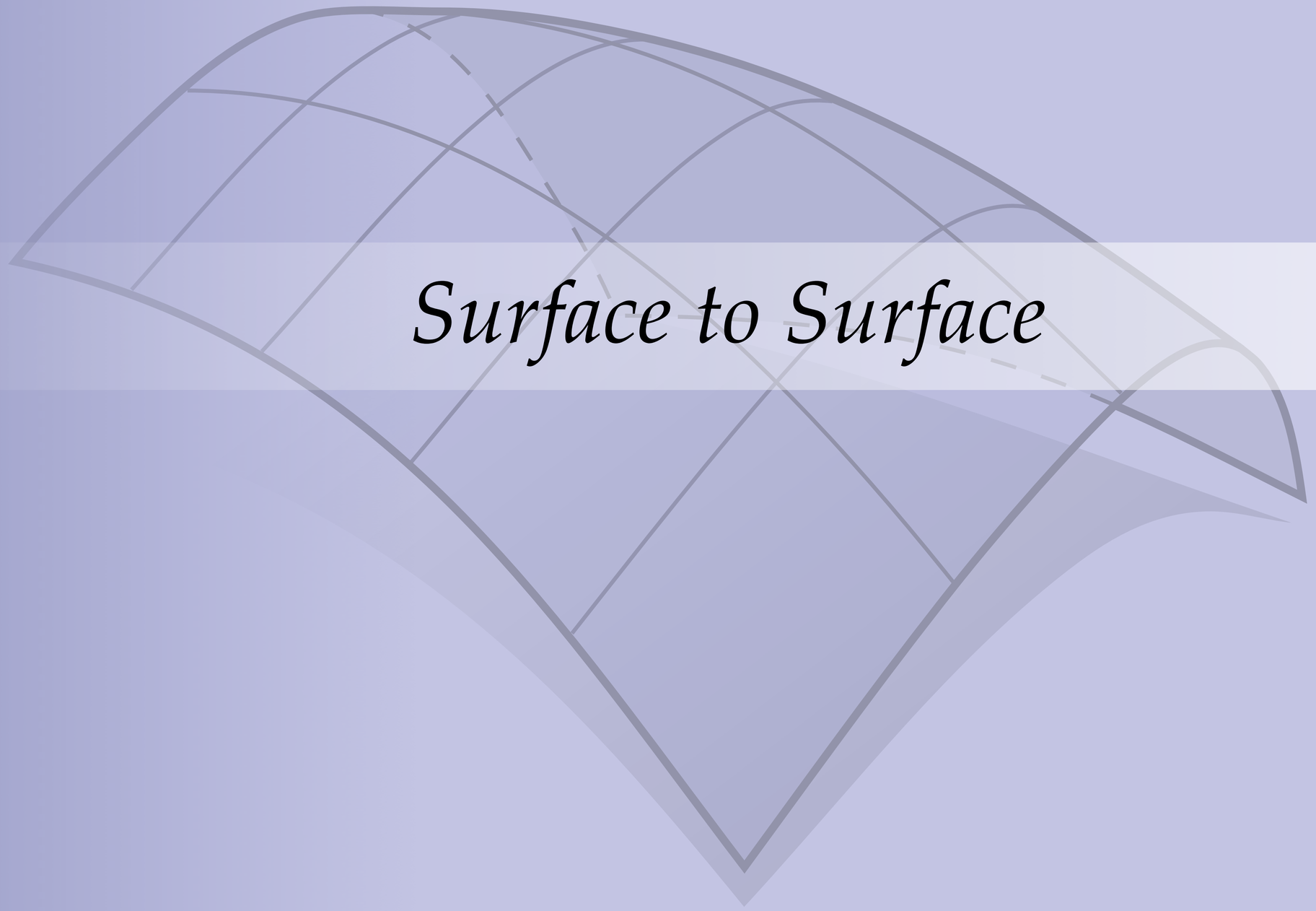



# *Why is Uniformization Useful?*

- Provides canonical domain for solving equations, comparing data, cross-parameterization, etc.
- *Careful:* still have a few degrees of freedom (e.g., Möbius transformations)





# *Surface to Surface*

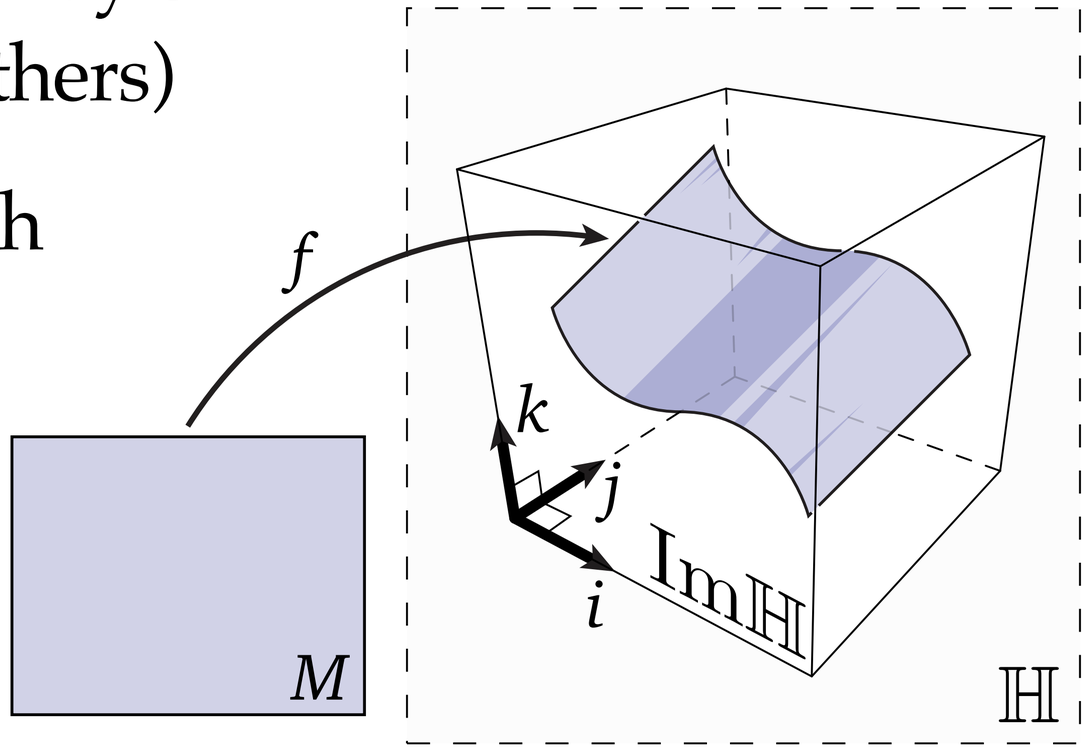
# *Surface to Surface*

- Conformal deformations of surfaces embedded in space
- Both surfaces can have arbitrary curvature (not just sphere, disk, *etc.*)
- Opens door to much broader geometry processing applications
- Very recent theory & algorithms (~1996/2011)
- Key equation: *time-independent Dirac equation*

Won't say too much today… see https://youtu.be/UQC_emOPVK8

# *Geometry in the Quaternions*

- Just as complex numbers helped with 2D transformations, *quaternions* provide natural language for 3D transformations
- Recent use of quaternions as alternative way of analyzing surfaces (Pedit, Pinkall, and others)
- Basic idea: points (a,b,c) get replaced with *imaginary* quaternions a$i$ + b$j$ + c$k$
- Surface is likewise an imaginary map $f$



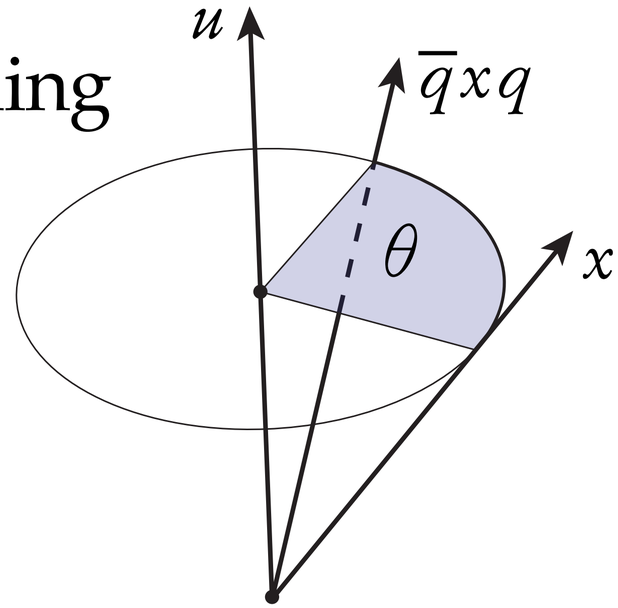
# *Stretch Rotations*

- How do we express rotation using quaternions?
- Similar to complex case, can rotate a vector $x$ using a unit quaternion $q$:

$$\tilde{x} = \bar{q} x q$$

**rotated** **original**

- If $q$ has non-unit magnitude, we get a rotation and scaling
- Should remind you of conformal map:
  *scaling* & *rotation* (but no shear)

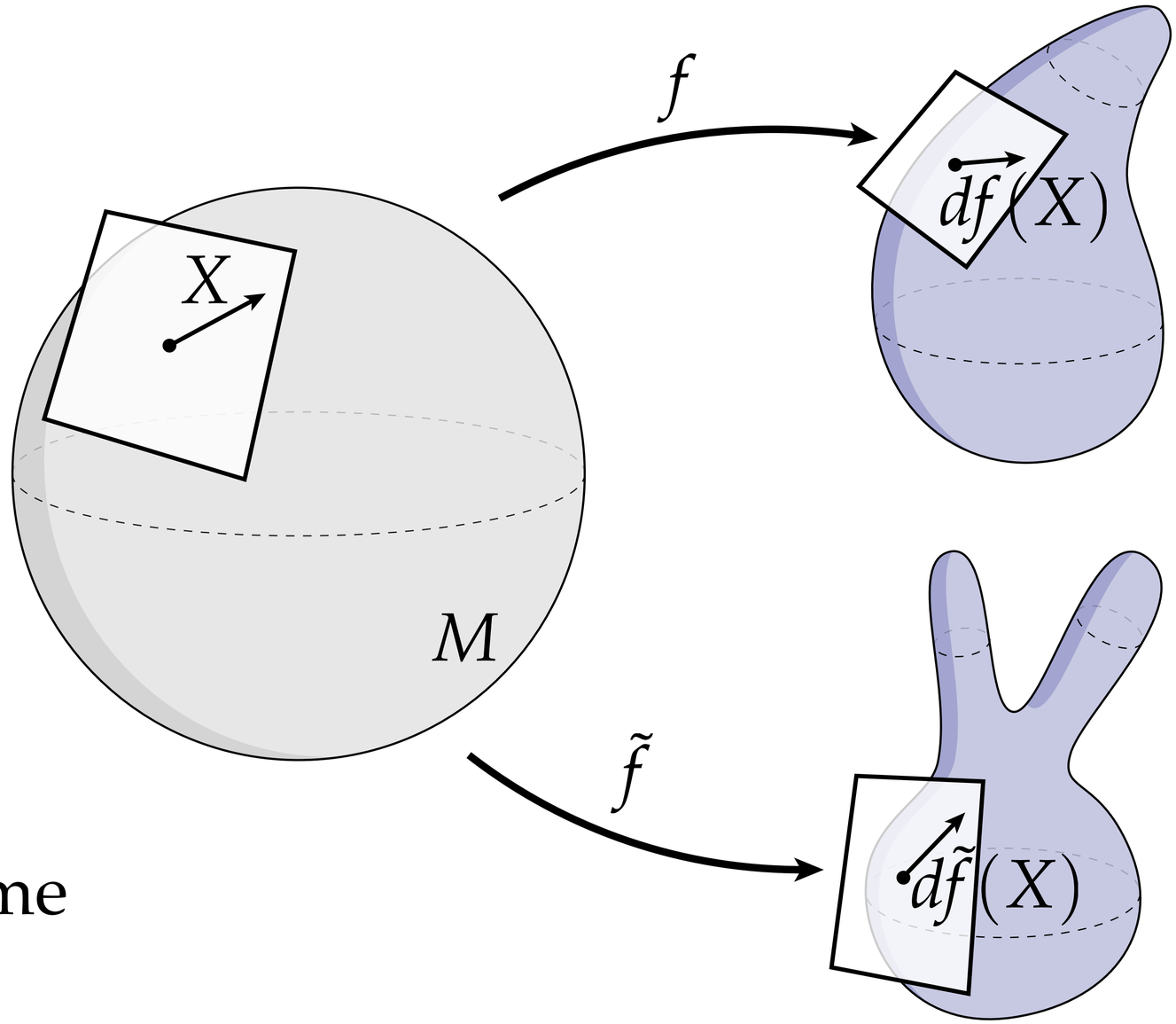


# *Spin Equivalence*

- From here, not hard to express conformal deformation of surfaces
- Two surfaces $f_0, f$ are *spin equivalent* if their tangent planes are related by a pure scaling and rotation at each point:

$$d\tilde{f}(X) = \bar{\psi}\, df(X)\psi$$

for all tangent vectors $X$ and some stretch rotation $\psi : M \to \mathbb{H}$



# *Dirac Equation*

- From here, one can derive the fundamental equation for conformal surface deformations, a *time-independent Dirac equation*

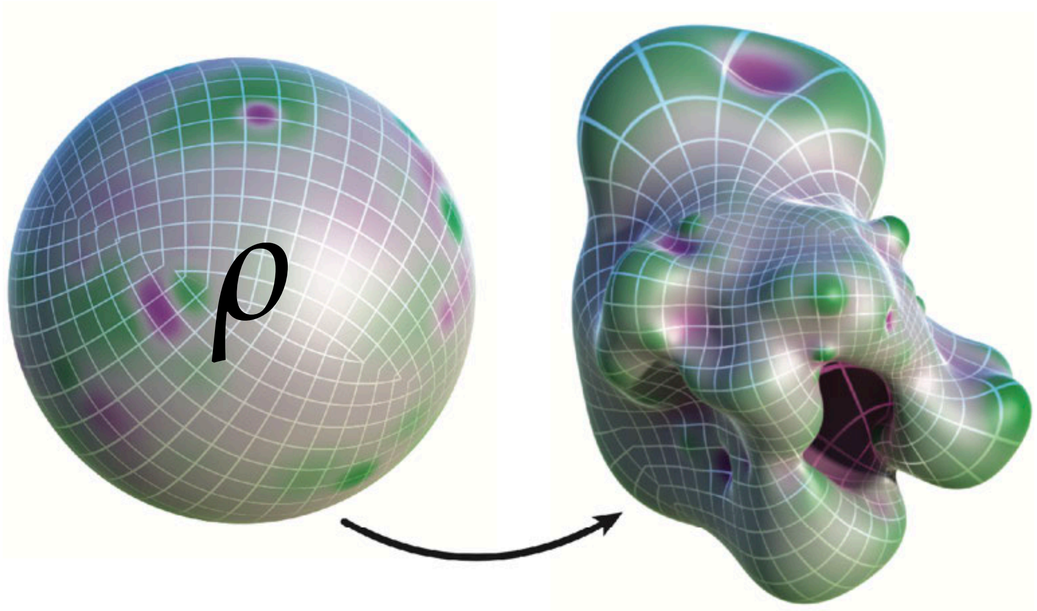
**change in curvature**

$$(D - \rho)\psi = 0$$

**quaternionic Dirac operator**

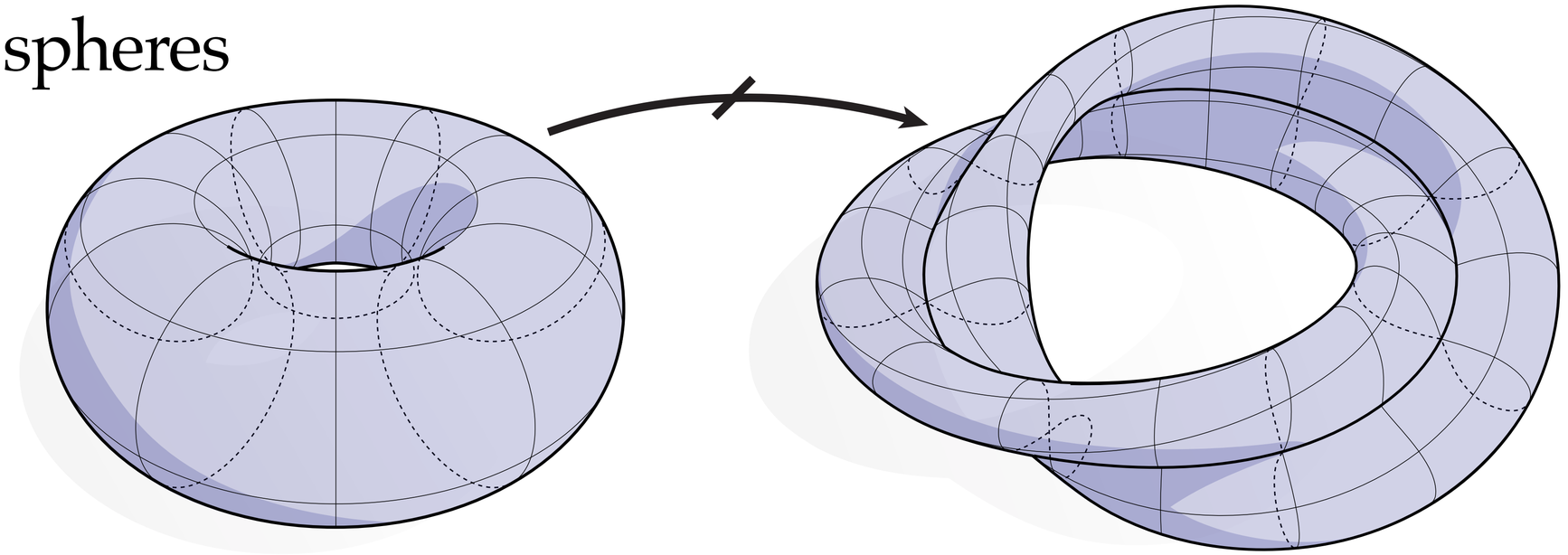
**stretch rotation**

$$D\psi := -\frac{df \wedge d\psi}{|df|^2}$$



# *Spin vs. Conformal Equivalence*

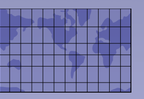
- Two surfaces that are spin equivalent are also conformally equivalent: tangent vectors just get *rotated* and *scaled*! (no shearing)
- Are conformally equivalent surfaces always spin equivalent?
  - **No** in general, *e.g.*, tori that are not *regularly homotopic* (below)
  - **Yes** for topological spheres

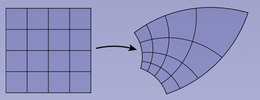


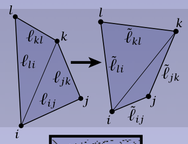
# *Why Not Just Optimize Angles?*

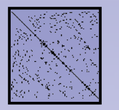
- Forget the mathematics—why not just optimize mesh to preserve angles?
- As discussed before, angle preservation is *too rigid!*
- E.g., convex surface *uniquely* determined by angles (up to rigid motion)

**original** **deformed** **optimize angles**

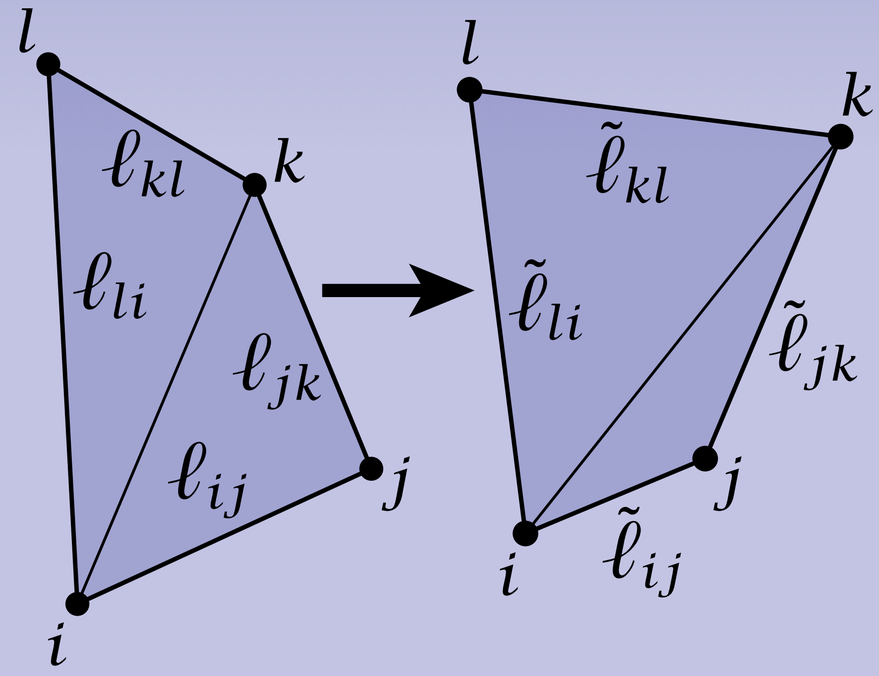

PART I: OVERVIEW


PART II: SMOOTH THEORY


PART III: DISCRETIZATION


PART IV: ALGORITHMS

# PART III: DISCRETIZATION

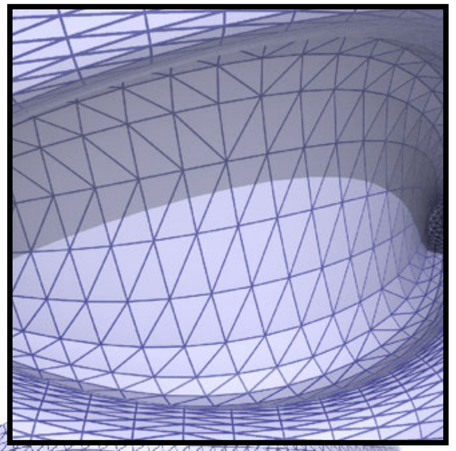



# CONFORMAL GEOMETRY PROCESSING

**Keenan Crane • CMU 15-458/858B • Fall 2017**

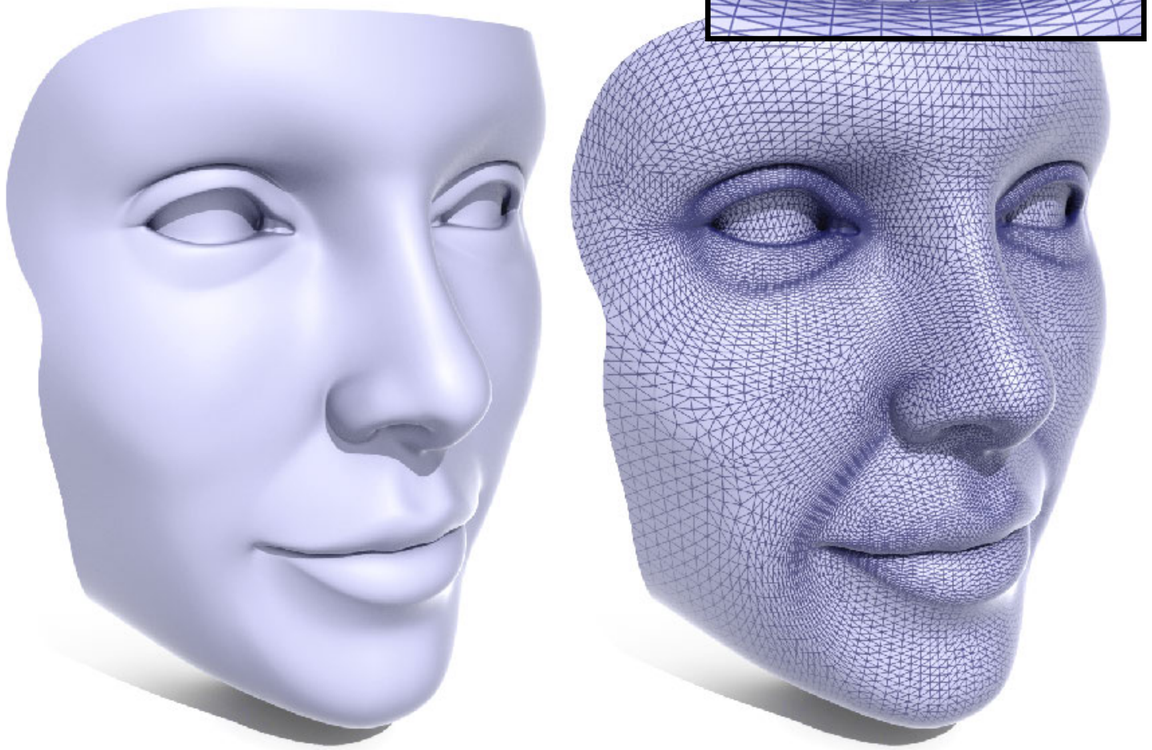
# *Surfaces as Triangle Meshes*

- For computation, need *finitely many* degrees of freedom
- Many ways to discretize—common choice is *triangle mesh*
  - No restrictions on geometry (height function, etc.)
  - Any polygon can be triangulated
  - Simple formulas (e.g., per triangle)
  - Efficient computation (sparse)



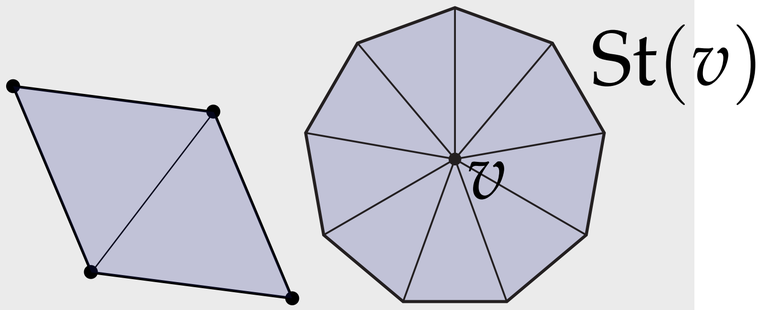
$$K = (V, E, F)$$

**mesh** **vertices** **edges** **faces**

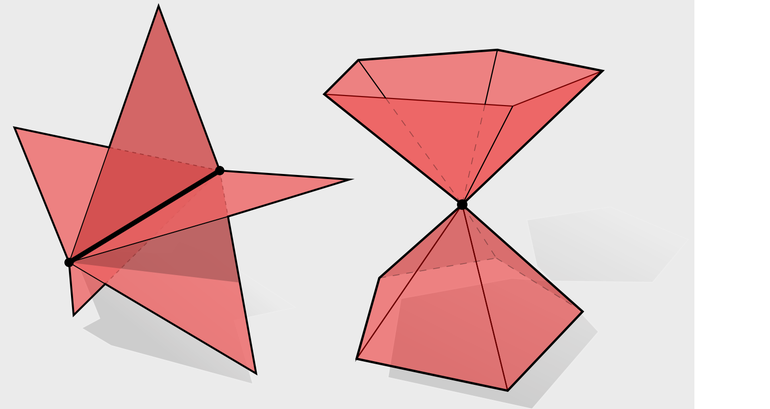


# *Manifold Triangle Mesh*

- Images: assume every pixel has four neighbors (keeps things simple!)
- Likewise, assume meshes are *manifold*
  - edges contained in no more than two faces
  - vertex contained in "fan" of triangles
  - formally: every *vertex star* St($v$) is a disk
- Keeps formulas simple
- Fewer special cases in code
- Easier to translate between smooth/discrete



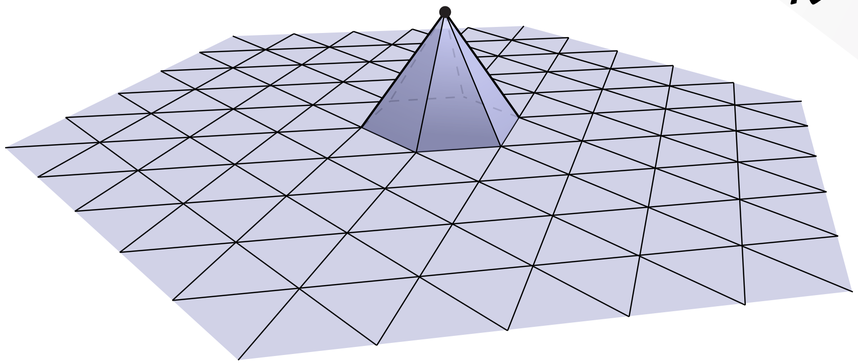

MANIFOLD

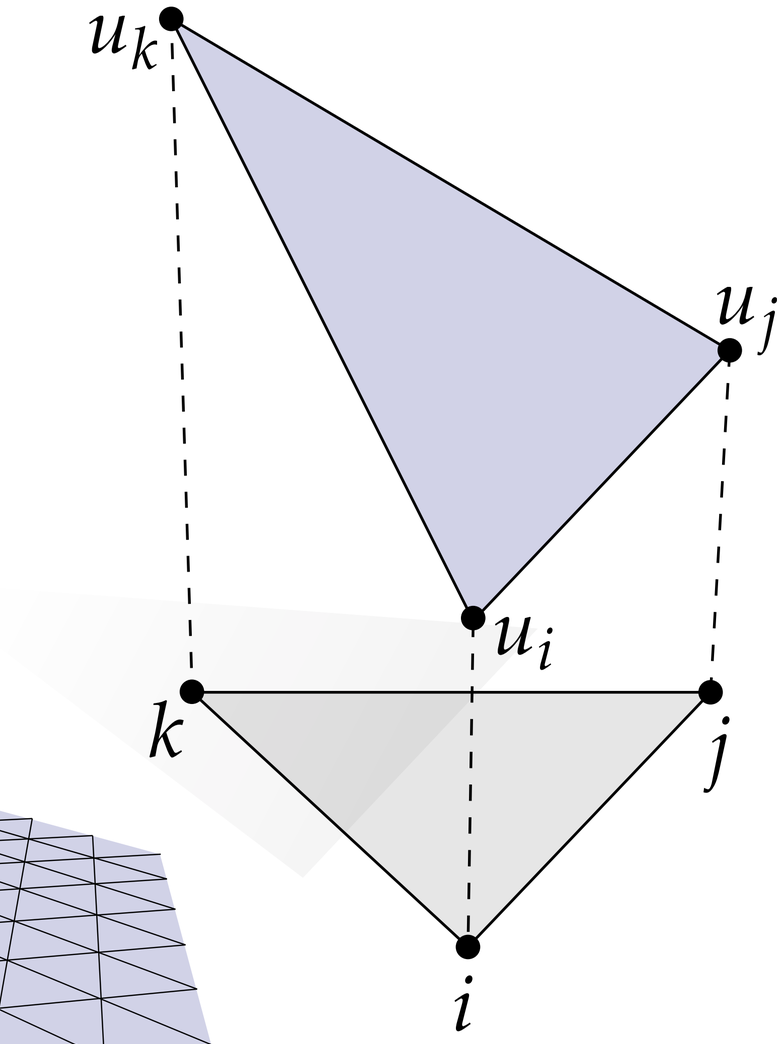


NONMANIFOLD

# *Piecewise Linear Function*

- Typical way to encode any function $u$ on a triangle mesh
- Store one value $u_i$ per vertex $i$
- “Extend” values linearly over each triangle
- More sophisticated schemes possible, but this one will take you surprisingly far…





# *"Discretized" vs. "Discrete"*

- Two high-level approaches to conformal maps on triangle meshes:

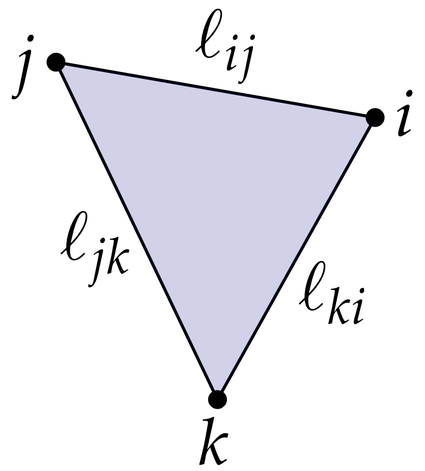
| DISCRETIZED | DISCRETE |
|---|---|
| properties satisfied only in limit of refinement (e.g., angle preservation) | quantities preserved exactly no matter how coarse (*e.g., length cross ratios*) |
| traditional perspective of scientific computing / finite element analysis | more recent perspective of *discrete differential geometry (DDG)* |
| often (but not always) leads to easy linear problems | can require slightly more difficult computation (e.g., convex optimization) |
| most of the algorithms we'll consider (*e.g.*, LSCM) | only a few algorithms: circle packing, CETM, inversive distance |

# *Discrete Metric*

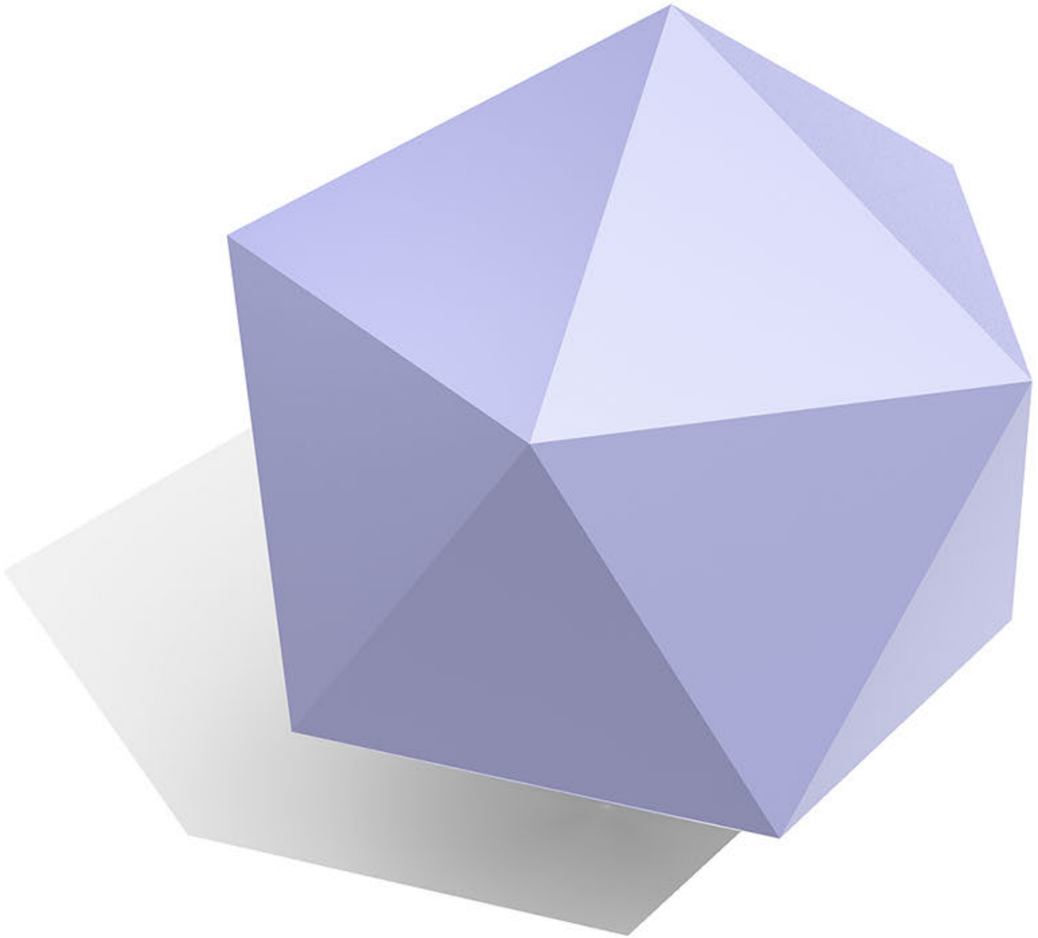
- "Discrete" point of view: try to **exactly** capture smooth relationship

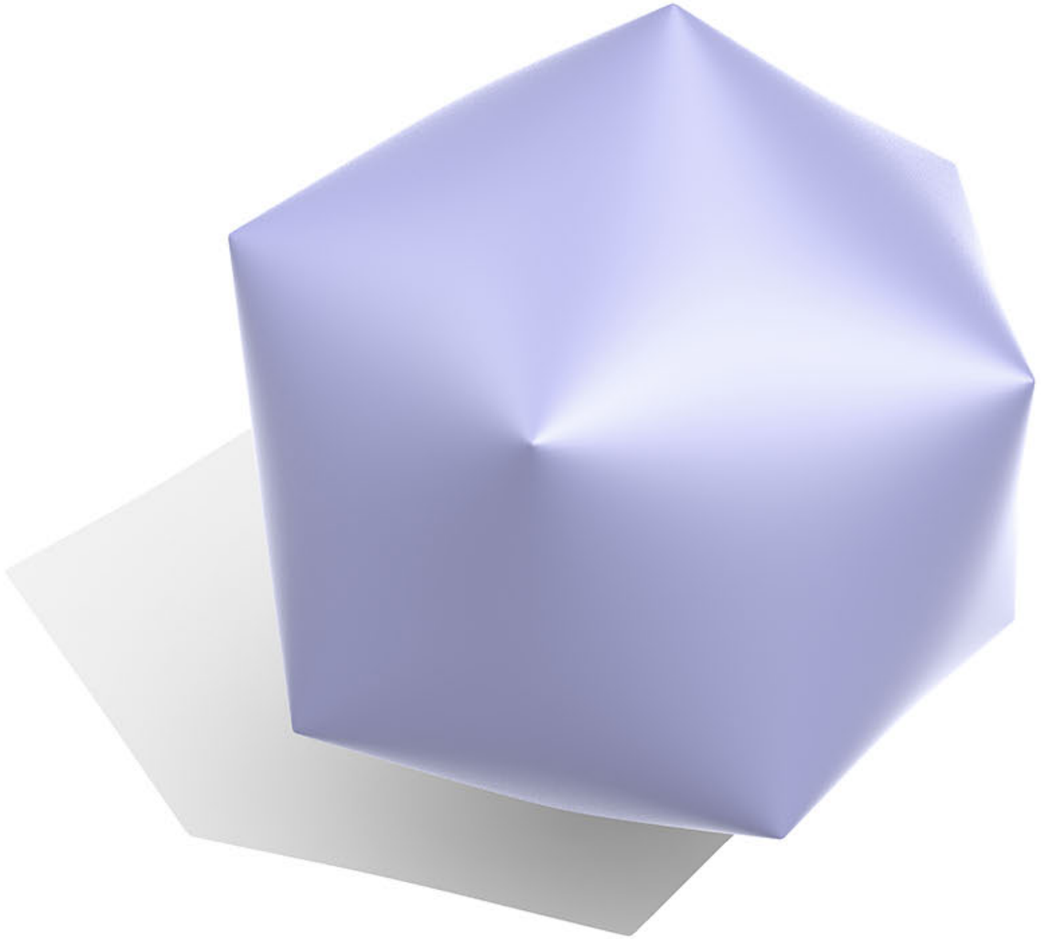
$$\tilde{g} = e^{2u} g$$

- What is a discrete metric?
  - Smooth metric allowed us to measure lengths: $|X| = \sqrt{g(X, X)}$
  - Discrete metric is simply length assigned to each edge: $\ell : E \to \mathbb{R}_{>0}$
  - Must also satisfy triangle inequality: $\ell_{ij} \leq \ell_{jk} + \ell_{ki}$
  - Can then be extended to Euclidean metric per triangle



# *Discrete Metric—Visualized*





(a.k.a. *"cone metric"*)

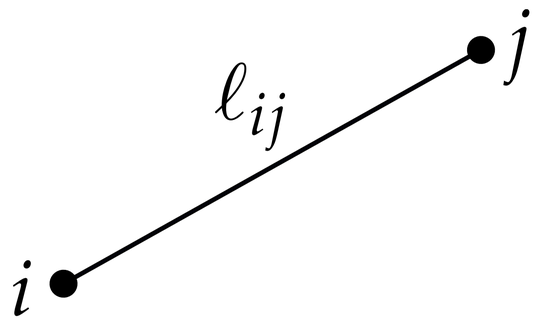
# *Conformal Equivalence of Triangle Meshes*

- "Discrete" point of view: try to **exactly** capture smooth relationship

$$\tilde{g} = e^{2u} g$$

- Discrete analogue: two discrete metrics are conformally equivalent if there is a function $u$ at vertices such that

$$\boxed{\tilde{\ell}_{ij} = e^{(u_i + u_j)/2} \ell_{ij}}$$



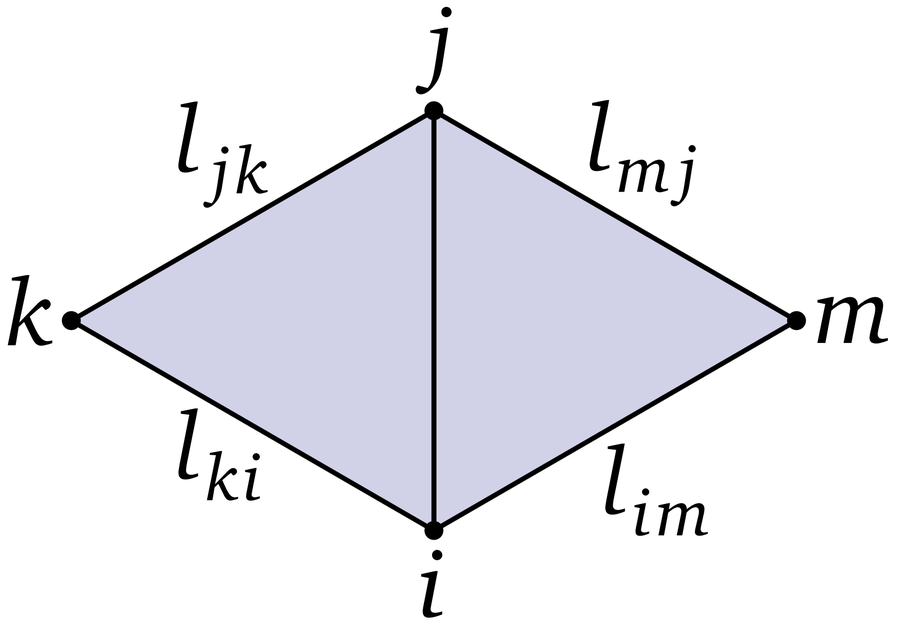

- Initially looks like naïve numerical approximation
- Turns out to provide complete discrete theory that (exactly) captures much of the behavior found in the smooth setting.

LUO, *"Combinatorial Yamabe Flow on Surfaces"* (2004)

# *Preservation of Length Cross Ratios*

**Fact.** (Springborn-Schröder-Pinkall)

If two discrete metrics are conformally equivalent, then they exhibit the same *length cross ratios*.




$$\mathfrak{c}_{ij} := \frac{l_{im}}{l_{mj}} \frac{l_{jk}}{l_{ki}}$$

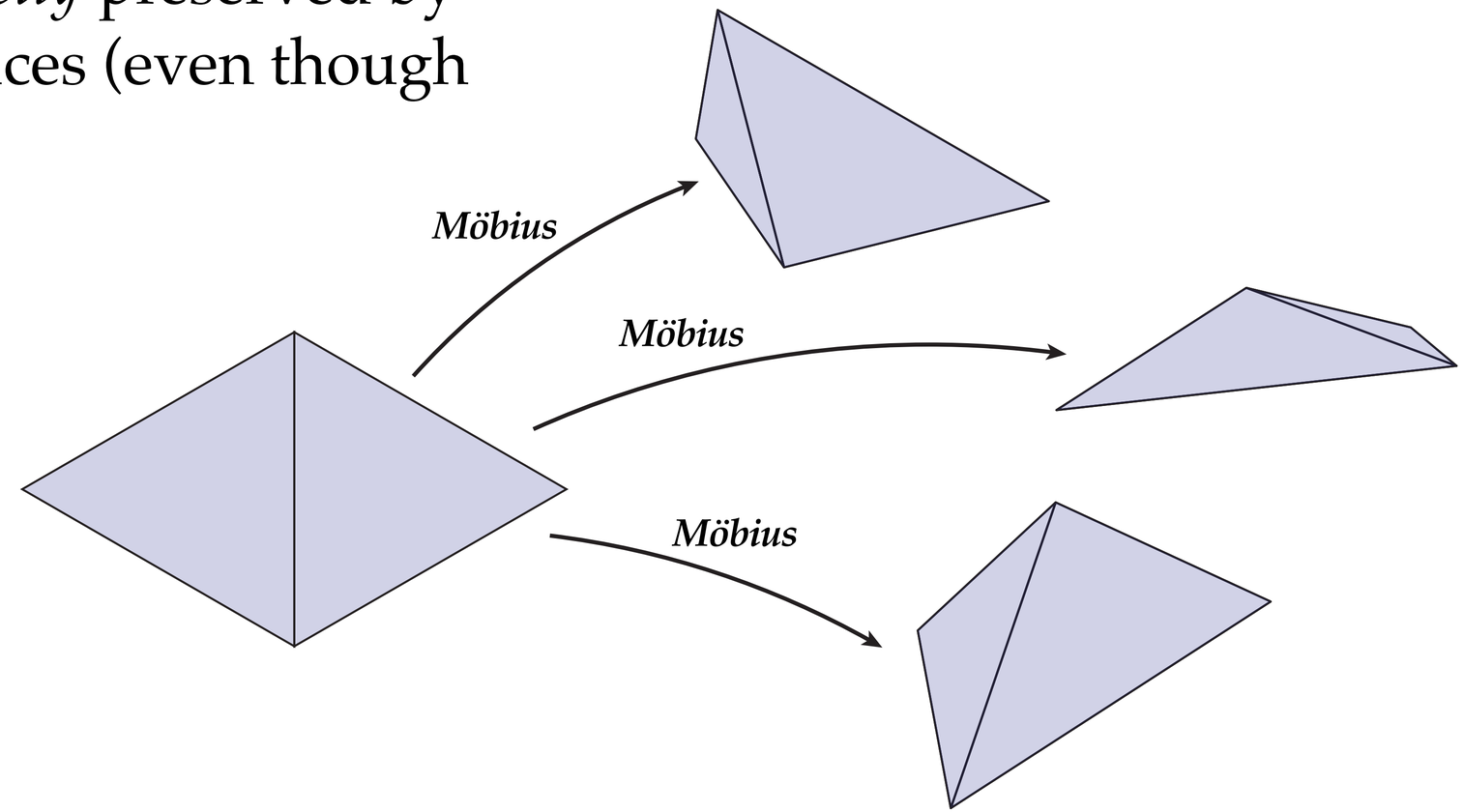
*length cross ratio*

$$\boxed{\mathfrak{c} \equiv \tilde{\mathfrak{c}}}$$

*discrete conformal equivalence*

# *Möbius Invariance of CETM*

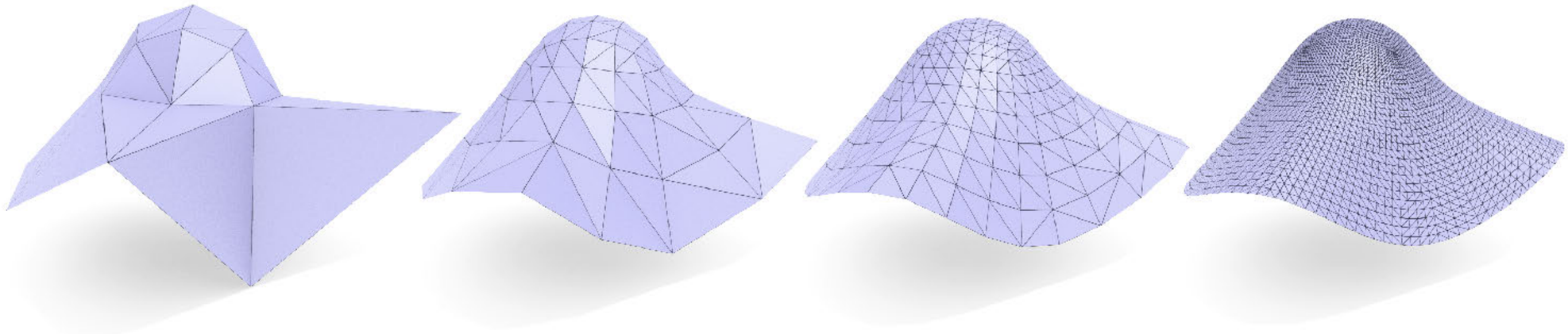
**Fact.** Length cross ratios are *exactly* preserved by Möbius transformations of vertices (even though *angles* are not!)

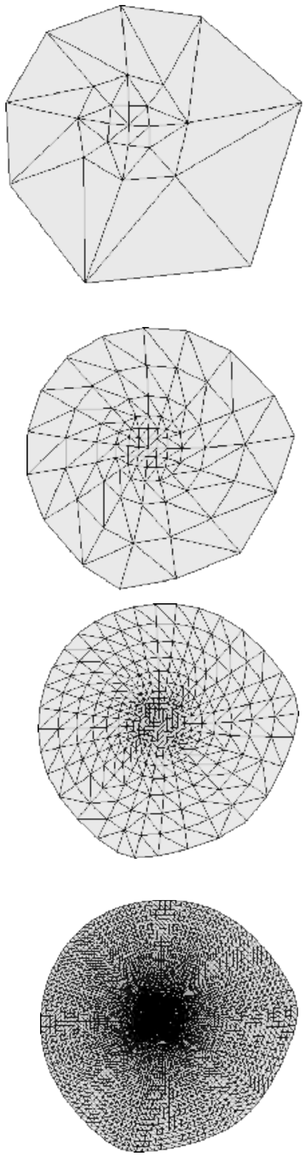


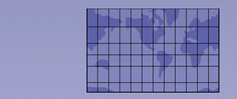

**Key idea:** discrete theory may not always capture "most obvious" properties (like angles); should try to think more broadly: *"what other characterizations are available?"*

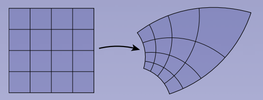
# *"Discretized" Conformal Maps?*

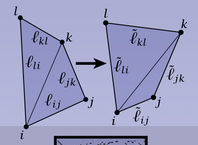
- Ok, that's the *"discrete"* definition…
- …What about *"discretized"* notions of conformal maps?
  - these are much easier to come by
  - basically anything that converges under refinement
  - will see more of this as we discuss algorithms

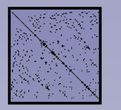




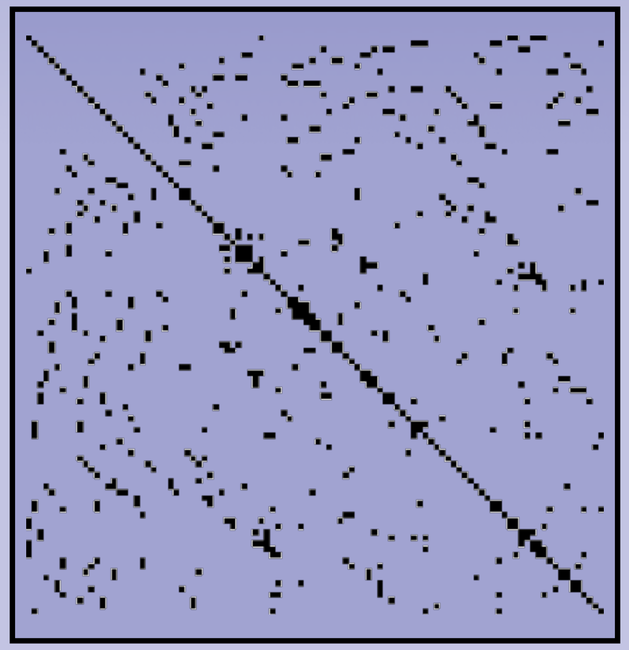

PART I: OVERVIEW


PART II: SMOOTH THEORY


PART III: DISCRETIZATION


PART IV: ALGORITHMS

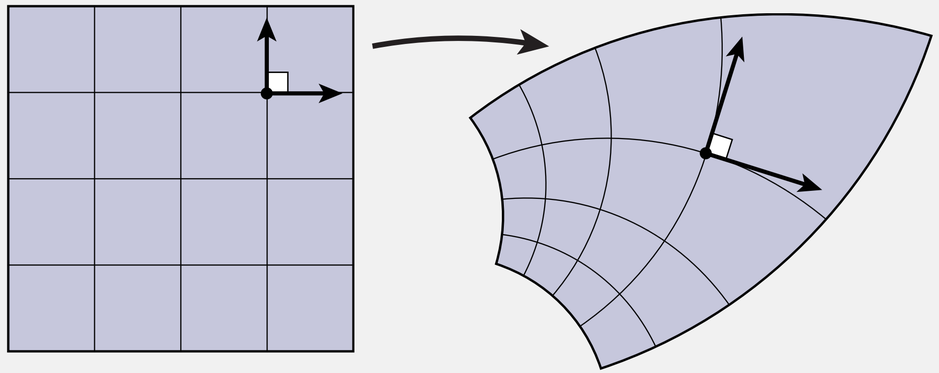
# PART IV: ALGORITHMS



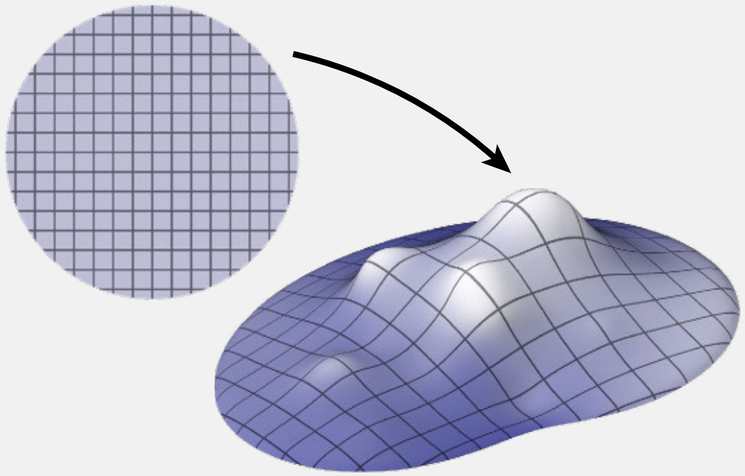
# CONFORMAL GEOMETRY PROCESSING

**Keenan Crane • CMU 15-458/858B • Fall 2017**

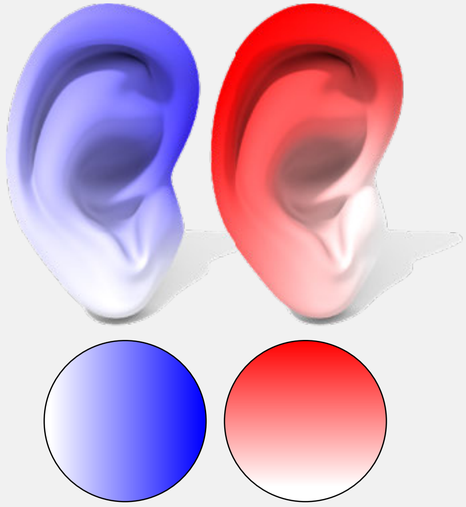
# *(Some) Characterizations of Conformal Maps*



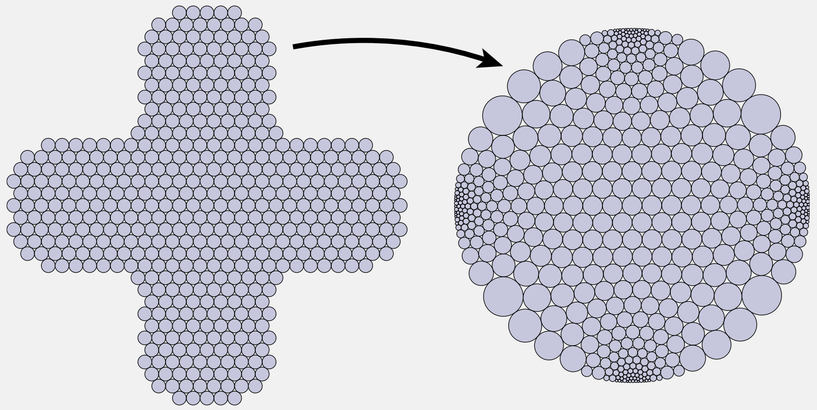
**angle preservation**



**metric rescaling**



**conjugate harmonic functions**



**preservation of circles**

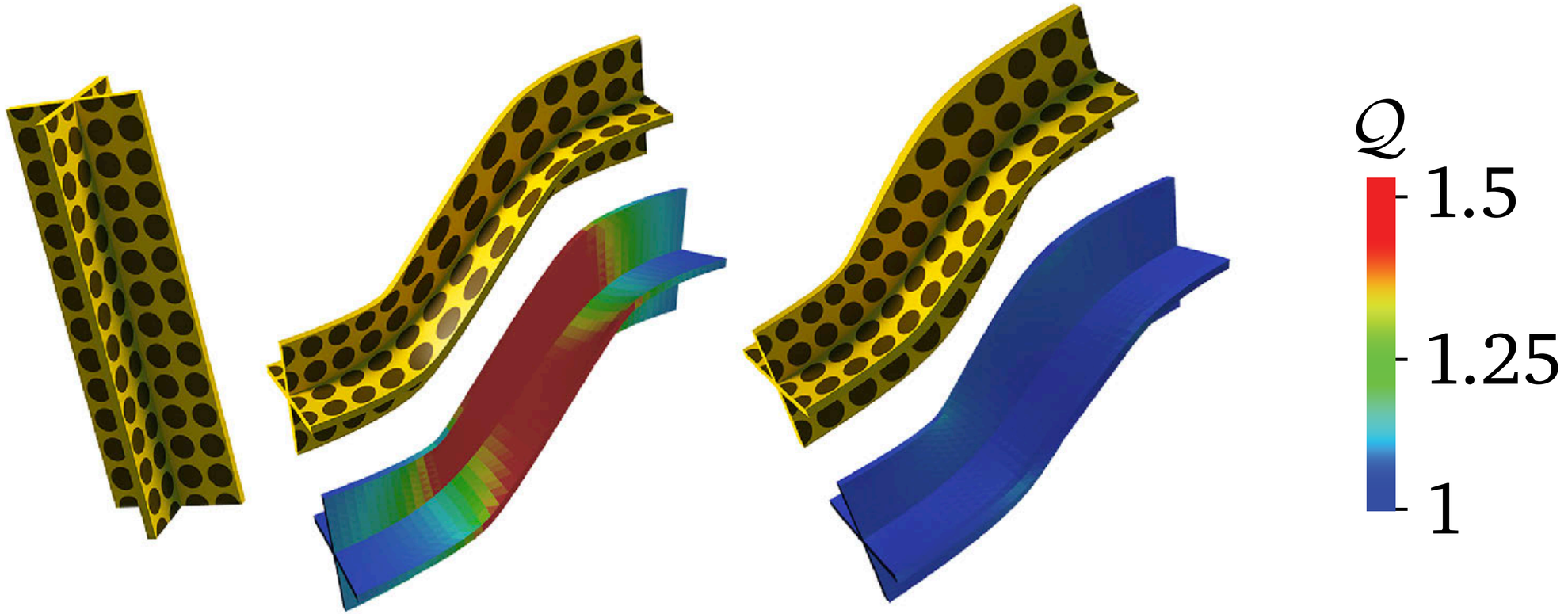
**critical points of Dirichlet energy**

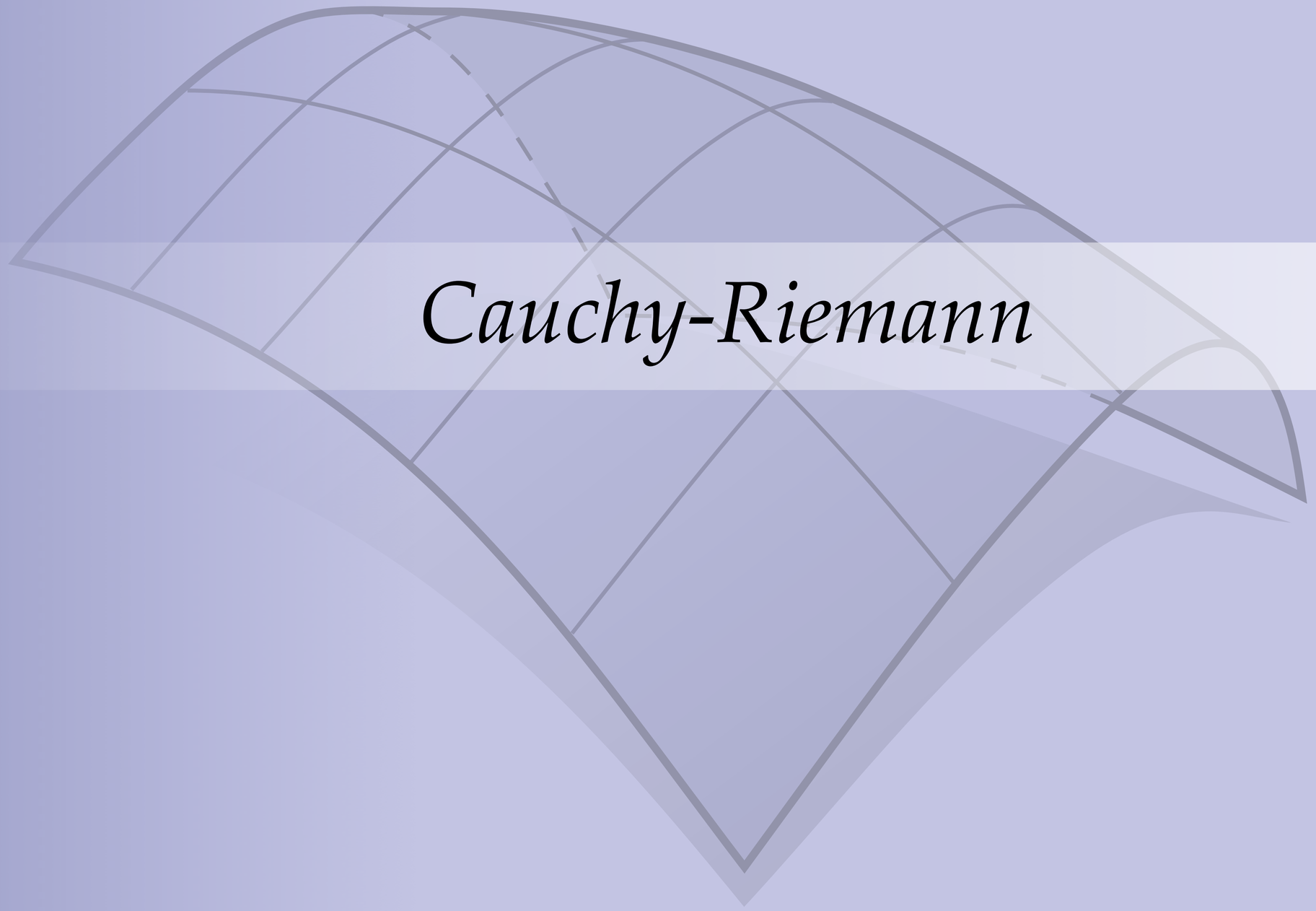
# *(Some) Conformal Geometry Algorithms*

| CHARACTERIZATION | ALGORITHMS |
|---|---|
| **Cauchy-Riemann** | *least square conformal maps (LSCM)* |
| **Dirichlet energy** | *discrete conformal parameterization (DCP)*<br>*genus zero surface conformal mapping (GZ)* |
| **angle preservation** | *angle based flattening (ABF)* |
| **circle preservation** | *circle packing*<br>*circle patterns (CP)* |
| **metric rescaling** | *conformal prescription with metric scaling (CPMS)*<br>*conformal equivalence of triangle meshes (CETM)* |
| **conjugate harmonic** | *boundary first flattening (BFF)* |

# *Quasiconformal Distortion*

- Only conformal map from triangle to triangle is *similarity* (rigid + scale)
- *Quasiconformal distortion (**Q**)* is ratio of singular values in each triangle
- Measures "how conformal"(want $Q = 1$ everywhere)

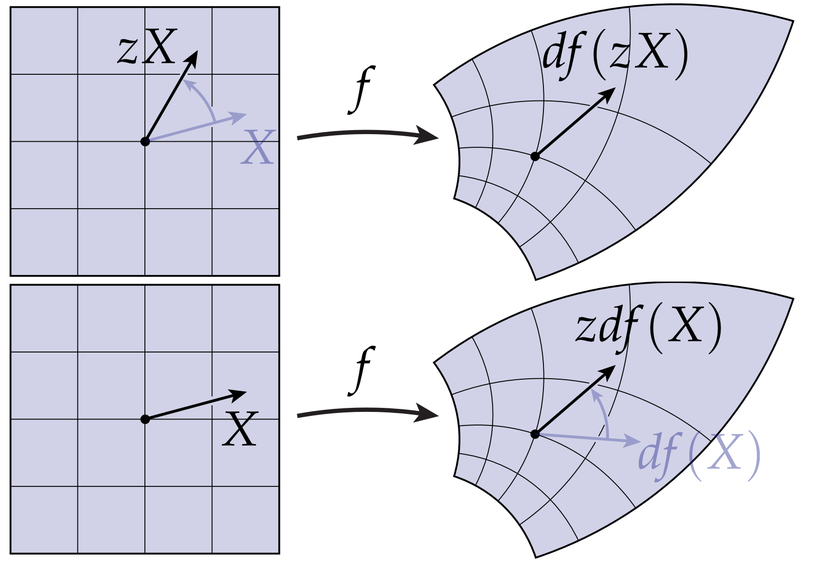




# *Cauchy-Riemann*

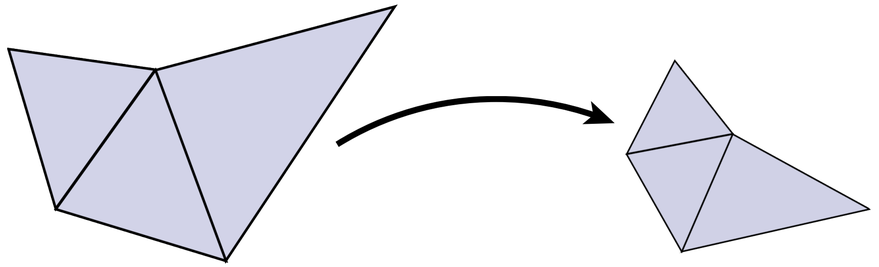
# *From Cauchy-Riemann to Algorithms*

- Natural starting point: solve Cauchy-Riemann equation
- Already know that there will be no *exact* solutions for a triangle mesh
- Instead, find solution that minimizes residual
- Leads to *least squares conformal map (LSCM)*
- Very popular; in *Maya*, *Blender*, `libigl`, …
- Fully automatic; *no control over target shape*




$$\boxed{df(zX) = zdf(X)}$$

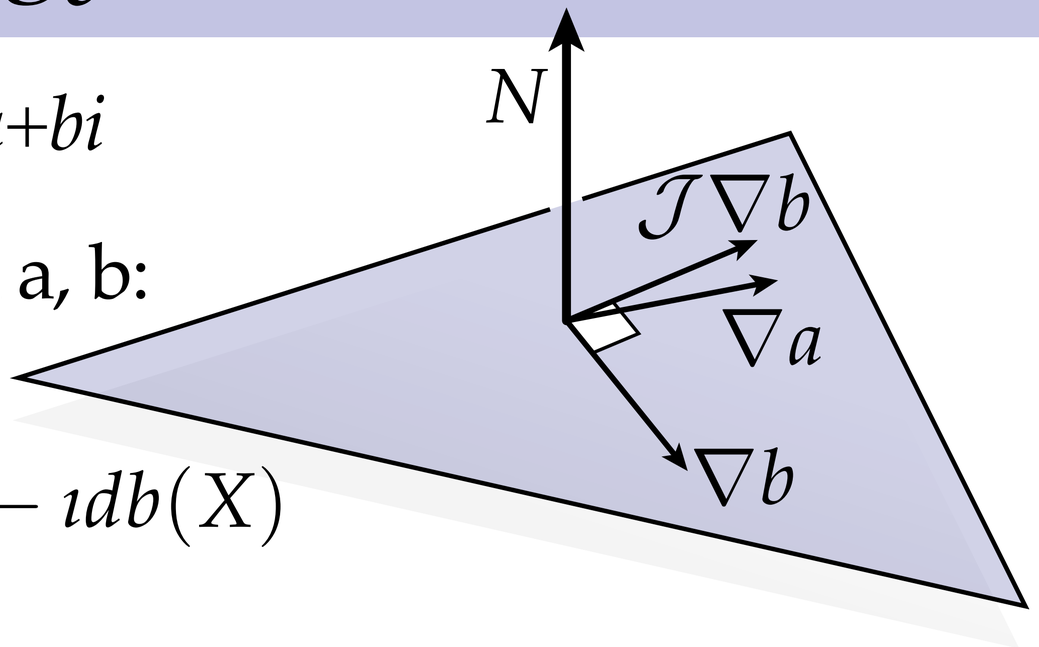
**CAUCHY-RIEMANN**



# *Least Square Conformal Energy*

- Write map as pair of real coordinates: $f = a+bi$
- Express Cauchy-Riemann as condition on a, b:

$$\begin{aligned} & df(\mathcal{J}X) &= & \imath df(X) \\ \iff \quad & da(\mathcal{J}X) + \imath db(\mathcal{J}X) &= & \imath da(X) - \imath db(X) \\ \iff \quad & \nabla a &= & -\mathcal{J}\nabla b \end{aligned}$$




- Sum failure of this relationship to hold over all triangles:

$$E_{\text{LSCM}}(a,b) := \sum_{ijk \in F} \mathcal{A}_{ijk} \left( (\nabla a)_{ijk} - N_{ijk} \times (\nabla b)_{ijk} \right)^2$$

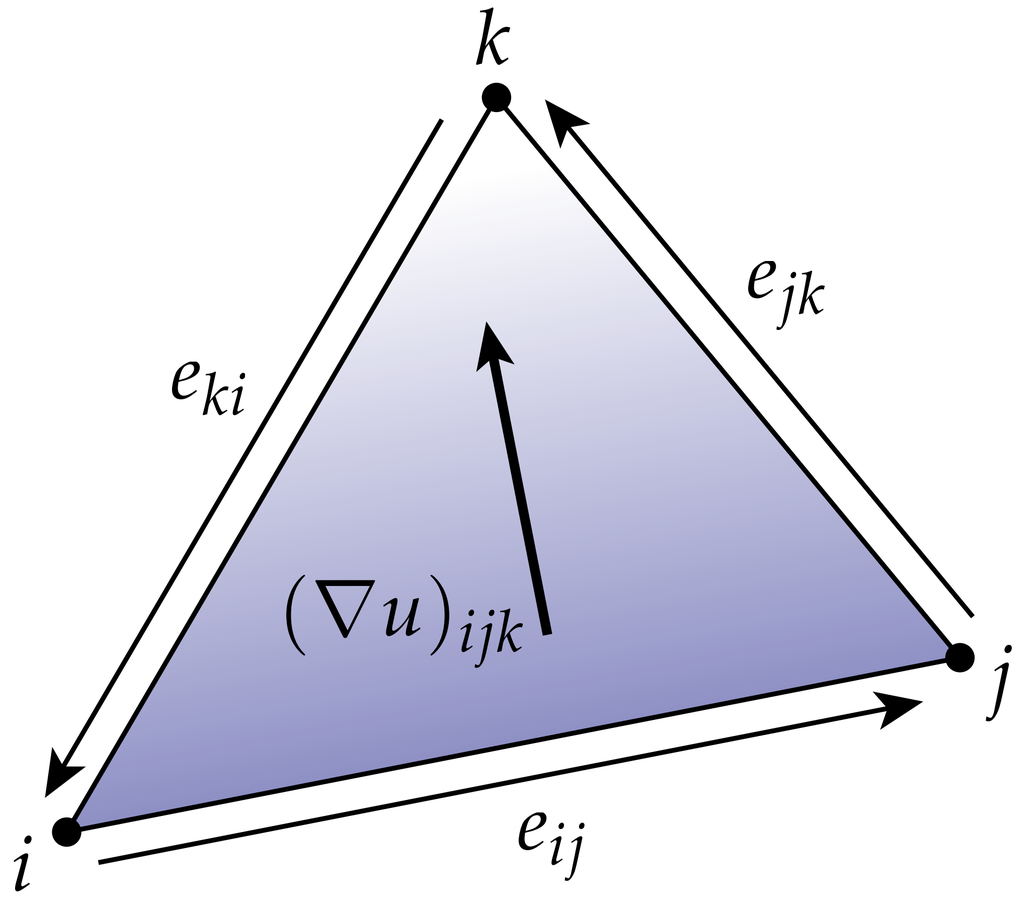
- Resulting energy is *convex* and *quadratic* (*i.e.*, "easy"!)

# *Gradient of a Piecewise Linear Function*

- Many geometry processing algorithms need *gradient* of a function (i.e., direction of "steepest increase")
- Easy formula on a triangle mesh:

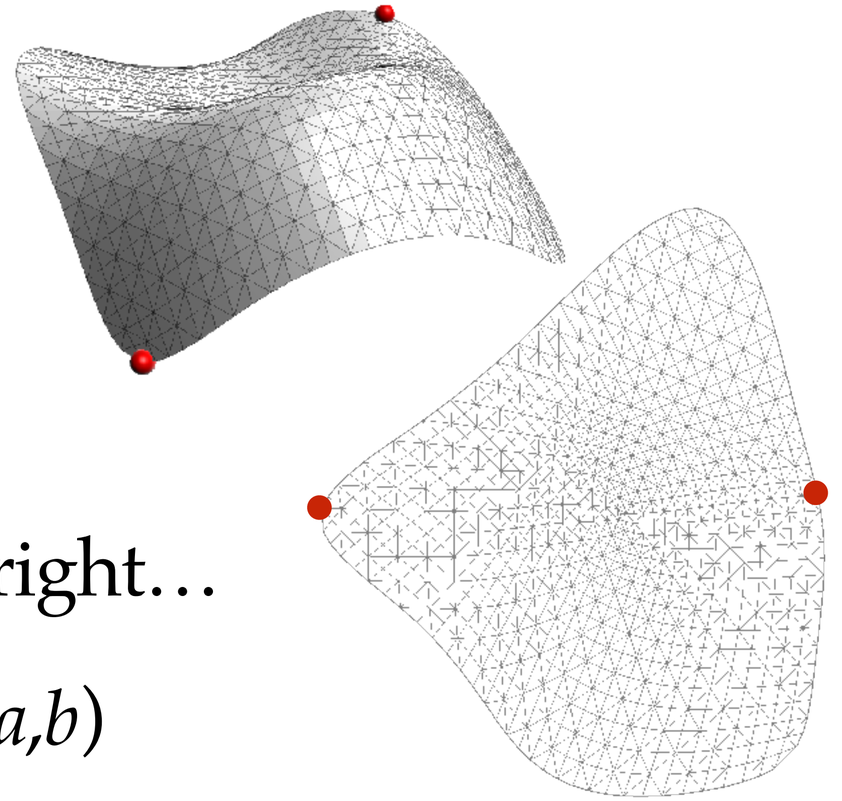
$$(\nabla u)_{ijk} = \frac{1}{2\mathcal{A}_{ijk}}(u_i e_{jk} + u_j e_{ki} + u_k e_{ij})$$

- Since function is *linear*, gradient is *constant* across each triangle.



# *Least Square Conformal Maps (LSCM)*

- Coordinate functions (*a*,*b*) that minimize $E_{\text{LSCM}}$ give the "best" map
- **Problem:** *constant* functions have zero energy!
- **Solution*:** *"pin"* two vertices to fixed locations
  - one vertex determines translation in plane
  - the other determines rotation & scale
- *Will see later that this solution is still not quite right…
- To minimize, set gradient to zero and solve for (*a*,*b*)
- Numerical problem is sparse linear system (very easy to solve)



# *Least Square Conformal Maps (LSCM)*

- Coordinate functions $(a,b)$ that minimize $E_{\text{LSCM}}$ give the "best" map
- Can encode energy as a quadratic form:

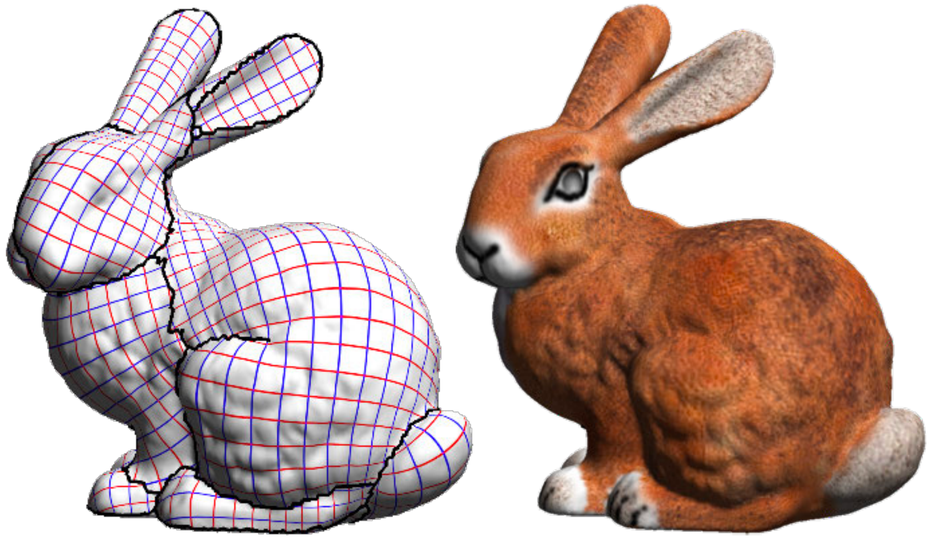
$$\mathsf{x} := \begin{bmatrix} a_1 & b_1 & \cdots & a_n & b_n \end{bmatrix}^{\mathsf{T}}$$

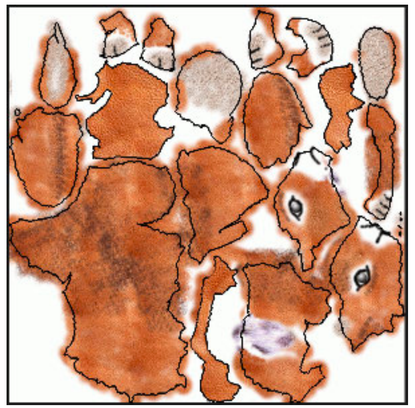
$$E_{\text{LSCM}}(a,b) = \tfrac{1}{2}\,\mathsf{x}^{\mathsf{T}}\mathsf{A}\,\mathsf{x}, \quad \mathsf{A} \in \mathbb{R}^{2n\times 2n}$$

- Minimize by setting gradient equal to zero:

$$\mathsf{A}\mathsf{x} = 0$$

- Just need to solve a linear system
- **Problem:** has trivial solution $\mathsf{x} = 0$!





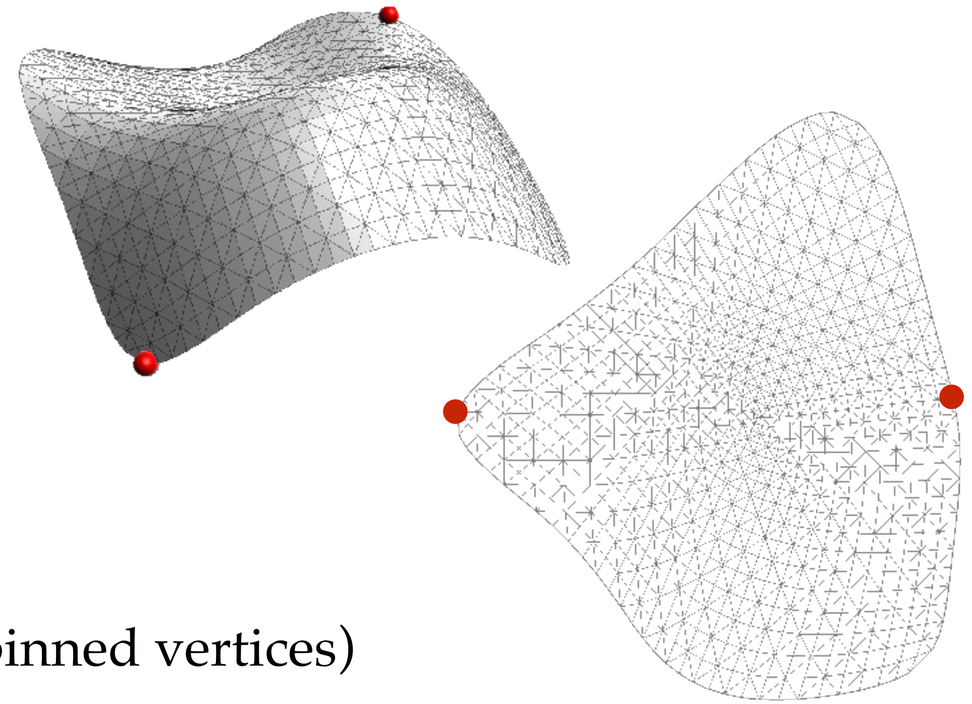
LÉVY, PETITJEAN, RAY, MAILLOT, *"Least Squares Conformal Maps for Automatic Texture Atlas Generation"* (2011)

# *LSCM—Nontrivial Solution via "Pinning"*

- In fact, any *constant* map will have zero energy, since gradient is zero:

$$E_{\text{LSCM}}(a,b) := \sum_{ijk \in F} \mathcal{A}_{ijk} \left( (\nabla a)_{ijk} - N_{ijk} \times (\nabla b)_{ijk} \right)^2$$

- Idea: "pin" any two vertices to arbitrary locations
  - one vertex determines global translation
  - another vertex determines scale/rotation



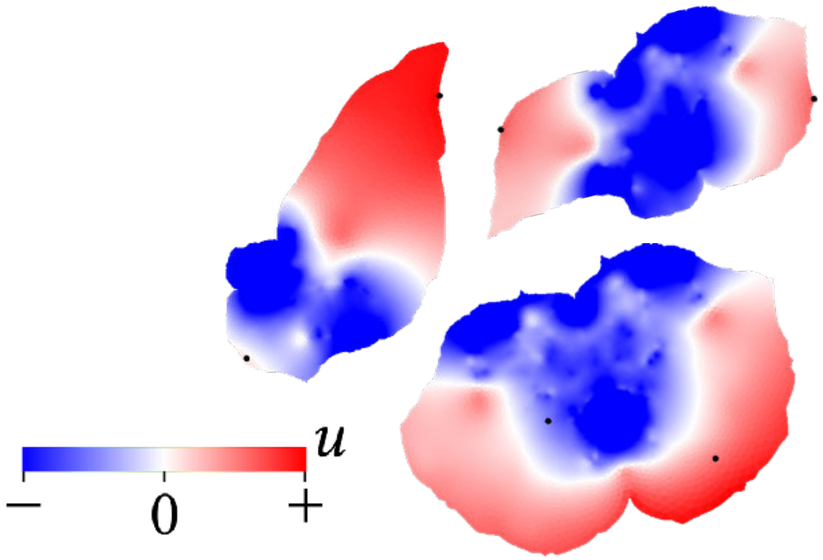
- Linear system now has nonzero RHS:

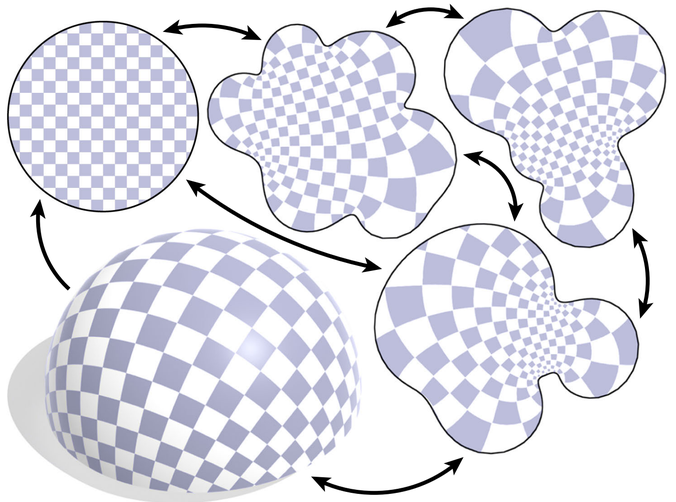
$$\hat{\mathsf{A}}\hat{\mathsf{x}} = \mathsf{b}$$

("hat" indicates removed rows/columns, corresponding to pinned vertices)

# *Problems with Pinning*

- To get a unique solution we "pinned down" two vertices
- Two problems with this approach:
  1. map can be unpredictable, distorted depending on choice of vertices
  2. we should have ***way*** more choice about what target shape looks like!



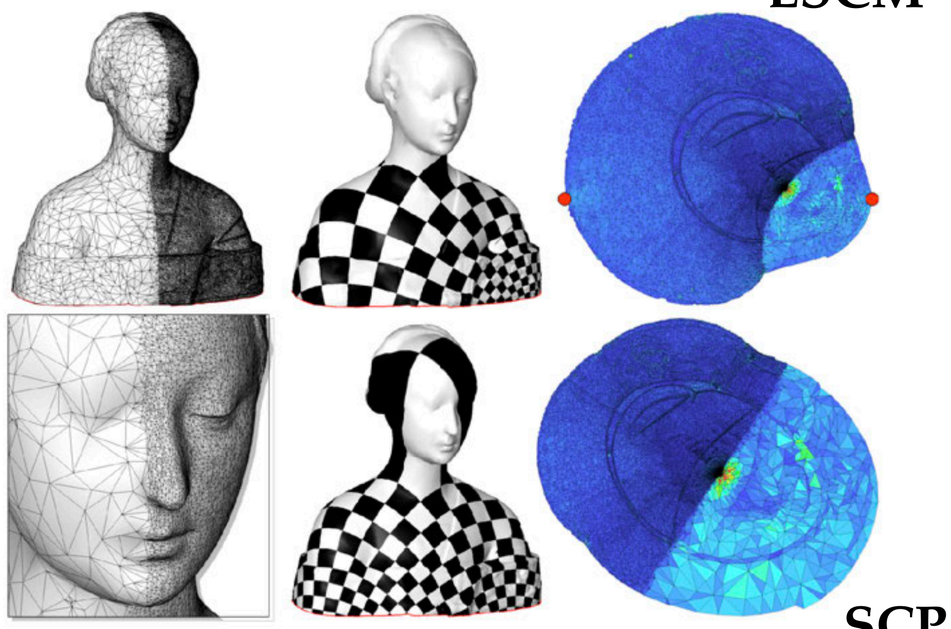
*Will address the first issue first…*

**[DEMO]**

# *Spectral Conformal Parameterization (SCP)*

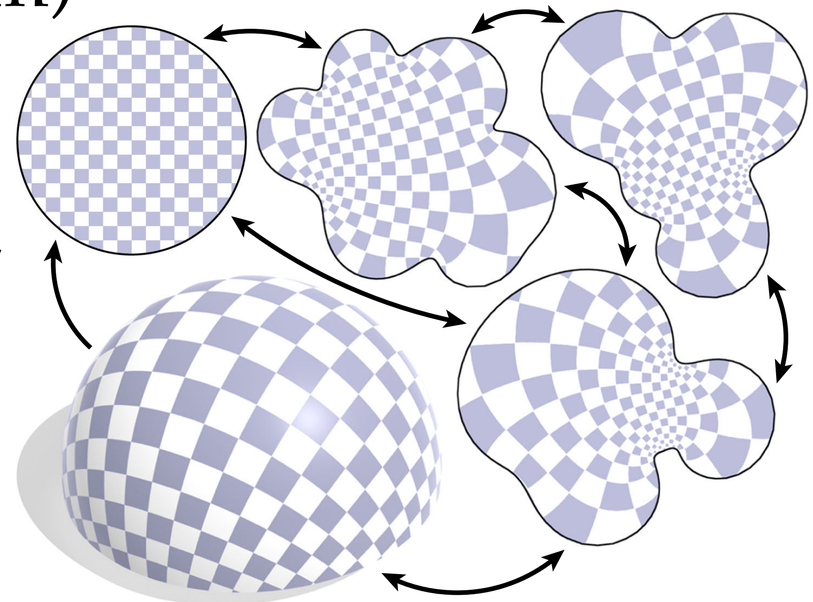
- "Pinning" was used to prevent degenerate (constant) solution
- Alternatively, can ask for smallest energy among all *unit-norm* solutions
- Compute principal eigenvector of energy matrix
- **Q:** Why does this work better?
  - *identical* from perspective of linear algebra
  - (much) better accuracy in floating-point

**[DEMO]**



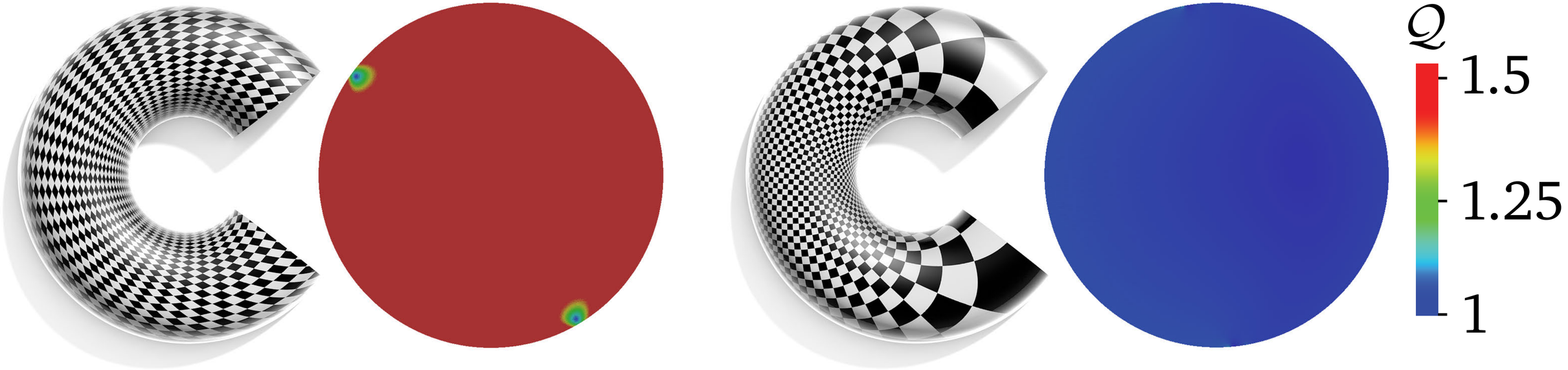
# *Conformal Maps—Boundary Conditions?*

- *Something is still wrong!*
  - In the discrete setting, specified just two points on boundary (just rigid motion & scaling in the plane)
  - In the smooth setting, there are **far** more ways to conformally flatten (Riemann Mapping Theorem)
- What happened here?
  - Among *piecewise linear* maps, "most conformal" solution is unique (up to rigid motion).
  - But what if we want to *control* target shape?



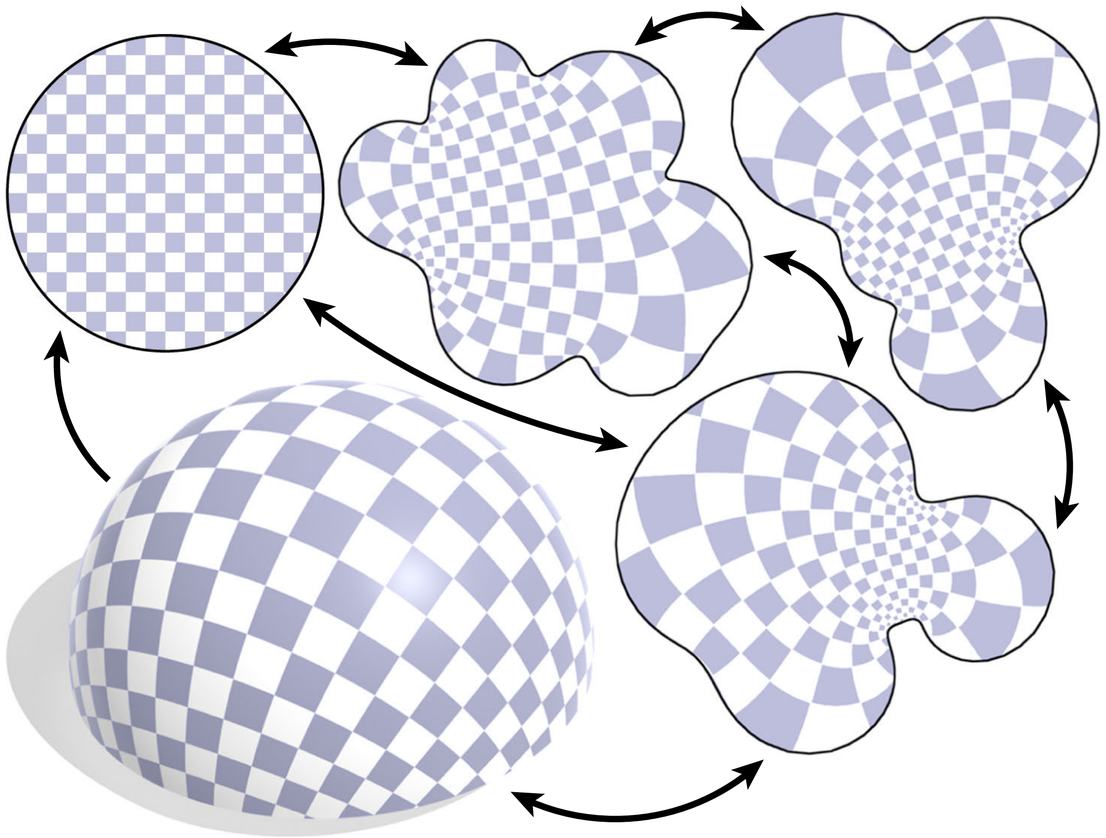
# *Prescribing the Entire Boundary Doesn't Work*

- **First attempt:** pin *all* boundary points to desired target shape
- **Problem:** In general there is no conformal map compatible with a given map along the boundary
- Least-squares yields *harmonic* map with severe angle distortion:

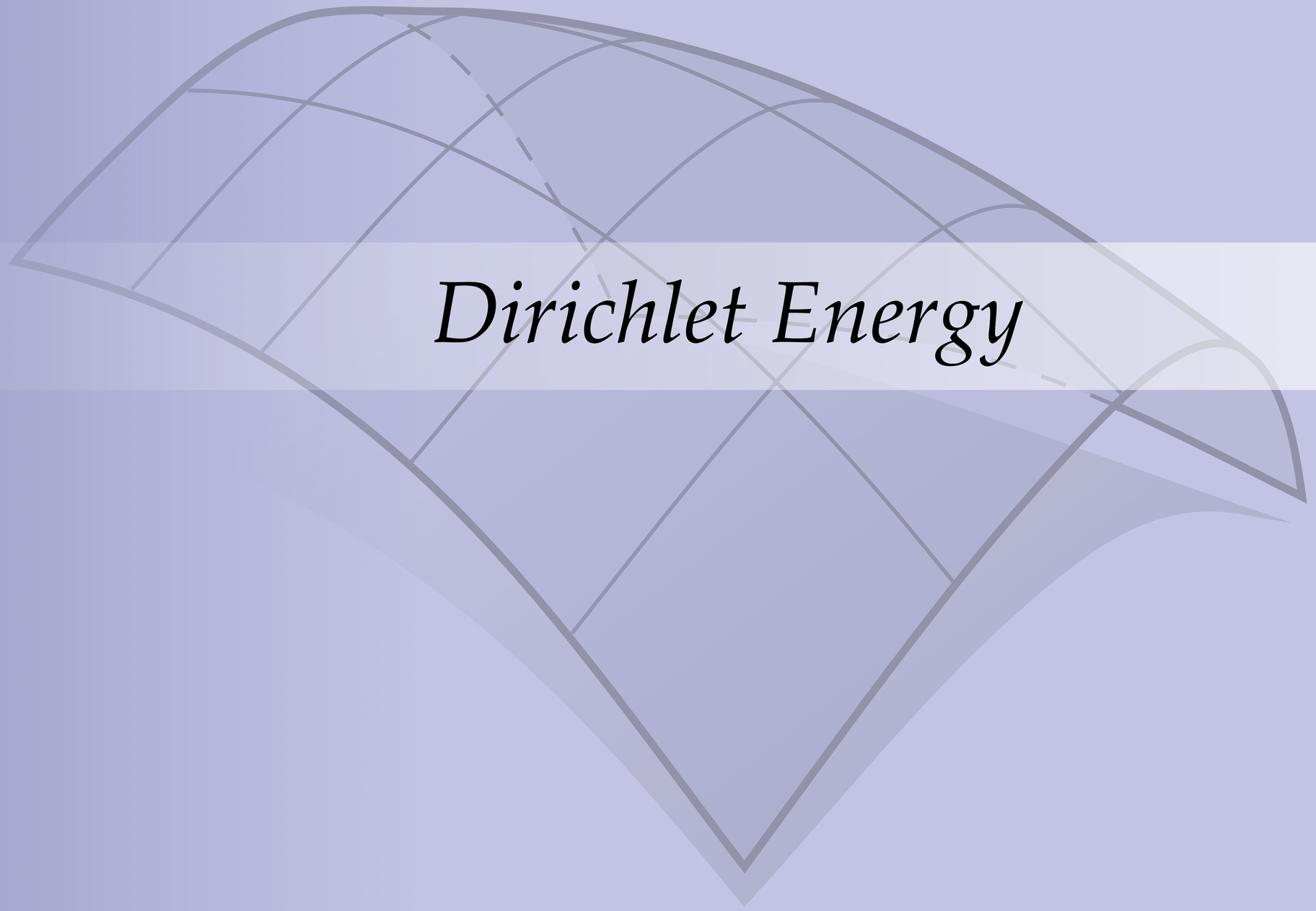



HARMONIC CONFORMAL

# *…So what if we want to control target shape?*



Will revisit this question later—when
we have more tools at our disposal!



# *Dirichlet Energy*

# *Dirichlet Energy*

- Different characterization of conformal maps: critical points of so-called *Dirichlet energy*
  - Physical analogy: elastic membrane that wants to have *zero* area
  - When this energy is minimized, we get a conformal map…
  - …under very special assumptions on the domain / boundary conditions!

- Alternative route to LSCM (a.k.a DCP) & other algorithms

# *Smooth Dirichlet Energy*

- Consider any map $f$ between manifolds $M$ and $N$
- *Dirichlet energy* is given by:

$$E_D(f) := \int_M |df|^2$$

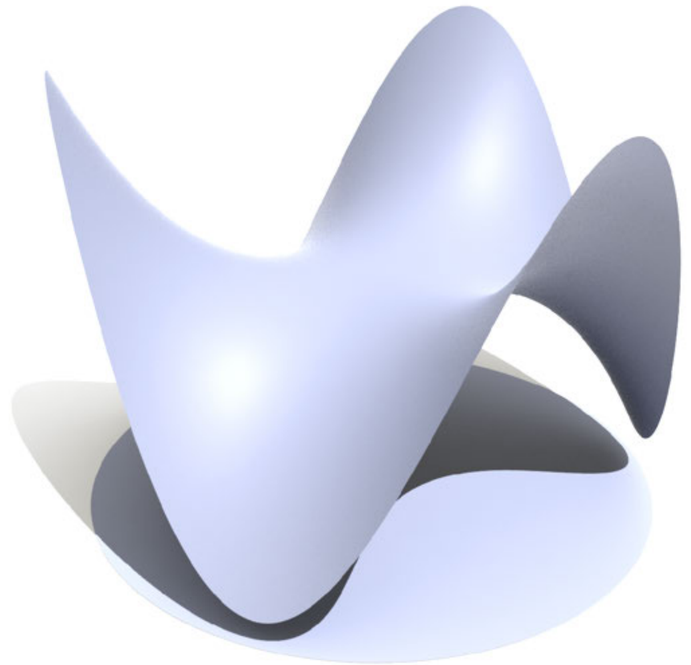
- Any critical point (*e.g.*, local minimum) is called a *harmonic map*.
- Perhaps most common case in geometry processing:
  - $M$ is a surface
  - $N$ is just the real line

# *Real Harmonic Functions*

- Intuitively, a *harmonic* function is the "smoothest" function that interpolates given values on the boundary; looks "saddle-like"
- A function is *harmonic* if applying the Laplacian yields zero
- E.g., in 2D:

$$f(x,y) = x^3 - 3x^2y - 3xy^2 + y^3$$

$$\begin{aligned} \Delta f &= \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \\ &= (6x - 6y) + (-6x + 6y) \\ &= 0 \end{aligned}$$



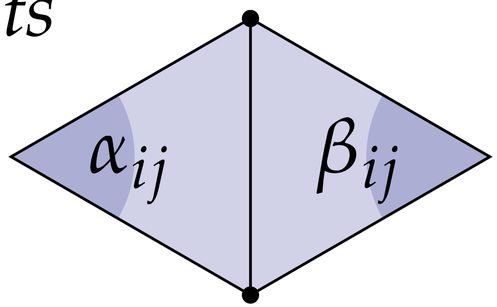
# *Discrete Harmonic Functions*

- Harmonic functions are easy to compute on a triangle mesh.
- Roughly speaking: every value is (weighted) average of its neighbors.
- More precisely, at every vertex $i$ we want

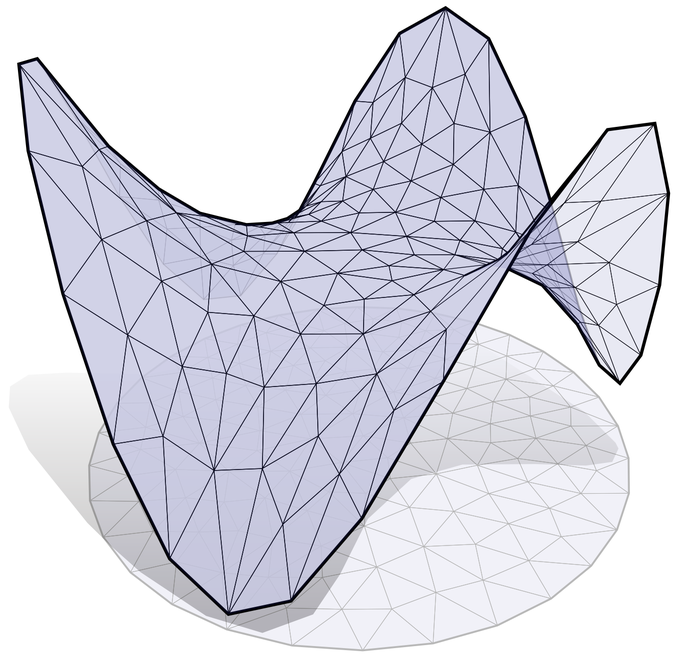
$$f_i = \sum_{ij} w_{ij} f_j \Big/ \sum_{ij} w_{ij}$$

- Typical choice for $w$ are *cotan weights*

$$w_{ij} = \tfrac{1}{2}(\cot \alpha_{ij} + \cot \beta_{ij})$$

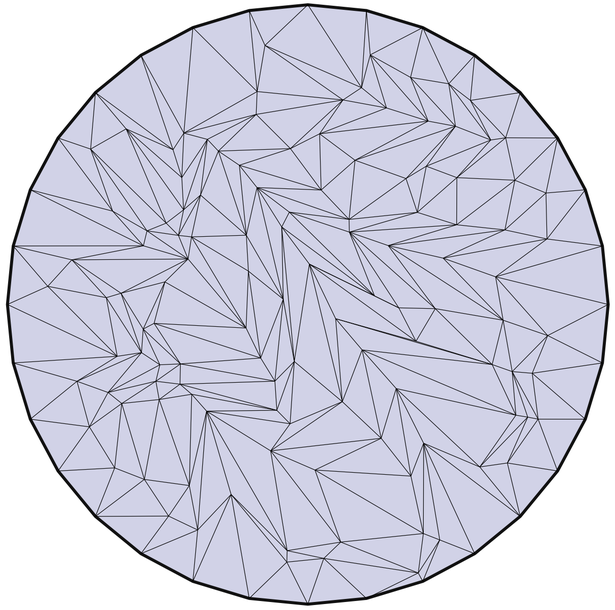



- Boundary values $f_i$ are fixed
- Sparse linear system; many (fast!) ways to solve

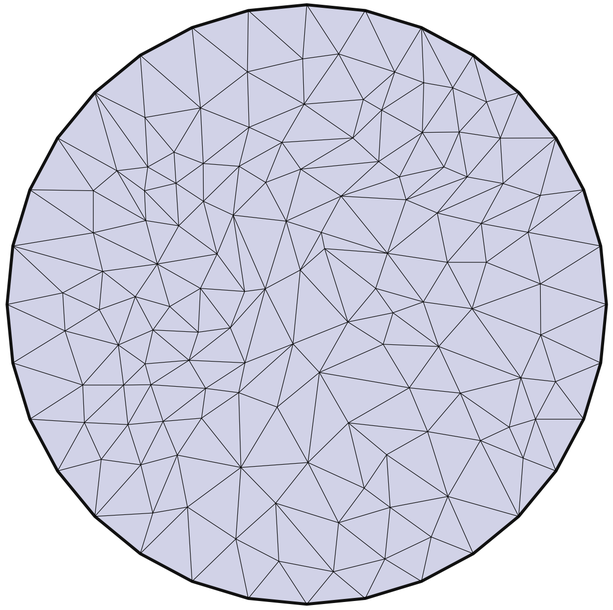


# *Discrete Harmonic Map—Neanderthal method*

- How can we actually *compute* a harmonic map?
- Simple but stupid idea: repeatedly average with neighbors (*Jacobi)*
- Much better idea: express as linear system and solve with a fast solver.

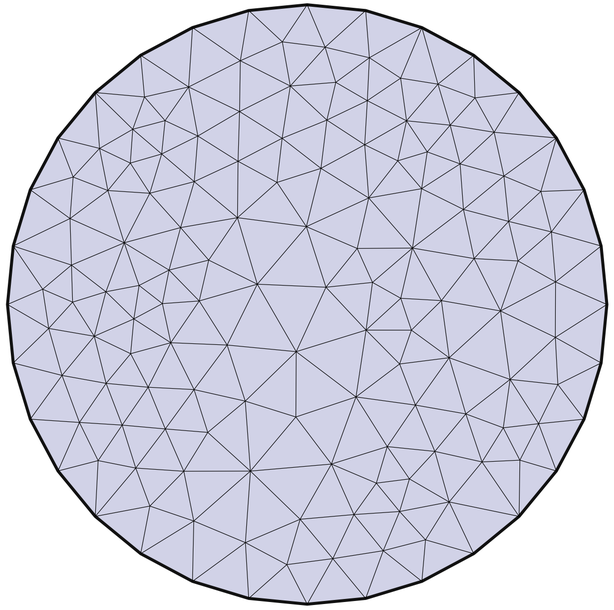


**input**



**iteration 1**

• • •

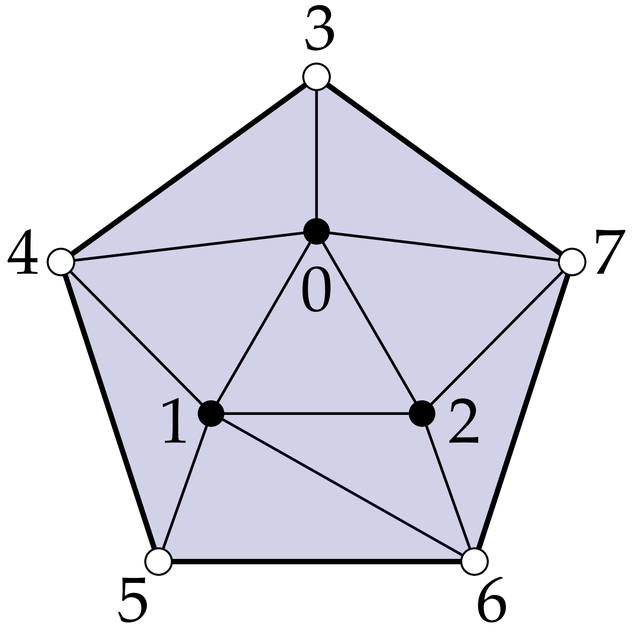


**iteration 736**
*(converged)*

**[DEMO]**

# *Meshes & Matrices*

- Common task in geometry processing: solve system of linear equations involving variables on vertices (or edges, or faces, …)
- Basic idea: give each mesh element a unique *index*; build a matrix encoding system of equations.
- E.g., find values $u$ for **black** vertices that are average of neighbors:




$$\begin{aligned} u_0 &= (u_1 + u_2 + u_3 + u_4 + u_7)/5 \\ u_1 &= (u_0 + u_2 + u_4 + u_5 + u_6)/5 \\ u_2 &= (u_0 + u_1 + u_6 + u_7)/4 \end{aligned}$$

$$\iff \begin{bmatrix} 5 & -1 & -1 \\ -1 & 5 & -1 \\ -1 & -1 & 4 \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} f_3 + f_4 + f_7 \\ f_4 + f_5 + f_6 \\ f_6 + f_7 \end{bmatrix}$$

(Now solve with a *fast* linear solver.)

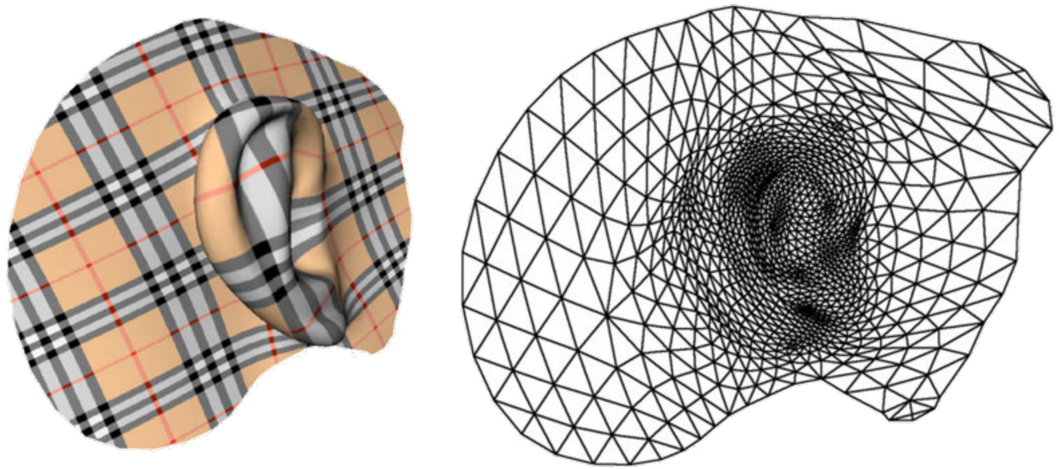
# *Dirichlet Energy and Harmonic Maps*

- **Fact*:** the residual of Cauchy-Riemann equations can be expressed as difference of Dirichlet energy and (signed) target area:

$$|| \star df - \imath df ||^2 = E_D(f) - \mathcal{A}(f)$$

*For a derivation, see Crane et al, *"Digital Geometry Processing with Discrete Exterior Calculus"*, Section 7.4

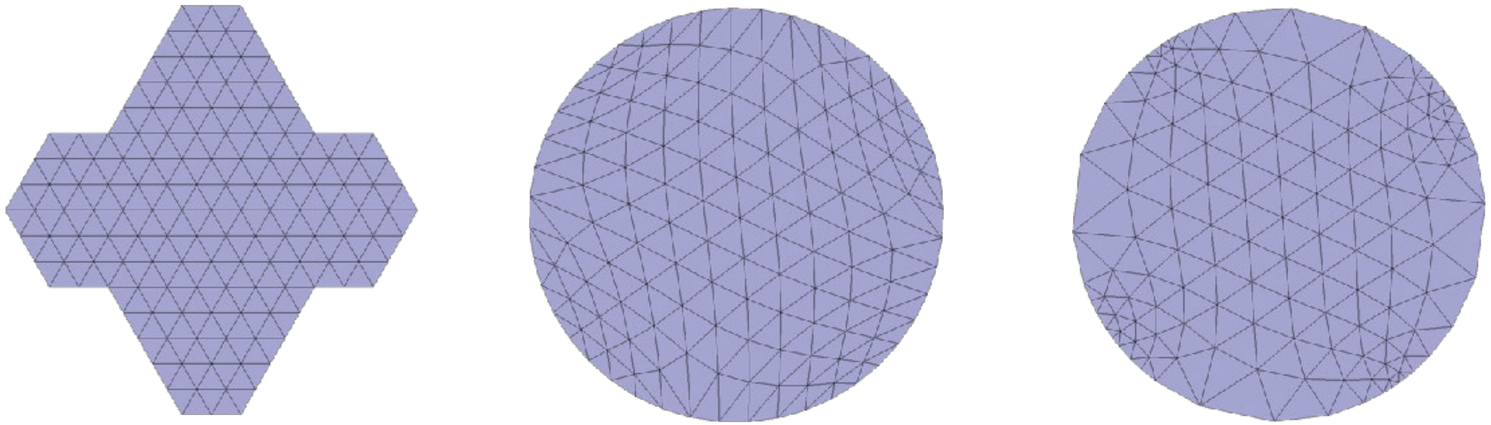
- Minimizing this energy turns out to be numerically equivalent to LSCM



DESBRUN, MEYER, ALLIEZ, *"Intrinsic Parameterizations of Surface Meshes "* (2002)

# *Harmonic Map with Fixed Area*

- **Special case:** if target area is fixed, one need only consider $E_D$
- *E.g.,* world's simplest algorithm for uniformization:
  - Iteratively average with neighbors
  - Project boundary vertices onto circle
- (Initialize by doing the same thing but with boundary fixed to circle)



*domain* *harmonic* *conformal*

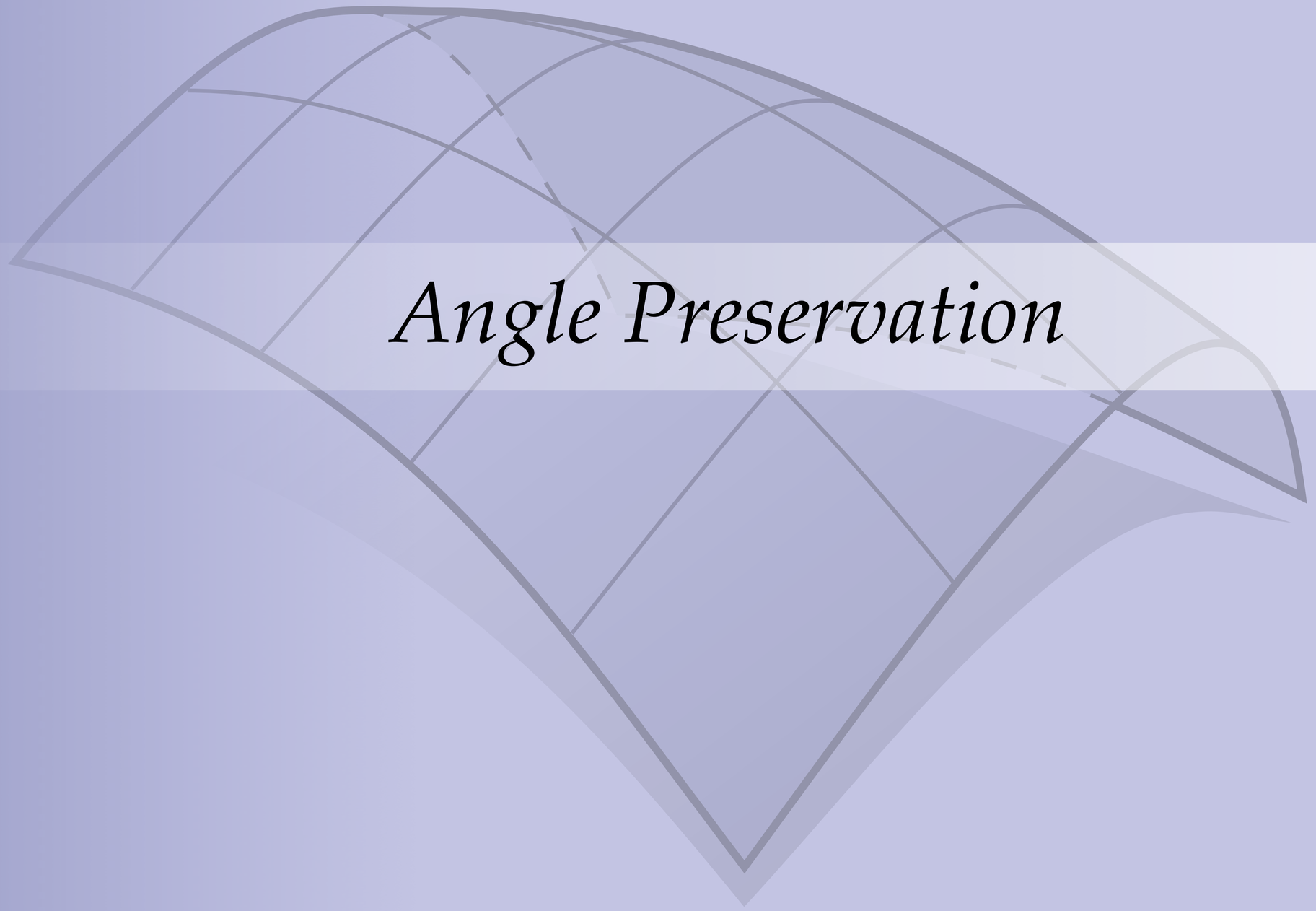
**[DEMO]**

**More sophisticated treatment:** HUTCHINSON, *"Computing Conformal Maps and Minimal Surfaces"* (1991)

# *Aside: When is a Harmonic Map Conformal?*

- When else can you play this "trick"? (*I.e.*, get a conformal map by just computing a harmonic map)
- Works for the sphere: just keep averaging w/neighbors, projecting
  - *Caveat*: may get stuck in a local minimum that is only holomorphic
  - As before, there are much more intelligent algorithms for the sphere!
- Full characterization given by Eells & Wood (1975):

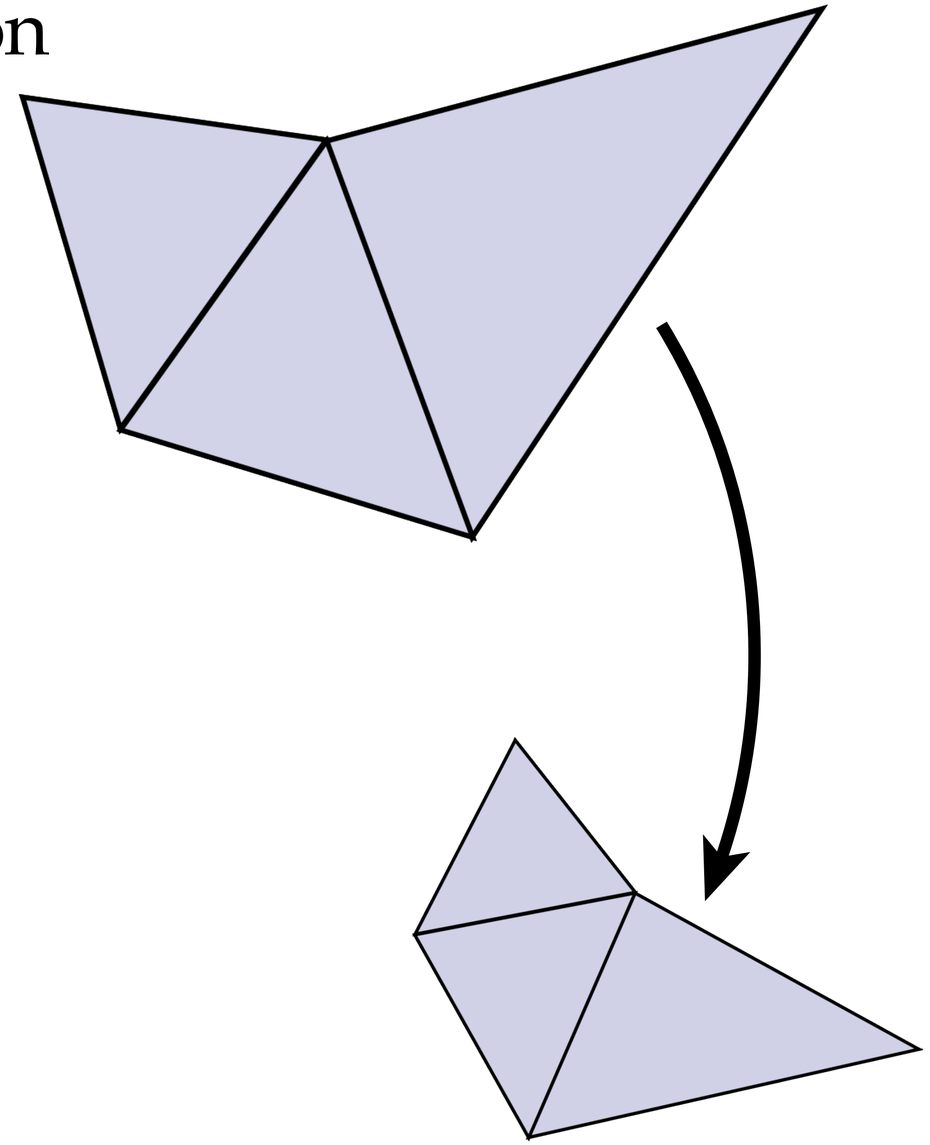
> THEOREM. *If $\varphi: X \to Y$ is a harmonic map relative to Riemannian metrics $g$ and $h$, and if $e(X) + |d_\varphi e(Y)| > 0$, then $\varphi$ is $\pm$ holomorphic relative to the complex structures determined by $g$ and $h$.*



# *Angle Preservation*

# *Angle Preservation*

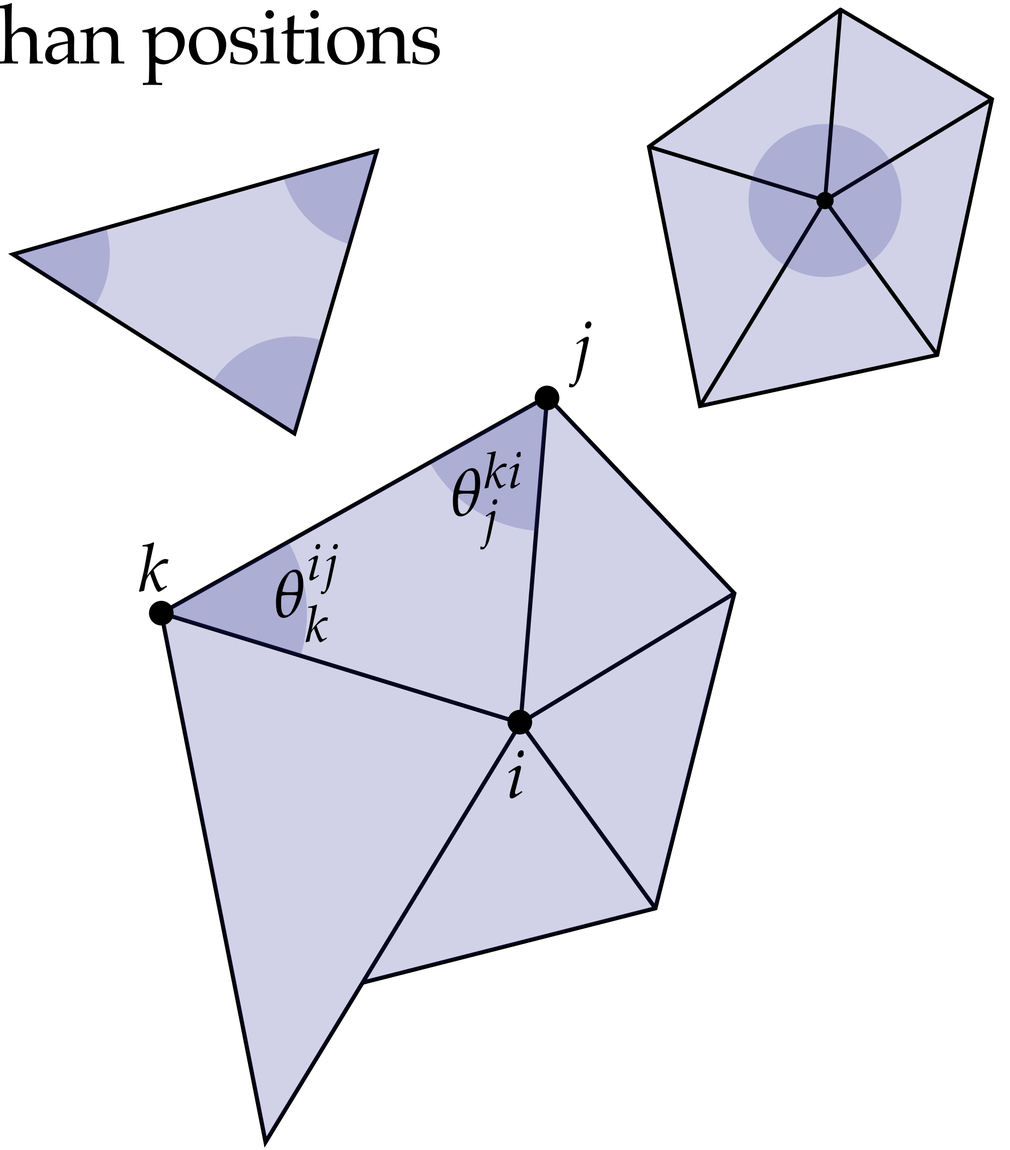
- As discussed earlier, exact angle preservation is *too rigid* (most meshes can't be flattened)
- But, can still continue down this path:
  - Find a collection of angles that describe a flat mesh
  - Approximate original angles "as well as possible"
  - Still provides good approximation of conformal map as we refine (*"discretized"*)



# *Compatibility of Angles*

- Encode flat mesh by *interior angles* rather than positions
- Must satisfy three conditions:
  1. Angles sum to $\pi$ in each triangle
  2. Sum to $2\pi$ around interior vertices
  3. Compatible lengths around vertices:

$$\prod_{ijk \in \mathrm{St}(i)} \frac{\sin \theta_j^{ki}}{\sin \theta_k^{ij}} = 1$$




**Note:** final condition is *nonlinear!*

# *Angle-Based Flattening*

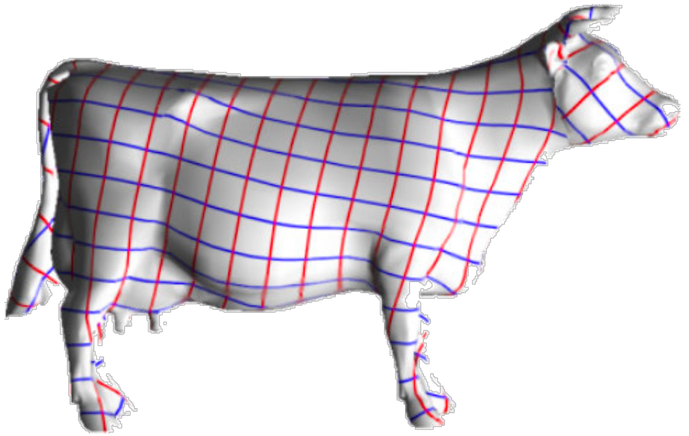
- **Given:** angles $\theta_0$ for original mesh (usually from embedding in 3-space)
- **Find:** closest angles $\theta$ that describe a flat mesh
- Compute by solving nonconvex optimization problem:

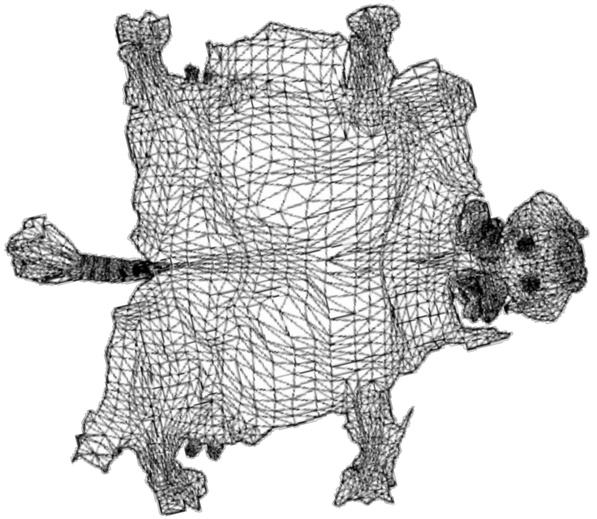
$$
\begin{aligned}
\min_{\theta \in \mathbb{R}^{3|F|}} \quad & \textstyle\sum_i (\tilde{\theta}_i - \theta_i)^2 & \\
\text{s.t.} \quad & \theta_i^{jk} + \theta_j^{ki} + \theta_k^{ij} = \pi, & \forall ijk \in F \\
& \textstyle\sum_{ijk \in \mathrm{St}(i)} \theta_i^{jk} = \pi, & \forall i \in V \\
& \textstyle\prod_{ijk \in \mathrm{St}(i)} \sin \theta_j^{ki} / \sin \theta_k^{ij} = 1, & \forall i \in V
\end{aligned}
$$

SHEFFER, DE STURLER, *"Parameterization of Faceted Surfaces for Meshing using Angle Based Flattening"* (2001)

# *Linear Angle Based Flattening*

- Original ABF problem is large, difficult to solve
- Approximate by a linear problem:
  - solve for *change* in angles that makes mesh flat
  - linearize nonlinear condition via log, Taylor series
- Results are nearly indistinguishable from original ABF

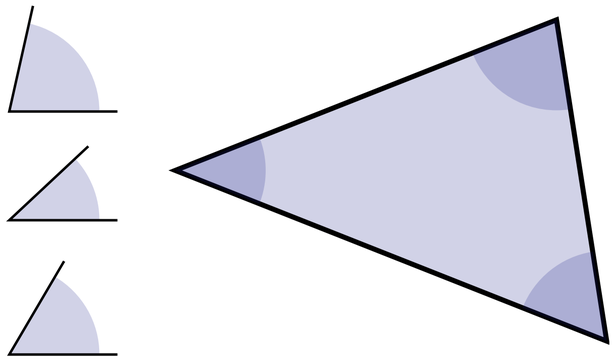


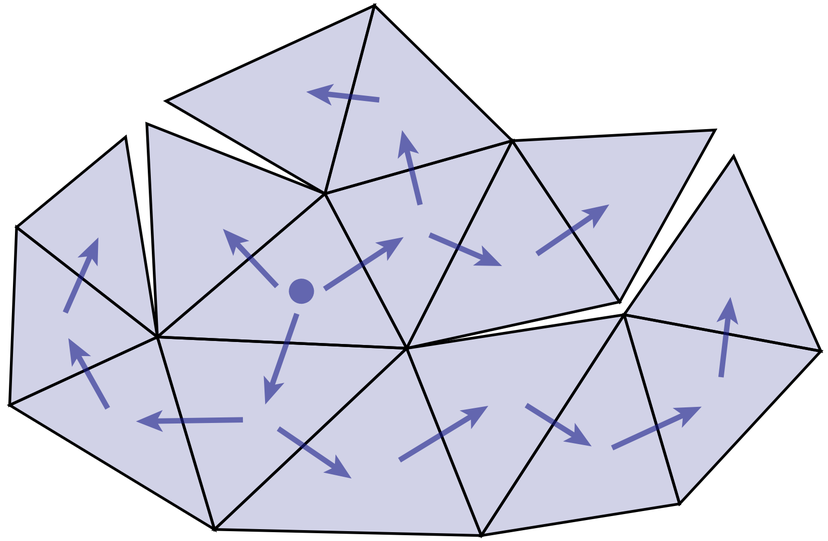


ZAYER, LÉVY, SEIDEL, *"Linear Angle Based Parameterization"* (2007)

# *Angle Layout Problem (Local Strategy)*

- **Given:** Angles that describe a flat triangulation
- **Find:** Vertex positions that exhibit these angles
- *Local strategy*: start at any triangle and "grow out"
  - first triangle determined up to scale by three angles
  - Problem: accumulation of numerical error can cause cracks

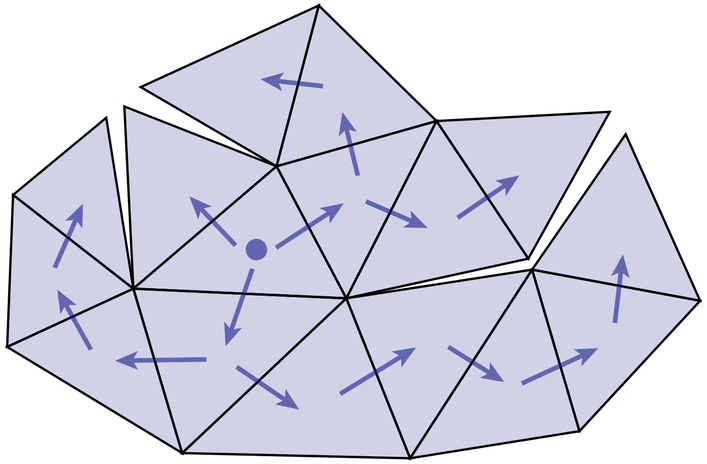




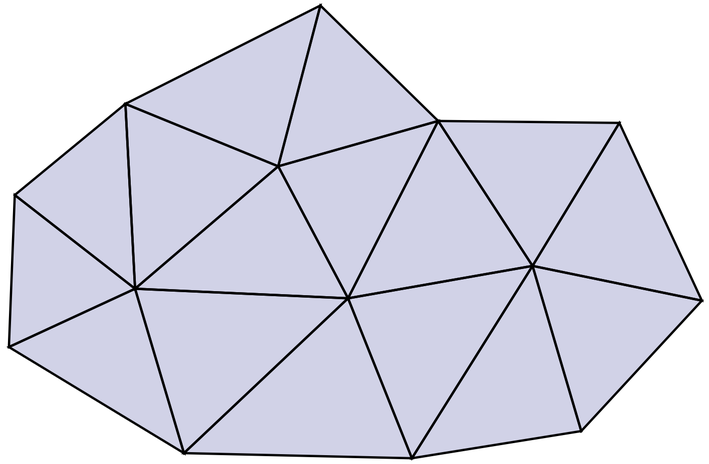
# *Angle Layout Problem (Global Strategy)*

- *Global strategy*: solve large linear system for vertex positions that best match the given angles (see ABF++)
- **Observation:** linear system is equivalent to computing edge lengths from angles, running LSCM on new edge lengths.
  - *Interpretation*: ABF++ intrinsically "deforms" metric to something nearly flat; still needs LSCM to get final (extrinsic) map to the plane

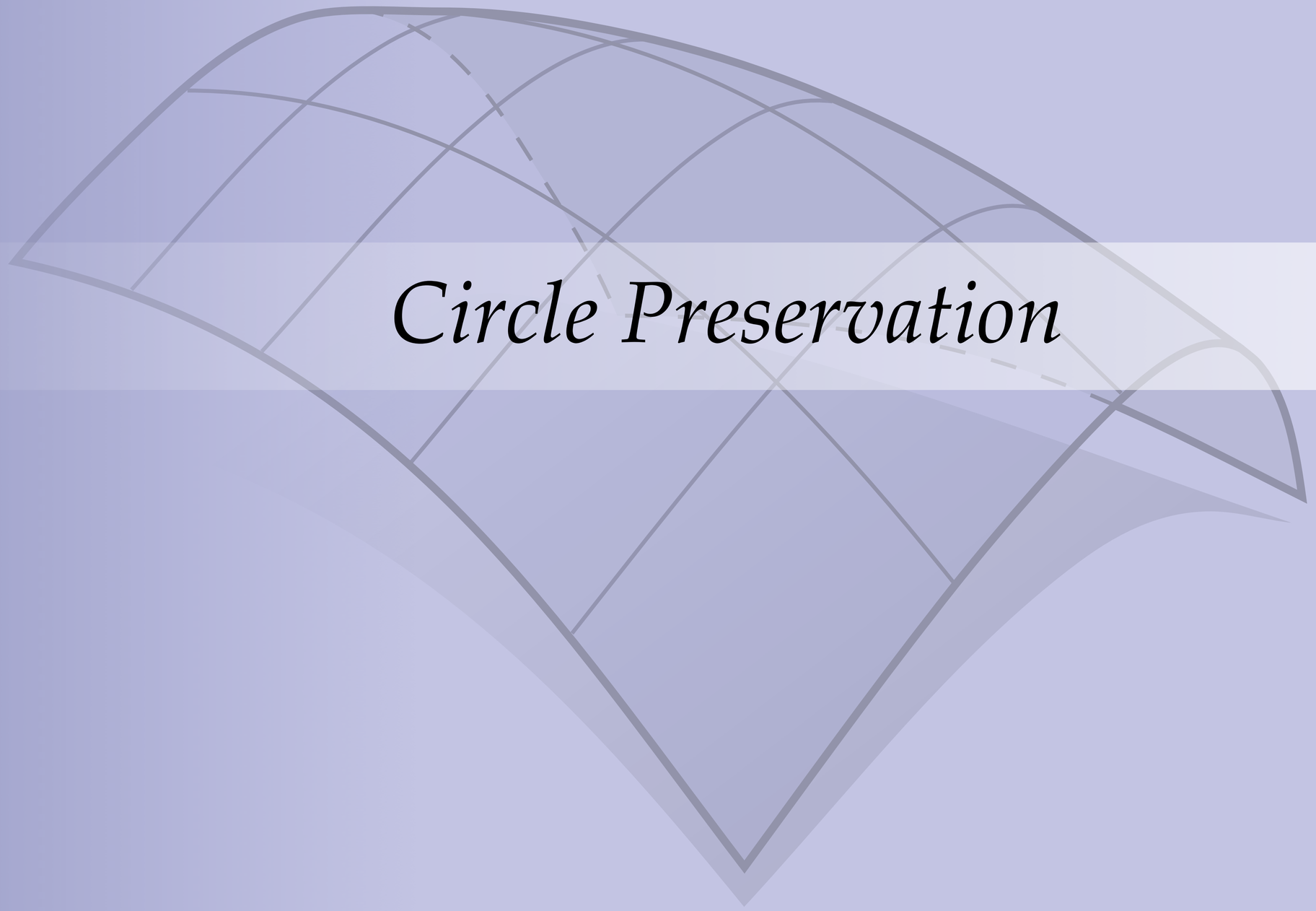
(Will see this strategy again later…)



LOCAL



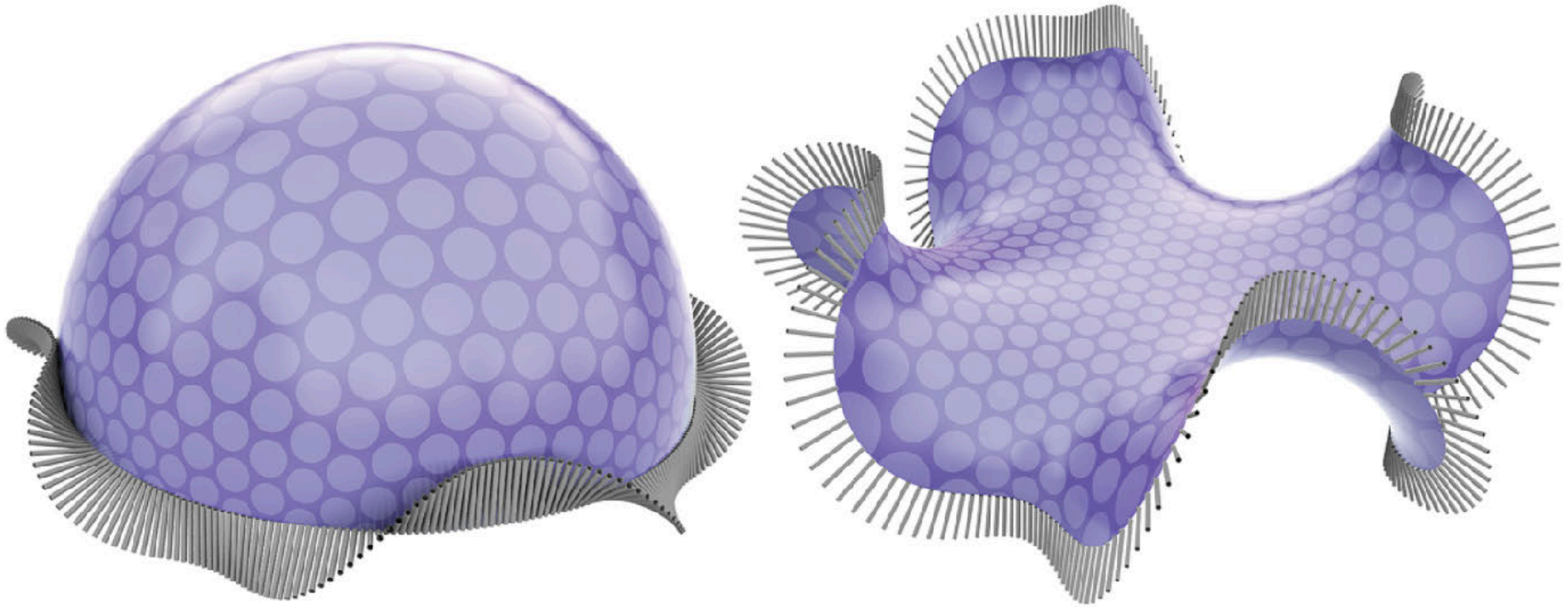
GLOBAL



# *Circle Preservation*

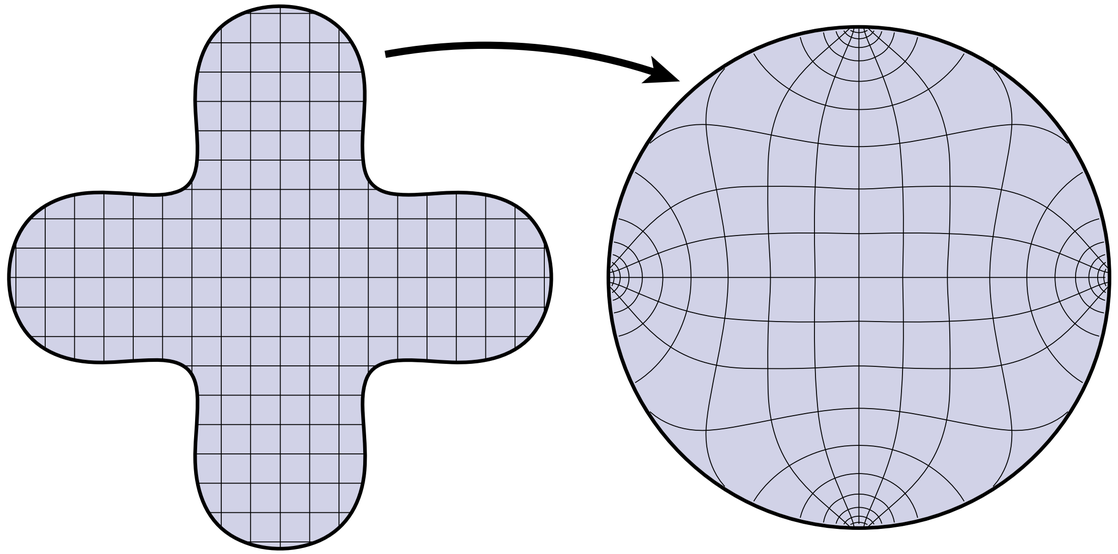
# *Circle Preservation*

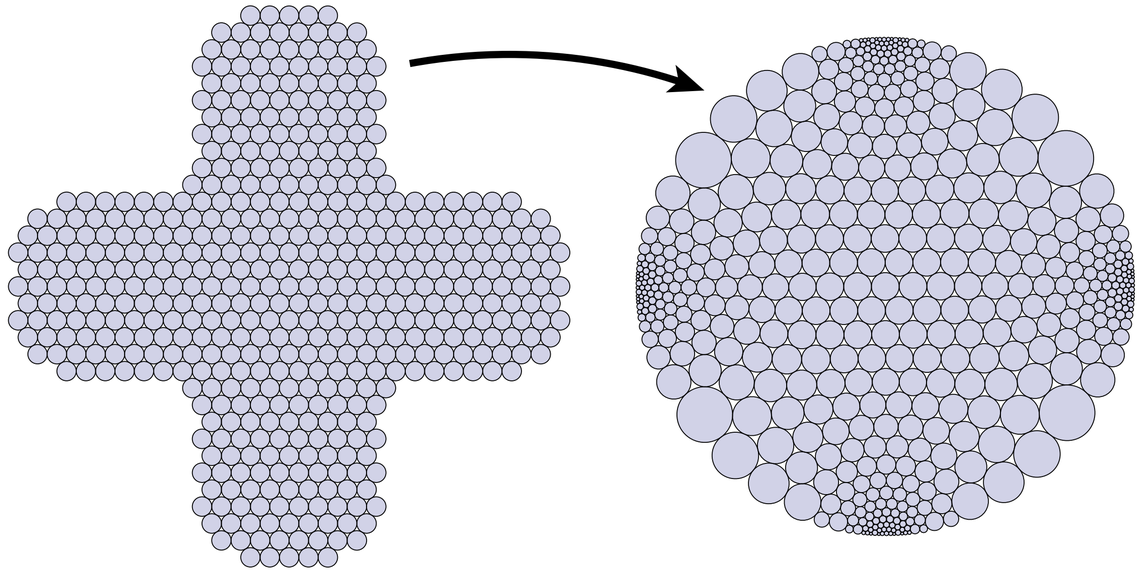
- **Smooth**: conformal maps preserve infinitesimal circles (why?)
- **Discrete**: try to preserve circles associated with mesh elements



# *Circle Packing*

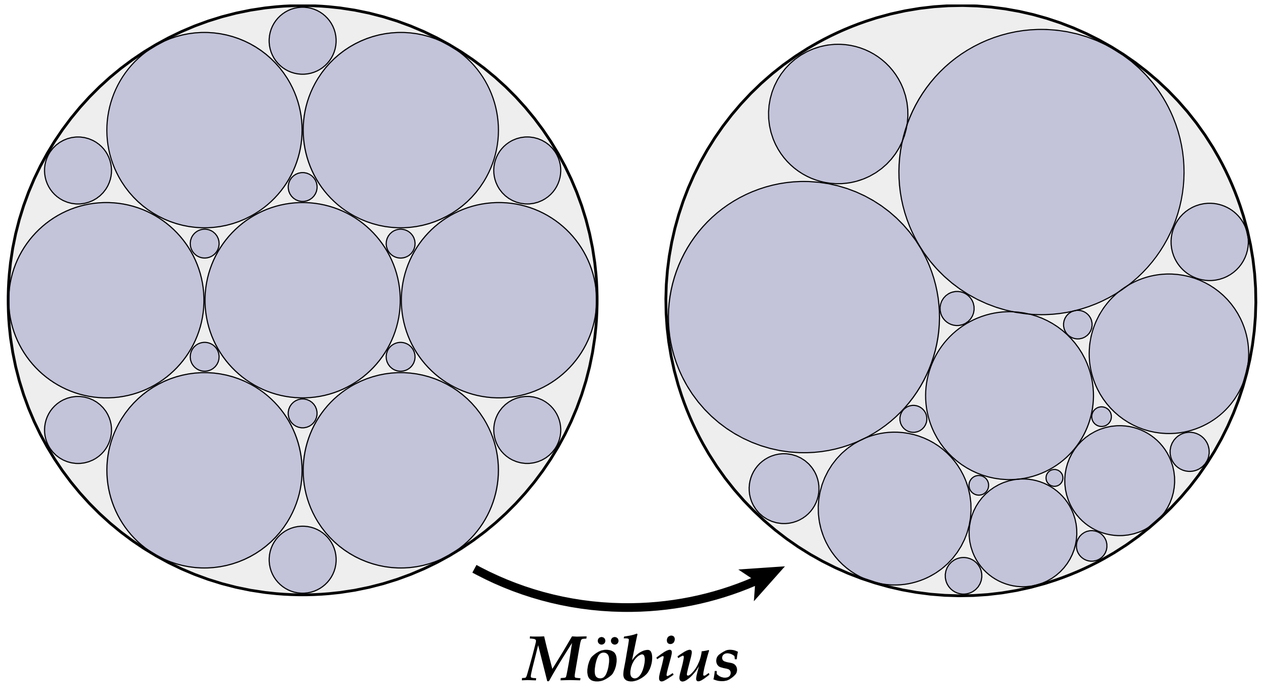
- *Koebe*: every planar graph can be realized as collection of circles
  - one circle per vertex; two circles are tangent if they share an edge
- *Thurston*: cover planar region by regular tiling of circles; now make boundary circles tangent to unit circle. This "circle packing" approximates a smooth conformal map (Rodin-Sullivan).





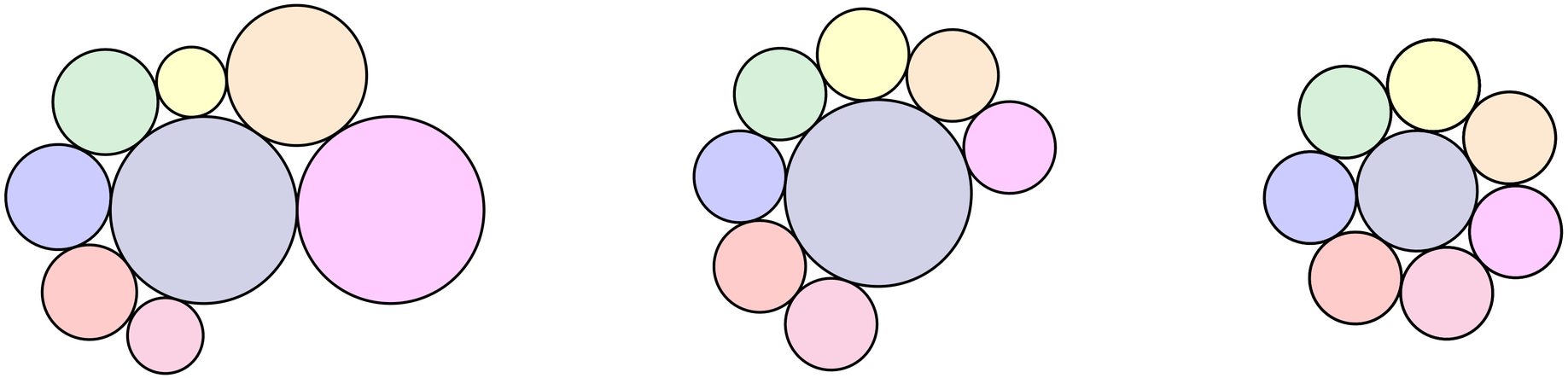
# *Circle Packing—Structure Preservation*

- Theories based on circles naturally preserve certain properties of smooth conformal maps
- *E.g.*, since Möbius transformations take circles to circles, circle packing preserves dimension of solutions to Riemann mapping



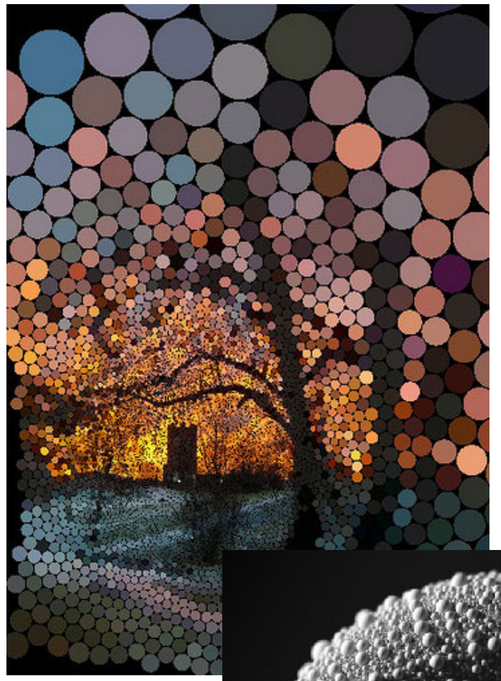
# *Circle Packing—Algorithm*

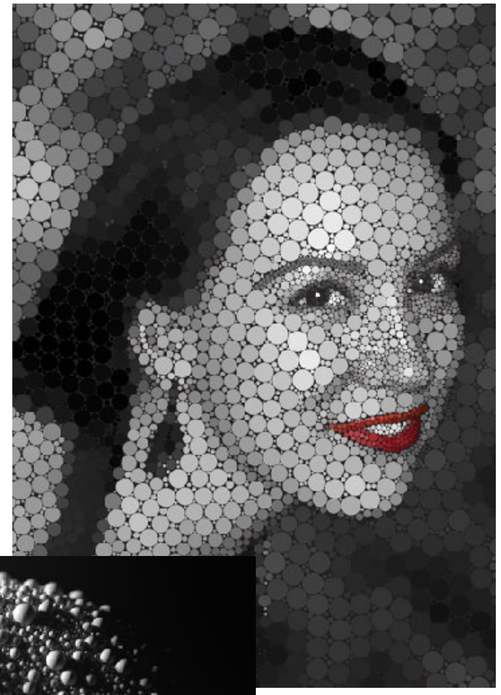
- Nonlinear problem, but simple iterative algorithm
- For each vertex $i$:
  - Let $\theta$ be total angle currently covered by $k$ neighbors
  - Let $r$ be radius such that $k$ neighbors of radius $r$ also cover $\theta$
  - Set new radius of $i$ such that $k$ neighbors of radius $r$ cover $2\pi$
- *Repeat!*

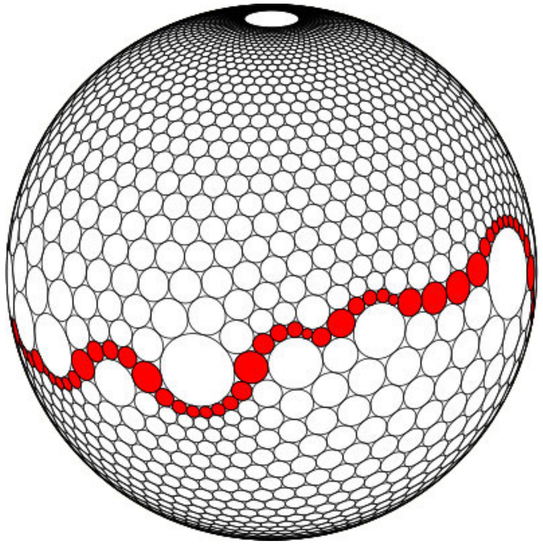


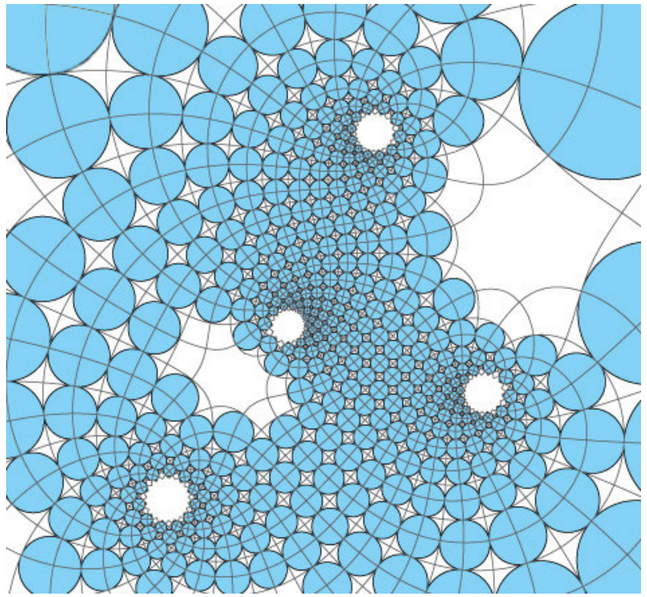
COLLINS, STEPHENSON, *"A Circle Packing Algorithm"* (2003)

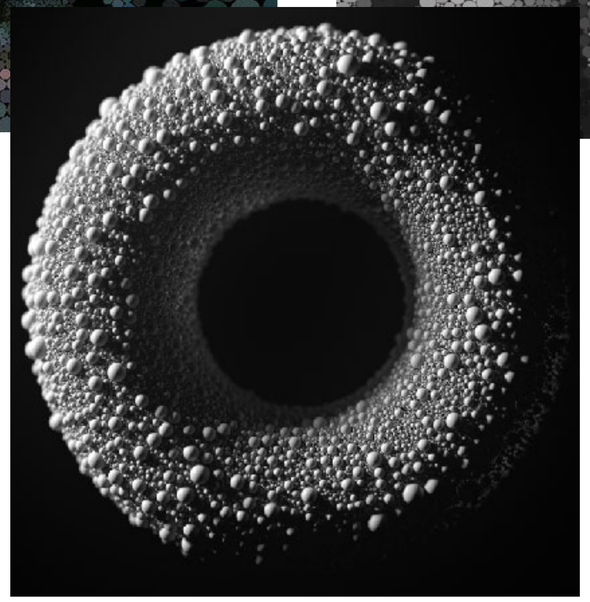
# *Circle Packing—Gallery*

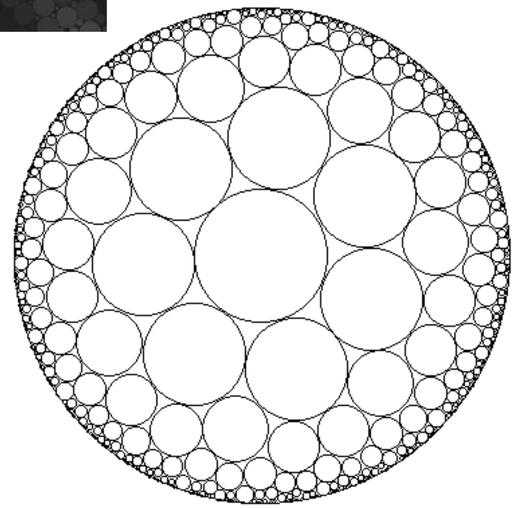


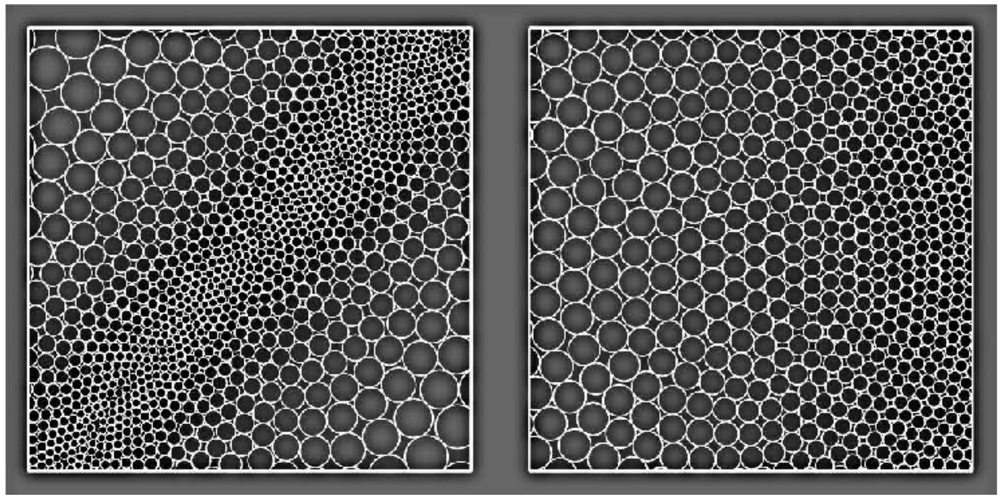






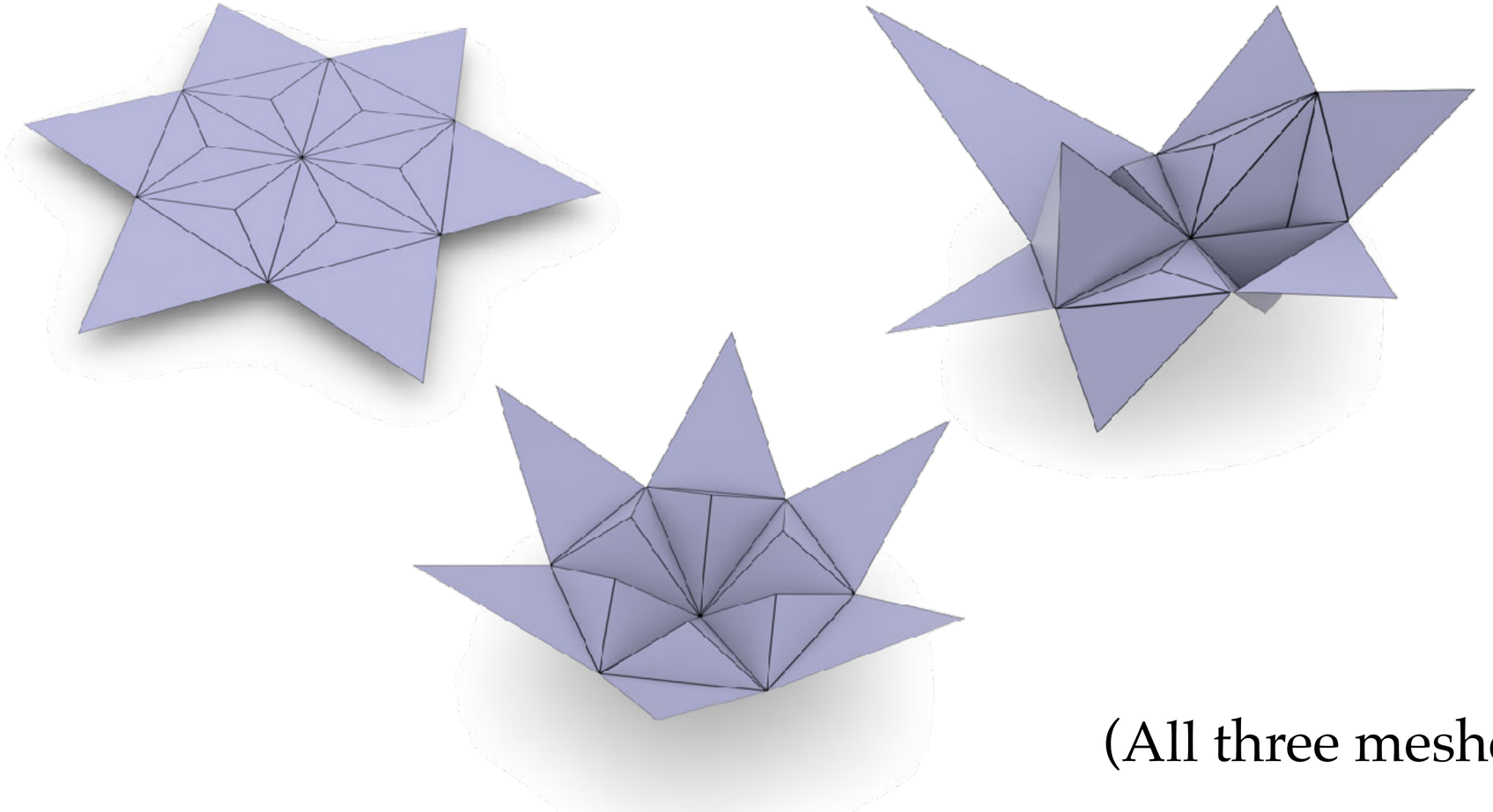


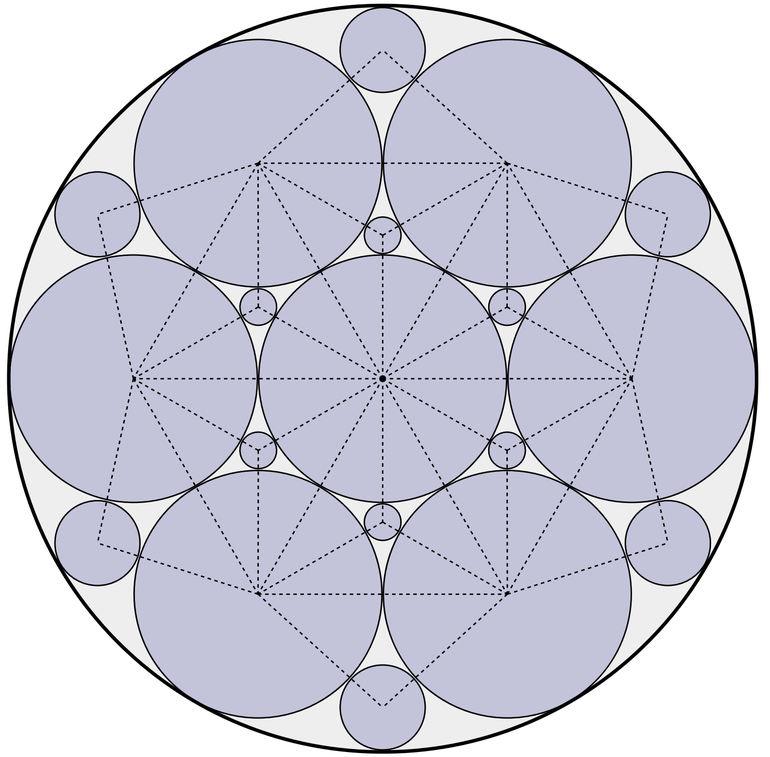




# *Circle Packings Ignore Geometry*

- Circle packing is purely combinatorial (neighboring circles are tangent)
- For *geometry* processing, need definition that incorporates *geometry!*

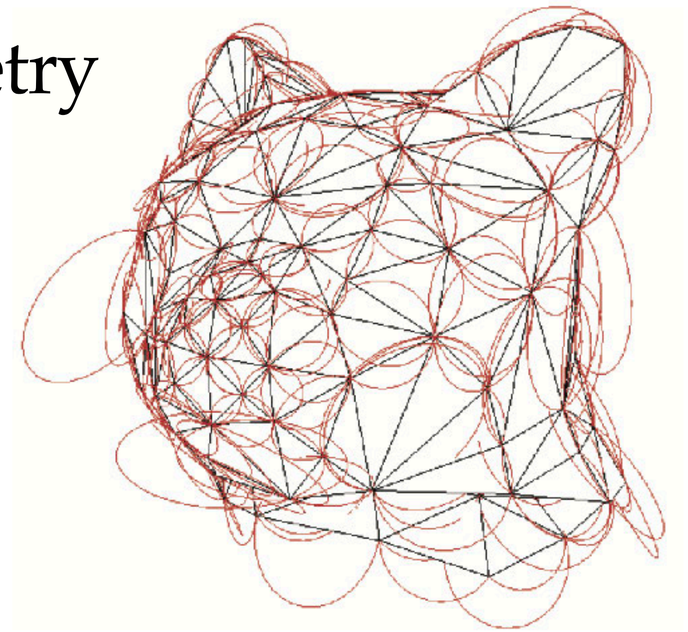


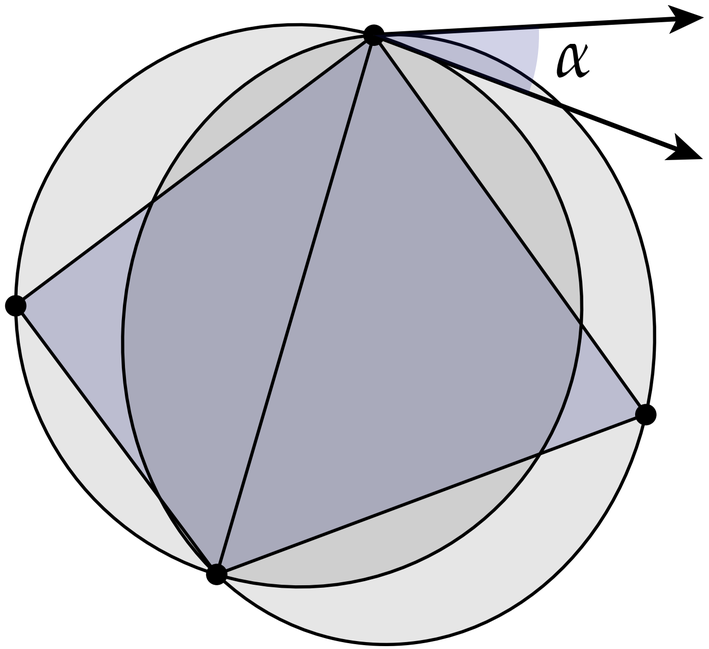


(All three meshes yield same circle packing.)

# *Circle Patterns*

- Different idea: *circle patterns*
  - associate each face with its circumcircle (circle through three vertices)
  - consider "conformal" if circle intersection angles are preserved
- Nicely incorporates geometry
- Convex optimization
- *Still rigid!* (not obvious)

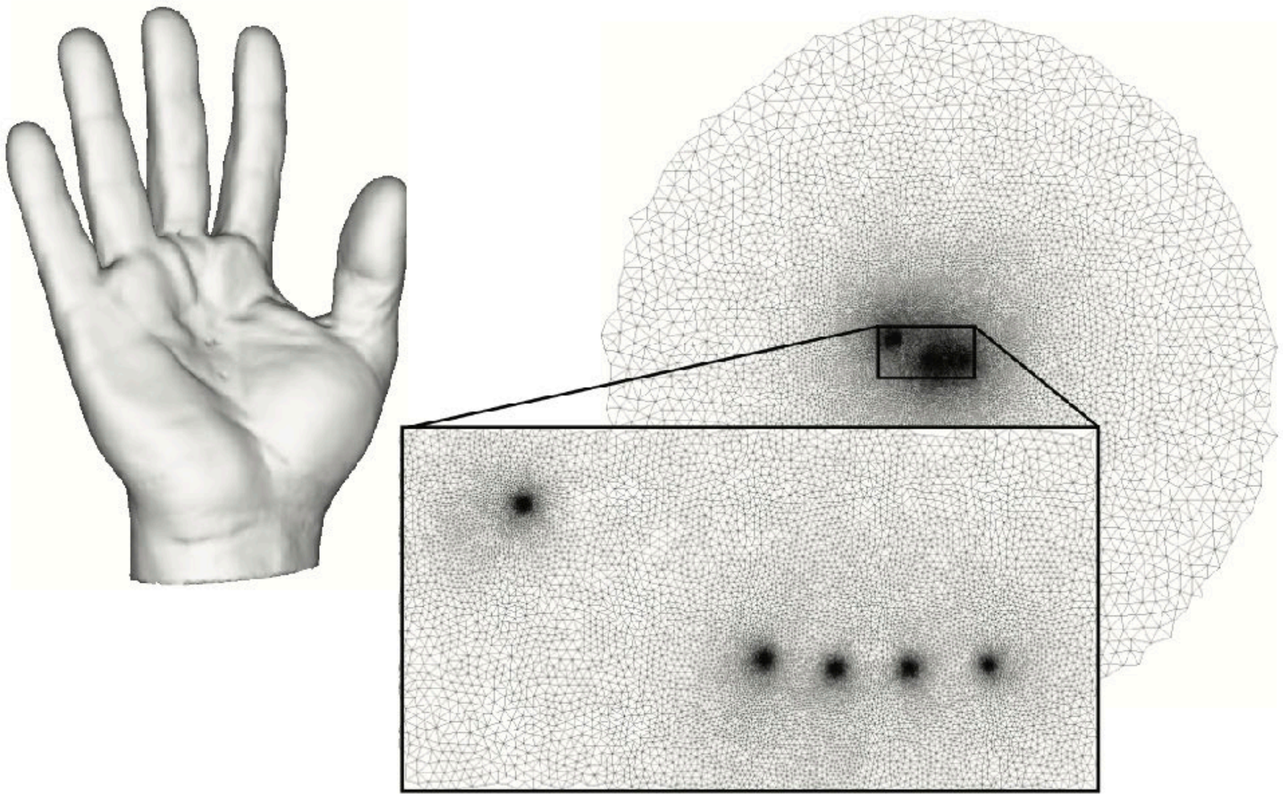


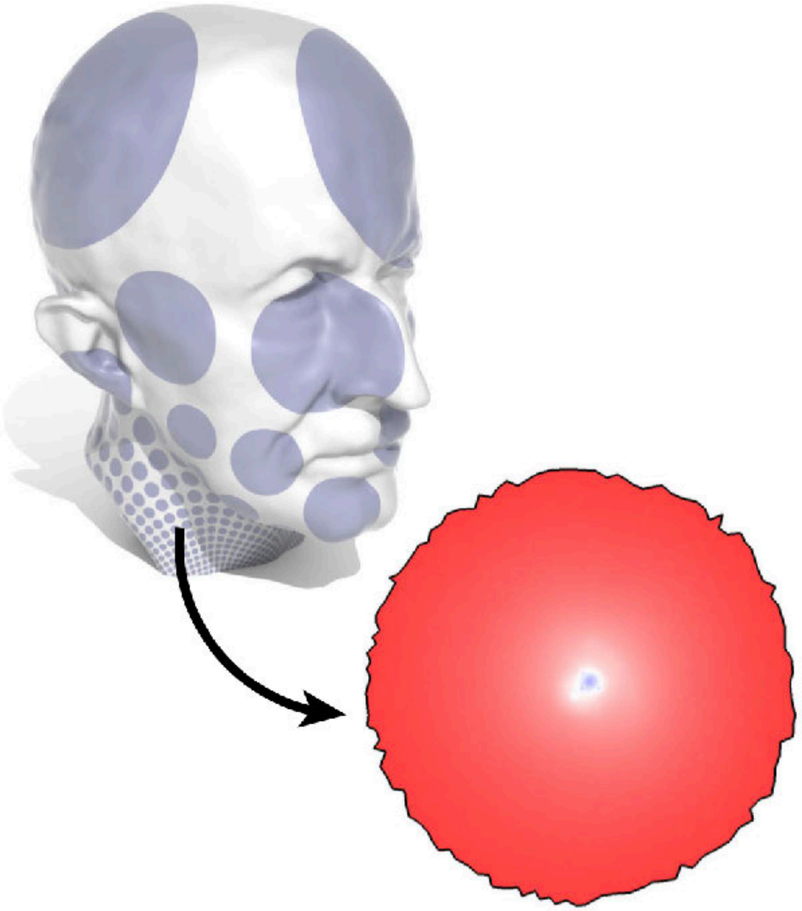



KHAREVYCH, SPRINGBORN, SCHRÖDER, *"Discrete Conformal Mappings via Circle Patterns"* (2006)

# *Cone Singularities—Motivation*

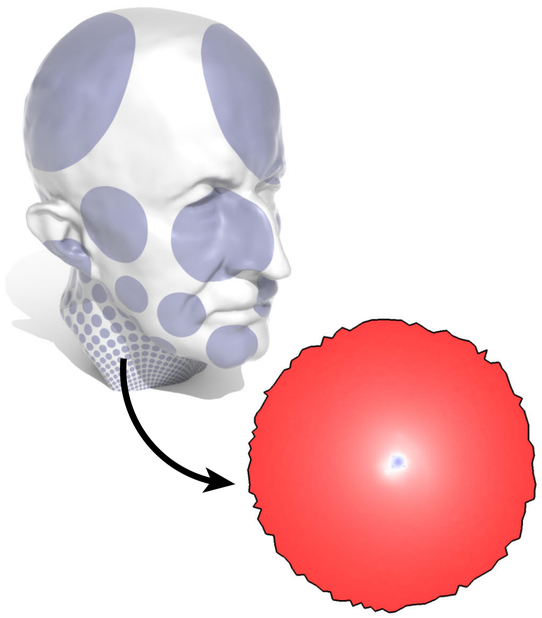
- Even in the best case, conformal flattening can exhibit significant area distortion:

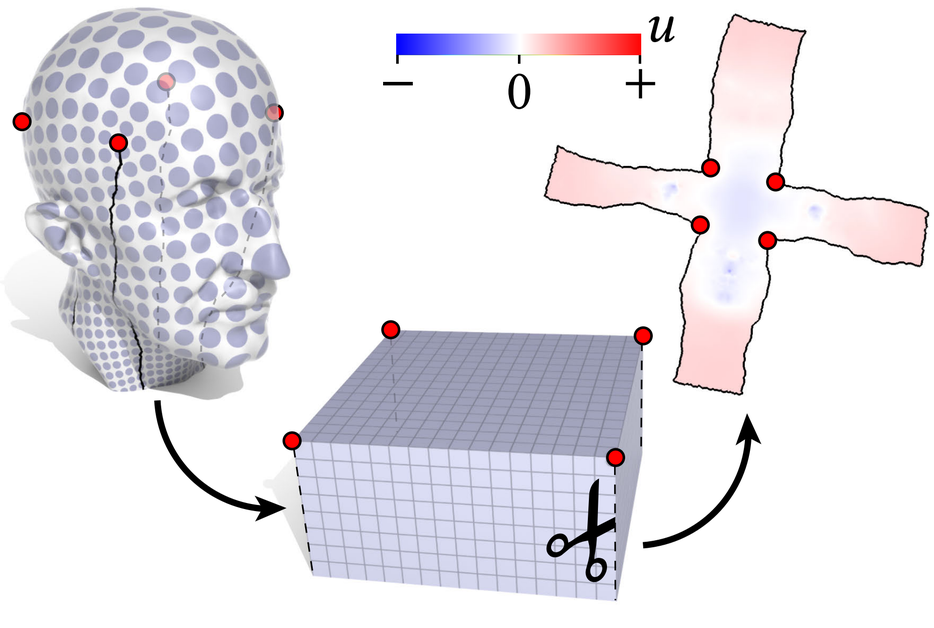




# *Cone Singularities*

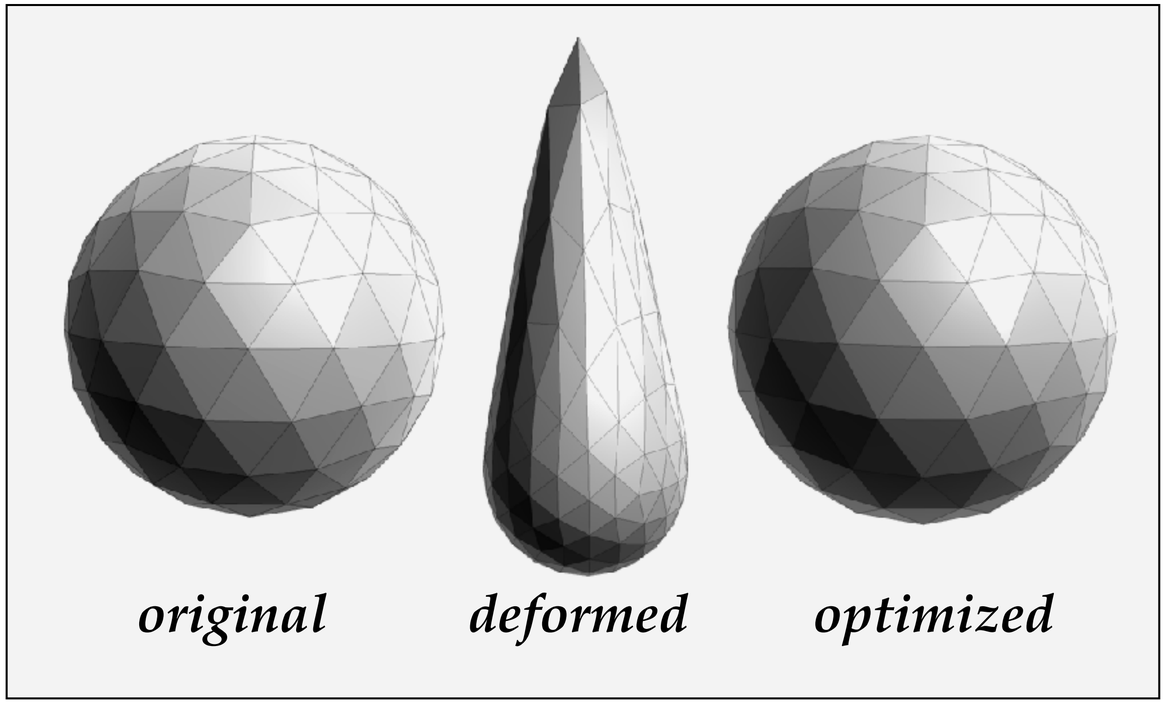
- **Idea:** *(Kharevych-Springborn-Schröder)*
  - first map to a surface that is flat except at a few "cone points"
  - then cut through cone points so that surface is flat everywhere
  - can now lay out in the plane with no additional stretching
- **Result:** lower overall area distortion (concentrated at cones)



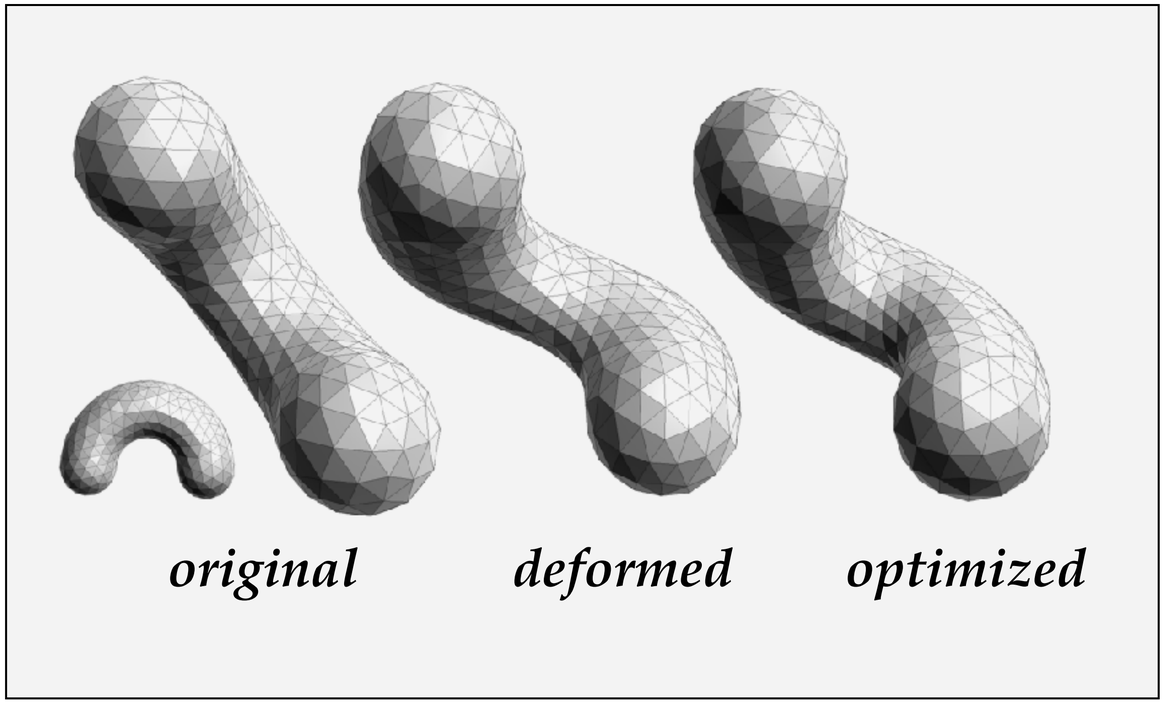


# *Rigidity of Circle Patterns*

**Experiment:** deform mesh, then find (numerically) nearby mesh with same circle intersection angles as original mesh.




**(CONVEX)**

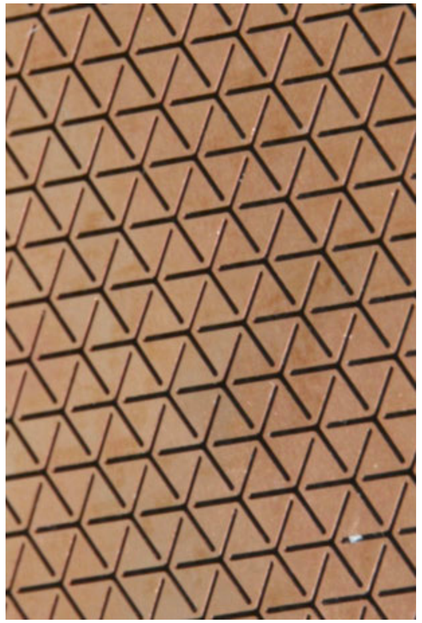



**(NONCONVEX)**

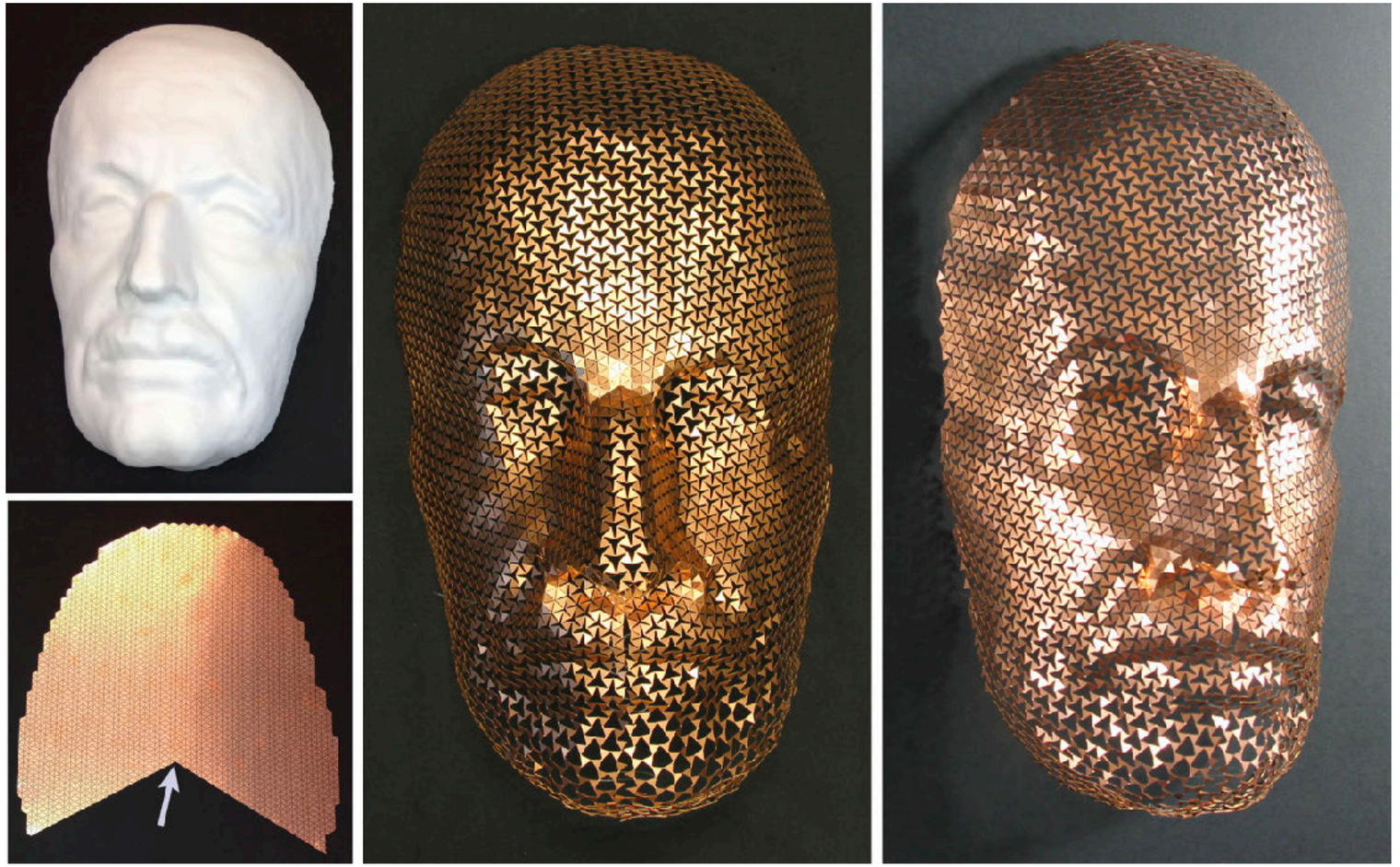
*…More flexible than angle preservation, less flexible than smooth conformal maps…*

# *Cone Singularities in Auxetic Design*

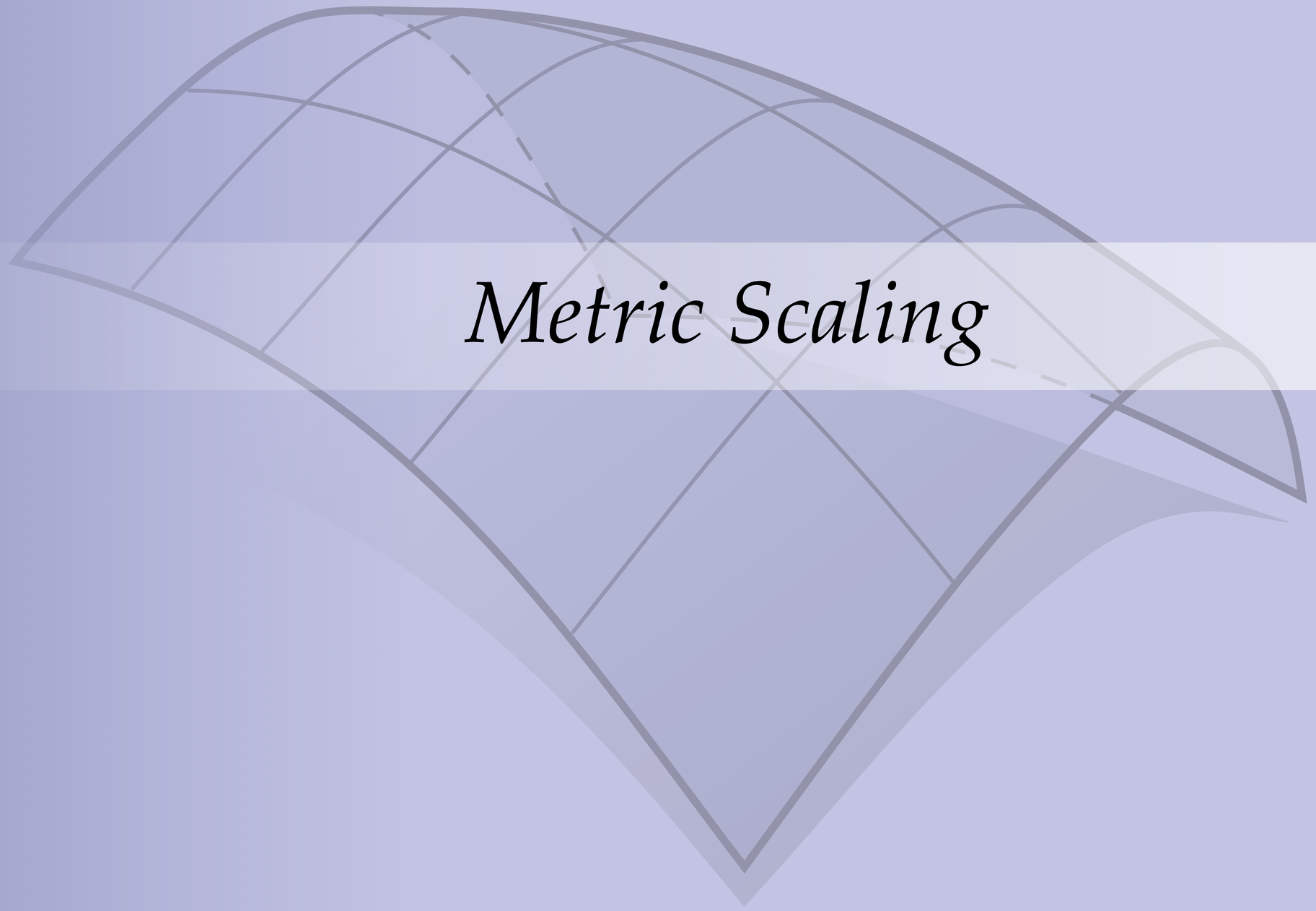
- Useful for manufacturing from materials with limited ability to stretch:



*(laser cut copper)*



KONAKOVIC, CRANE, DENG, BOUAZIZ, PIKER, PAULY, *"… Computational Design … with Auxetic Materials"* (2016)



# *Metric Scaling*

# *Discrete Conformal Flattening*

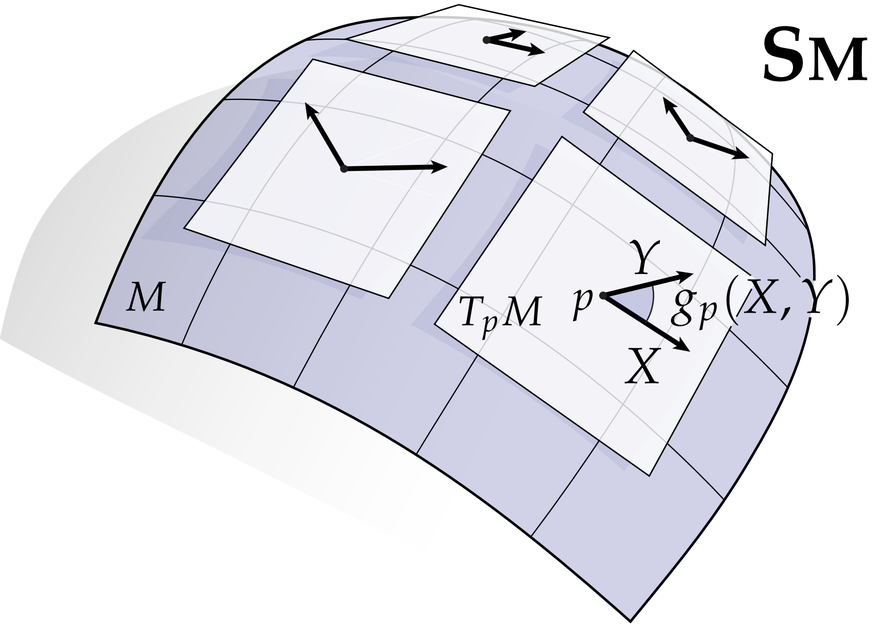
- Recall that two metrics are conformally equivalent if…

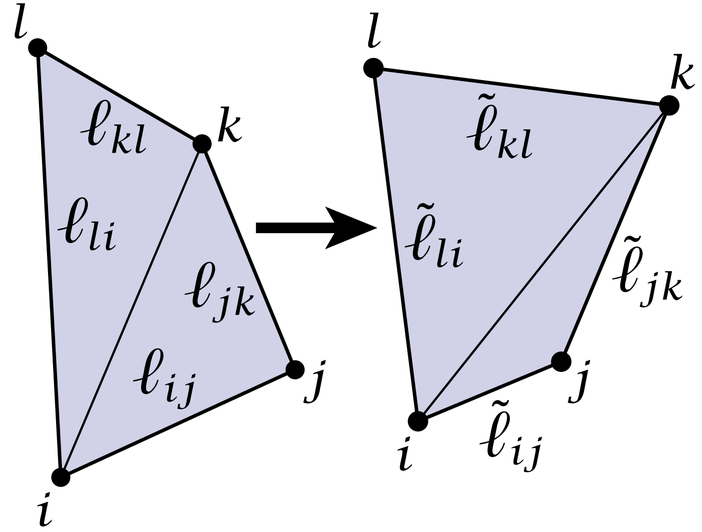
$$\tilde{g} = e^{2u} g$$

**SMOOTH**

$$\tilde{\ell}_{ij} = e^{(u_i + u_j)/2} \ell_{ij}$$

**DISCRETE**



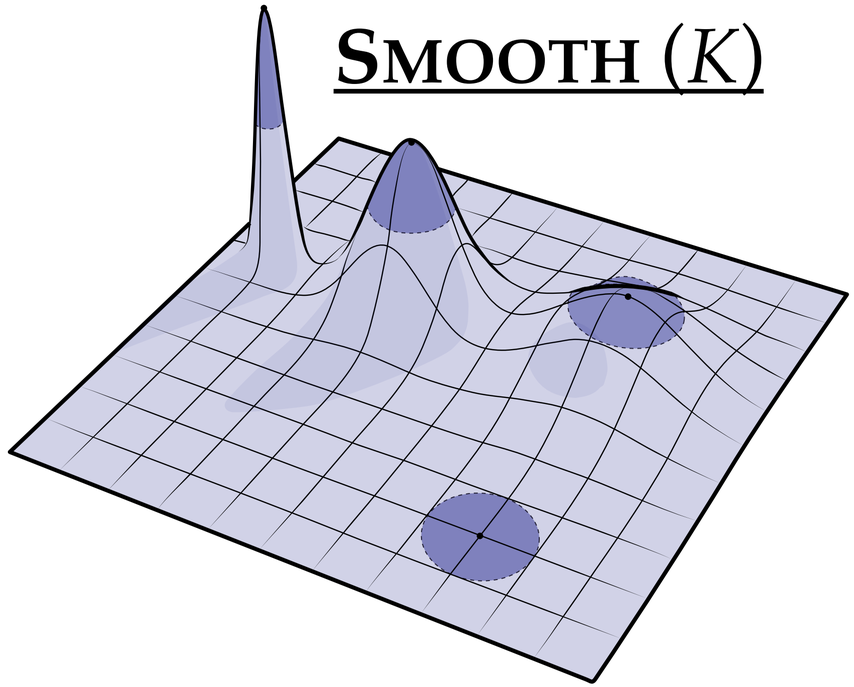




How do we *compute* a flattening that is conformally equivalent in this sense?

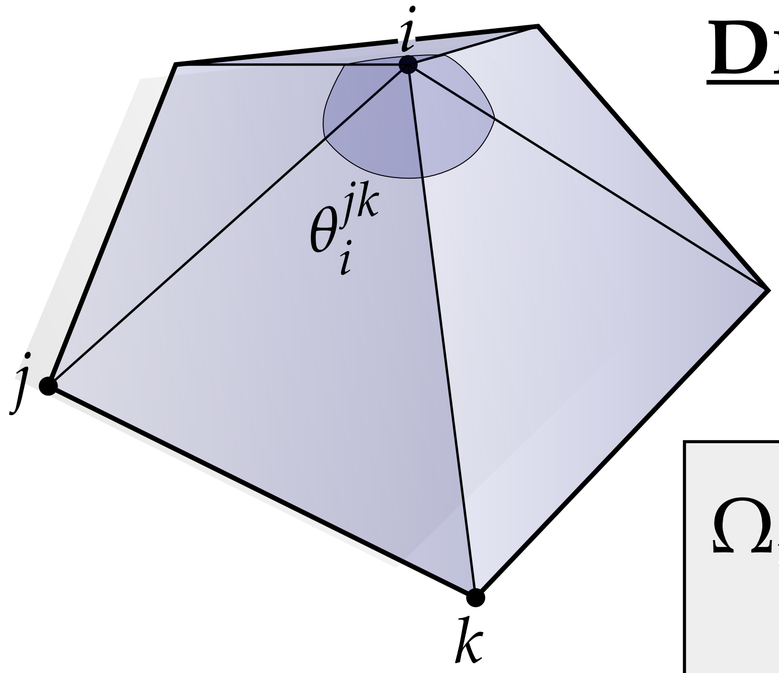
# *(Discrete) Gaussian Curvature*

- Useful to take a moment to say what we mean by "flat"!
- *Gaussian curvature* $K$ measures how hard it is to flatten a piece of material
- *Discrete* Gaussian curvature is just deviation from planar angle sum $2\pi$:

**SMOOTH** ($K$)



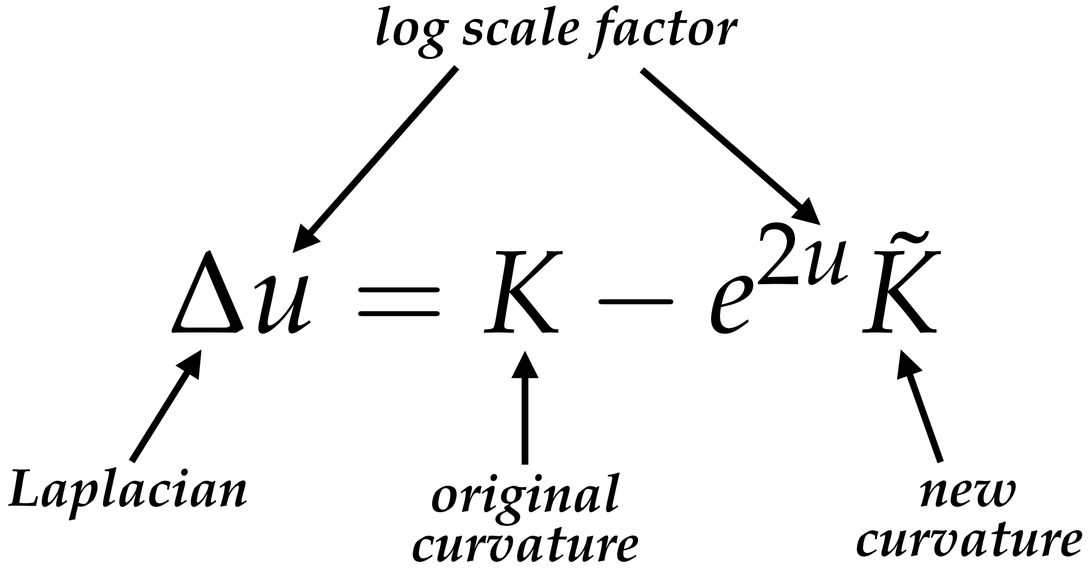
**DISCRETE**




$$\Omega_i := 2\pi - \sum_{ijk \in F} \theta_i^{jk}$$

# *Yamabe Problem*

- In the smooth setting, the *Yamabe equation* gives an explicit relationship between a conformal scaling of the metric, and the change in Gaussian curvature:



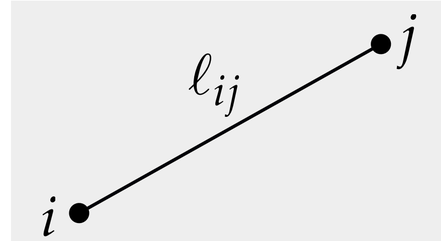

- Nonlinear due to $e^{2u}$ term on right-hand side; hard to solve directly.

# *Discrete Yamabe Flow*

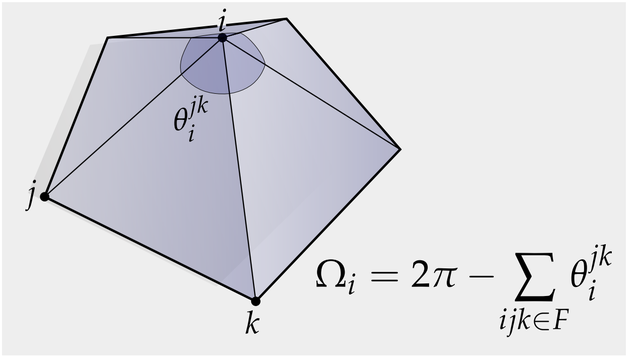
- Instead, flow toward scale factors that give desired curvature
- *Discrete case*: scale factors determine new lengths, which determine new angles, which determine angle defect
- *Basic idea*: differentiate curvature with respect to $u$
- End up with so-called (discrete) *Yamabe flow:*

$$\boxed{\frac{d}{dt}u(t) = \Omega^* - \Omega(t)}$$

- (Here for *any* target curvature $\Omega^*$, not just flat)




$$\tilde{\ell}_{ij} = e^{(u_i+u_j)/2}\ell_{ij}$$




$$\Omega_i = 2\pi - \sum_{ijk \in F} \theta_i^{jk}$$

# *CETM Algorithm*

- Flow can also be interpreted as a gradient of convex energy
  - Hessian of this energy is infamous *"cotan Laplacian"*
- Makes the flow more practical for geometry processing algorithms
- Sophisticated control over boundary shape, cone singularities, *etc.*

SPRINGBORN, SCHRÖDER, PINKALL, *"Conformal Equivalence of Triangle Meshes"* (2008)

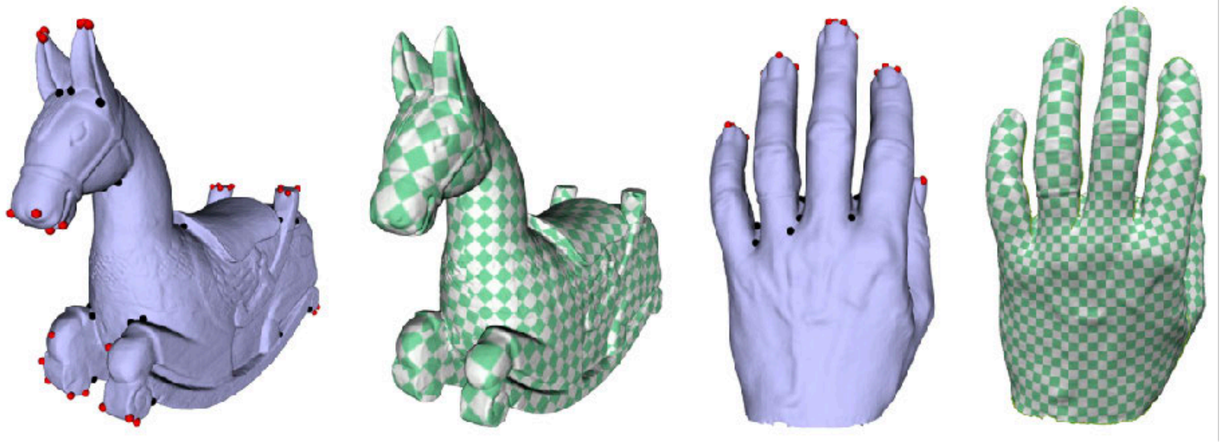
# *Curvature Prescription & Metric Scaling (CPMS)*

- *Alternatively*: linearize Yamabe equation and solve in one step:

***assume log factor is fixed, or zero***

$$\Delta u = K - e^{2u}\tilde{K}$$

- Reasonable assumption when target curvature describes *cone metric*.



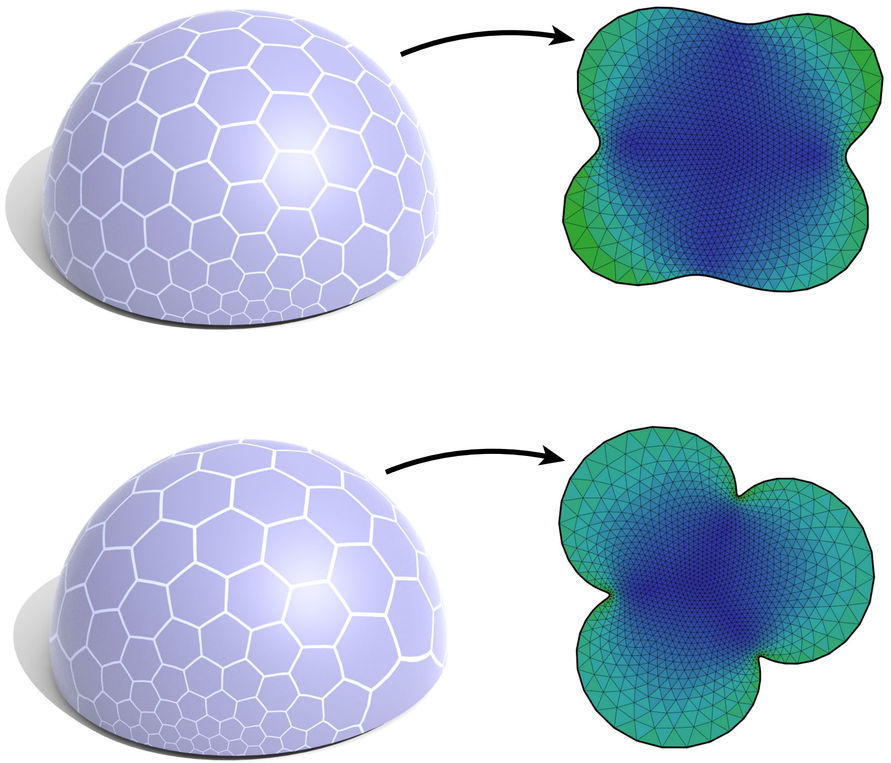
BEN-CHEN, GOTSMAN, BUNIN, *"Conformal Flattening by Curvature Prescription and Metric Scaling"* (2008)

# *Cherrier Formula*

- Yamabe equation was actually incomplete—what happens at the boundary?
- Answer given by *Cherrier equation*

$$\begin{aligned} \Delta u &= K - e^{2u}\tilde{K} \quad \text{on } M \\ \frac{\partial u}{\partial n} &= \kappa - e^{u}\tilde{\kappa} \quad \text{on } \partial M \end{aligned}$$

- Implies we can prescribe ***either*** the curvature $\kappa$ ***or*** the scale factor $u$ along the boundary—*but not both!*



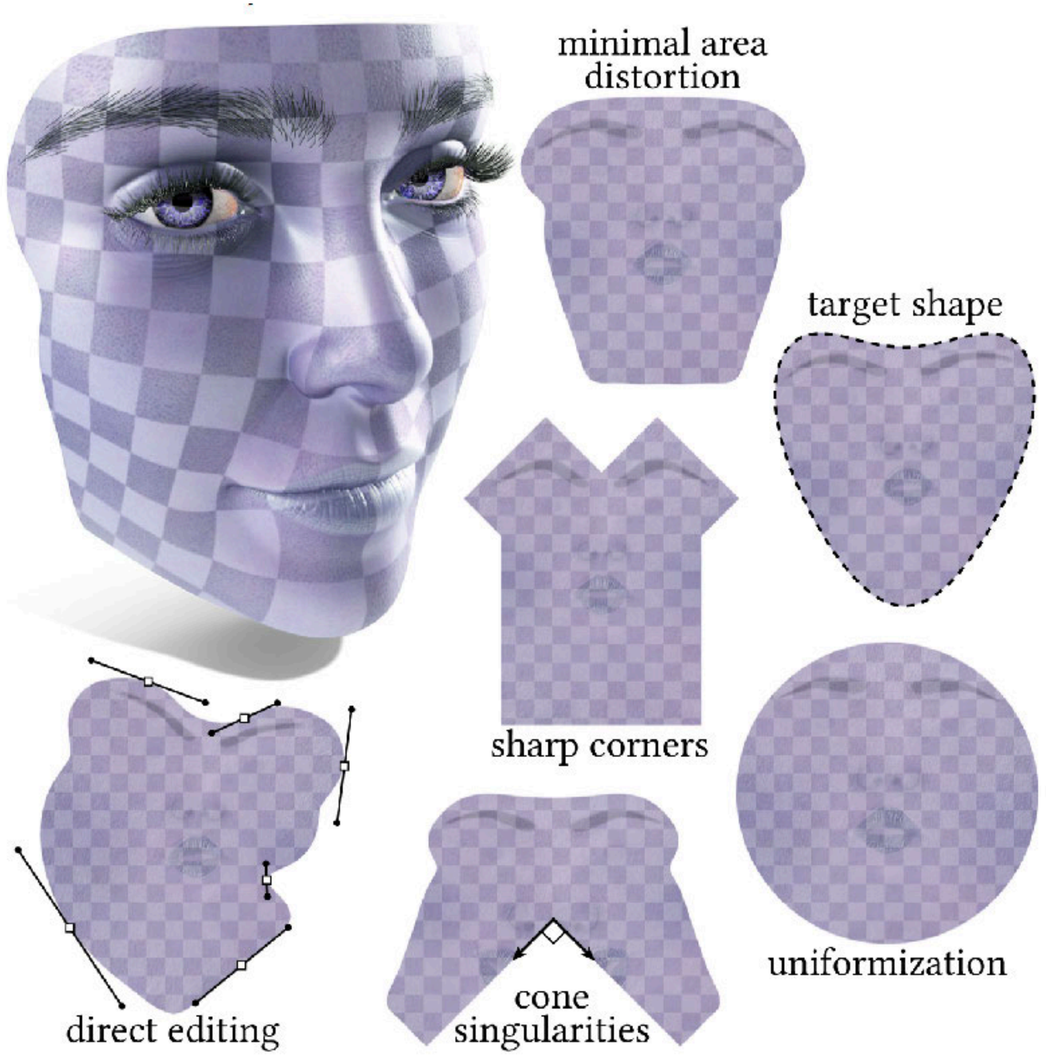
CHERRIER, *"Problèms de Neumann non linéaires sur les variétés Riemanniennes"* (1984)

# *Boundary First Flattening (BFF)*

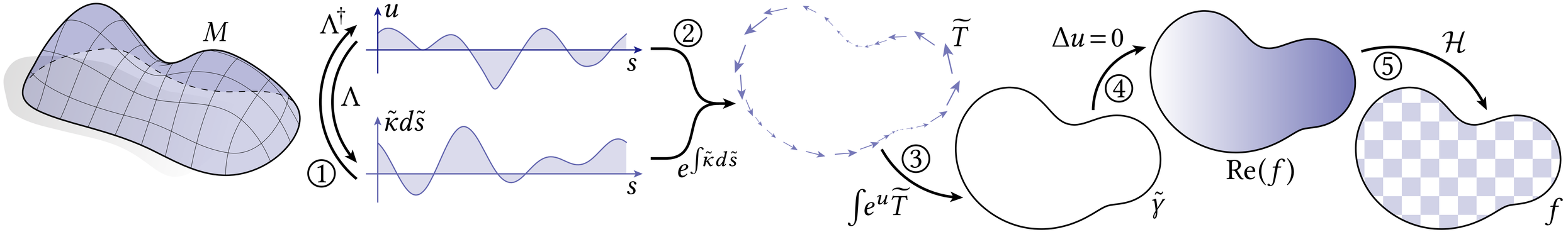
- Brand new algorithm (2017) based on Cherrier plus some other tricks…
- Complete control over boundary shape
- Faster than LSCM; *much* faster than CETM (but with comparable quality)
- Lots of bonus features (optimal area distortion, cone singularities, …)

https://arxiv.org/abs/1704.06873

**[DEMO]**



# *Boundary First Flattening—Rough Outline*




- Given a surface, specify either length ***or*** curvature of target curve
- Solve *Cherrier problem* to get complementary data (curvature or length)
- Integrate boundary data to get boundary curve
- Extend boundary curve to a pair of *conjugate harmonic functions*

# *From Cauchy-Riemann to Conjugate Harmonic*

- Starting with Cauchy-Riemann:

$$df(\mathcal{J}X) = \imath df(X)$$

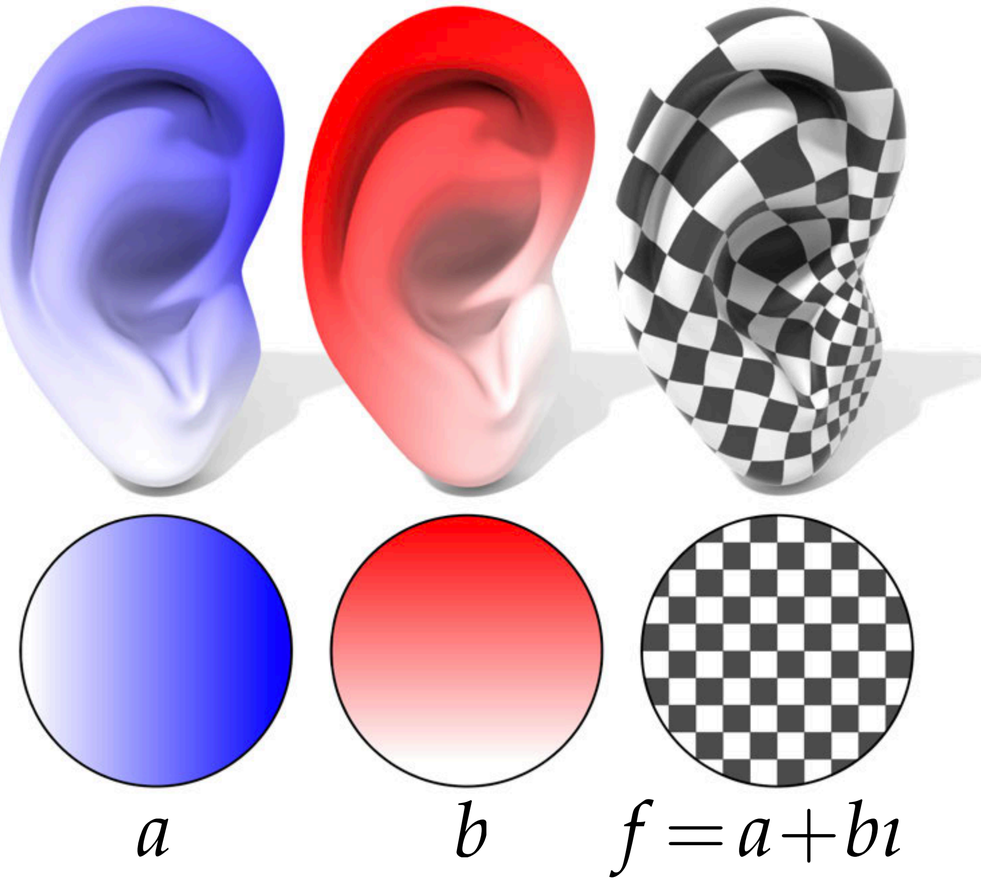
$$da(\mathcal{J}X) + \imath db(\mathcal{J}X) = \imath da(X) - \imath db(X)$$

$$\nabla a = -J\nabla b$$

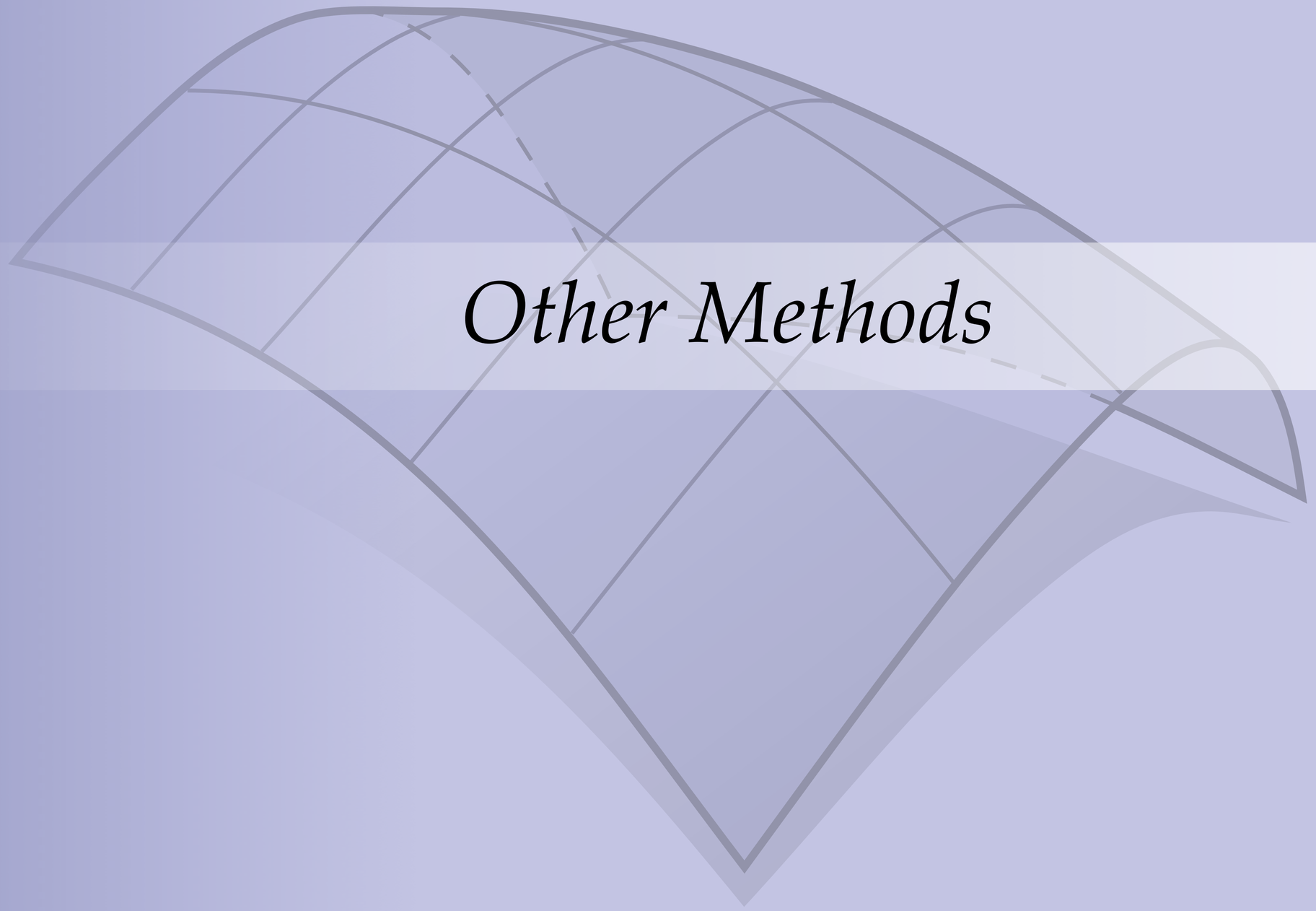
$$\underbrace{\nabla \cdot \nabla}_{\Delta} a = -\underbrace{\nabla \cdot (\nabla b)}_{=0}$$

$$\begin{aligned} \Delta a &= 0 \\ \Delta b &= 0 \\ \mathcal{J}\nabla a &= \nabla b \end{aligned}$$

**CONJUGATE HARMONIC PAIR**




*(How do you conjugate a piecewise linear function? See BFF paper!)*

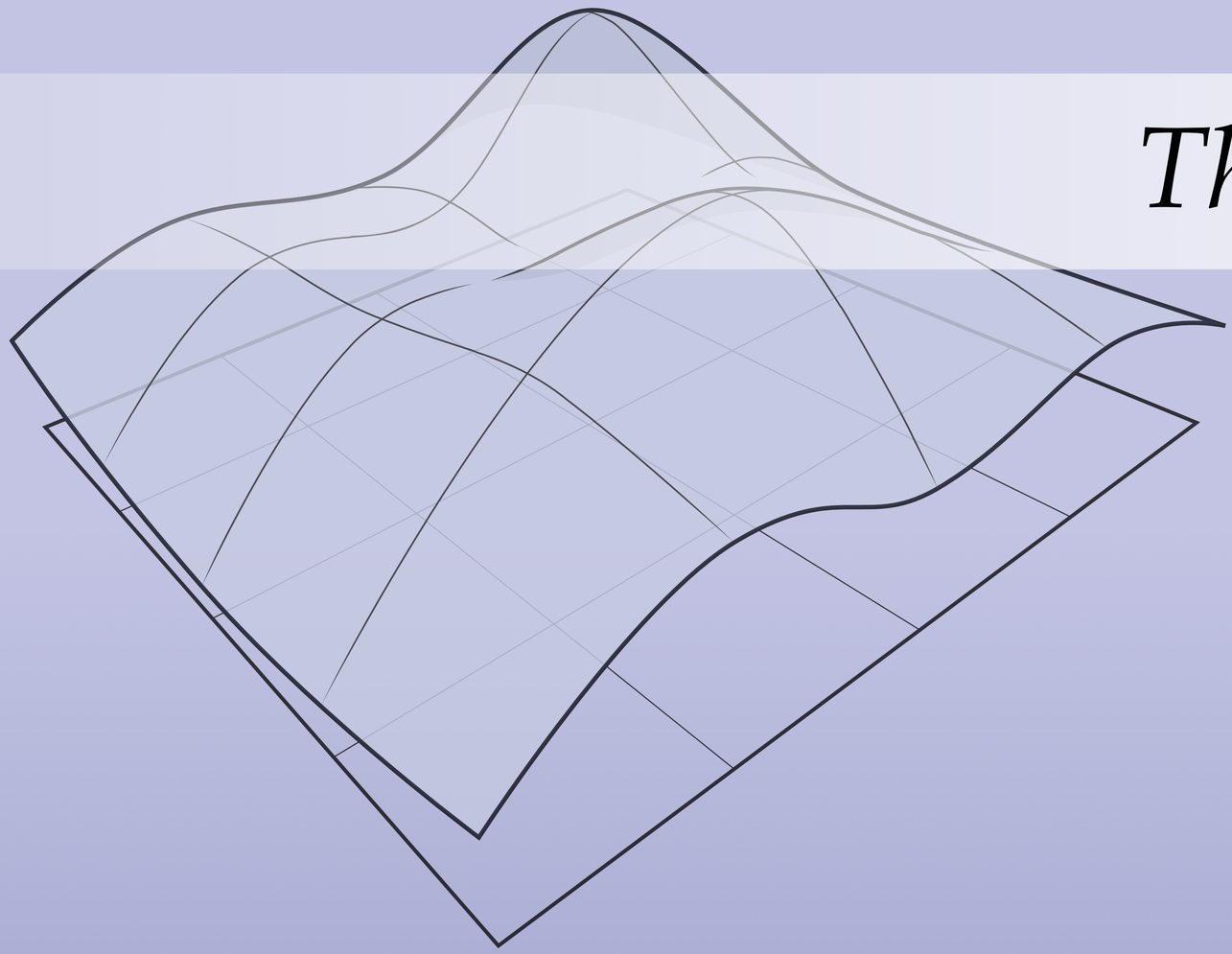


# *Other Methods*

# *So much more!*

- **Many** ideas/algorithms we didn't cover…
  - in plane: *Schwarz-Christoffel, Cauchy-Green coordinates, …*
  - inversive distance [Guo et al 2009]
  - primal-dual length ratio / discrete Riemann surfaces [Mercat 2001]
  - facewise Möbius transformations [Vaxman et al 2015]
  - in the plane: *Schwarz-Christoffel, Cauchy-Green coordinates, …*
- Also, didn't get to see many of the (*beautiful!*) things people are doing with conformal maps…

*Thanks!*



# Discrete Differential Geometry: An Applied Introduction

**Keenan Crane • CMU 15-869(J) • Spring 2016**