中国科学技术大学
University of Science and Technology of China

GAMES102

GAMES 102在线课程

几何建模与处理基础

刘利刚

中国科学技术大学

中国科学技术大学
University of Science and Technology of China

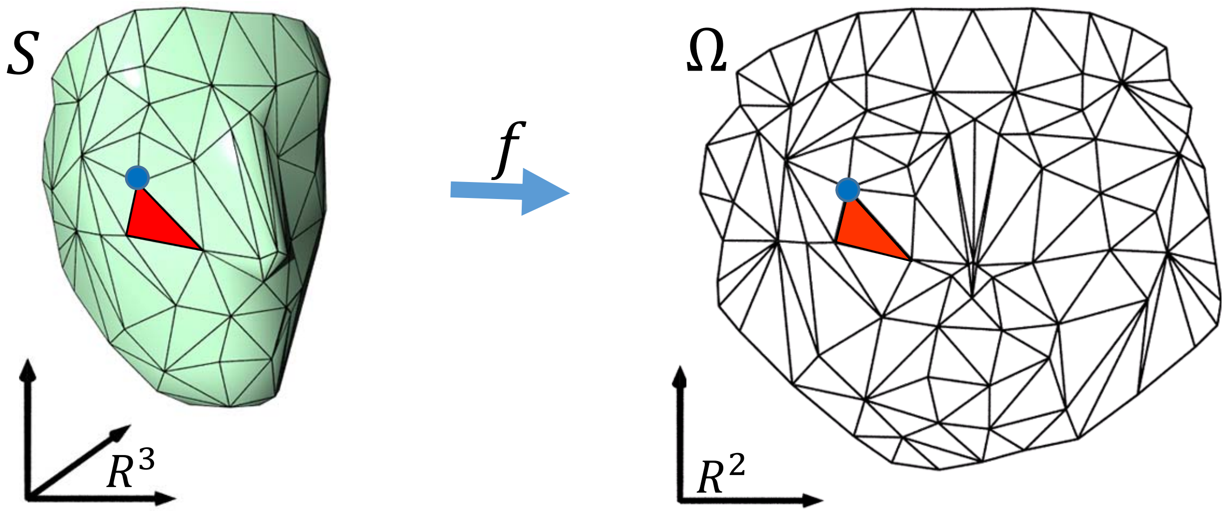
GAMES102

GAMES 102在线课程：几何建模与处理基础

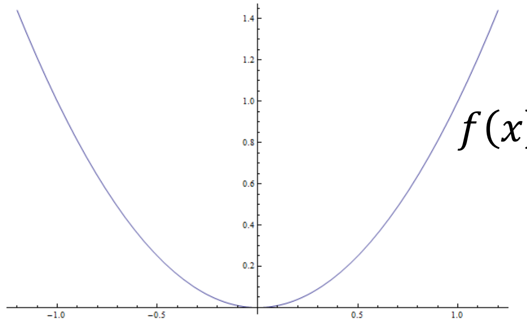
几何映射

回顾： 曲面参数化

- 问题：将空间曲面展开到二维平面
- 本质：寻求一个映射 $f: S \subset R^3 \rightarrow \Omega \subset R^2$

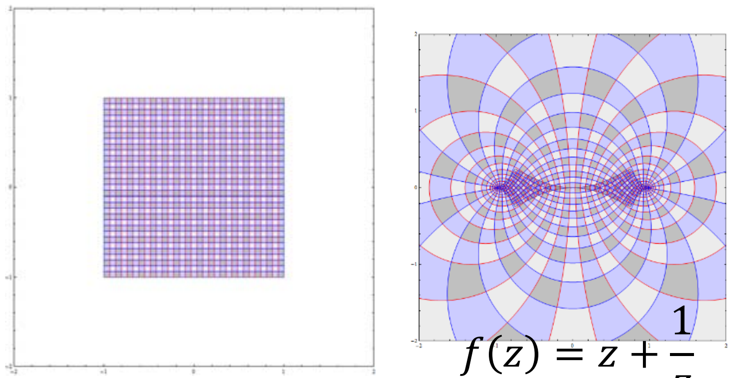


映射 （ Mapping / Map ）



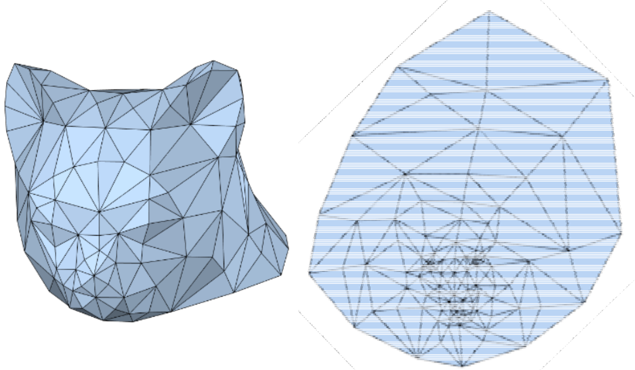
$f(x) = x^2$

$f: \mathbb{R} \to \mathbb{R}$

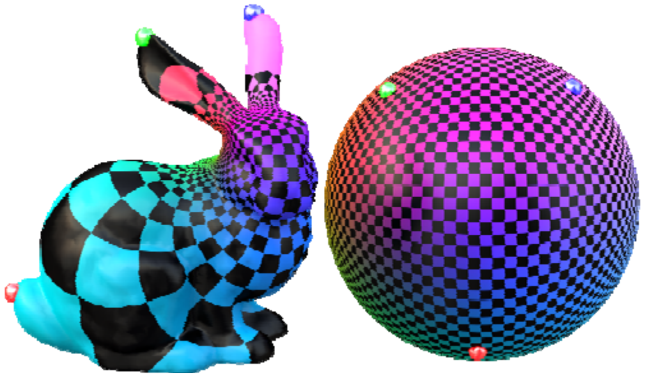


$f(z) = z + \frac{1}{z}$

$f: \mathbb{R}^2 \to \mathbb{R}^2$



$f: \mathbb{M} \to \mathbb{R}^2$

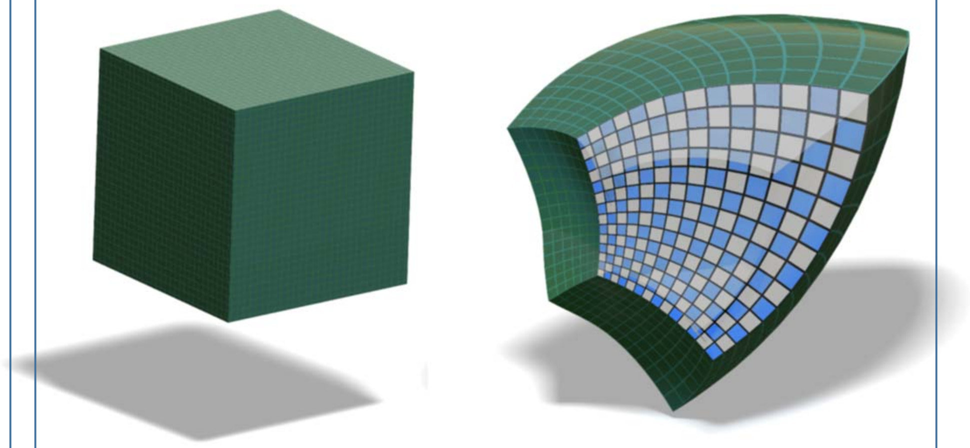


$f: \mathbb{M} \to \mathbb{S}^2$

Courtesy of Roi Poranne and Shahar Kovalsky

映射 （ Mapping / Map ）

$f: \mathbb{M} \rightarrow \mathbb{M}'$



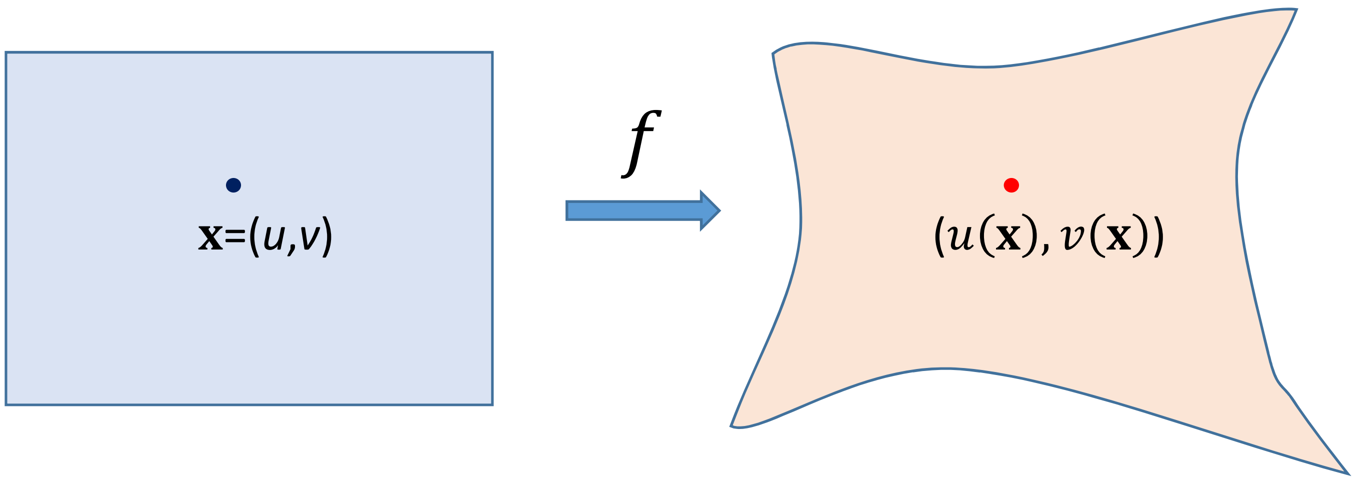
$f: \mathbb{R}^3 \rightarrow \mathbb{R}^3$

本节课：平面几何映射

$$f: \mathbb{R}^2 \to \mathbb{R}^2$$

- 映射表达：

$$\boxed{\mathbf{f}(\mathbf{x}) = \begin{pmatrix} u(\mathbf{x}) \\ v(\mathbf{x}) \end{pmatrix}}$$



映射的表达

映射的表达：化繁为简

- 映射表达为基本映射（基函数）的线性组合
  - 函数的分解

- 映射表达为小区域（三角形区域）上映射的拼接
  - 区域的分解（映射的离散）

映射的表达：化繁为简

- 映射表达为基本映射（基函数）的线性组合
  - 函数的分解

- 映射表达为小区域（三角形区域）上映射的拼接
  - 区域的分解（映射的离散）

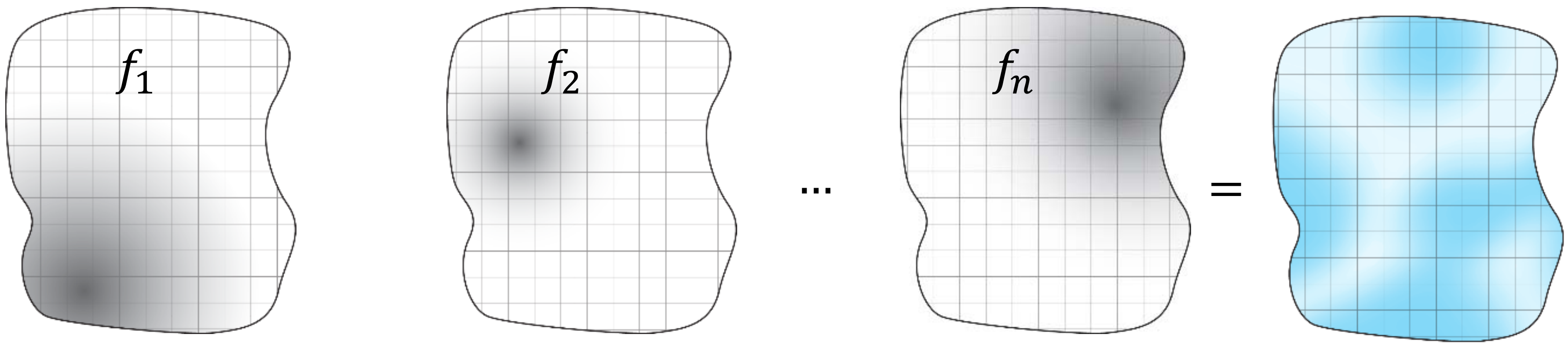
映射： 基函数的线性组合

- 基函数(basis functions):

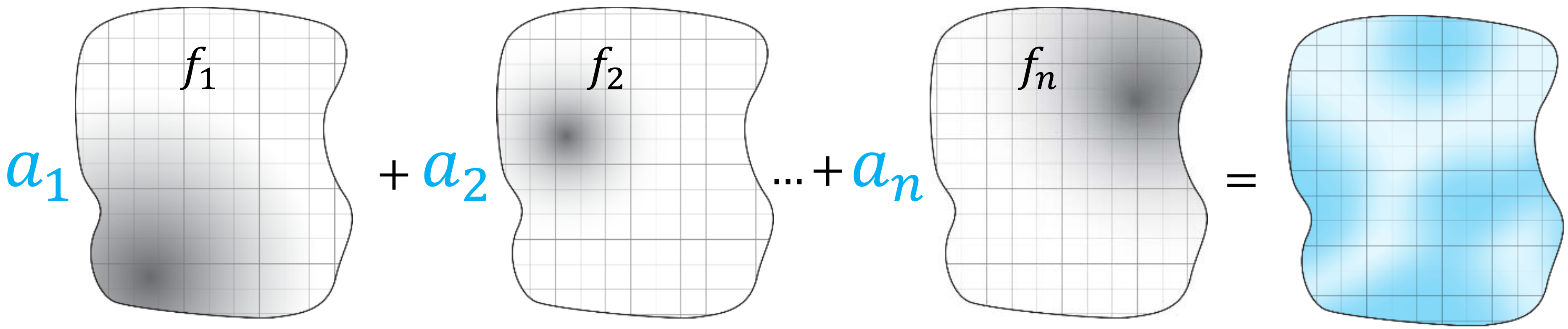
$$f_1, f_2, f_3, \cdots, f_n$$

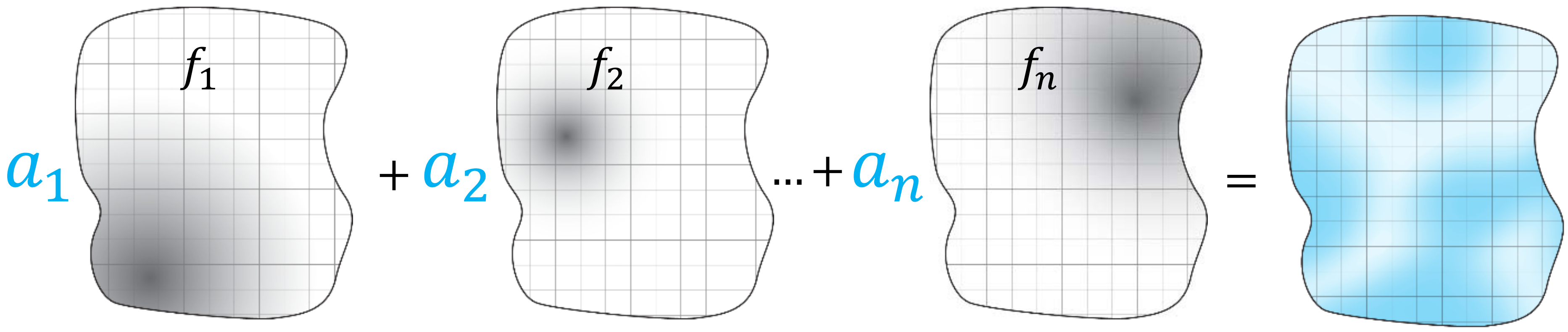
- 基函数的线性组合：

$$\mathbf{f}(\mathbf{x}) = \begin{pmatrix} u(\mathbf{x}) \\ v(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} \sum a_i f_i(\mathbf{x}) \\ \sum b_i f_i(\mathbf{x}) \end{pmatrix}$$

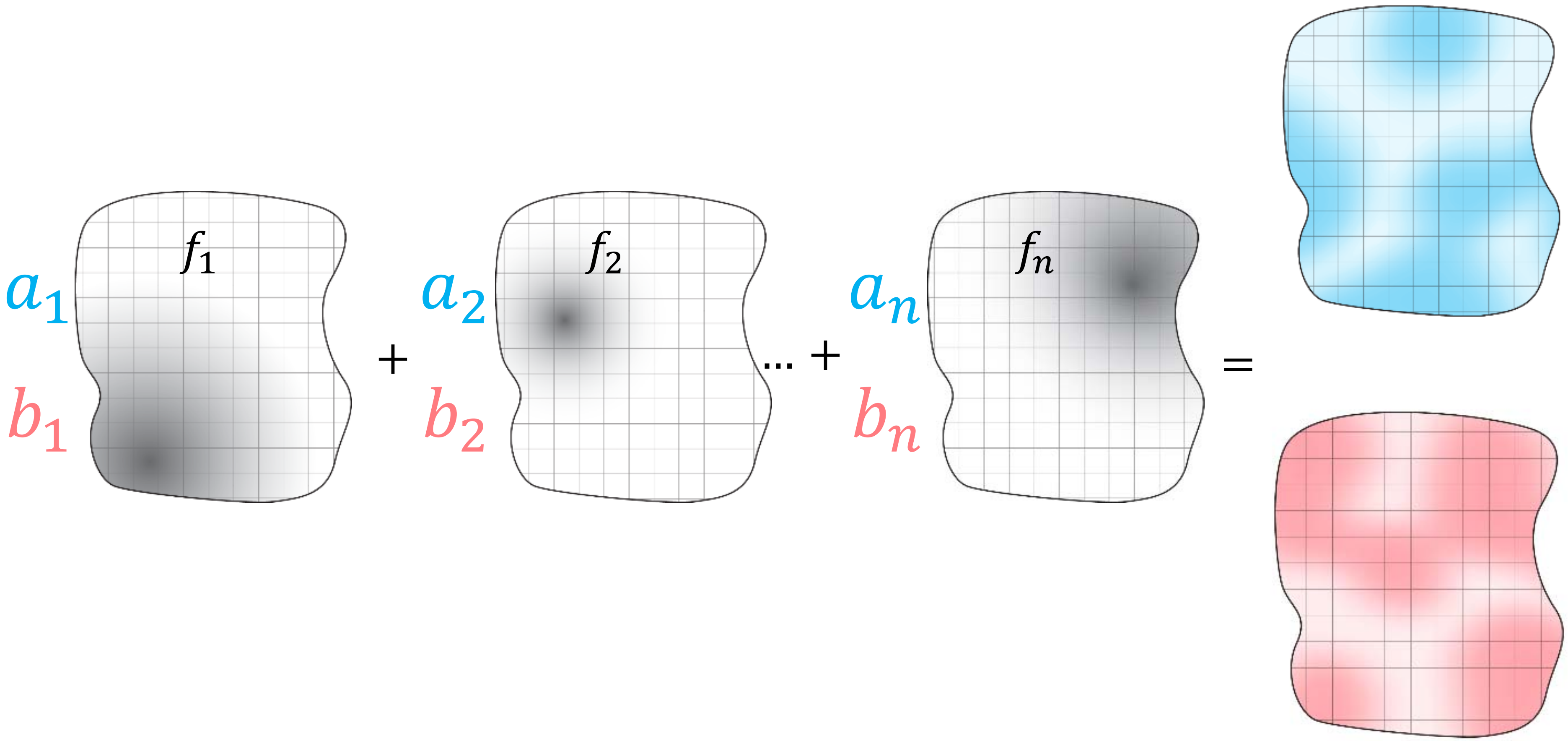


f_1
f_2
...
f_n
=

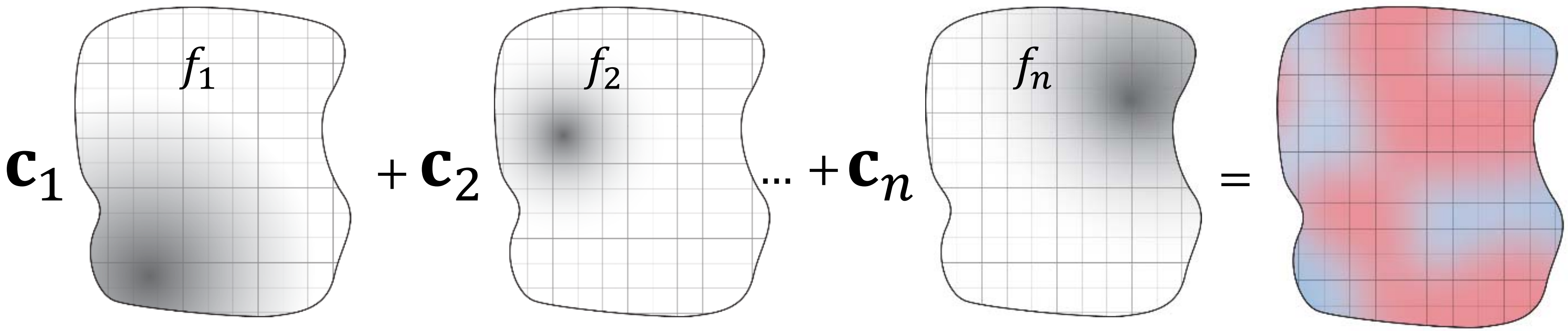

a_1
f_1
$+ a_2$
f_2
$\ldots + a_n$
f_n
$=$



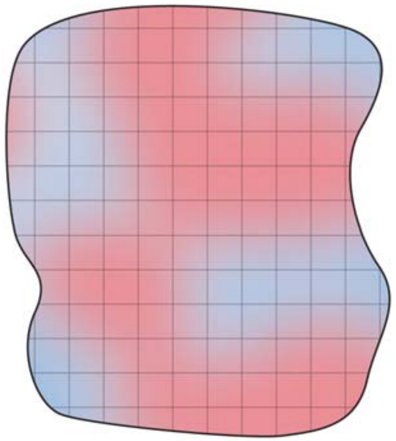
a_1
f_1
$+ a_2$
f_2
$\dots + a_n$
f_n
$=$



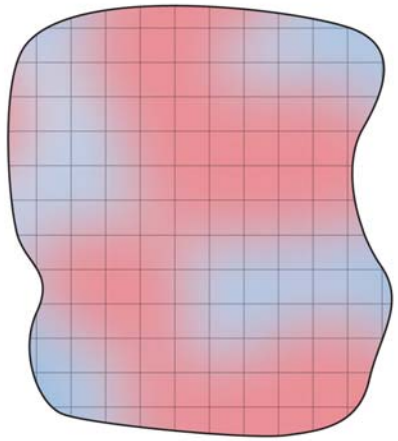
a_1
b_1
f_1
$+$
a_2
b_2
f_2
$\ldots +$
a_n
b_n
f_n
$=$


$\mathbf{c}_1$
f_1
$+\mathbf{c}_2$
f_2
$\cdots+\mathbf{c}_n$
f_n
$=$

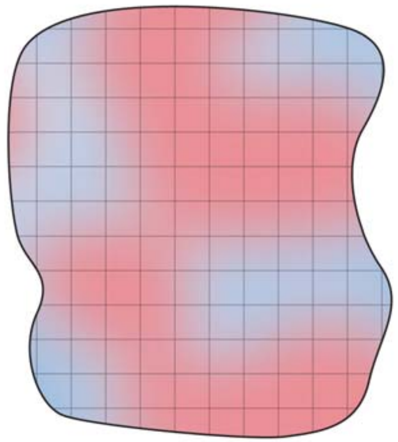
$$\mathbf{f}(\mathbf{x}) = \begin{pmatrix} \sum a_i f_i(\mathbf{x}) \\ = \sum \mathbf{c}_i f_i(\mathbf{x}) \\ \sum b_i f_i(\mathbf{x}) \end{pmatrix} =$$

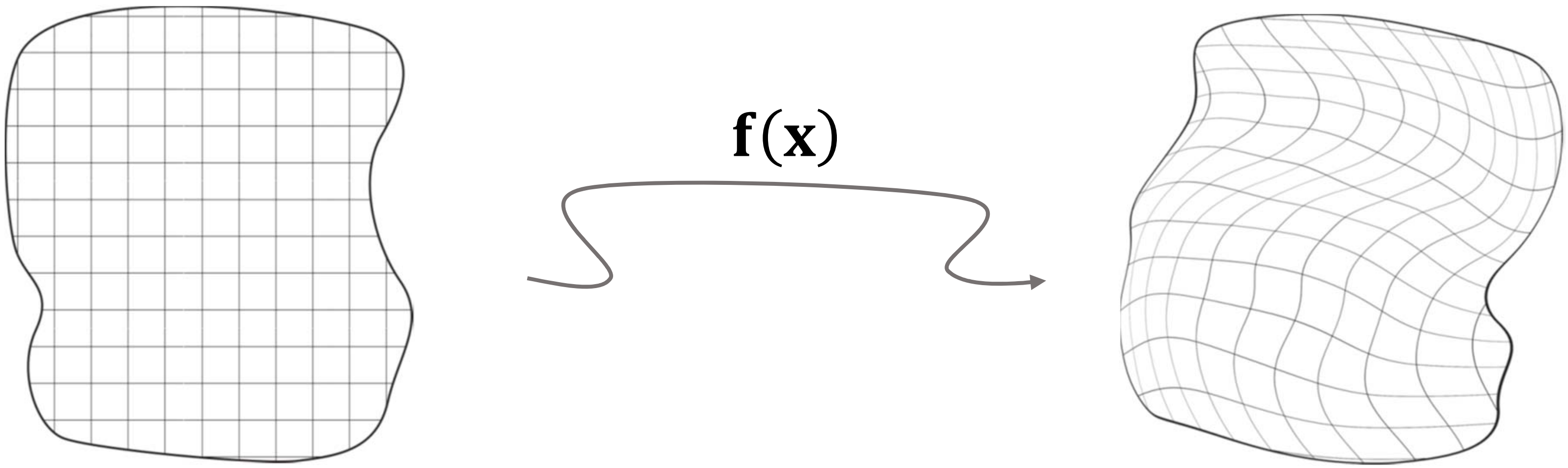


$$\mathbf{f}(\mathbf{x}) = \begin{pmatrix} \sum a_i f_i(\mathbf{x}) \\ \sum b_i f_i(\mathbf{x}) \end{pmatrix} = \sum \mathbf{c}_i f_i(\mathbf{x}) =$$



$$\mathbf{f}(\mathbf{x}) = \begin{pmatrix} \sum a_i f_i(\mathbf{x}) \\ \sum b_i f_i(\mathbf{x}) \end{pmatrix} = \sum \mathbf{c}_i f_i(\mathbf{x}) =$$



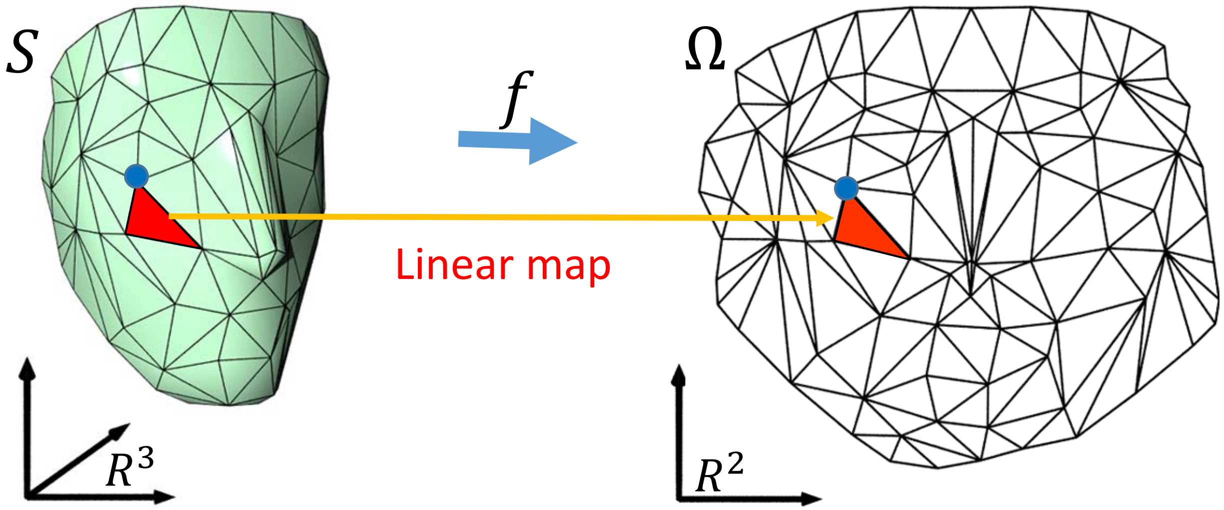


映射的表达：化繁为简

- 映射表达为基本映射（基函数）的线性组合
  - 函数的分解
- 映射表达为小区域（三角形区域）上映射的拼接
  - 区域的分解（映射的离散）

映射：简单区域上映射的连续组合

- f is approximated by piecewise linear maps between pairs of triangles



几何映射的例子

例1： 2D变形

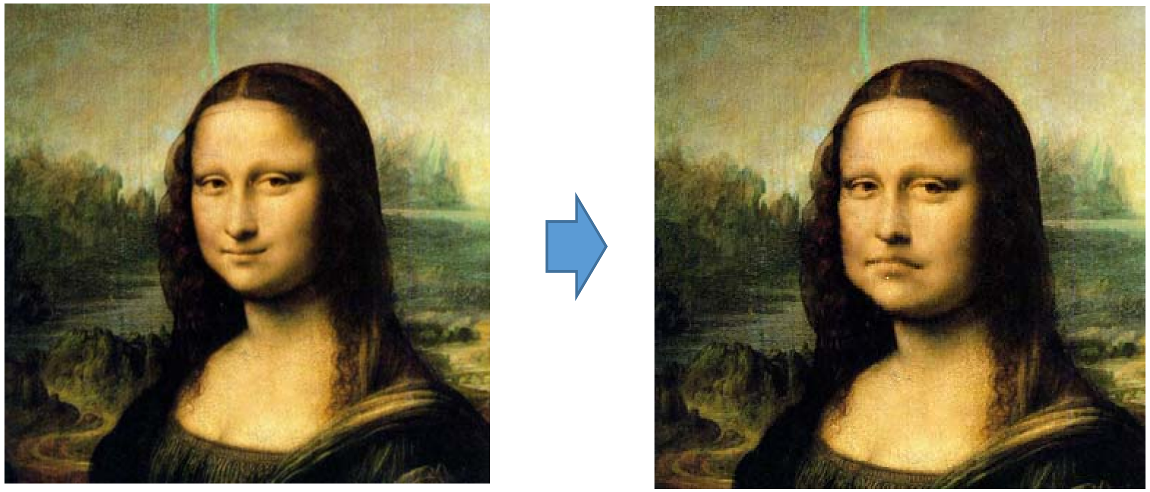
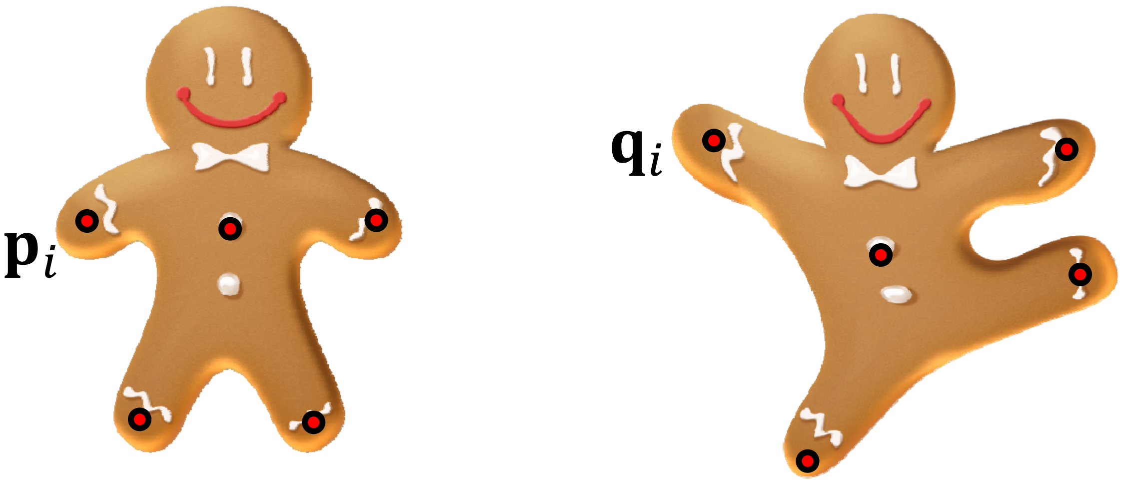
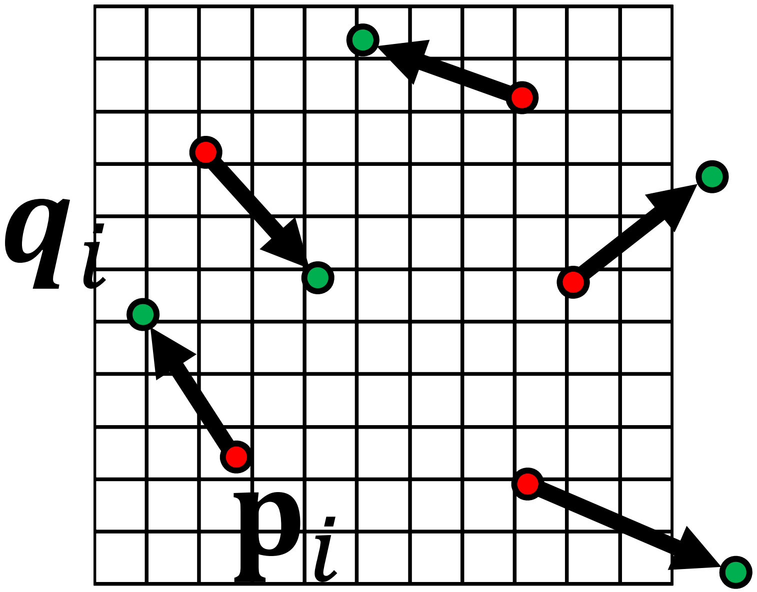


Image warping



Shape deformation

本质：插值问题



$$\mathbf{f}(\mathbf{p}_i) = \sum \mathbf{c}_i f_i(\mathbf{x})$$

$$\mathbf{c}_i = ?$$

$$\mathbf{f}(\mathbf{p}_i) = \mathbf{q}_i, \forall i$$

$$\sum \mathbf{c}_i f_i(\mathbf{p}_i) = \mathbf{q}_i, \forall i$$

求解

- 插值法（比如，RBF插值）

$$\mathbf{f}(\mathbf{p}_i) = \mathbf{c}_0 + \mathbf{c}_x \boldsymbol{x} + \mathbf{c}_y \boldsymbol{y} + \sum \mathbf{c}_i \phi(\|\mathbf{x} - \mathbf{p}_i\|)$$

$$\phi(r) = r^2 \log r$$

$$\mathbf{f}(\mathbf{p}_i) = \mathbf{q}_i, \forall i$$

- 逼近法（能量极小法）

$$\min E_{\mathrm{TPS}}(\mathbf{f}) = \iint \left[\left(\frac{\partial^2 \mathbf{f}}{\partial x^2} \right)^2 + 2 \left(\frac{\partial^2 \mathbf{f}}{\partial x \partial y} \right)^2 + \left(\frac{\partial^2 \mathbf{f}}{\partial y^2} \right)^2 \right]$$

Bending energy

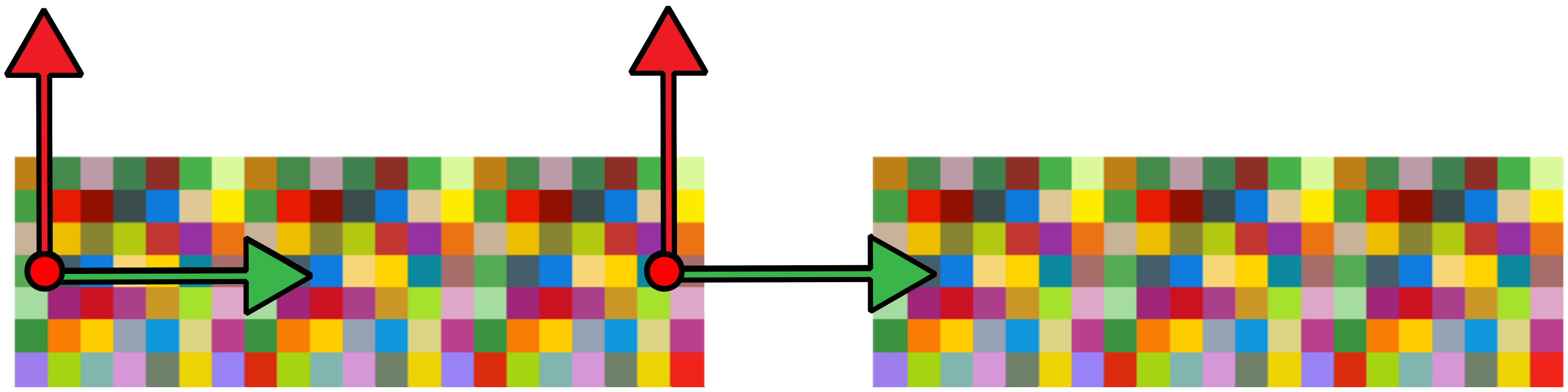
$$\text{s.t. } \mathbf{f}(\mathbf{p}_i) = \mathbf{q}_i, \forall i$$

更多约束

- Hermite插值：插值梯度

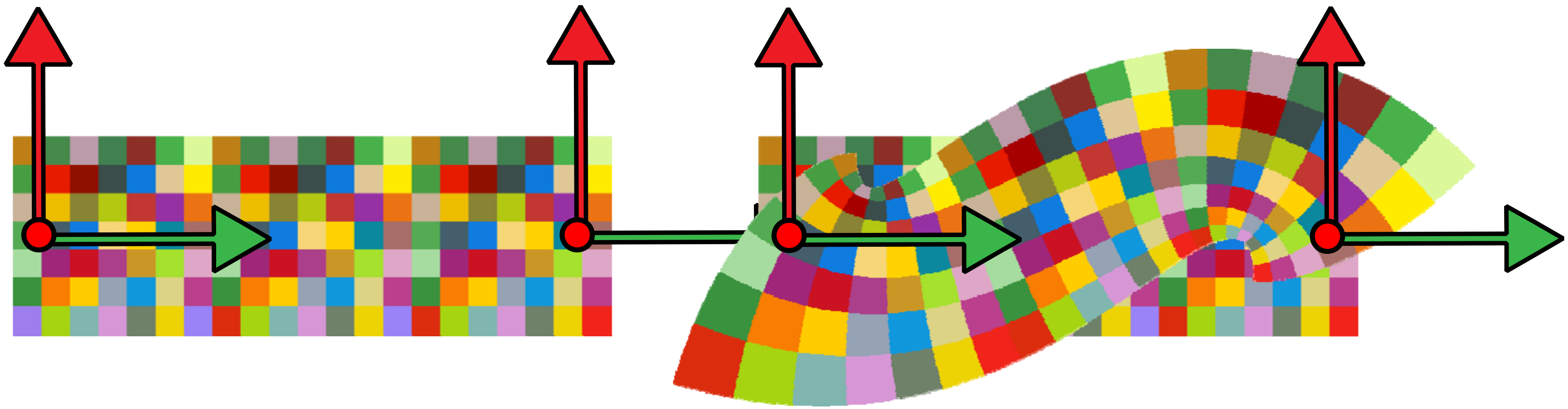
更多约束

- Hermite插值：插值梯度



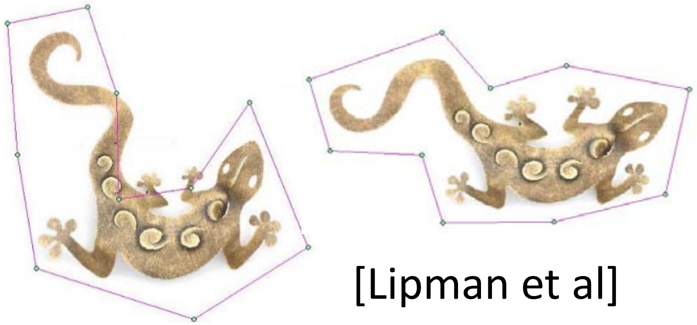
更多约束

- Hermite插值：插值梯度

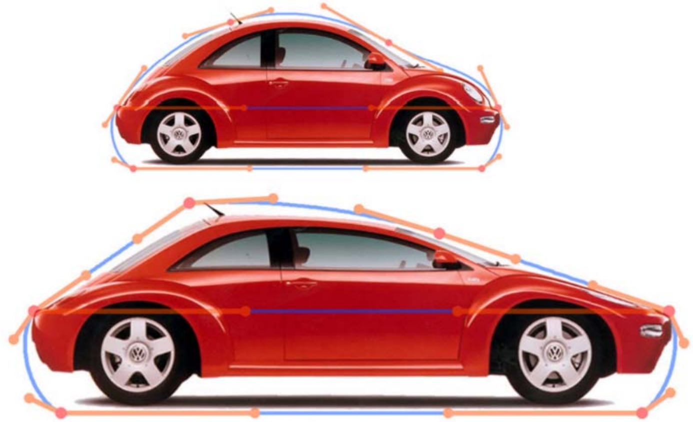


$$\mathbf{f}(\mathbf{p}_i) = \mathbf{q}_i \qquad \mathbf{Df}(\mathbf{p}_i) = \mathbf{D}_i$$

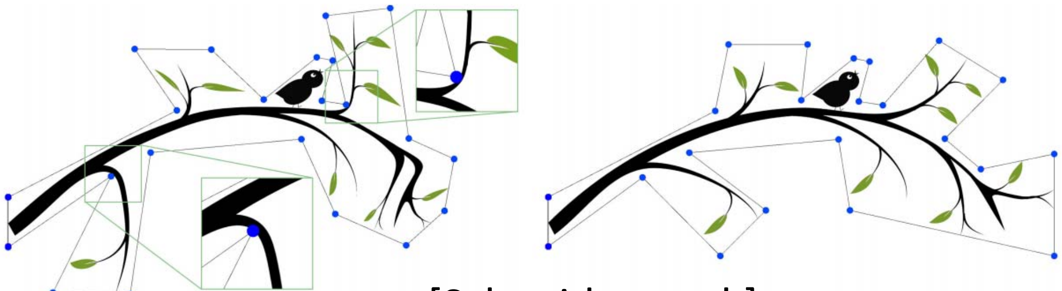
例2: Barycentric Coordinates



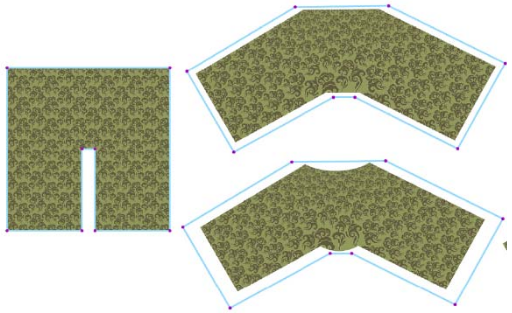
[Lipman et al]



[Li et al.]

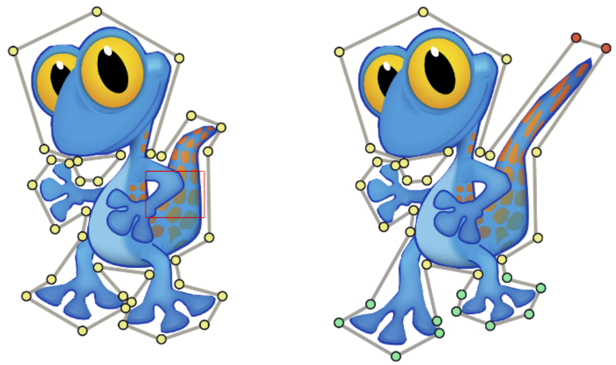


[Schneider et al.]



[Weber et al.]

[Weber et. al]

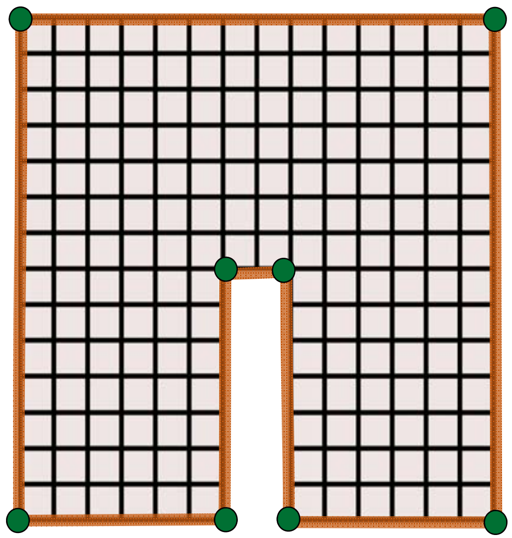


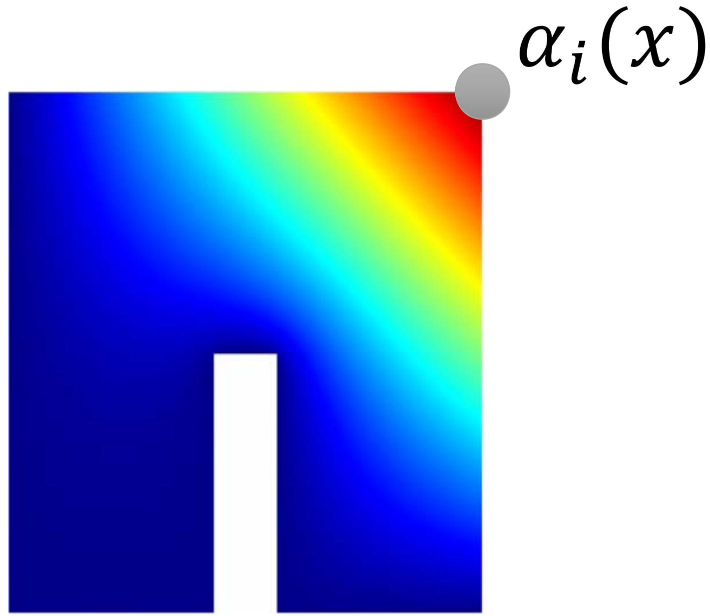
[Li et al.]

例2: Barycentric Coordinates

Stages:

- Source shape
- Polygonal cage
- Coordinates



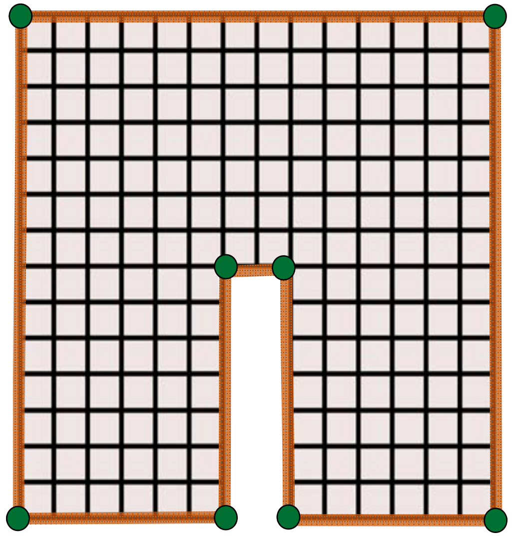


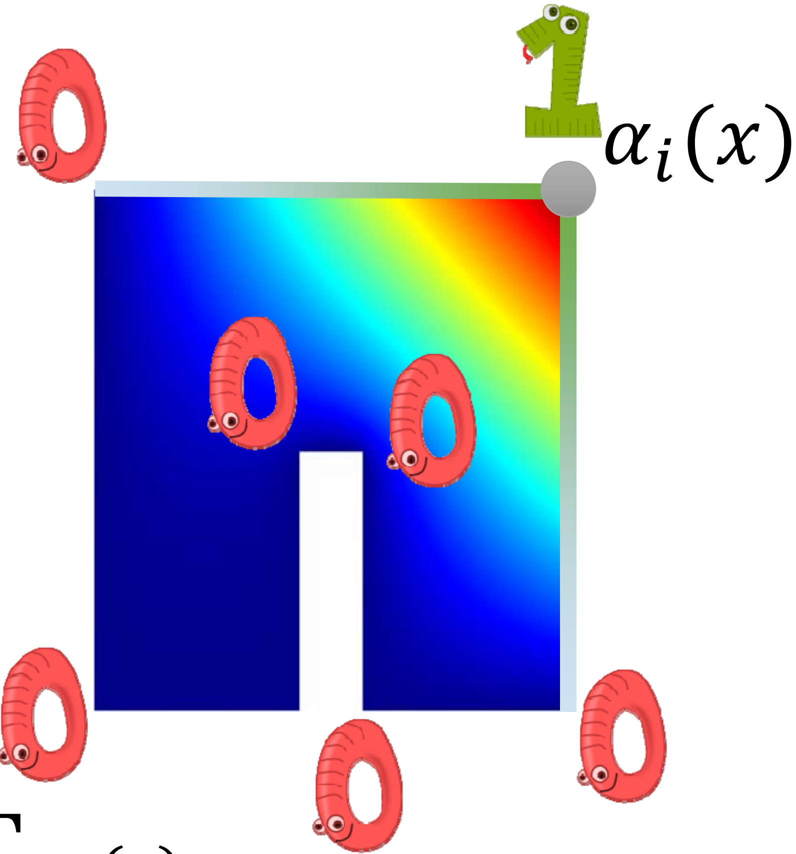
$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i(\mathbf{x})\mathbf{q}_i$$

例2: Barycentric Coordinates

Stages:

- Source shape
- Polygonal cage
- Coordinates



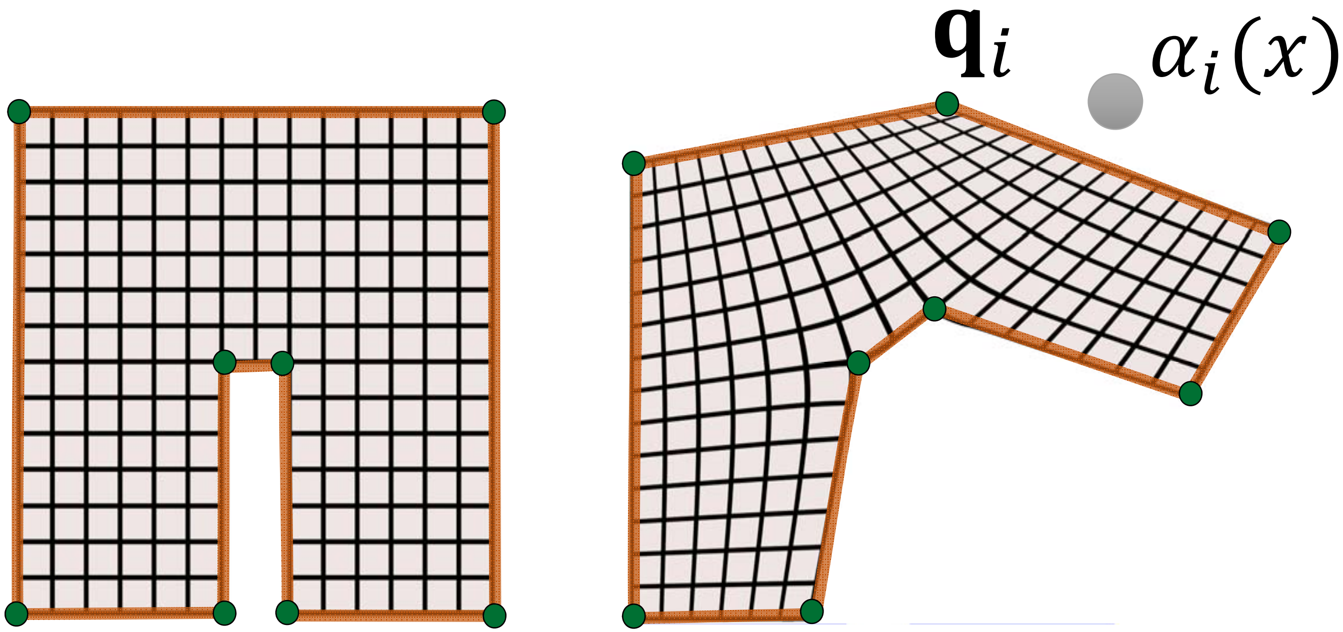


$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i(\mathbf{x})\mathbf{q}_i$$

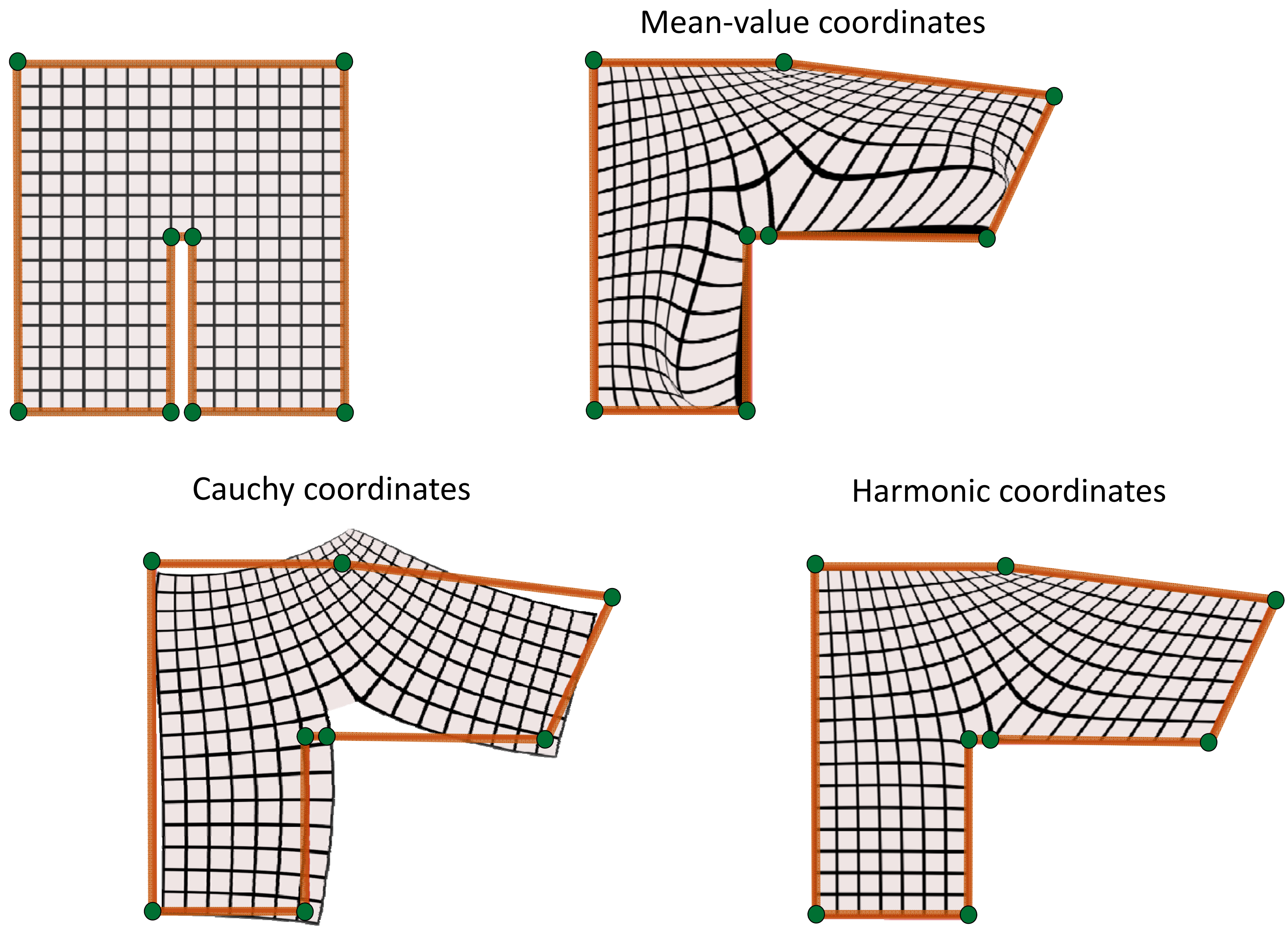
例2: Barycentric Coordinates

Stages:

- Source shape
- Polygonal cage
- Coordinates
- Manipulate cage
- Apply deformation



$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i(\mathbf{x})\mathbf{q}_i$$


Mean-value coordinates
Cauchy coordinates
Harmonic coordinates

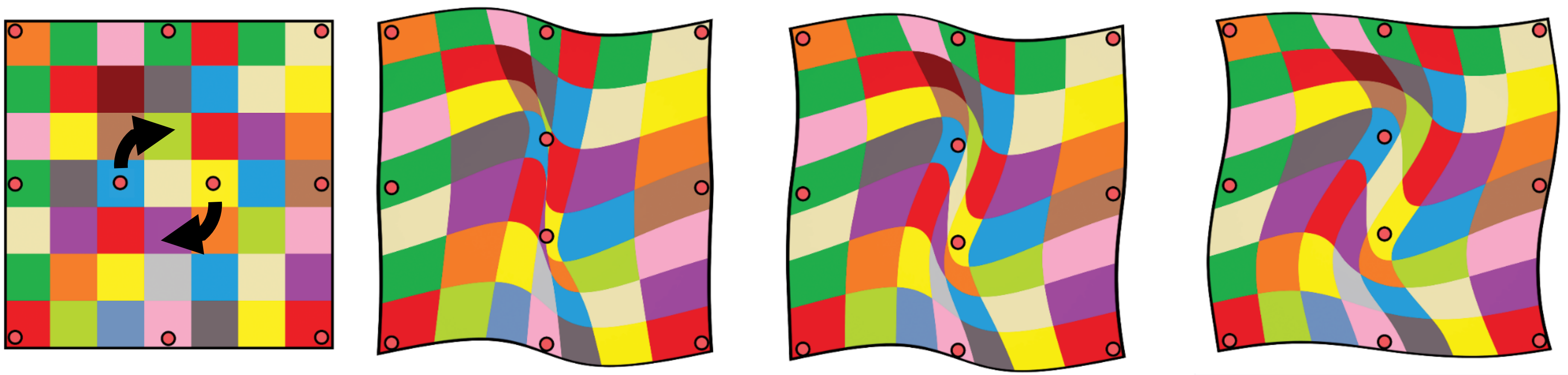
映射的性质

What are good maps?

Local

Bijectivity

Low distortion

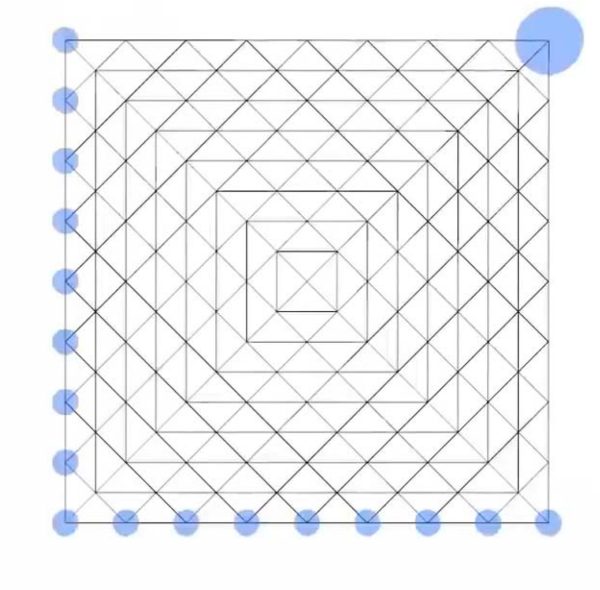


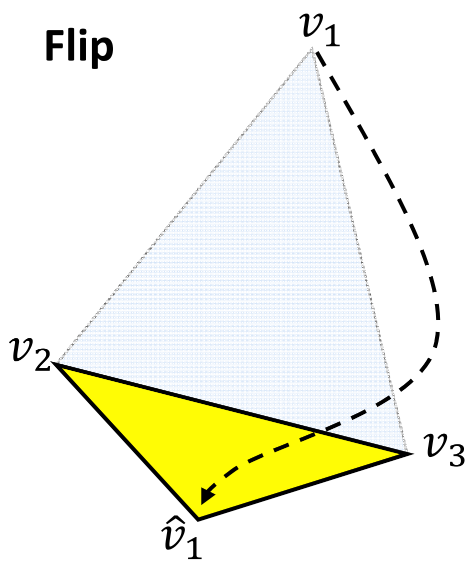
Not Bijective

Bijective

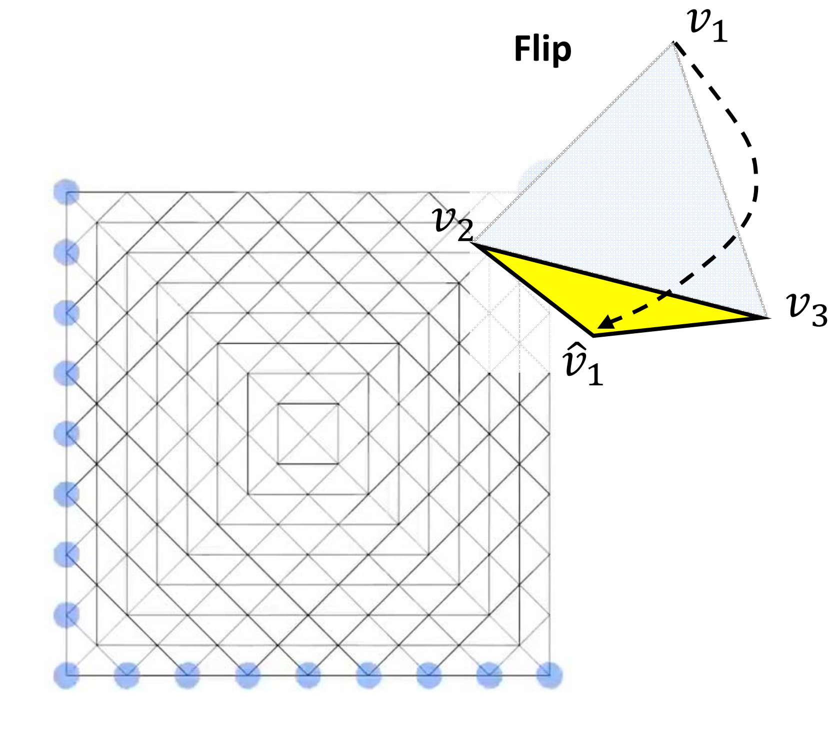
Lower distortion

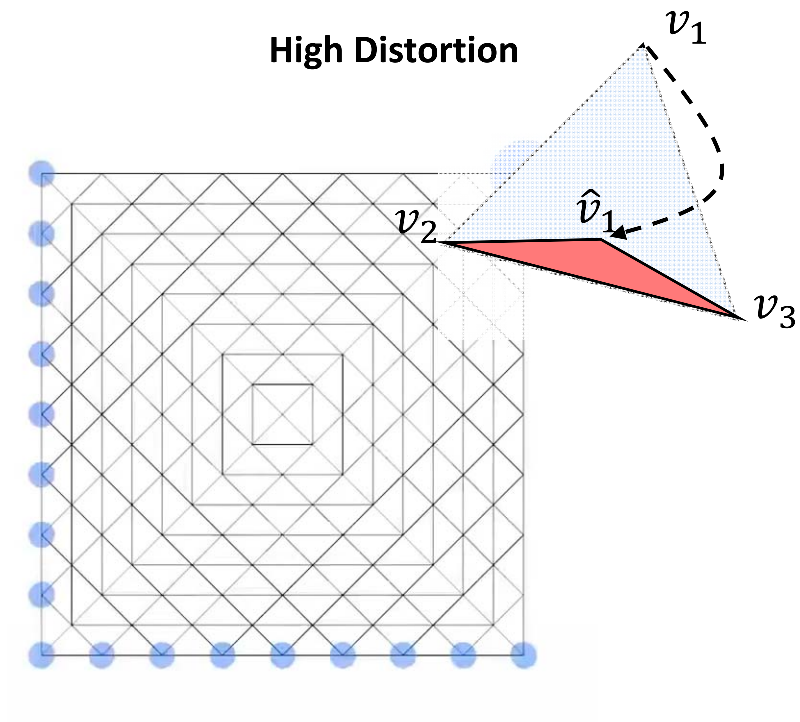
Flip (foldover) triangles in mapping





Locally injective mappings





Flip-free (foldover-free) mapping

Globally Bijective VS. Locally Bijective

Globally Bijective $\Rightarrow$ Locally Bijective

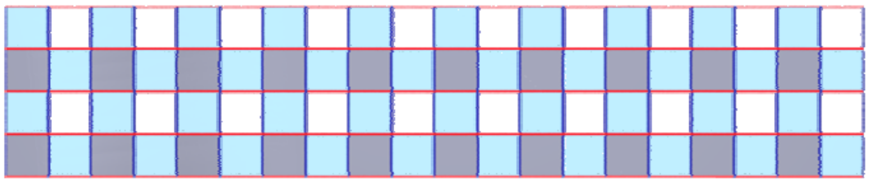
=Injective

$\mathbf{f}$ is bijective

$\mathbf{f}: \mathbf{U} \rightarrow \mathbf{f}(\mathbf{U})$ is bijective

Globally Bijective VS. Locally Bijective

Globally Bijective	Locally Bijective =Injective
$\mathbf{f}$ is bijective	$\mathbf{f}: \mathbf{U} \rightarrow \mathbf{f}(\mathbf{U})$ is bijective



Globally Bijective VS. Locally Bijective

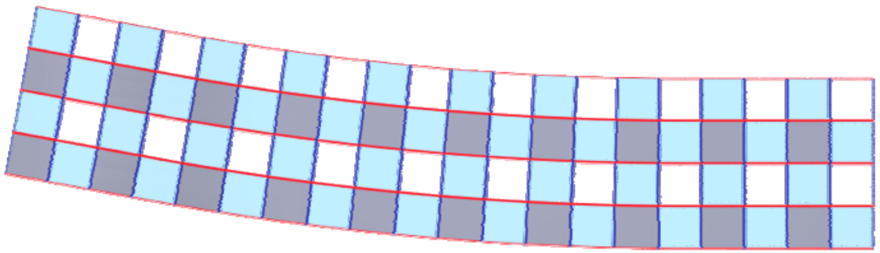
Globally Bijective

$\mathbf{f}$ is bijective

Locally Bijective

=Injective

$\mathbf{f}: \mathbf{U} \to \mathbf{f}(\mathbf{U})$ is bijective



Globally Bijective VS. Locally Bijective

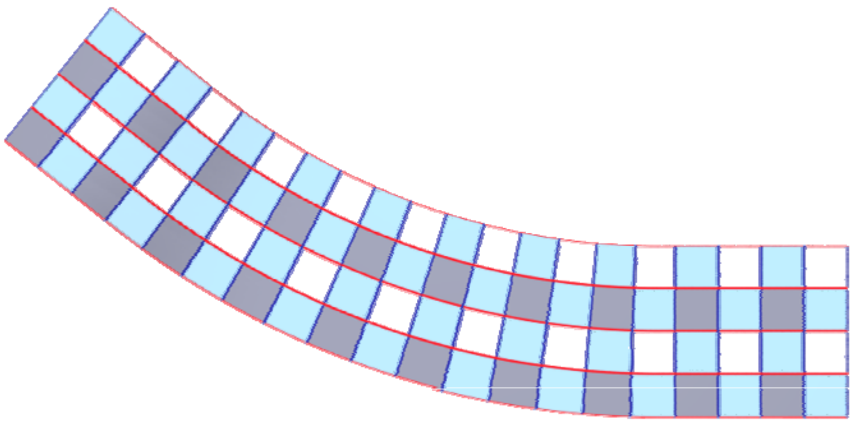
Globally Bijective

$\mathbf{f}$ is bijective

Locally Bijective

=Injective

$\mathbf{f}: \mathbf{U} \to \mathbf{f}(\mathbf{U})$ is bijective



Globally Bijective VS. Locally Bijective

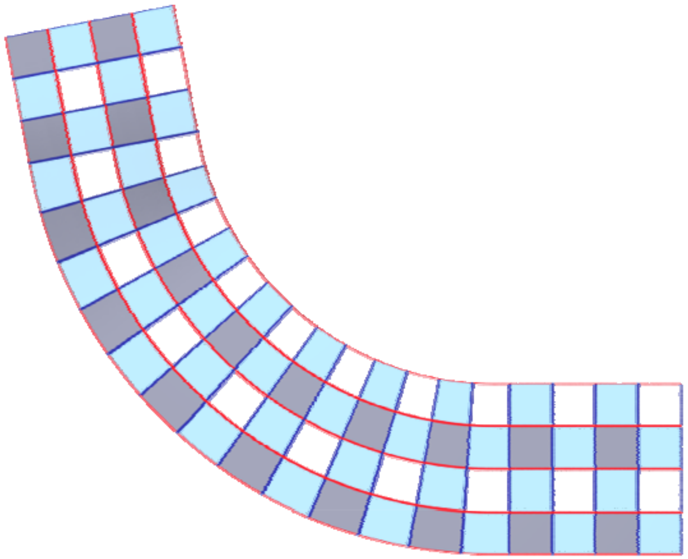
Globally Bijective

Locally Bijective

=Injective

$\mathbf{f}$ is bijective

$\mathbf{f}: \mathbf{U} \to \mathbf{f}(\mathbf{U})$ is bijective



Globally Bijective VS. Locally Bijective

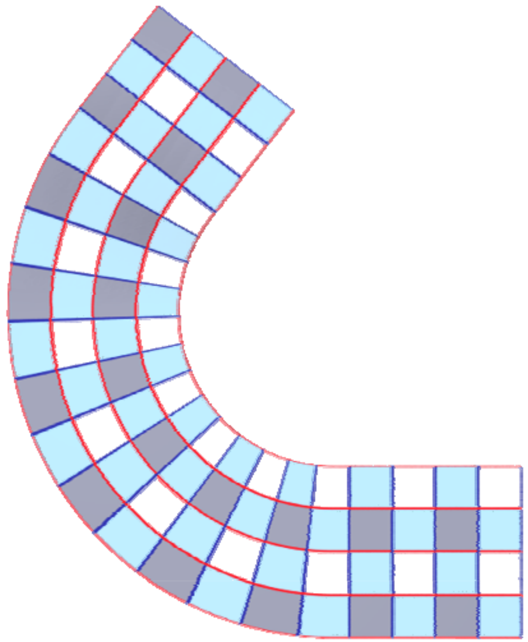
Globally Bijective

Locally Bijective

=Injective

$\mathbf{f}$ is bijective

$\mathbf{f}\colon \mathbf{U} \to \mathbf{f}(\mathbf{U})$ is bijective



Globally Bijective VS. Locally Bijective

Globally Bijective

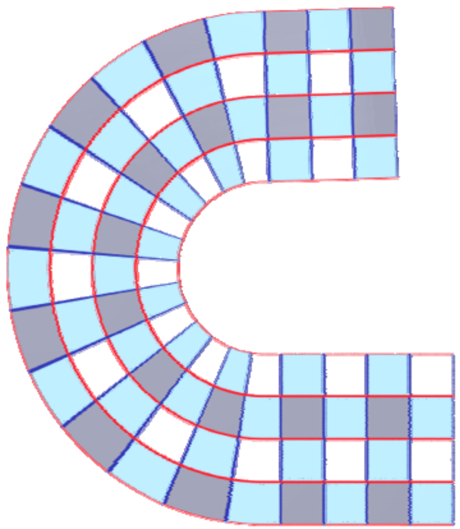
Locally Bijective

=Injective

$\mathbf{f}$ is bijective

$\mathbf{f}: \mathbf{U} \rightarrow \mathbf{f}(\mathbf{U})$ is bijective

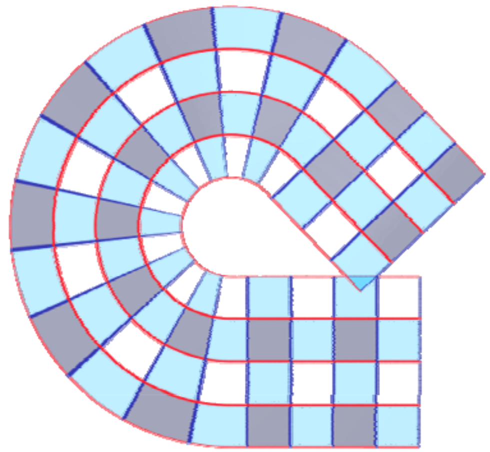
Still Bijective!



Globally Bijective VS. Locally Bijective

Globally Bijective	Locally Bijective
$\mathbf{f}$ is bijective	$\mathbf{f}: \mathbf{U} \rightarrow \mathbf{f}(\mathbf{U})$ is bijective

=Injective



Globally Bijective VS. Locally Bijective

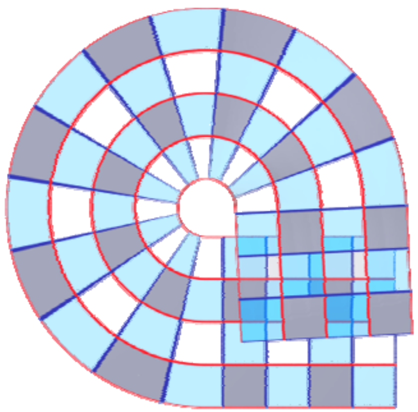
Globally Bijective

$\mathbf{f}$ is bijective

Locally Bijective

=Injective

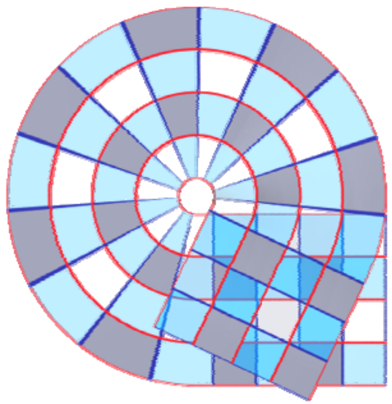
$\mathbf{f}: \mathbf{U} \rightarrow \mathbf{f}(\mathbf{U})$ is bijective



Globally Bijective VS. Locally Bijective

Globally Bijective	Locally Bijective
$\mathbf{f}$ is bijective	$\mathbf{f}: \mathbf{U} \rightarrow \mathbf{f}(\mathbf{U})$ is bijective

=Injective



Globally Bijective VS. Locally Bijective

Globally Bijective

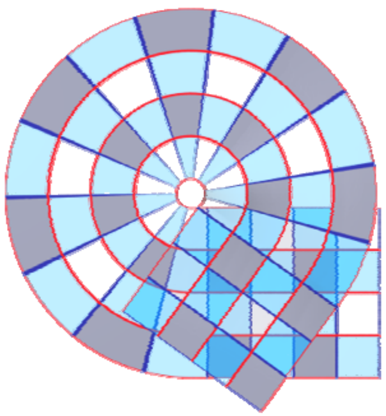
$\mathbf{f}$ is bijective

Not Bijective!

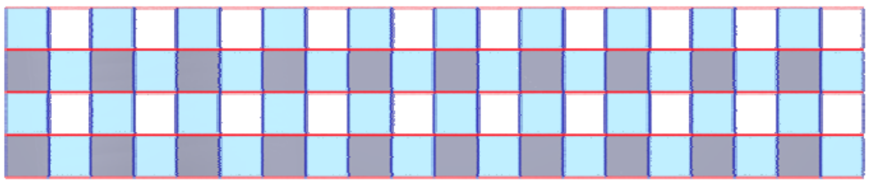
Locally Bijective

=Injective

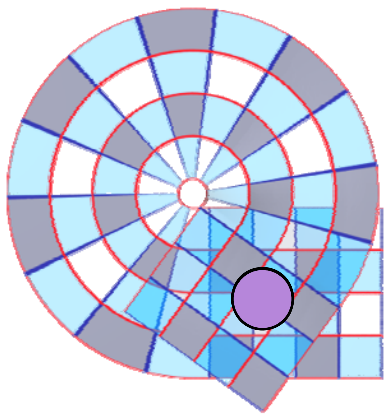
$\mathbf{f}: \mathbf{U} \rightarrow \mathbf{f}(\mathbf{U})$ is bijective



Globally Bijective VS. Locally Bijective

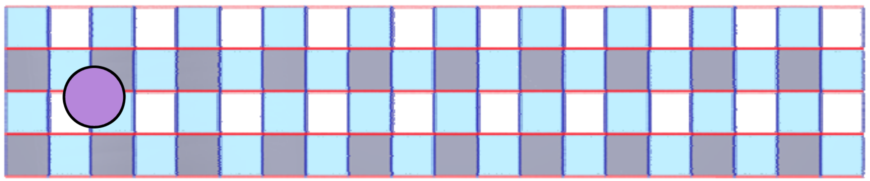


Not Bijective!

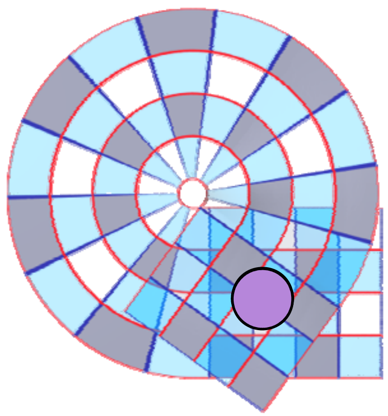


Two Pre-images

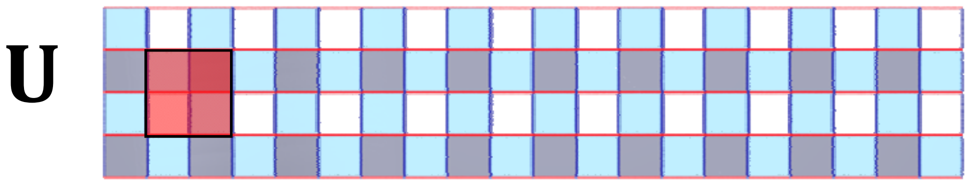
Globally Bijective VS. Locally Bijective



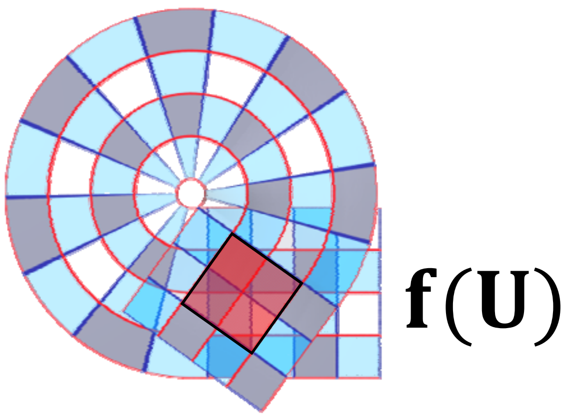
Not Bijective!



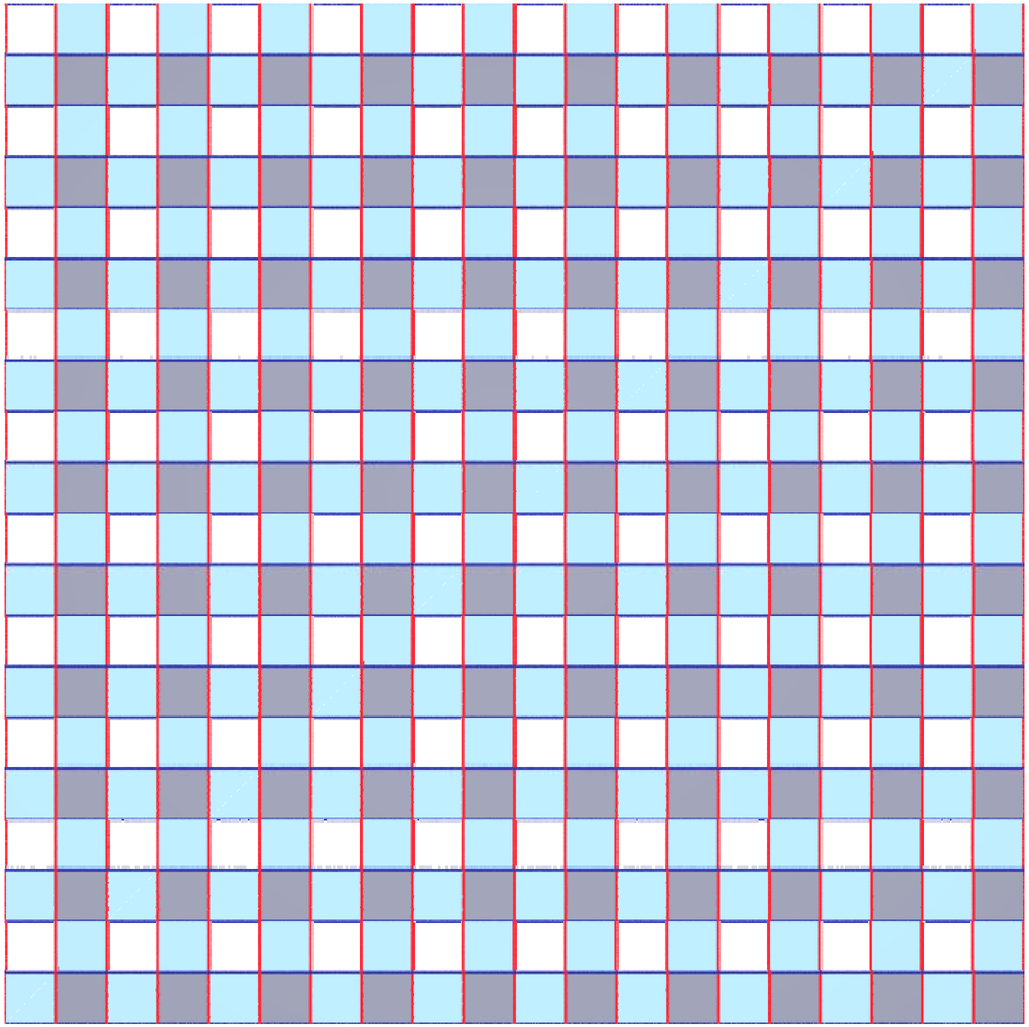
Globally Bijective VS. Locally Bijective



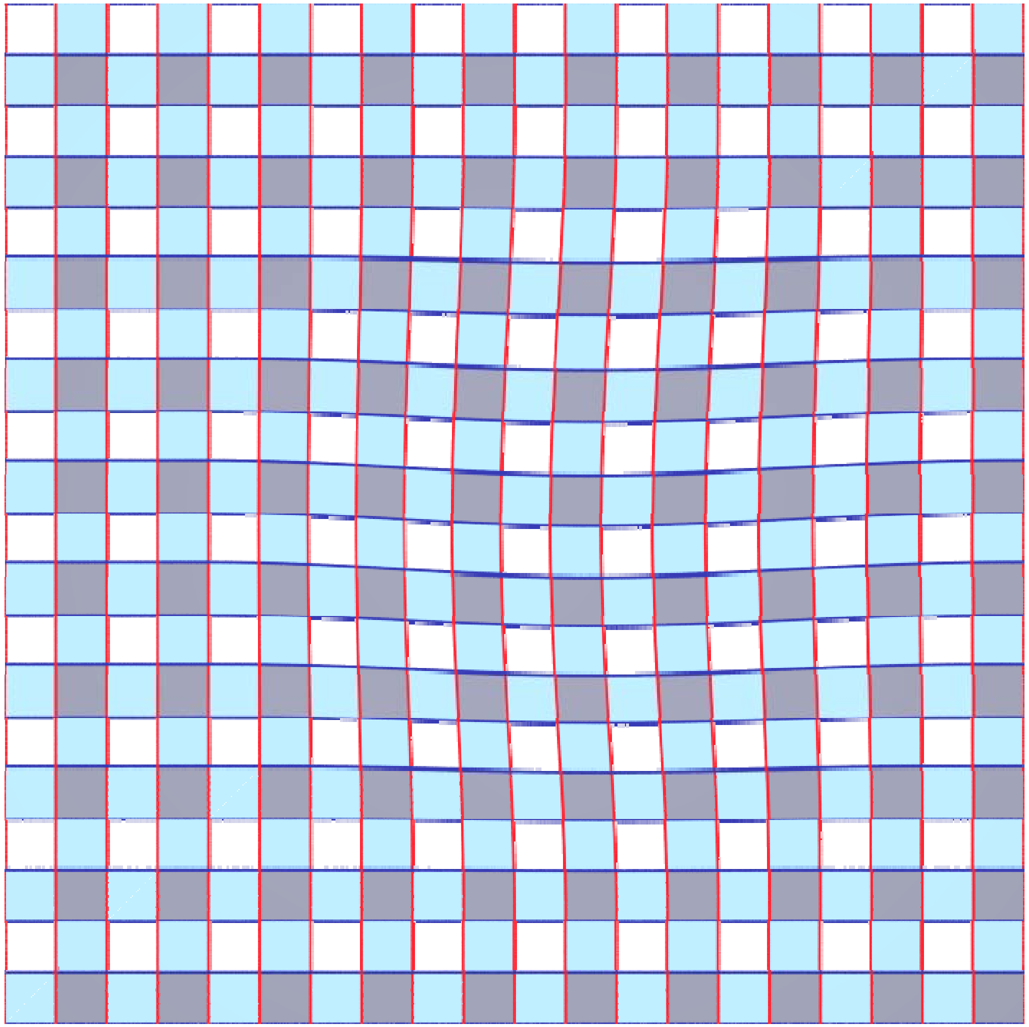
Only Locally
Bijective!



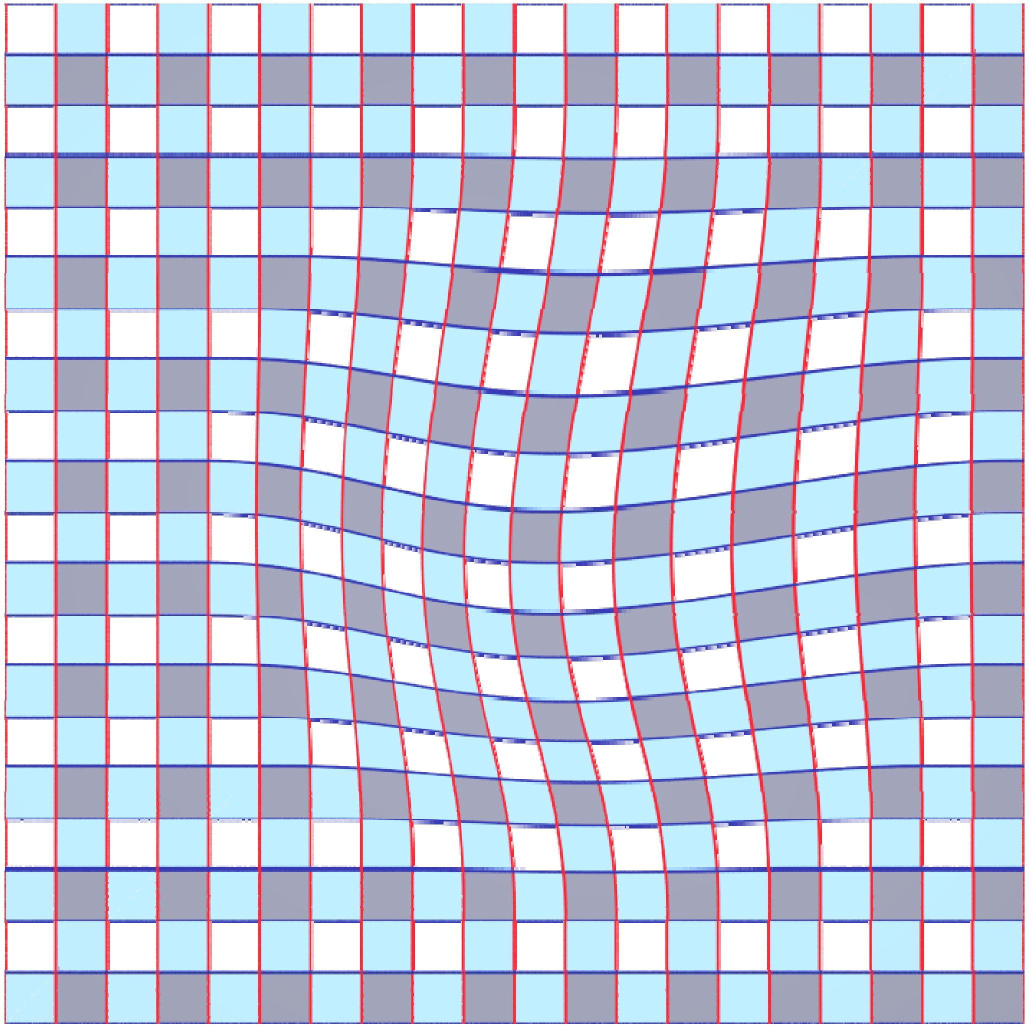
Locally Bijection – Non-example



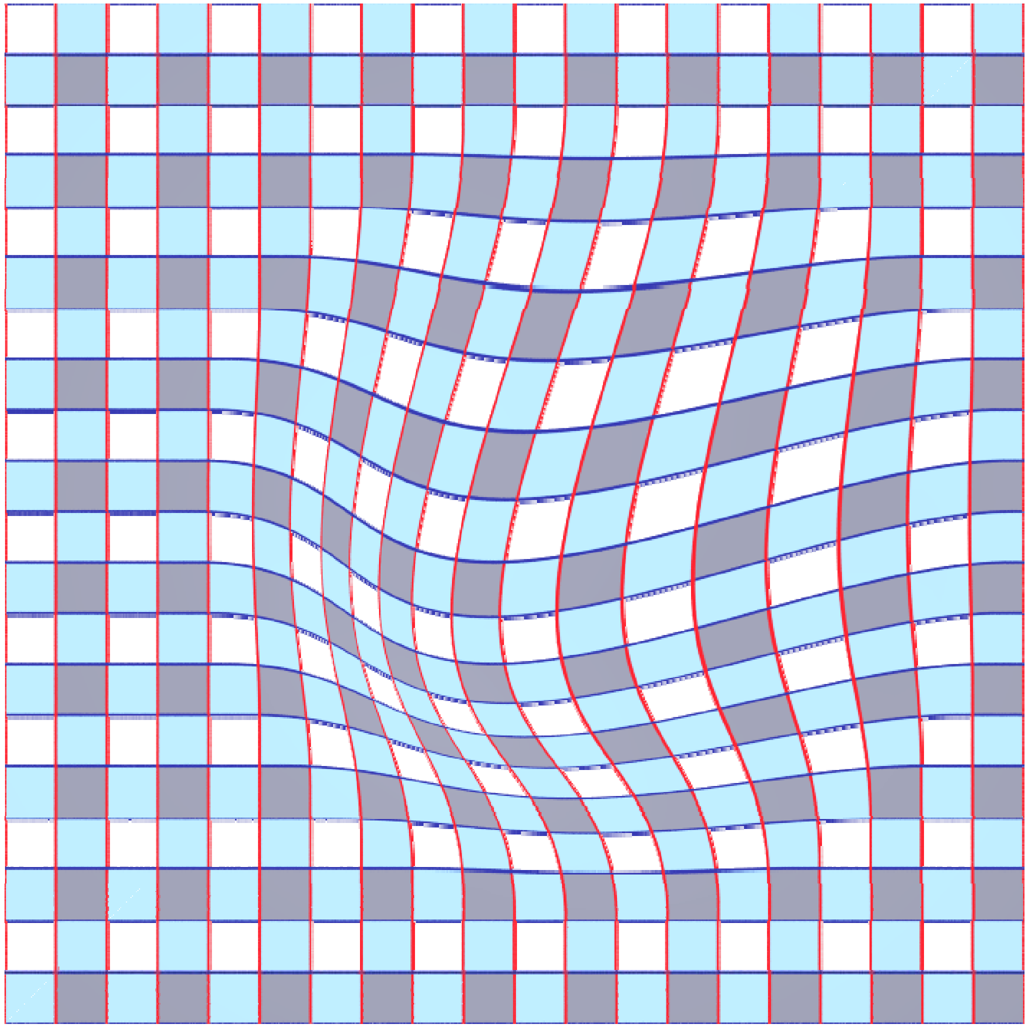
Locally Bijection – Non-example



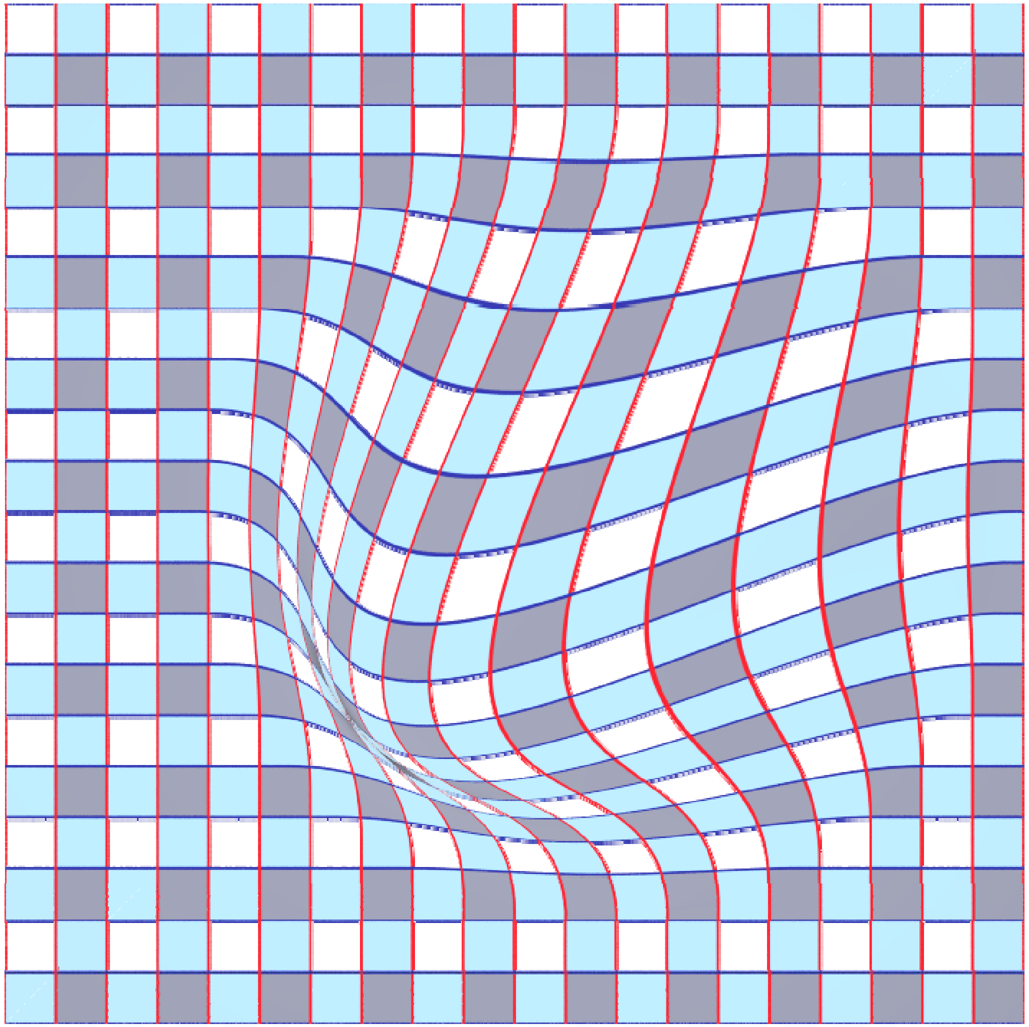
Locally Bijection – Non-example



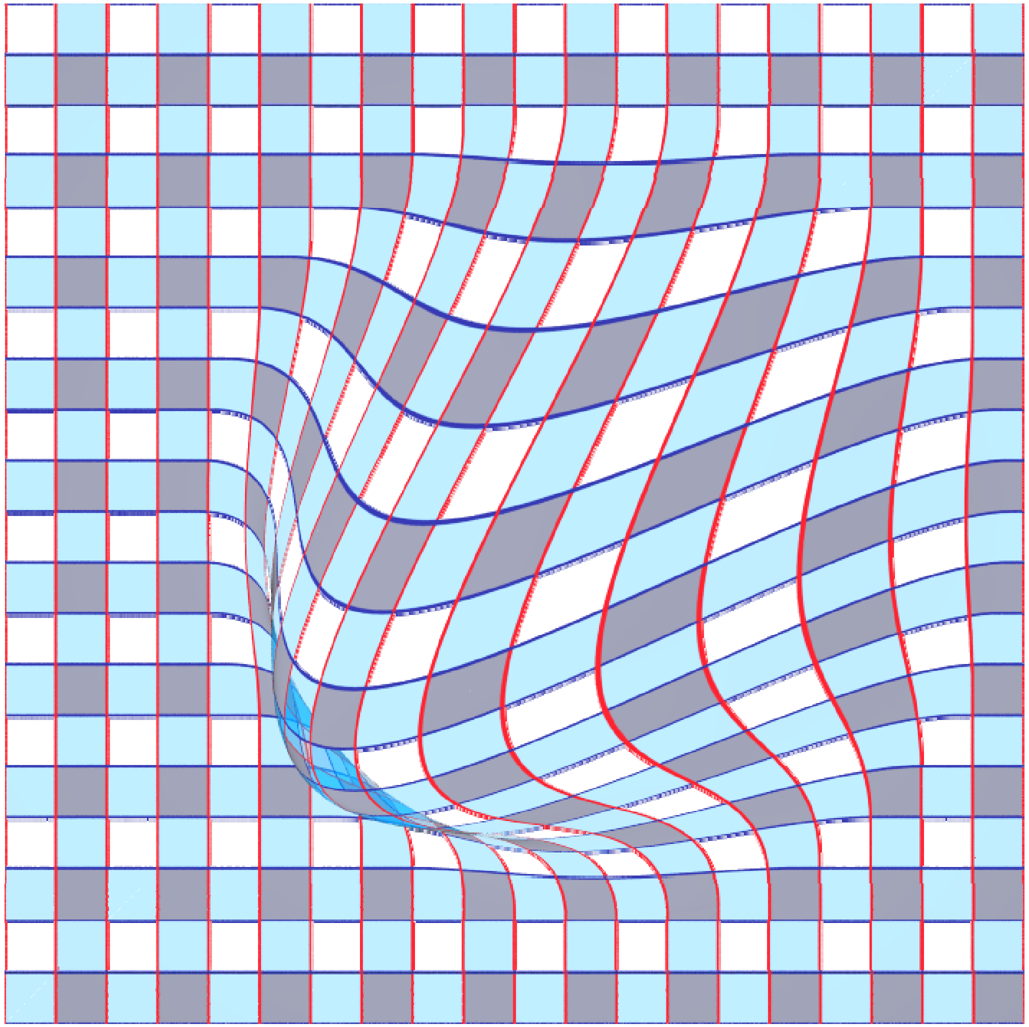
Locally Bijection – Non-example



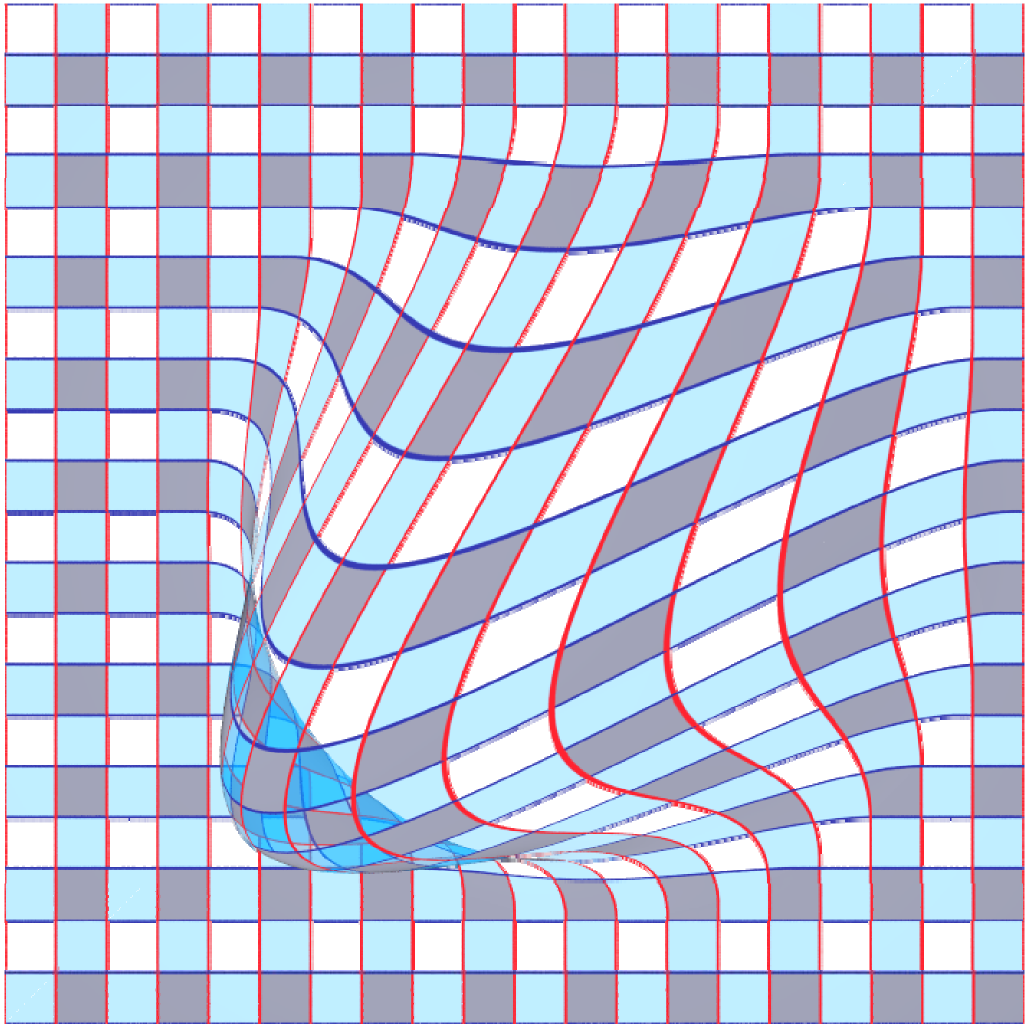
Locally Bijection – Non-example



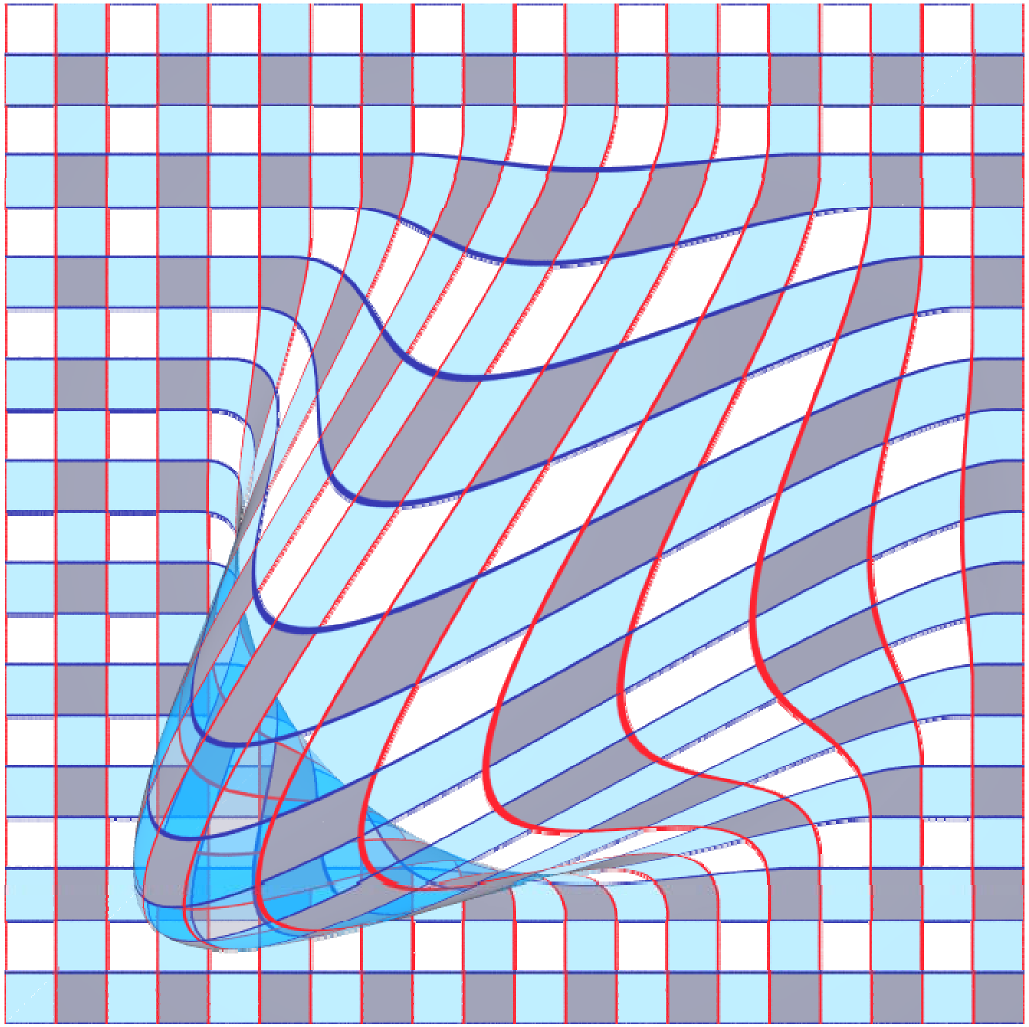
Locally Bijection – Non-example



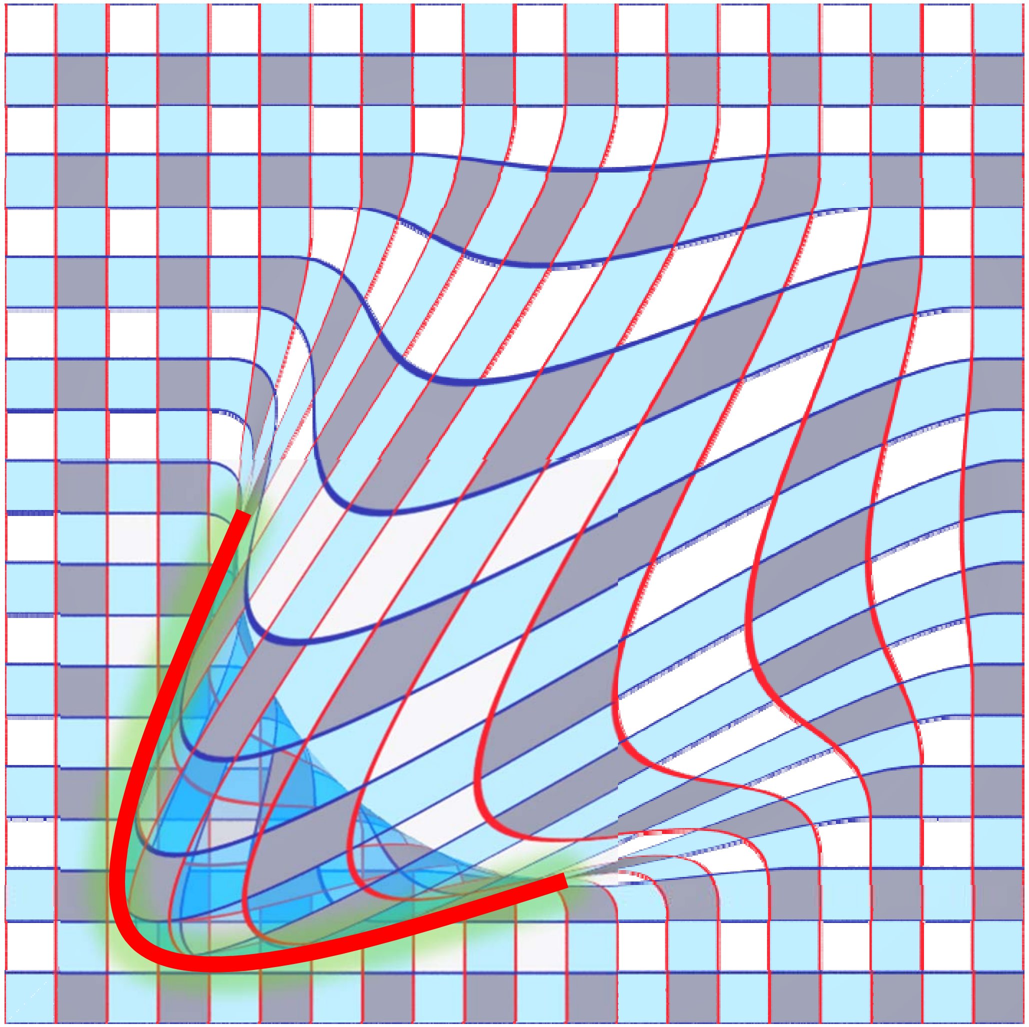
Locally Bijection – Non-example



Locally Bijection – Non-example

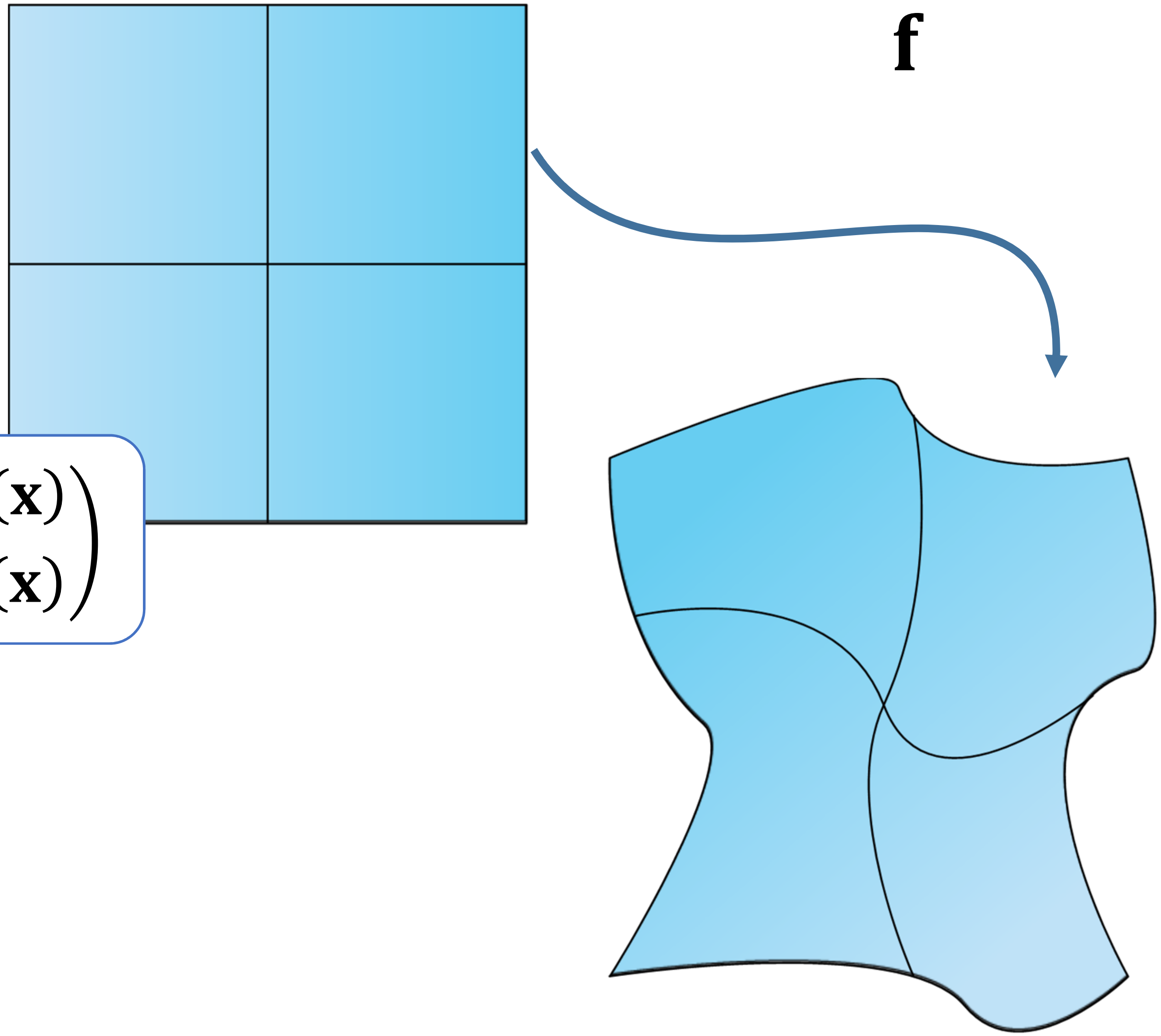


Locally Bijection – Non-example



Locally Bijection – Sufficient condition

$$\mathbf{f}(\mathbf{x}) = \begin{pmatrix} u(\mathbf{x}) \\ v(\mathbf{x}) \end{pmatrix}$$

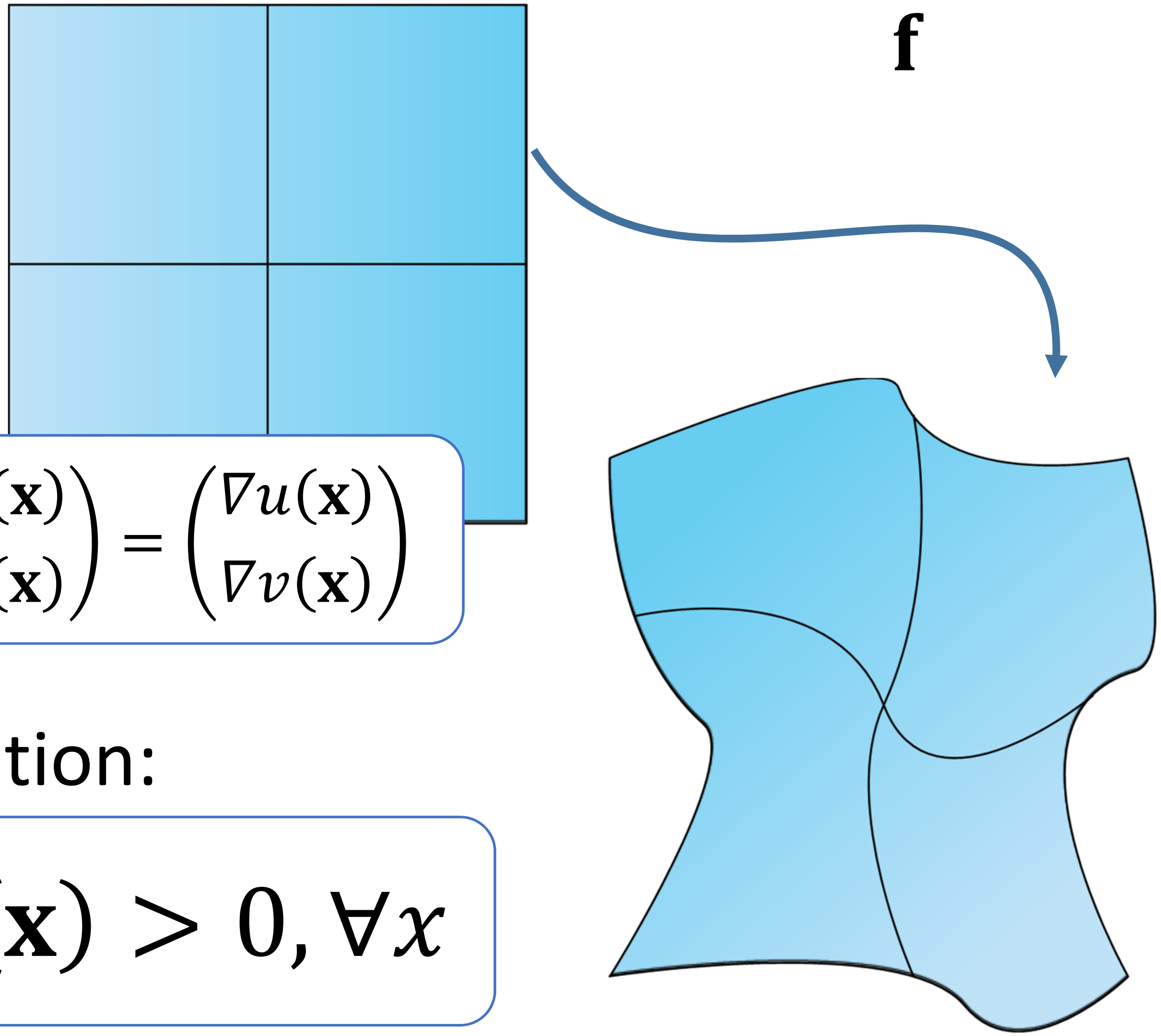


The Jacobian:

$$\mathcal{J}\mathbf{f}(\mathbf{x}) = \begin{pmatrix} \partial_x u(\mathbf{x}) & \partial_y u(\mathbf{x}) \\ \partial_x v(\mathbf{x}) & \partial_y v(\mathbf{x}) \end{pmatrix}$$

Locally Bijection – Sufficient condition

$$\mathbf{f}(\mathbf{x}) = \begin{pmatrix} u(\mathbf{x}) \\ v(\mathbf{x}) \end{pmatrix}$$



The Jacobian:

$$\mathcal{J}\mathbf{f}(\mathbf{x}) = \begin{pmatrix} \partial_x u(\mathbf{x}) & \partial_y u(\mathbf{x}) \\ \partial_x v(\mathbf{x}) & \partial_y v(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} \nabla u(\mathbf{x}) \\ \nabla v(\mathbf{x}) \end{pmatrix}$$

The Condition:

$$\det \mathcal{J}\mathbf{f}(\mathbf{x}) > 0, \forall x$$

Globally Bijective VS. Locally Bijective

Globally Bijective $\Longrightarrow$ Locally Bijective

$\mathbf{f}$ is bijective

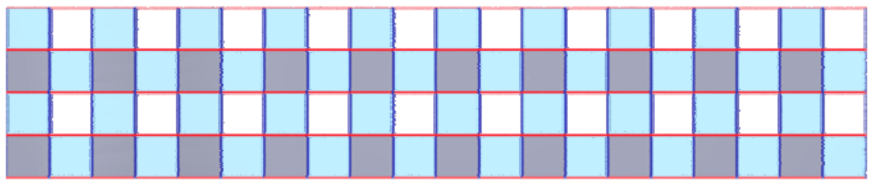
$\mathbf{f}: \mathbf{U} \rightarrow \mathbf{f}(\mathbf{U})$ is bijective

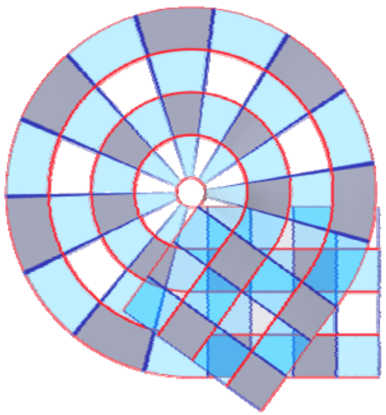
Globally Bijective VS. Locally Bijective

Globally Bijective ← Locally Bijective ?

$\mathbf{f}$ is bijective

$\mathbf{f}: \mathbf{U} \rightarrow \mathbf{f}(\mathbf{U})$ is bijective





Globally Bijective VS. Locally Bijective

Globally Bijective $\leftarrow$ Locally Bijective

$\mathbf{f}$ is bijective

$\mathbf{f}: \mathbf{U} \rightarrow \mathbf{f}(\mathbf{U})$ is bijective

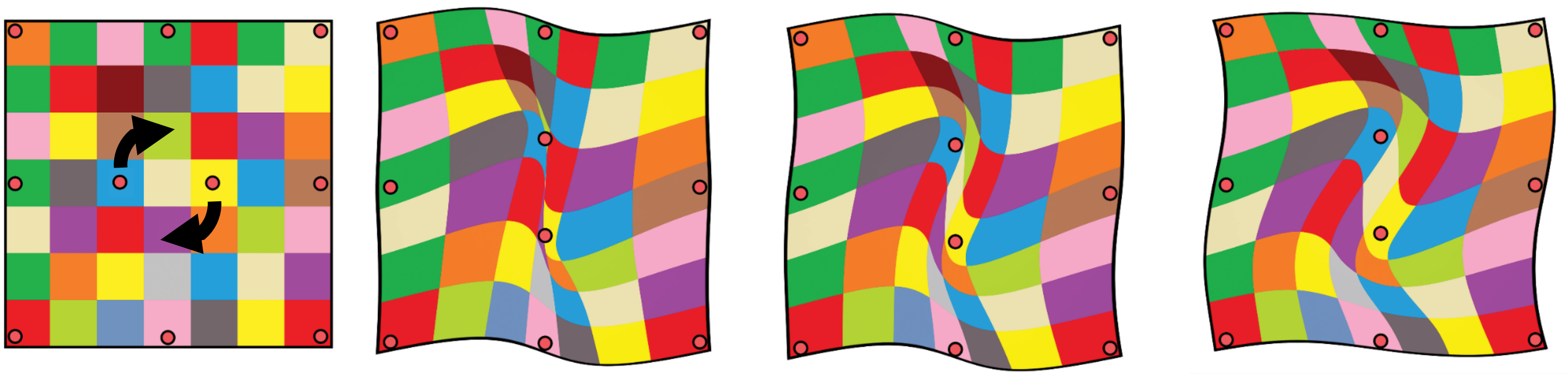
Google: “Global inversion theorems”

What are good maps?

Local

Bijectivity

Low distortion



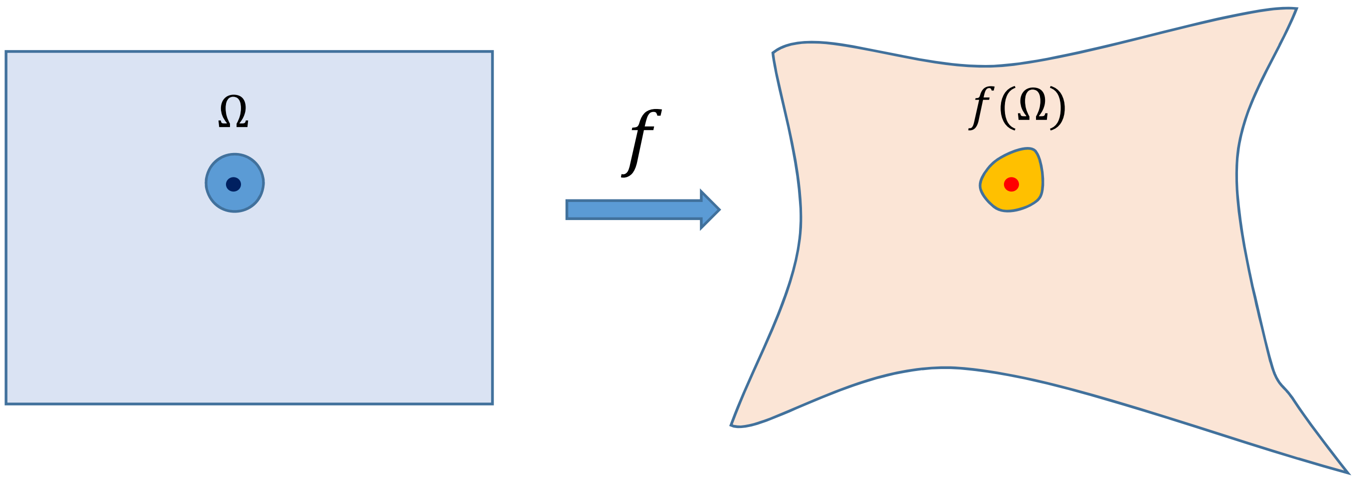
Not Bijective

Bijective

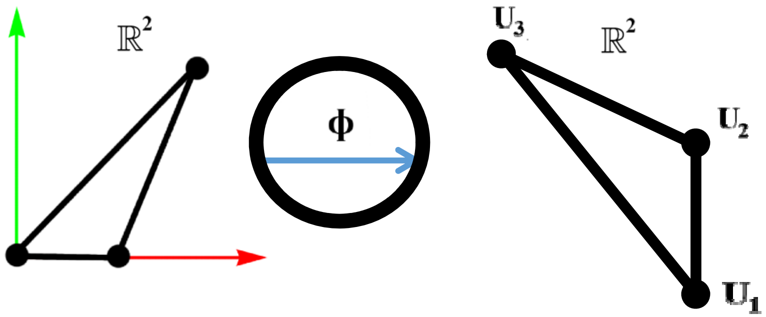
Lower distortion

Jacobian的几何意义

- 函数在某点的Jacobian度量了其局部的形变量

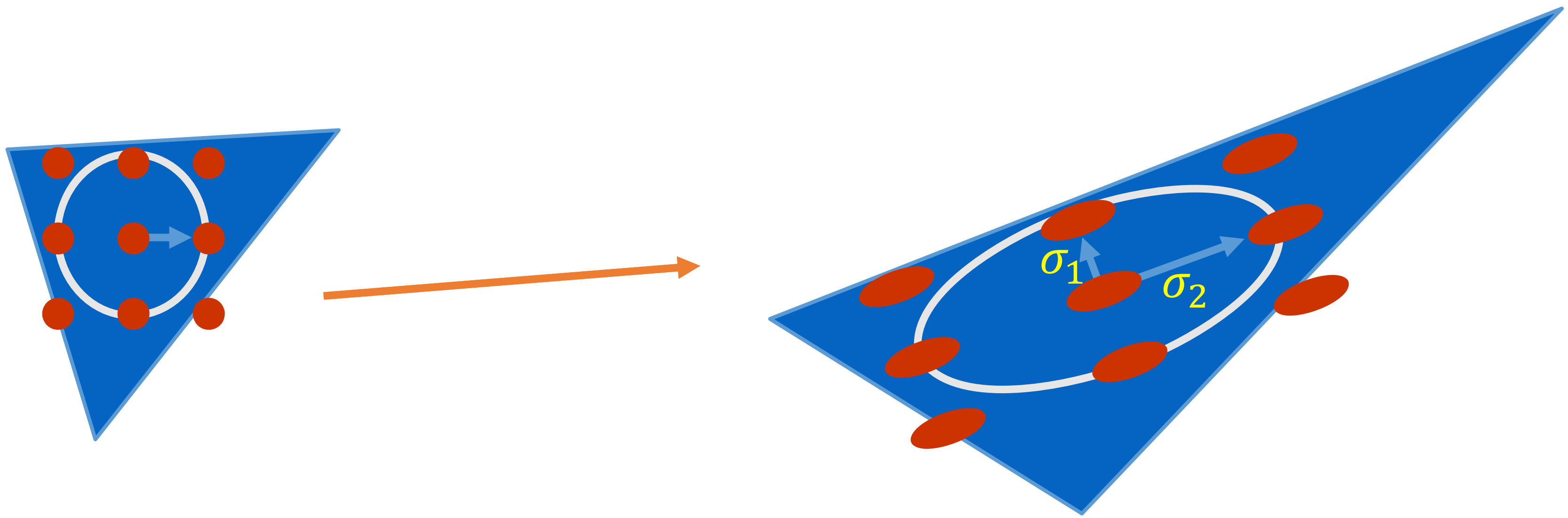


Distortion Measure

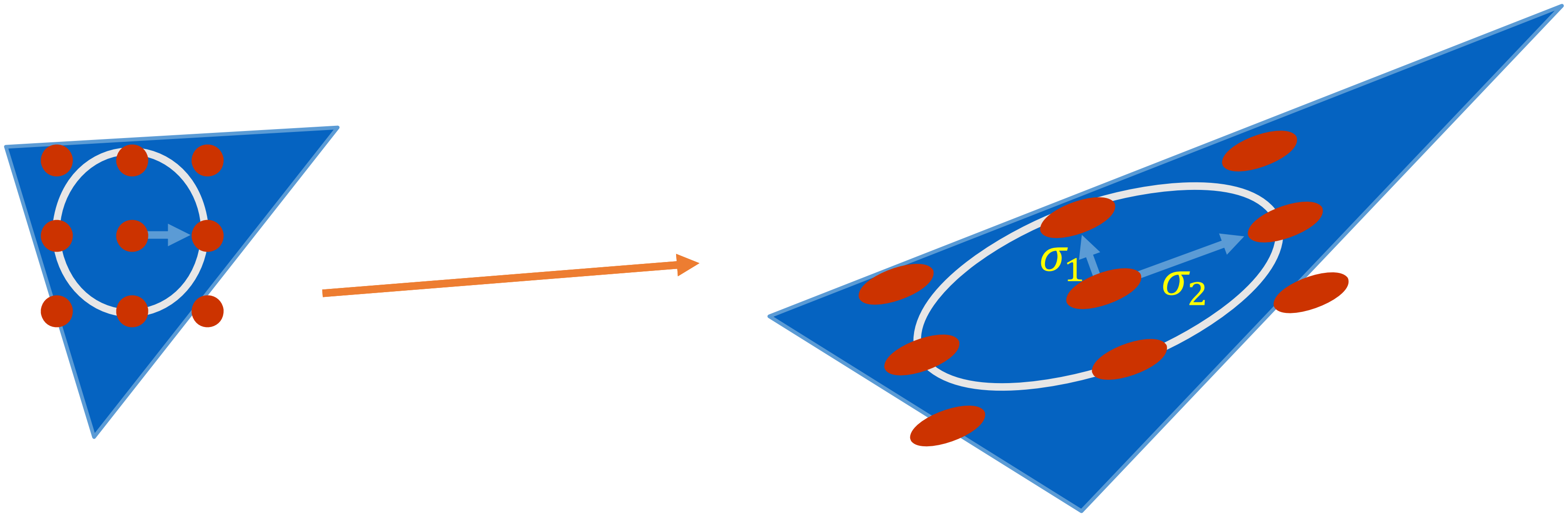


$$L = U \begin{pmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{pmatrix} V^*$$

$$\sigma_2 \geq \sigma_1$$



Distortion Measure



- angle-preserving (*conformal*) $\sigma_1 = \sigma_2$
- area-preserving (authalic) $\sigma_1 \sigma_2 = 1$
- length-preserving (*isometric*) $\sigma_1 = \sigma_2 = 1$

Distortion Metric

- Conformal

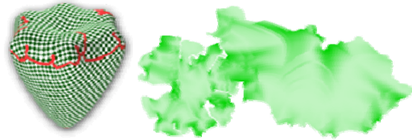
  [Degener et al. 2003]

  $$\frac{\sigma_2}{\sigma_1}$$

- Maximal Isometric Distortion

  [Sorkine et al. 2002]

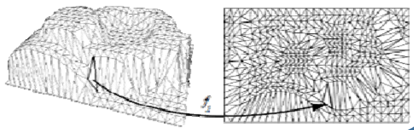
  $$\max(\sigma_2, \frac{1}{\sigma_1})$$



- MIPS

  [Hormann and Greiner 2000]

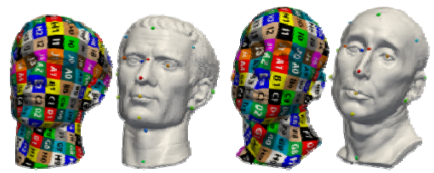
  $$\frac{\sigma_1}{\sigma_2} + \frac{\sigma_2}{\sigma_1}$$



- Isometric

  [Aigermann et al. 2014]

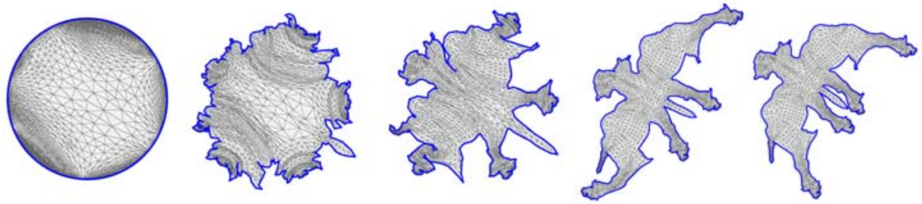
  $$\sqrt{\sigma_2^2 + \frac{1}{\sigma_1^2}}$$



- Symmetric Dirichlet energy

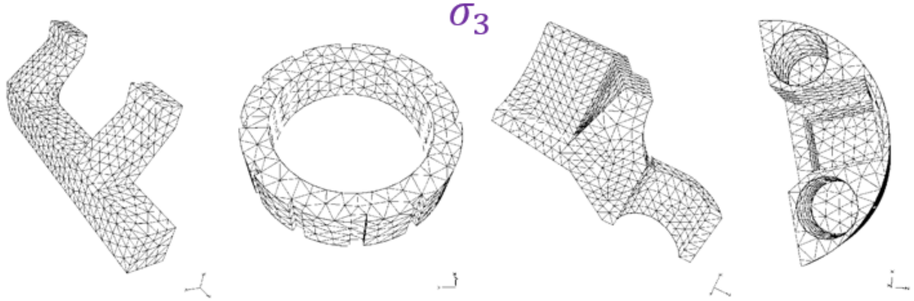
  [Smith and Schaefer 2015]

  $$\sigma_1^2 + \frac{1}{\sigma_1^2} + \sigma_2^2 + \frac{1}{\sigma_2^2}$$



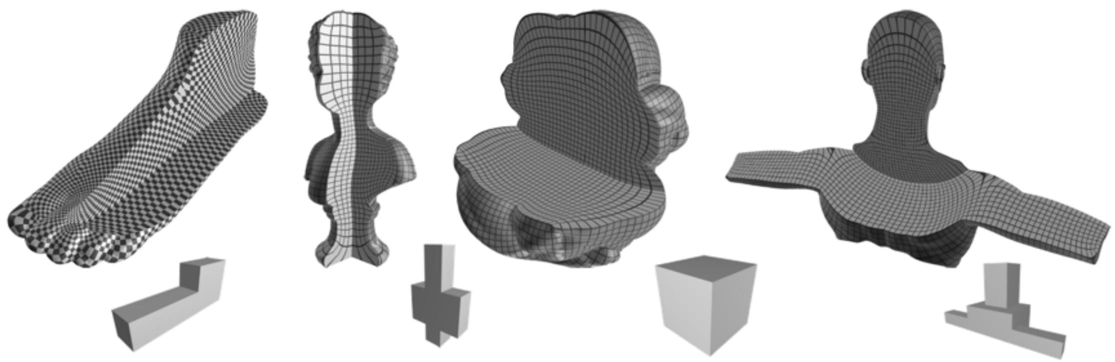
Distortion Metric in 3D

$$\frac{\sigma_1}{\sigma_3}$$



[Freitag & Knupp 2002]

$$\frac{\sigma_1}{\sigma_3} + \frac{\sigma_3}{\sigma_1}; \quad \sigma_1\sigma_2\sigma_3 + \frac{1}{\sigma_1\sigma_2\sigma_3}$$



[Paillé & Poulin 2012]

映射的优化模型

Recap: Formulation of Parameterization

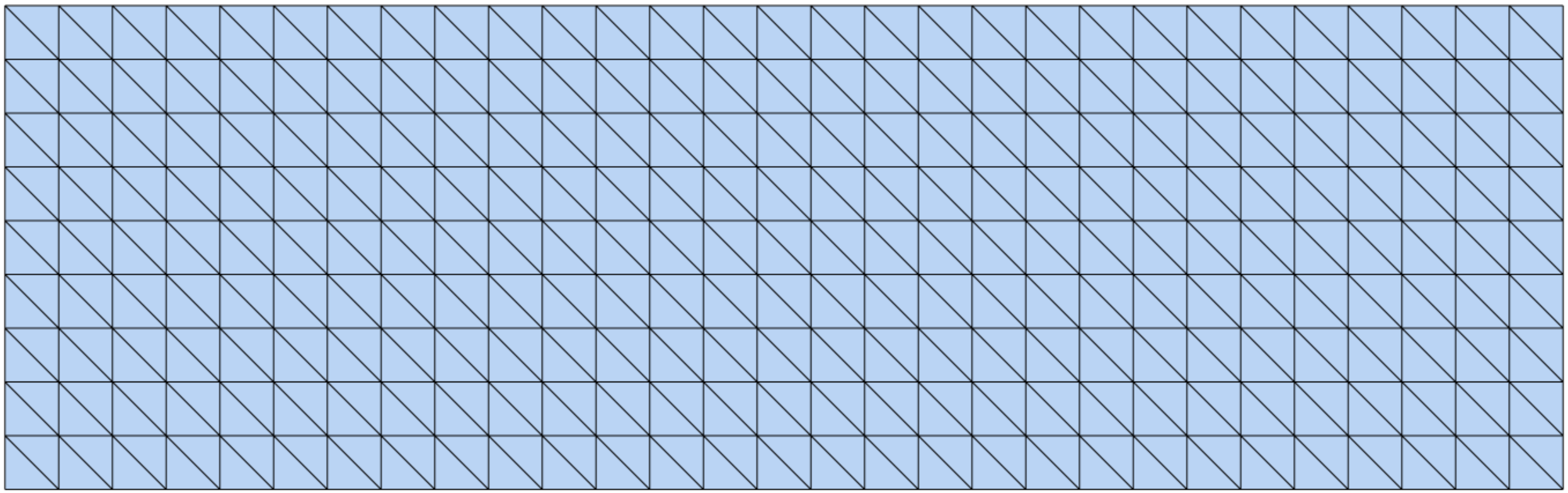
$$\min_{V} E(V) = \sum_{t \in T} (\sigma_1^2 + \frac{1}{\sigma_1^2} + \sigma_2^2 + \frac{1}{\sigma_2^2})$$

s.t. $\quad \sigma_1 \sigma_2 > 0, \quad \forall t$

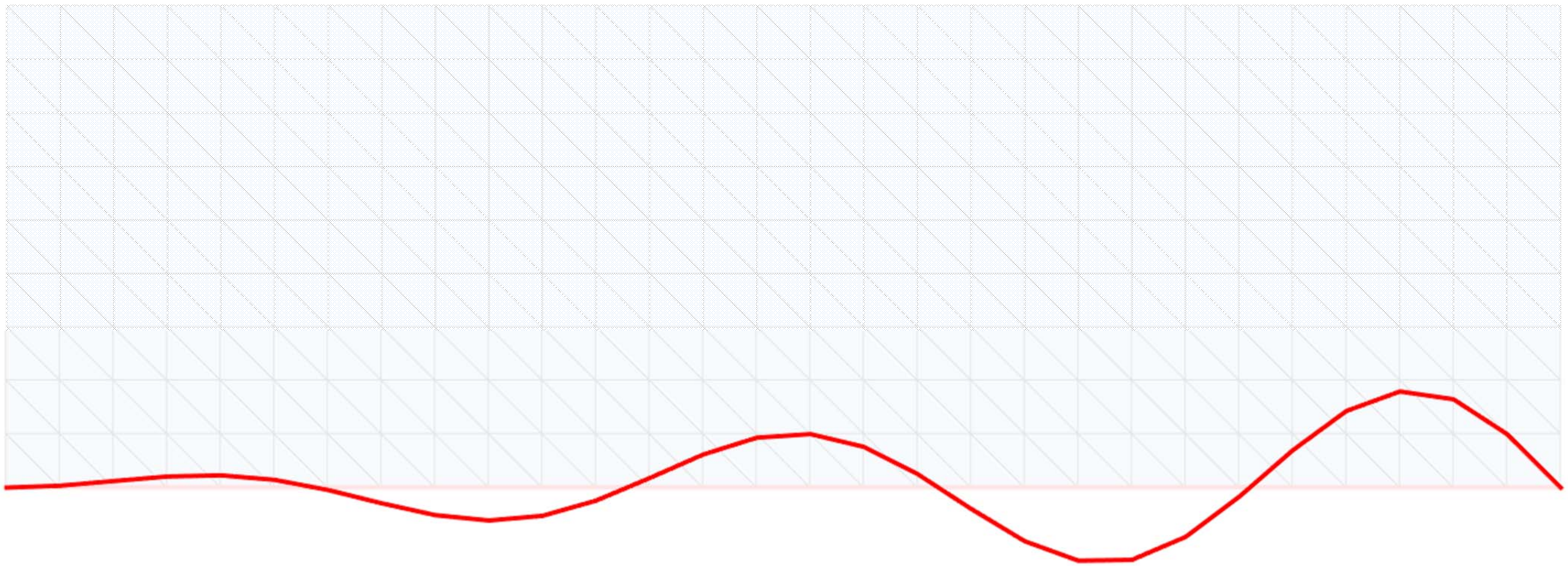
- The cost function is highly nonlinear and nonconvex
- The constraints are nonlinear
- The Heissian matrix is highly non-definite

Computationally expensive for large scale meshes!

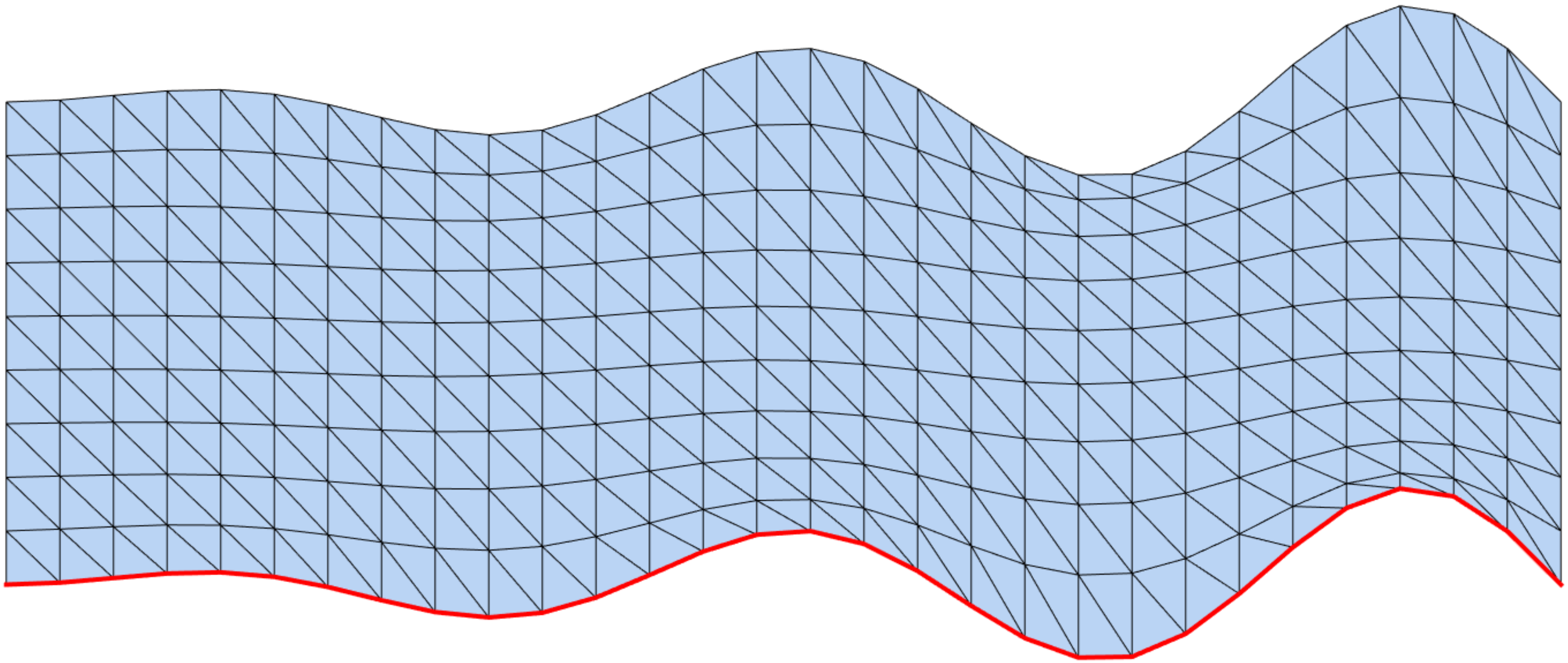
Computing maps



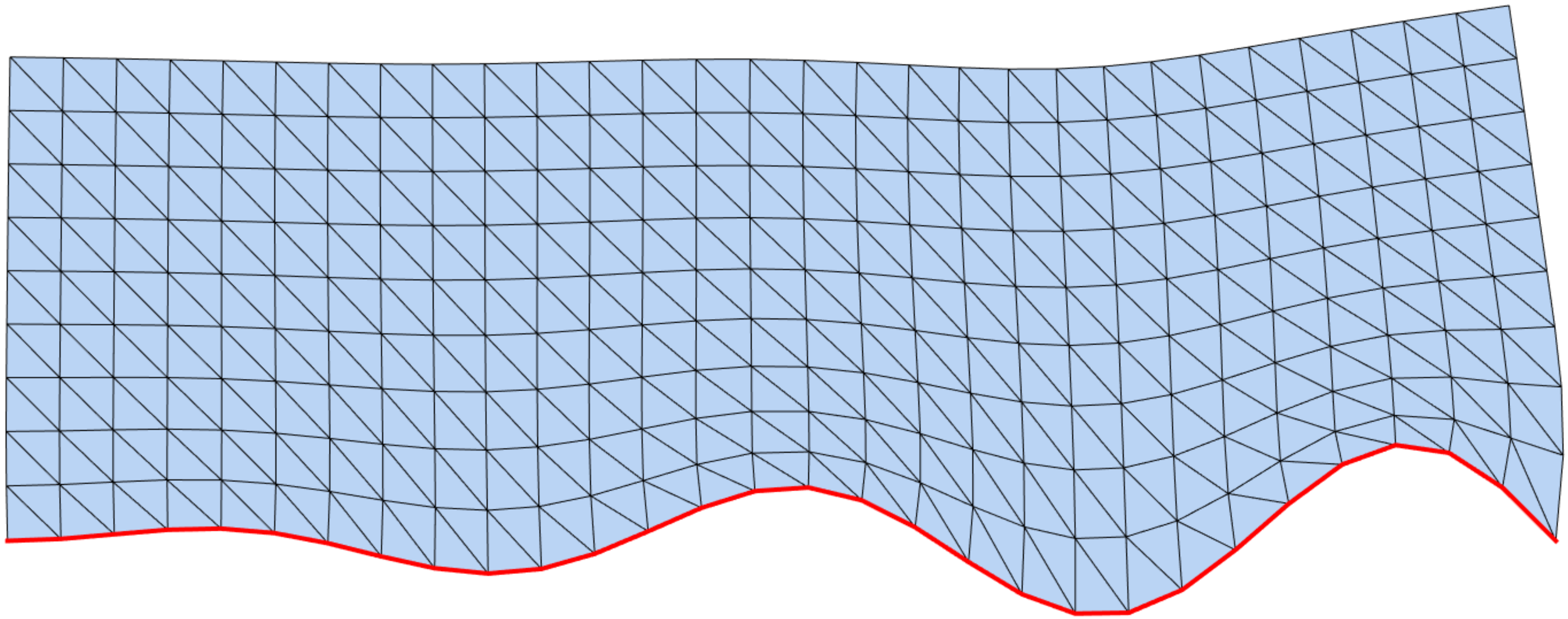
Computing maps



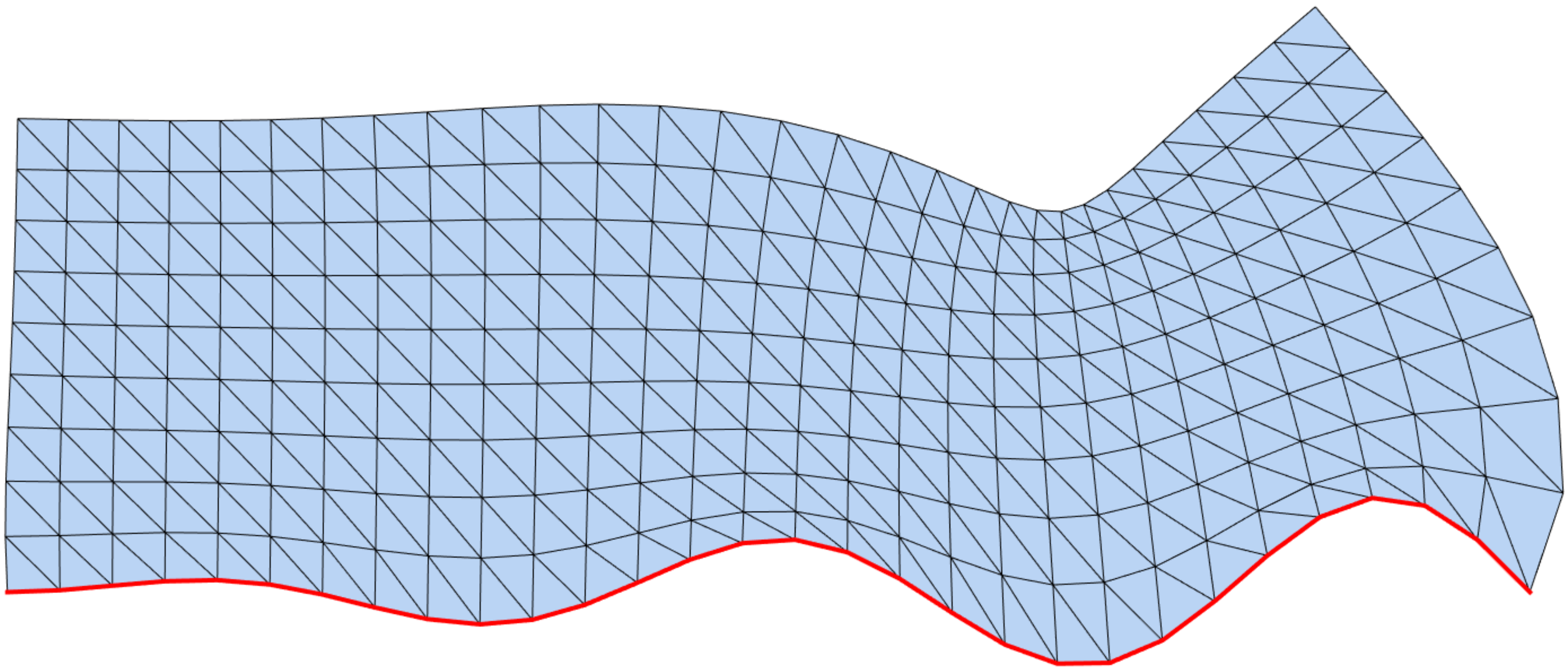
Computing maps



Computing maps



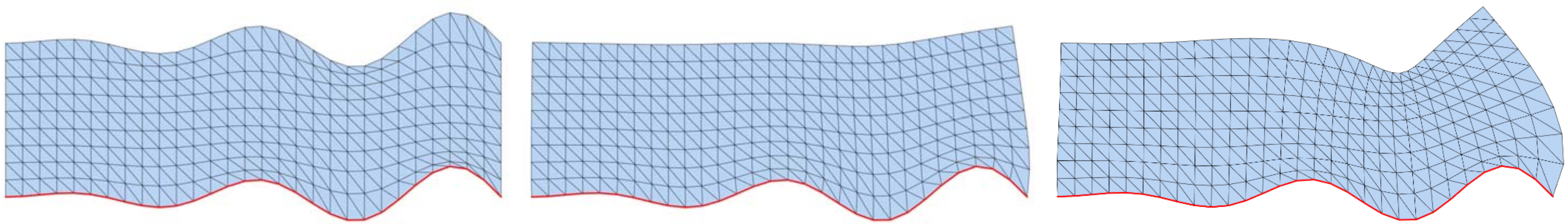
Computing maps



Computing maps

- Imposing constraints

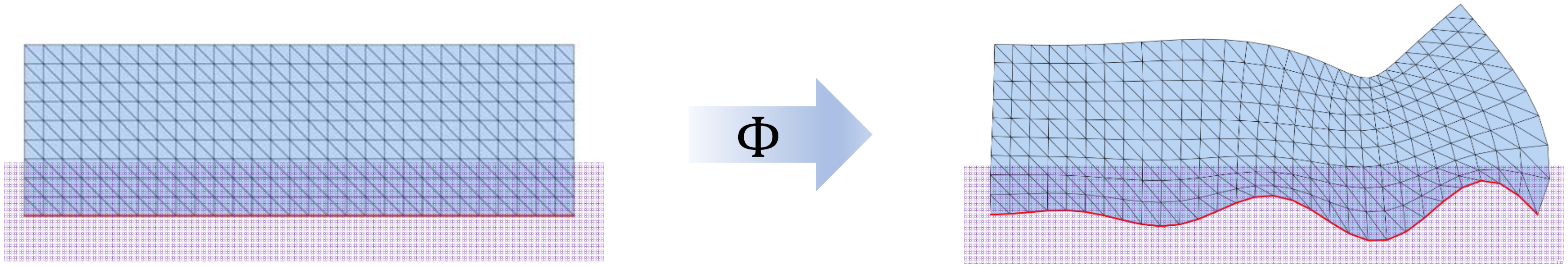
- Finding maps that are most…



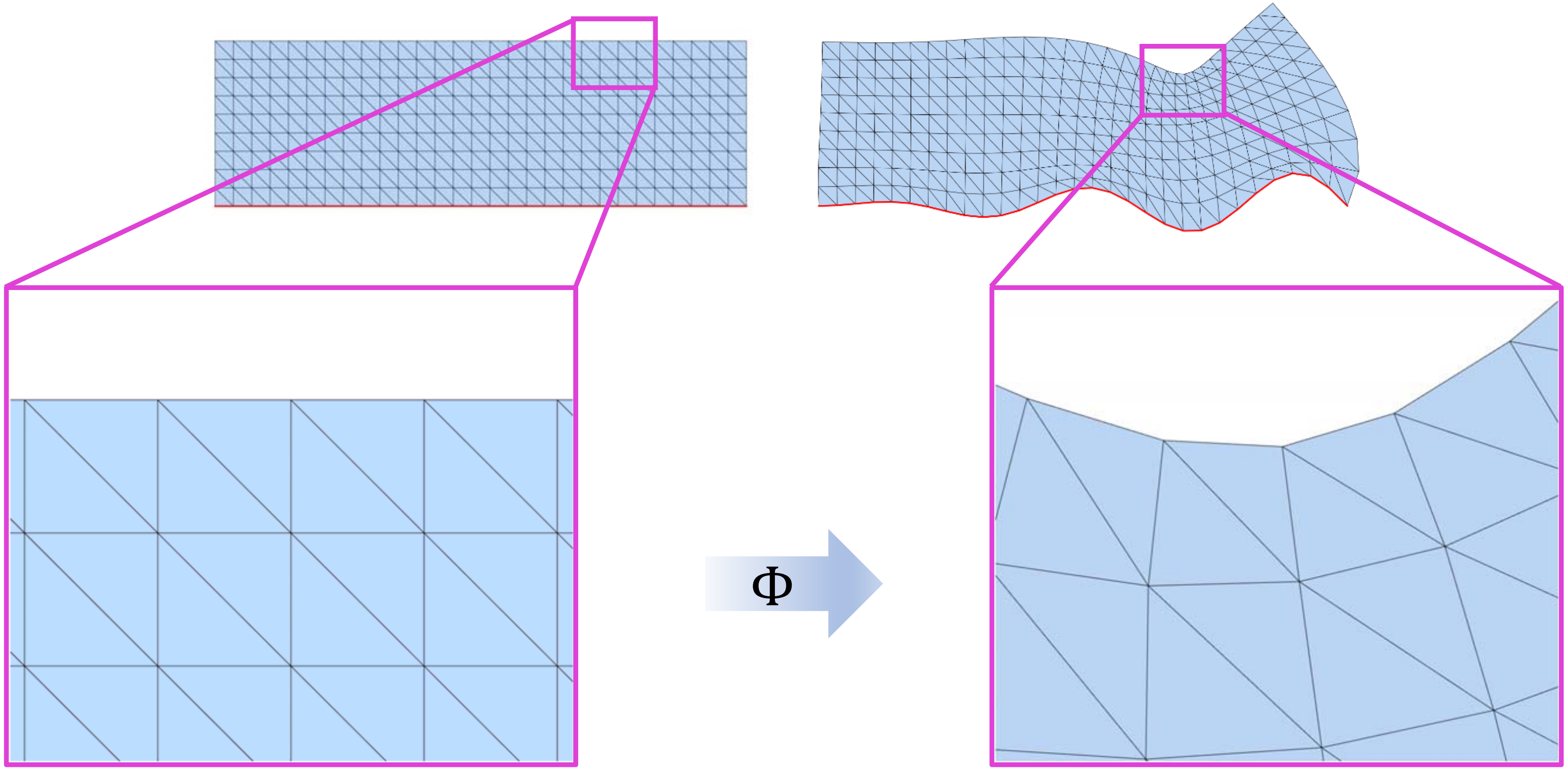
Constrained Optimization

$$\begin{aligned} &\underset{\Phi}{\operatorname{argmin}} \quad E(\Phi) \\ &\text{s.t.} \qquad\quad \Phi \in K \end{aligned}$$

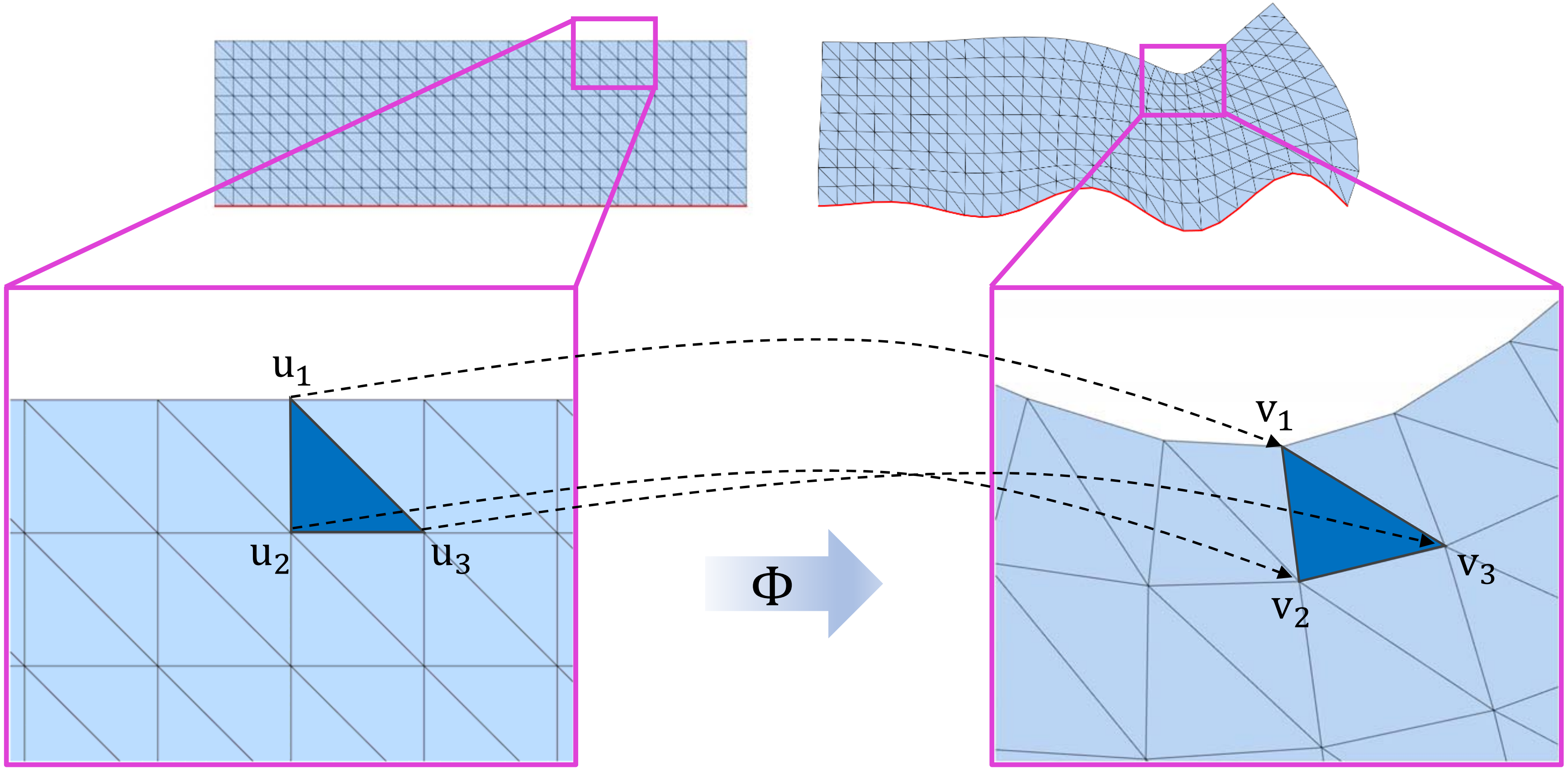
$\underset{\Phi}{\operatorname{argmin}}\ E(\Phi)$	Energy
s.t. $\Phi \in K$	Constraints



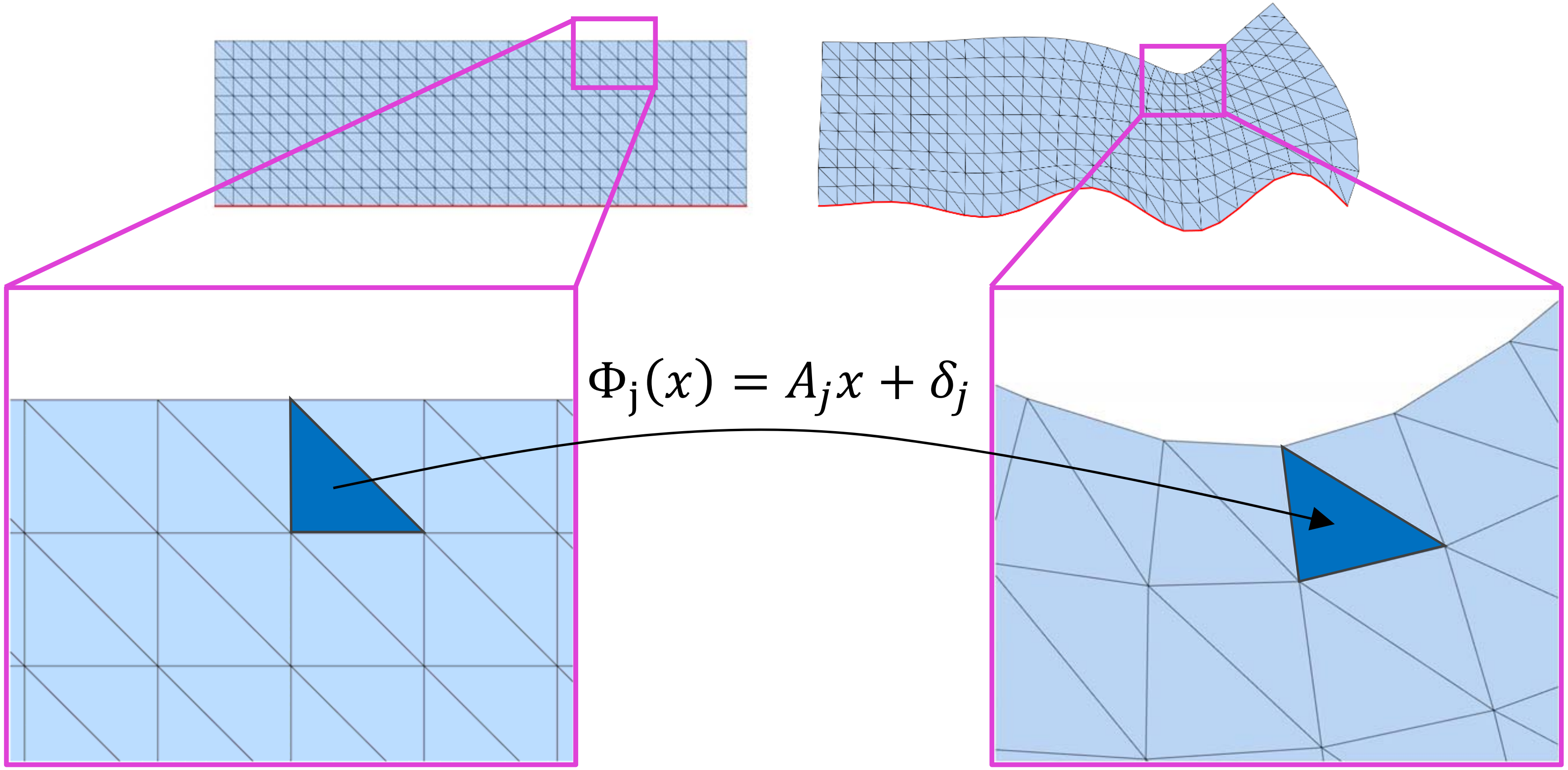
Energy



Energy



Energy

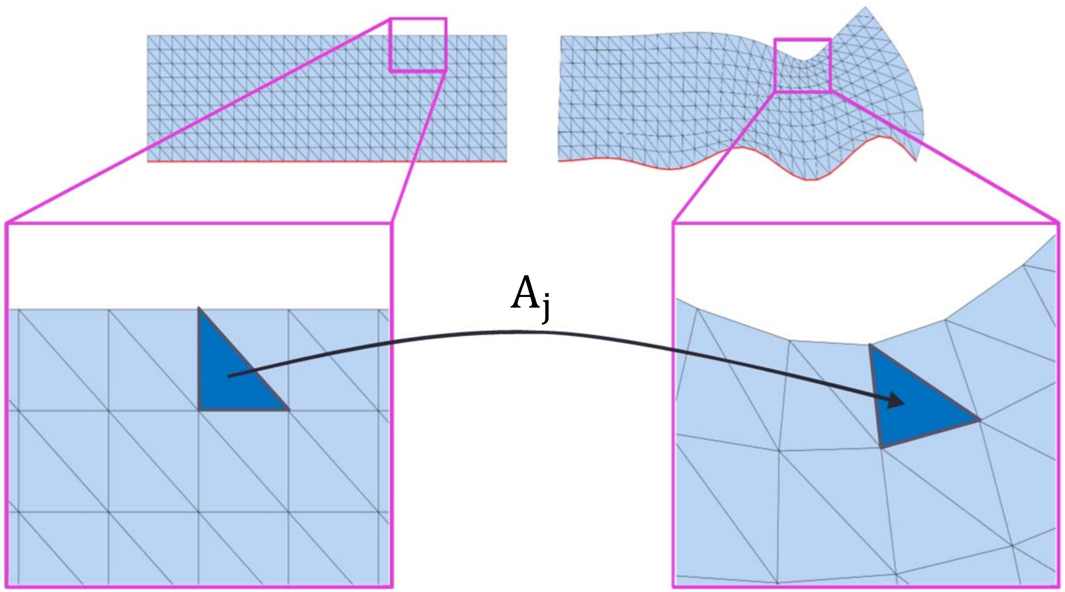


$$E(\Phi) = E(A_1, \ldots, A_m)$$

Map optimization

- In terms of differentials:

$$\operatorname{argmin} E(\mathrm{A}_1, \ldots, \mathrm{A}_\mathrm{m})$$

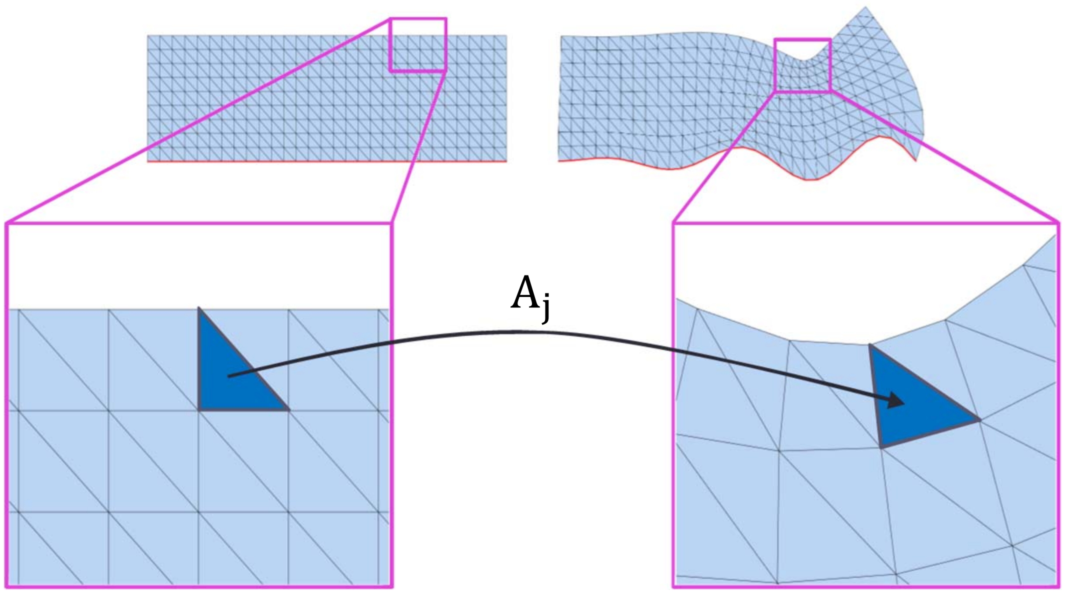


Map optimization

- In terms of differentials:

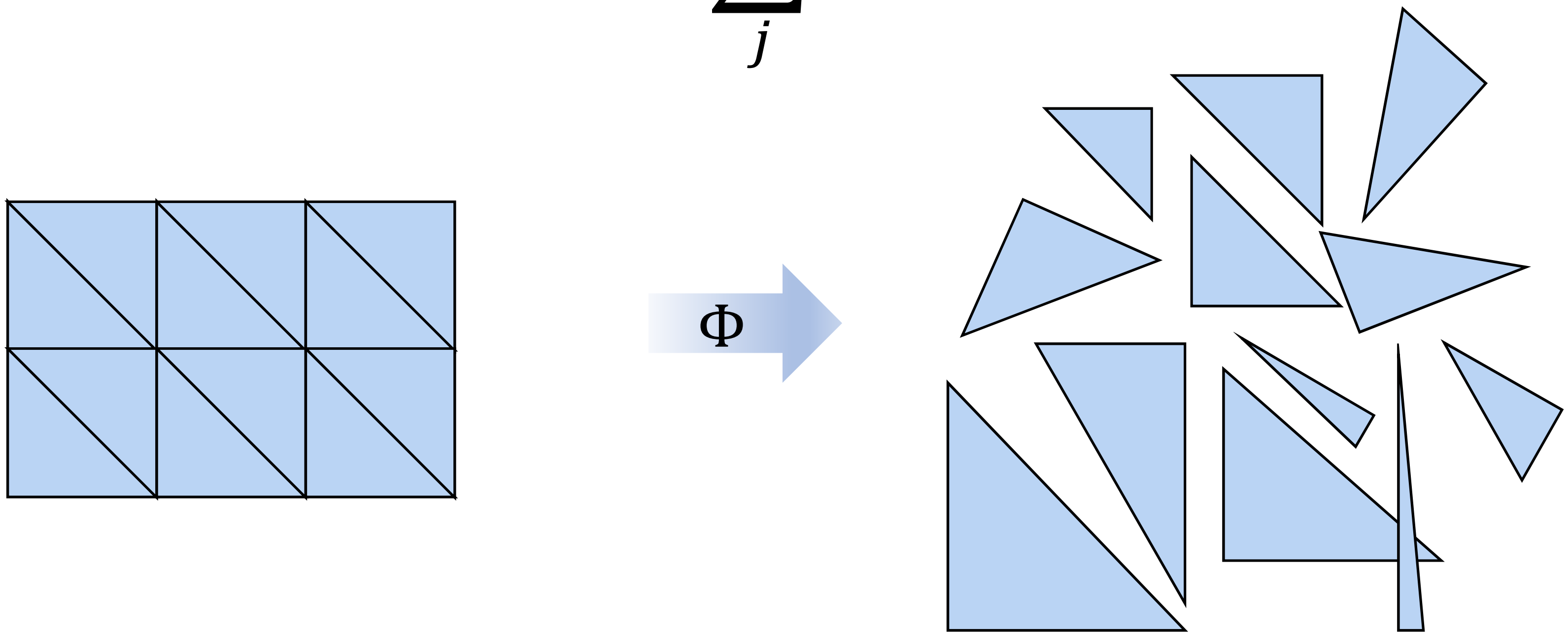
$$\operatorname{argmin} \sum_j f(A_j)$$

Separable



Map optimization

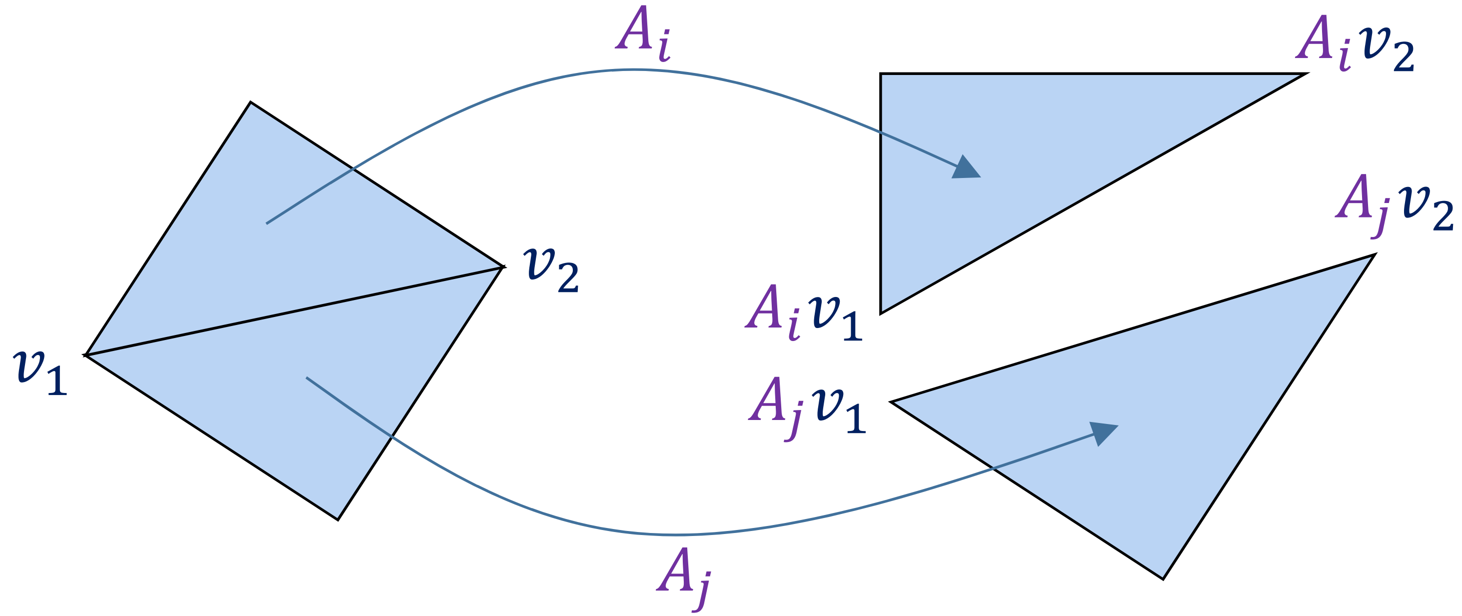
$$\operatorname{argmin} \sum_j f(A_j)$$



Must impose continuity!

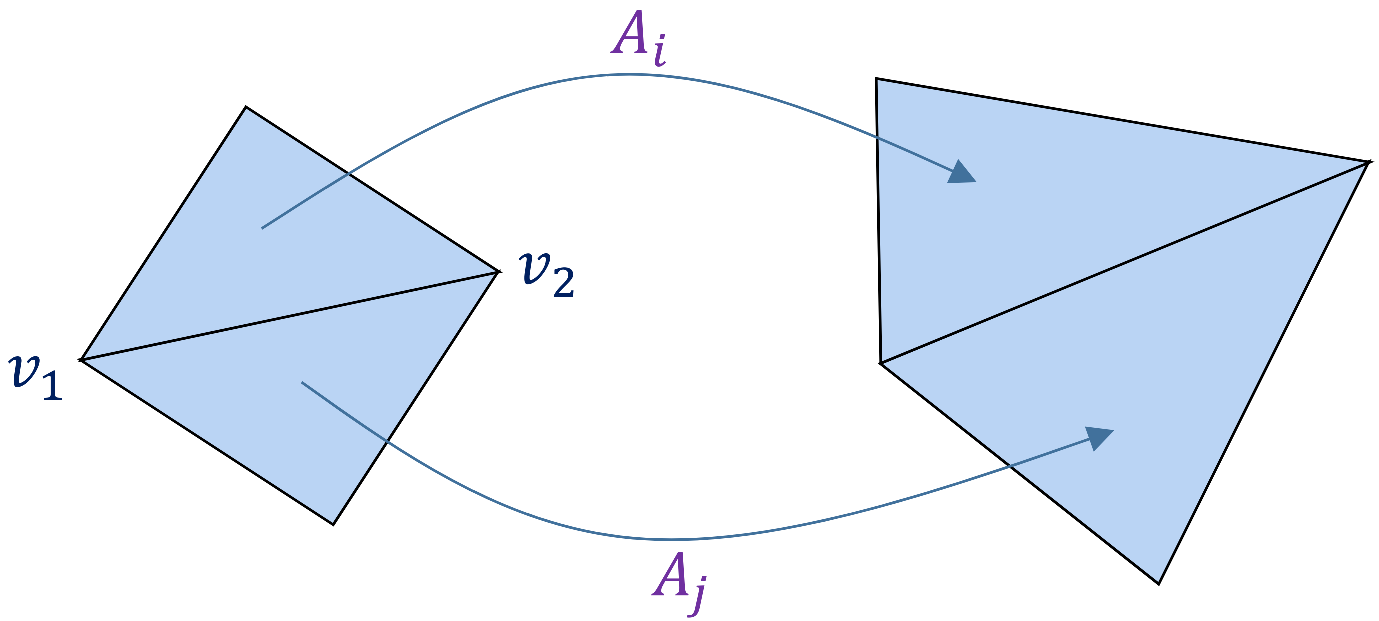
Explicit continuity

- Optimization variables: $A_1, A_2, \dots, A_m$
- Adjacent A_j's must agree



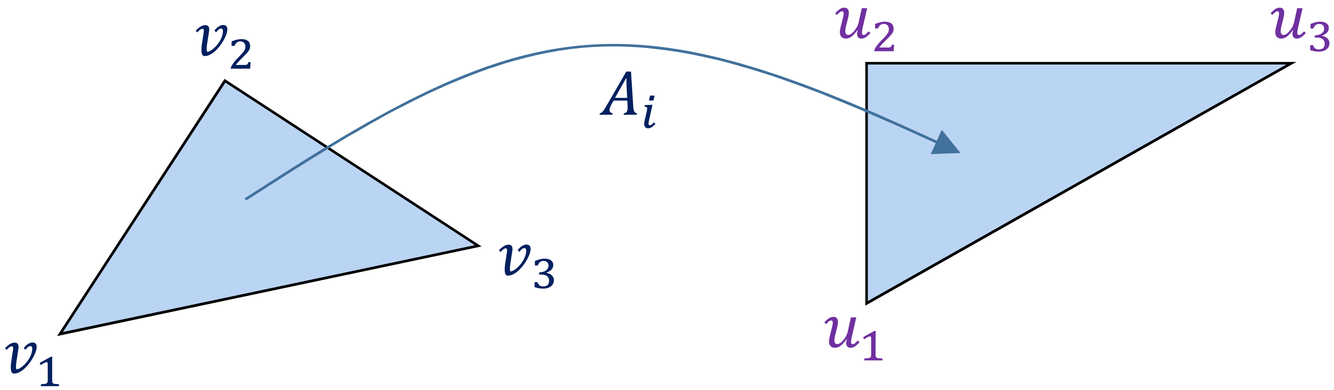
Explicit continuity

- Optimization variables: $A_1, A_2, \ldots, A_m$
- Adjacent A_j's must agree



$$A_i v_1 = A_j v_1$$
$$A_i v_2 = A_j v_2$$

Implicit continuity



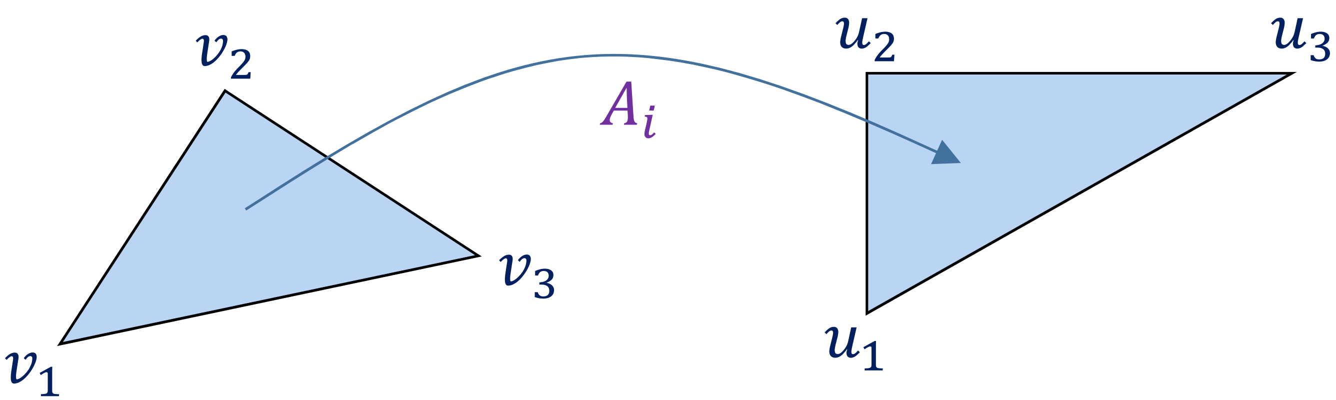
$$A_i \overline{[v_1\ v_2\ v_3]} = \overline{[u_1\ u_2\ u_3]}$$

$$A_i = \overline{[u_1\ u_2\ u_3]}\ \overline{[v_1\ v_2\ v_3]}^{\dagger}$$

$$A_i = A_i(U)$$

Linearly express A_i's in terms of U

Implicit continuity



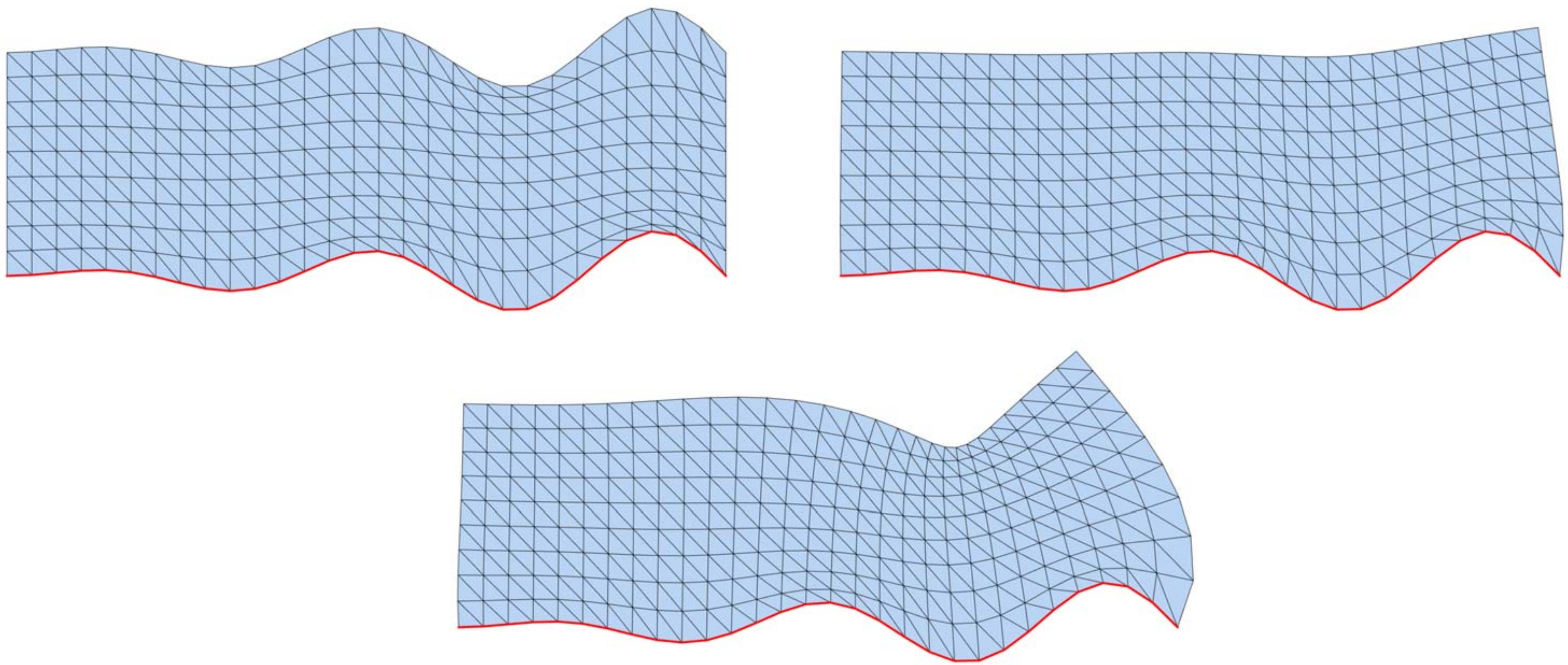
- Optimization variables: $u_1, u_2, \dots, u_n \quad (U)$

$$E(\Phi) = \sum_j f\left(A_j(U)\right)$$

几何优化的求解

Popular energies

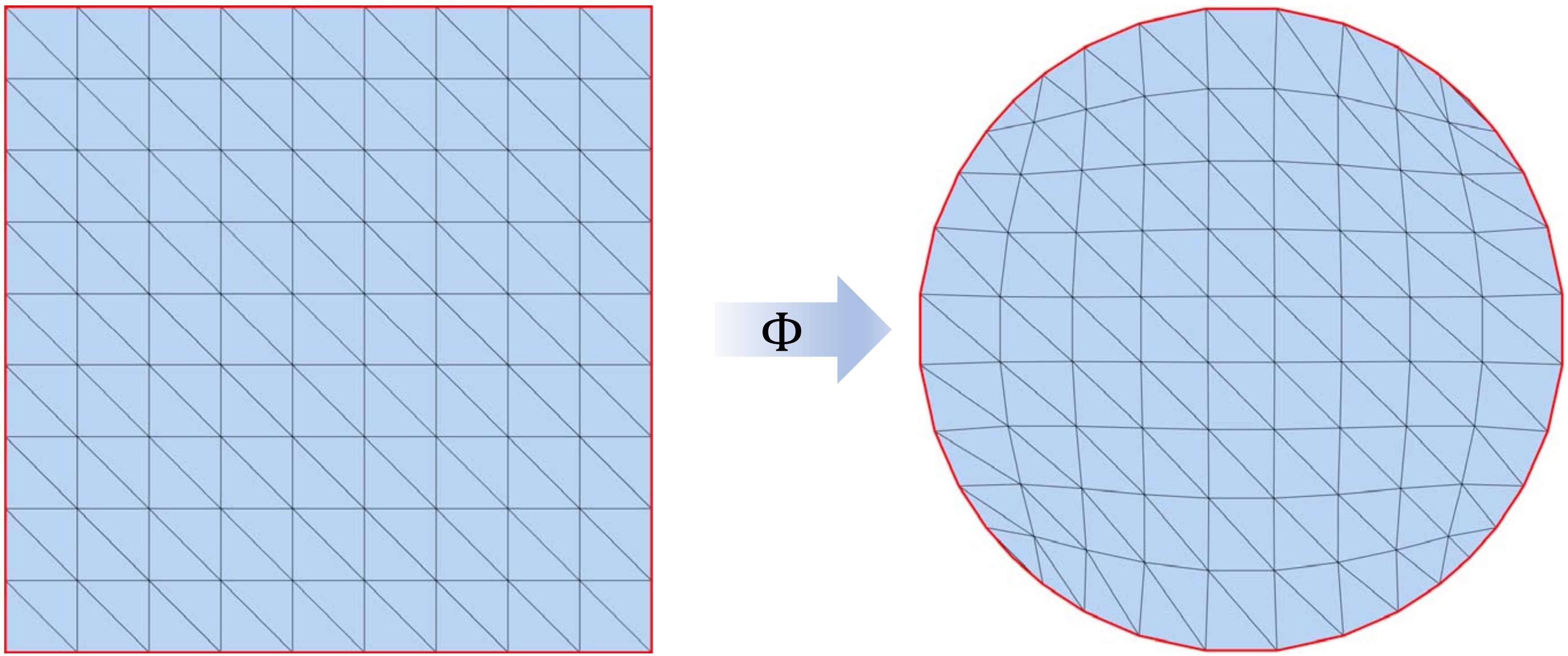
$$\operatorname{argmin} \sum_j f(A_j)$$



Dirichlet

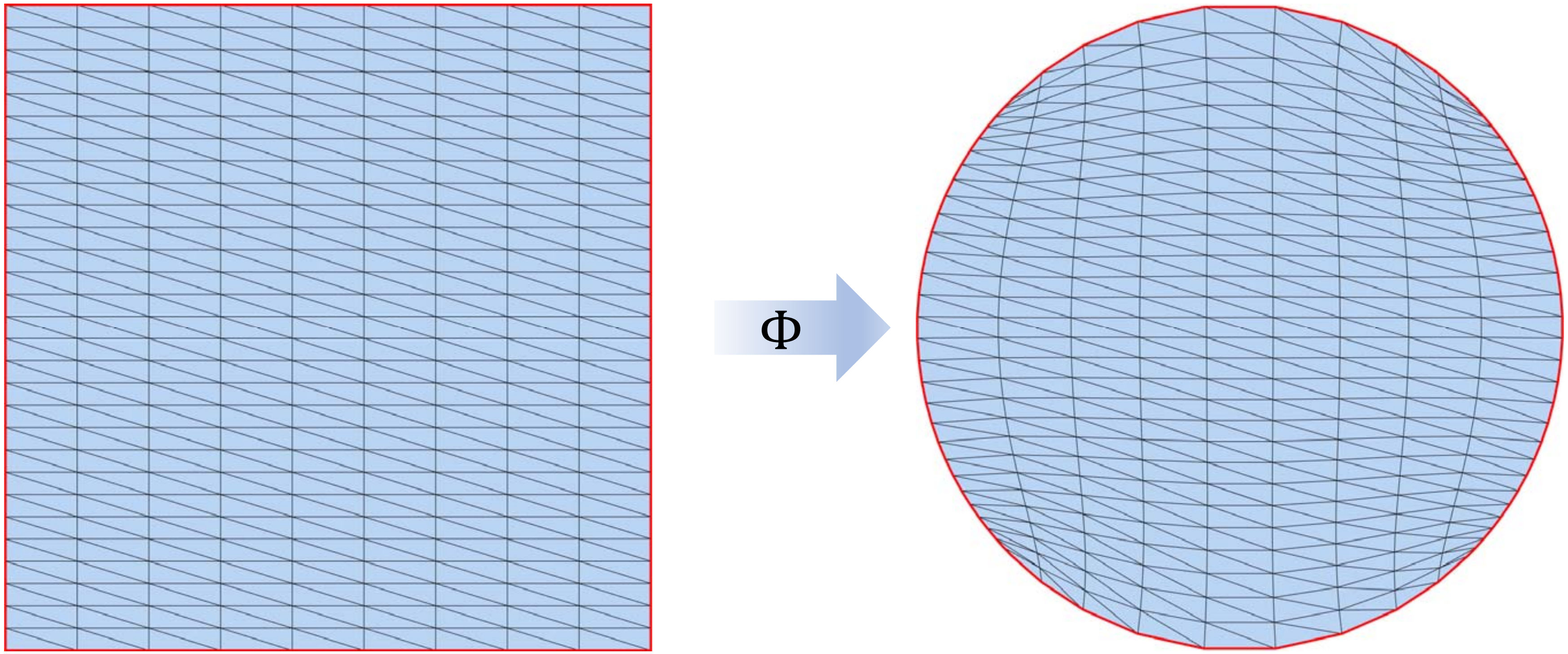
area / volume

$$E_D = \sum_j w_j \|A_j\|_F^2$$



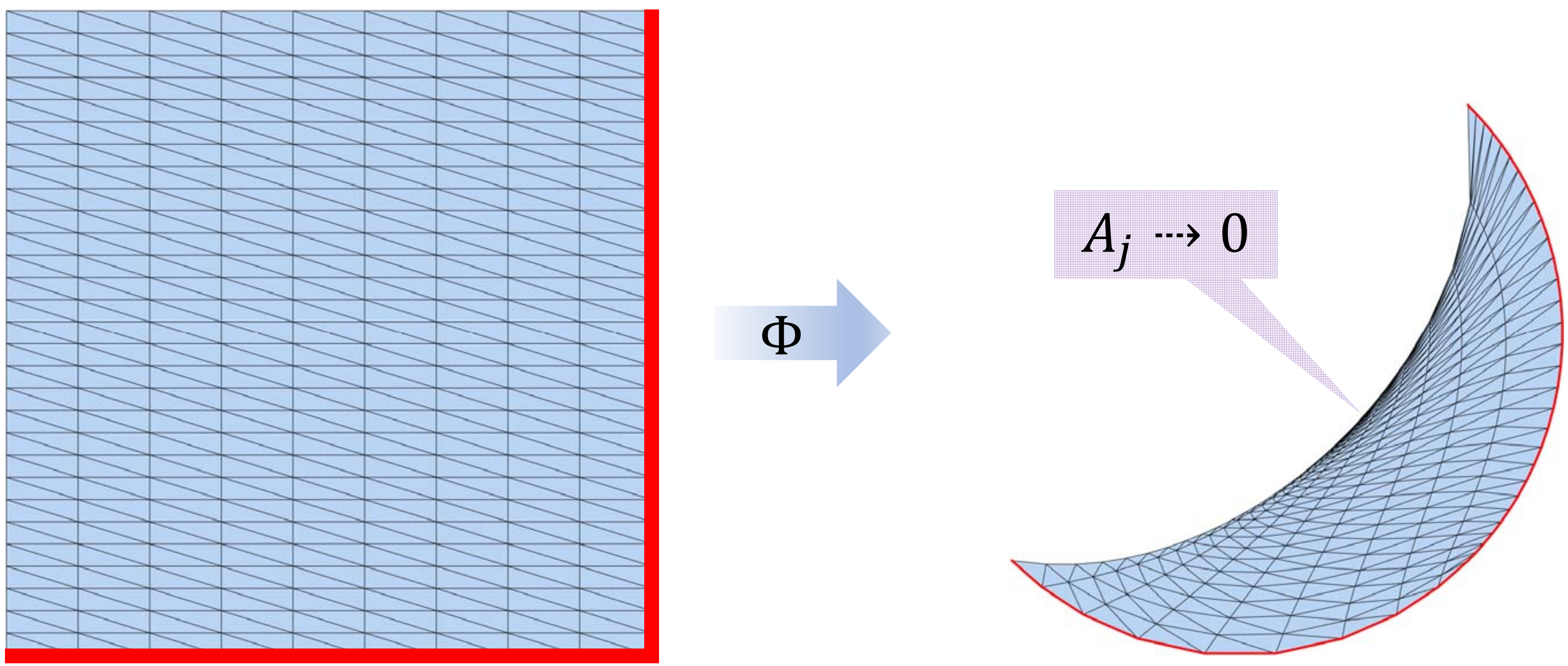
Dirichlet

$$E_D = \sum_j w_j \left\| A_j \right\|_F^2$$



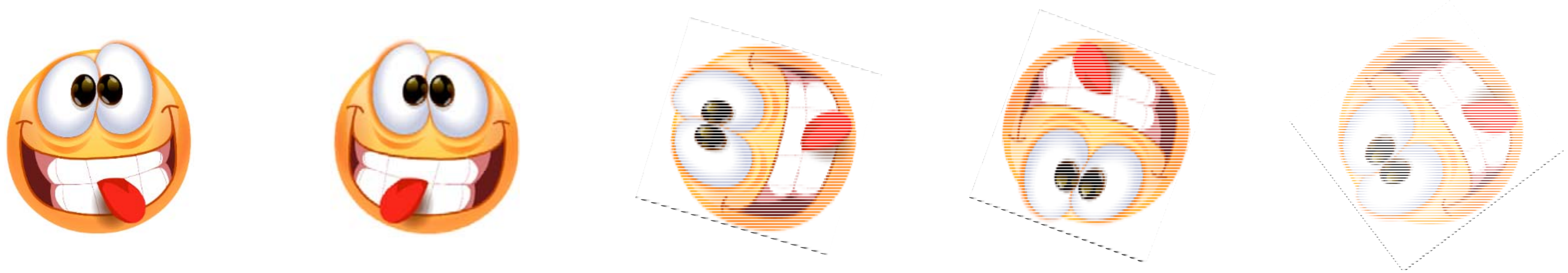
Dirichlet

$$E_D = \sum_j w_j \|A_j\|_F^2$$

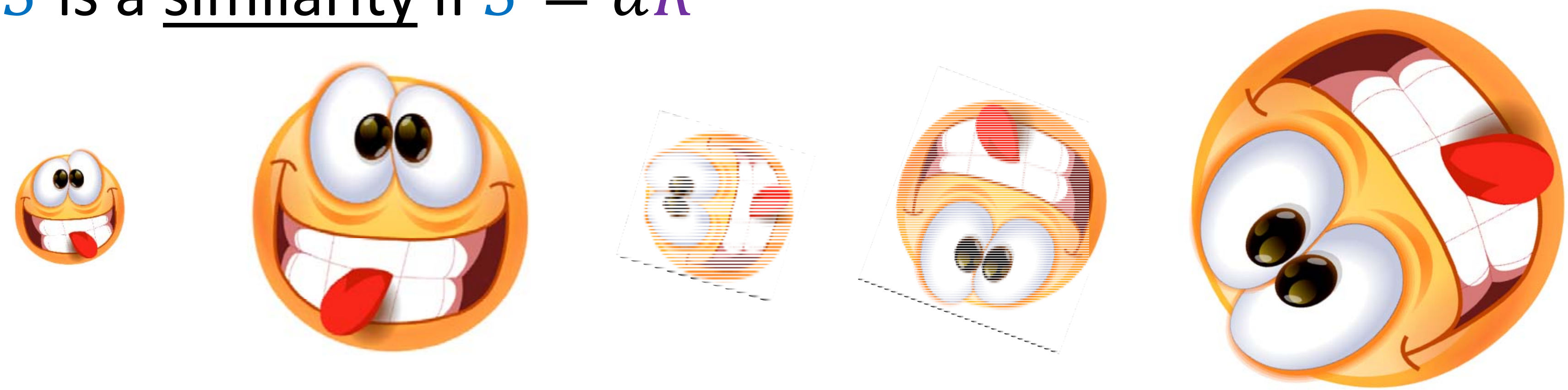


Orthogonal and Similarity

- R is orthogonal if $R^T = R^{-1}$
  (rotation if $\det R > 0$)



- S is a similarity if $S = \alpha R$



Closest R and S

- $\mathcal{R}(A)$ = closest orthogonal/rotation matrix to A
- $\mathcal{S}(A)$ = closest similarity matrix to A

- Computable using SVD/SSVD:

$$A = U\Sigma V^T; \qquad \Sigma = \mathrm{diag}(\sigma_1, \dots, \sigma_n)$$

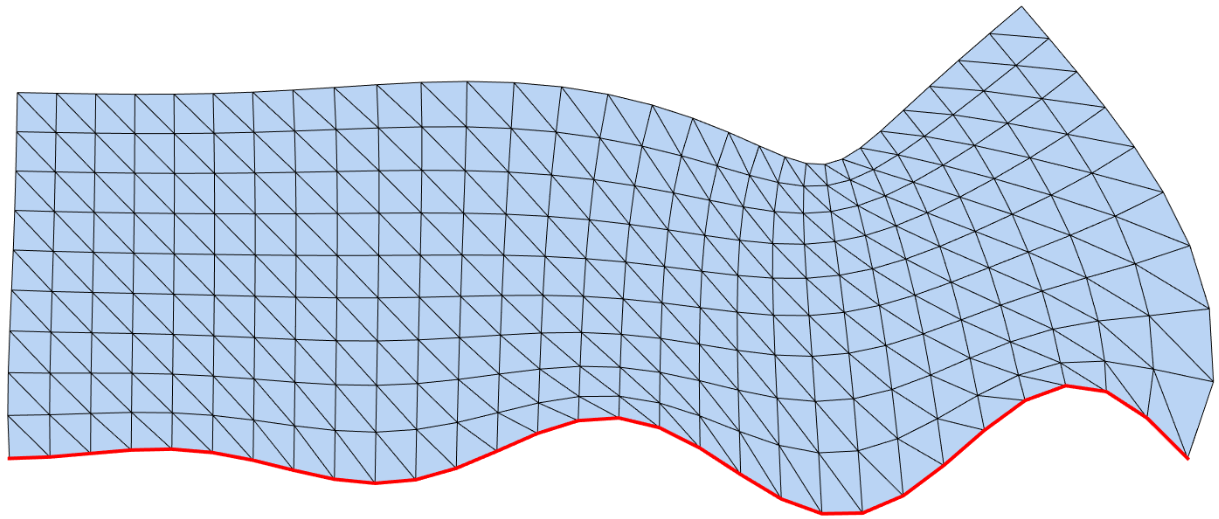
- $\mathcal{R}(A) = U\cancel{\Sigma}V^T = UV^T$
- $\mathcal{S}(A) = \overline{\sigma}UV^T$

mean of SVs

As-Similar-As-Possible (ASAP)

$$E_L = \sum_j w_j \left\| A_j - \mathcal{S}(A_j) \right\|_F^2$$

closest similarity

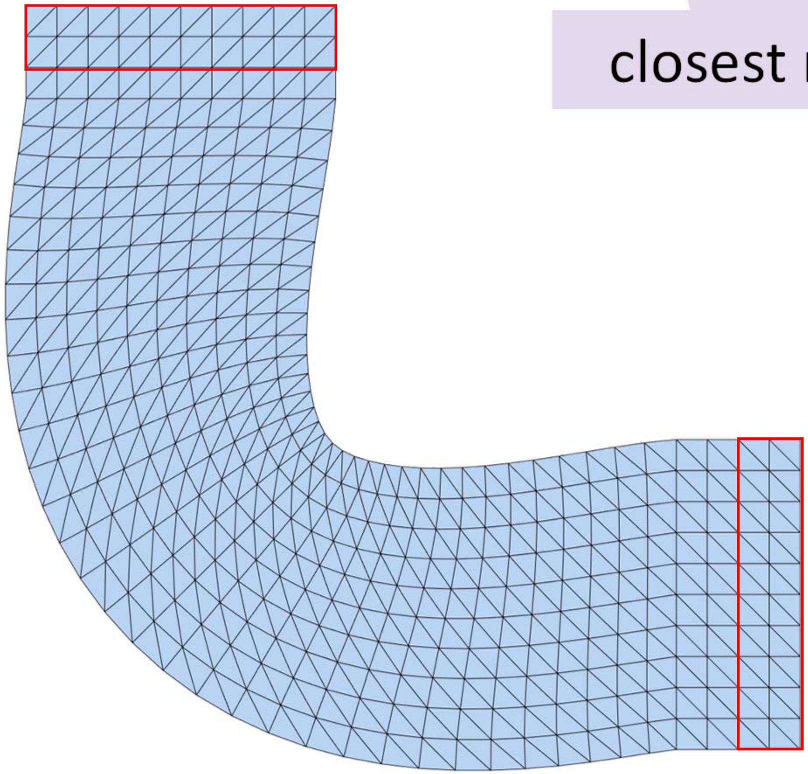


Solving sparse linear system!

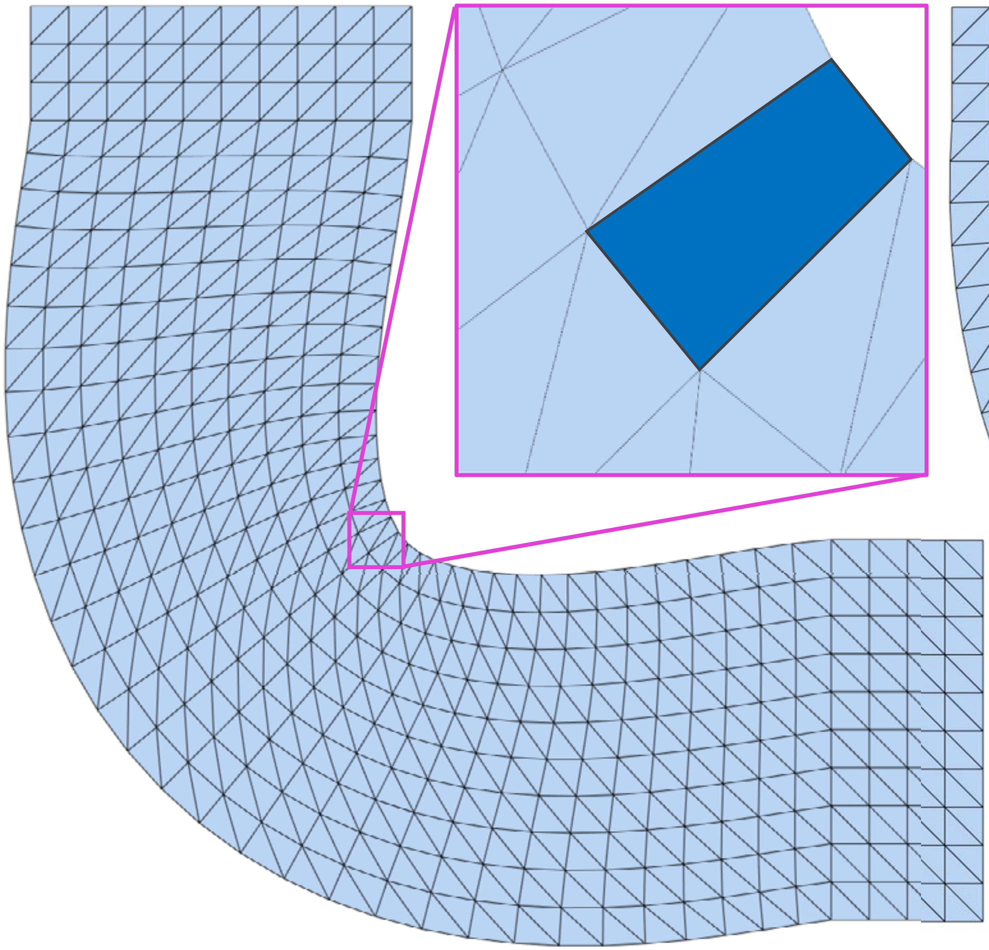
As-Rigid-As-Possible (ARAP)

$$E_R = \sum_j w_j \left\| A_j - \mathcal{R}(A_j) \right\|_F^2$$

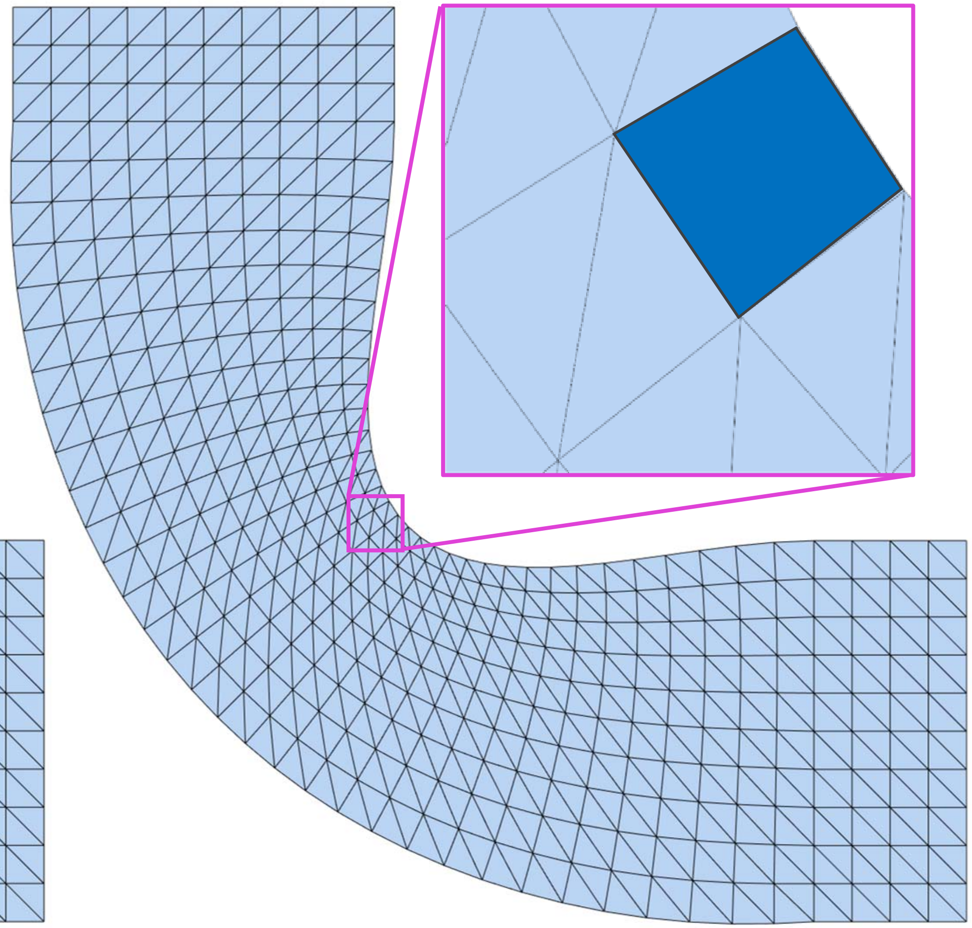
closest rigid transformation



ARAP vs. ASAP

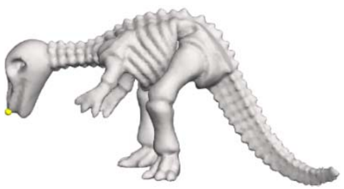


ARAP



ASAP

Singular values perspective

Dirichlet		$\lVert A \rVert_F^2$	$\sum_k \sigma_k^2$
LSCM		$\lVert A - \mathcal{S}(A) \rVert_F^2$	$\sum_k (\sigma_k - \overline{\sigma})^2$ mean of SVs
ARAP		$\lVert A - \mathcal{R}(A) \rVert_F^2$	$\sum_k (\sigma_k - 1)^2$

ARAP: Alternating Optimization

$$E_R = \sum_j w_j \left\| A_j - \mathcal{R}(A_j) \right\|_F^2$$

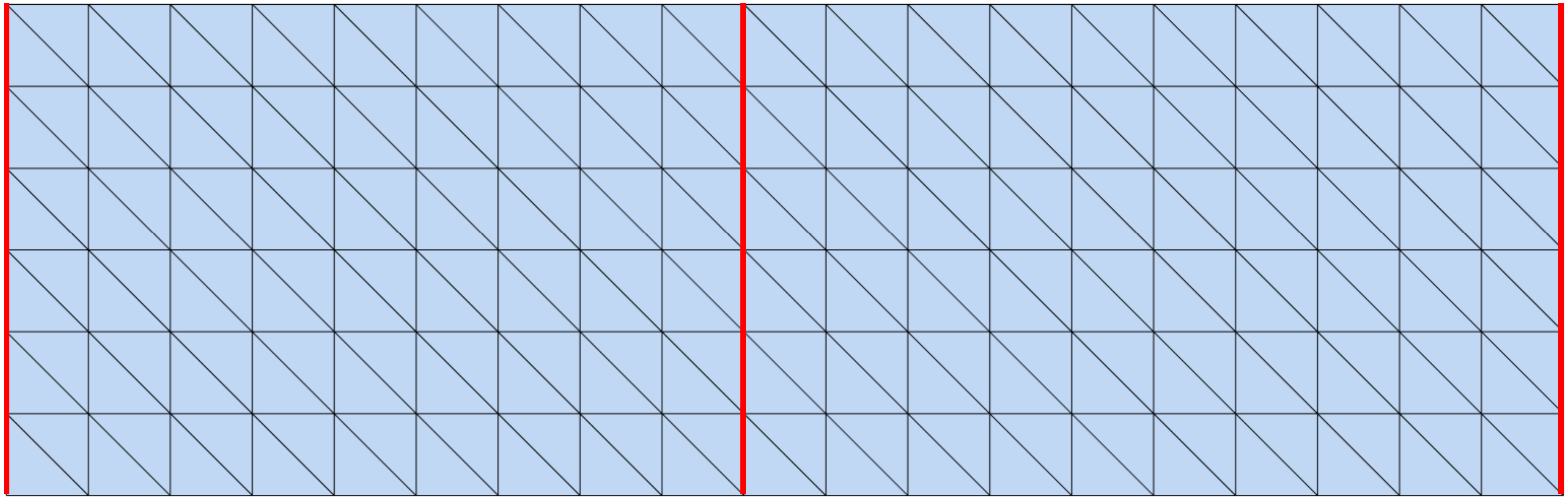
- Iteratively:
  - Compute and fix $R_j = \mathcal{R}(A_j)$
  - Minimize

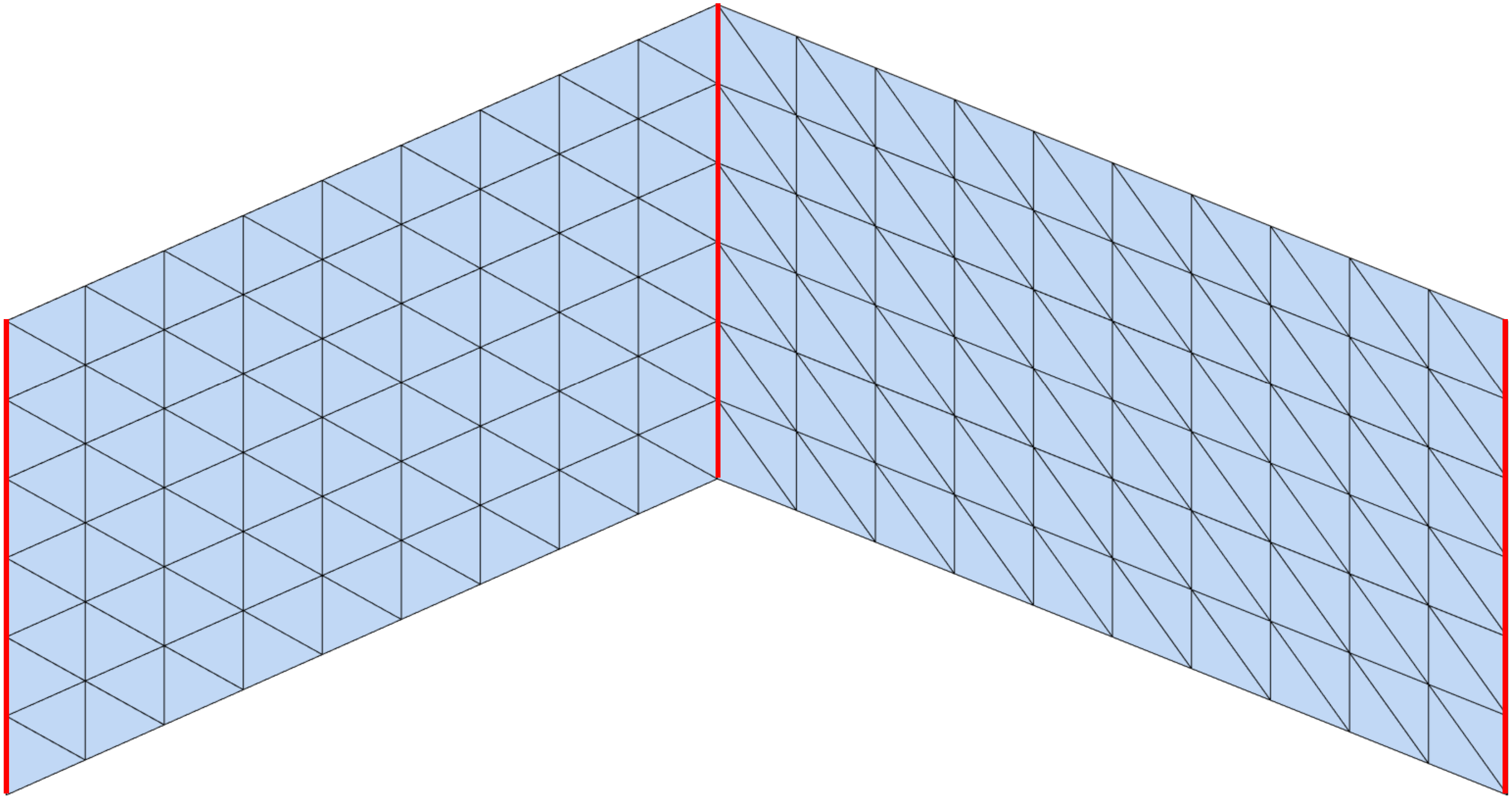
$$\sum_j w_j \left\| A_j - R_j \right\|_F^2$$

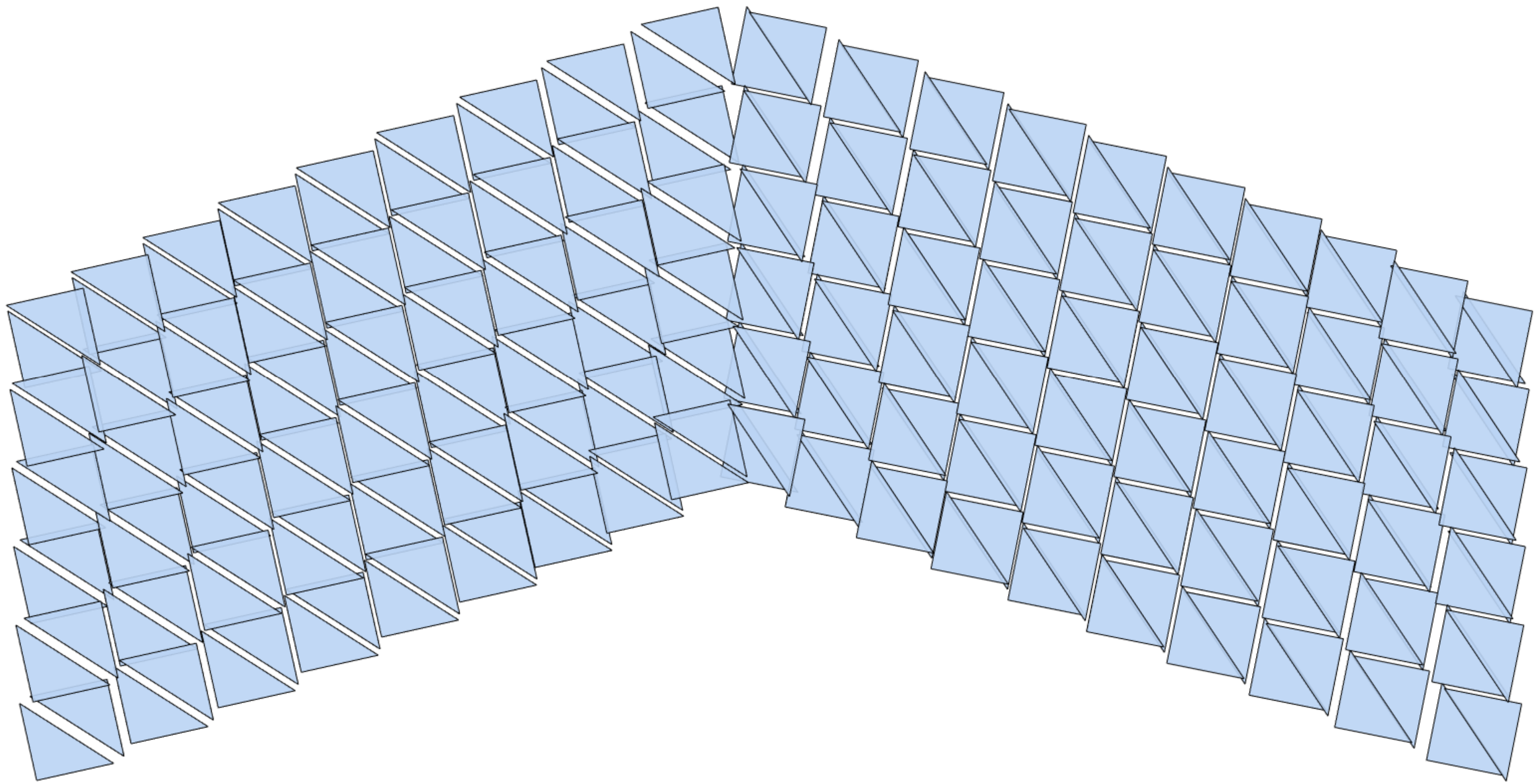
Local step

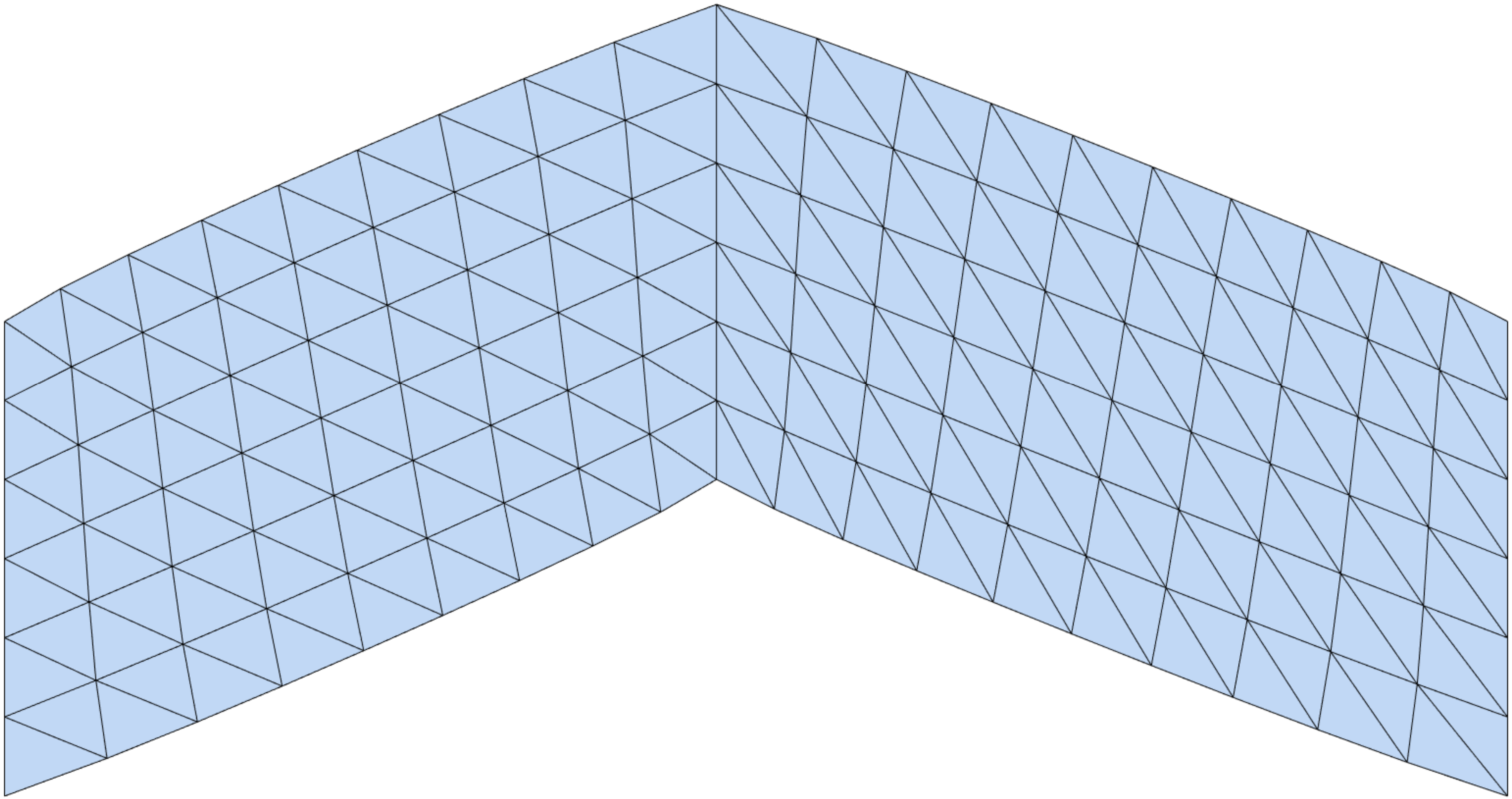
Global step

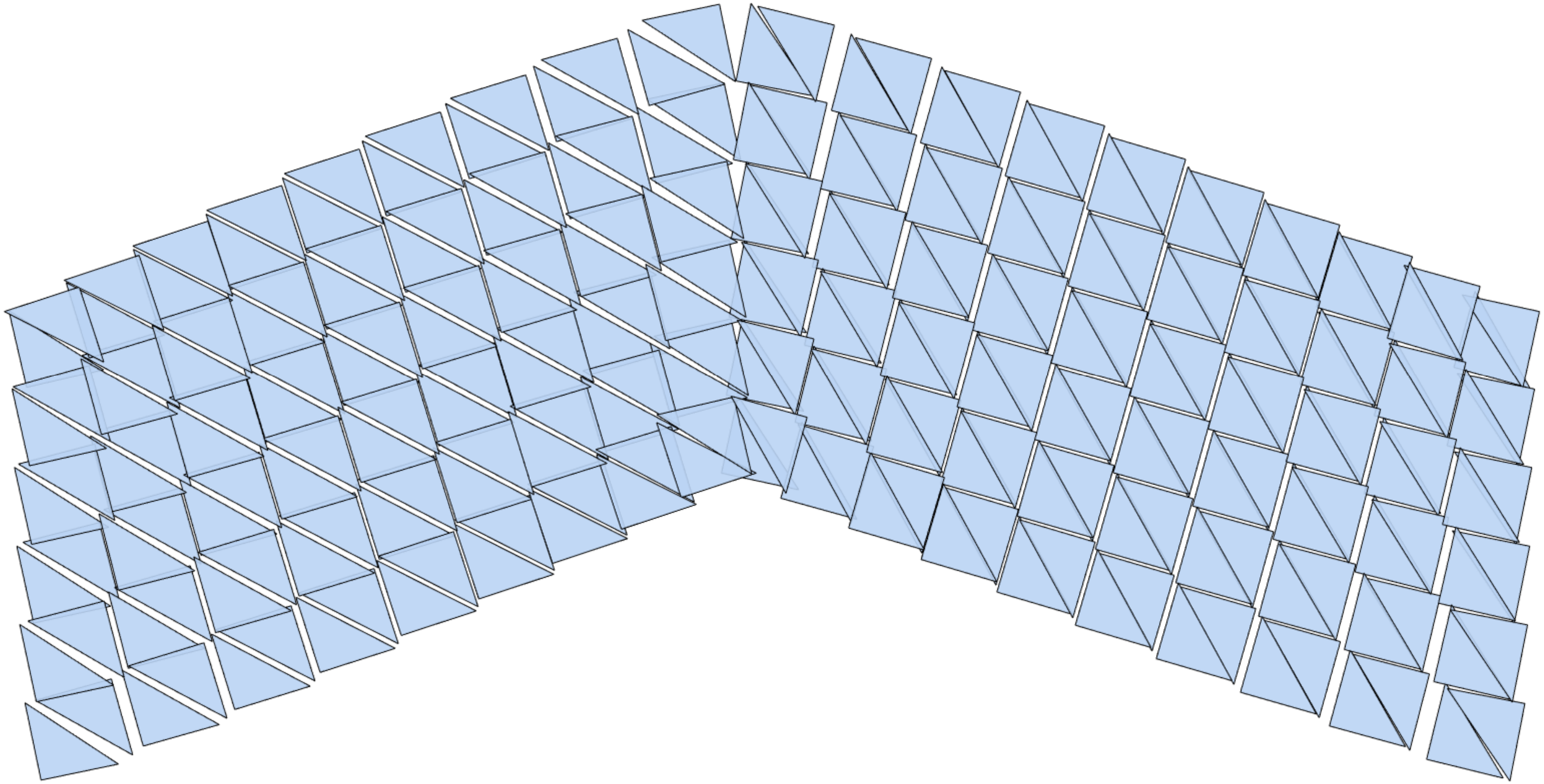
[Liu et al. A Local/Global Approach to Mesh Parameterization. SGP 2008]

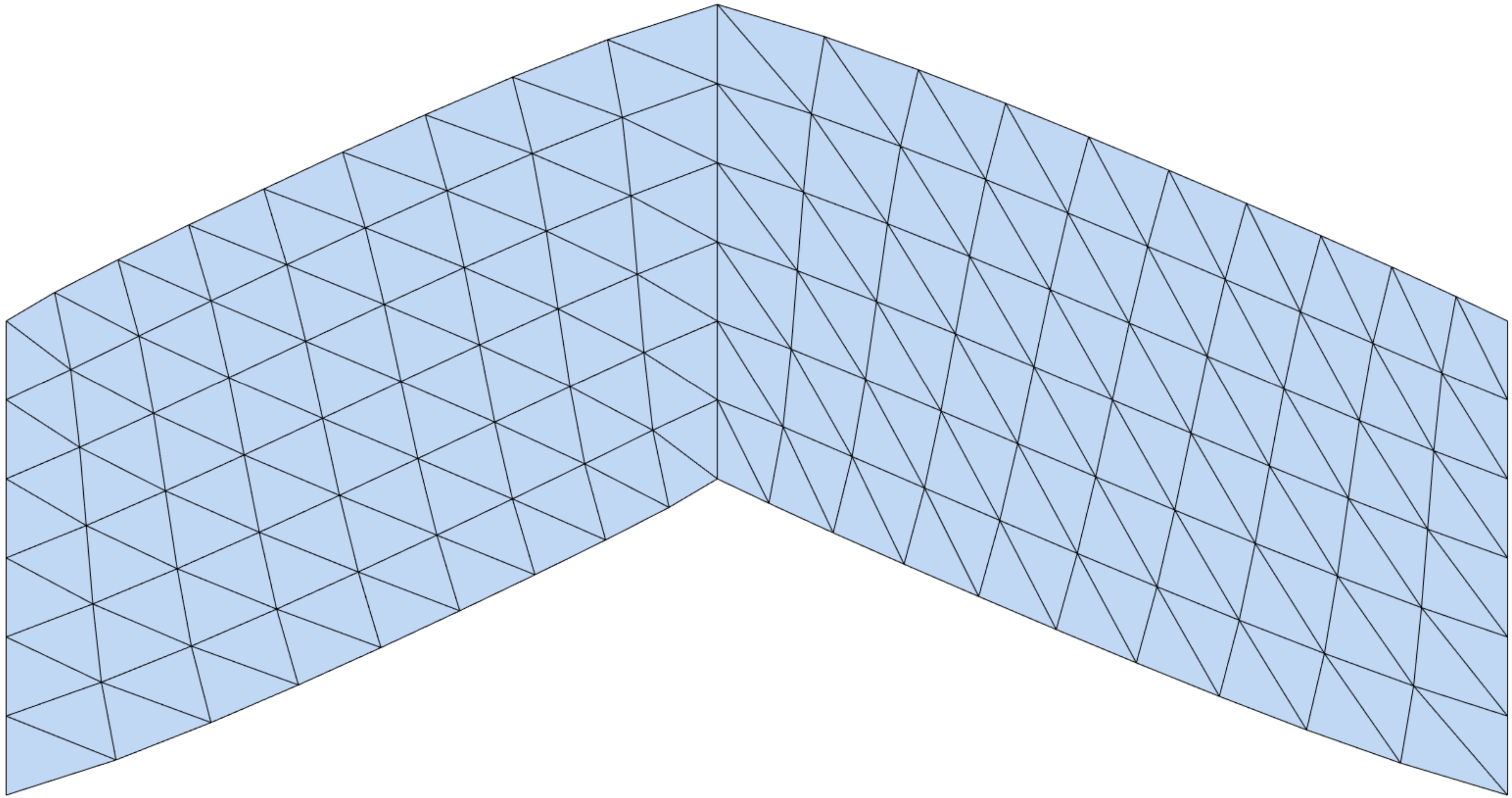


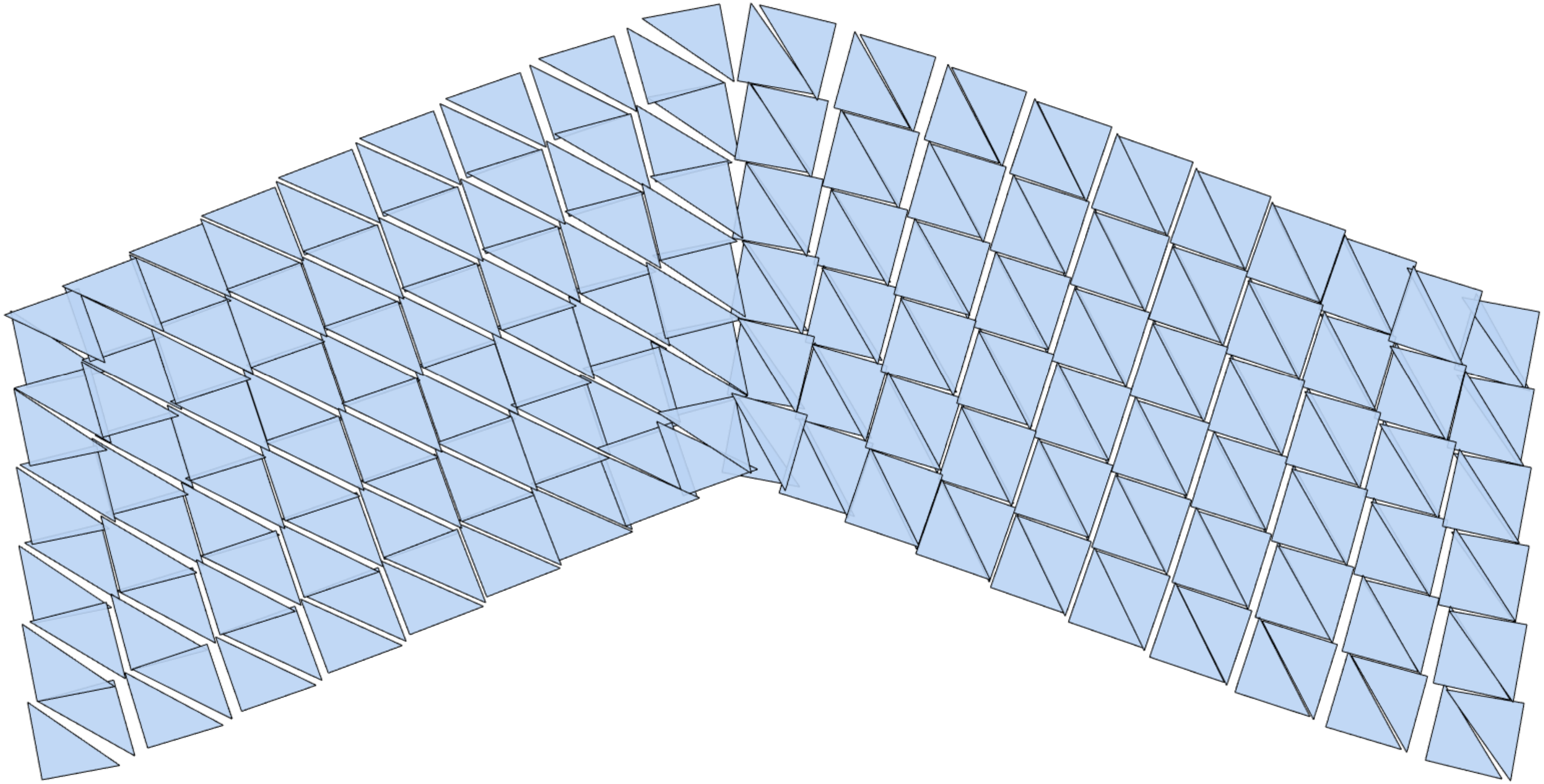


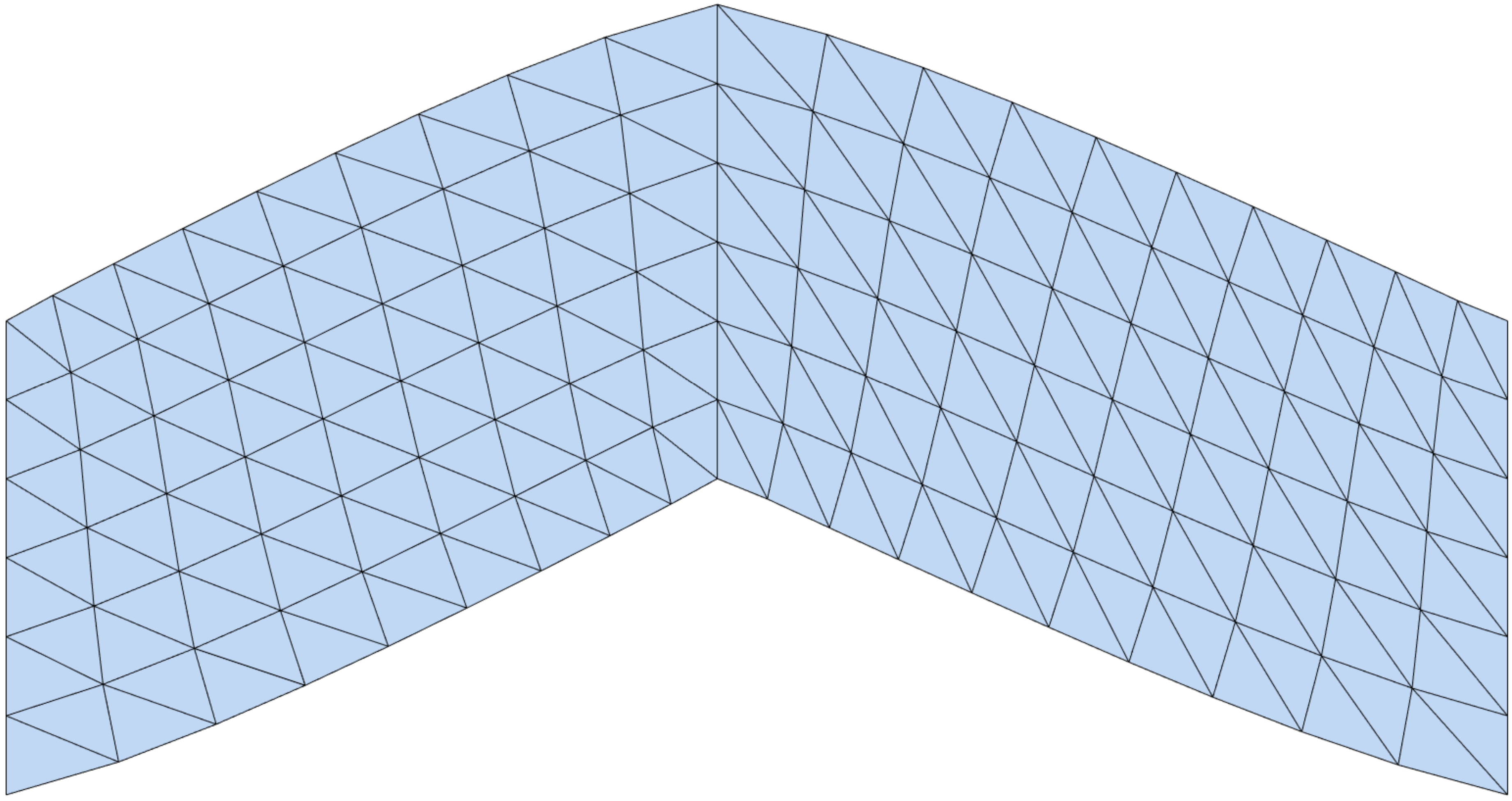


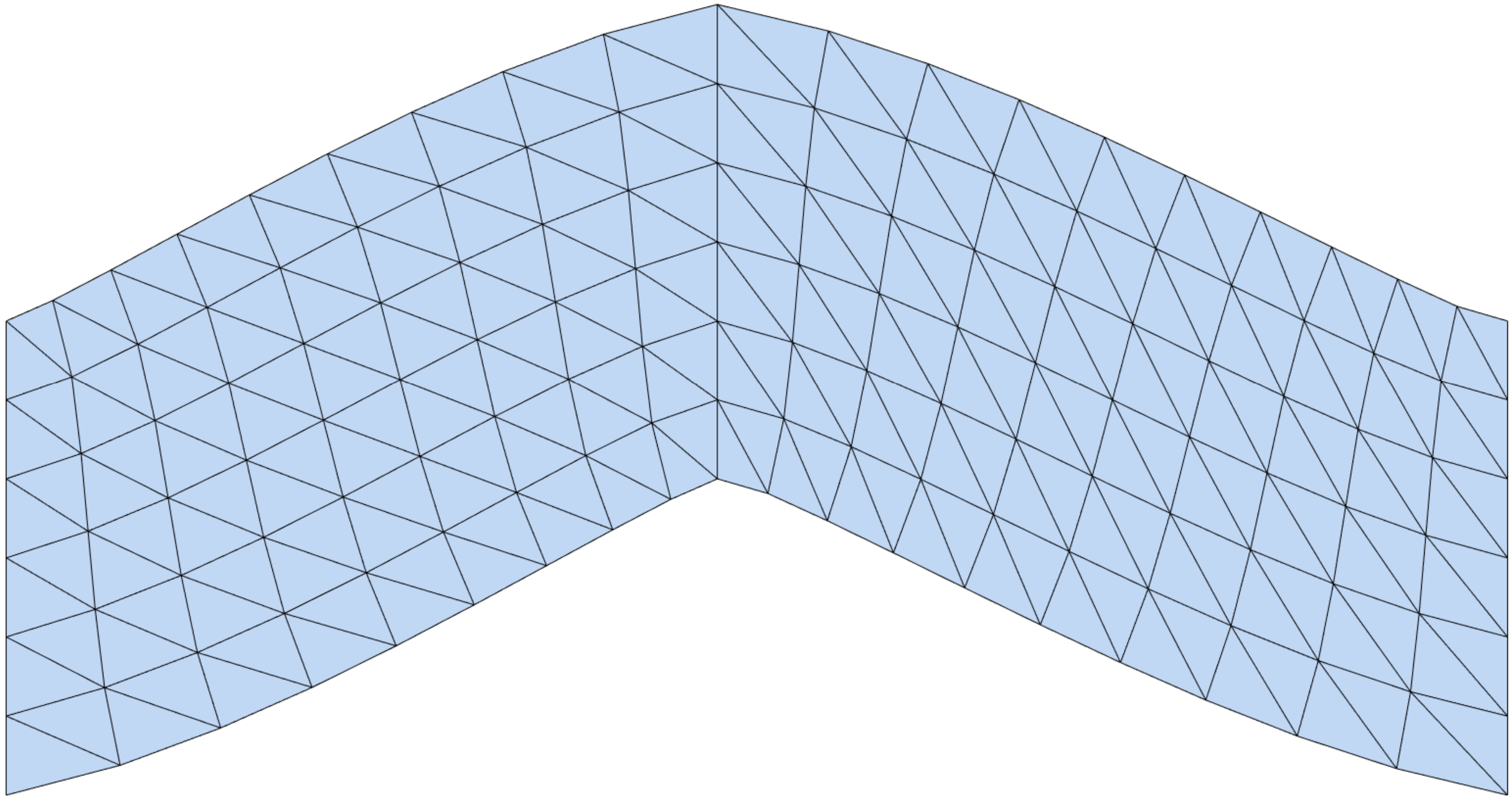


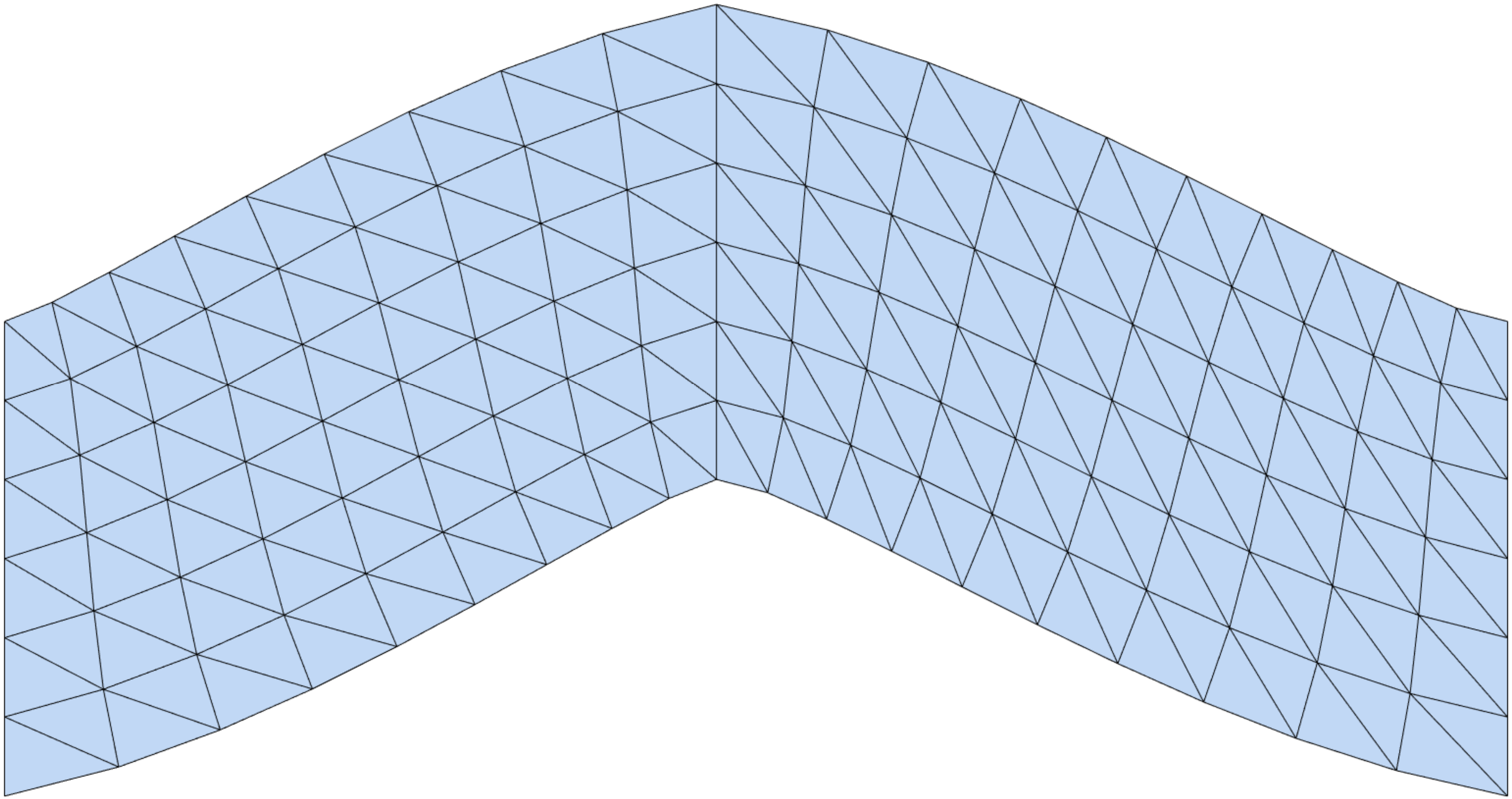


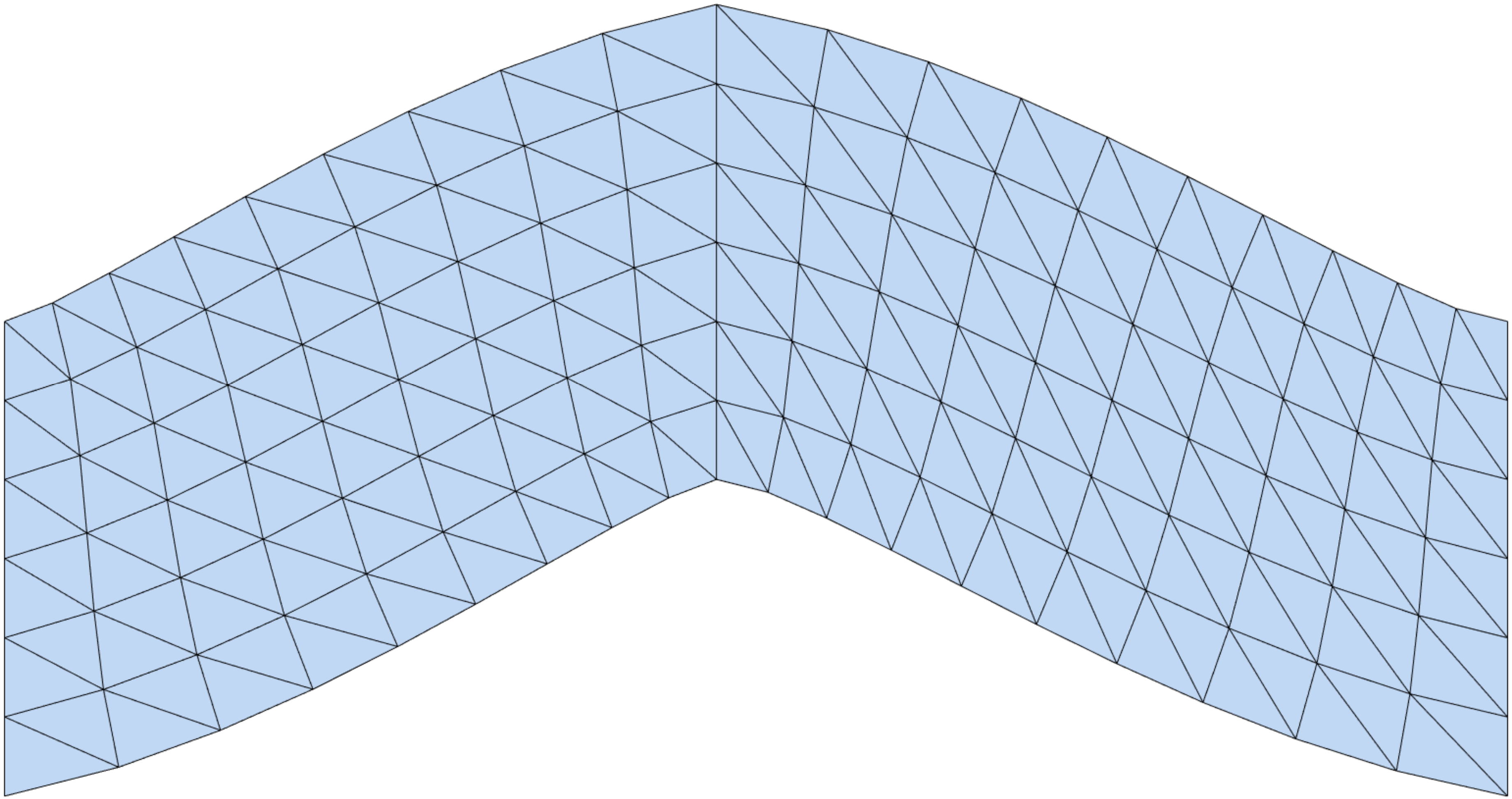


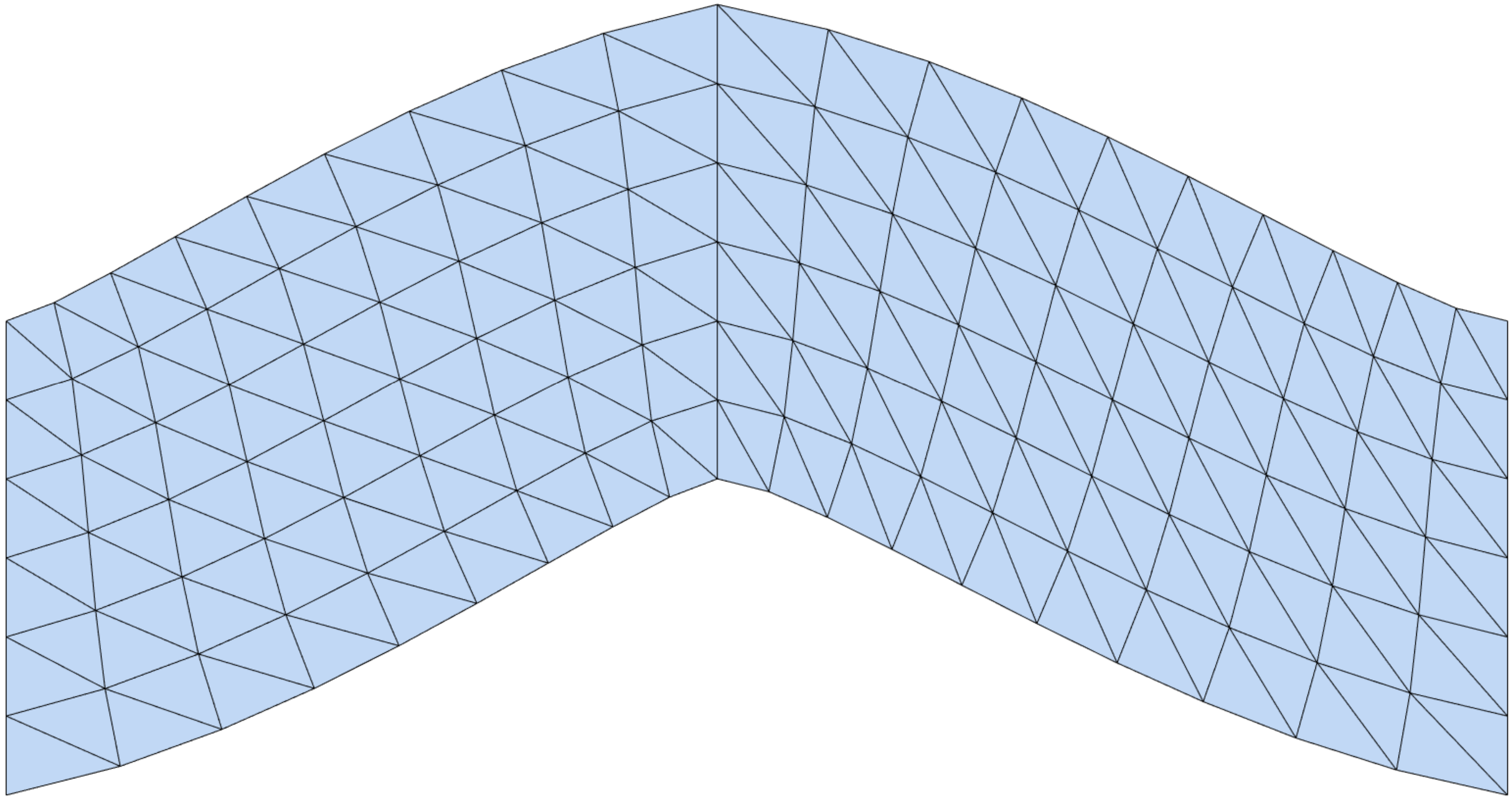






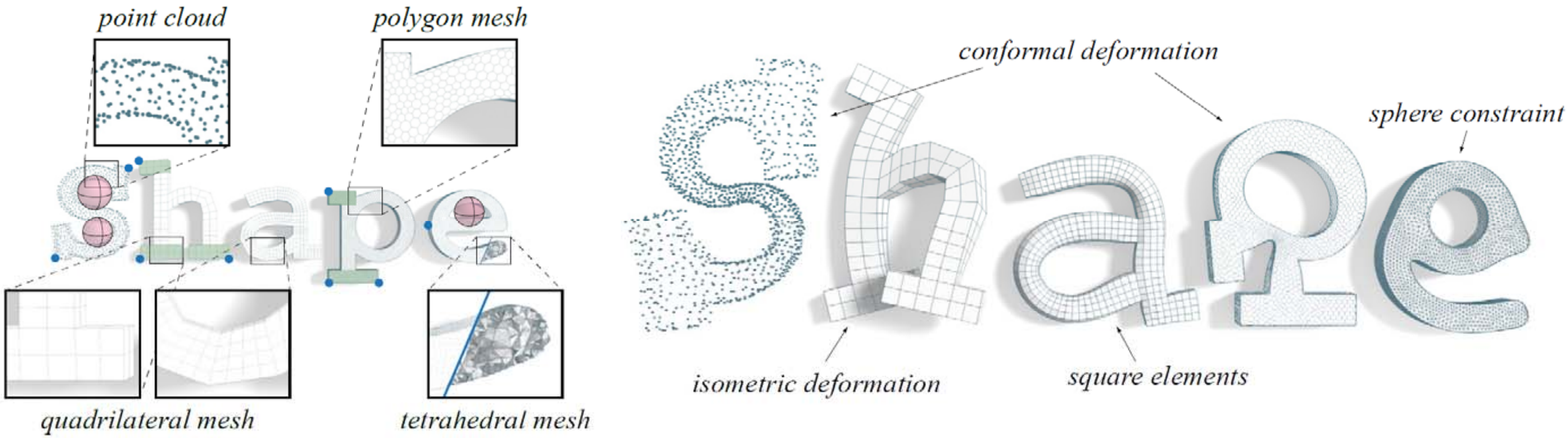






Alternating optimization

- Very general

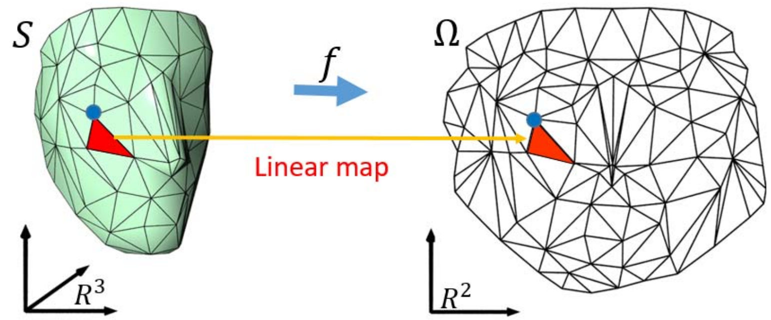


[Bouaziz et al. 2012]

- Related jargon:
  gradient descent, global-local, alternating projections

Summary: Geometric Mapping

- Discrete Mapping



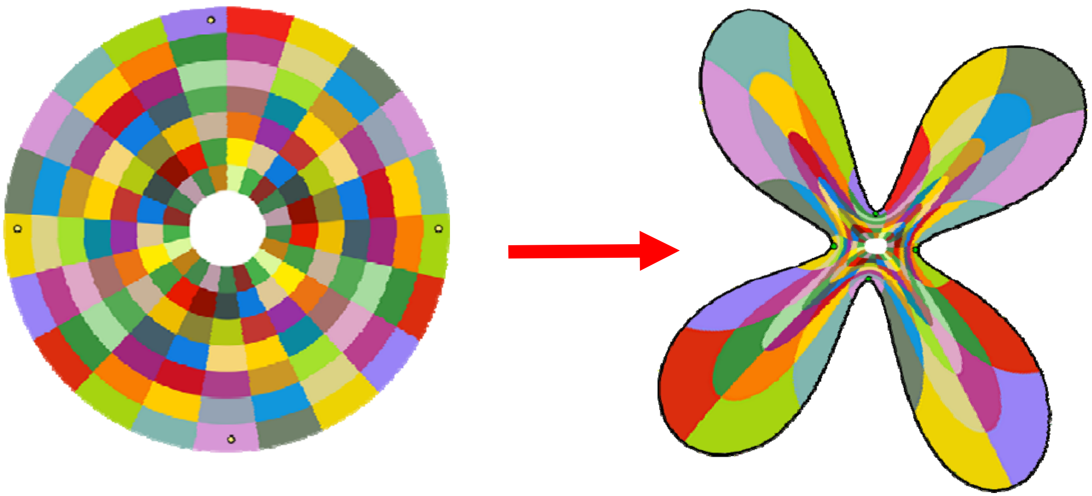
- Discrete formulation

$$\underset{\Phi}{\operatorname{argmin}} \quad E(\Phi)$$
$$\text{s.t.} \quad \Phi \in K$$

Separable

- Nonlinear and nonconvex
- Computationally expensive for large scale meshes!

Meshless mappings

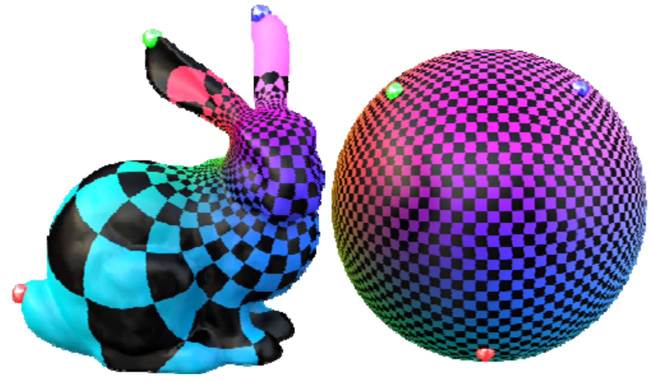


$$f(\boldsymbol{x}) = \sum_{i=1}^{m} \boldsymbol{c}_i B_i(\boldsymbol{x})$$

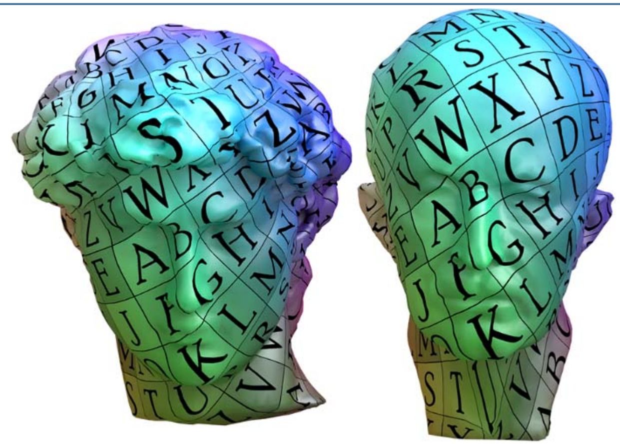
- Low distortion
- Flip-free
- Bijective

Geometric Mapping

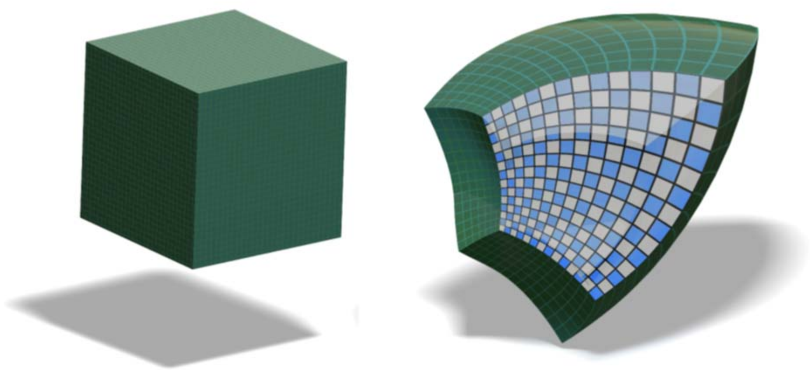
- 其他区域间的映射求解
    - 离散形式
    - 约束条件



$f: \mathbb{M} \to \mathbb{S}^2$



$f: \mathbb{M} \to \mathbb{M}'$



$f: \mathbb{R}^3 \to \mathbb{R}^3$

Fu et al. Inversion-free Geometric Mapping Construction: A Survey. CVM, 2021

中国科学技术大学
University of Science and Technology of China

谢　谢　！