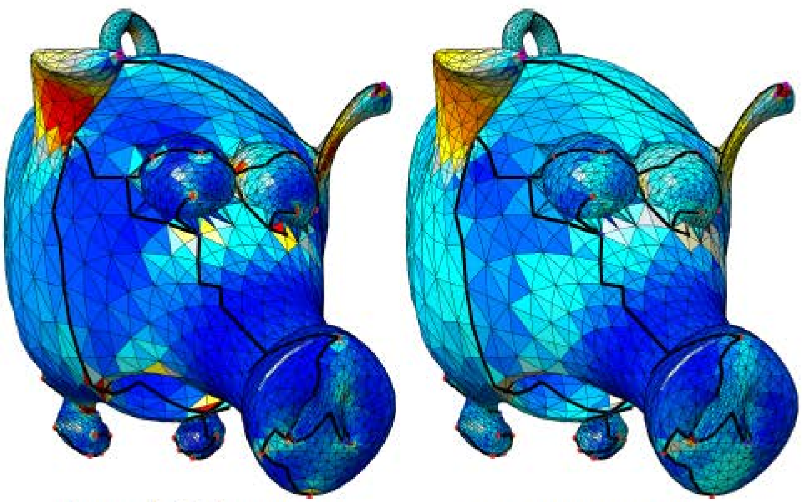
Strict Minimizers For Geometric Optimization

Zohar Levi
New York University

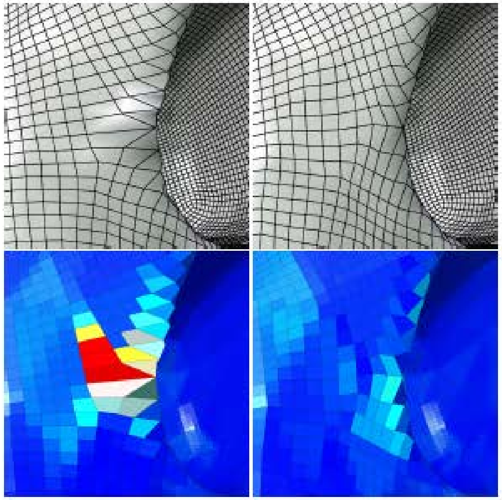
Denis Zorin
New York University

Motivation

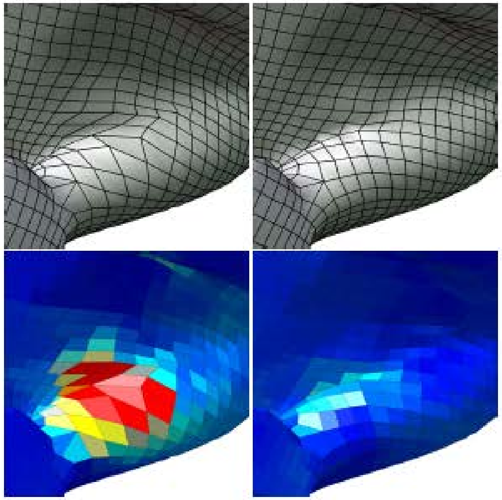
[Aigerman and Lipman 2013] vs. our strict minimizer



bounded distortion *strict minimizer*

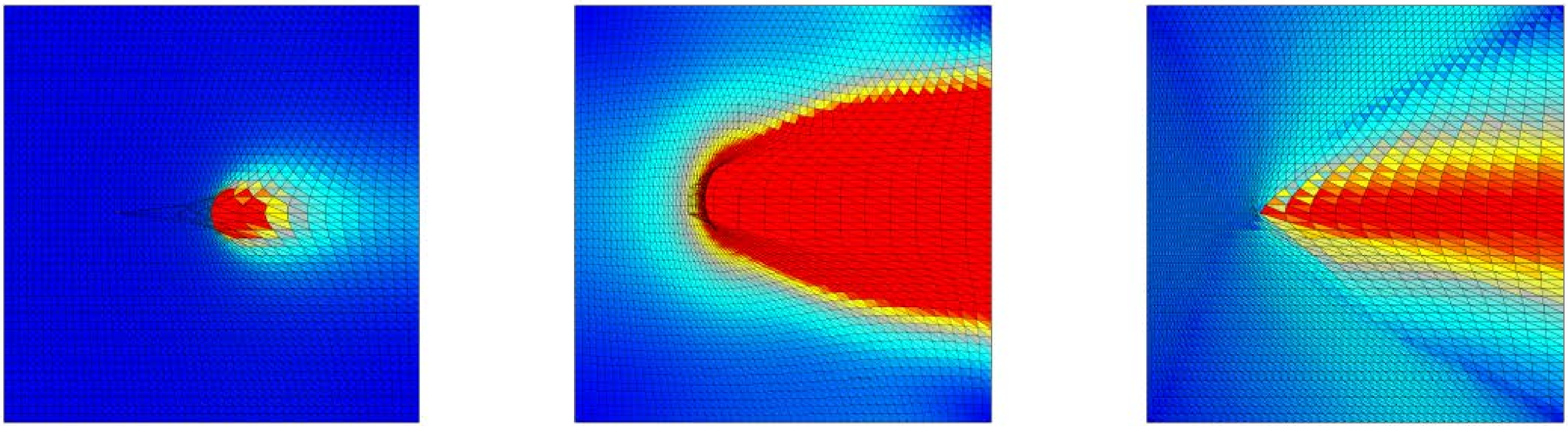


bounded distortion *strict minimizer*



bounded distortion *strict minimizer*

A toy example



L_2 L_∞ $strict$

Overall idea

1. Find L_∞ solution with smallest max distortion set of facets
2. Fix the distortion on facets with max distortion
3. Repeat until all facets are fixed

Questions

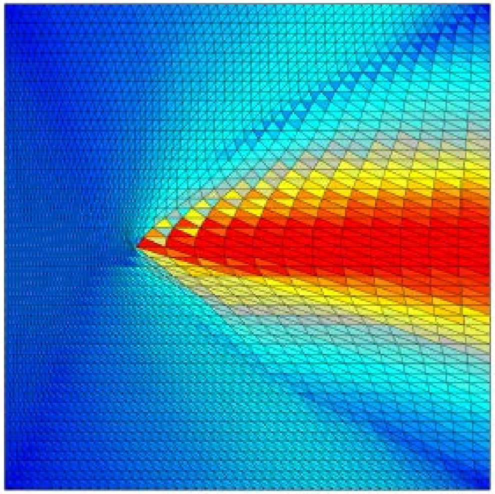
- Can we define "the best" L_∞ solution independent of the algorithm?
- Is it unique?
- Is there an efficient algorithm?

Questions

- Can we define “the best” L_∞ solution independent of the algorithm?
- Is it unique?
- Is there an efficient algorithm?

Lexicographical ordering

Vector of facets' distortion for a mapping $\boldsymbol{f}$



$$D[\mathbf{f}] = [D_1[\mathbf{f}], \ldots D_N[\mathbf{f}]]$$

$D_i(f)$ is the distortion on facet i.

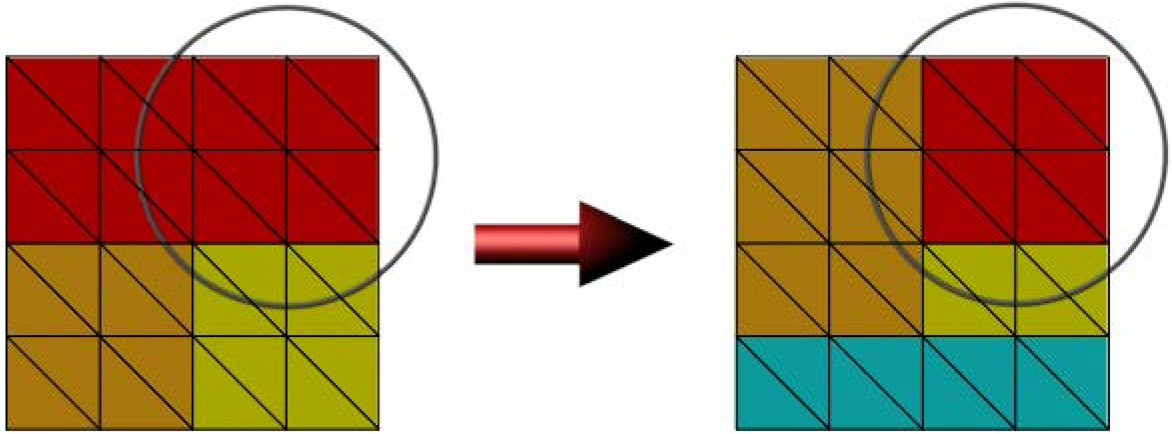
Define an ordering based on this (sorted) vector

(For simplicity, we consider a mesh with similar facets.)

(Lexicographical is a dictionary order: “3322” < “3331”)

How to compute this in theory

1. Solve L_∞
2. Figure out minimal subset of facets that must have max distortion
3. Fix the distortion on facets with max distortion
4. Repeat until all facets are fixed

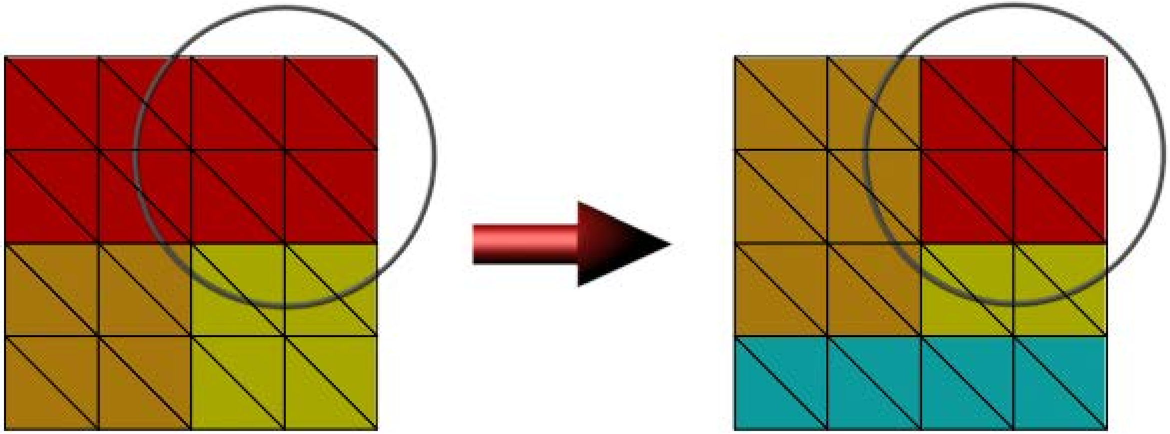


Minimize the distortion of free facets

Consider the set of solutions X:

$$k = \min \max_{j \notin I} D_j[\mathbf{f}], \ \text{subject to } D_i[\mathbf{f}] \leq k_i, i \in I$$

I is the set of facets already constrained to have an upper bound k_i on the distortion.



Questions

- Can we define “the best” L_∞ solution independent of the algorithm?
- Is it unique?
- Is there an efficient algorithm?

Essential facets

Facet j is essential if for all L_∞ solutions in X.

$$D_j[\mathbf{f}] = k$$

Proposition 1. The essential set is not empty.

The essential set is unique by definition.

Theoretical algorithm

1. Solve L_∞
2. Find the essential set
3. Add essential set to the fixed set
4. Repeat until all facets are fixed

Uniqueness

Proposition 2. Any strict minimizer has the same distortion vector

$$D[\mathbf{f}^*] = [k_1, \ldots, k_N]$$

Uniqueness of solutions vs. vectors of distortion

Questions

- Can we define “the best” L_∞ solution independent of the algorithm?
- Is it unique?
- Is there an efficient algorithm?

L_∞ optimization

$$\min k, \qquad \text{subject to}$$

$$D_i^{FD}(\mathbf{q}) = \|J_i(\mathbf{q}) - R_i\|_F \leq k, \qquad i = 1 \ldots N$$

- Isometric or quasiconformal (ASAP)
- A SOCP problem, which can be solve efficiently
- Always feasible
- Can operate inside a local/global framework
- Distortion below 1 means no foldovers

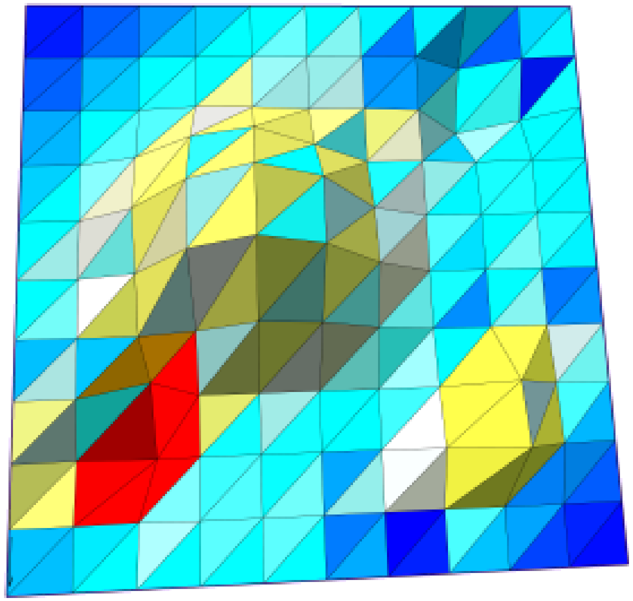
Theoretical algorithm

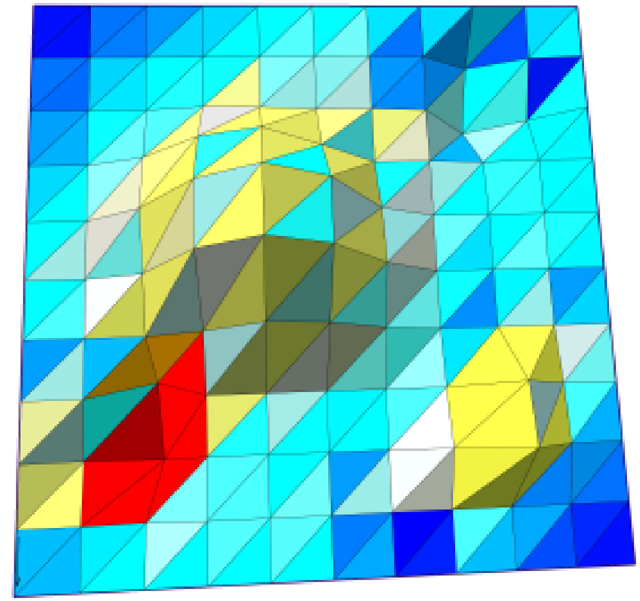
1. Solve L_∞
2. Find the essential set
3. Add essential set to the fixed set
4. Repeat until all facets are fixed

First approximate algorithm

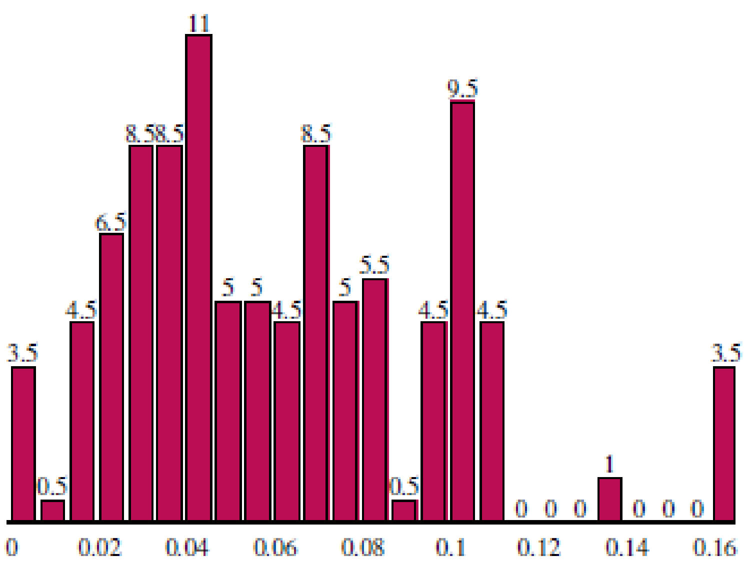
1. Solve L_∞ for upper bound k on free facets
2. Solve L_2, temporarily constraining free facets to k
3. Estimate essential set as facets with distortion k-Δ
4. Add essential set to the fixed set
5. Repeat until all facets are fixed

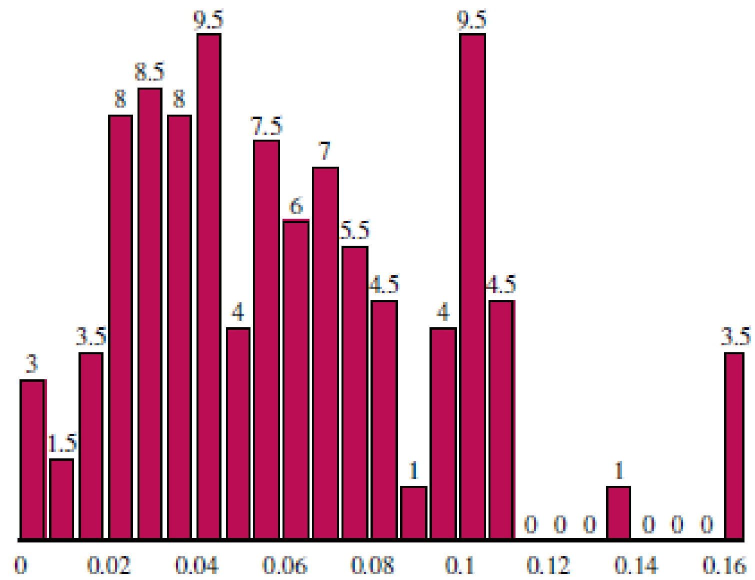
Approximate





Exact



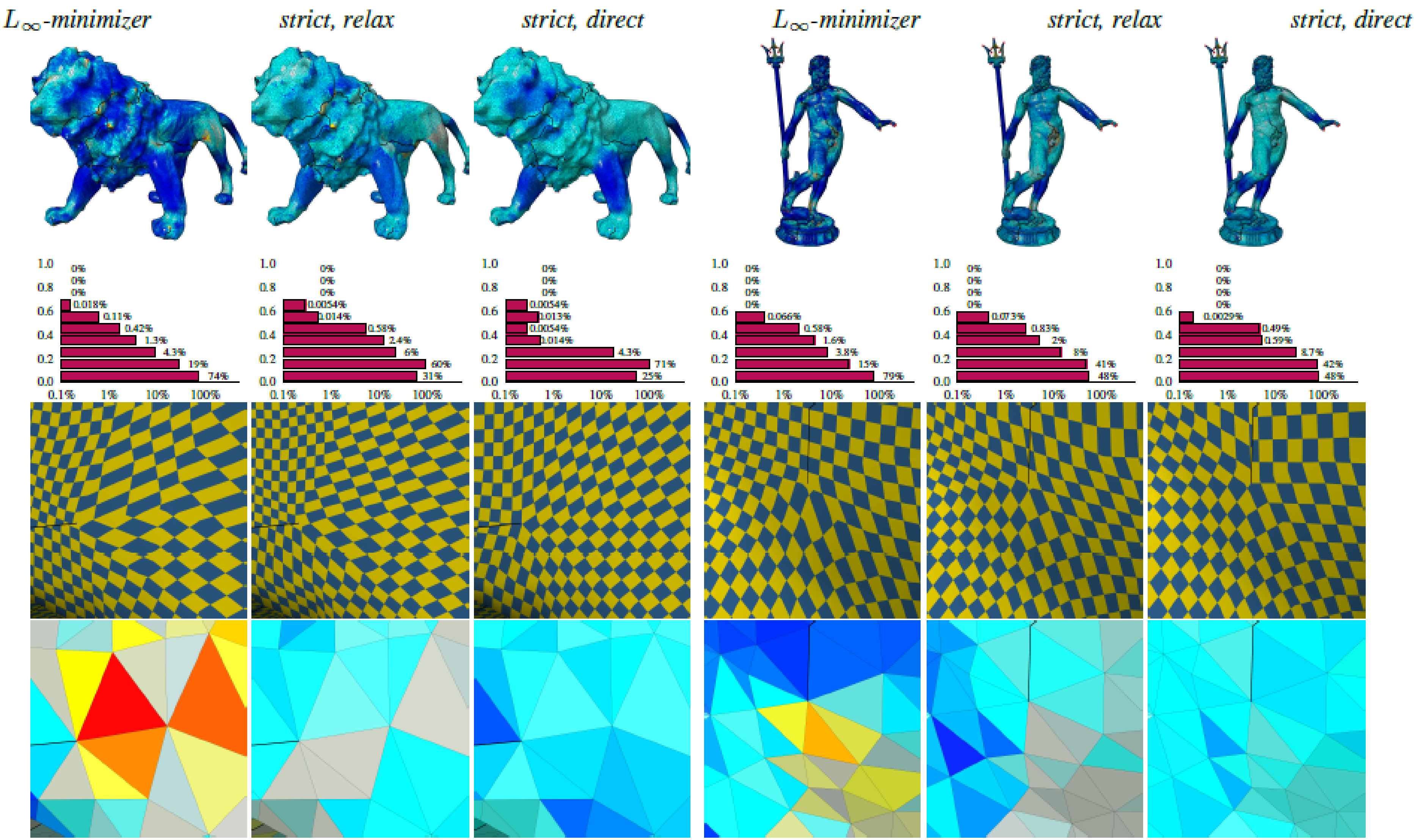


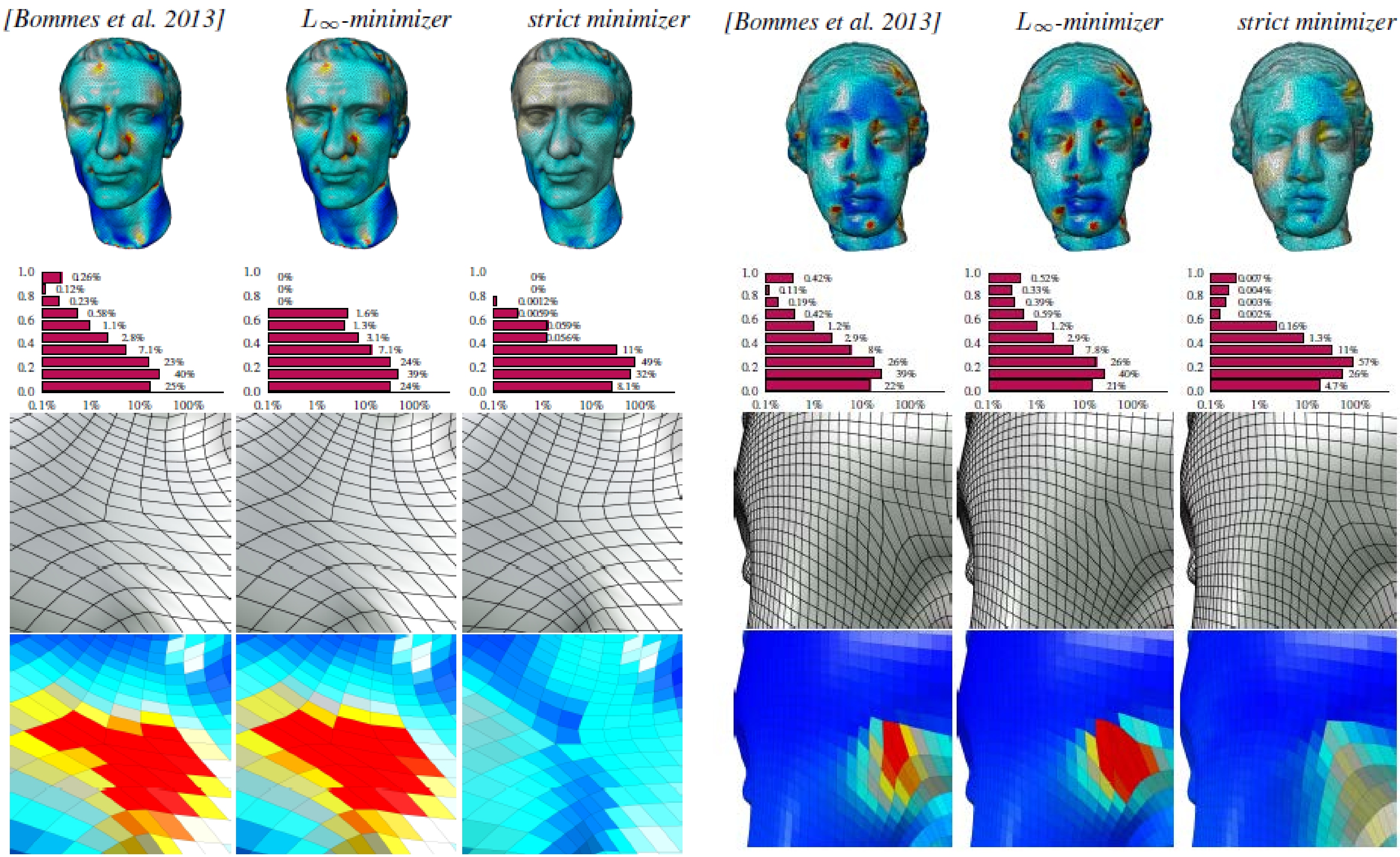
Second approximate algorithm

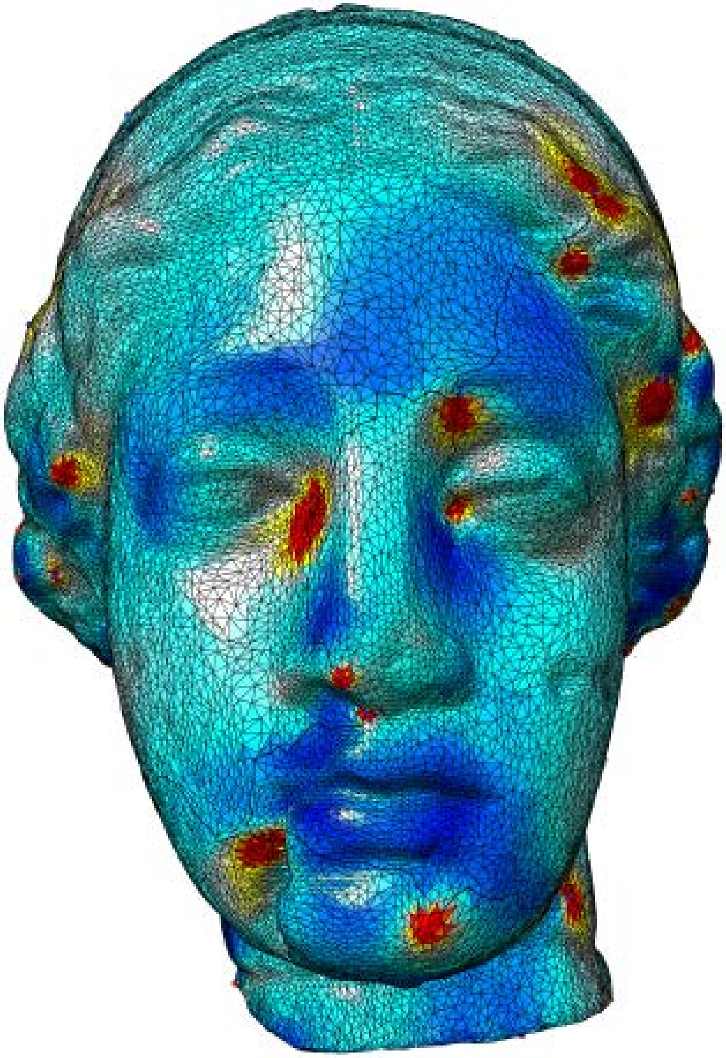
- Solve L_∞ to obtain an initial solution.
- **repeat** 50 times
  - **for** each vertex v
    - Solve 1-ring L_∞ problem on v

Properties

- Coordinate descent
- One vertex at a time
- Problem can be solved semi-explicitly


L_∞-minimizer
strict, relax
strict, direct
L_∞-minimizer
strict, relax
strict, direct
1.0
0.8
0.6
0.4
0.2
0.0
0.1%
1%
10%
100%
0%
0.018%
0.11%
0.42%
1.3%
4.3%
19%
74%
0.0054%
0.014%
0.58%
2.4%
6%
60%
31%
0.013%
0.014%
4.3%
71%
25%
0.066%
0.58%
1.6%
3.8%
15%
79%
0.073%
0.83%
2%
8%
41%
48%
0.0029%
0.49%
0.59%
8.7%
42%

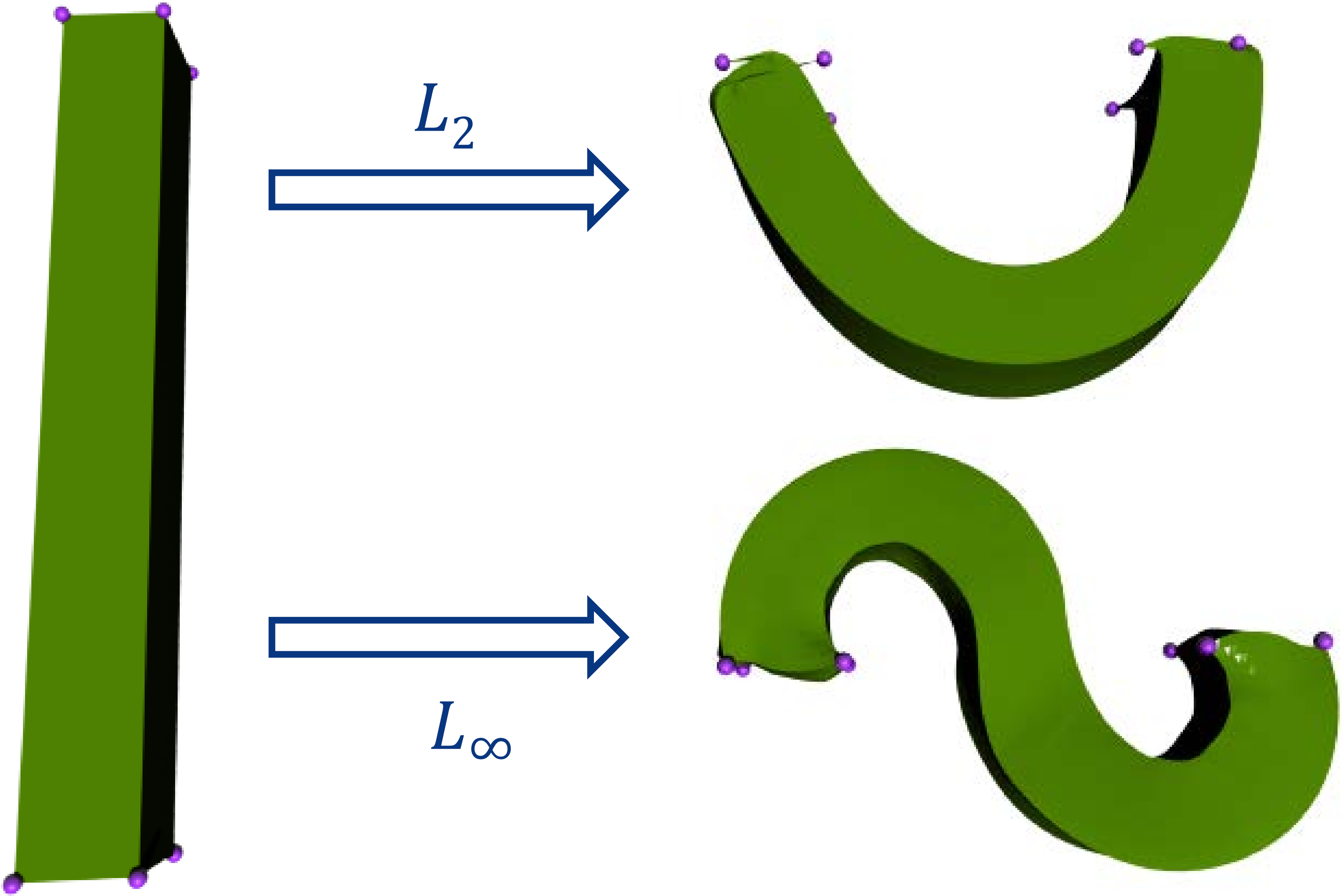

[Bommes et al. 2013]
L_∞-minimizer
strict minimizer
[Bommes et al. 2013]
L_∞-minimizer
strict minimizer
0.26% 0.12% 0.23% 0.58% 1.1% 2.8% 7.1% 23% 40% 25%
0% 0% 0% 1.6% 1.3% 3.1% 7.1% 24% 39% 24%
0% 0% 0.0012% 0.0059% 0.059% 0.056% 11% 49% 32% 8.1%
0.42% 0.11% 0.19% 0.42% 1.2% 2.9% 8% 26% 39% 22%
0.52% 0.33% 0.39% 0.59% 1.2% 2.9% 7.8% 26% 40% 21%
0.007% 0.004% 0.003% 0.002% 0.16% 1.3% 11% 57% 26% 4.7%
1.0 0.8 0.6 0.4 0.2 0.0
0.1% 1% 10% 100%



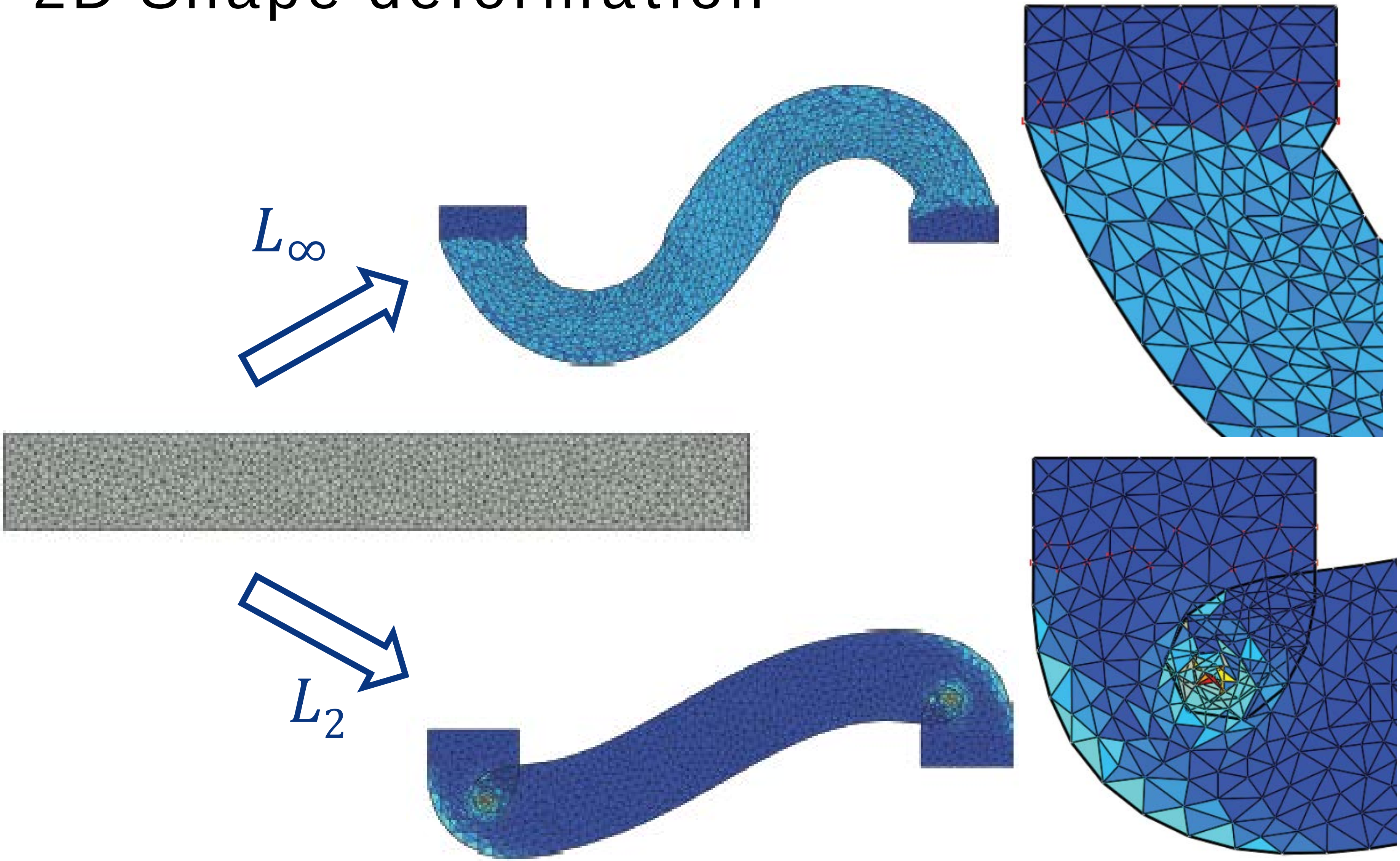
L_∞

strict min

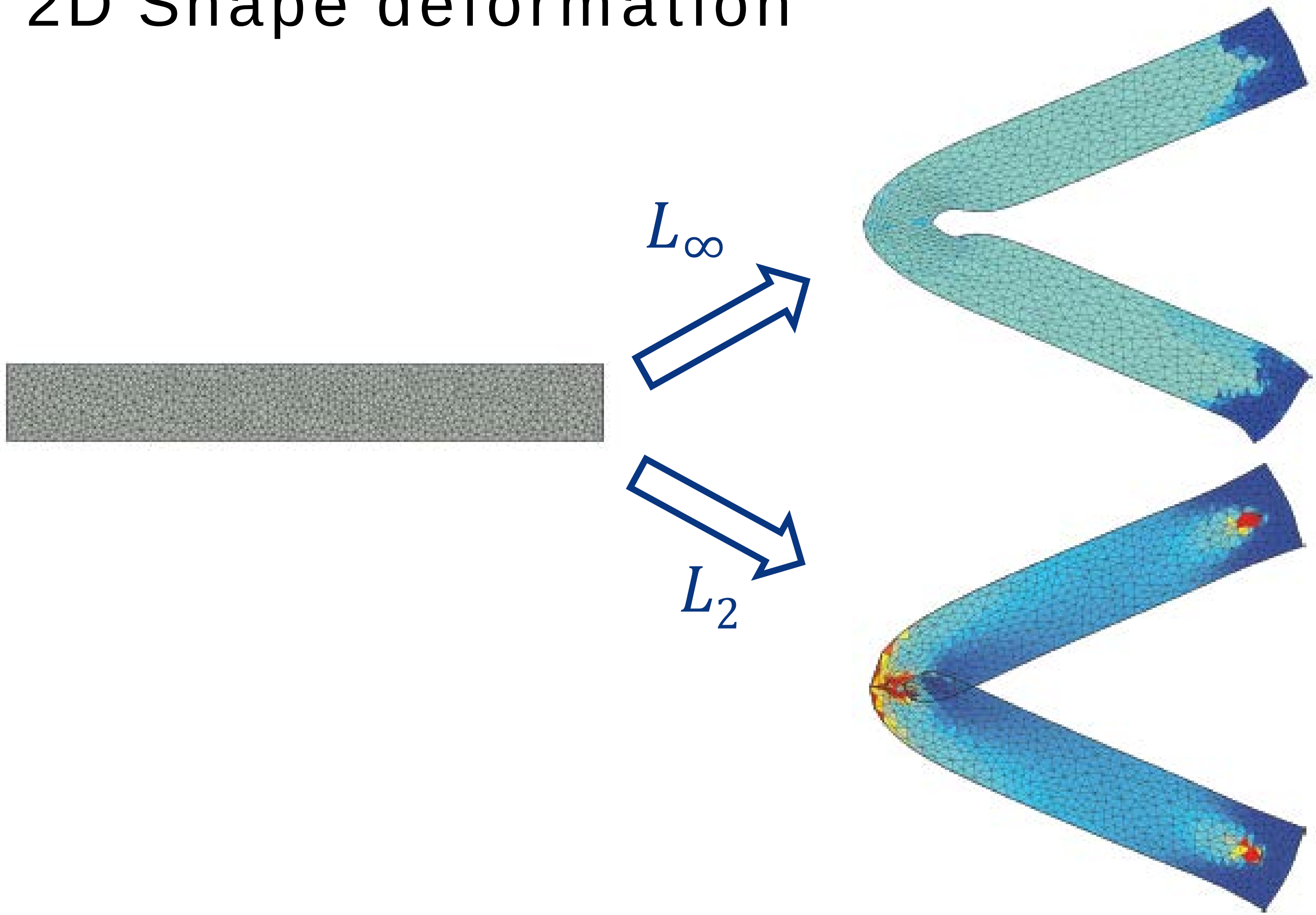
3D Shape deformation



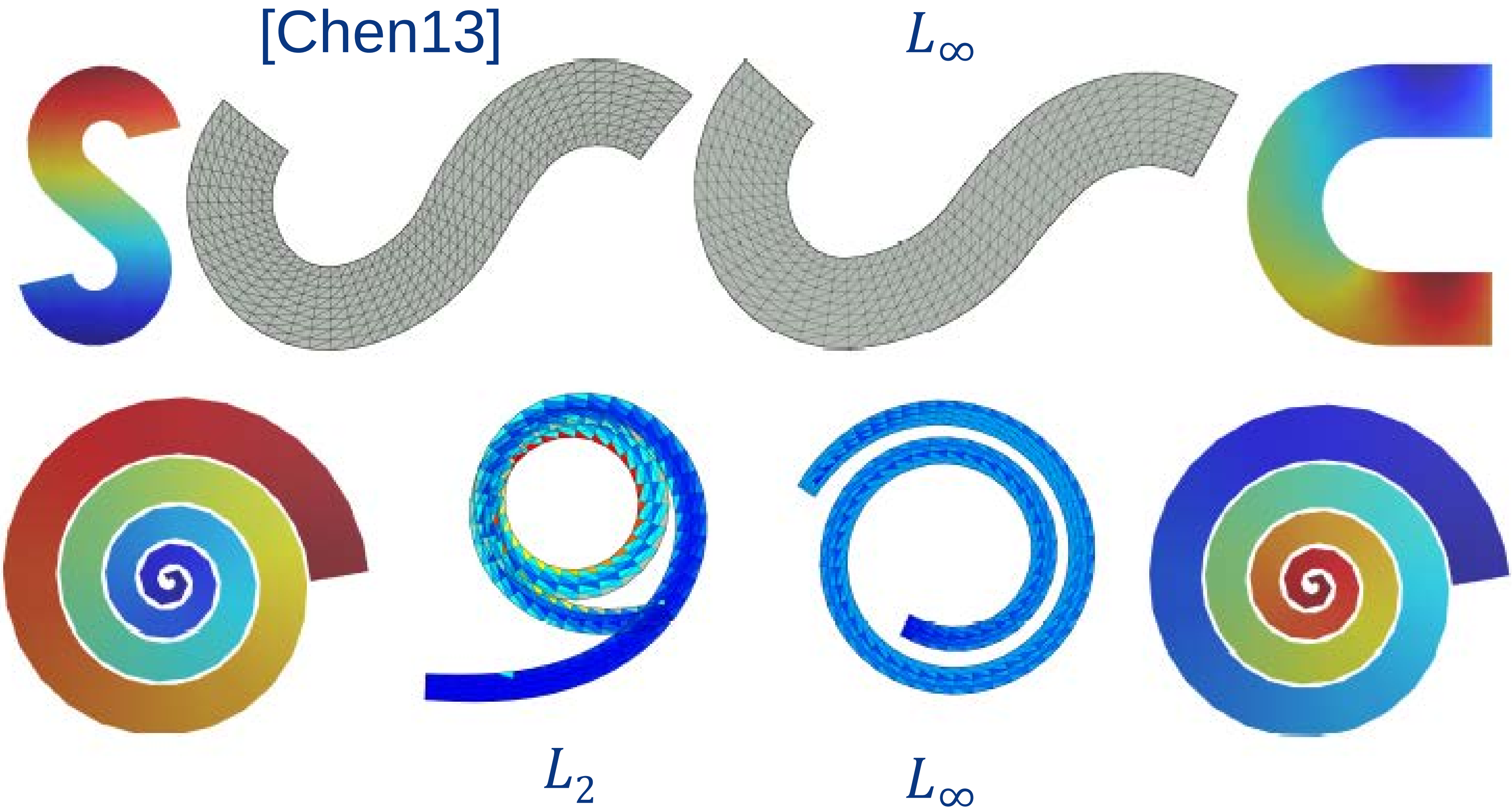
2D Shape deformation



2D Shape deformation

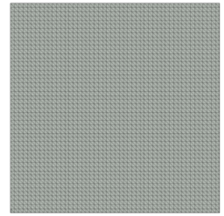


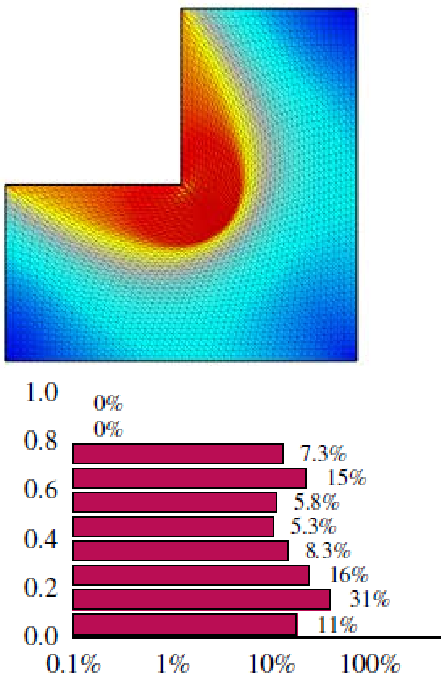
Shape interpolation



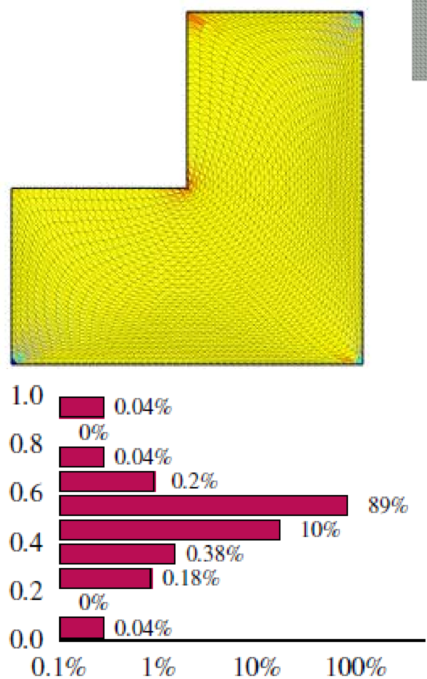
Teichmuller

Source

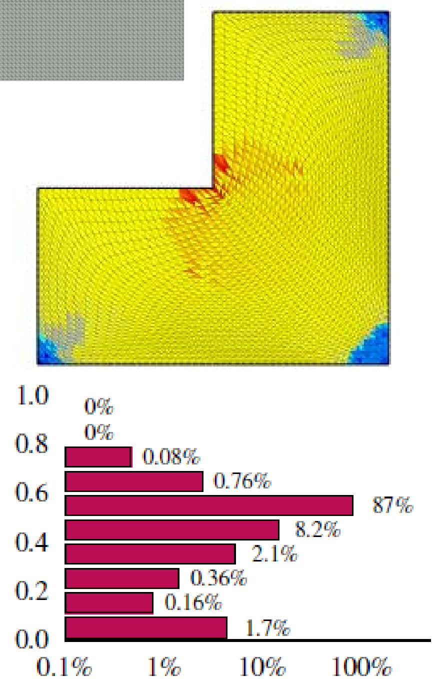




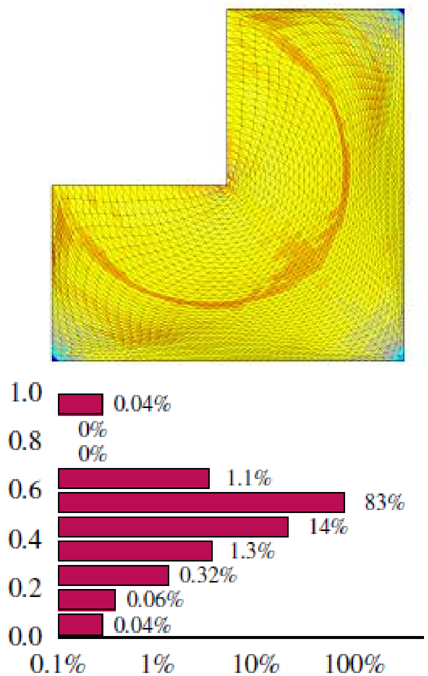
[Lipman12]



[Weber12]



Strict min [Lipman12]



Strict min ASAP

- Conclusions
  - Defined strict minimizer – ‘best’ L_∞ solution
  - Uniqueness
  - Efficient approximate algorithms

- Acknowledgments

  - Julian Panetta
  - NSF awards IIS-1320635 and IIS-1247240