Black-Box Analysis:

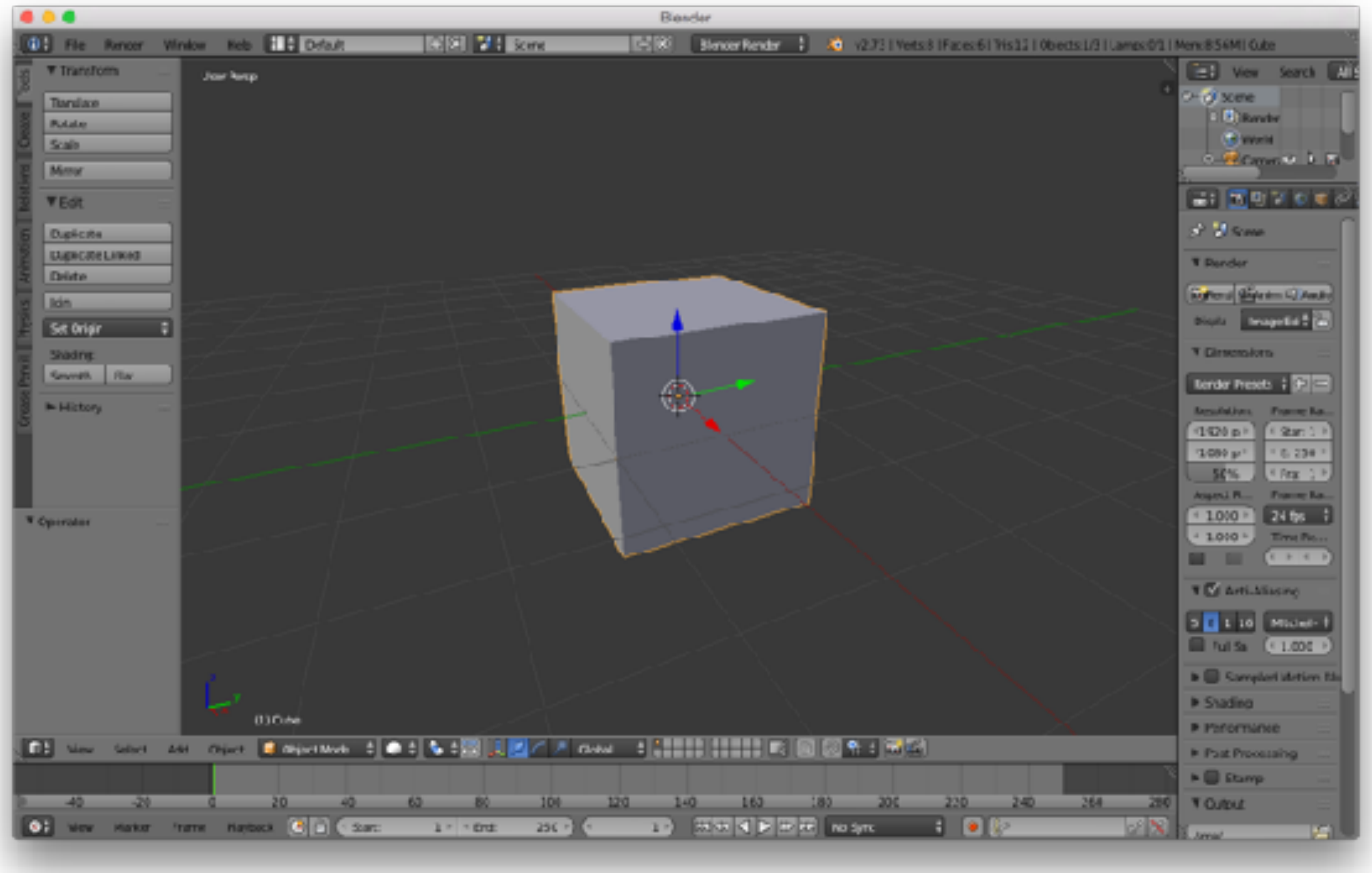
From **Theory** to Practice

Teseo Schneider

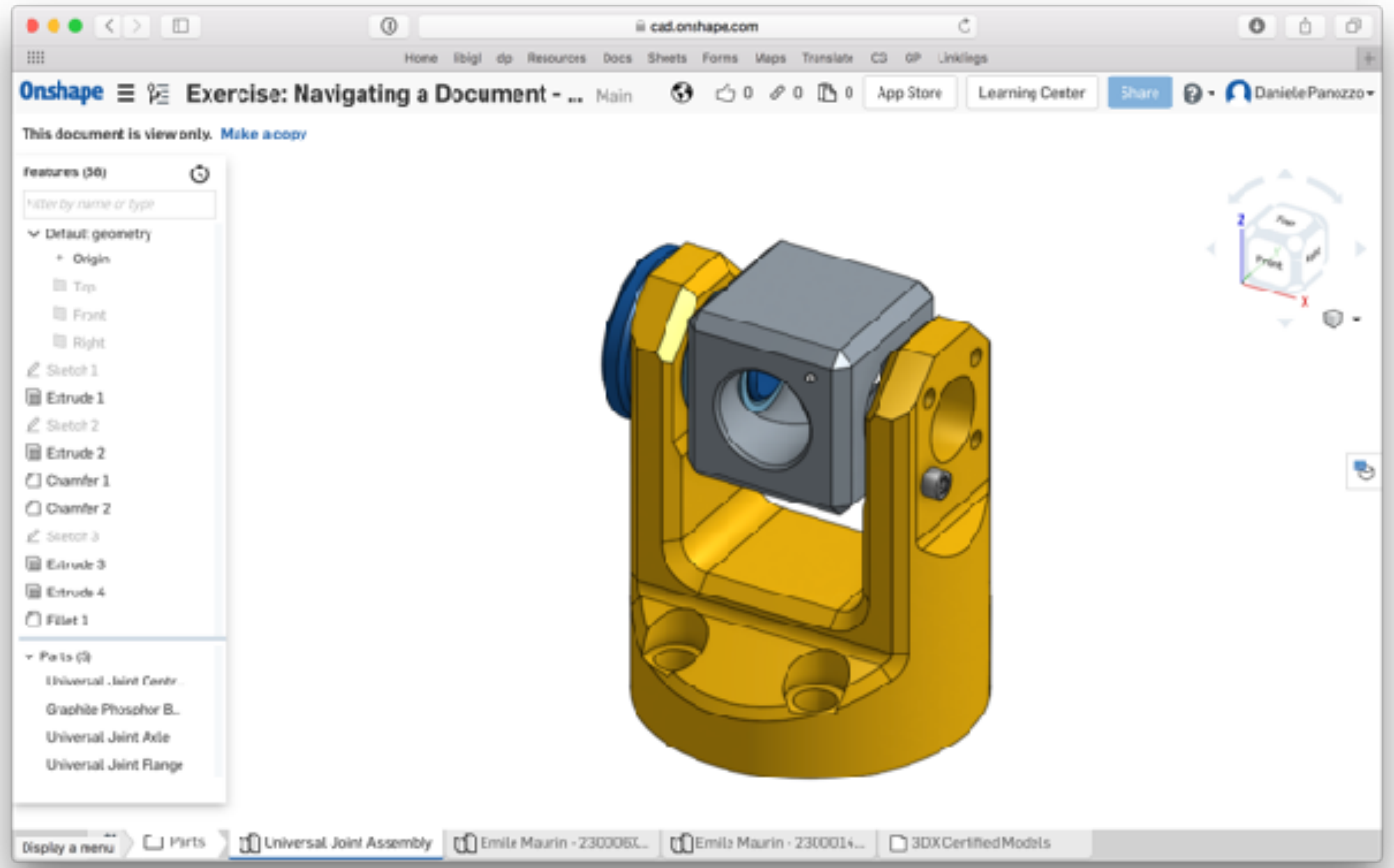
https://cs.nyu.edu/~teseo/

NYU | COURANT INSTITUTE OF MATHEMATICAL SCIENCES

Research Overview



Blender



OnShape

Geometry

Appearance

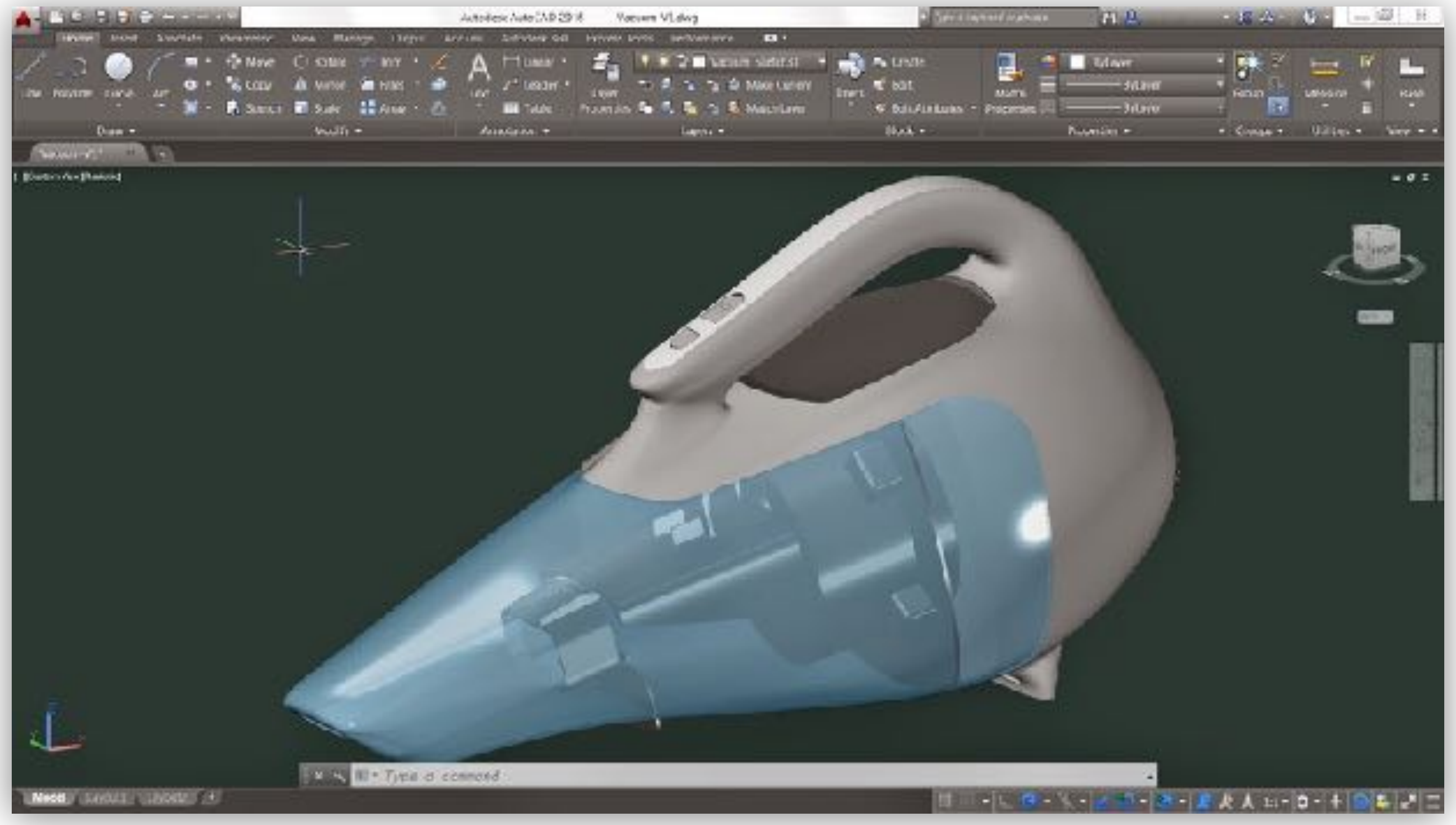
Fabricability

Stability

Robustness

Cost

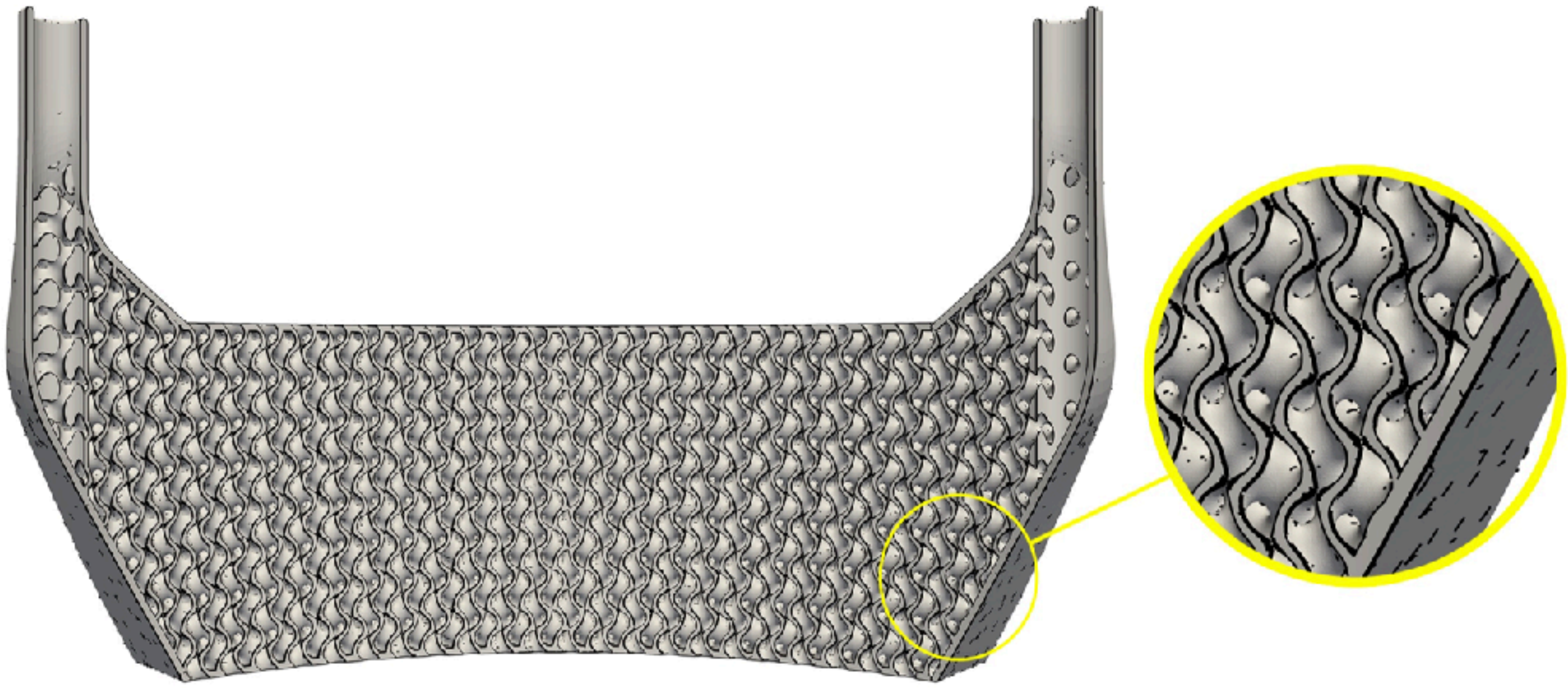
Maya



AutoCAD

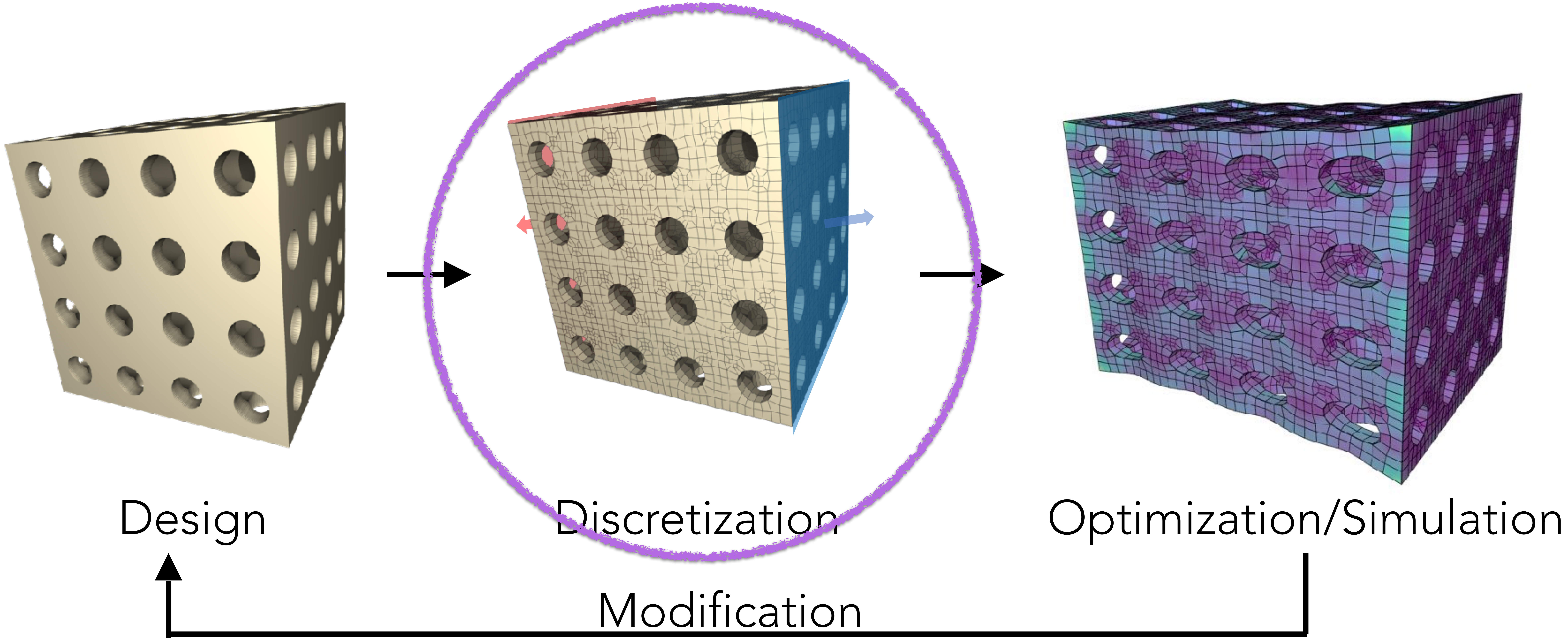
Examples

Volume Reduction



Heat Flux Increase

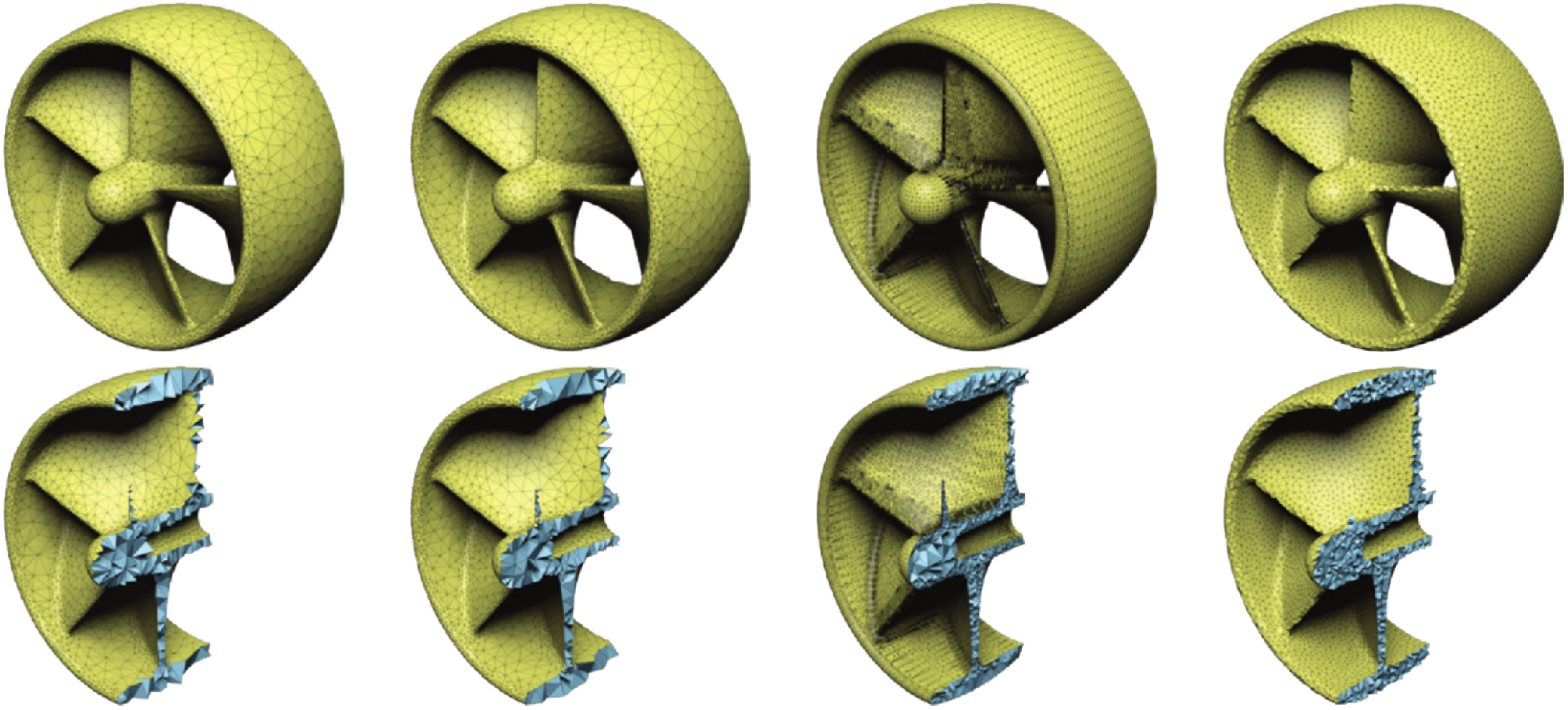
Design Pipeline



3D Geometry Is Challenging

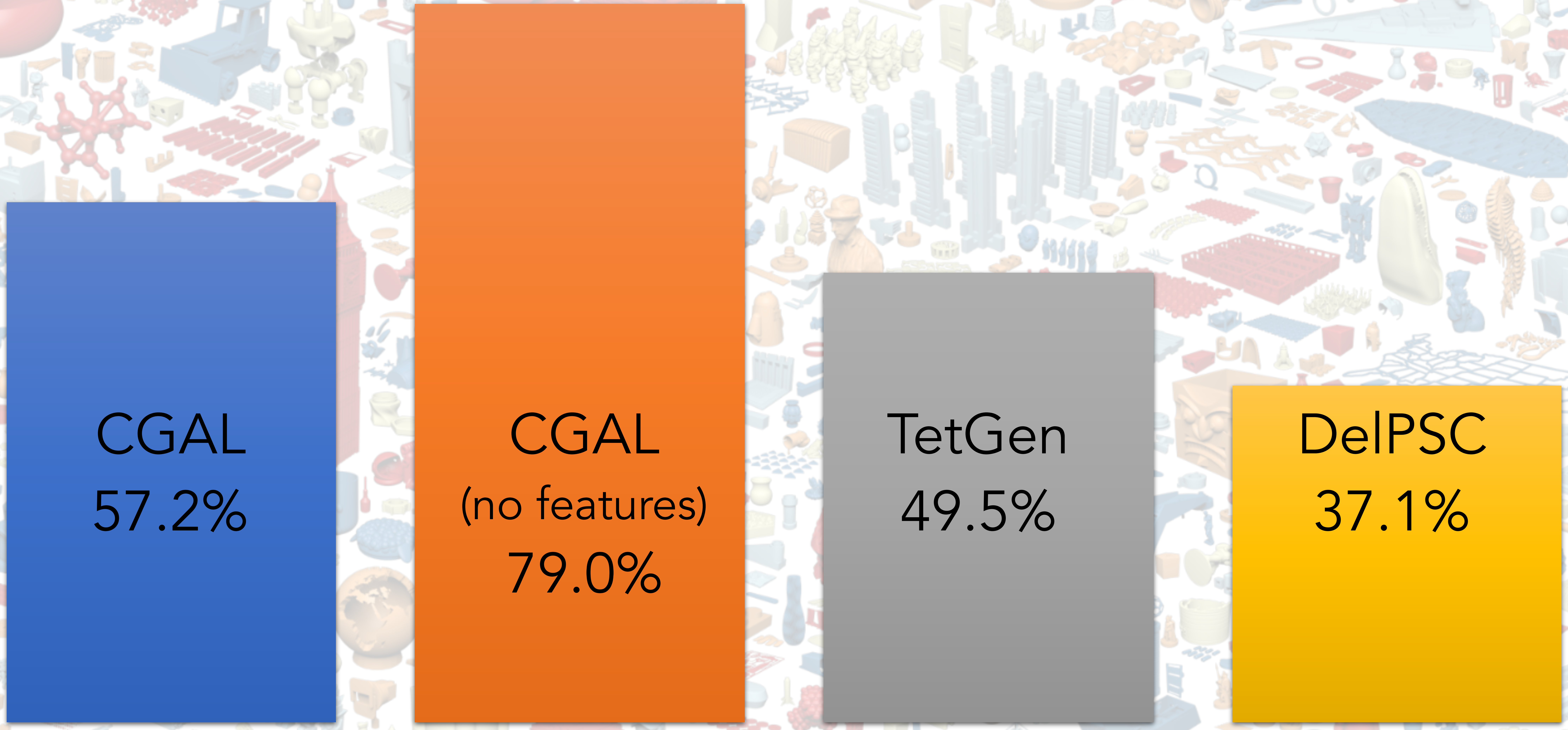
- A canonical representation does not exist
- Most operations are not closed under the floating point representation:
  - Not handling this results in lack of robustness
  - Handling it increases dramatically the algorithmic complexity, increasing the chances of implementation errors (which are a nightmare to debug)

Case Study: Tetrahedral Meshing



CGAL | CGAL (without feature) | TetGen | DelPSC

Success Rate



Why?

- Problem statement imposes **strong assumptions on the input**, which are rare in real-data

- Modeling tools use **operations not closed under the representation** (for example trimming for NURBS), introducing a plethora of degenerate configurations

- Implementation of a complex algorithm in floating point is a major challenge, even if the algorithm is provably correct in arbitrary precision

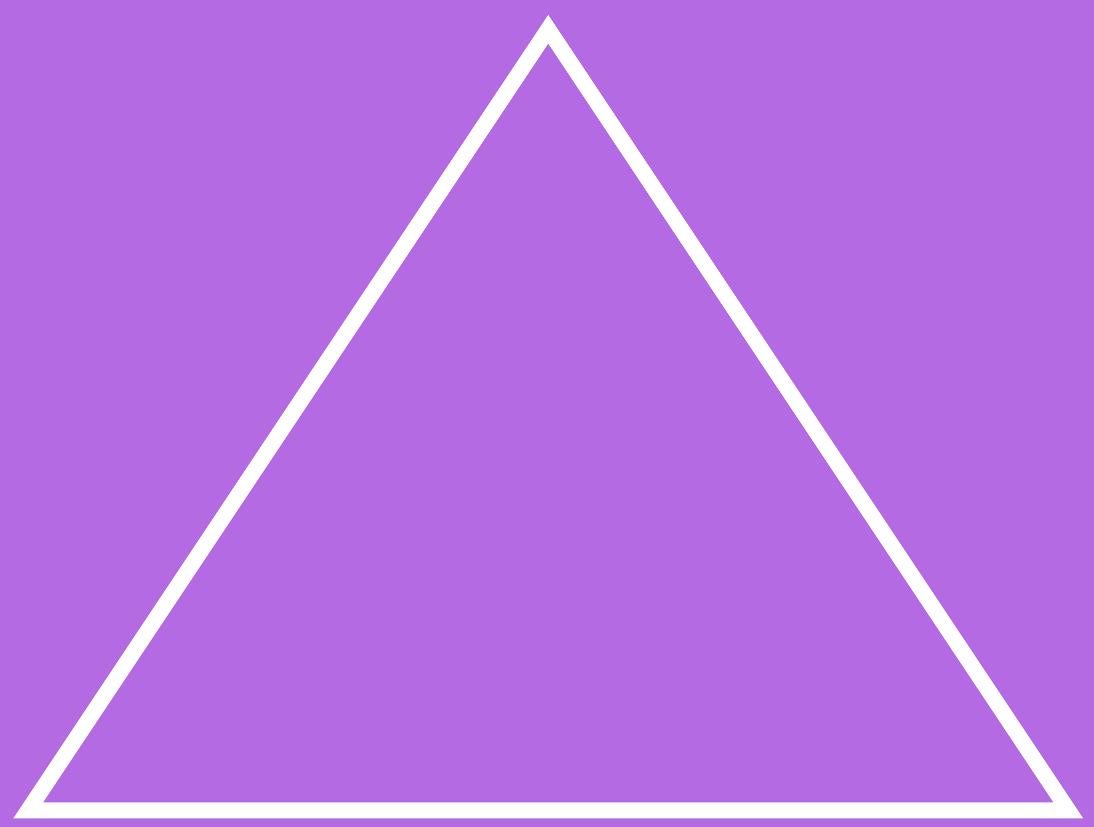
- Large collections of data was not available during the development of these methods

Let’s do it again

- High running times are *preferable* than a failure, since they enable **automation**
- **If** robust floating-point computation is difficult to get right, **exact** computation leads to simpler, but slower, algorithms
- Exact geometry is often **not required** (and sometimes not desired)

Which element is more accurate for a non-linear elasticity problem given a fixed wall clock time budget?

1

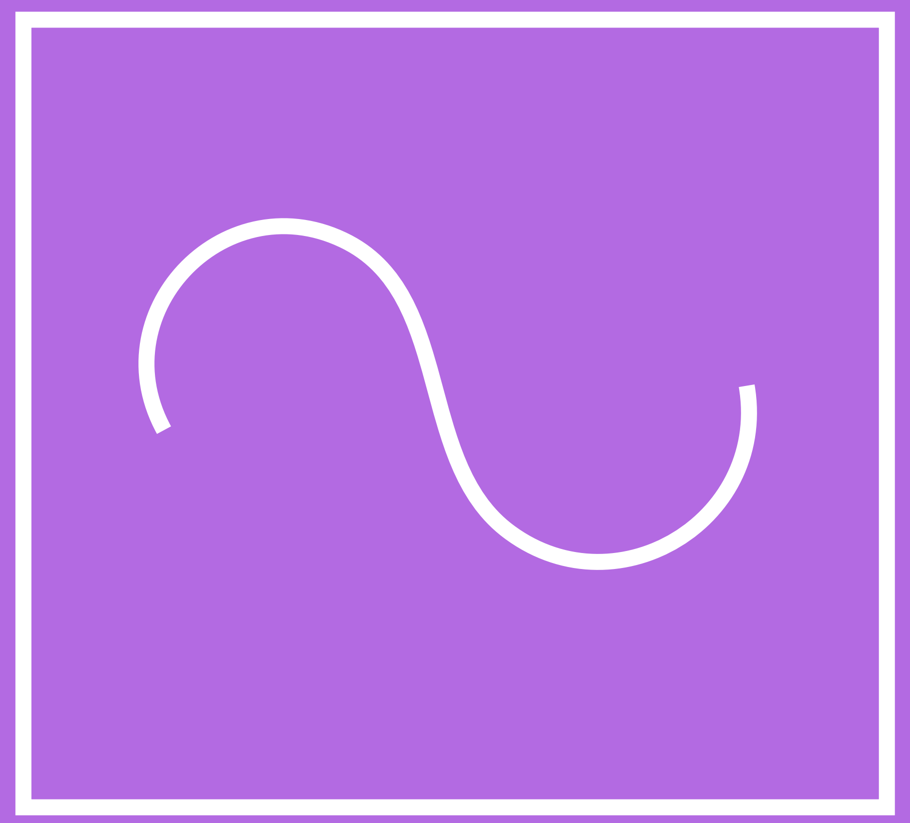


Quadratic Lagrangian Tetrahedra

2

Quadratic Lagrangian/Serendipity Hexahedra

3



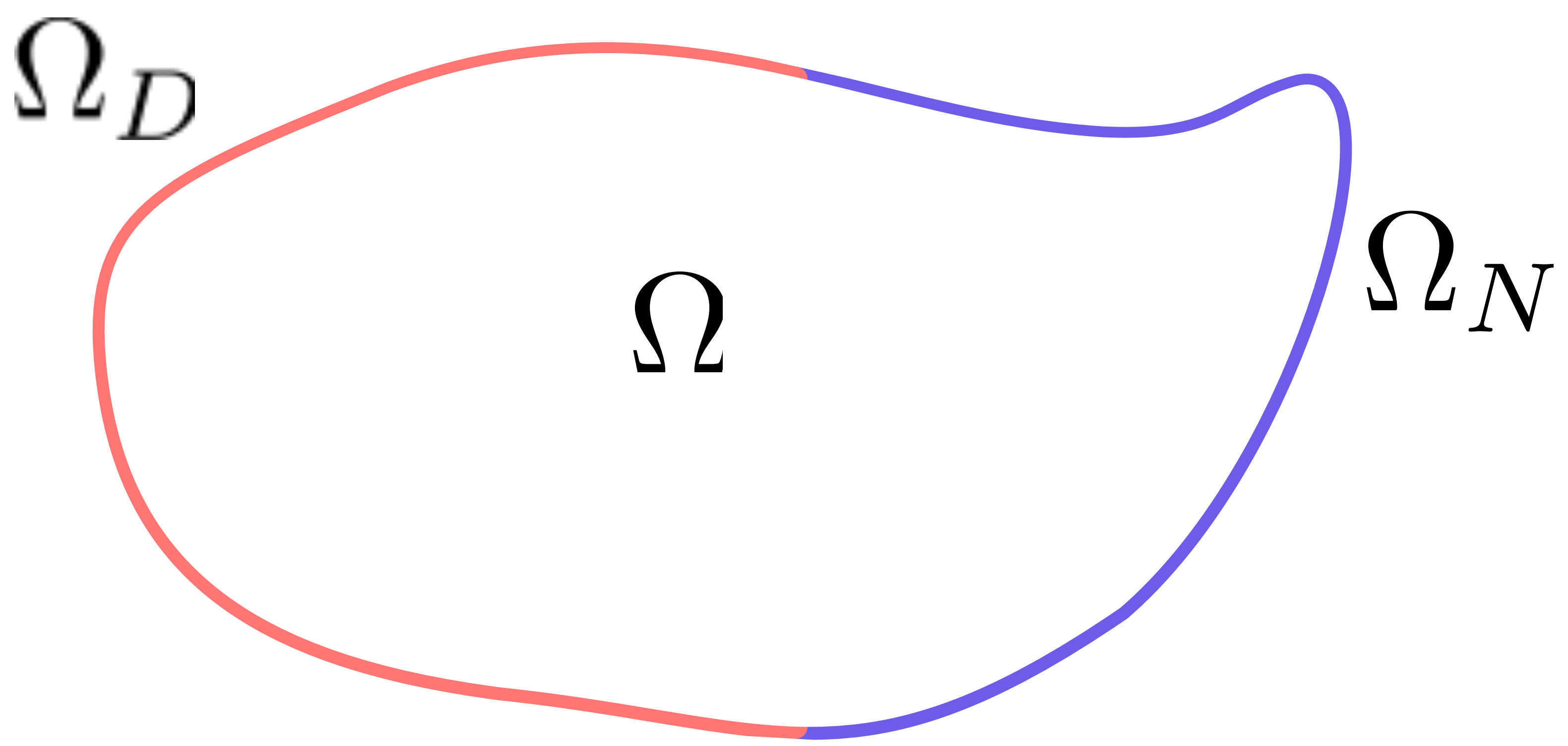
Quadratic Splines on Hexahedra (IGA)

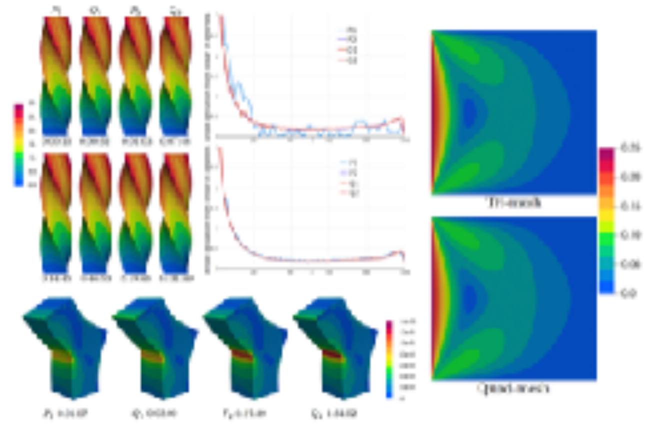
Problem

- Solve elliptic PDE $\mathcal{F}(x, u, \nabla u, D^2 u) = b$

  subject to $u = d$ on $\partial\Omega_D$ and $\nabla u \cdot n = f$ on $\partial\Omega_N$

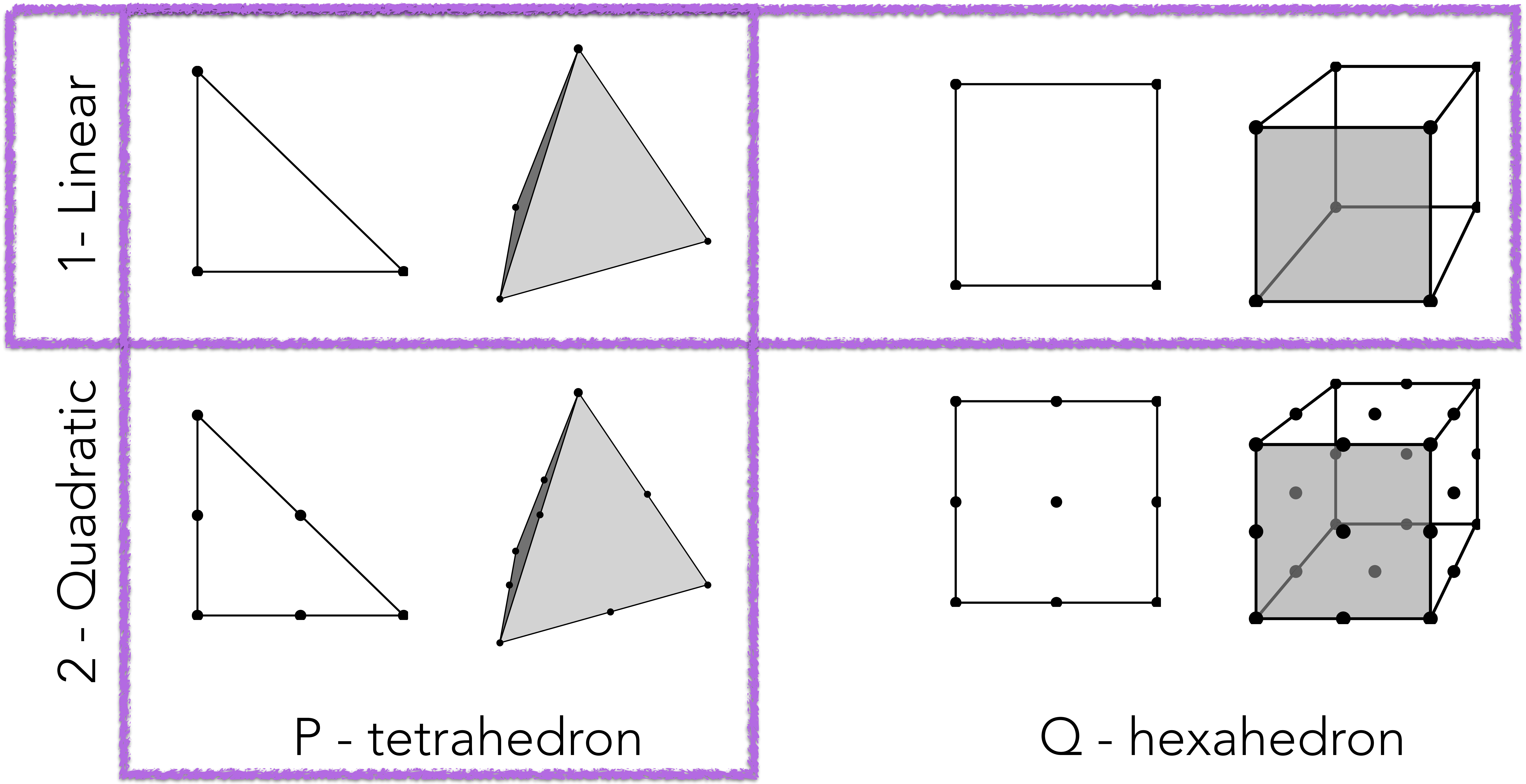
- For common elliptic PDEs
  - Elasticity (Linear and Non-Linear)
  - Poisson



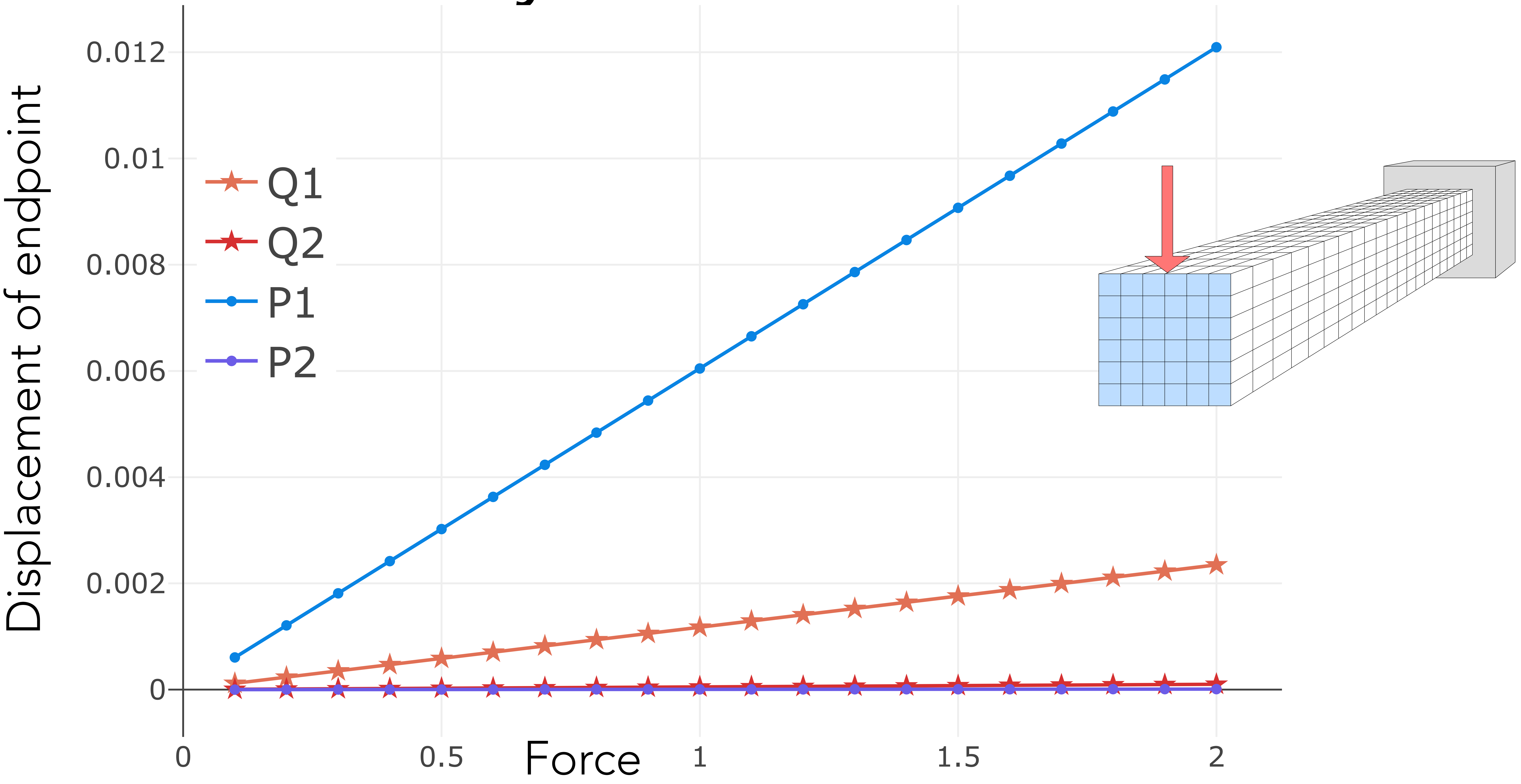


A Large Scale Comparison of Tetrahedral and Hexahedral Elements for Finite Element Analysis
Teseo Schneider, Yixin Hu, Xifeng Gao, Jeremie Dumas, Denis Zorin, Daniele Panozzo,
submitted, 2019
[Paper] [Code] [Data]

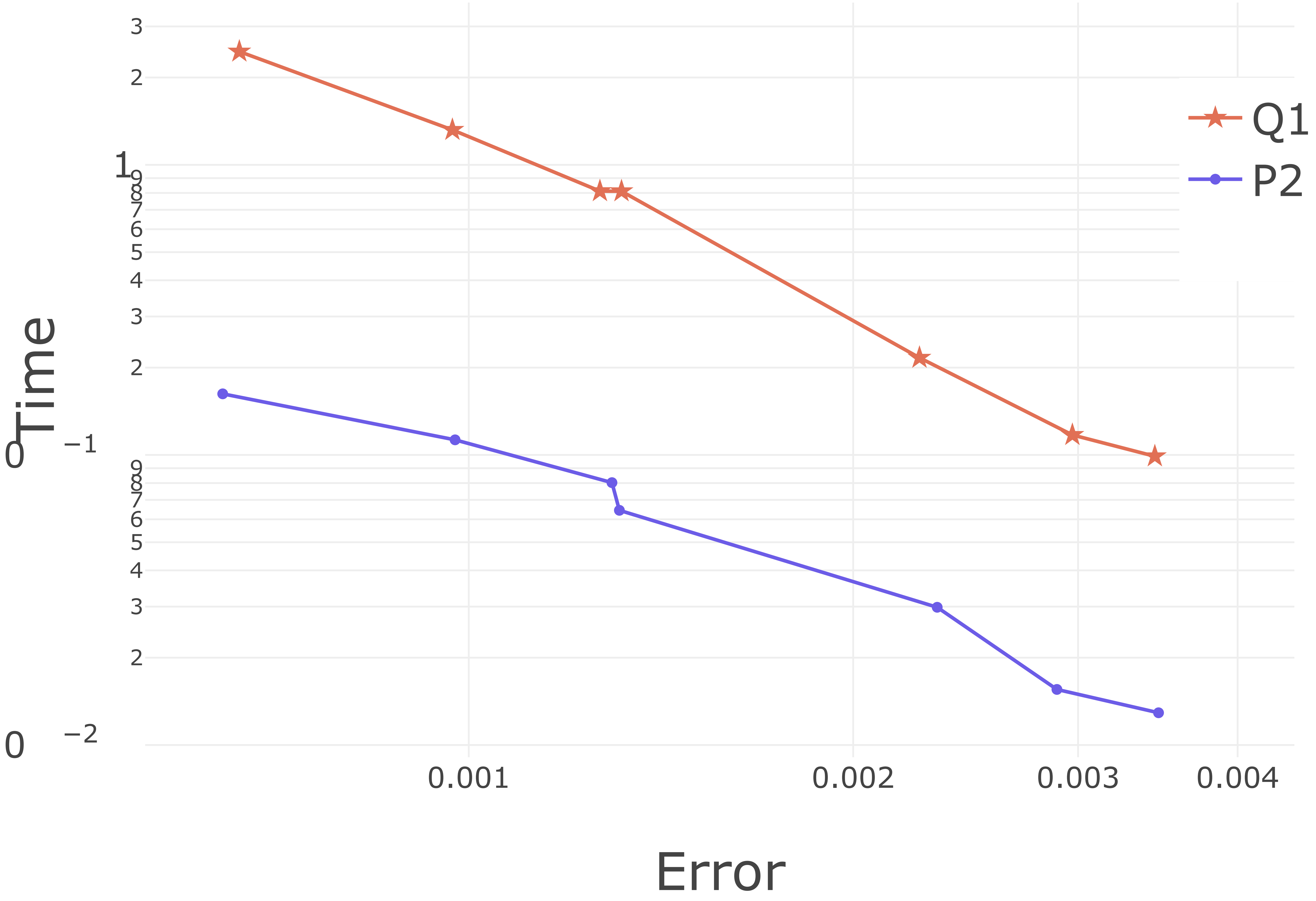
Choice of Basis



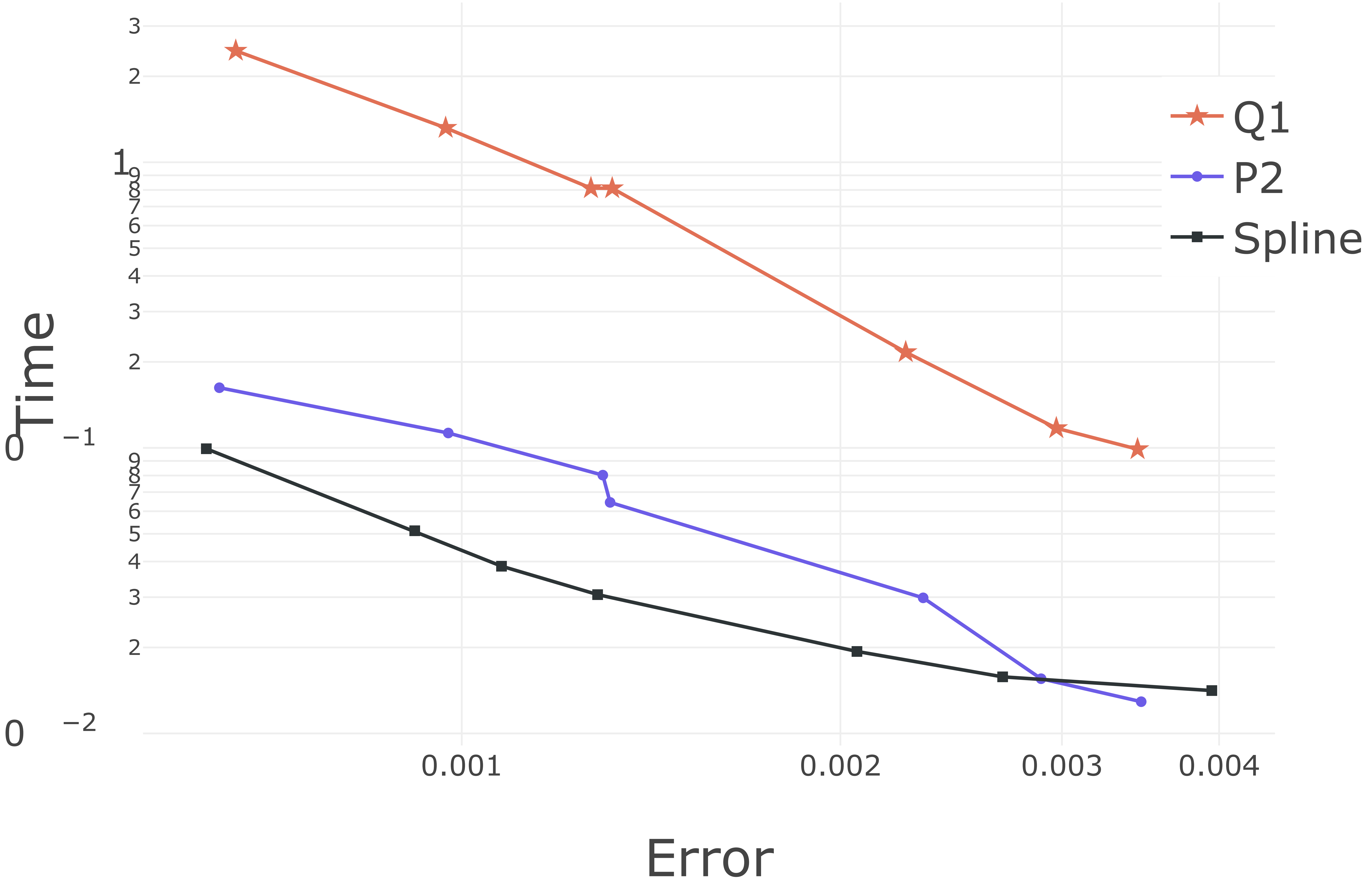
Elasticity – Bended Bar



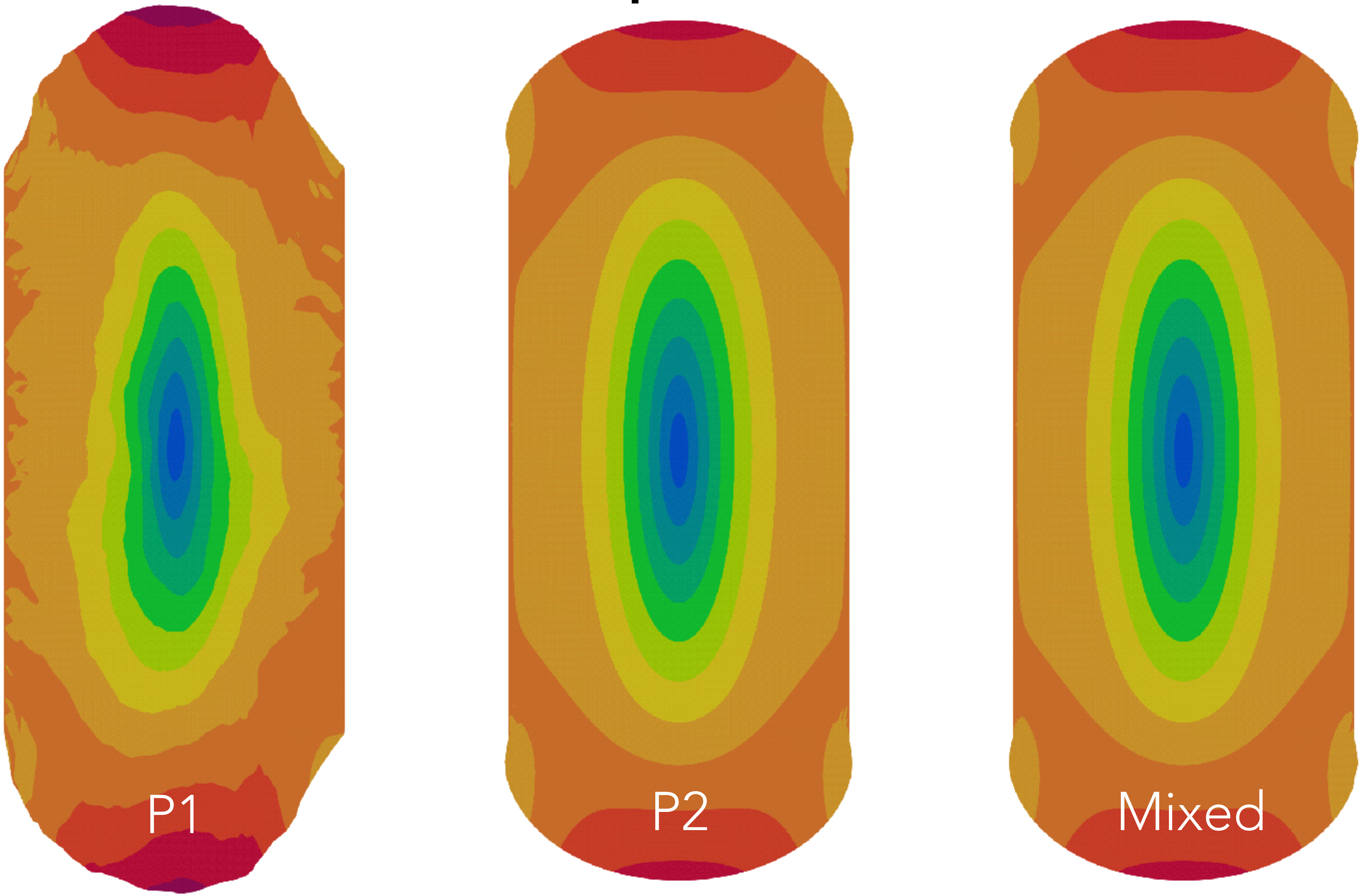
Elasticity – Bended Bar



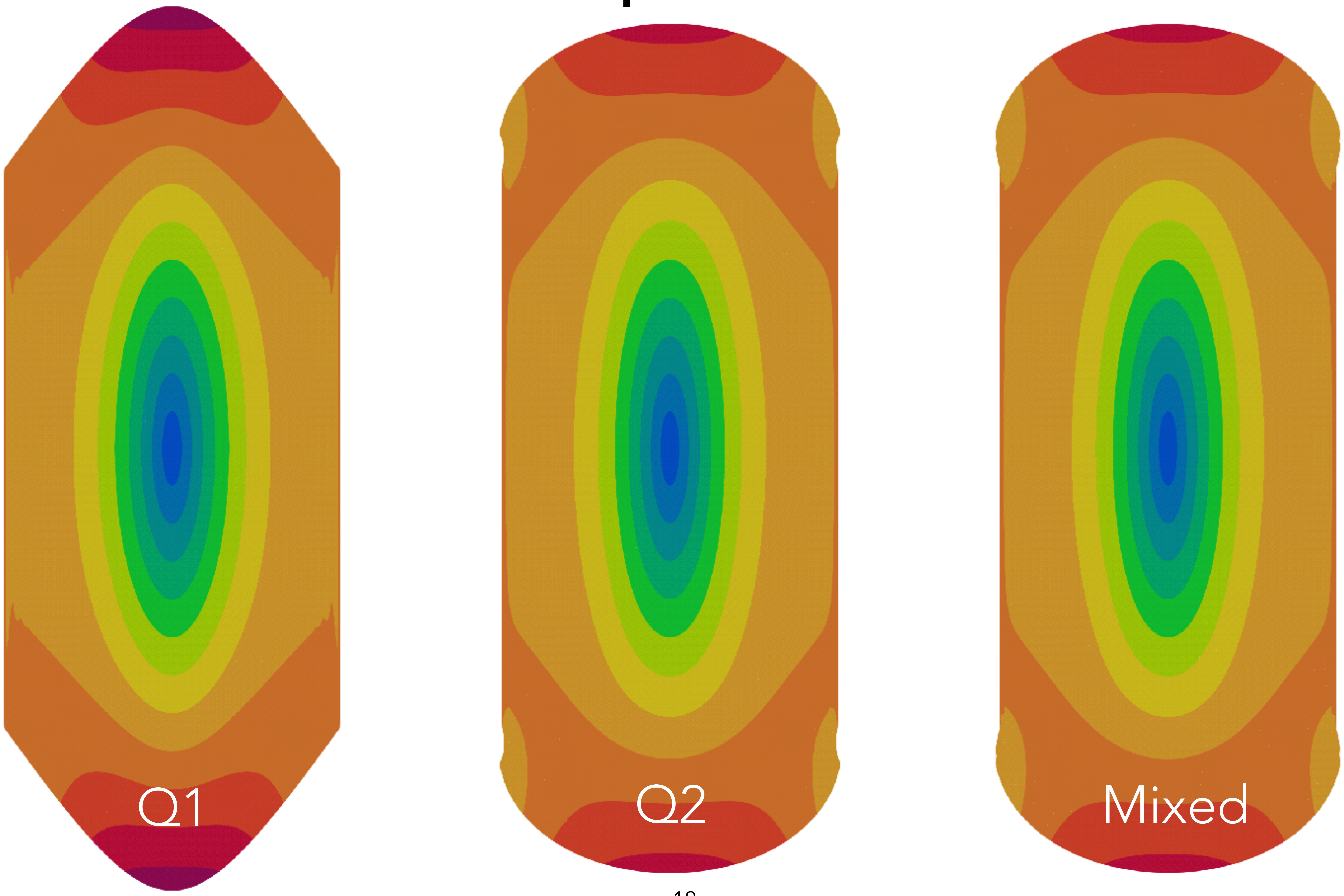
Elasticity – Bended Bar



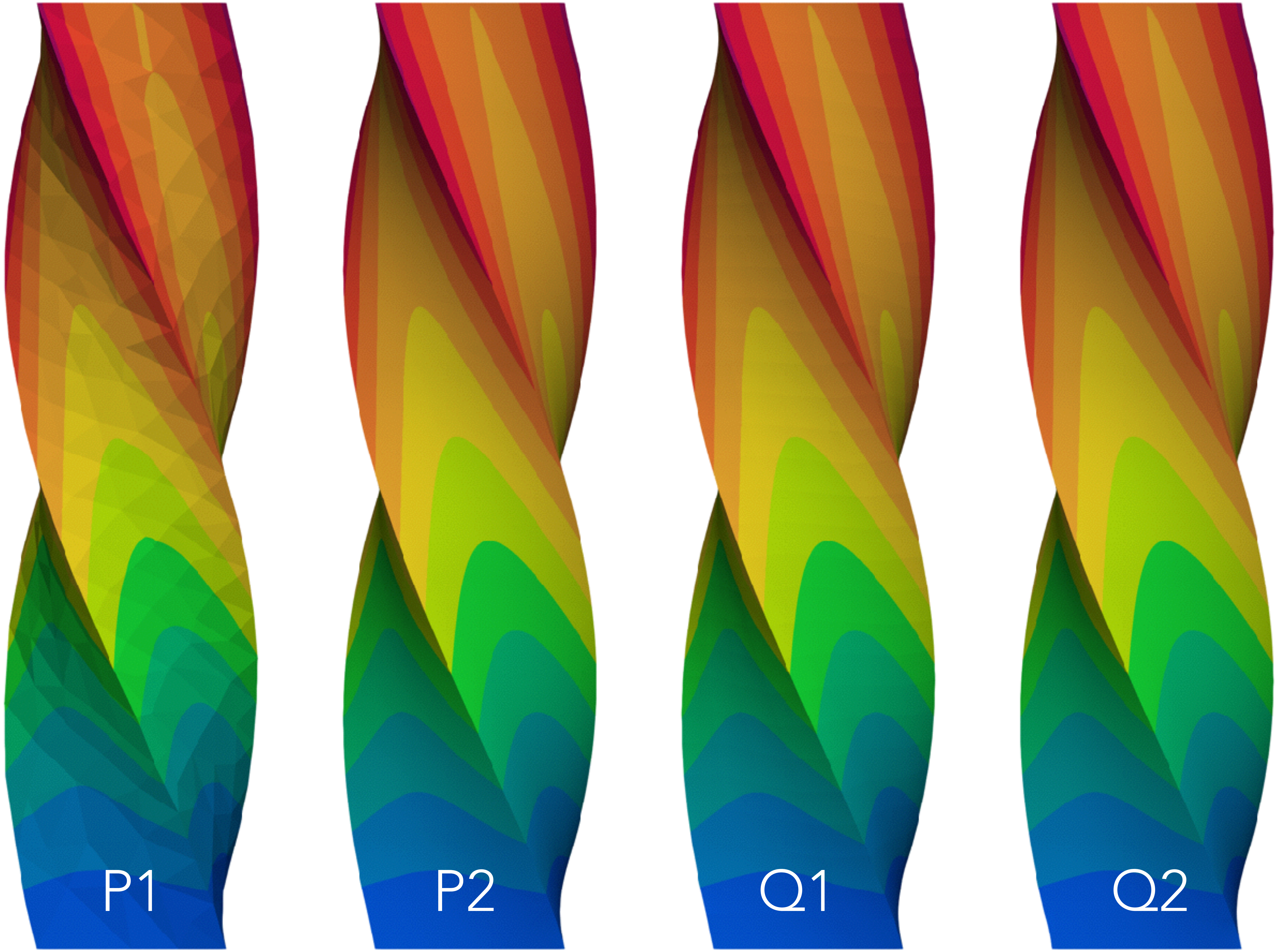
Incompressible



Incompressible

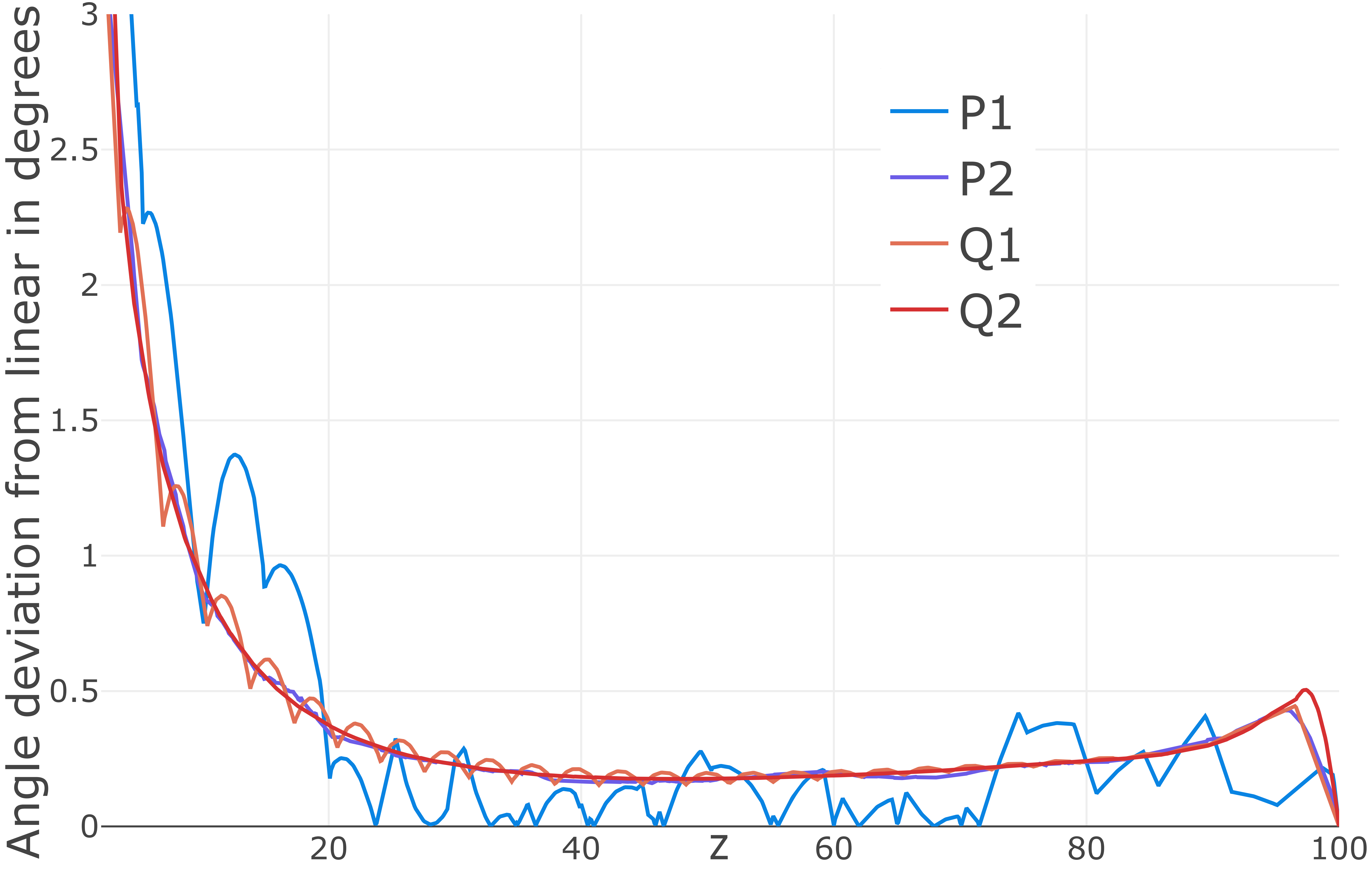


Neo-Hooke – Coarse

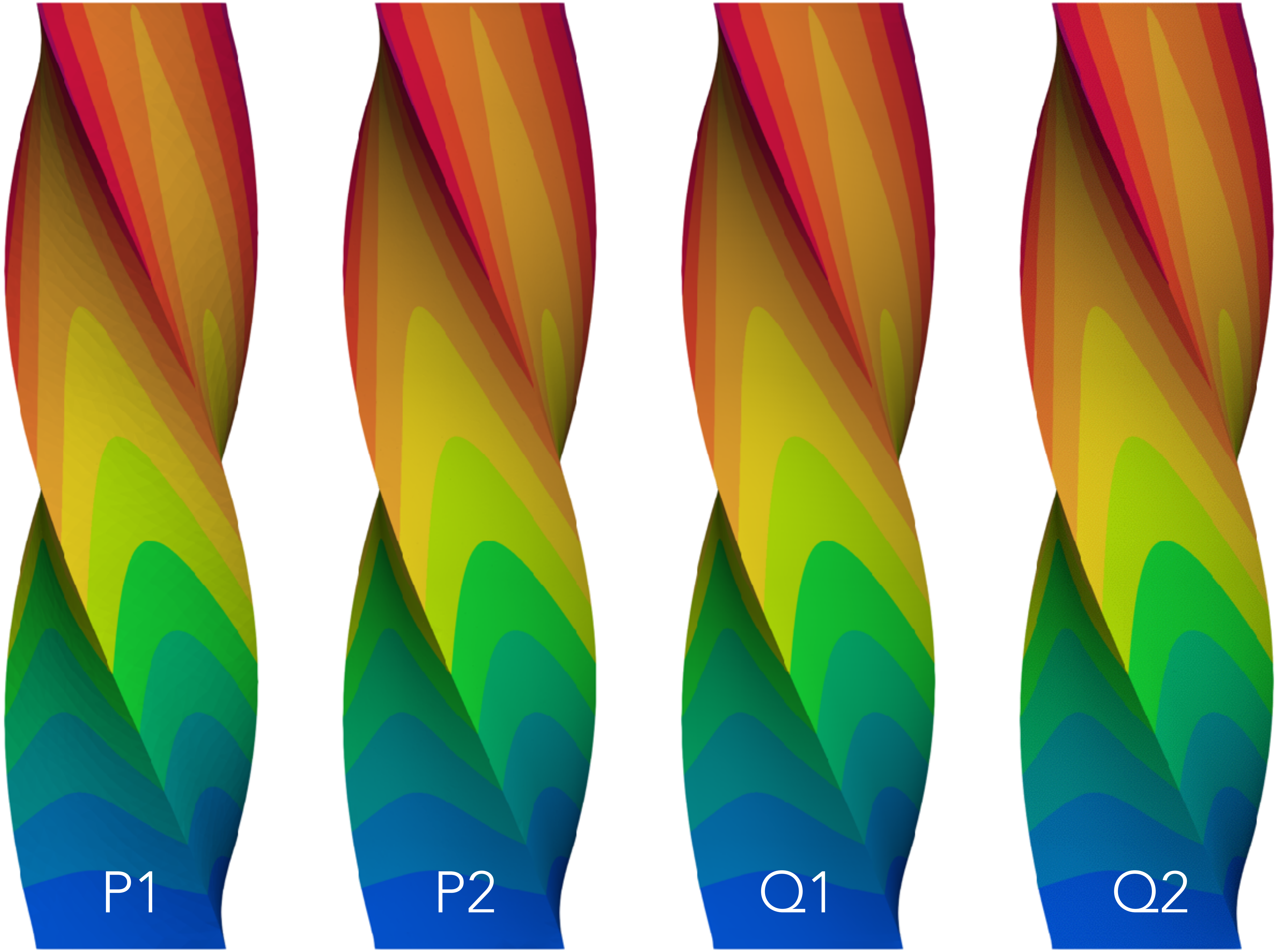


0:00:10 0:01:03 0:00:32 0:07:46

Neo-Hooke – Coarse

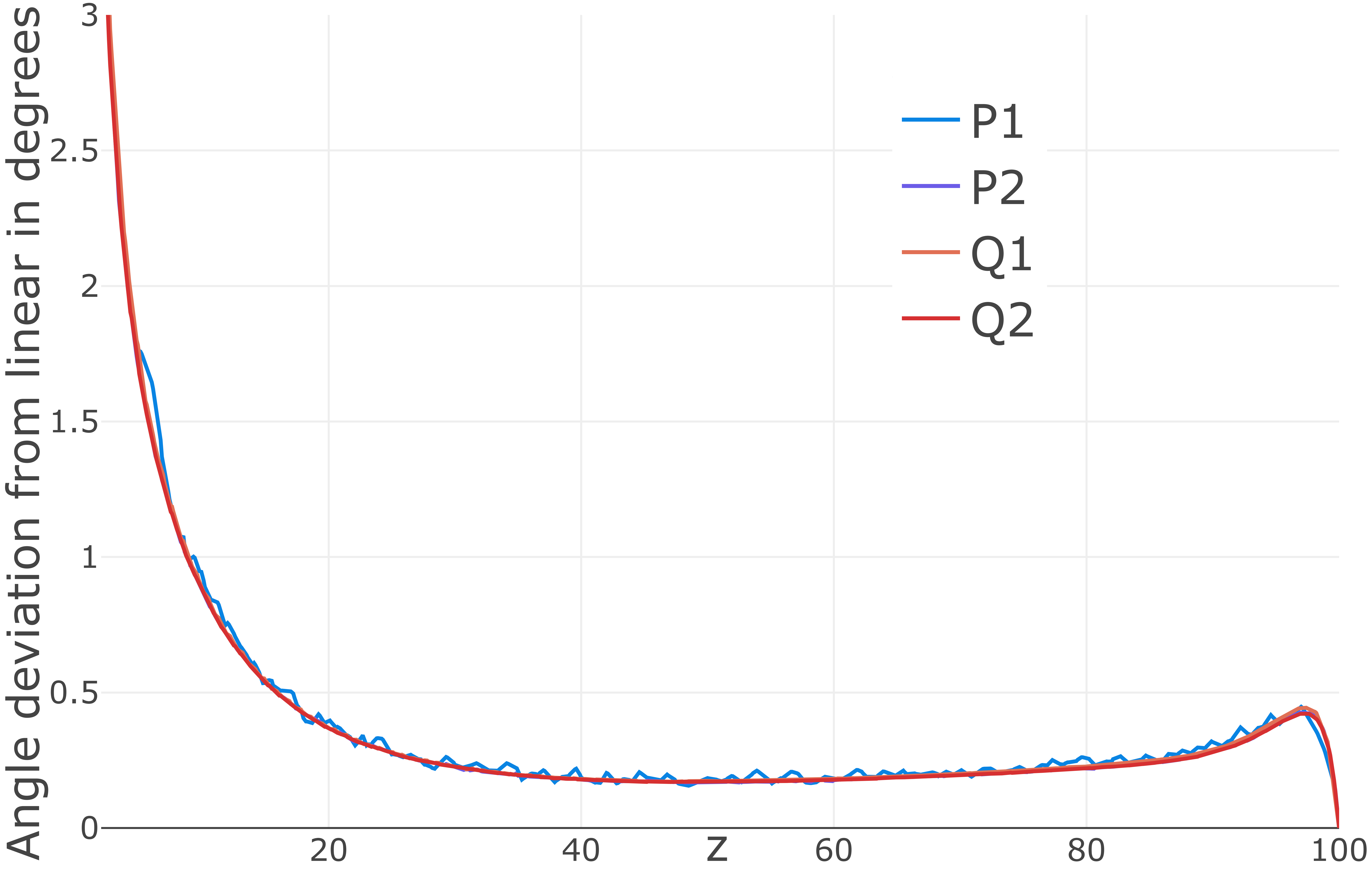


Neo-Hooke – Dense



0:14:45 5:19:43 0:46:53 31:31:49

Neo-Hooke – Dense



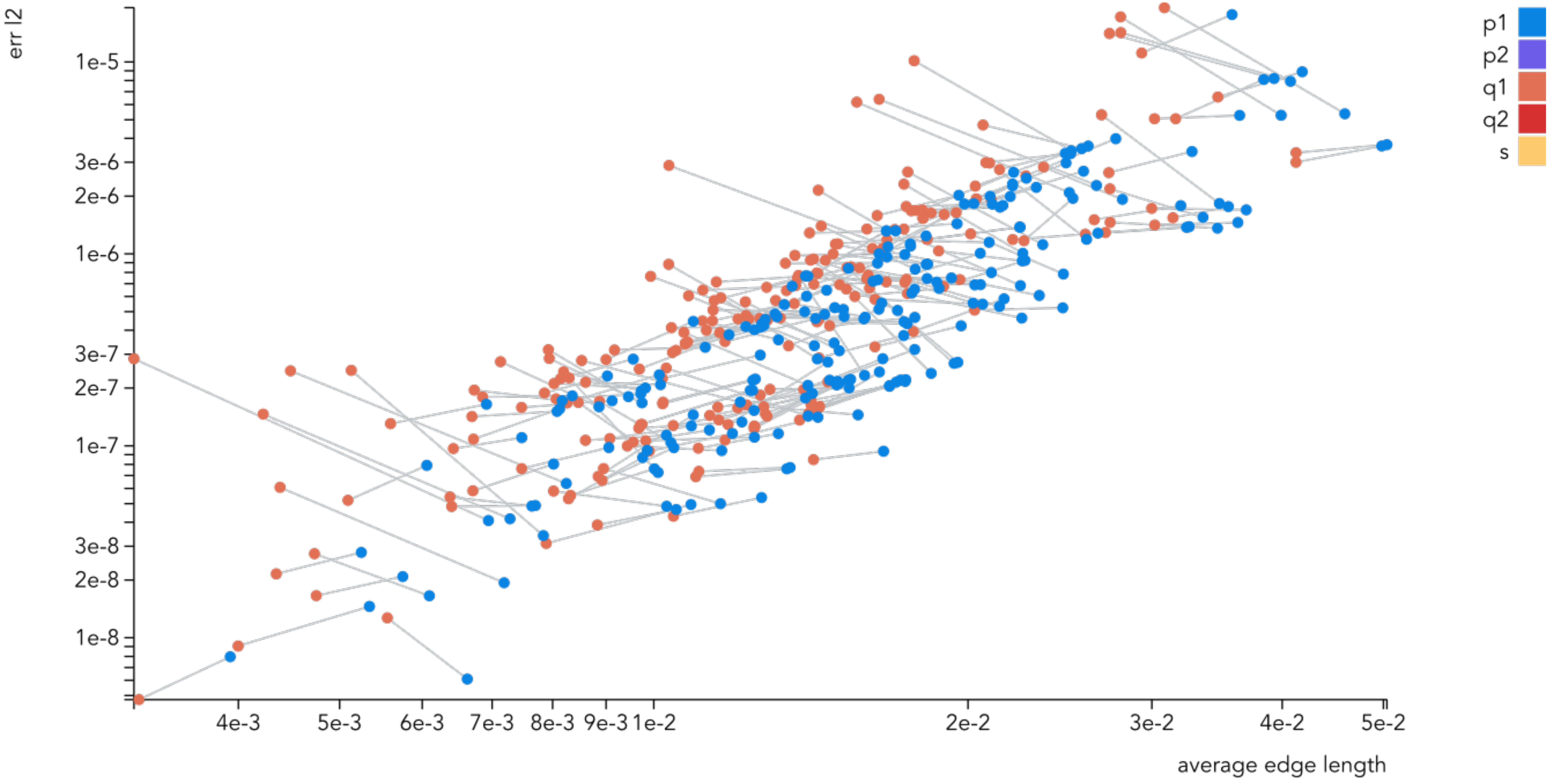
Dataset

- Hexalab https://www.hexalab.net/
  - 16 state-of-the-art hex-meshing algorithms
  - 237 meshes
  - 8 flips 3.4%
- Thingi10k
  - 3200 meshes with MeshGems
  - 577 flips 18.0%
- For a given hex mesh, we generate a tetrahedral mesh with the same number of vertices

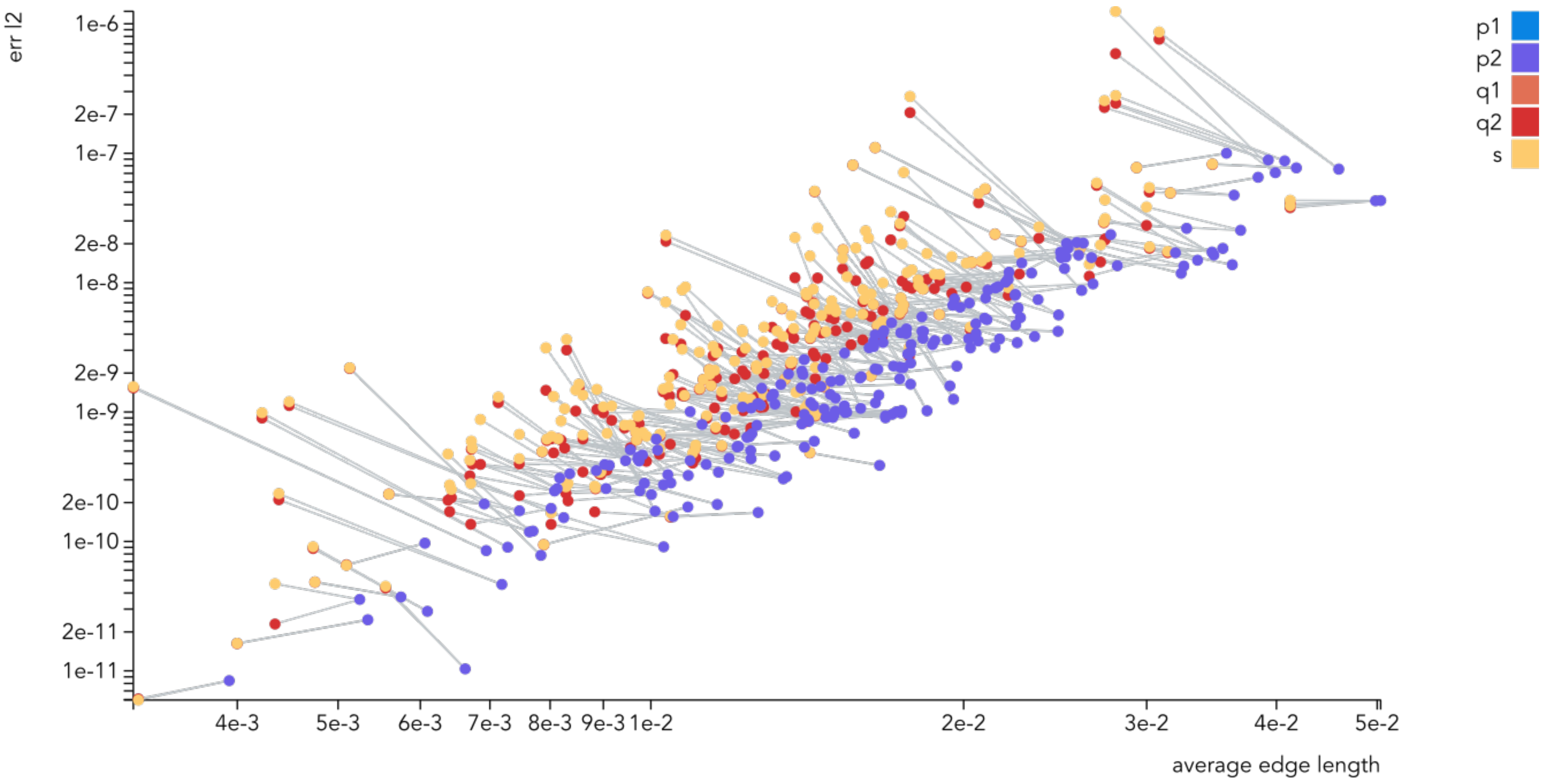
Interactive Plot

https://polyfem.github.io/tet-vs-hex/plot.html

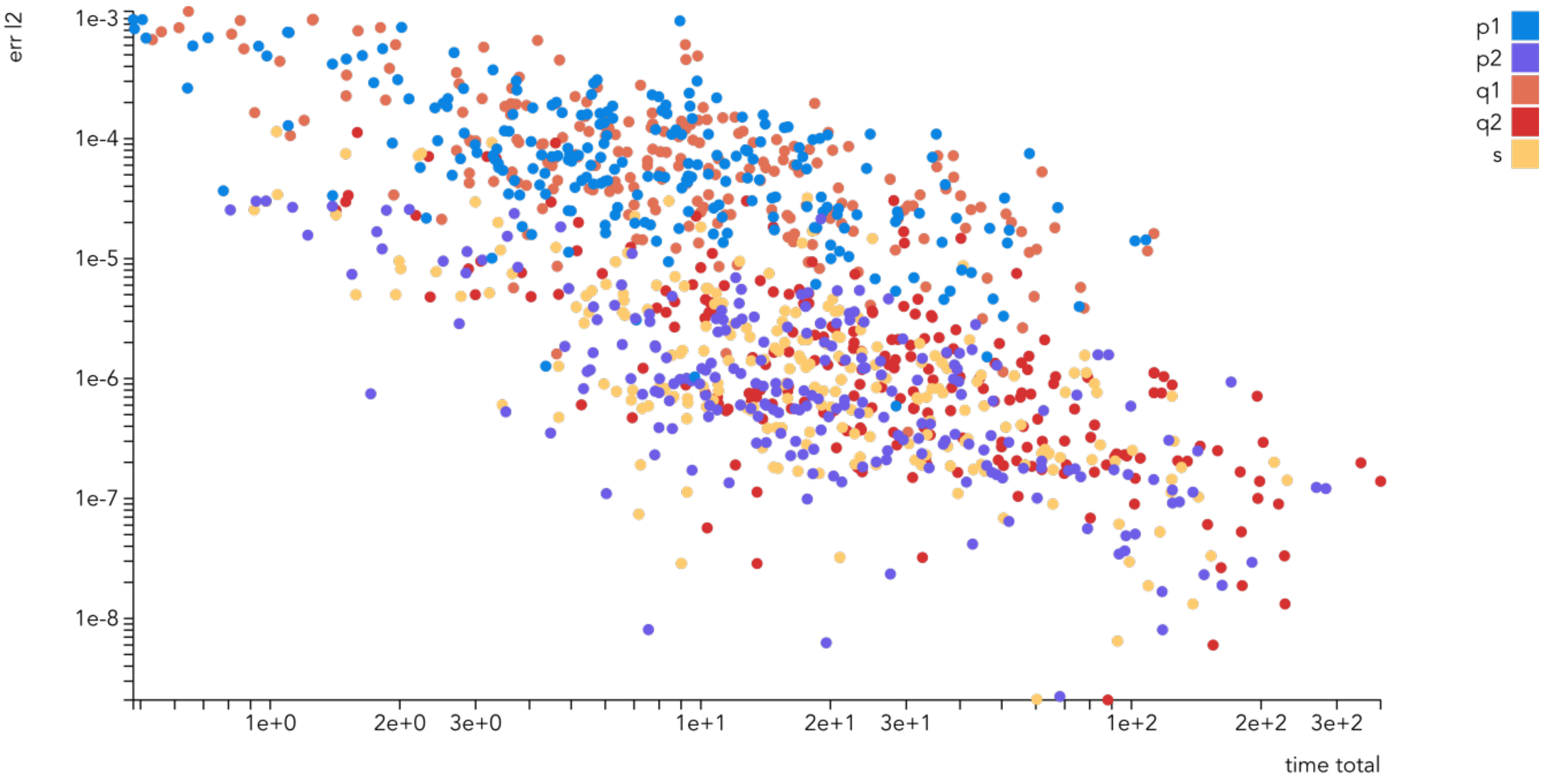
Hexalab – no-flips



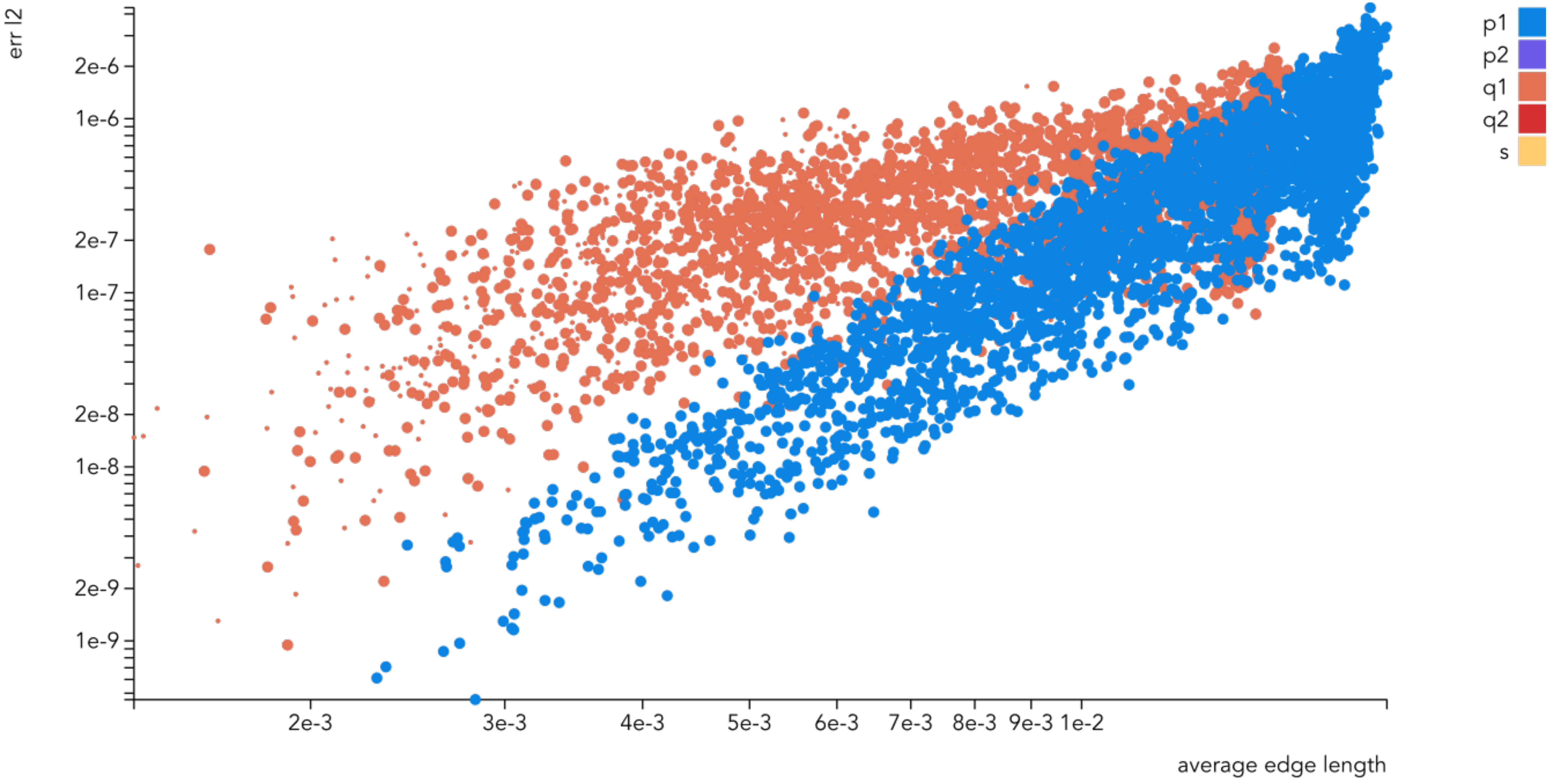
Hexalab – no-flips



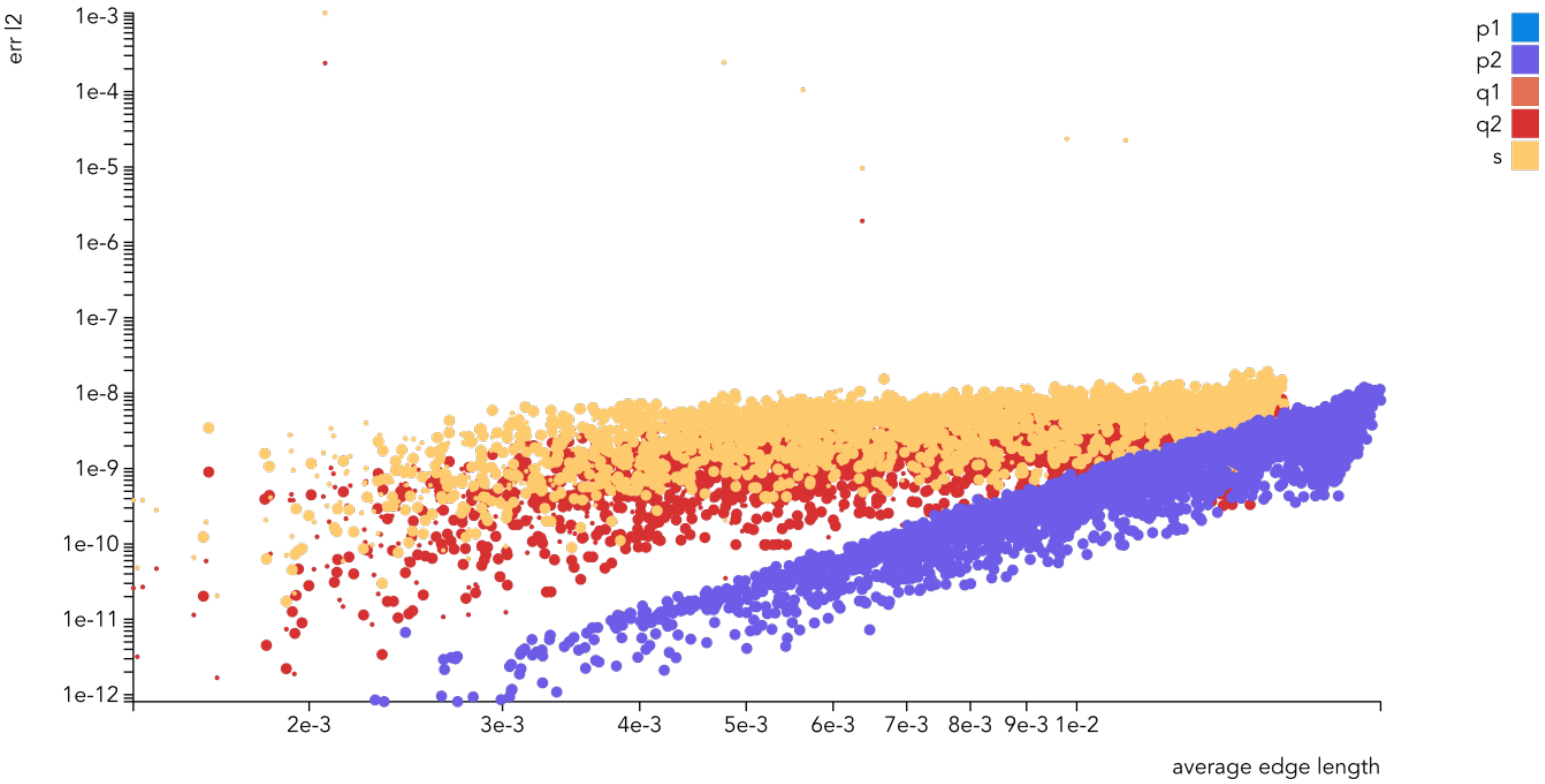
Hexalab – no-flips



Thingi10k

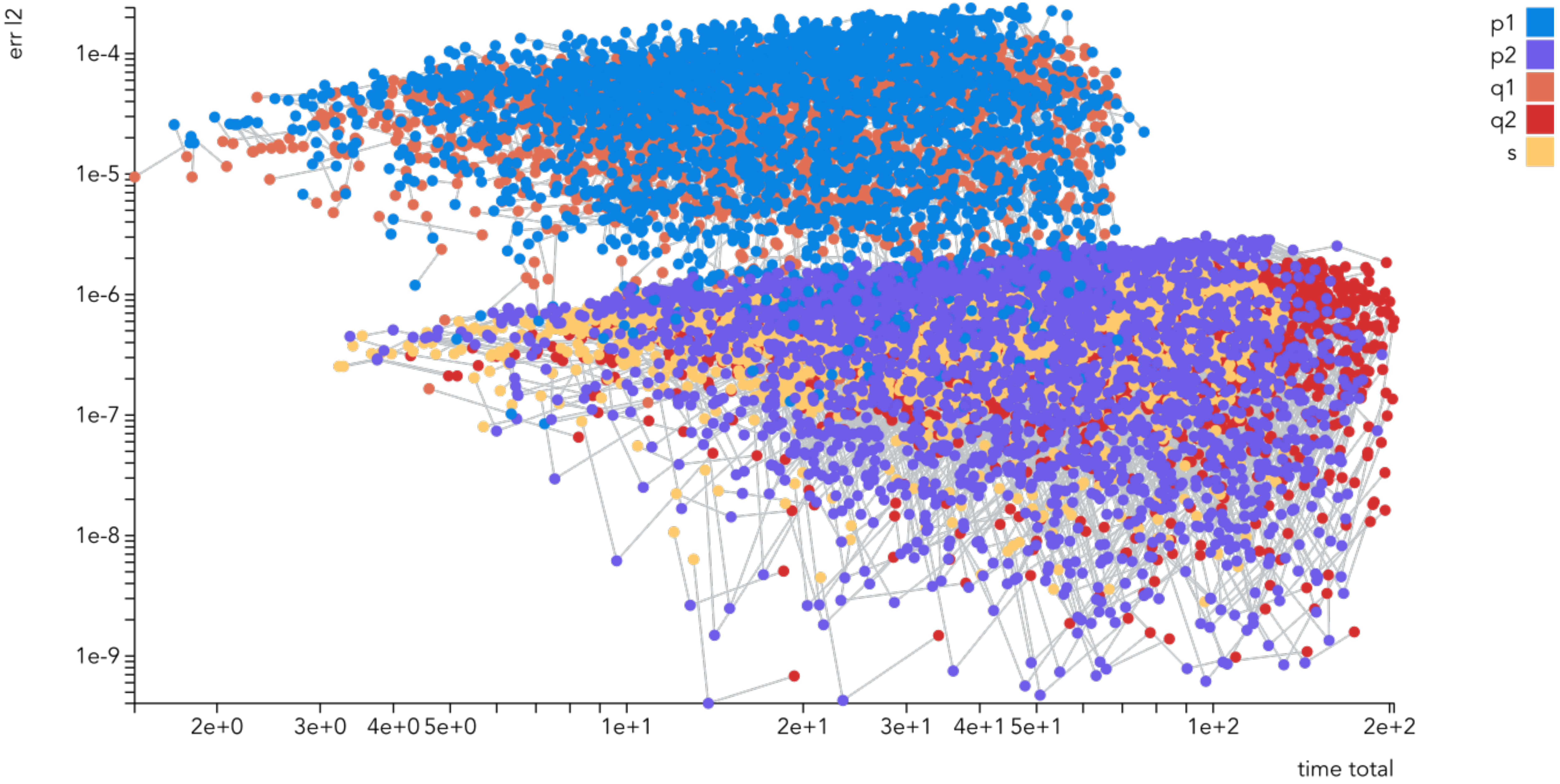

err l2
2e-6
1e-6
2e-7
1e-7
2e-8
1e-8
2e-9
1e-9
2e-3
3e-3
4e-3
5e-3
6e-3
7e-3
8e-3
9e-3
1e-2
average edge length
p1
p2
q1
q2
s

Thingi10k



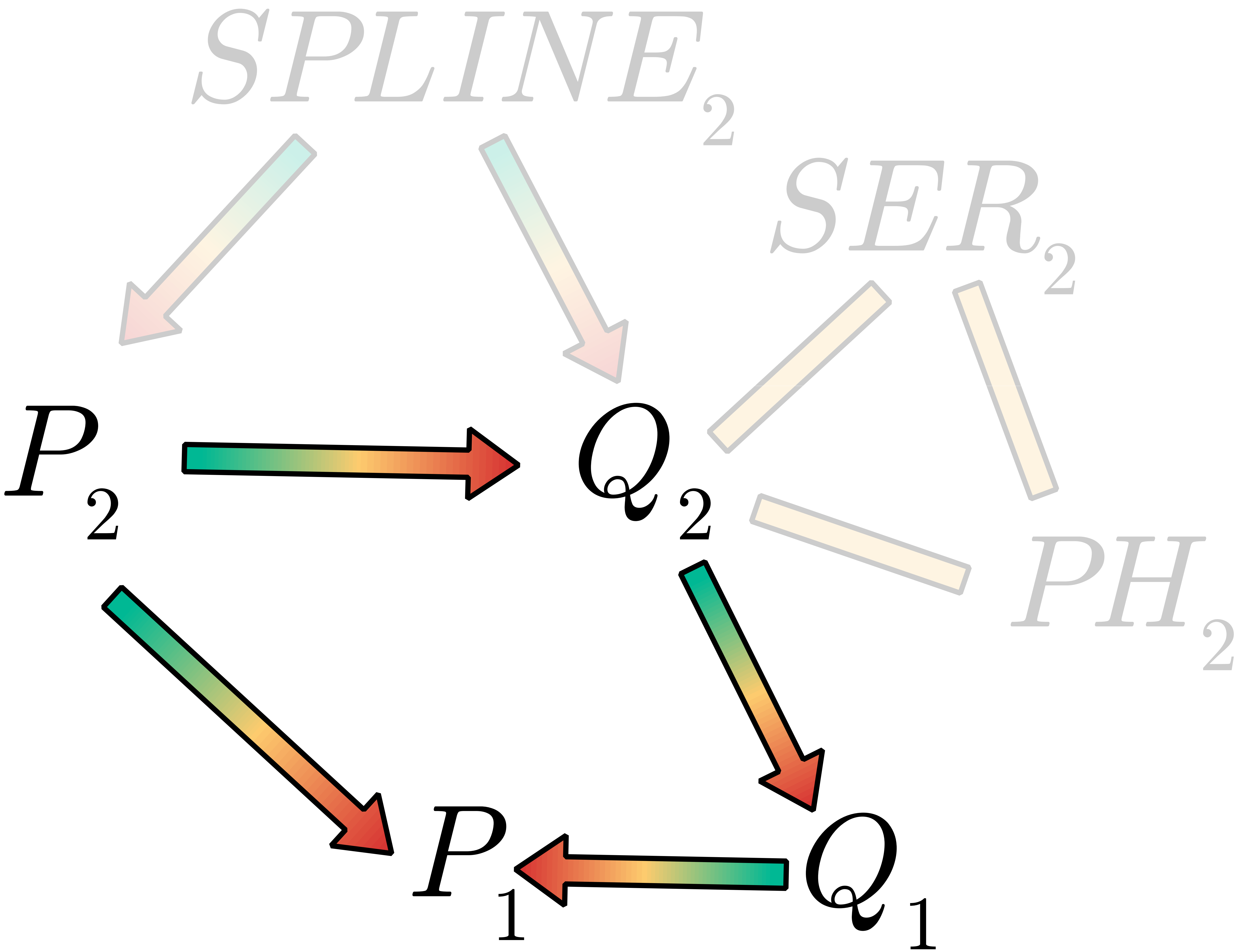
err l2
1e-3
1e-4
1e-5
1e-6
1e-7
1e-8
1e-9
1e-10
1e-11
1e-12
2e-3
3e-3
4e-3
5e-3
6e-3
7e-3
8e-3
9e-3
1e-2
average edge length
p1
p2
q1
q2
s

Thingi10k

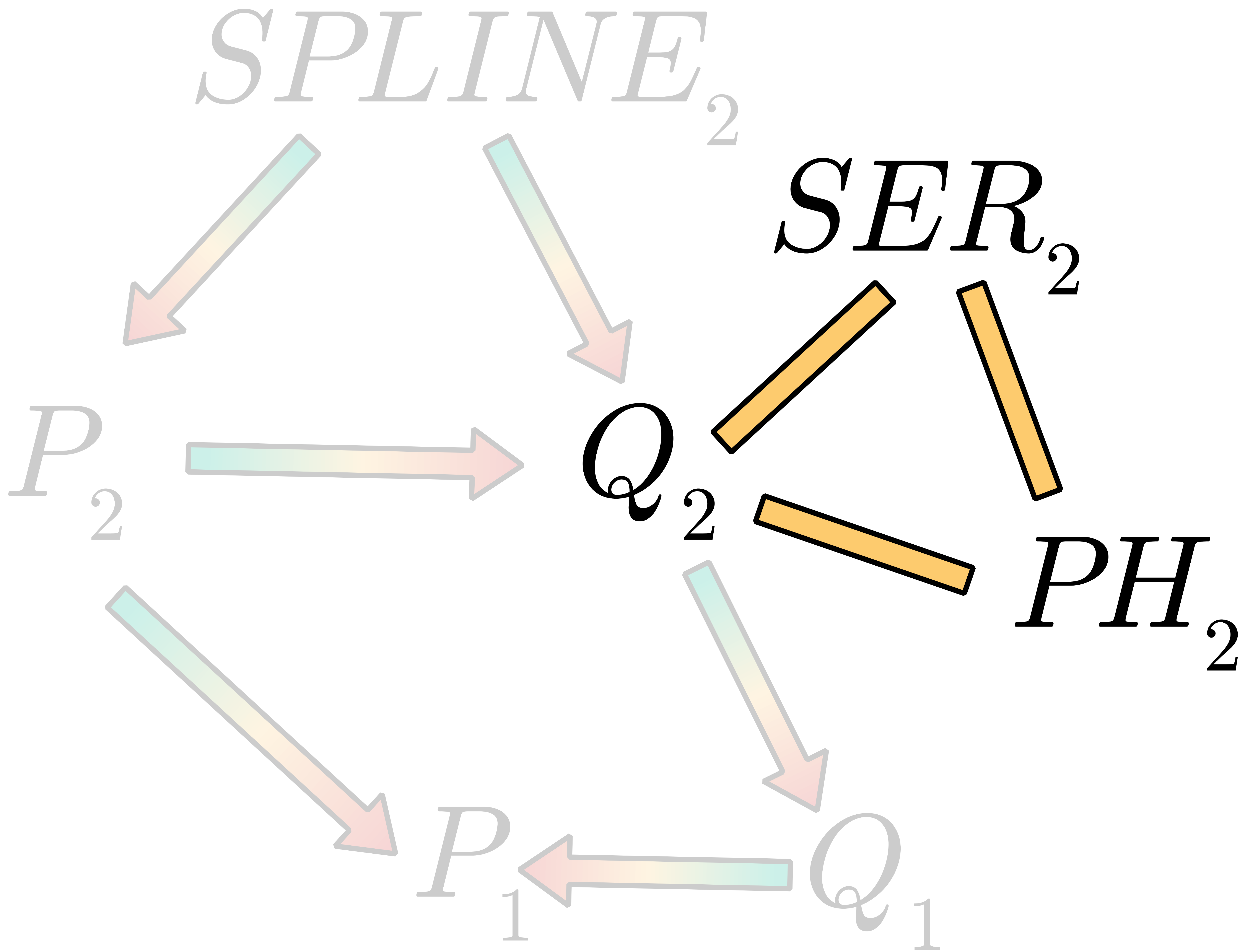


err l2
1e-4
1e-5
1e-6
1e-7
1e-8
1e-9
2e+0
3e+0
4e+0
5e+0
1e+1
2e+1
3e+1
4e+1
5e+1
1e+2
2e+2
time total
p1
p2
q1
q2
s

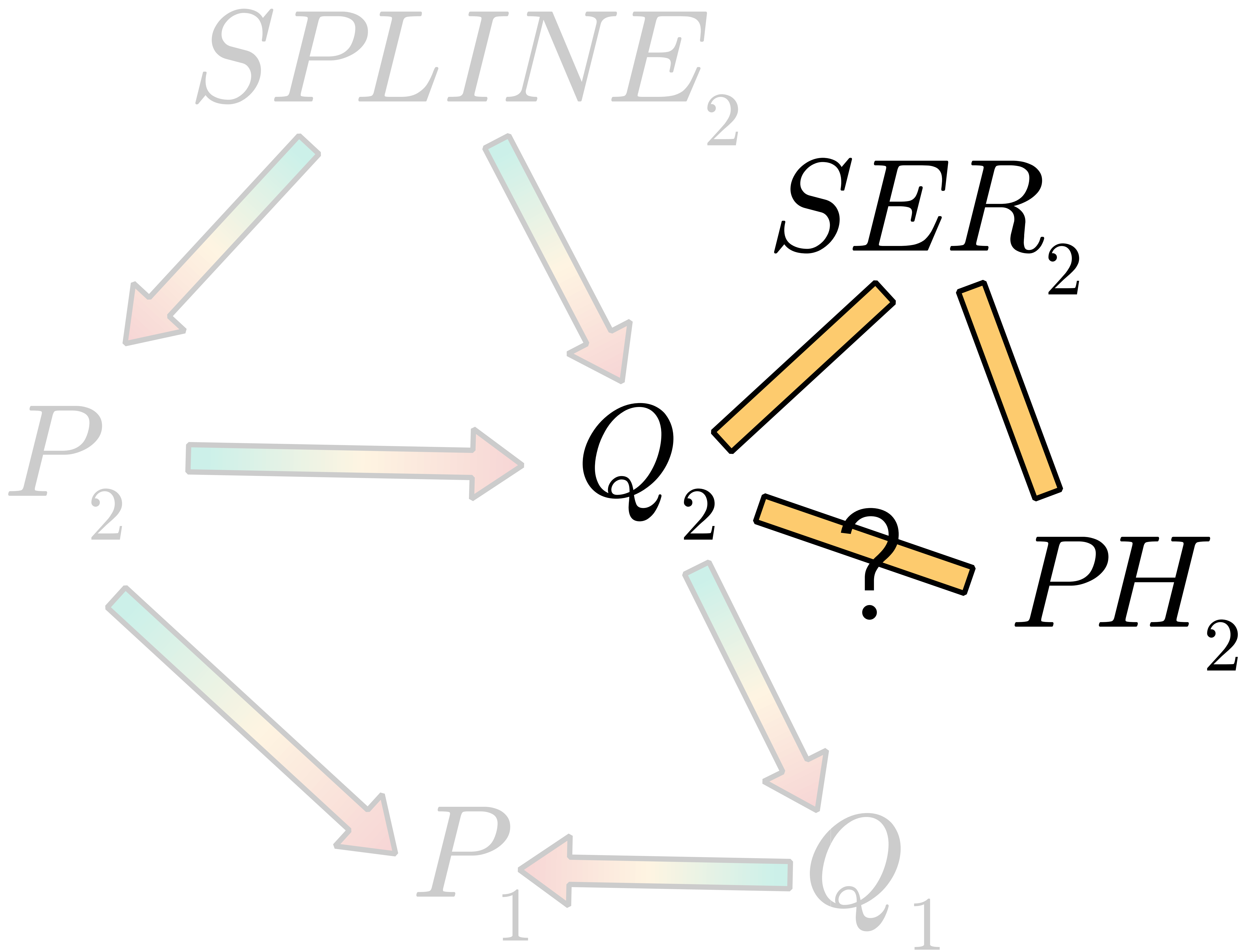
Element Summary



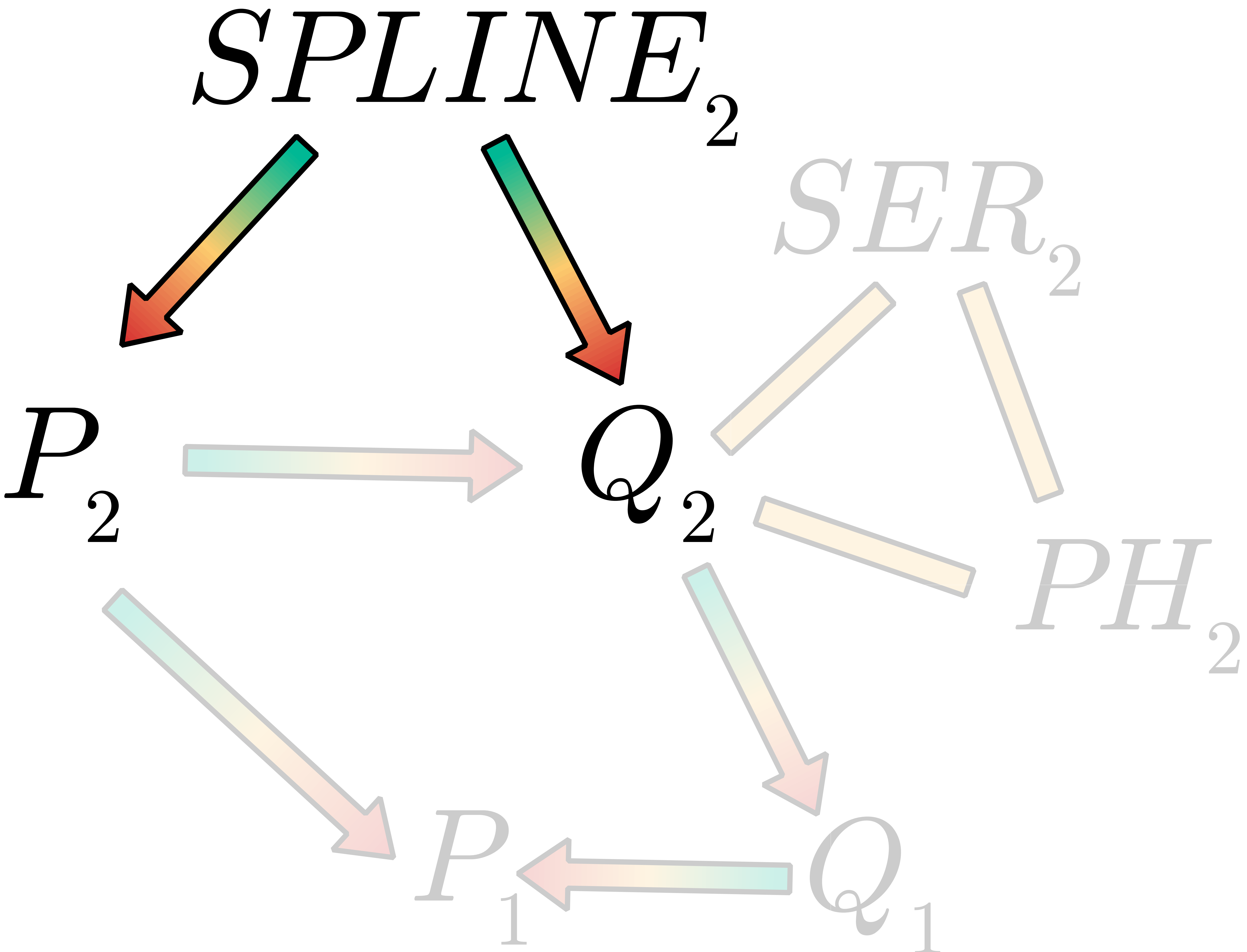
Element Summary



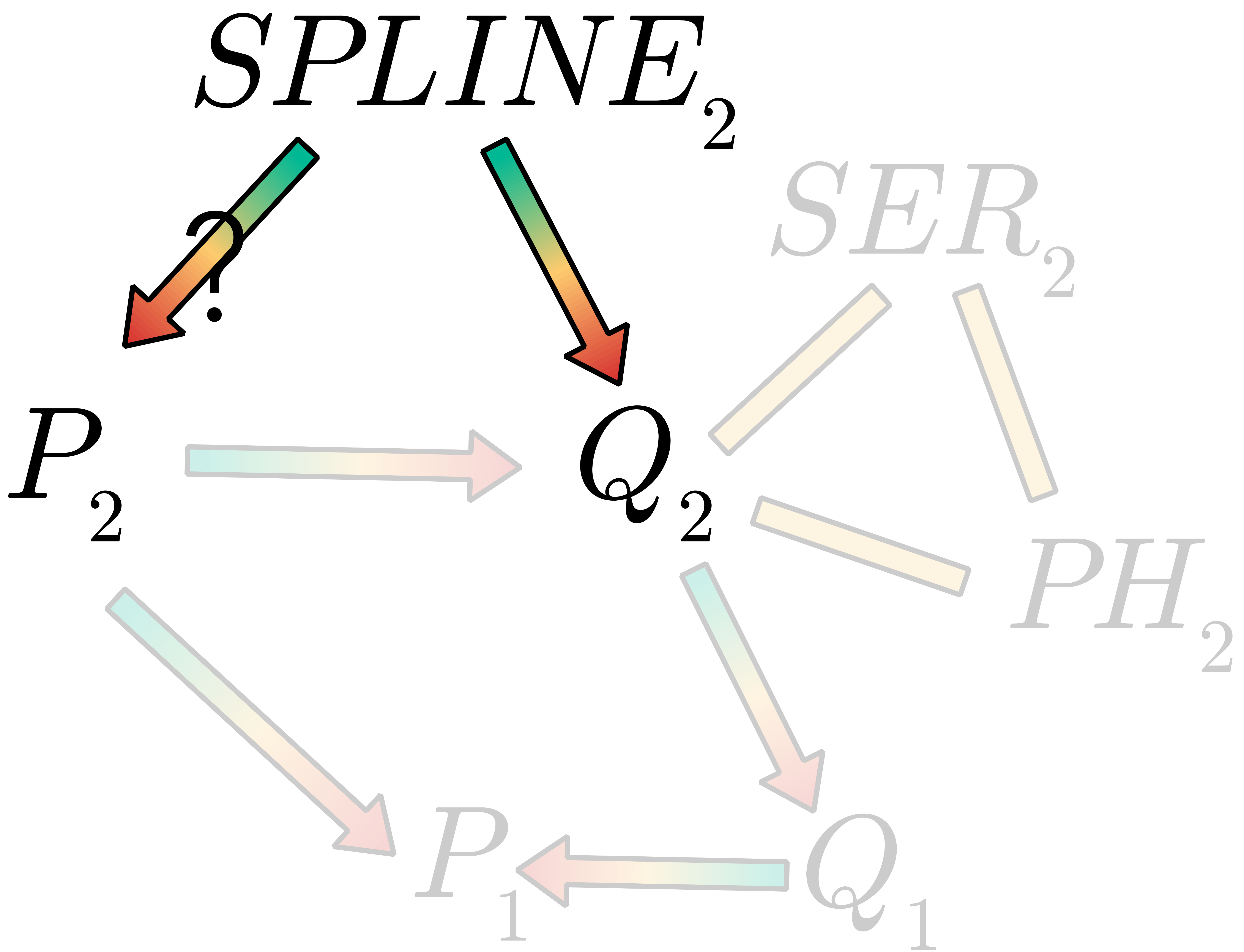
Element Summary



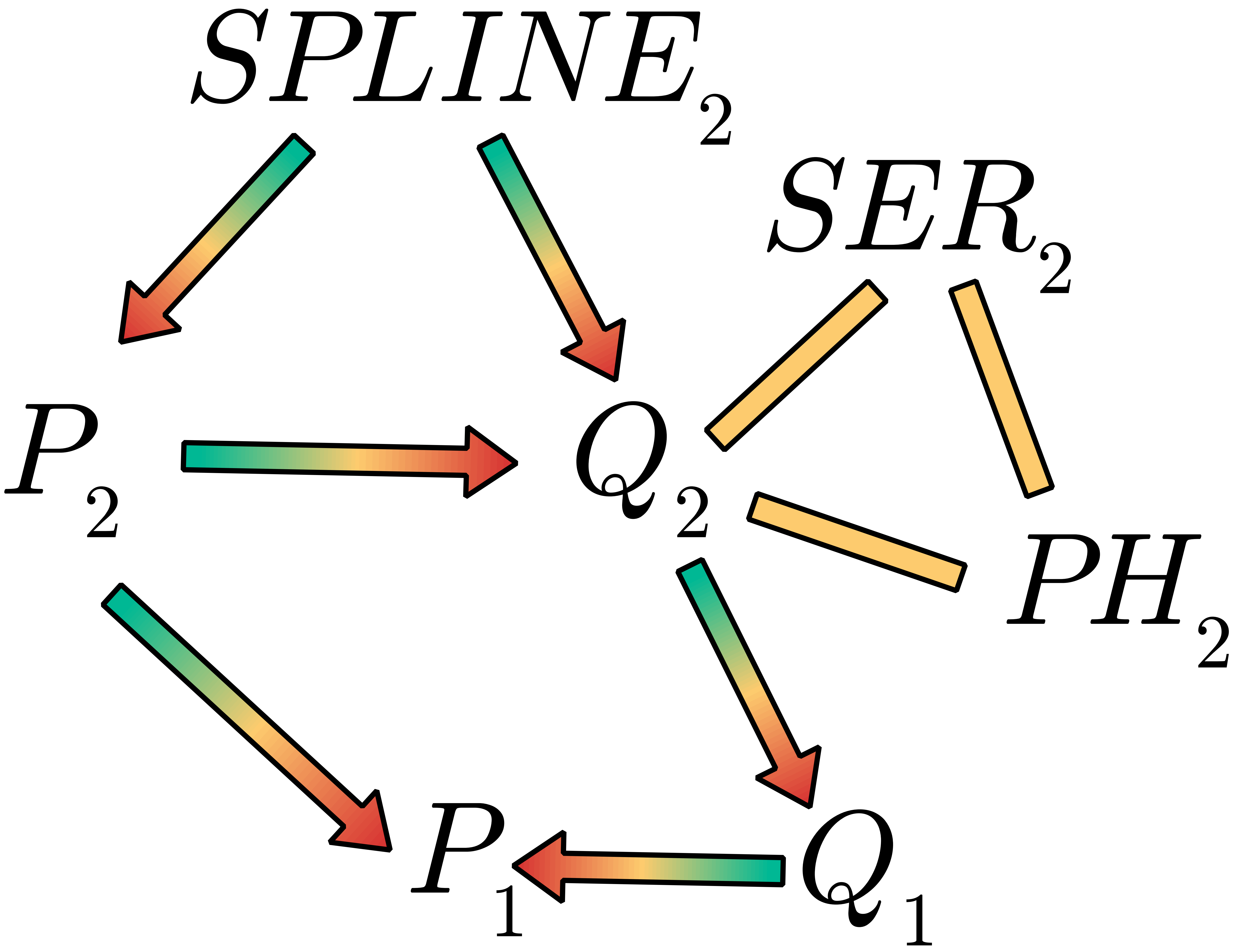
Element Summary



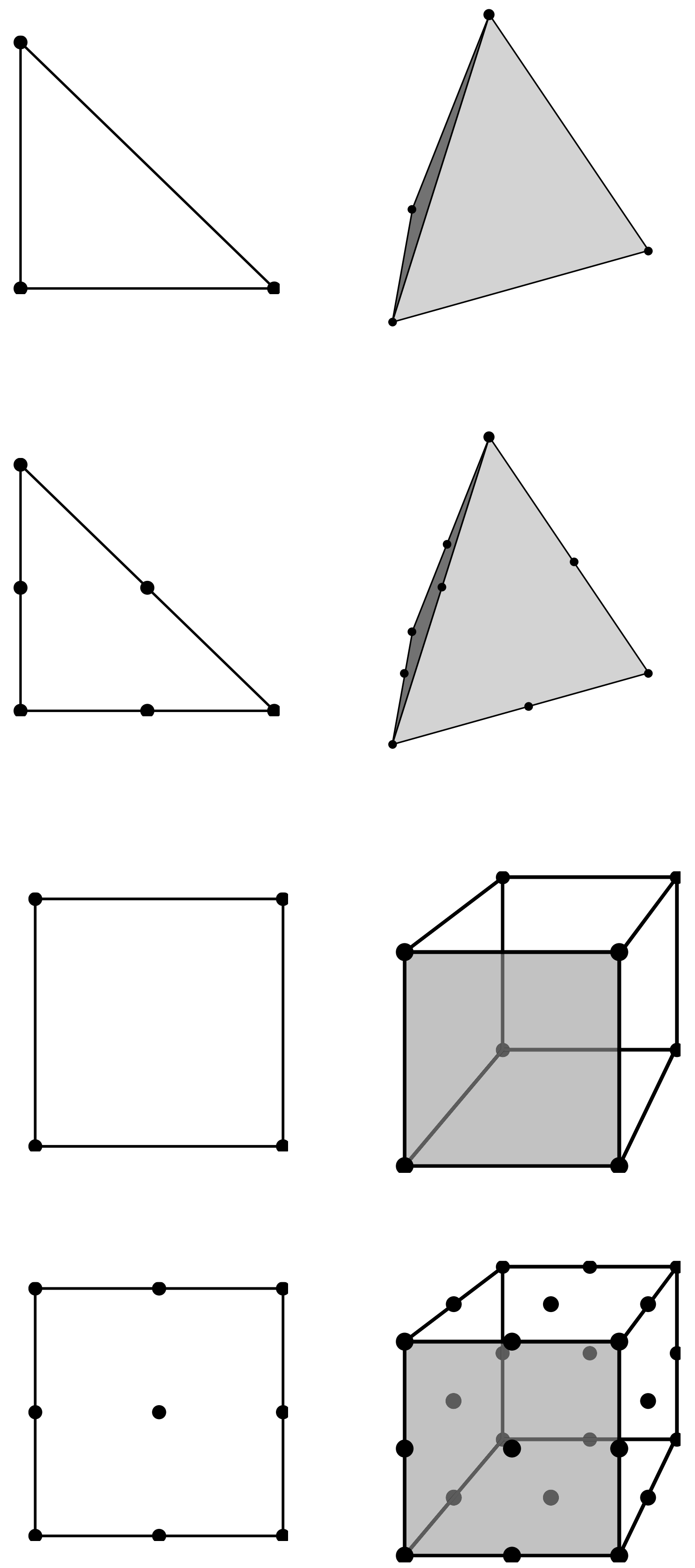
Element Summary



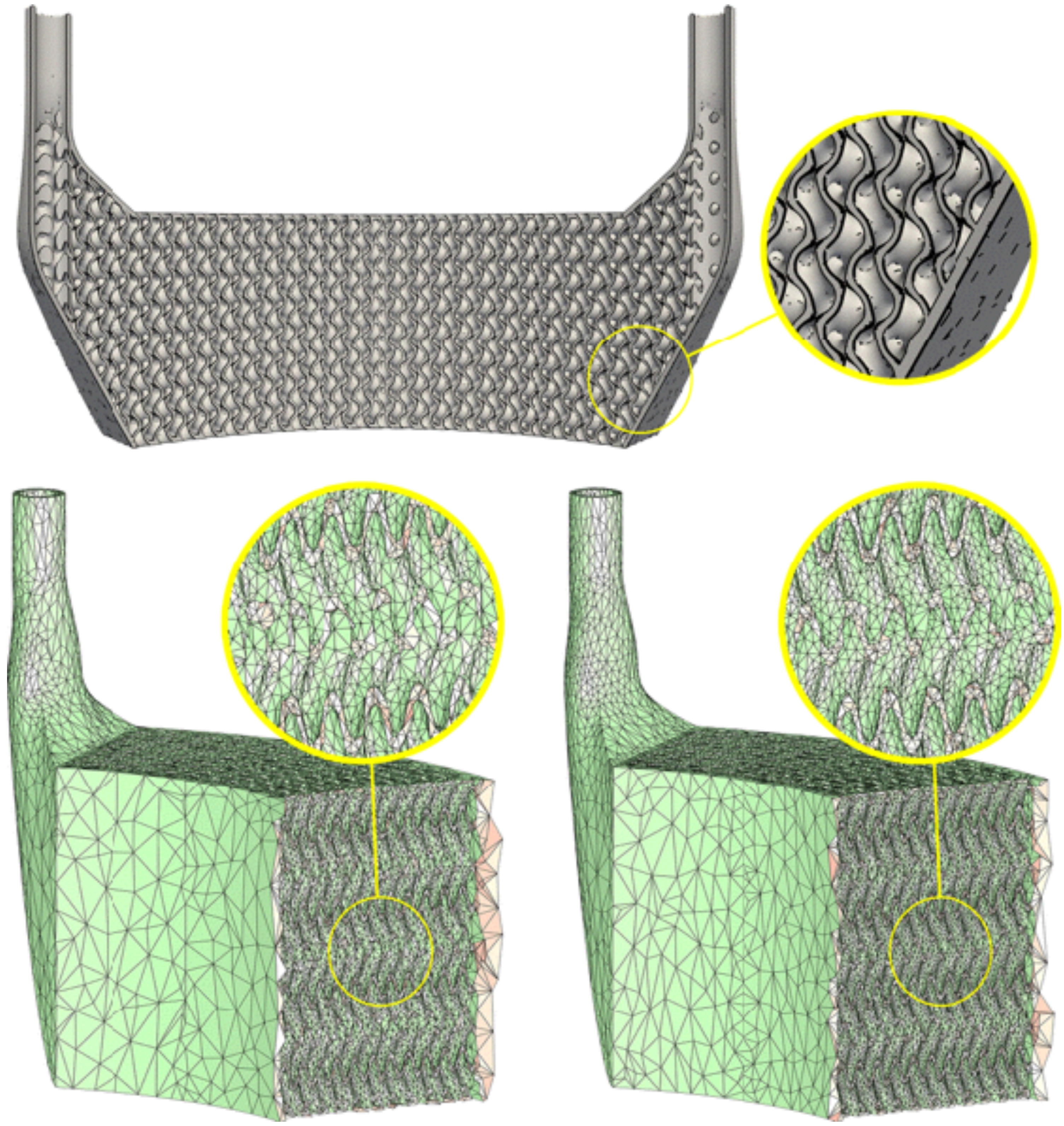
Element Summary



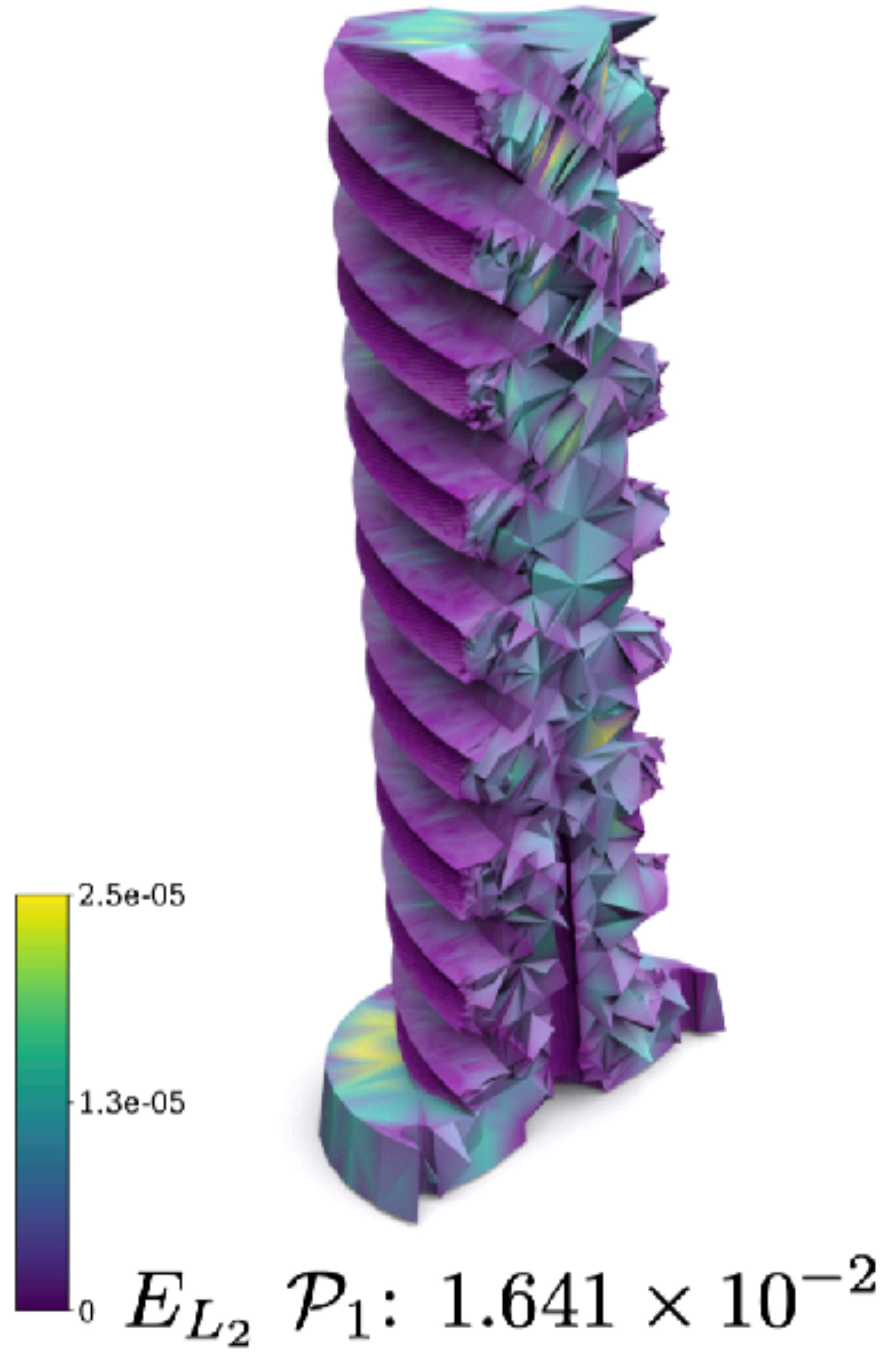
Overview



Which discretization provides lower running time for a fixed accuracy?

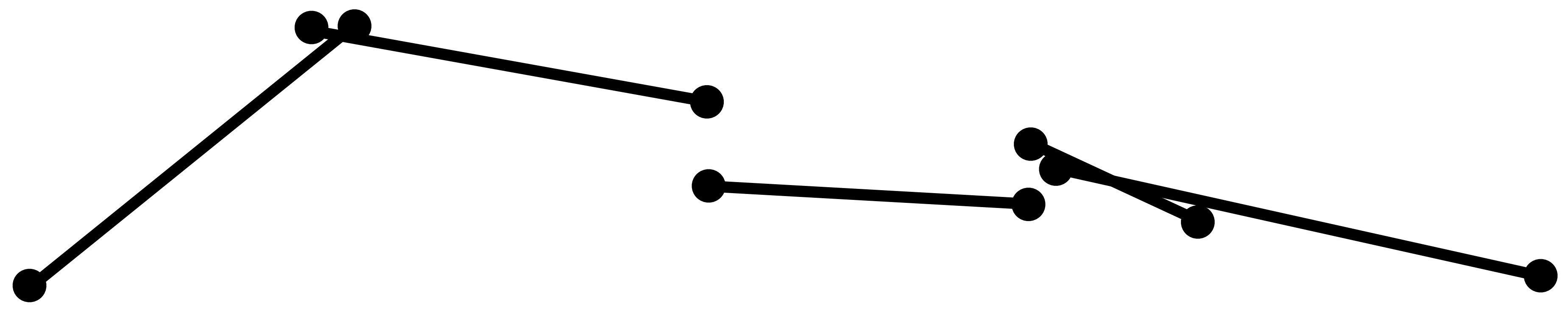


Can you mesh robustly without any assumption on the input?

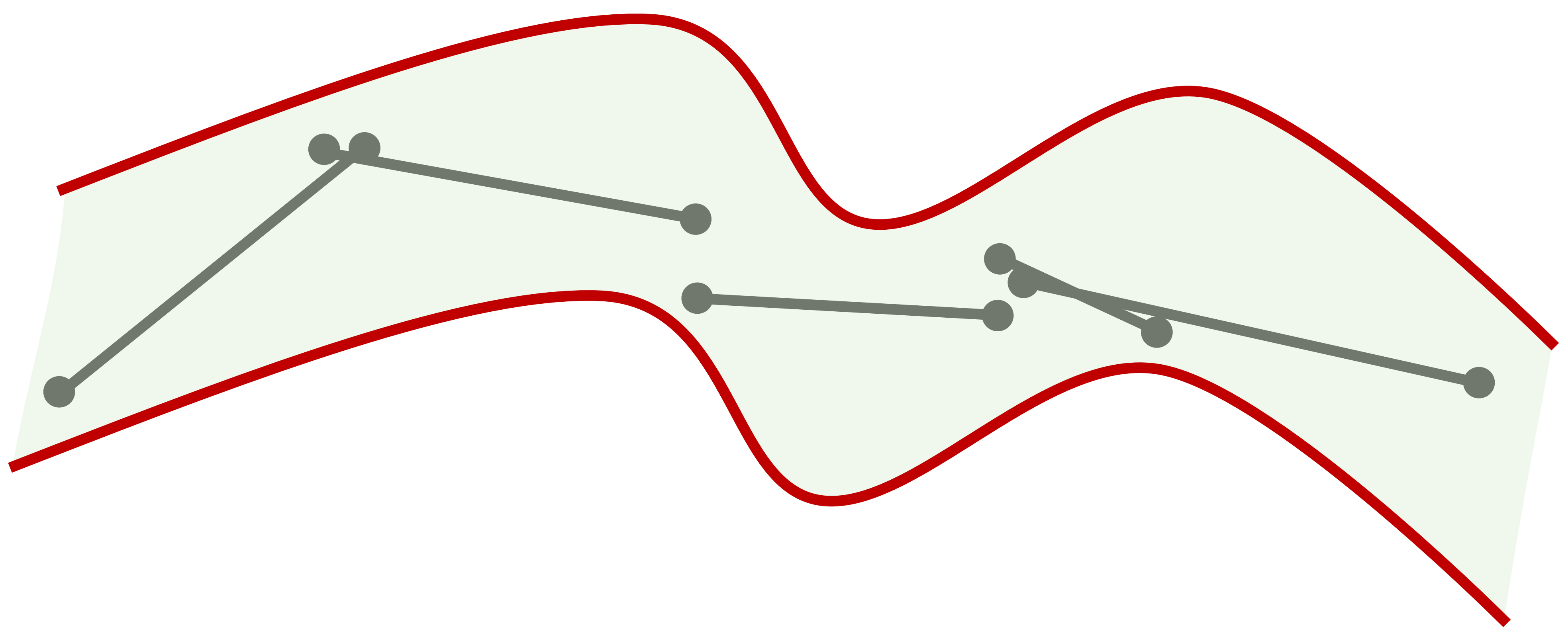


Does mesh quality affect the accuracy of the FEM solution?

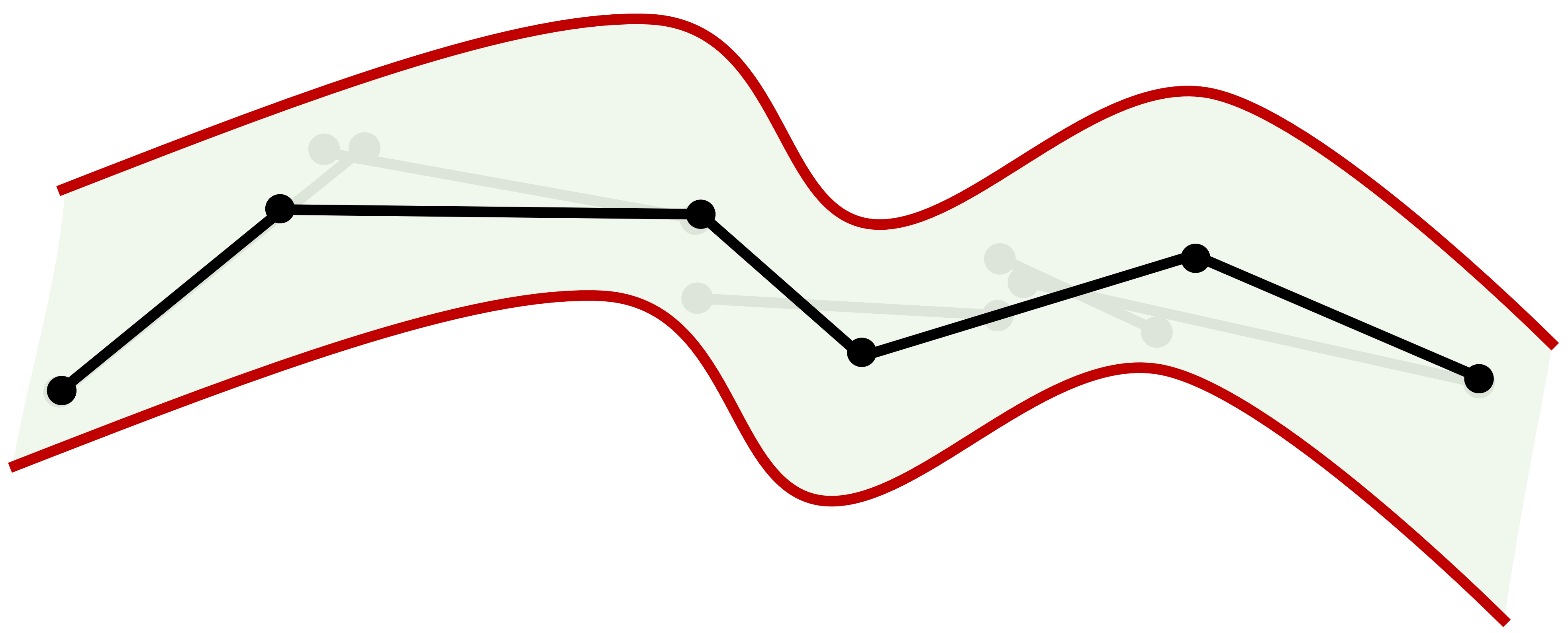
Envelope



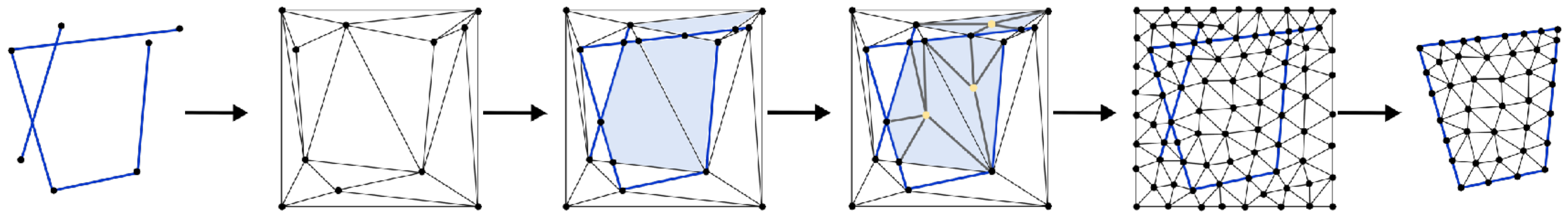
Envelope



Envelope

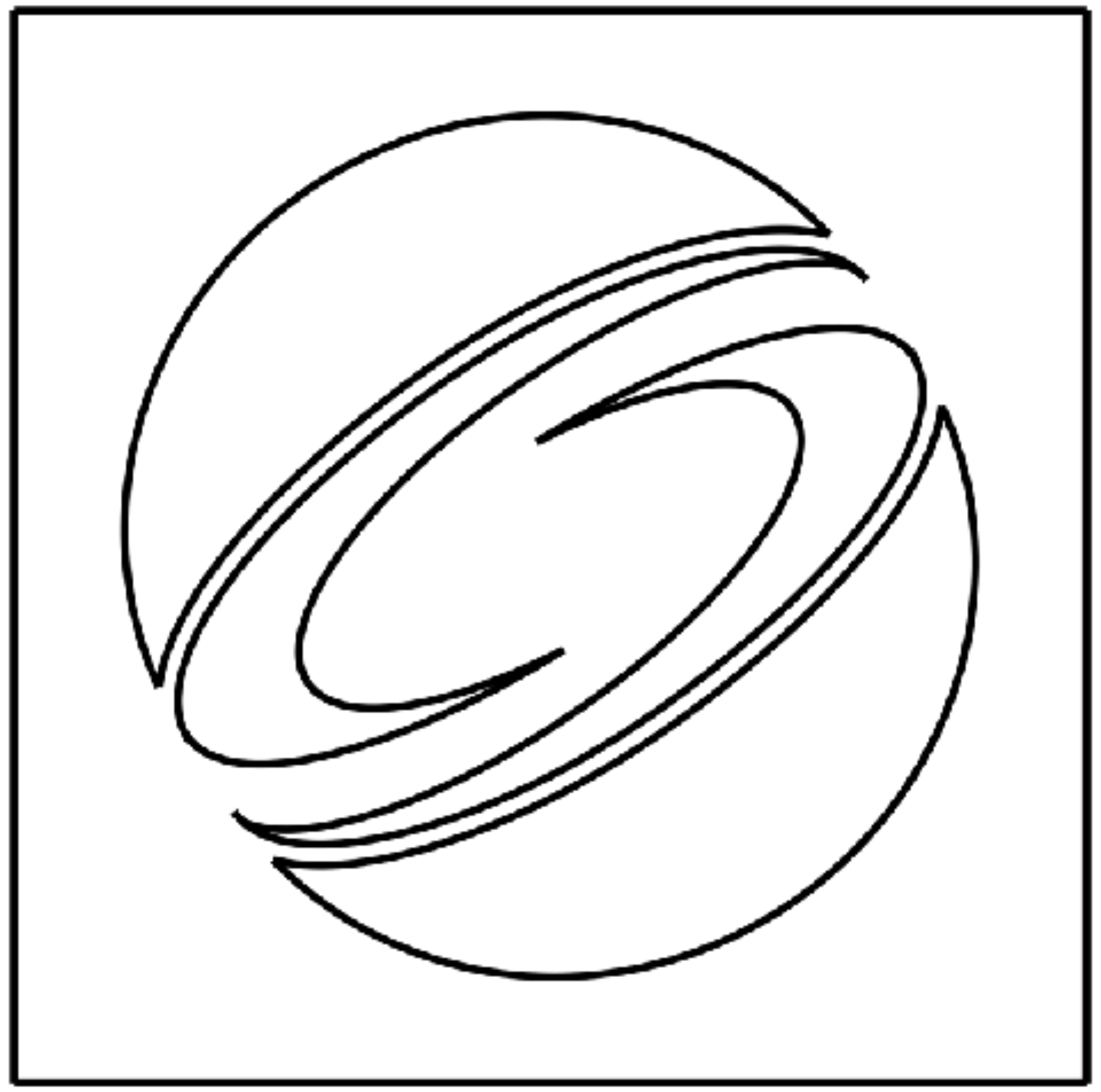


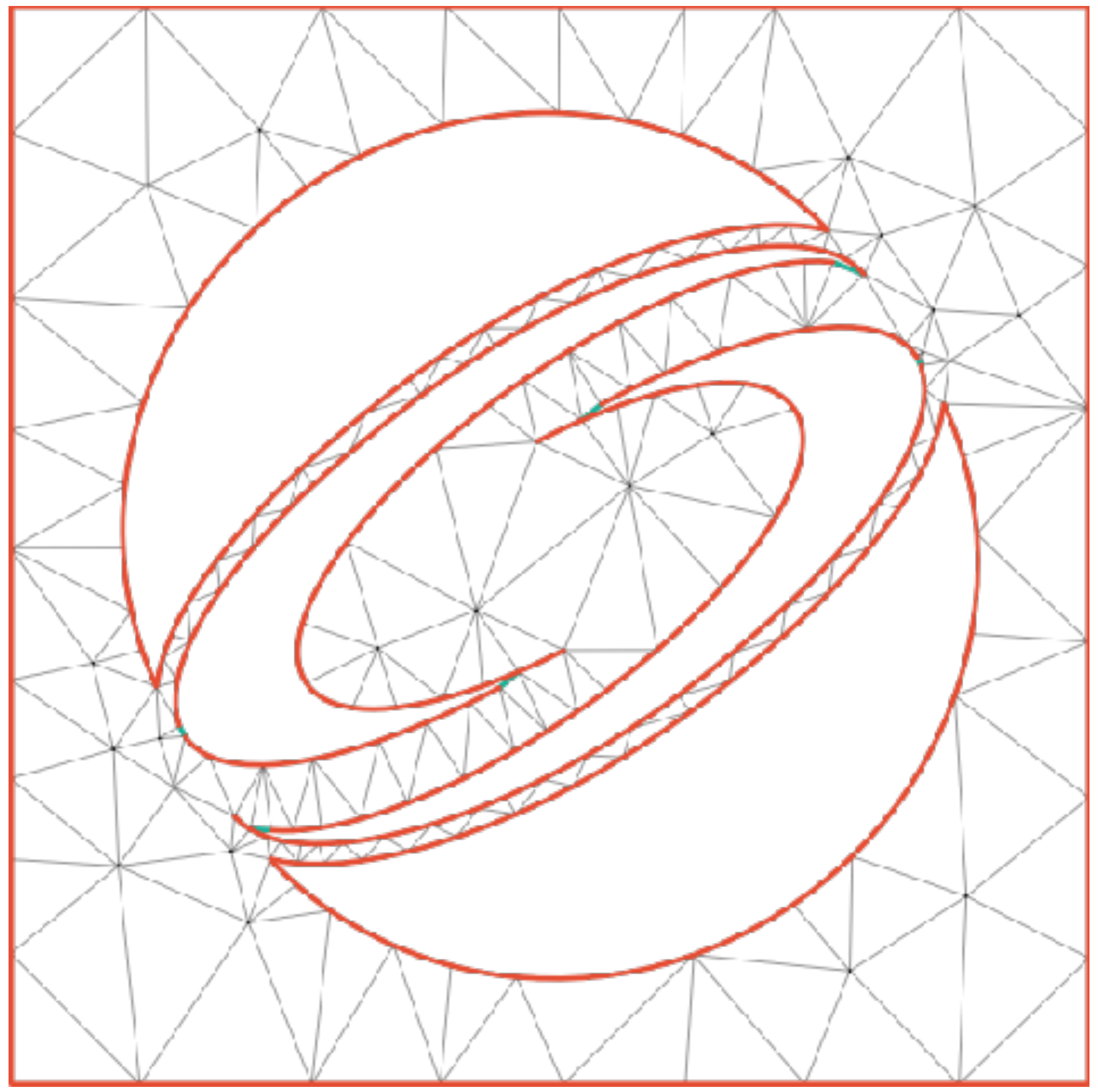
Fast Triangulation in the Wild

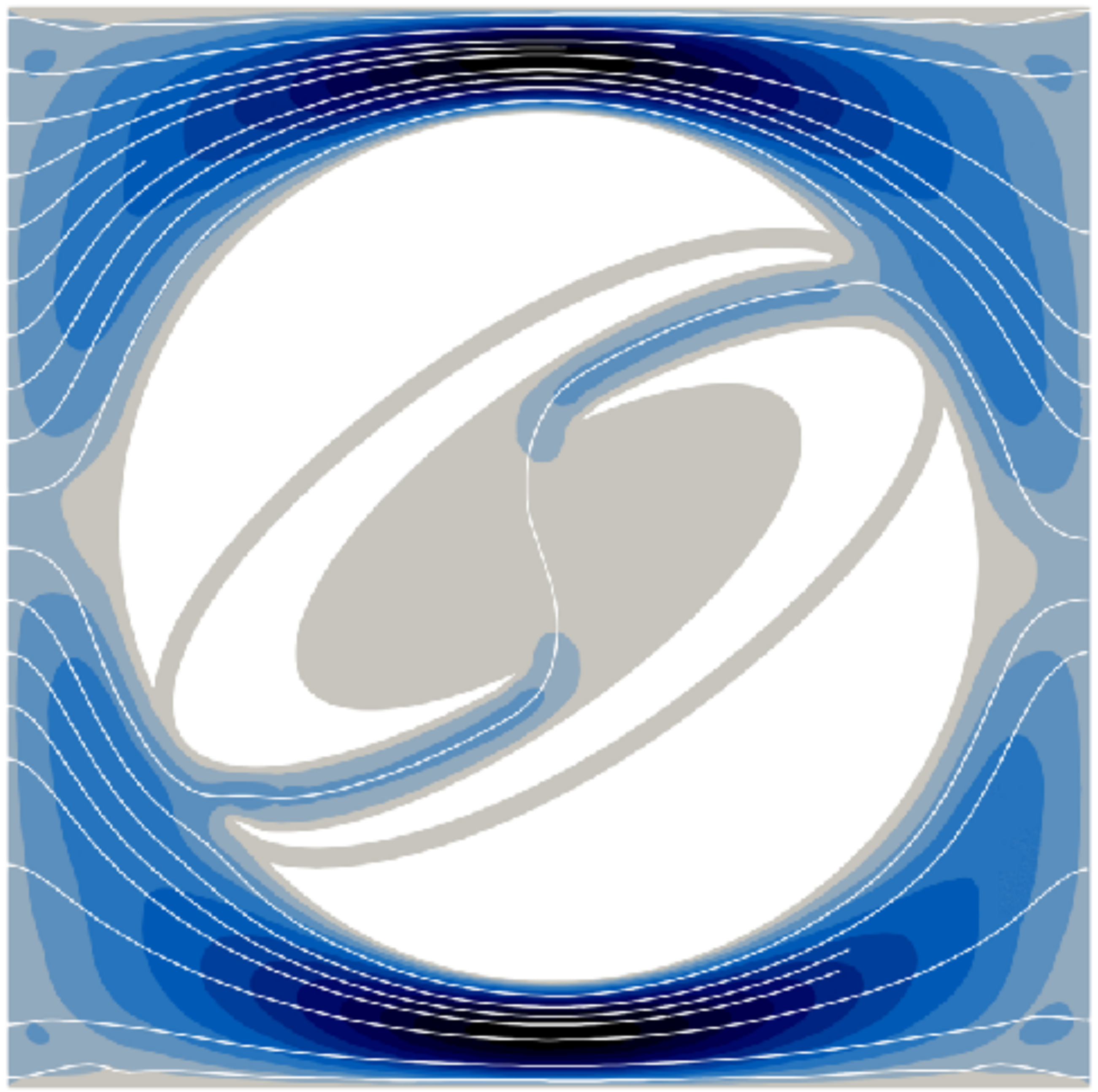


Input Initial mesh Hybrid mesh Valid mesh Optimized mesh Output

2D Triangulation







Input 2D Boundary

Coarser → Faster

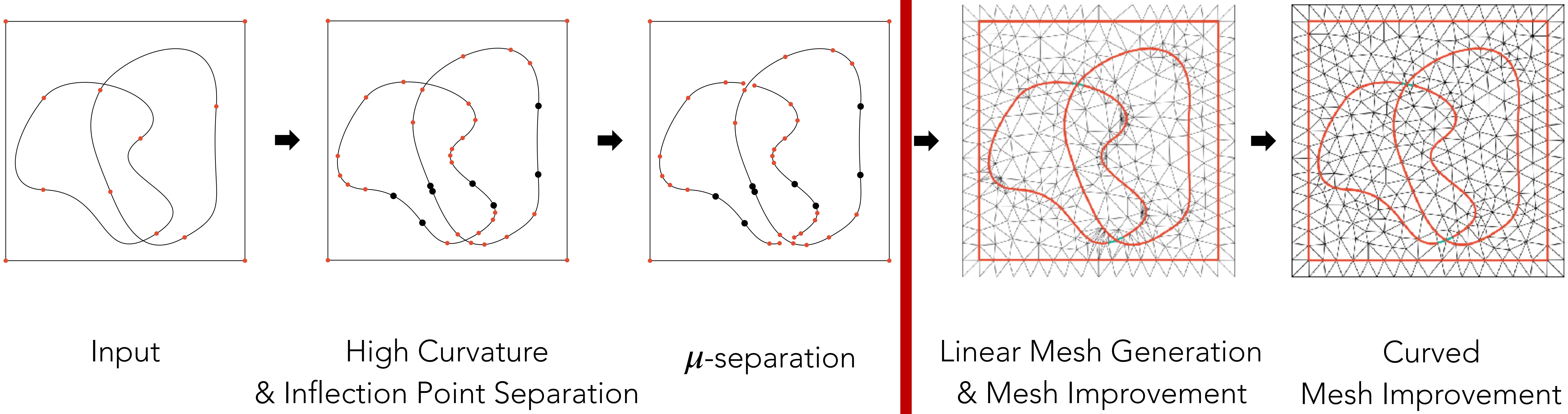
Output Triangle Mesh

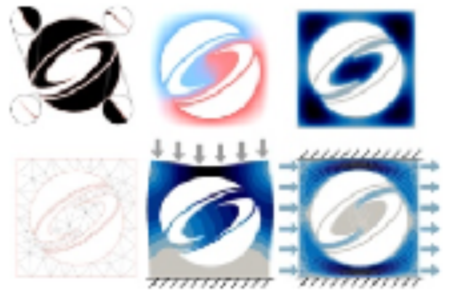
Physical Simulation

Conforming → Accurate

Curved 2D Triangulation: TriWild

"Cleanup" the input curves.

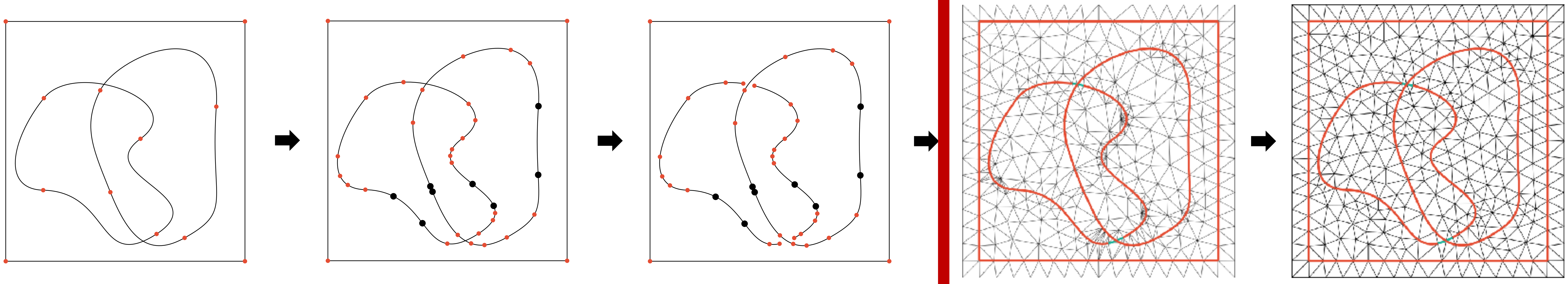




TriWild: Robust Triangulation with Curve Constraints
Yixin Hu, Teseo Schneider, Xifeng Gao, Qingnan Zhou, Alec Jacobson, Denis Zorin, Daniele Panozzo,
ACM Transaction on Graphics (SIGGRAPH), 2019
[Paper] [Code] [Data]

TriWild

Linear mesh for easier curving.



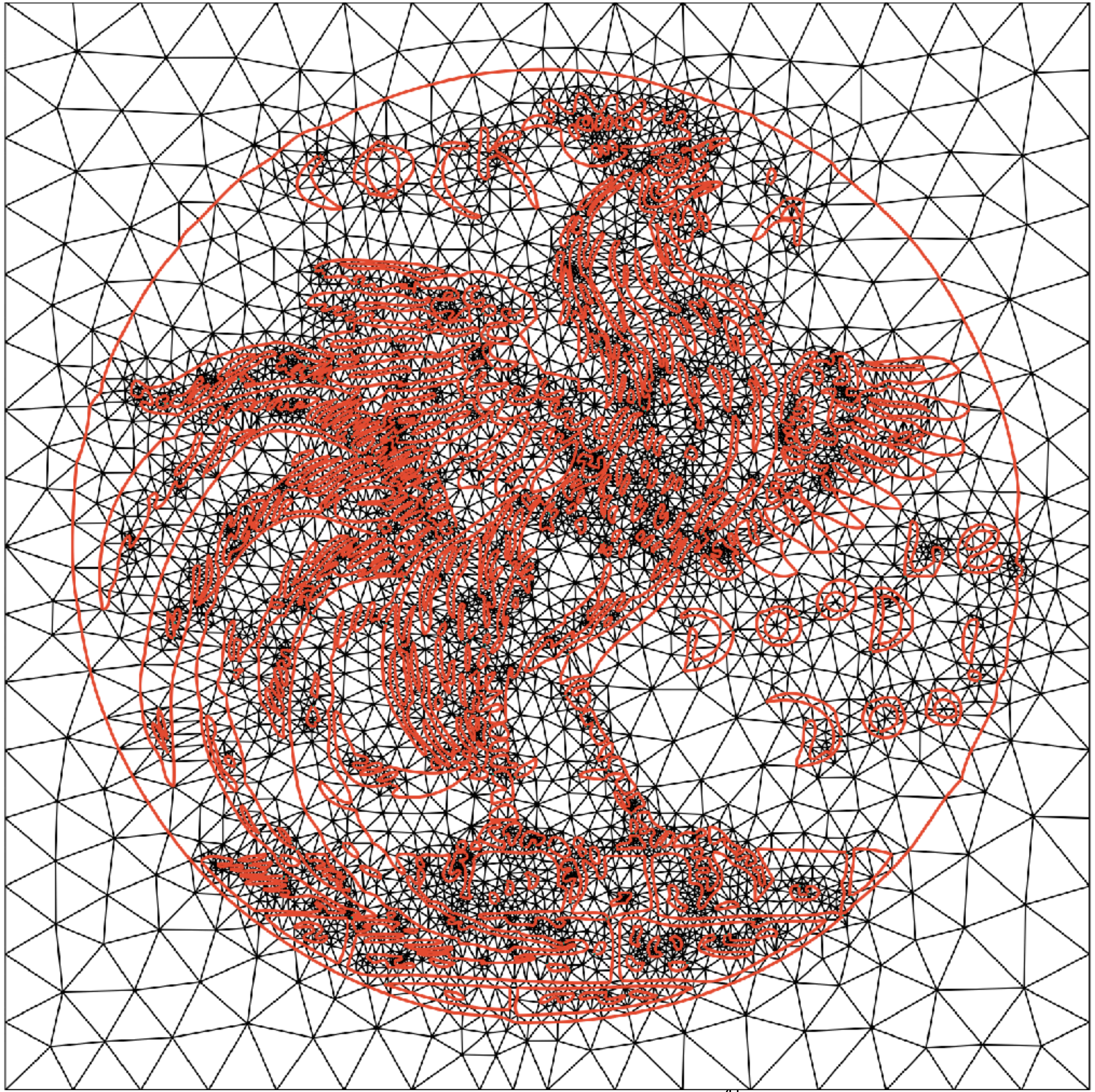
Input

High Curvature
& Inflection Point Separation

μ-separation

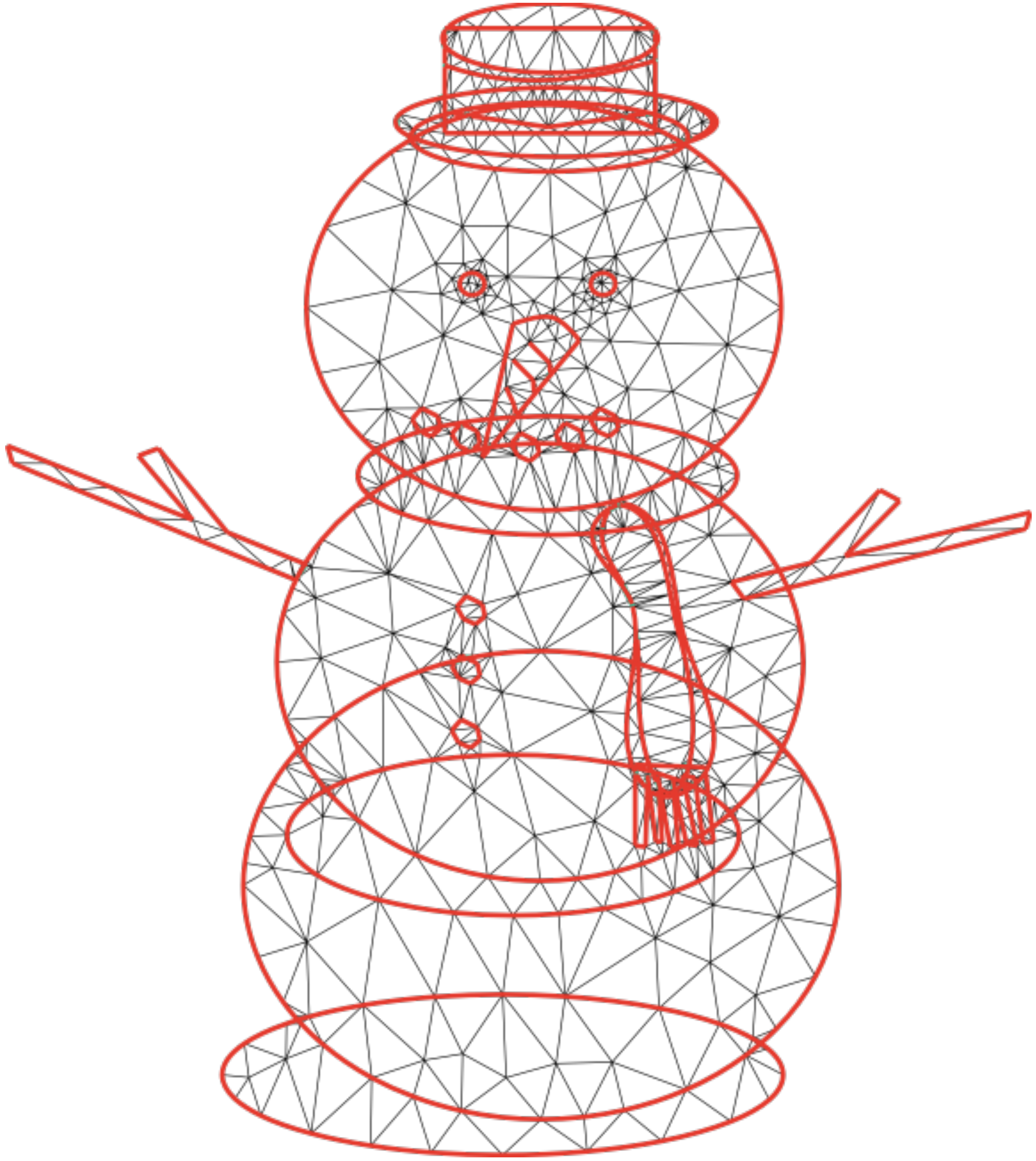
Linear Mesh Generation
& Mesh Improvement

Curved
Mesh Improvement

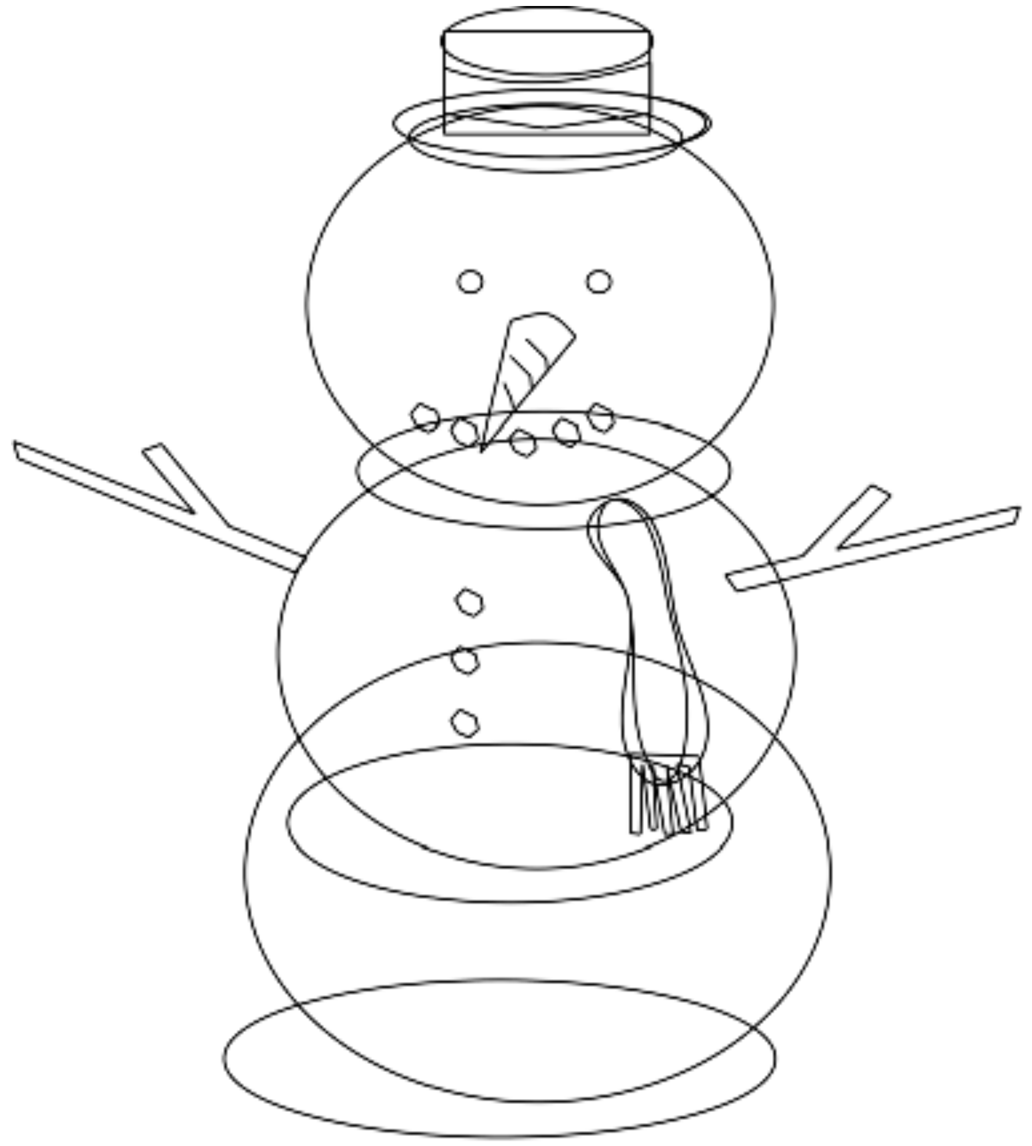


Input:

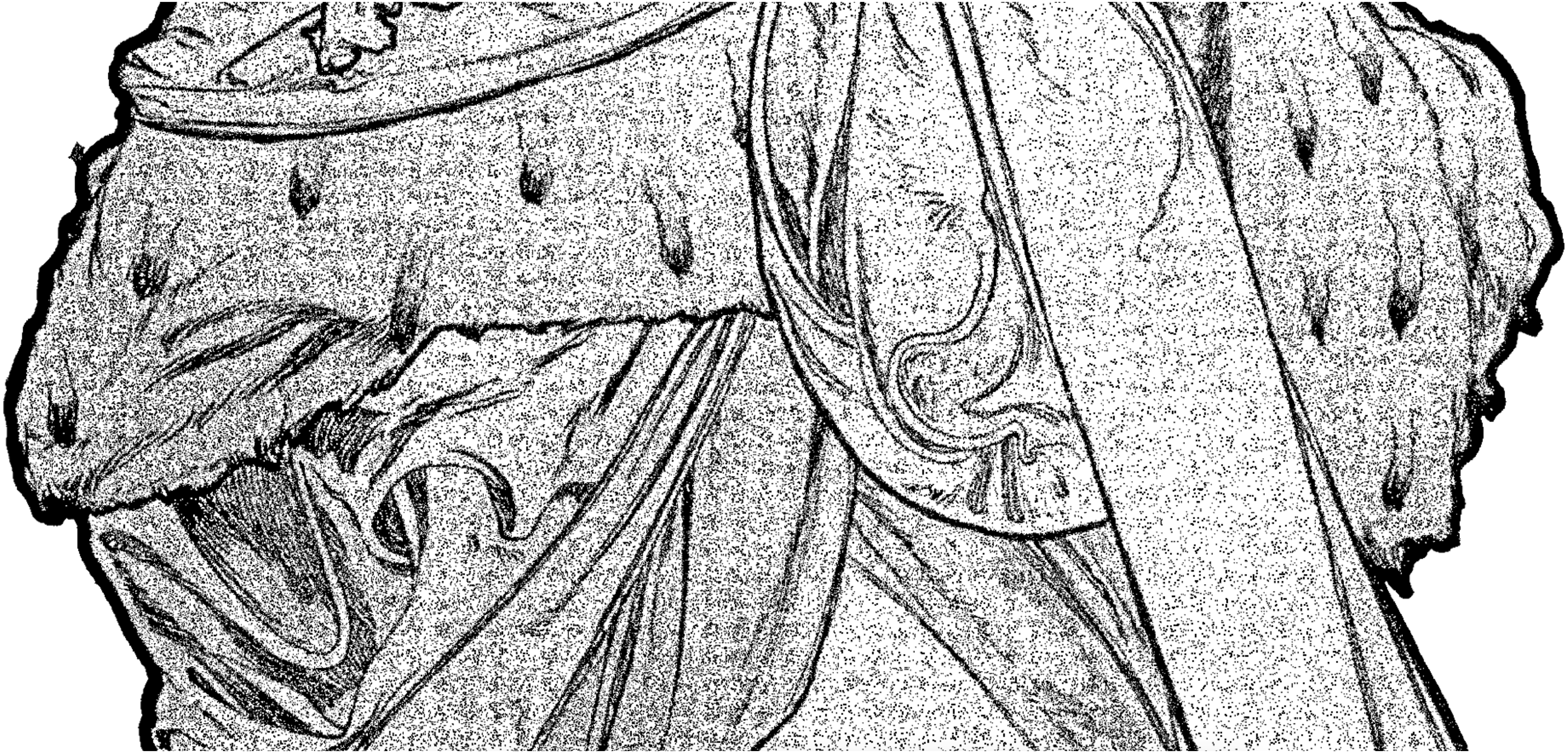
(Generated by TriWild)

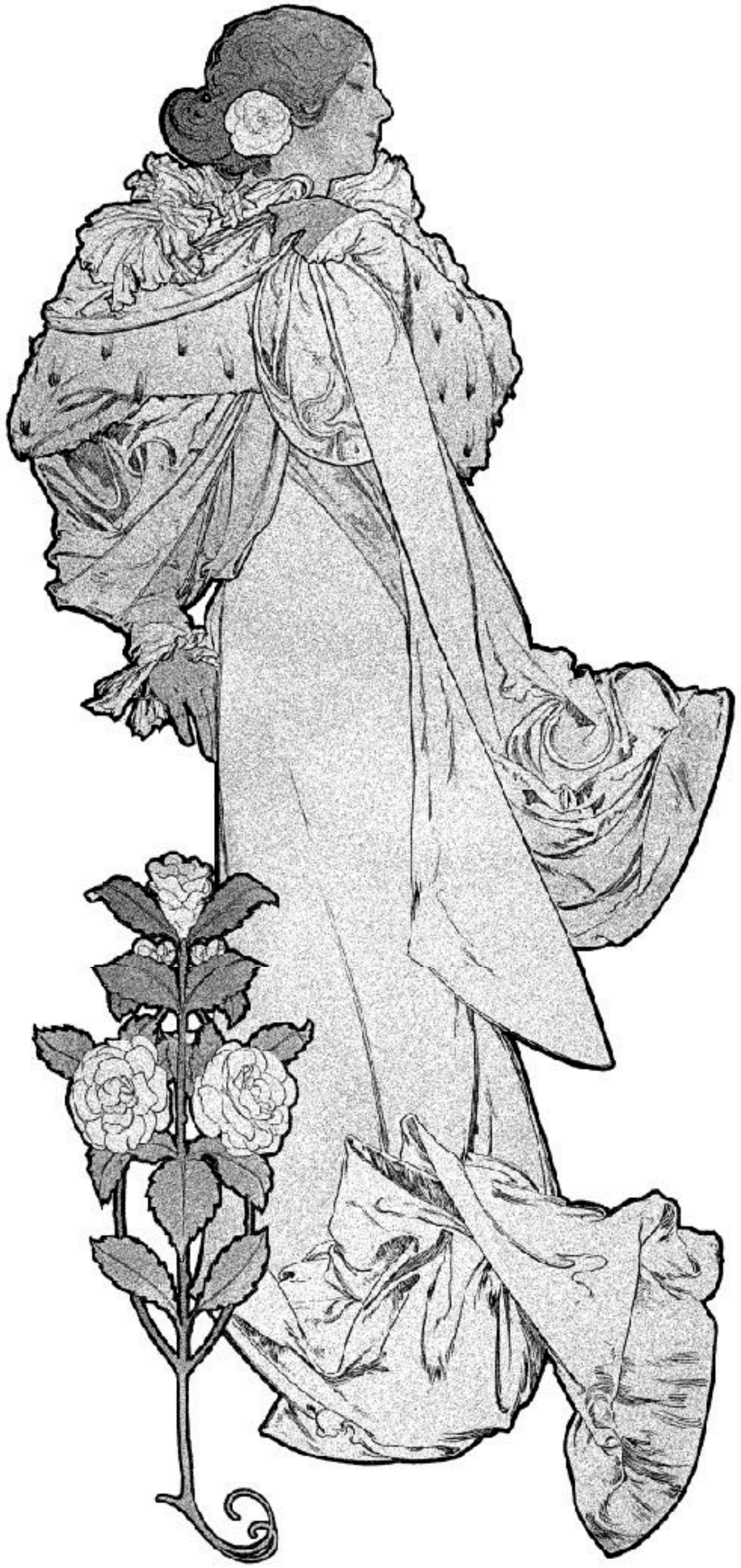


Input:

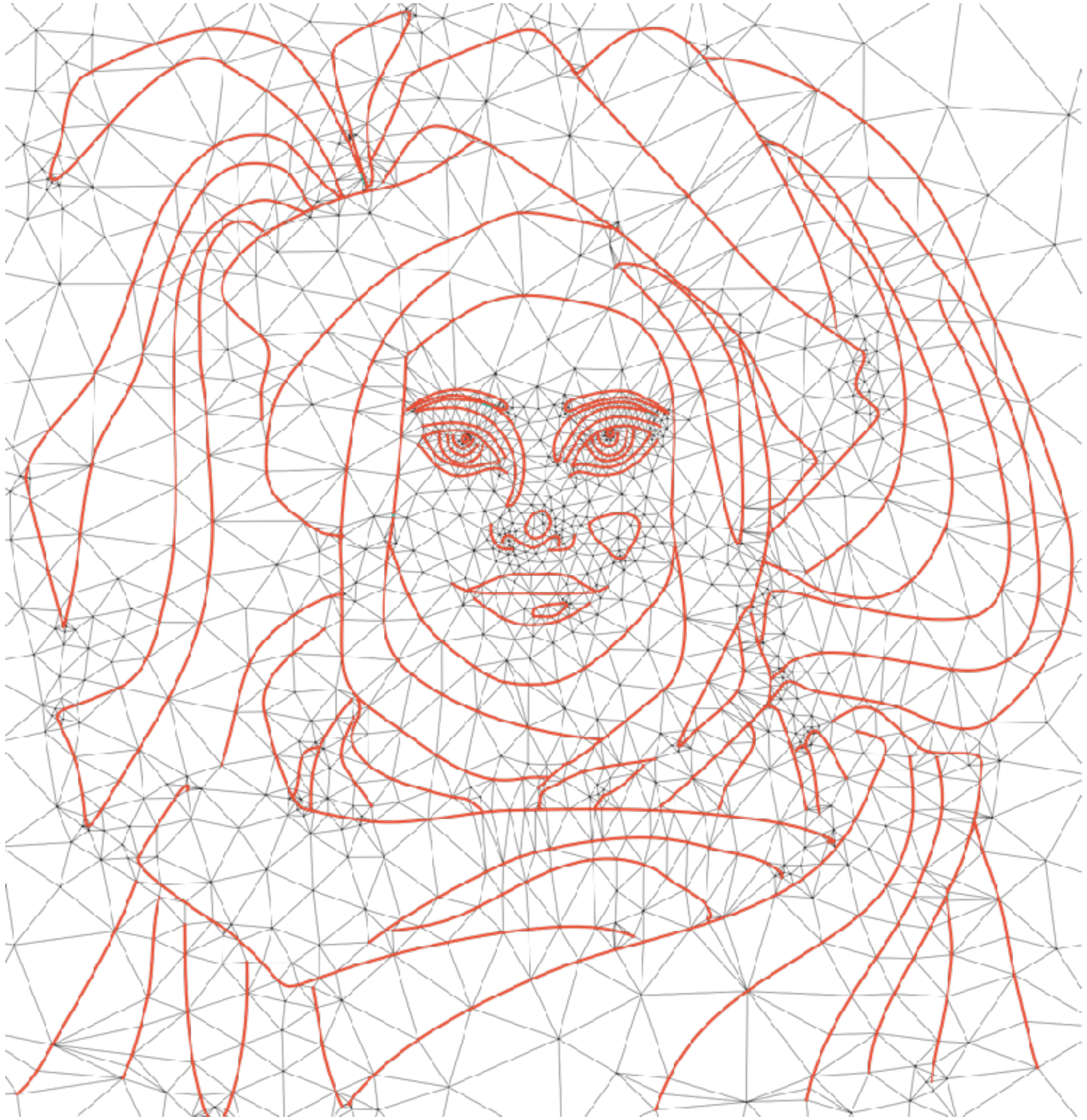


(Generated by TriWild)



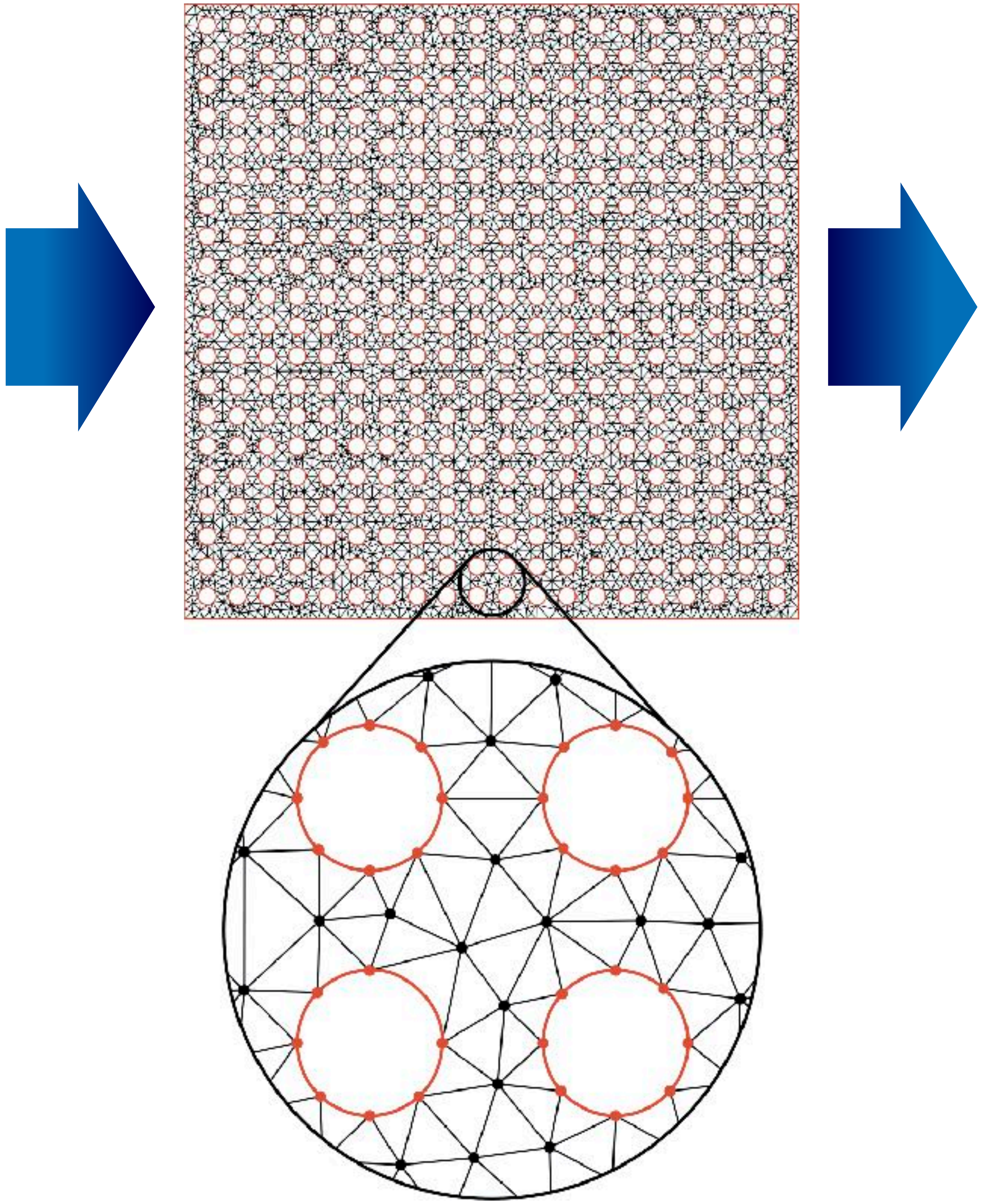


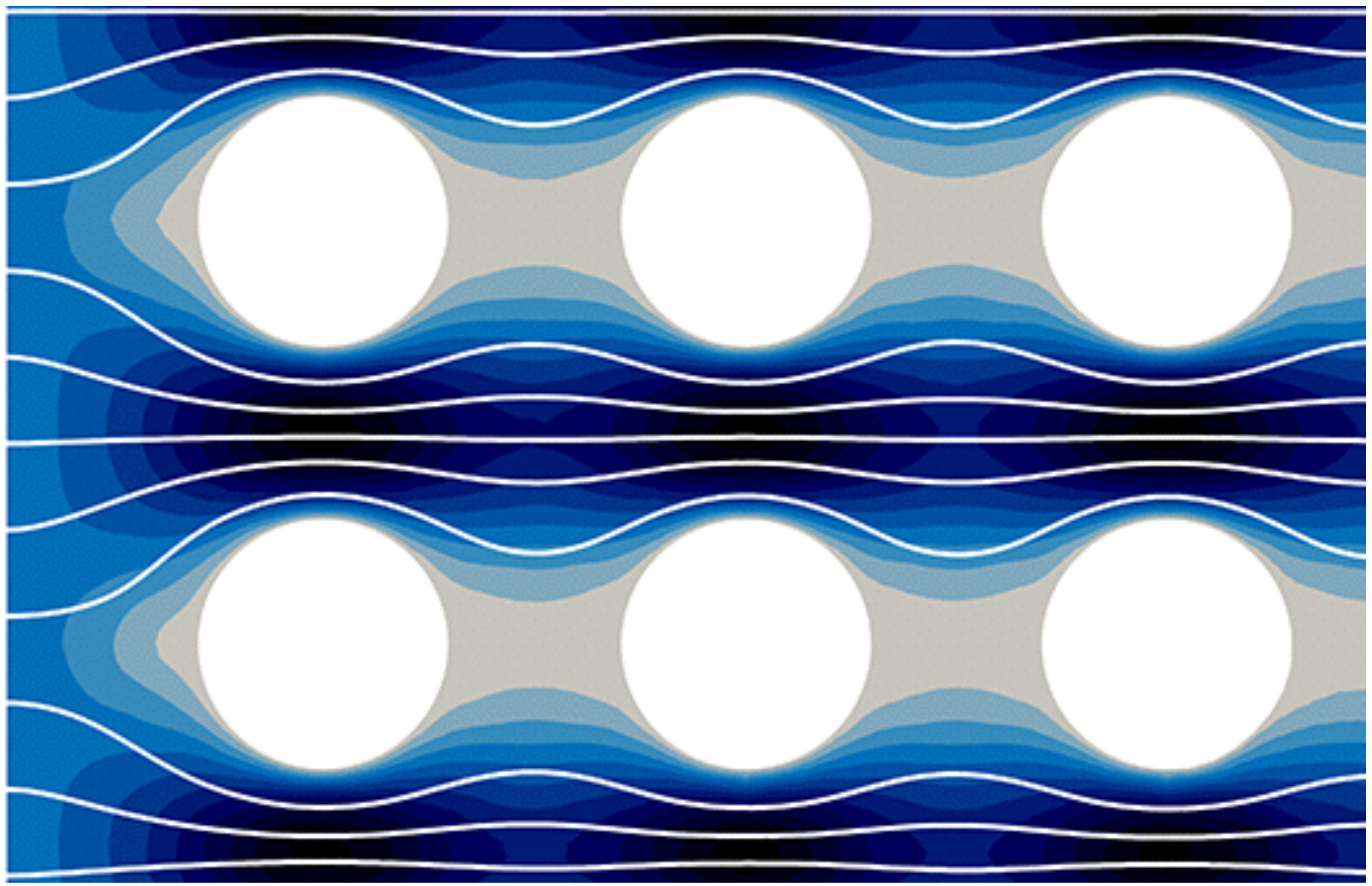
Application – Diffusion Curves



Application — Diffusion Curve

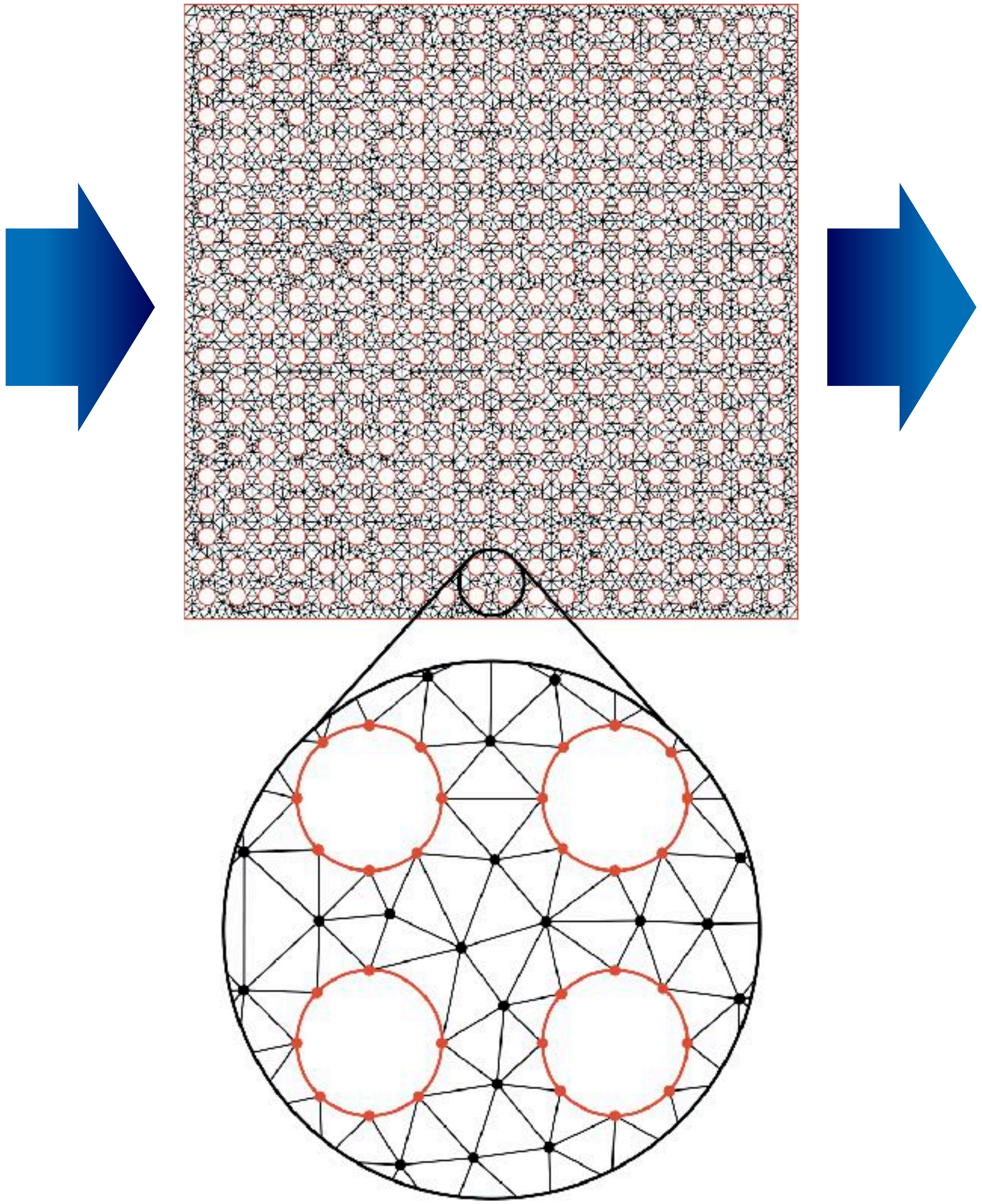
Application – Stokes

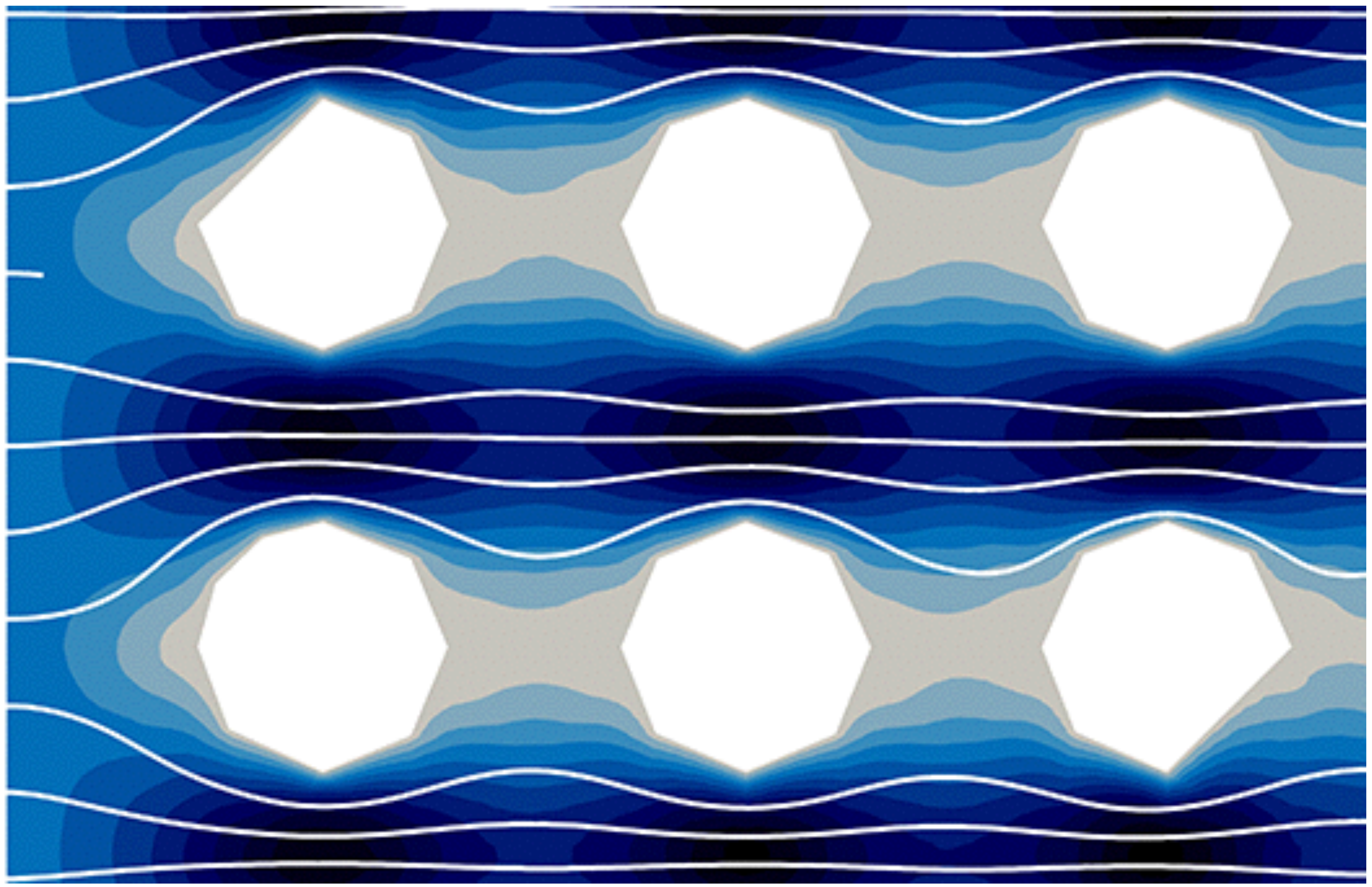




Using Curved Mesh

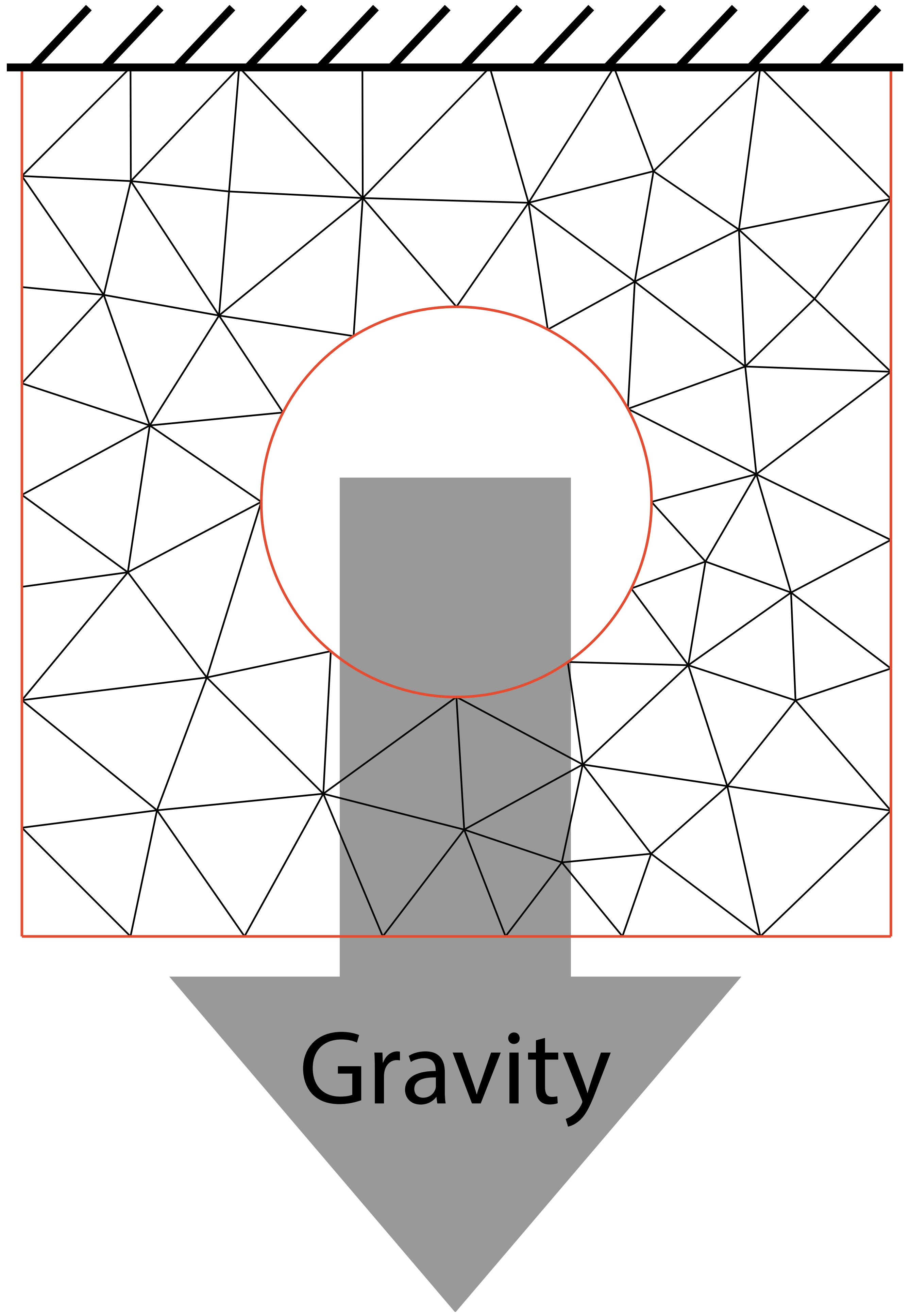
Application – Stokes





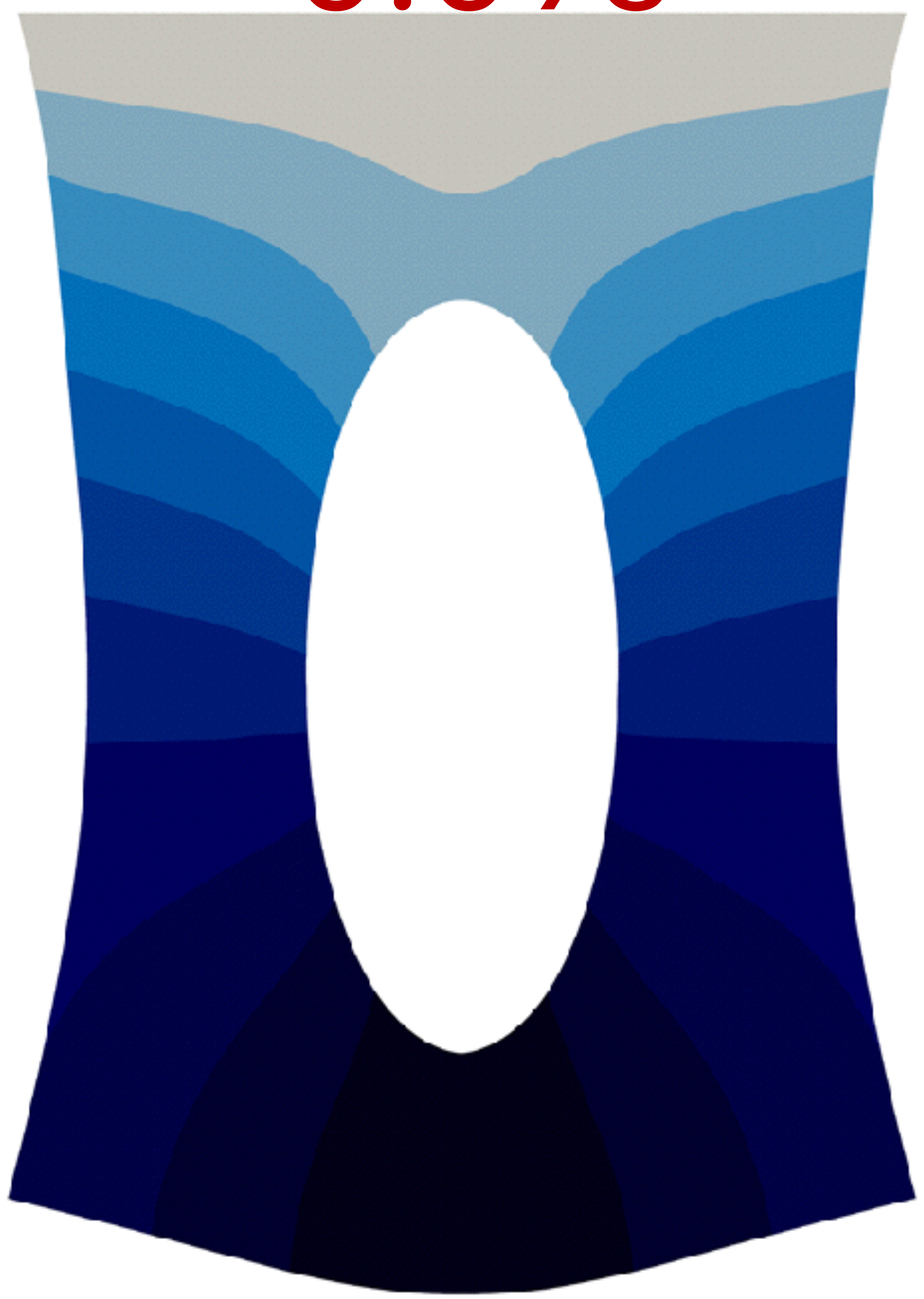
Using Linear Mesh

Application – Elasticity



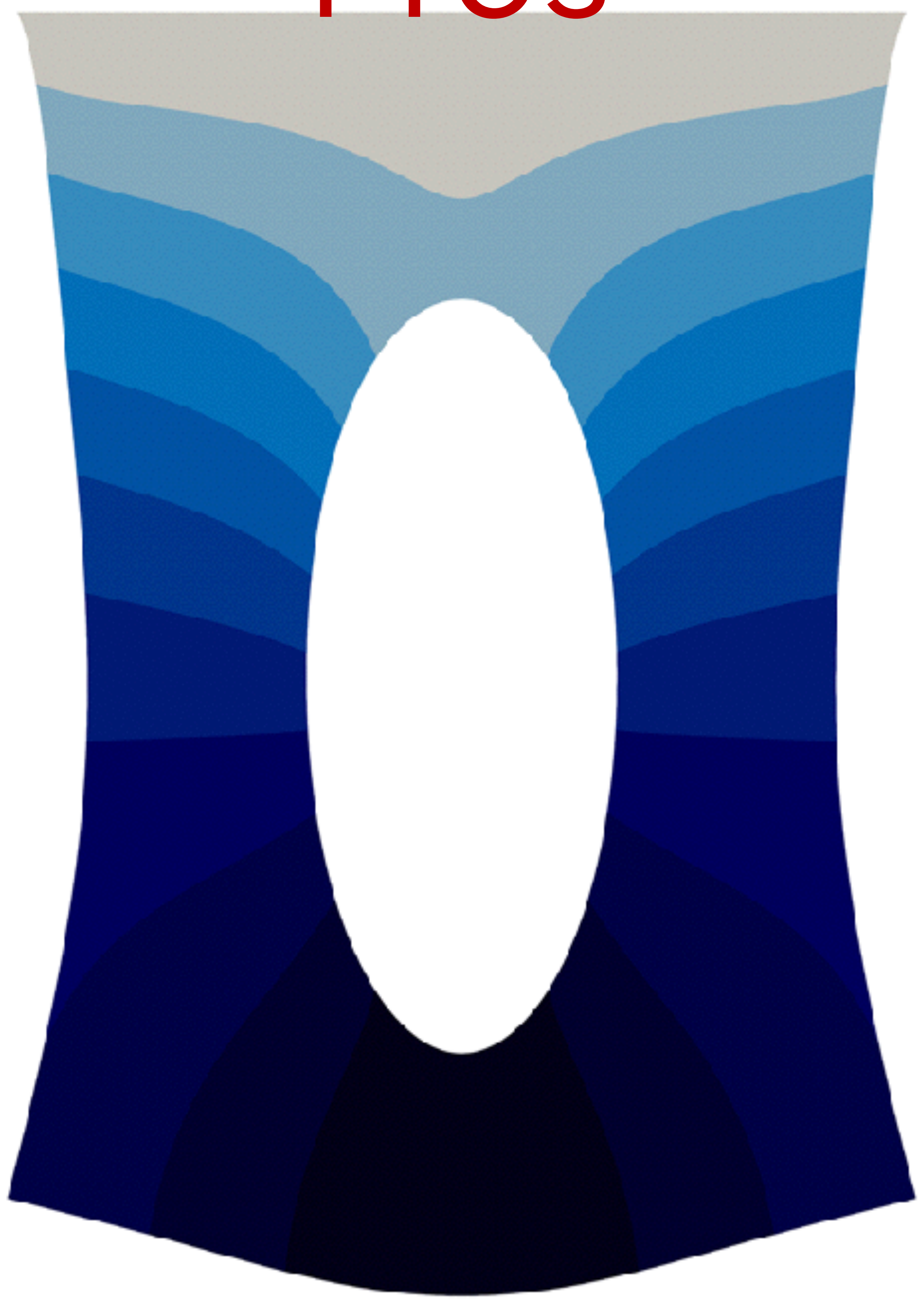
Curved Mesh

0.09s



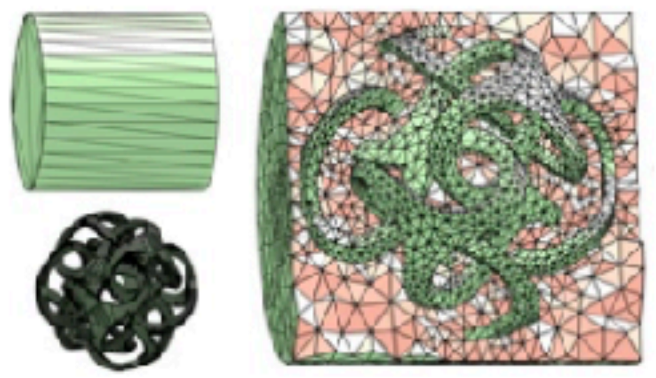
Using Curved Mesh

115s

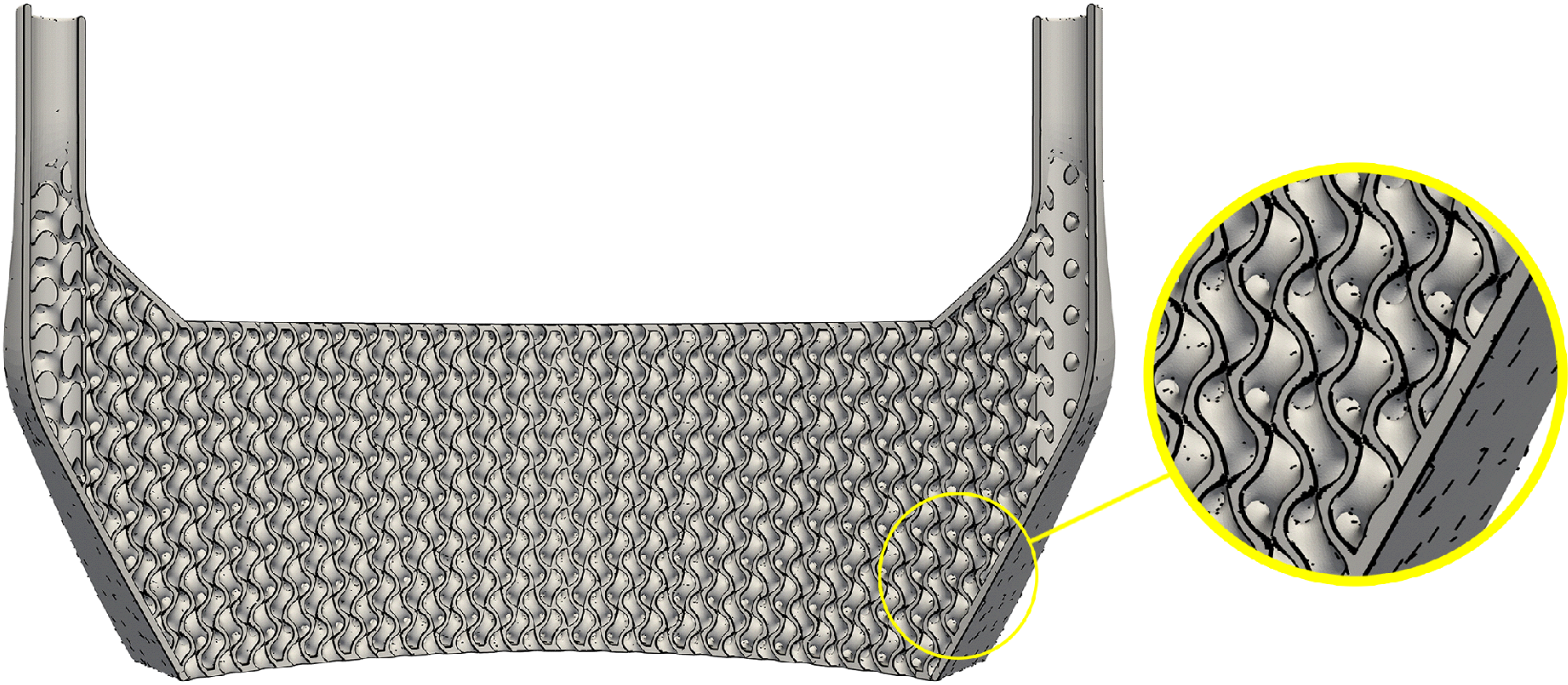


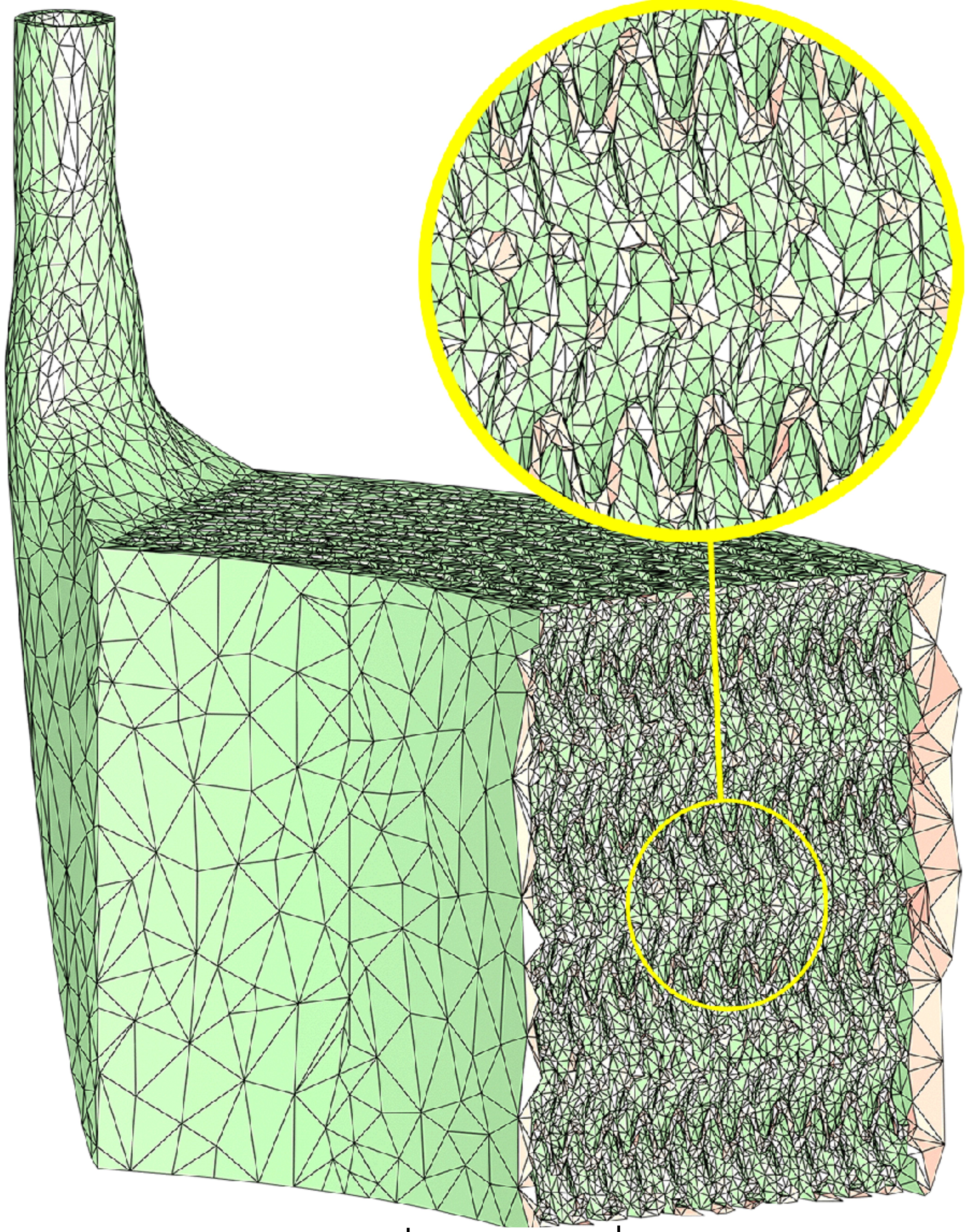
Using **Dense Linear** Mesh

Fast Tetrahedral Meshing in the Wild

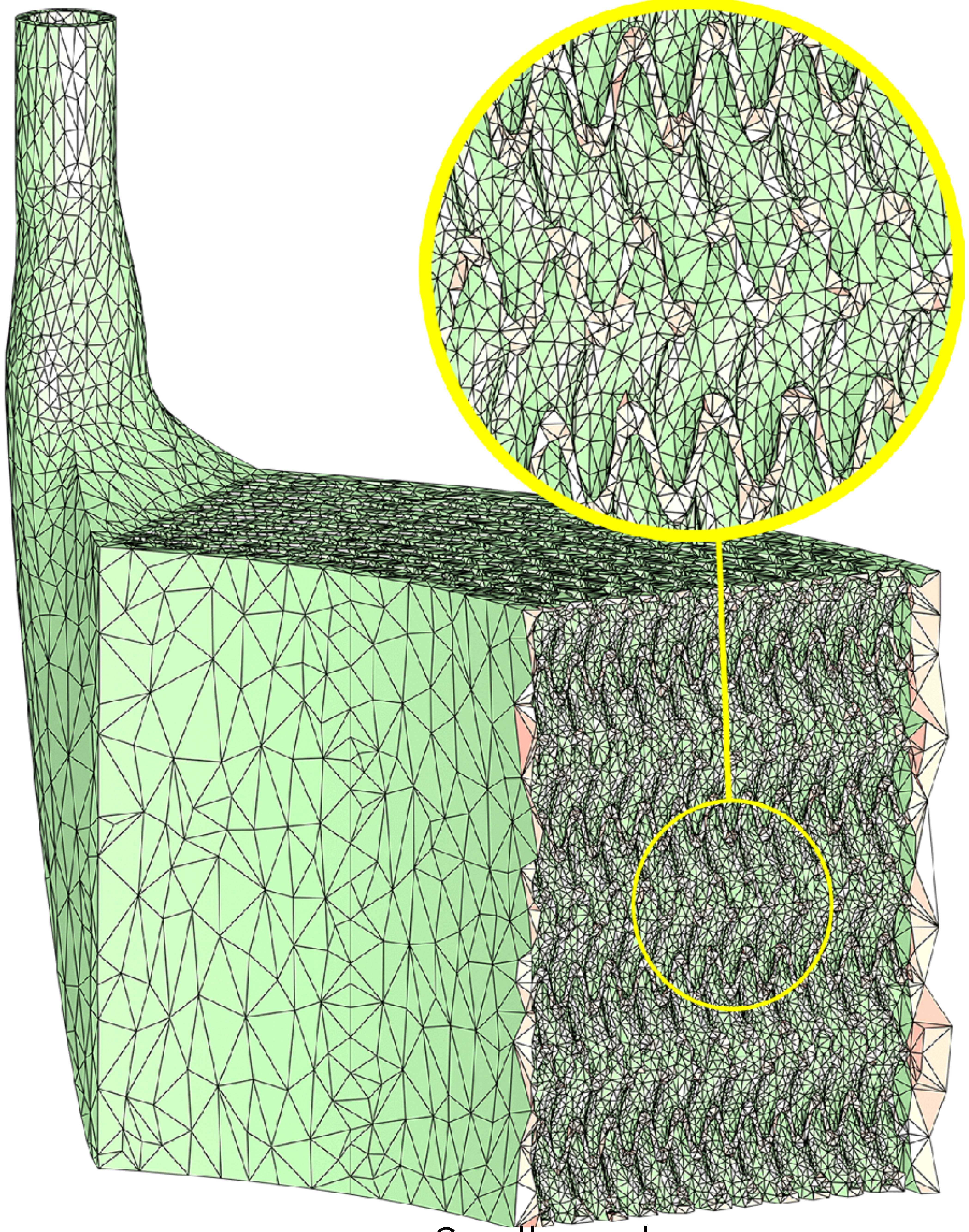


Fast Tetrahedral Meshing in the Wild
[Yixin Hu], [Teseo Schneider], Bolun Wang, [Denis Zorin], [Daniele Panozzo],
Arxiv, 2019
[Paper] [Code]



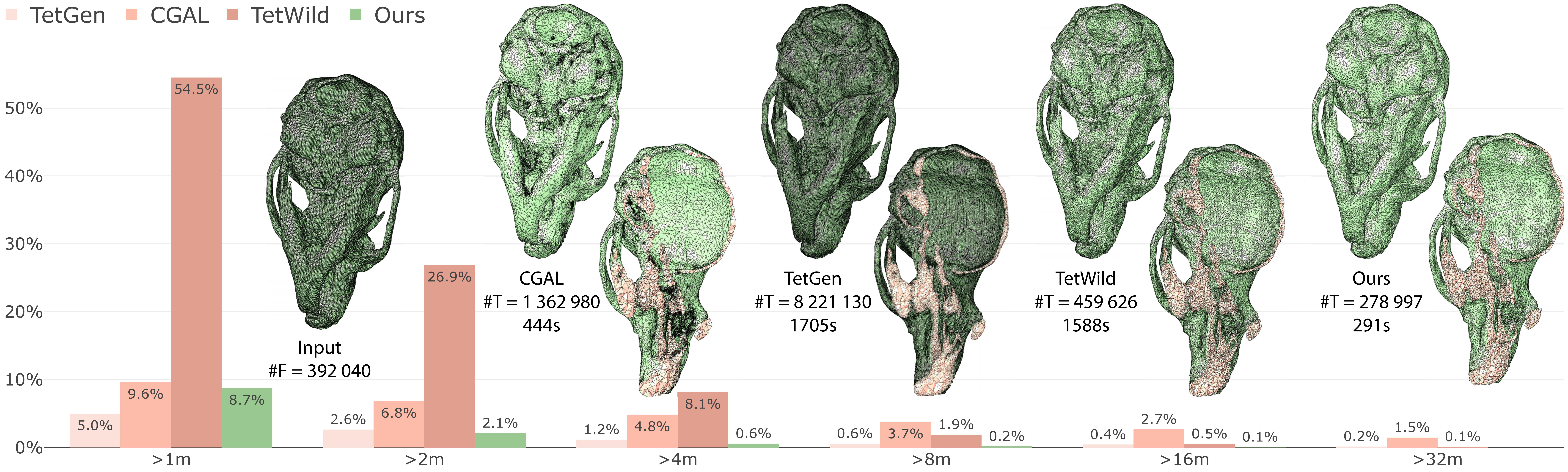


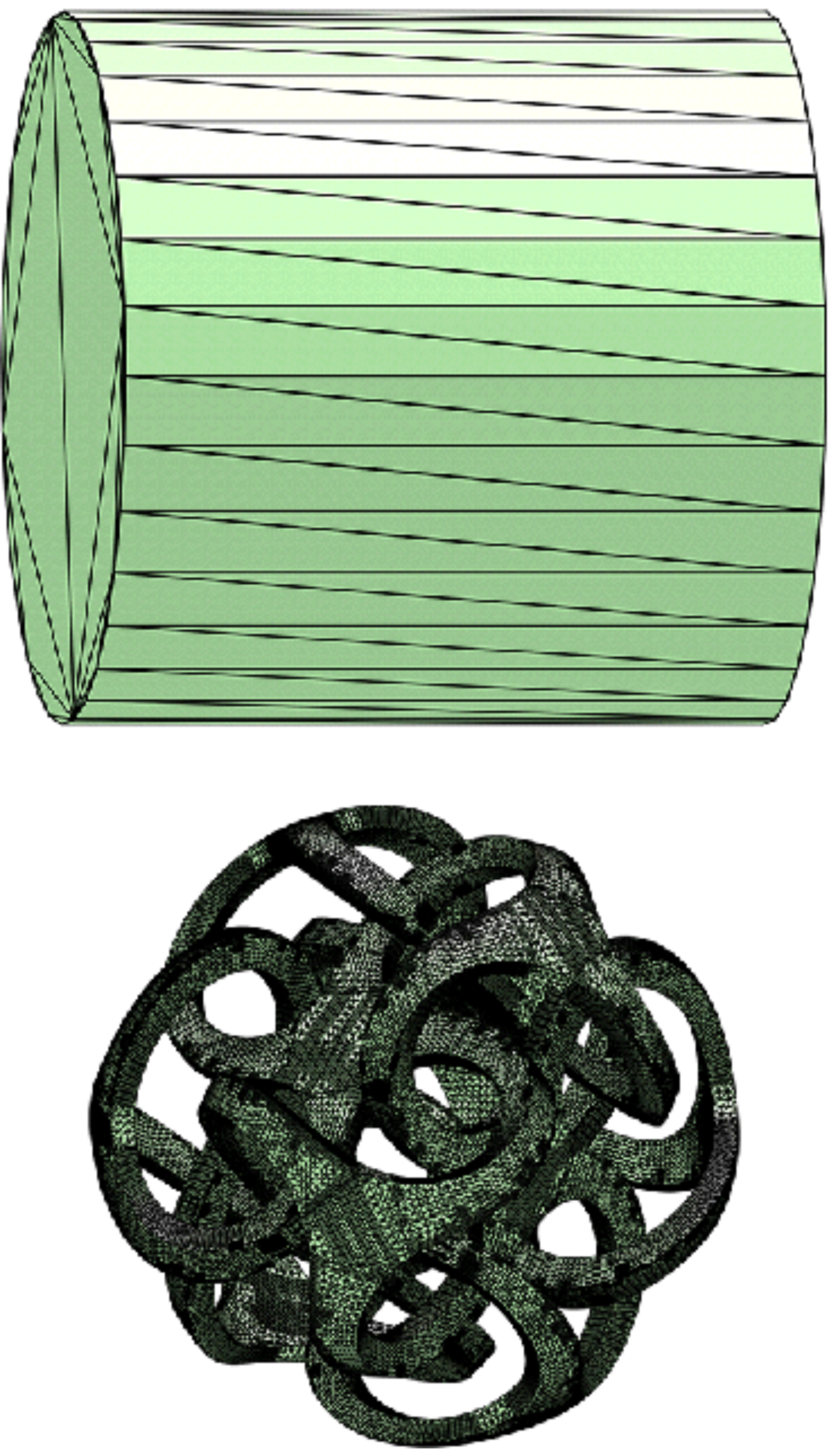
Large envelope
54m

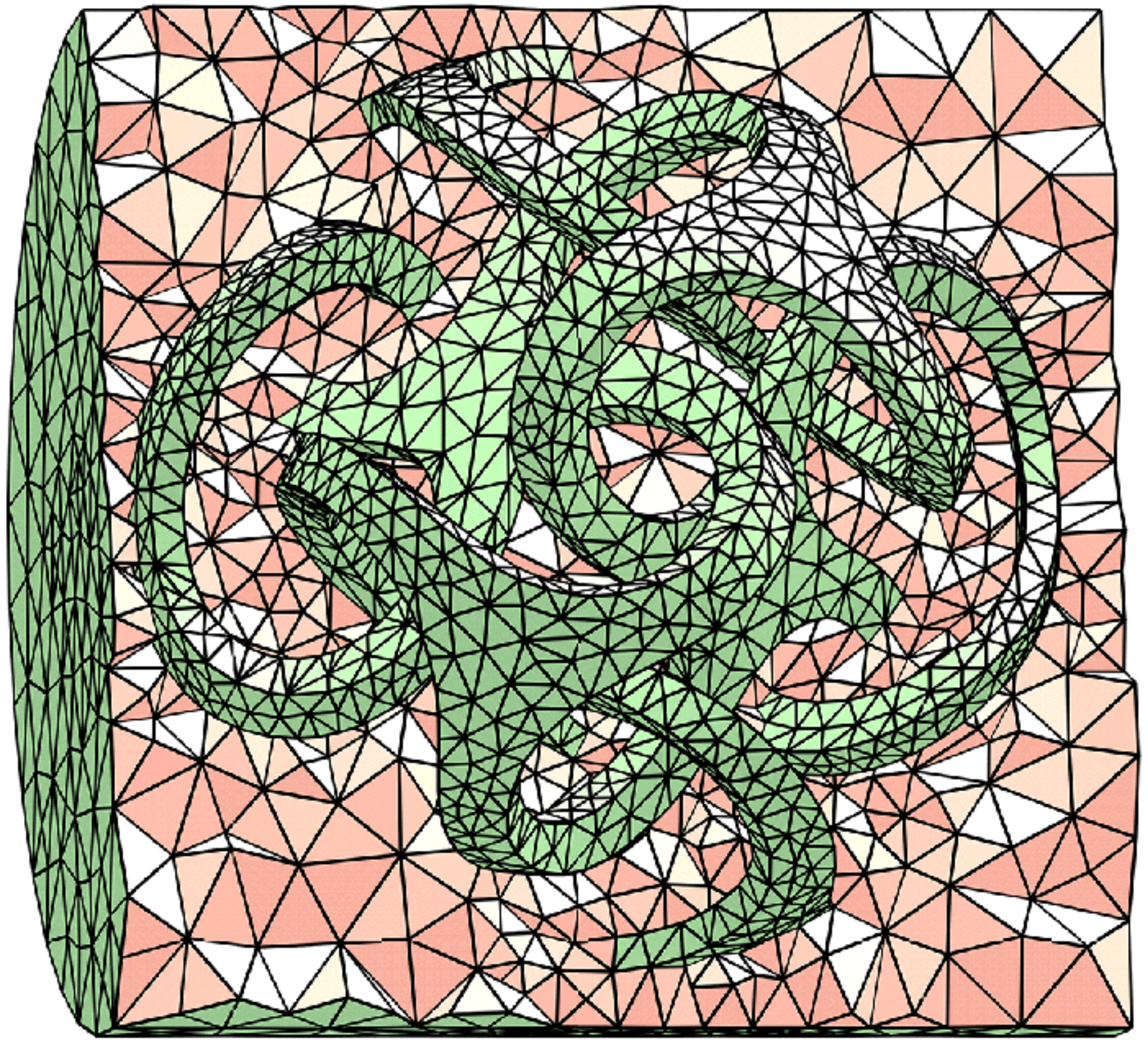


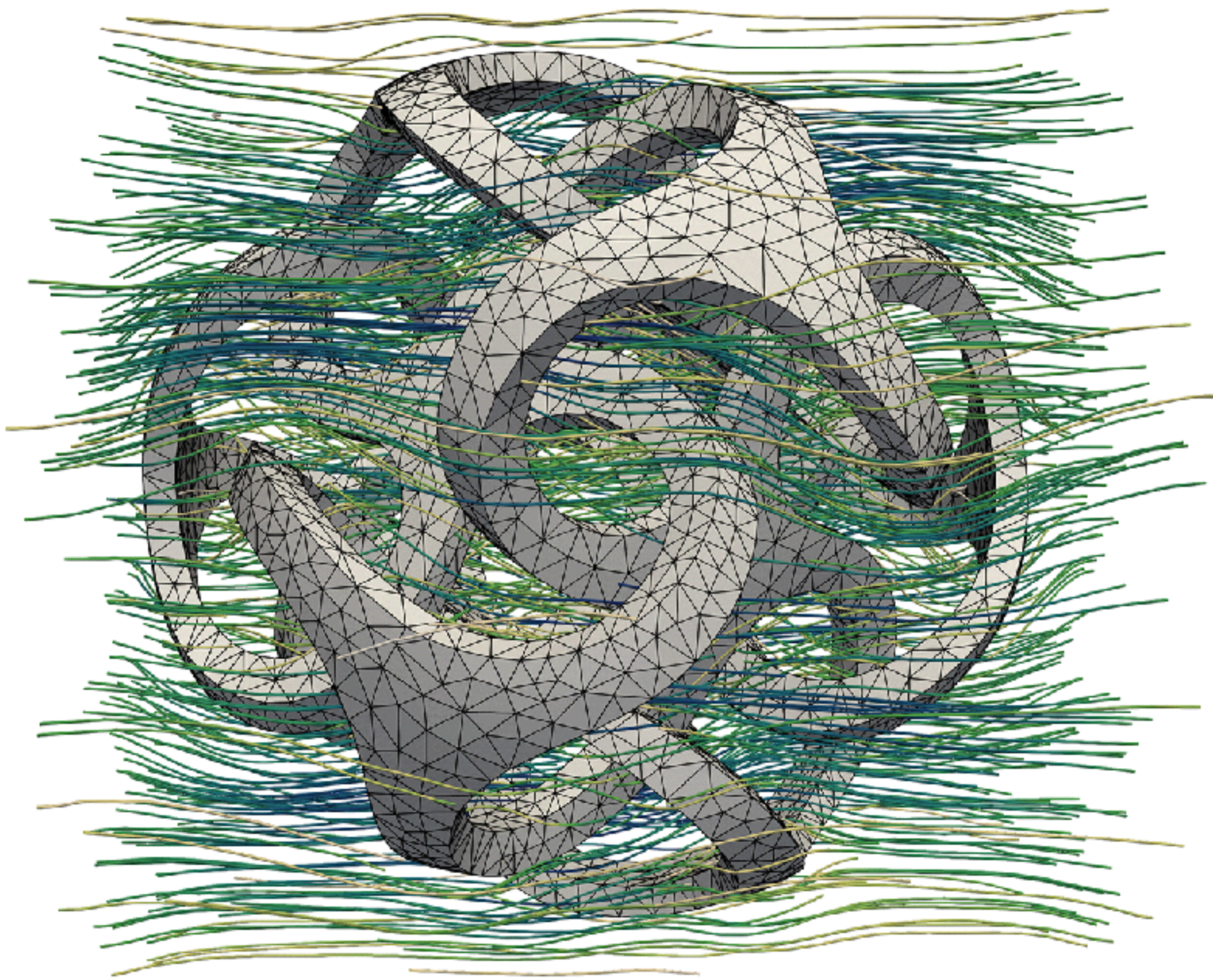
Small envelope
1h 37m

Faster than Tetgen!

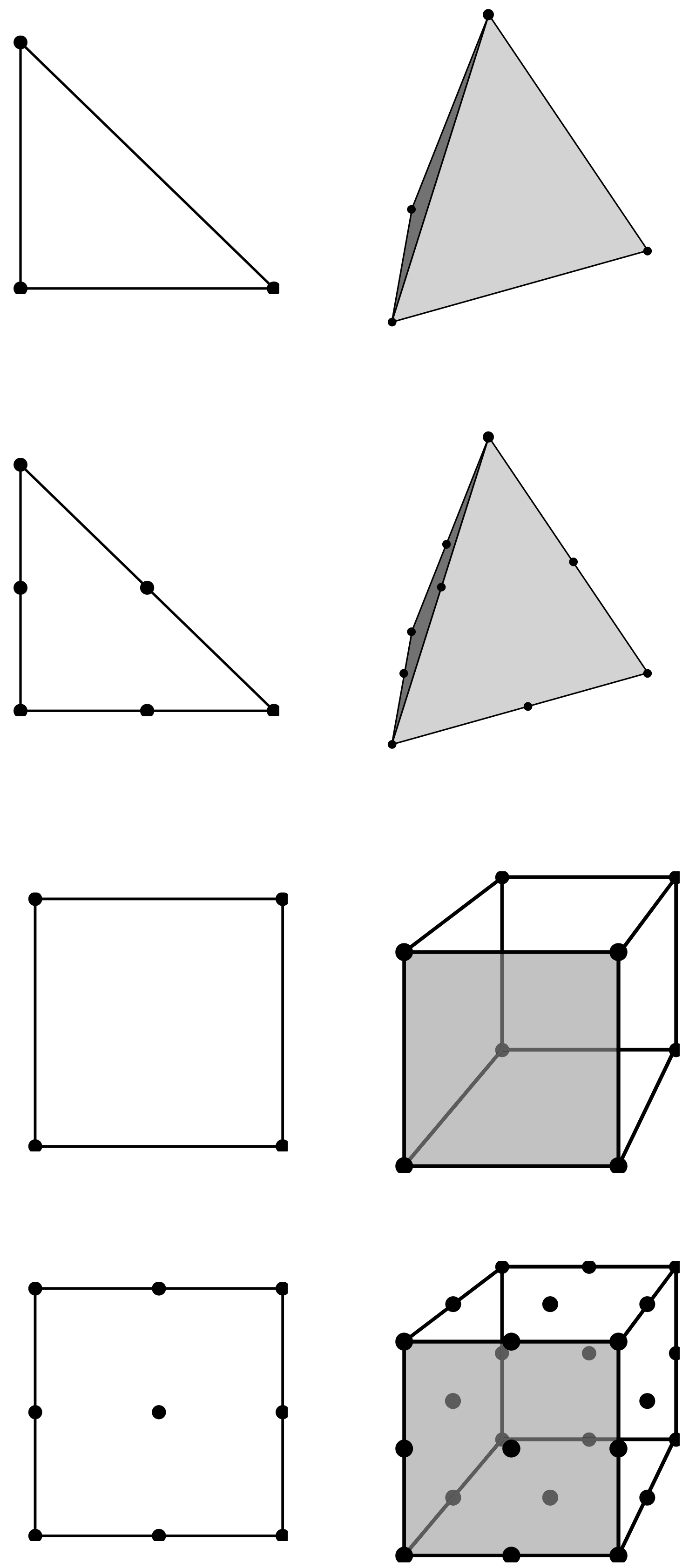




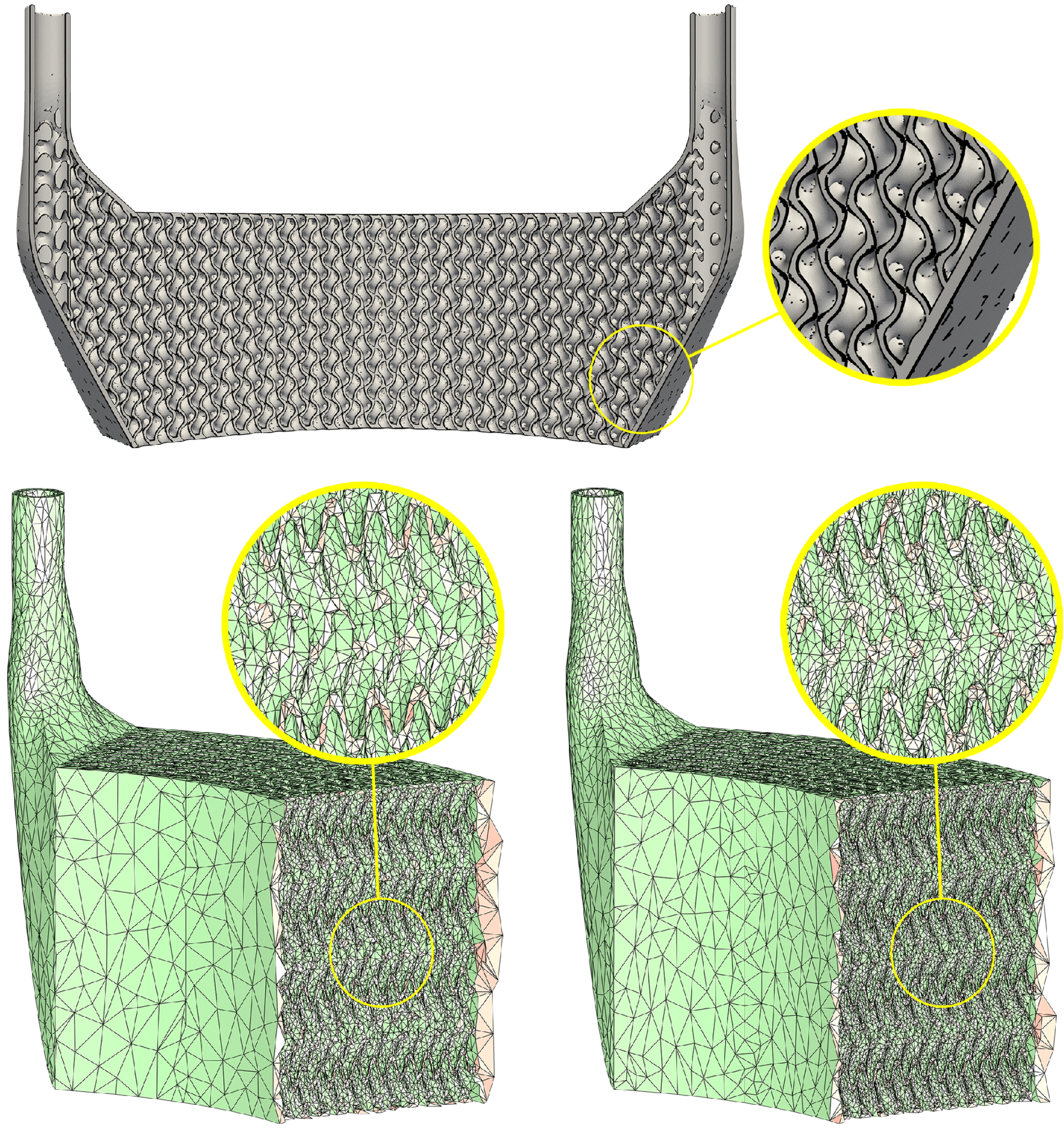




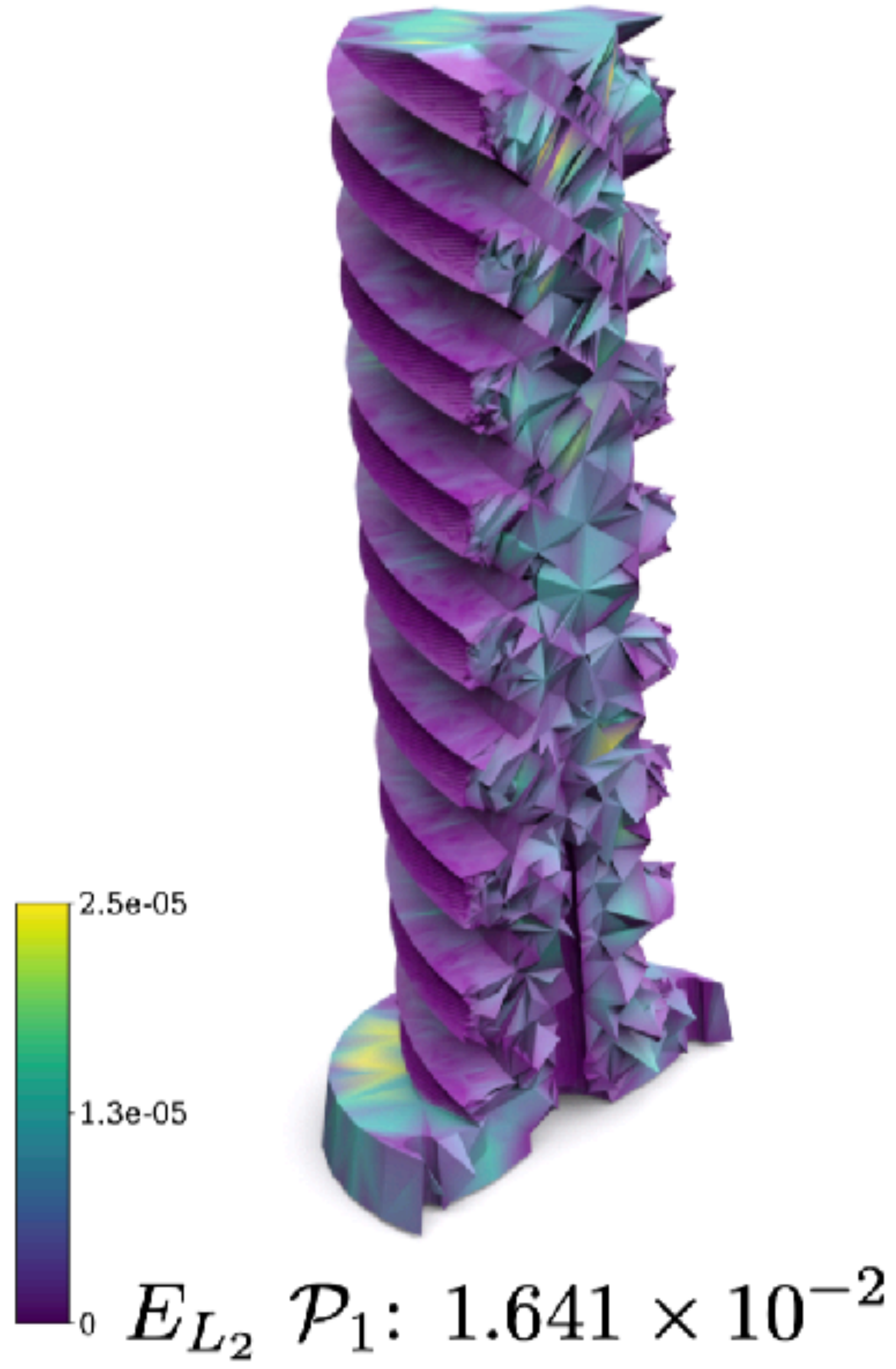
Overview



Which discretization provides lower running time for a fixed accuracy?

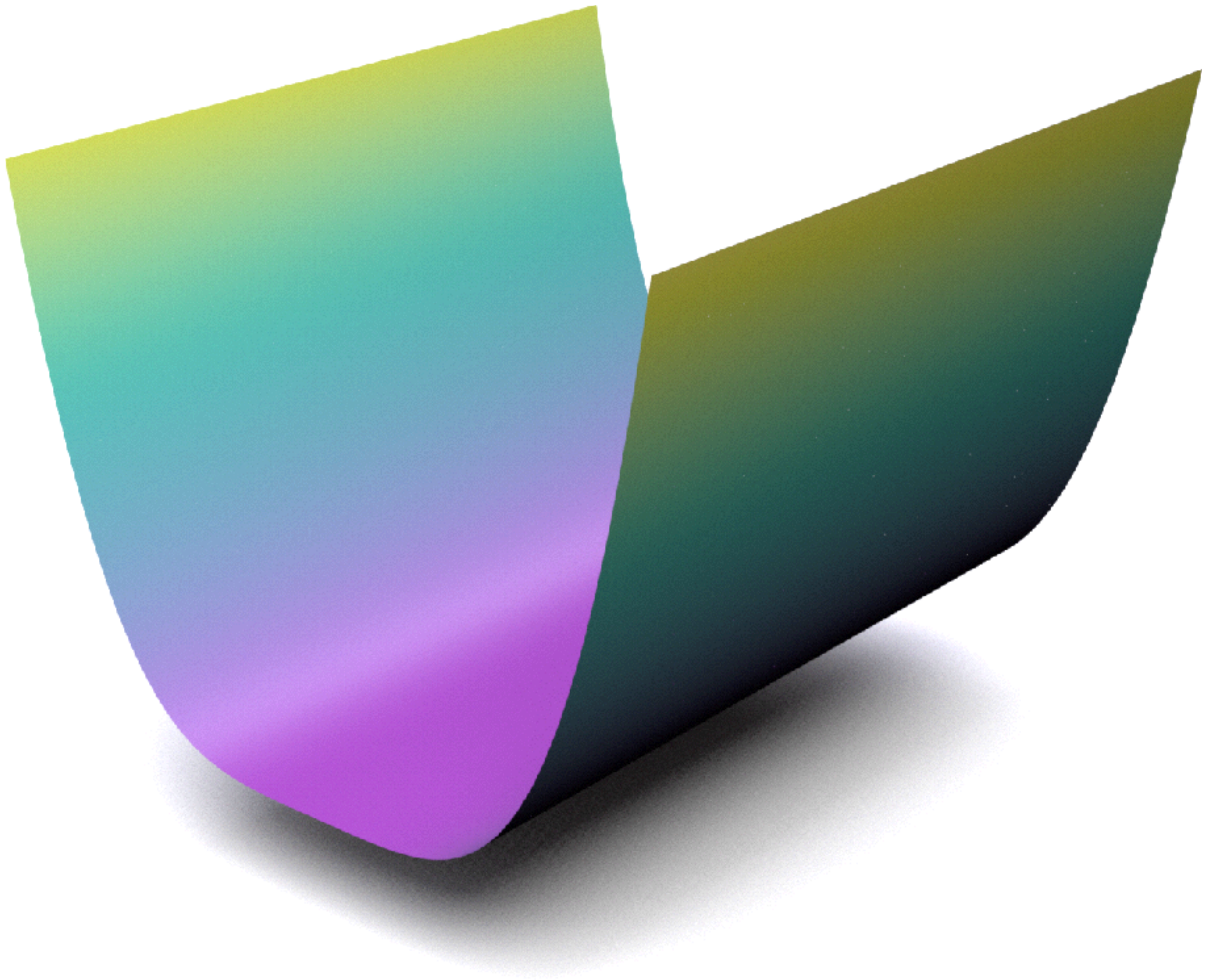


Can you mesh robustly without any assumption on the input?



Does mesh quality affect the accuracy of the FEM solution?

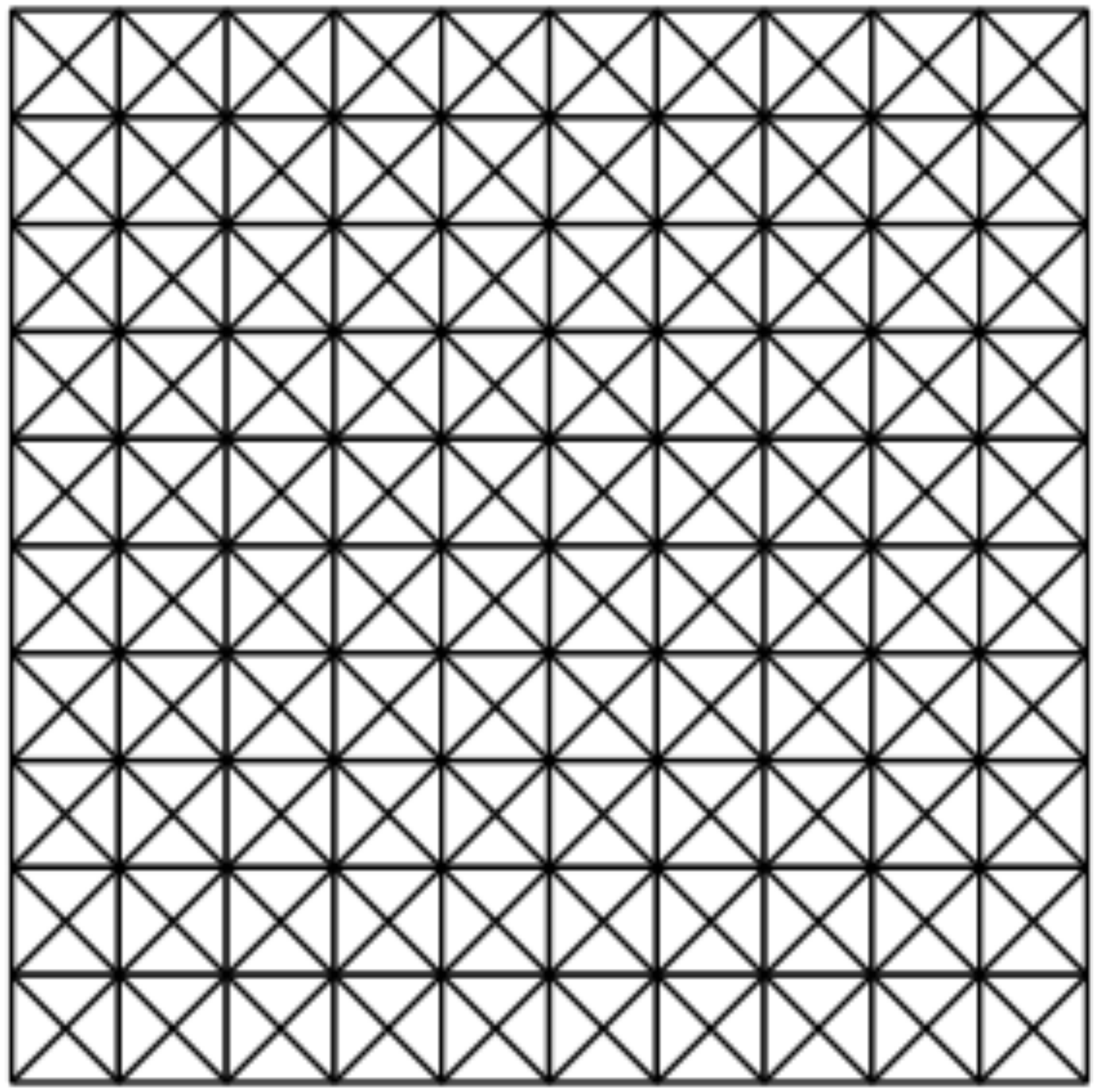
Does Quality Matter?



$$\Delta u = f, 12x^2 f = 12x^2$$

Solution

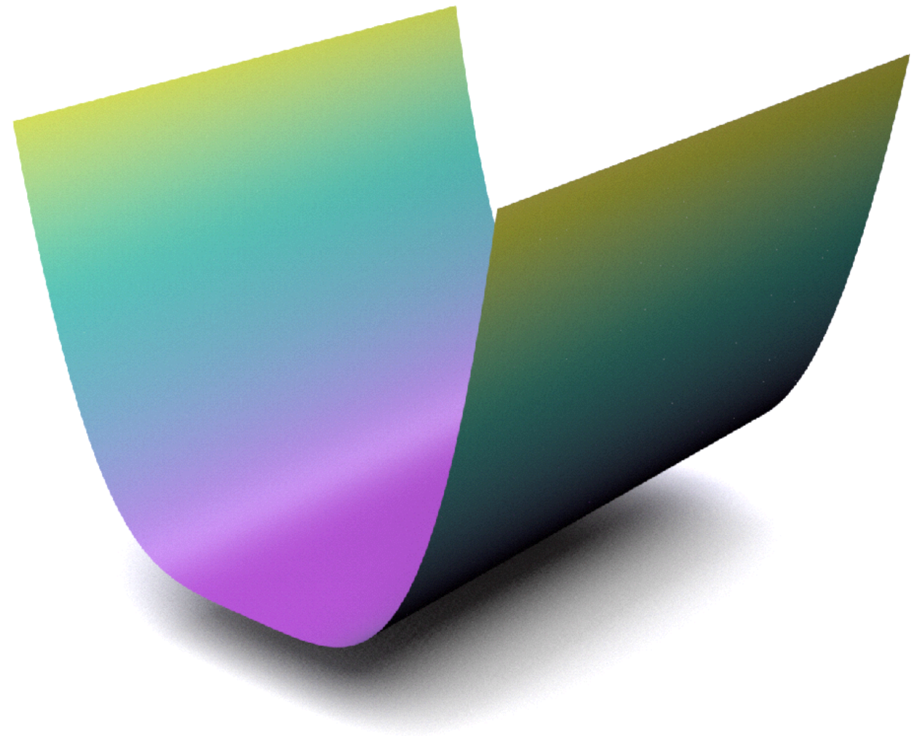
$$u = x^4$$

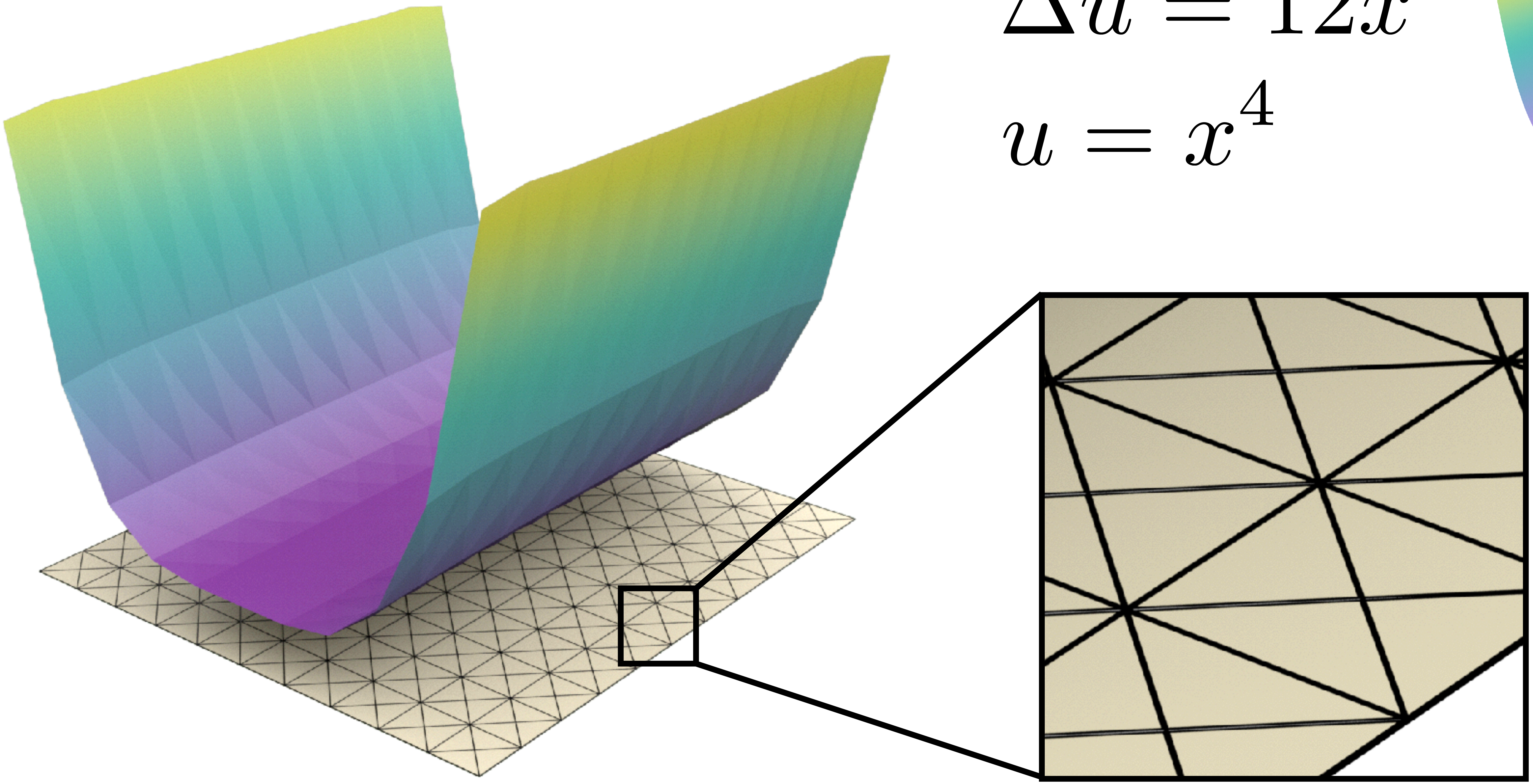


Does Quality Matter?

$$\Delta u = 12x^2$$

$$u = x^4$$

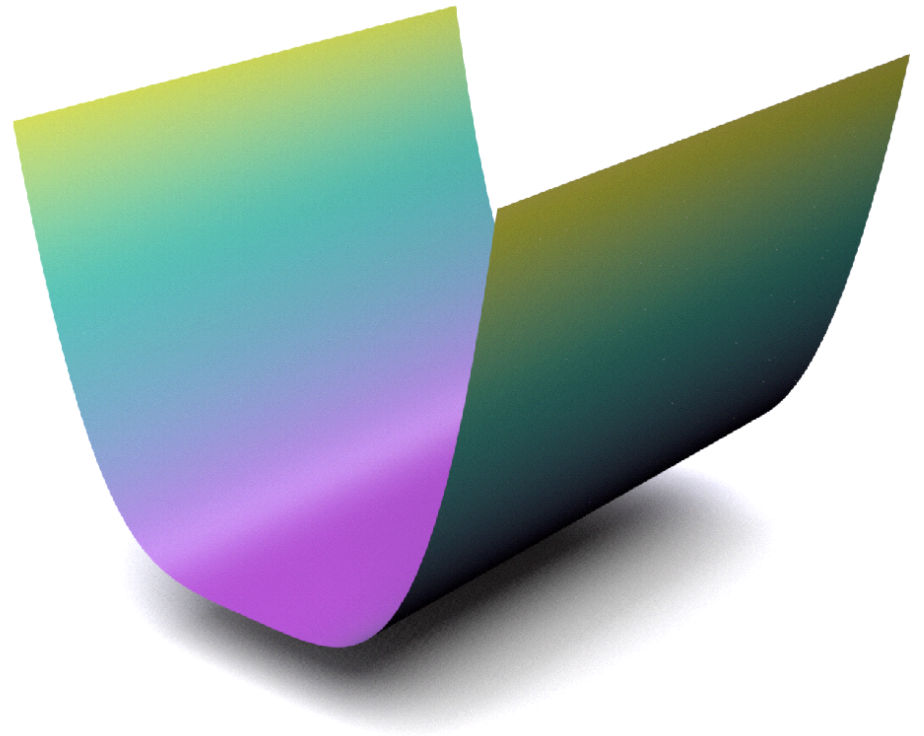


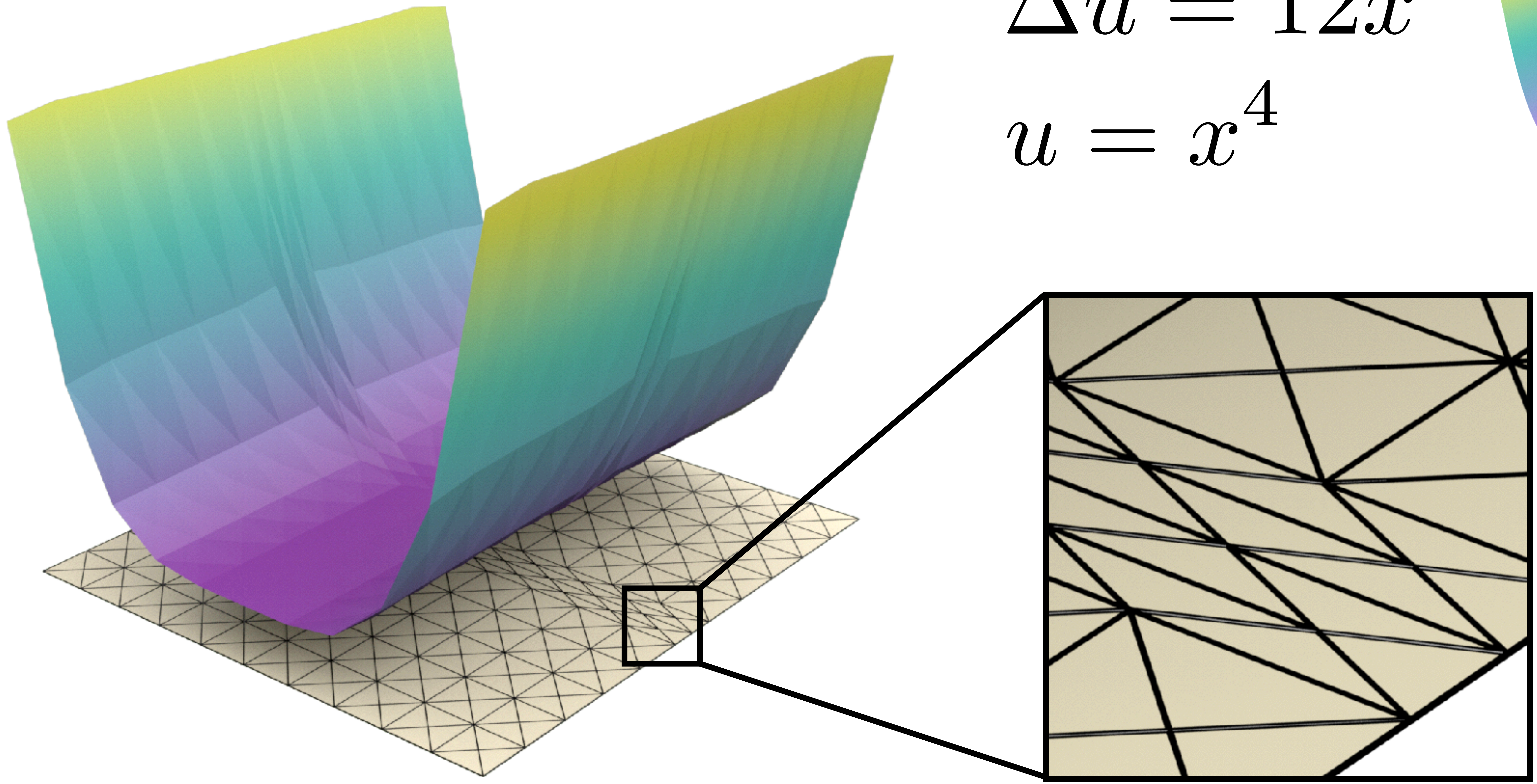


Does Quality Matter?

$$\Delta u = 12x^2$$

$$u = x^4$$

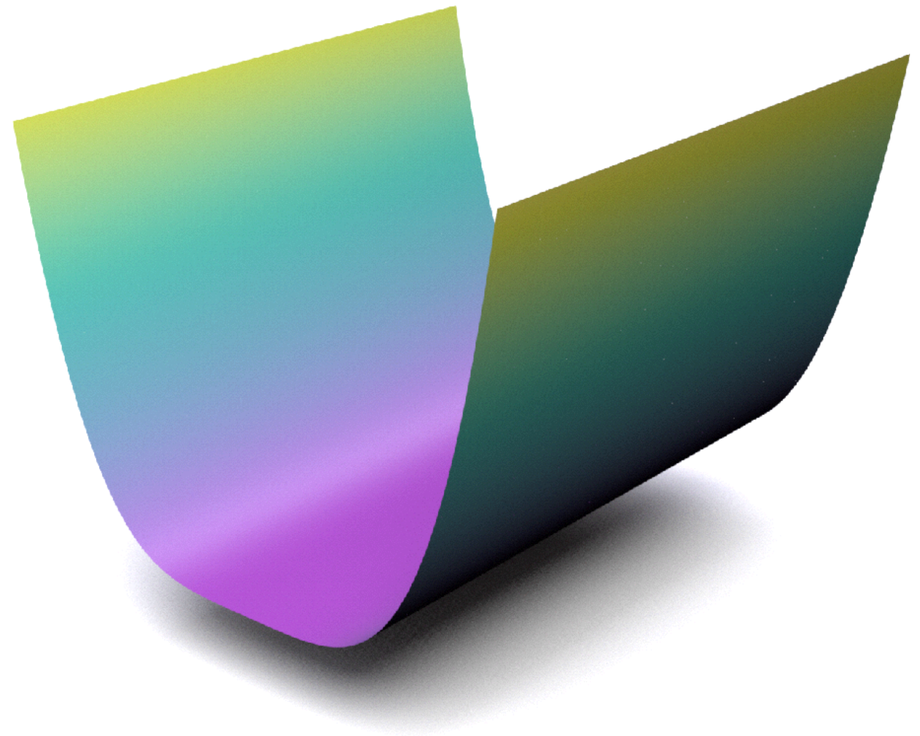


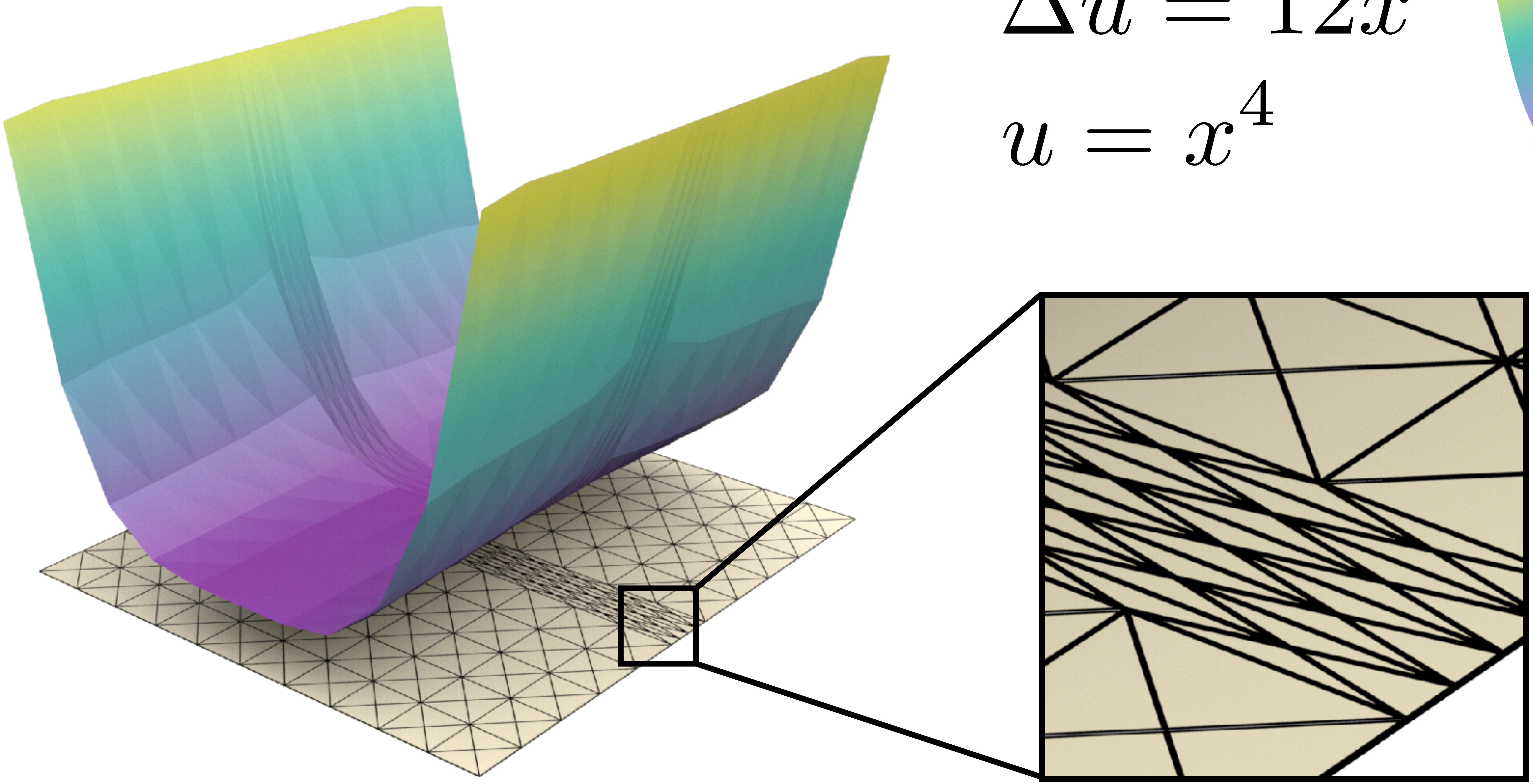


Does Quality Matter?

$$\Delta u = 12x^2$$

$$u = x^4$$

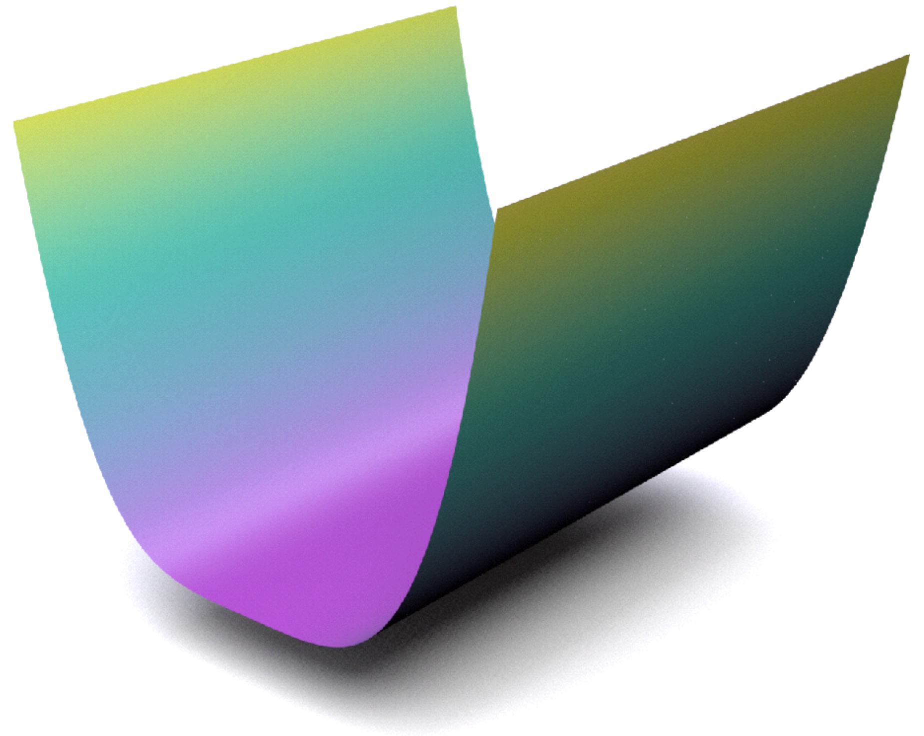


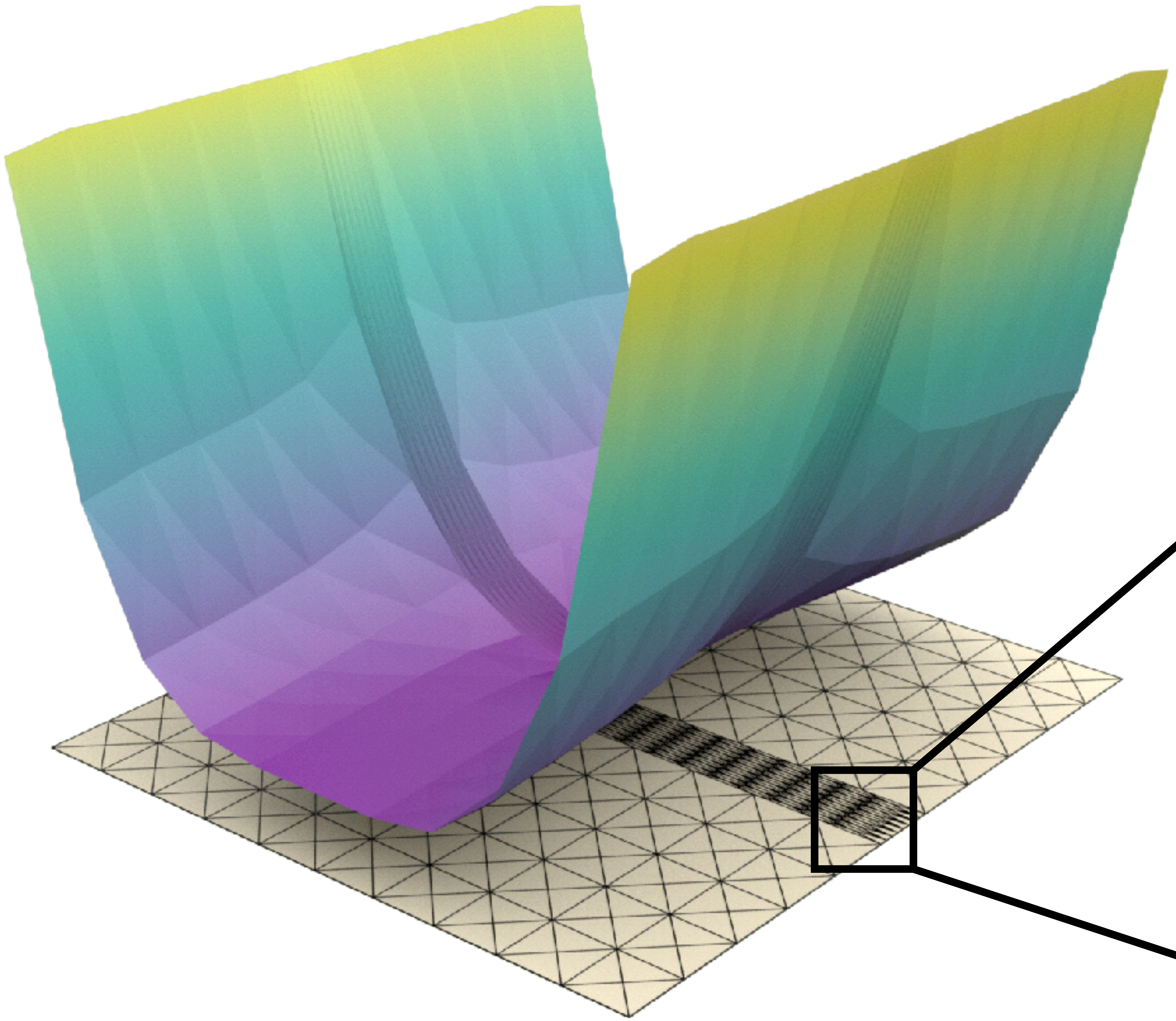


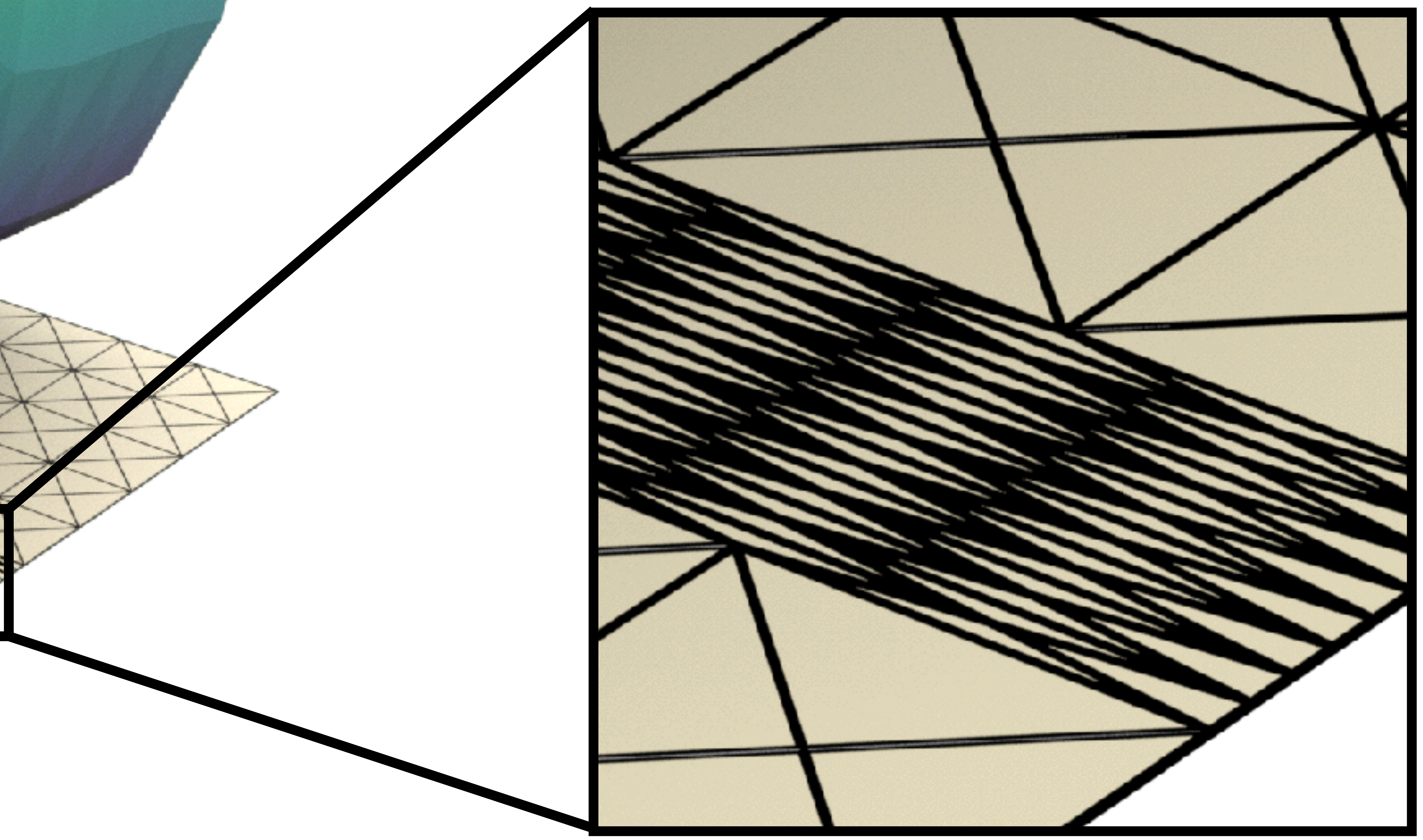
Does Quality Matter?

$$\Delta u = 12x^2$$

$$u = x^4$$



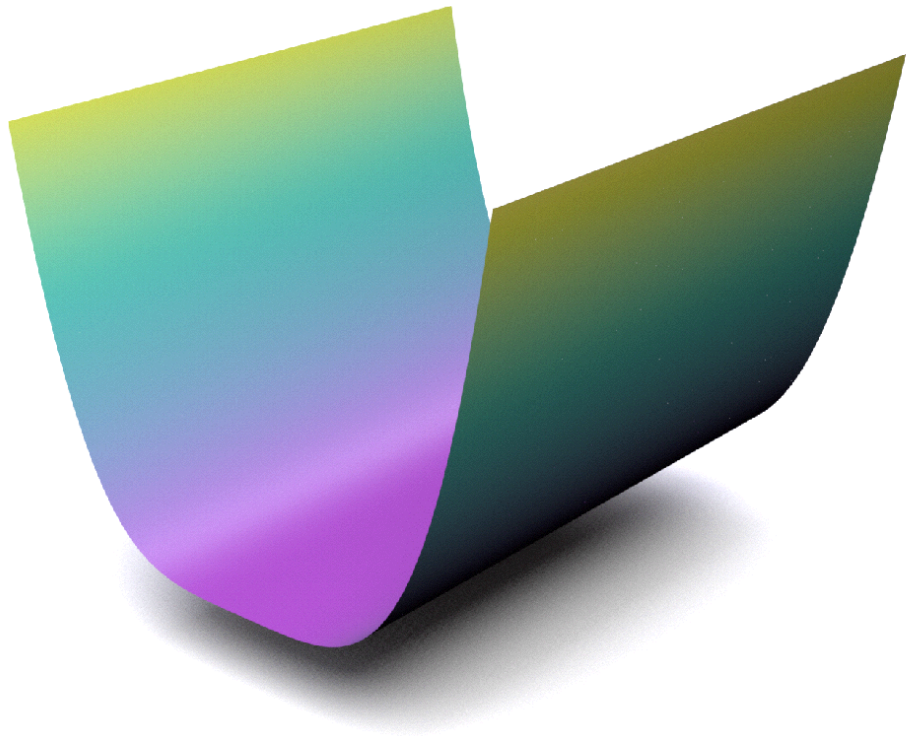


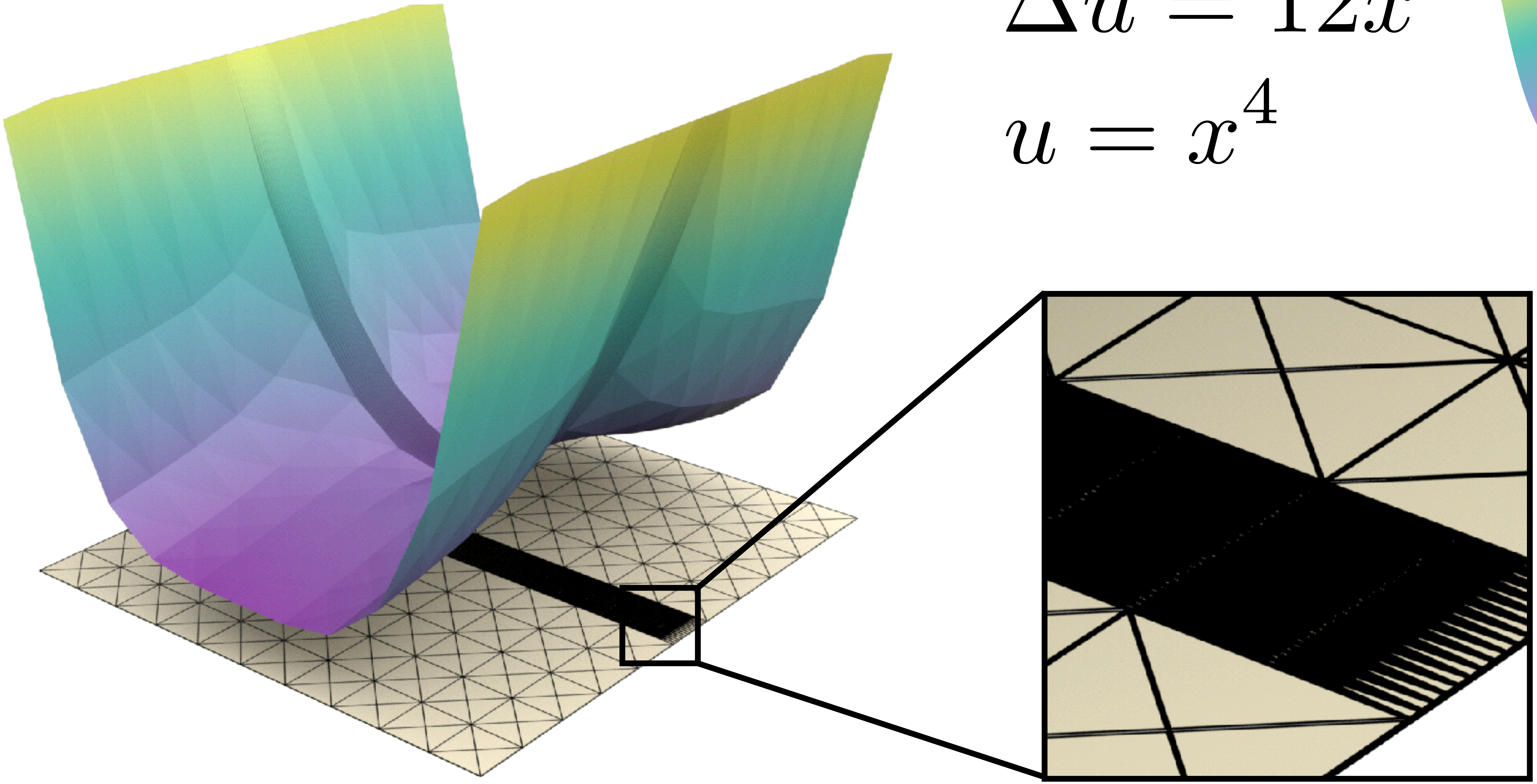


Does Quality Matter?

$$\Delta u = 12x^2$$

$$u = x^4$$

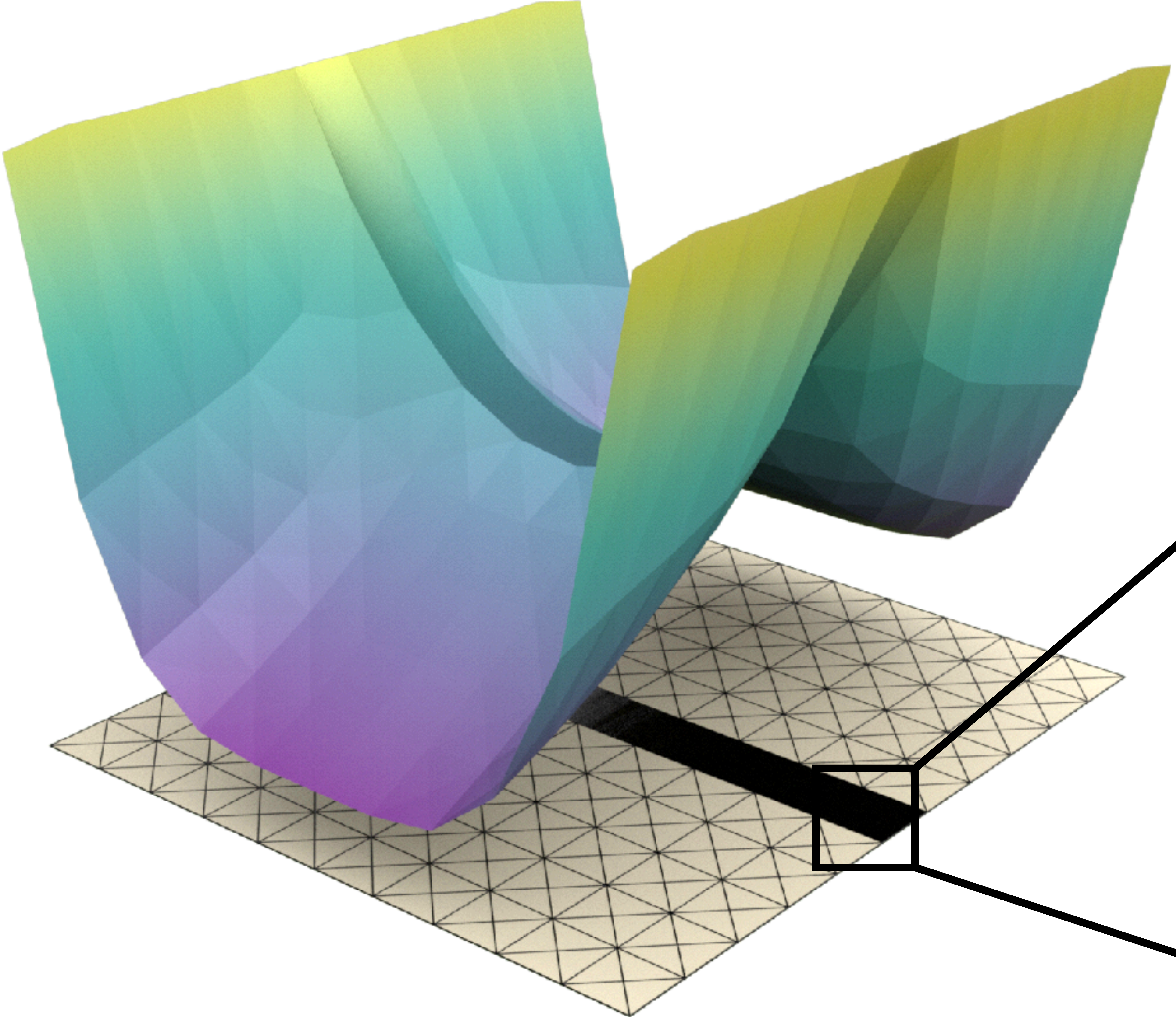


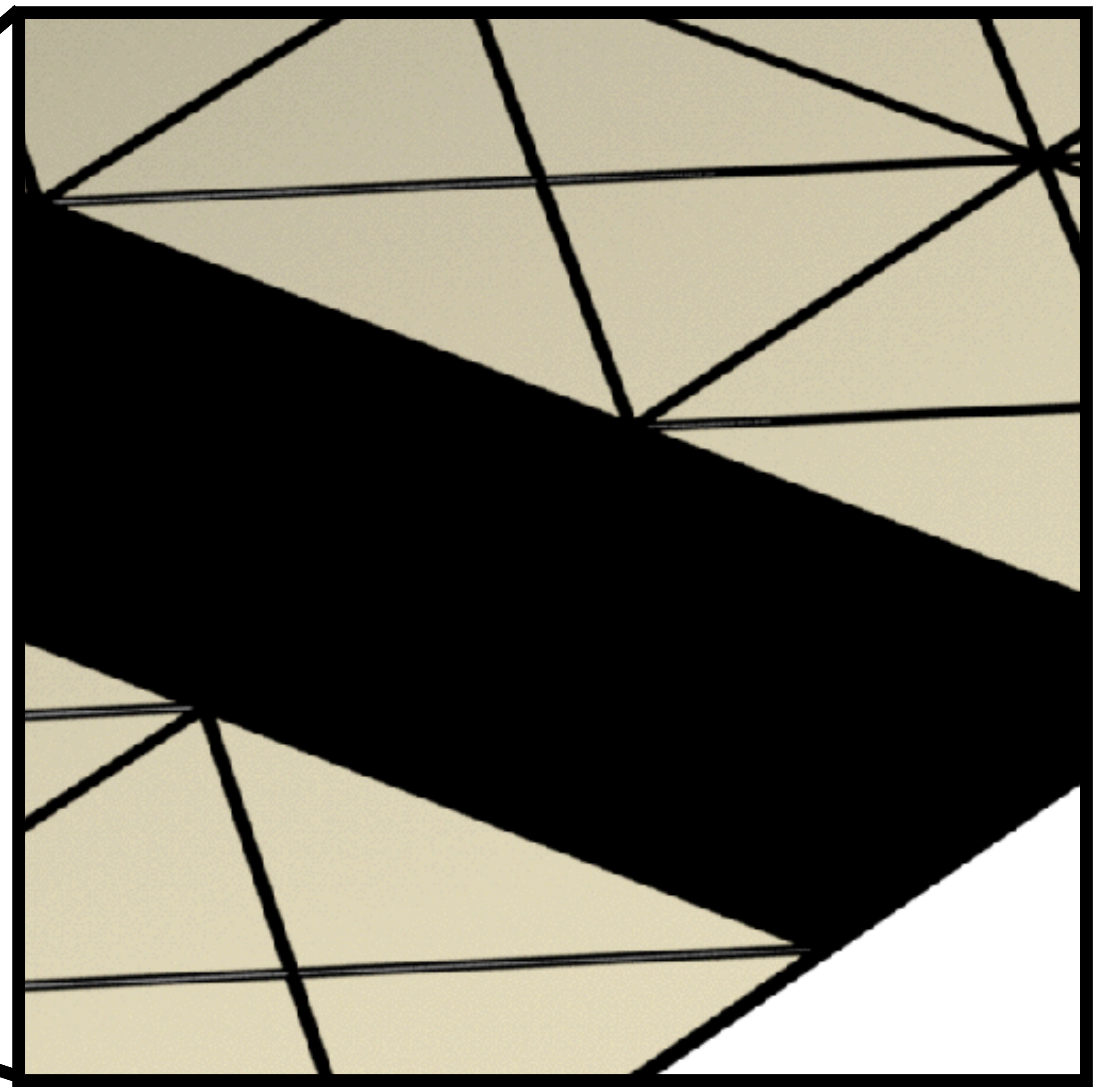


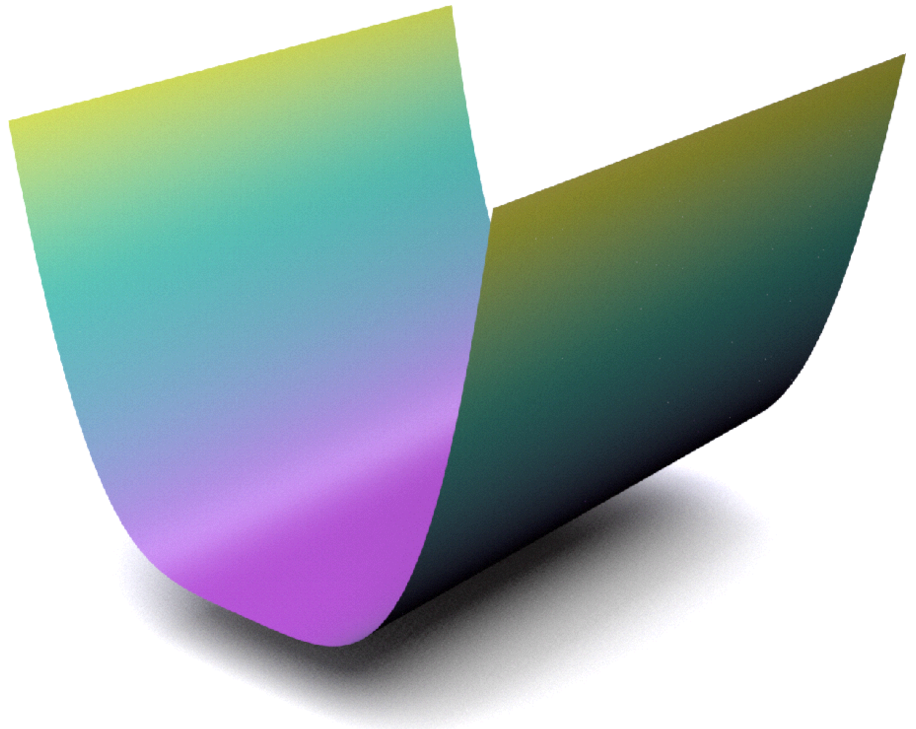
Does Quality Matter?

$$\Delta u = 12x^2$$

$$u = x^4$$



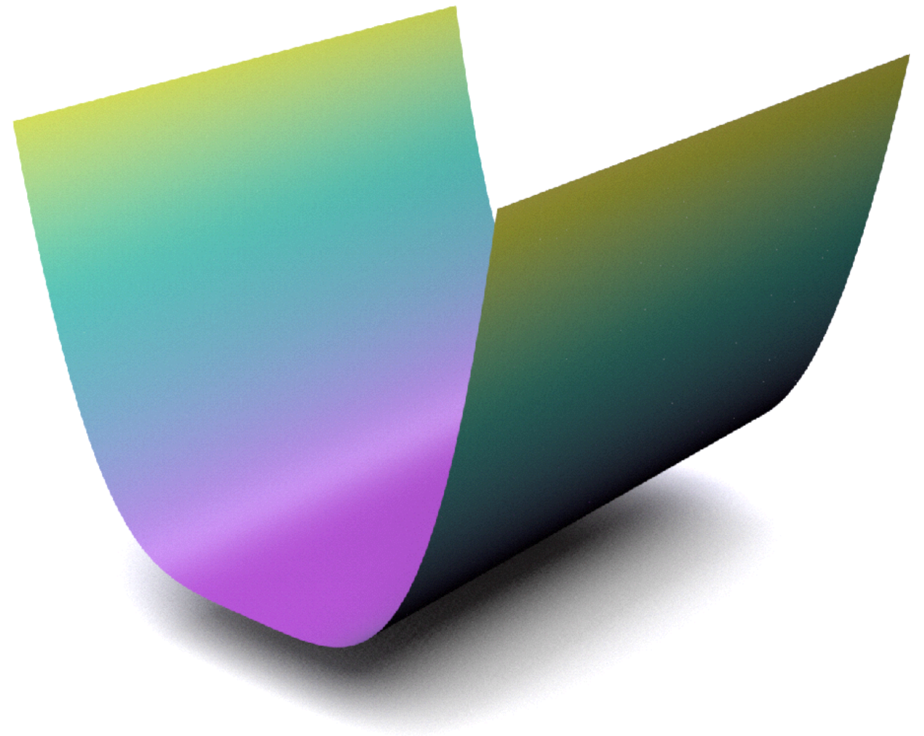


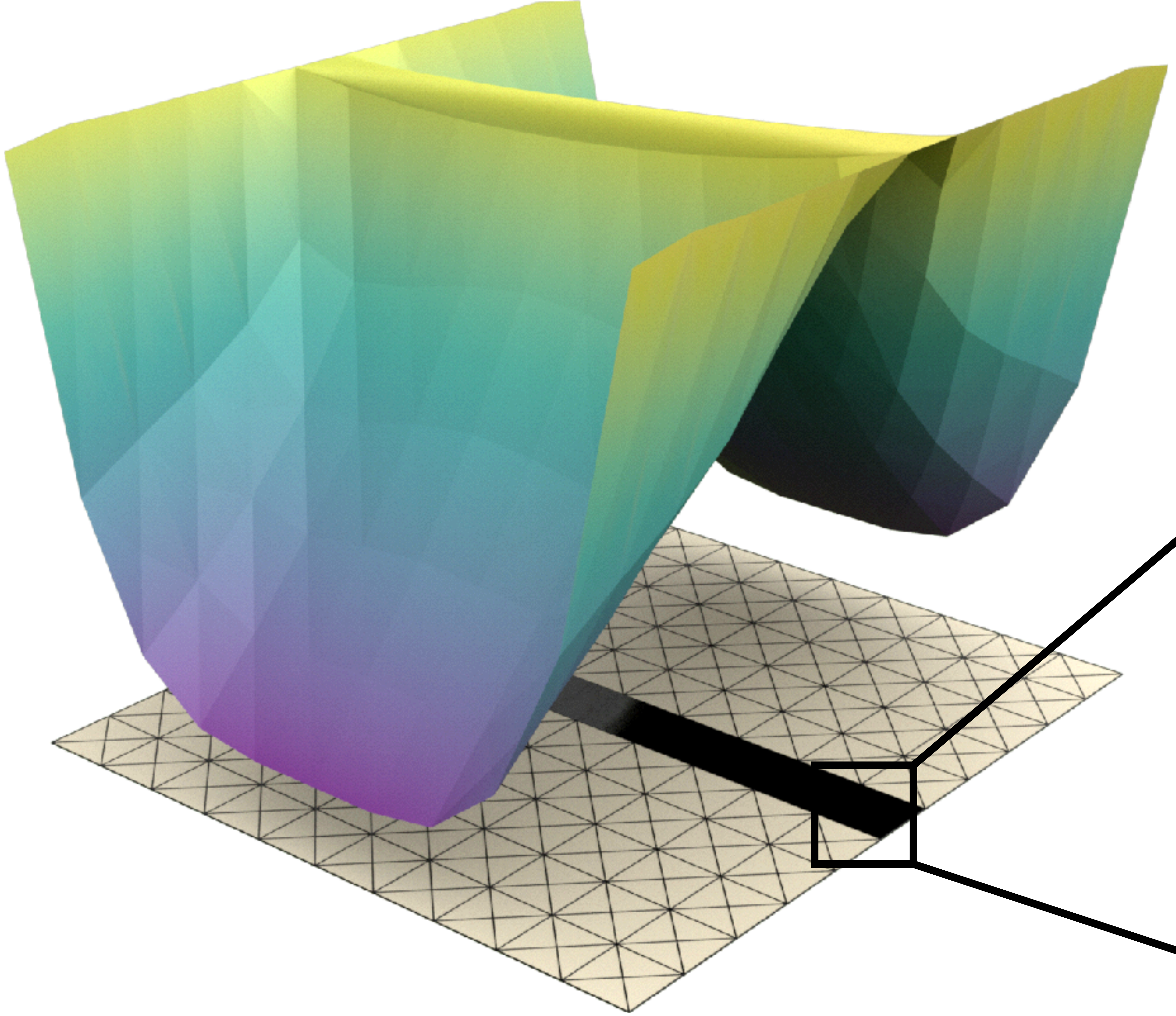


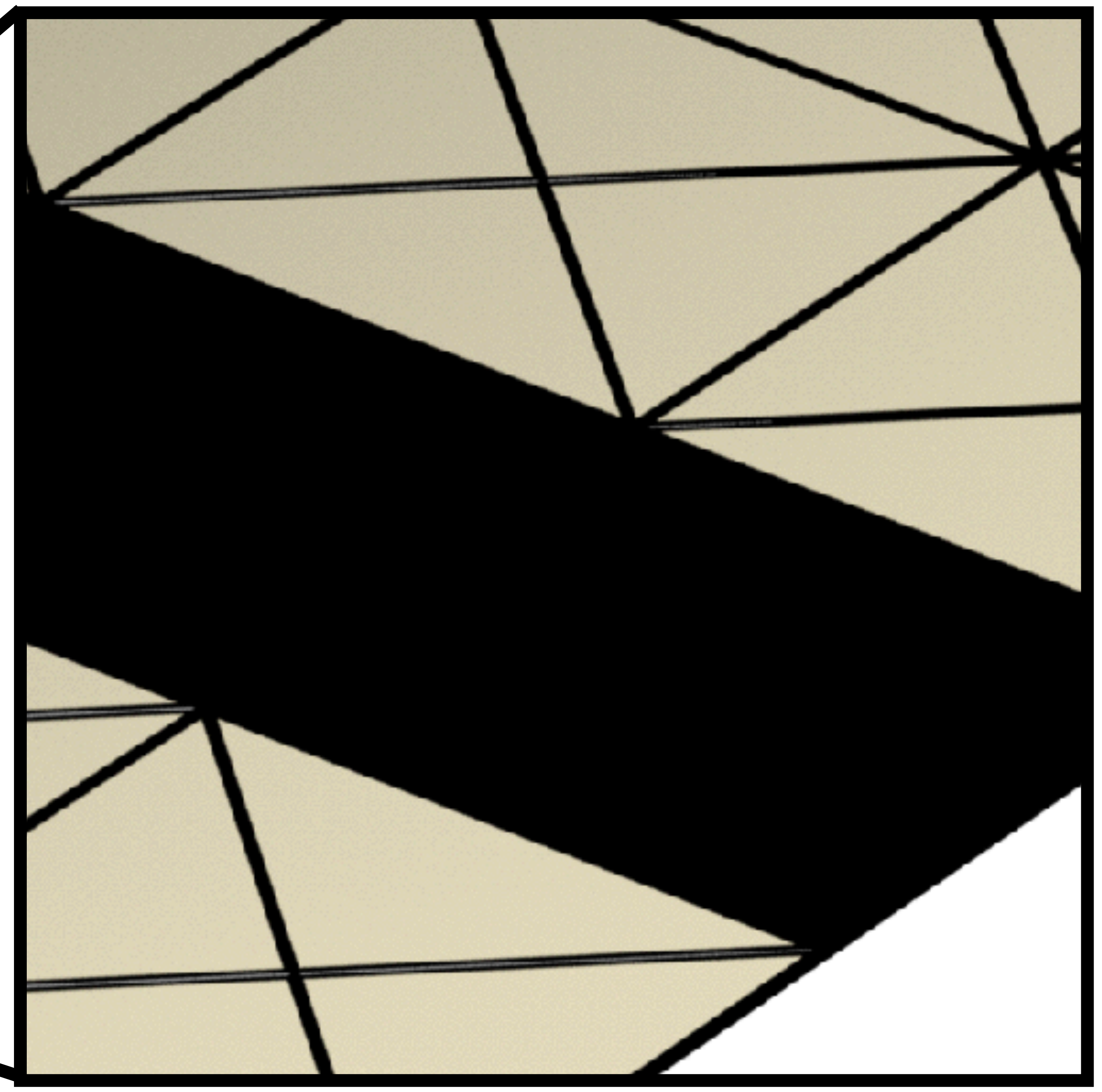
Does Quality Matter?

$$\Delta u = 12x^2$$

$$u = x^4$$

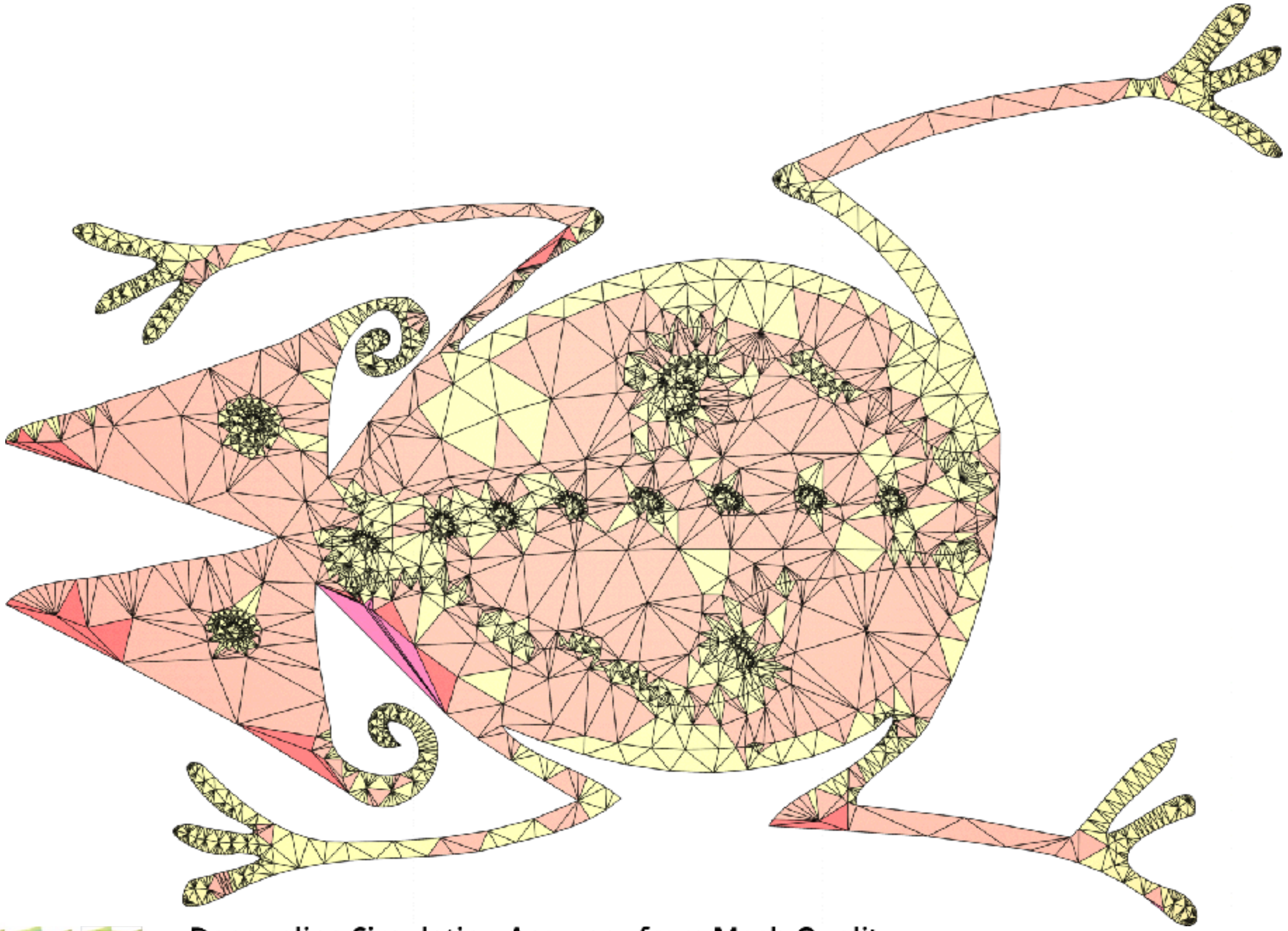


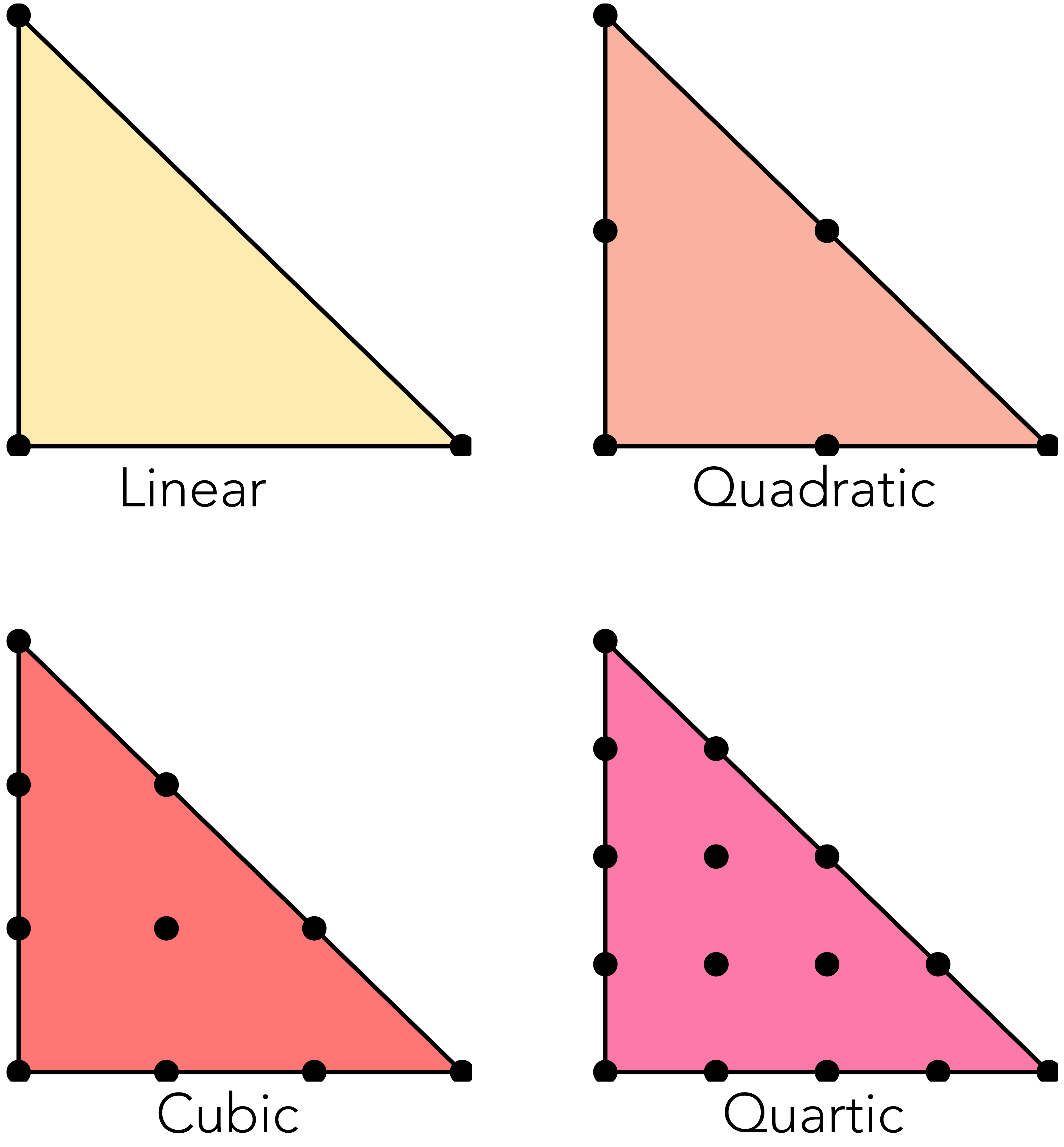


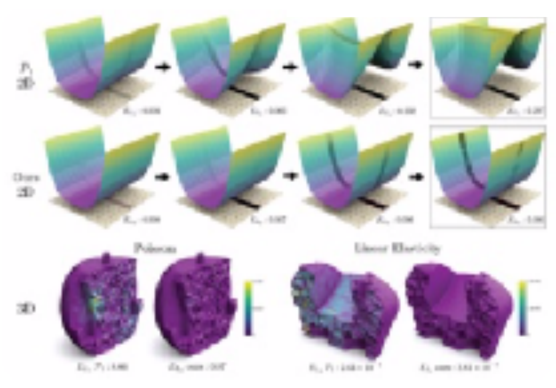


Our Solution

Locally increase the order of elements







Decoupling Simulation Accuracy from Mesh Quality

Teseo Schneider, Yixin Hu, Jeremie Dumas, Xifeng Gao, Daniele Panozzo, Denis Zorin,

ACM Transaction on Graphics (SIGGRAPH Asia), 2018

[Paper] [Code]

Refinement

- A posteriori h-refinement
  - Increase the mesh resolution locally

  [Wu 01], [Simnett 09], [Wicke 10], [Pfaff 14], …

- A posteriori p-refinement
  - Solve, then increase order where necessary

  [Babuška 94], [Kaufmann 13], [Bargteil 14], [Edwards 14], …

- Ours is a priori p-refinement
  - We increase order only based on the input

Overview

$$k = \frac{\ln\left(B\hat{h}^{\hat{k}+1}\frac{\sigma_E^2}{\hat{\sigma}^2}\right) - \ln\, h_E}{\ln\, h_E}$$

1. Use formula

Magic Formula

Order of an element

$$\boxed{k} = \frac{\ln\left(B\hat{h}^{\hat{k}+1}\frac{\sigma_E^2}{\hat{\sigma}^2}\right) - \ln\, h_E}{\ln\, h_E}$$

Magic Formula

User parameter, = 3

$$k = \frac{\ln\left(\boxed{B}\hat{h}^{\hat{k}+1}\frac{\sigma_E^2}{\hat{\sigma}^2}\right) - \ln h_E}{\ln h_E}$$

Magic Formula

Average edge length

$$k = \frac{\ln\left(B\hat{h}^{\hat{k}+1}\frac{\sigma_E^2}{\hat{\sigma}^2}\right) - \ln h_E}{\ln h_E}$$

Magic Formula

Base order, usually 1

$$k = \frac{\ln\left(B\hat{h}^{\hat{k}+1}\frac{\sigma_E^2}{\hat{\sigma}^2}\right) - \ln h_E}{\ln h_E}$$

Magic Formula

$$k = \frac{\ln\left(B\hat{h}^{\hat{k}+1}\frac{\sigma_E^2}{\hat{\sigma}^2}\right) - \ln h_E}{\ln h_E}$$

$$\hat{\sigma}_{2D} = \sqrt{3}/6$$

$$\hat{\sigma}_{3D} = \sqrt{6}/12$$

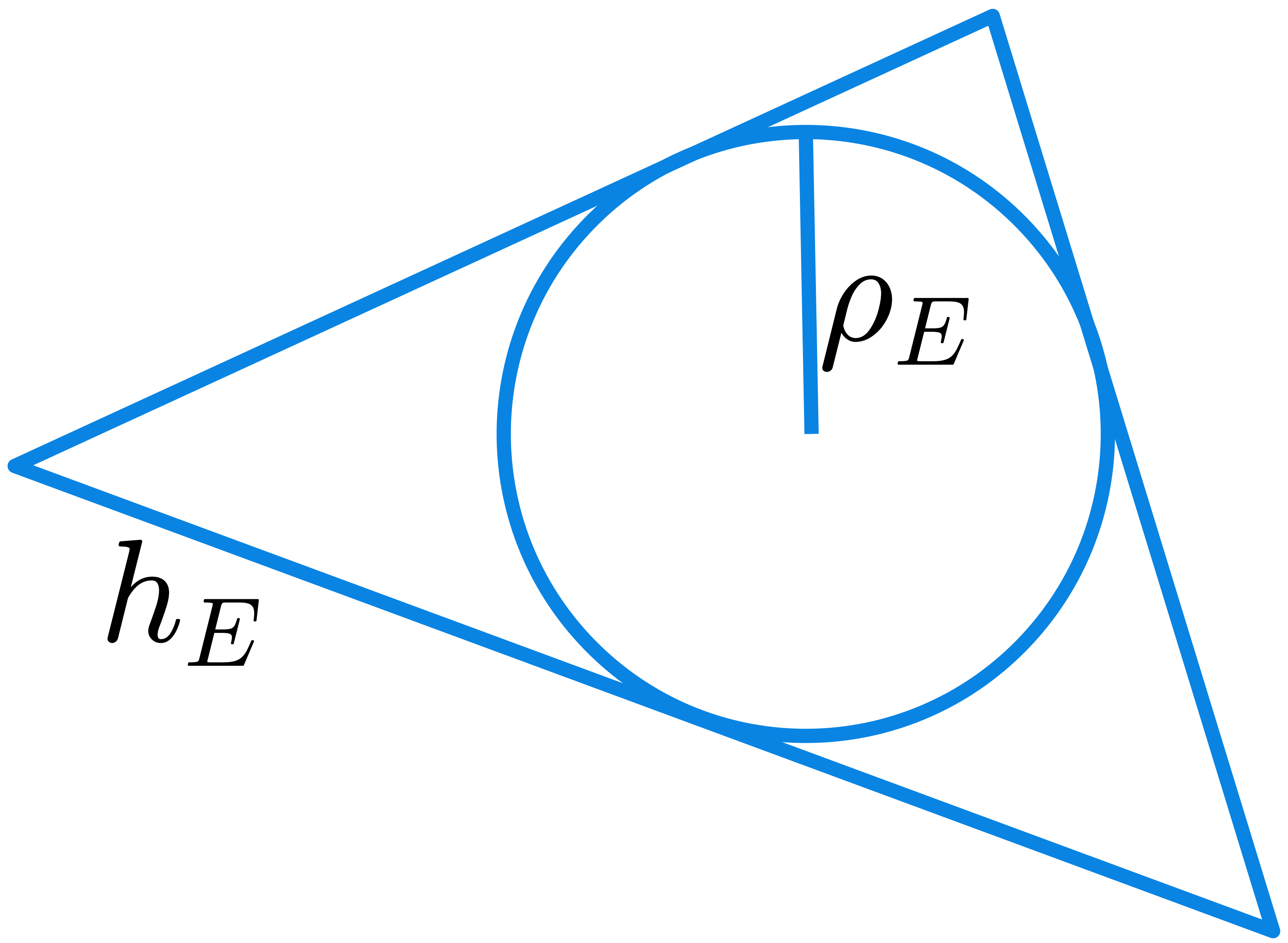
Magic Formula

$$\hat{\sigma}_{2D} = \sqrt{3}/6$$

$$\hat{\sigma}_{3D} = \sqrt{6}/12$$

$$k = \frac{\ln\left(B\hat{h}^{\hat{k}+1}\frac{\sigma_E^2}{\hat{\sigma}^2}\right) - \ln h_E}{\ln h_E}$$

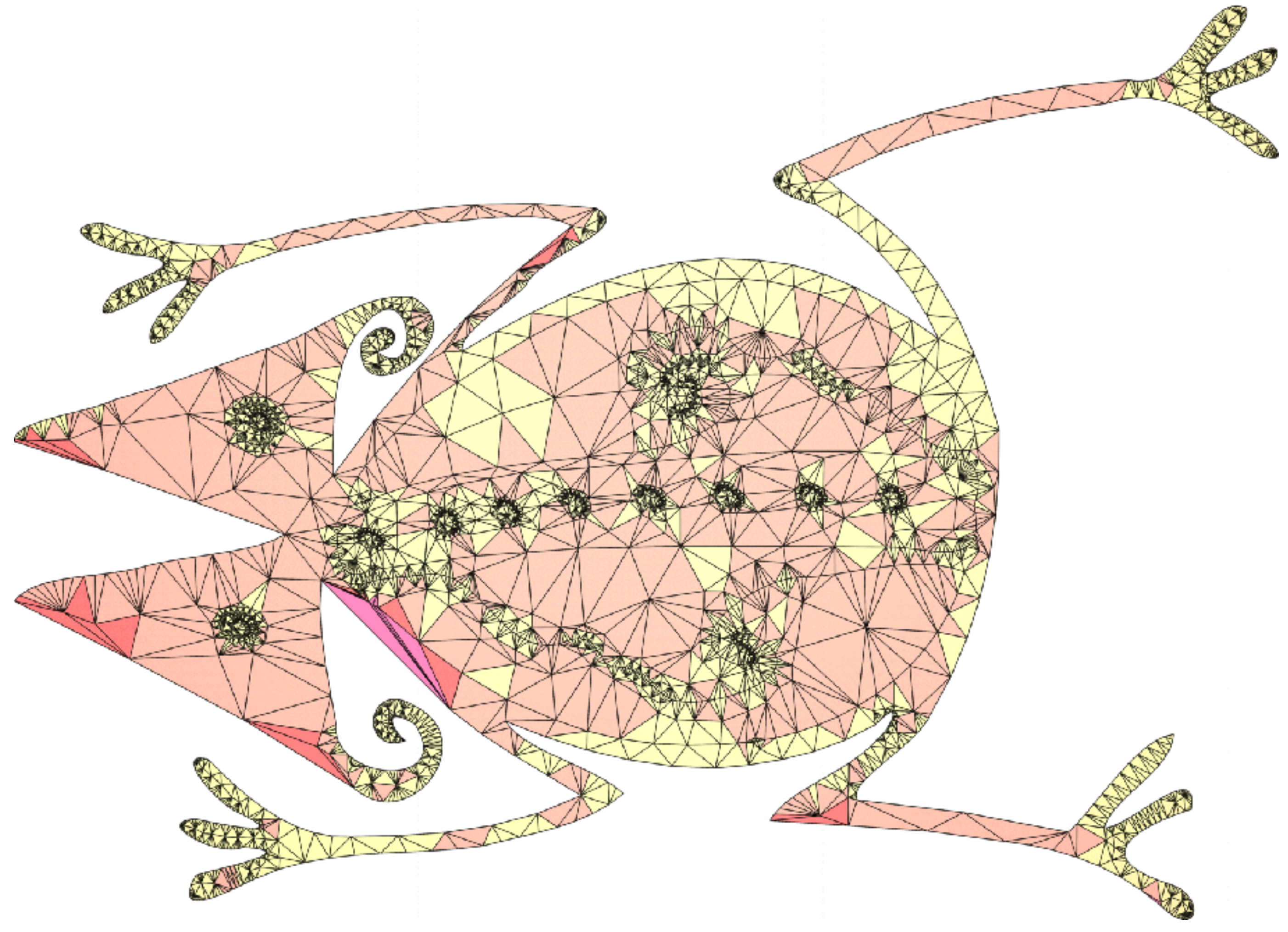
$$\sigma_E = \frac{\rho_E}{h_E}$$



Overview

$$k = \frac{\ln\left(B\hat{h}^{\hat{k}+1}\frac{\sigma_E^2}{\hat{\sigma}^2}\right) - \ln\, h_E}{\ln\, h_E}$$

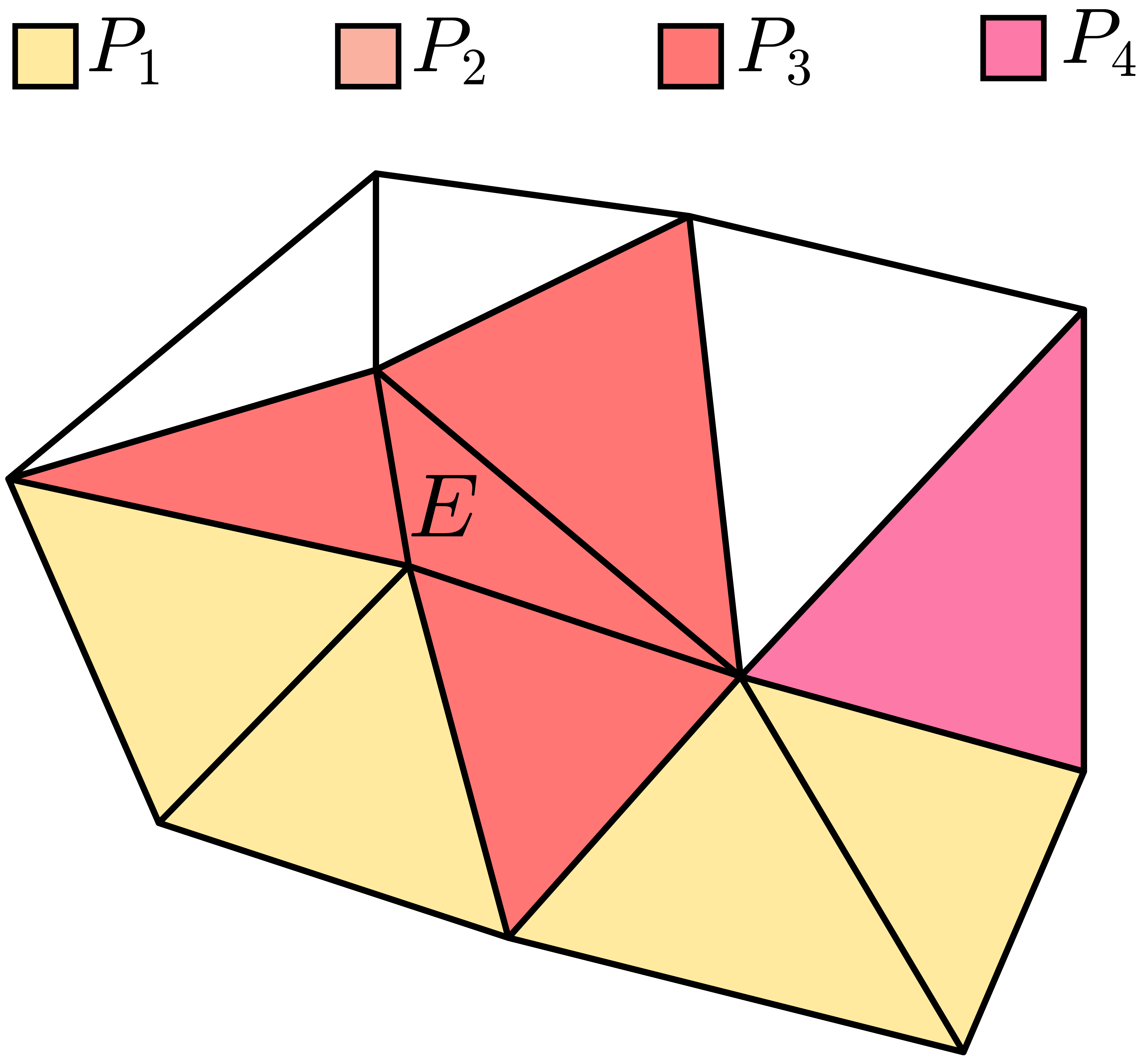
1. Use formula



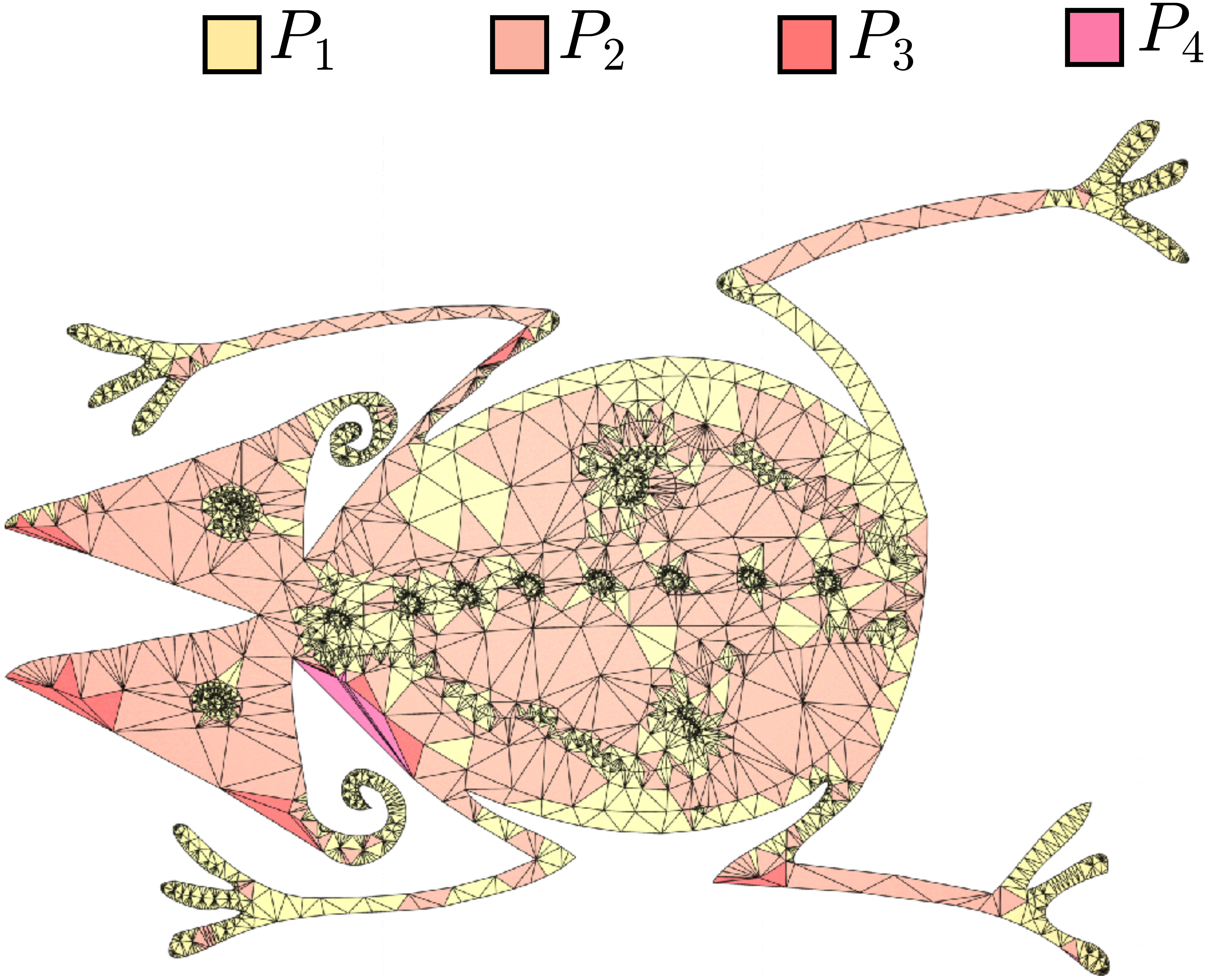
2. Propagate degrees

Degree Propagation

- For each element E
- Compute k_E using formula
- Increase the order (if necessary) of:
  - The element E
  - All edge/face neighbors



Degree Propagation

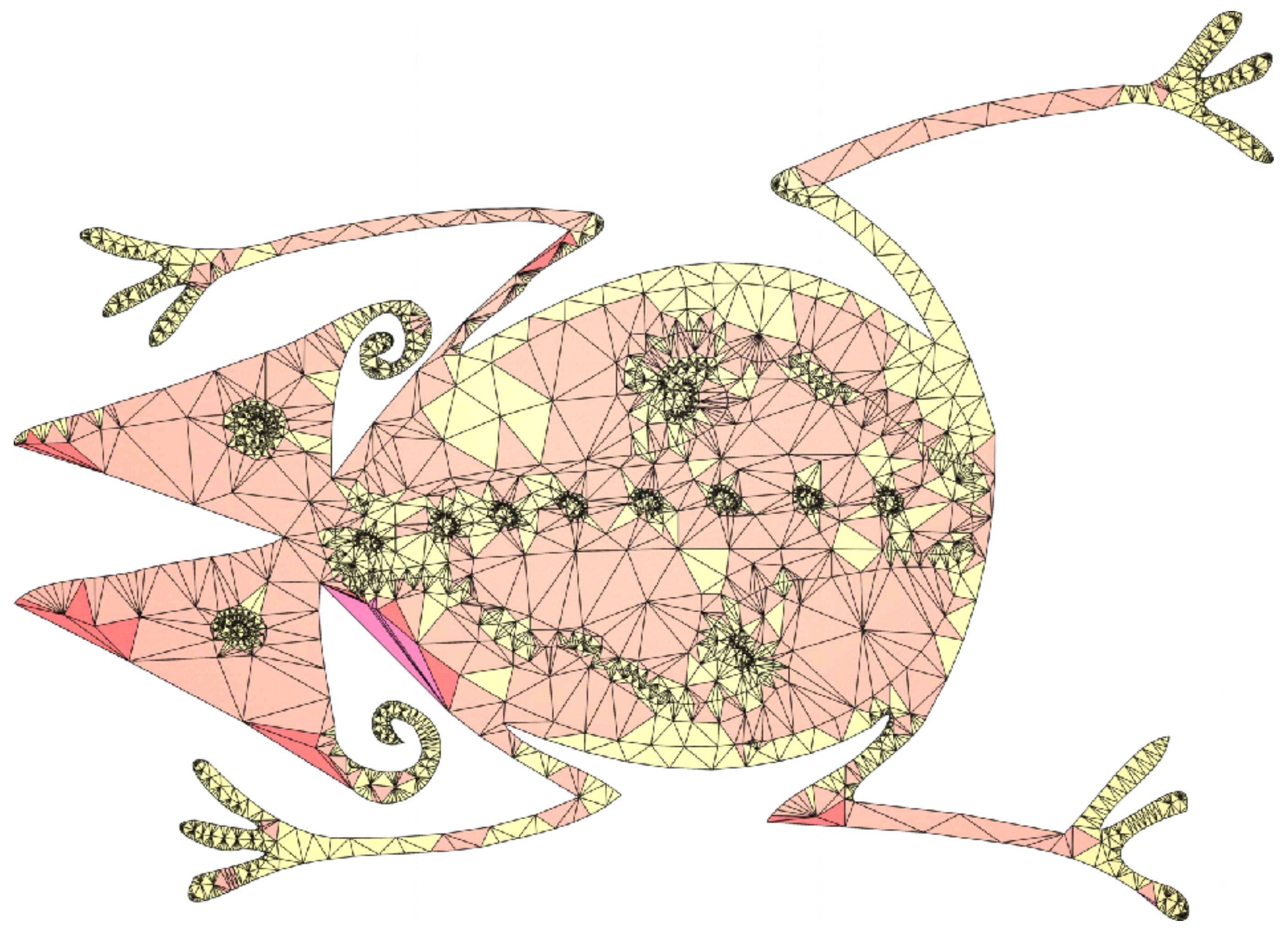


- For each element E
- Compute k_E using formula
- Increase the order (if necessary) of:
  - The element E
  - All edge/face neighbors

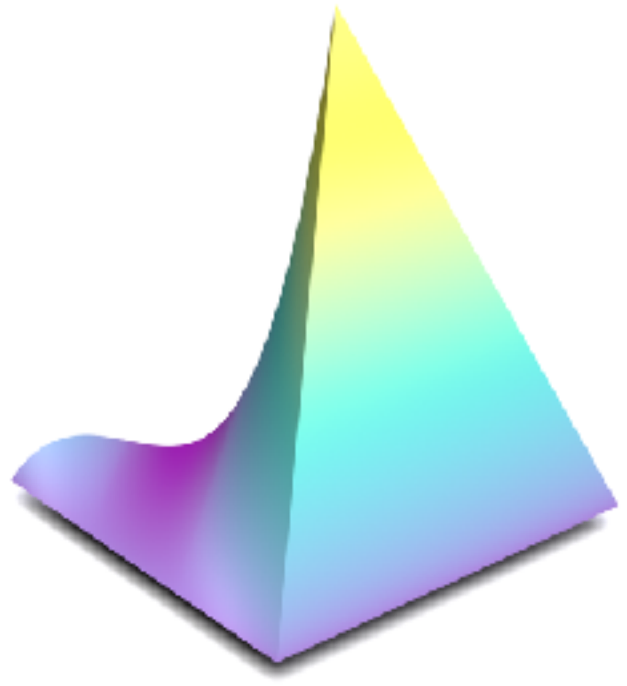
Overview

$$k = \frac{\ln\left(B\hat{h}^{\hat{k}+1}\frac{\sigma_E^2}{\hat{\sigma}^2}\right) - \ln\, h_E}{\ln\, h_E}$$

1. Use formula

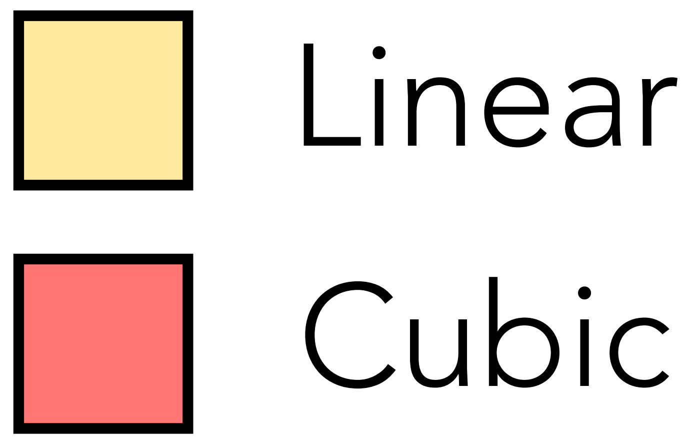


2. Propagate degrees



3. Construct C^0 basis

Building Continuous Basis

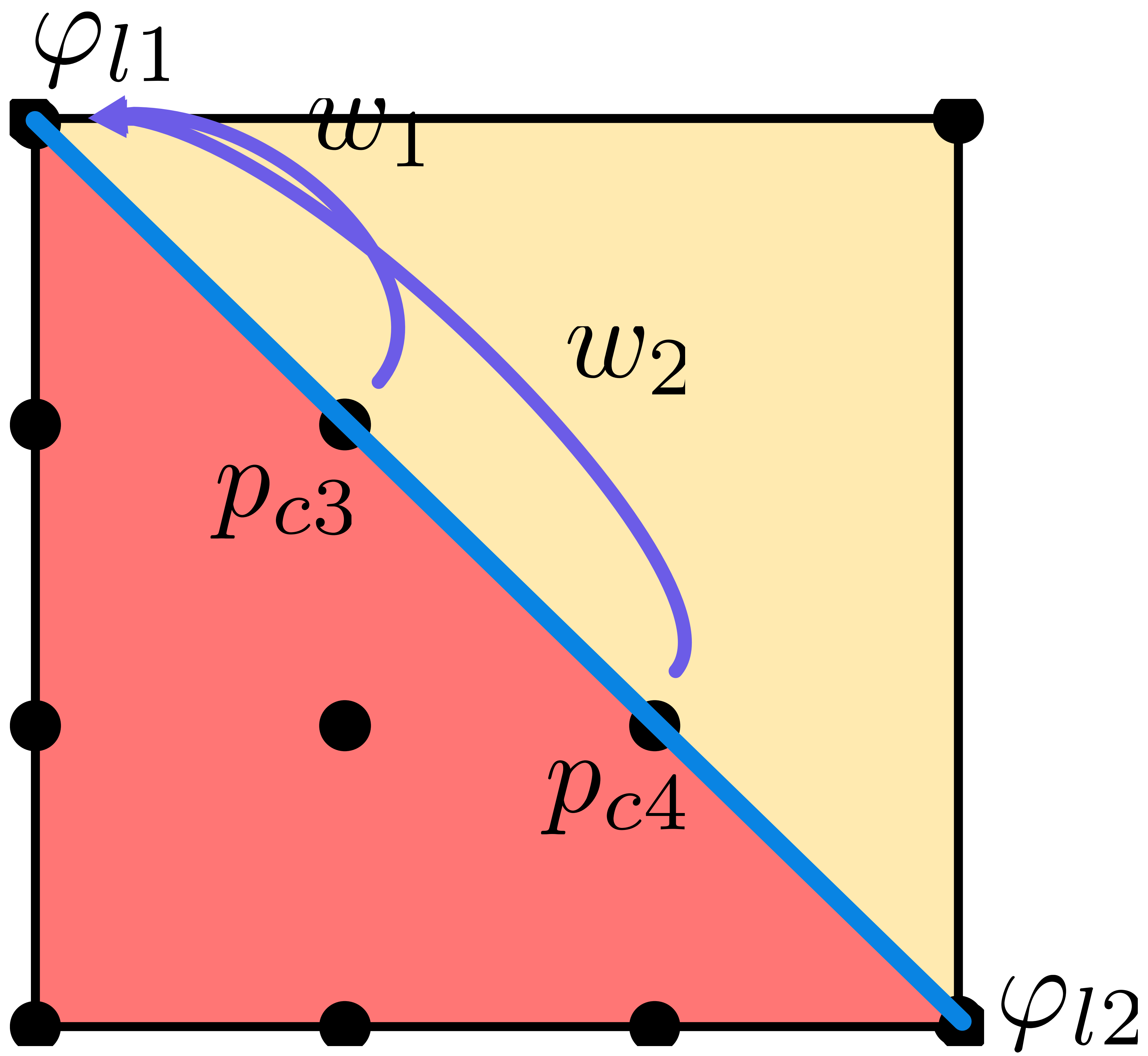


Linear

$$a + bt + 0t^2 + 0t^3$$

$$\varphi_{l1}(p_{c3}) = w_1 = \frac{2}{3}$$

$$\varphi_{l1}(p_{c4}) = w_2 = \frac{1}{3}$$



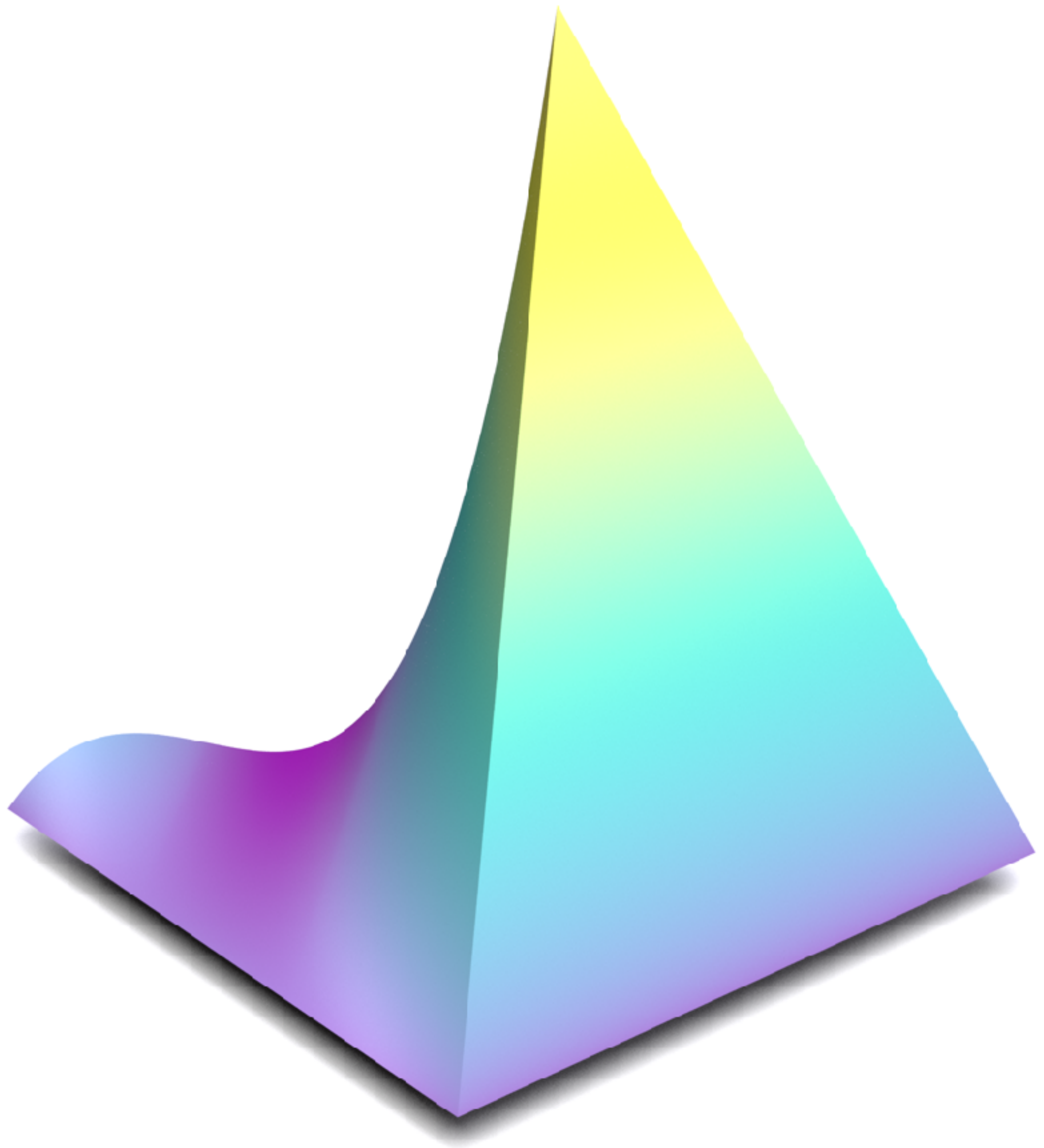
Building Continuous Basis

Linear

Cubic

Linear

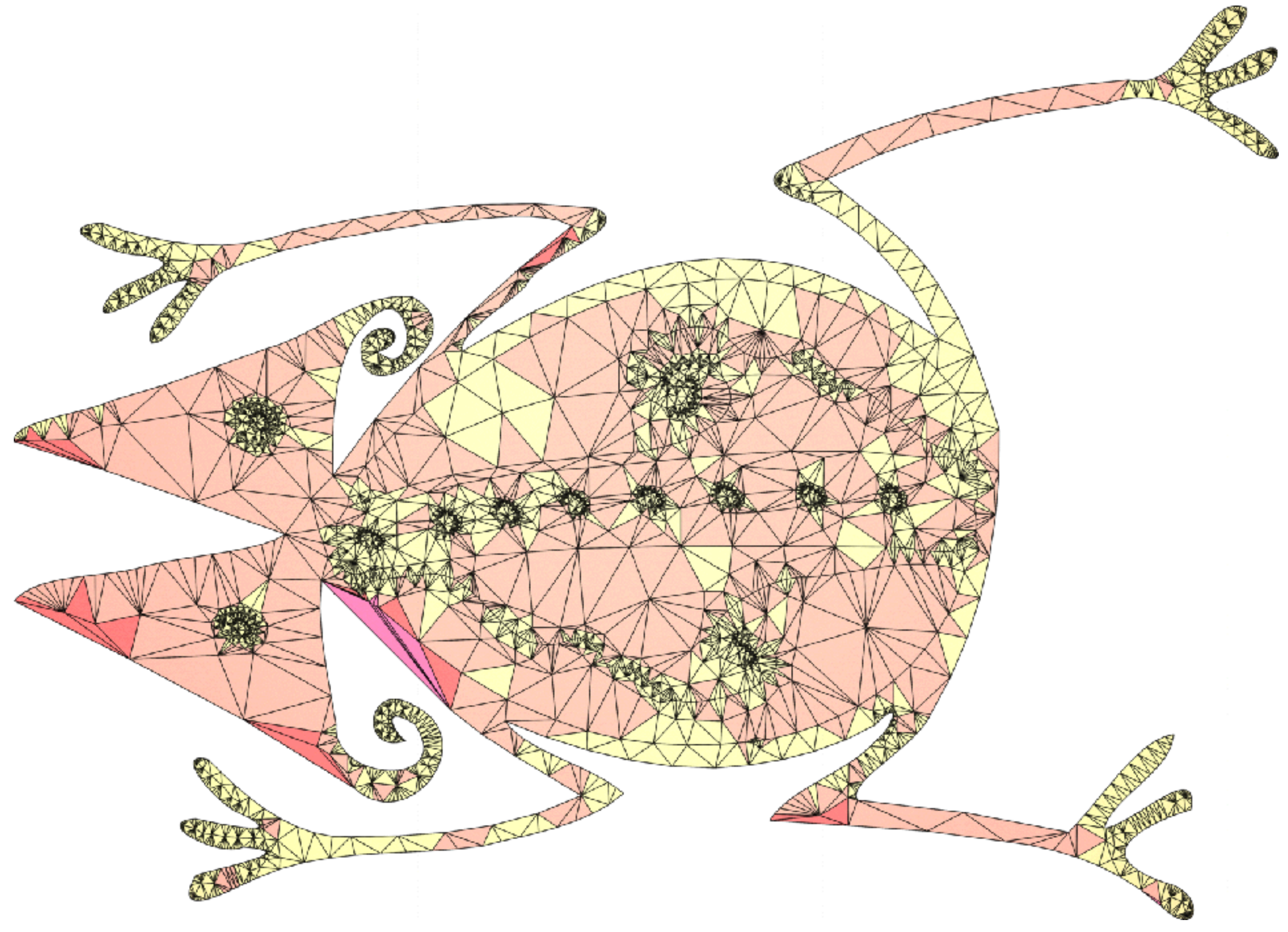
$$a + bt + 0t^2 + 0t^3$$



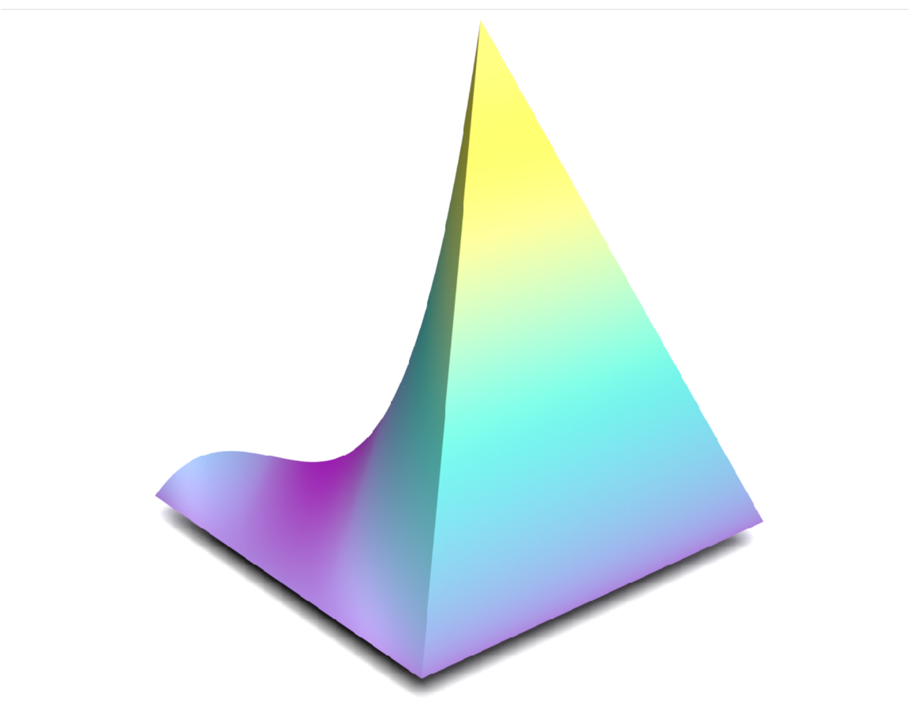
Overview

$$k = \frac{\ln\left(B\hat{h}^{\hat{k}+1}\frac{\sigma_E^2}{\hat{\sigma}^2}\right) - \ln\, h_E}{\ln\, h_E}$$

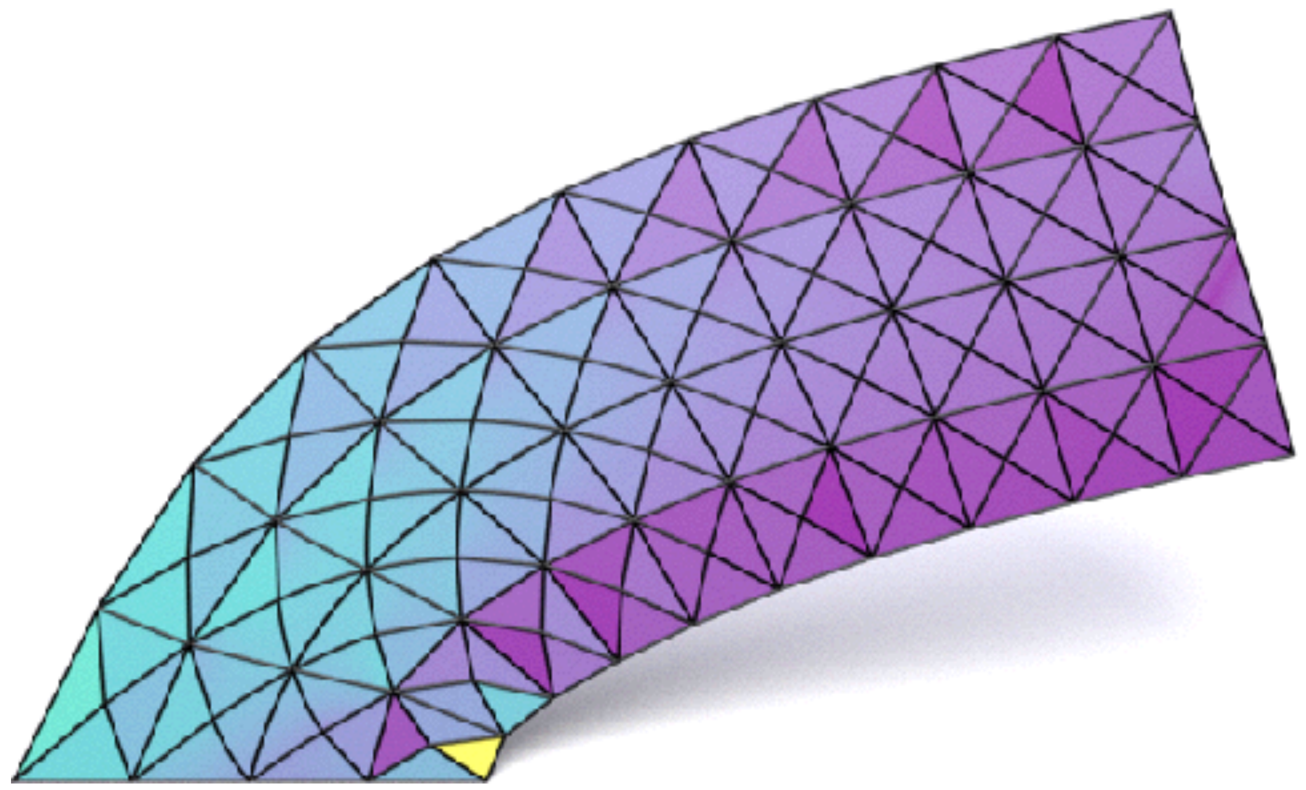
1. Use formula



2. Propagate degrees



3. Construct C^0 basis



4. Simulate!

Back to Laplace

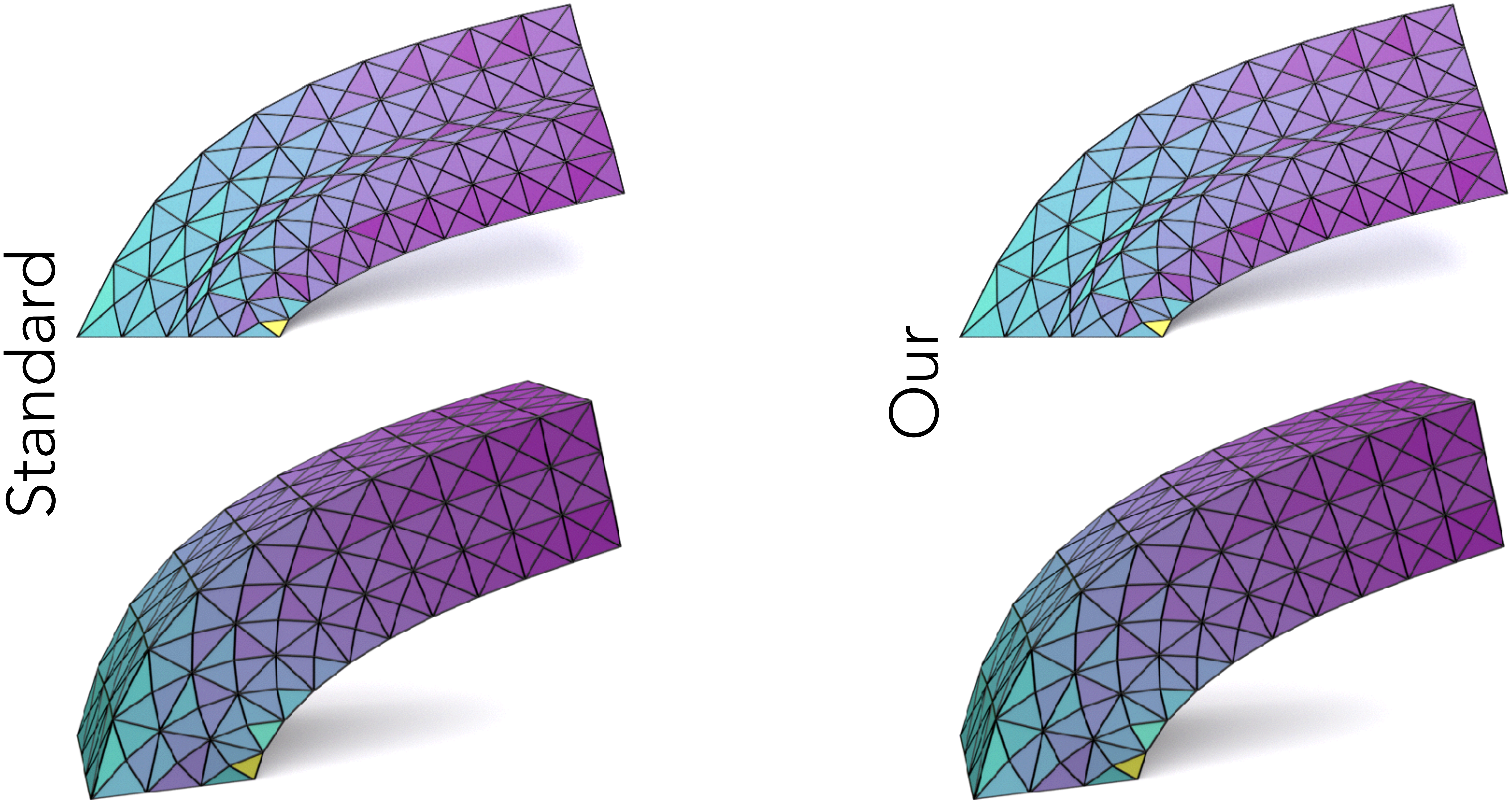
Back to Laplace

Back to Laplace

Standard

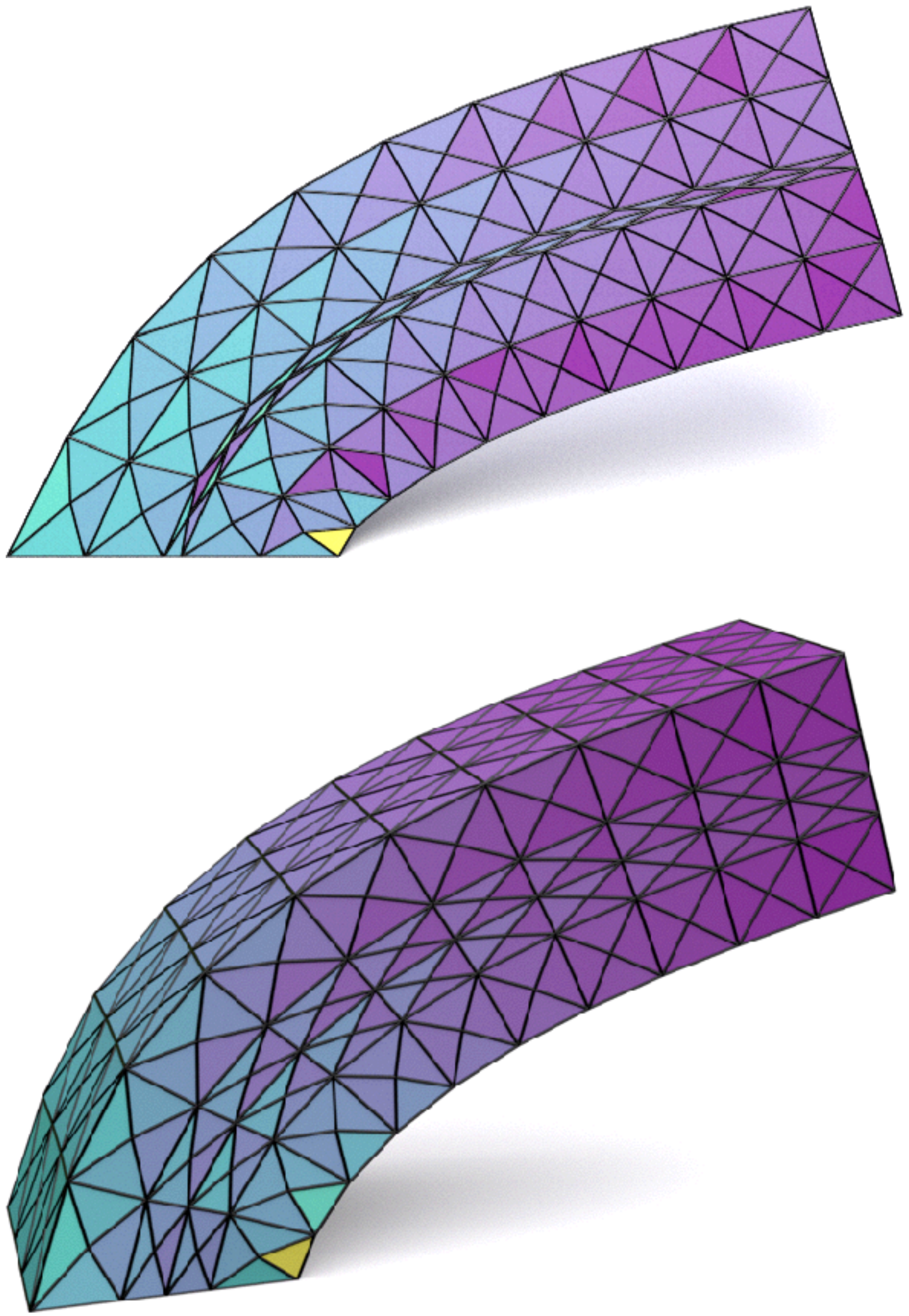
Our

Neo-Hookean Elasticity

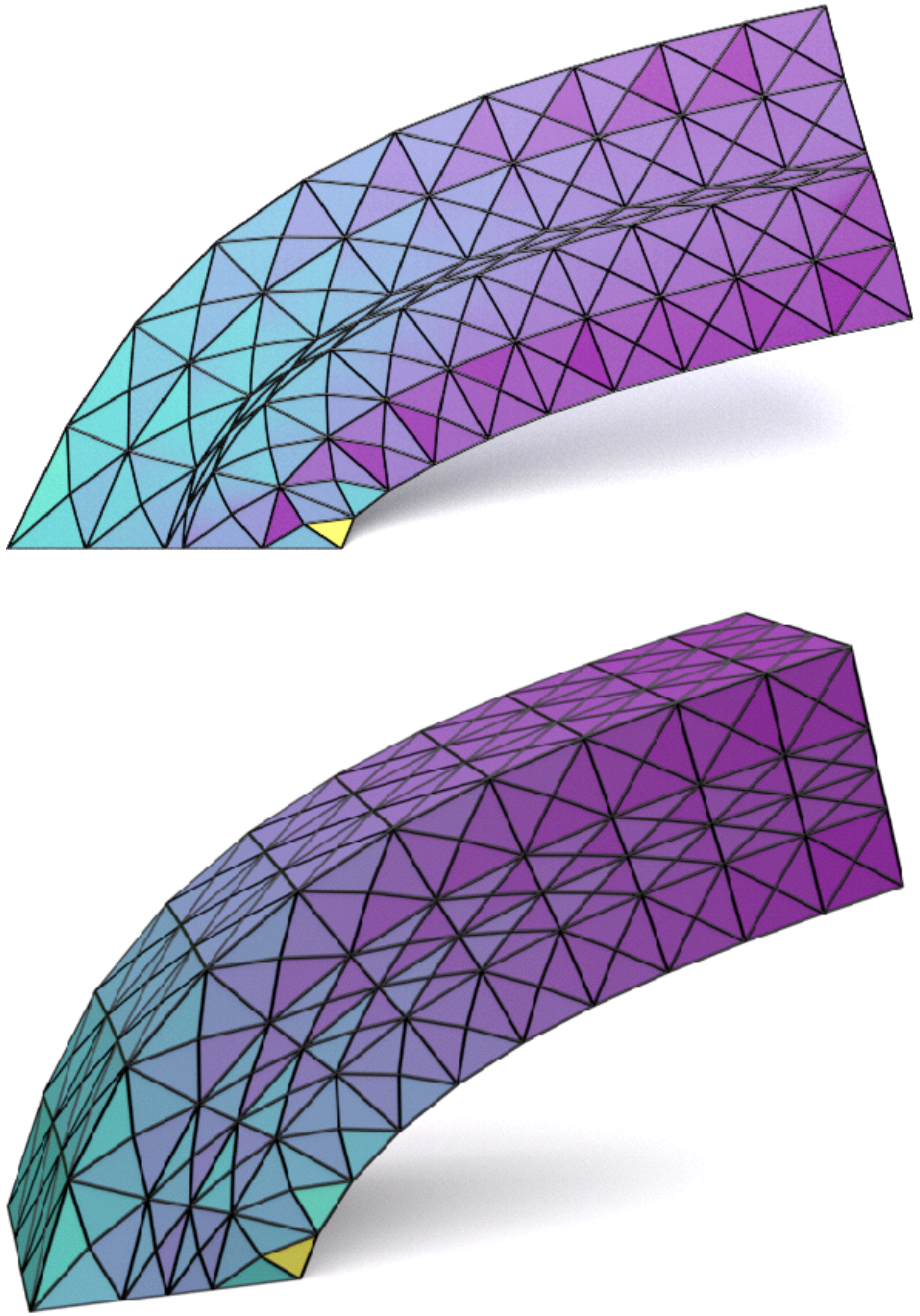


Neo-Hookean Elasticity

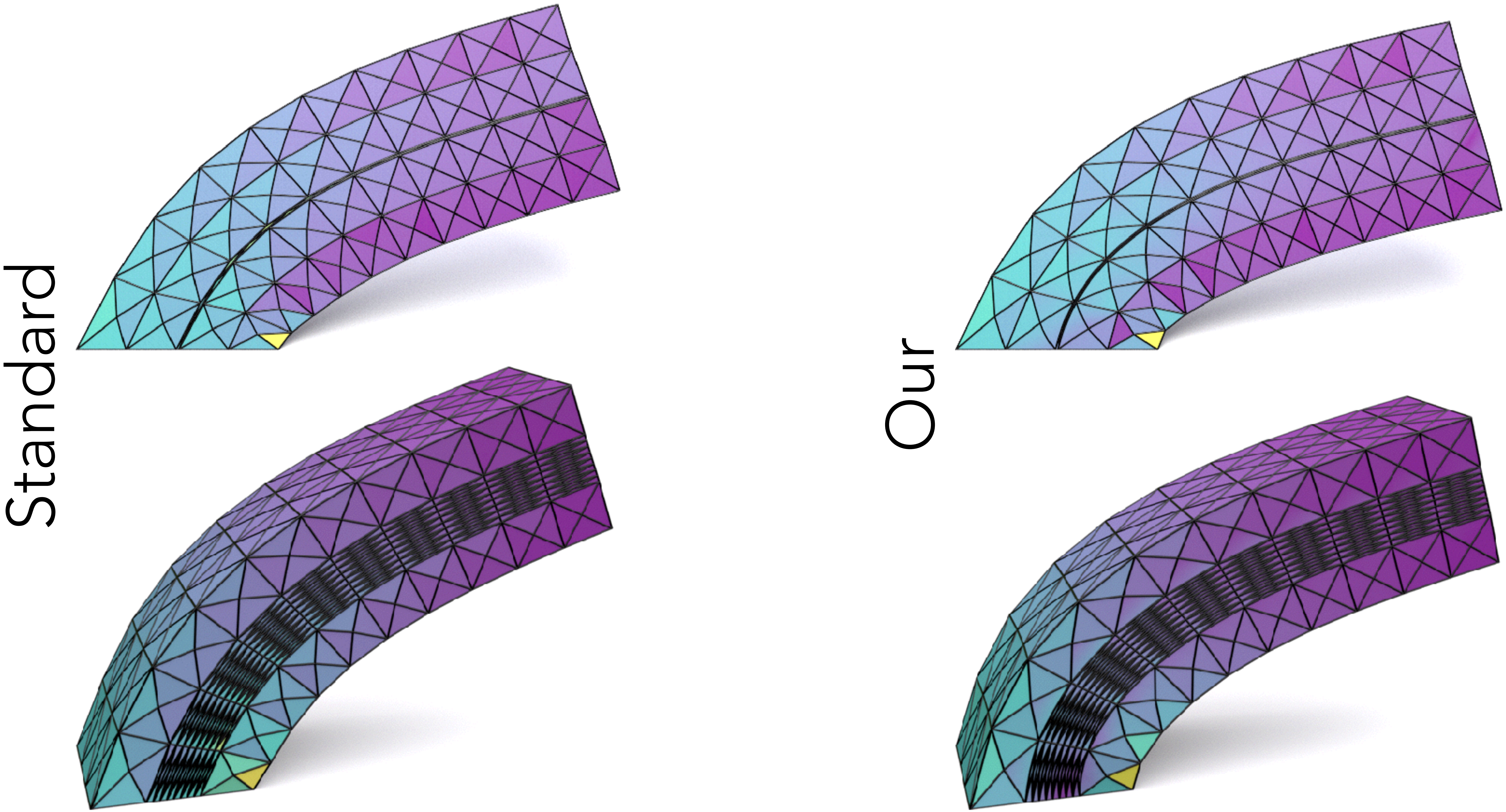
Standard



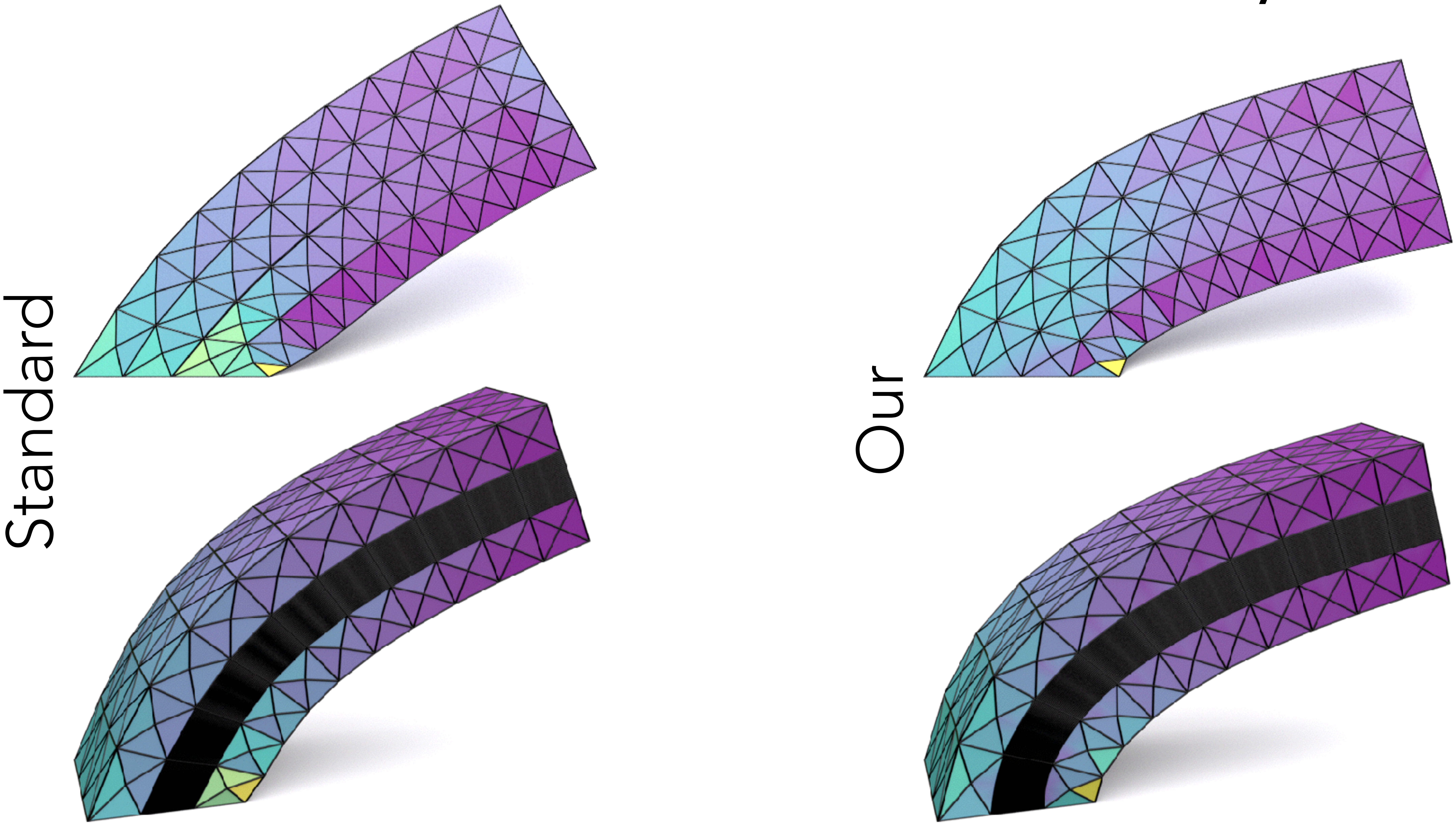
Our



Neo-Hookean Elasticity

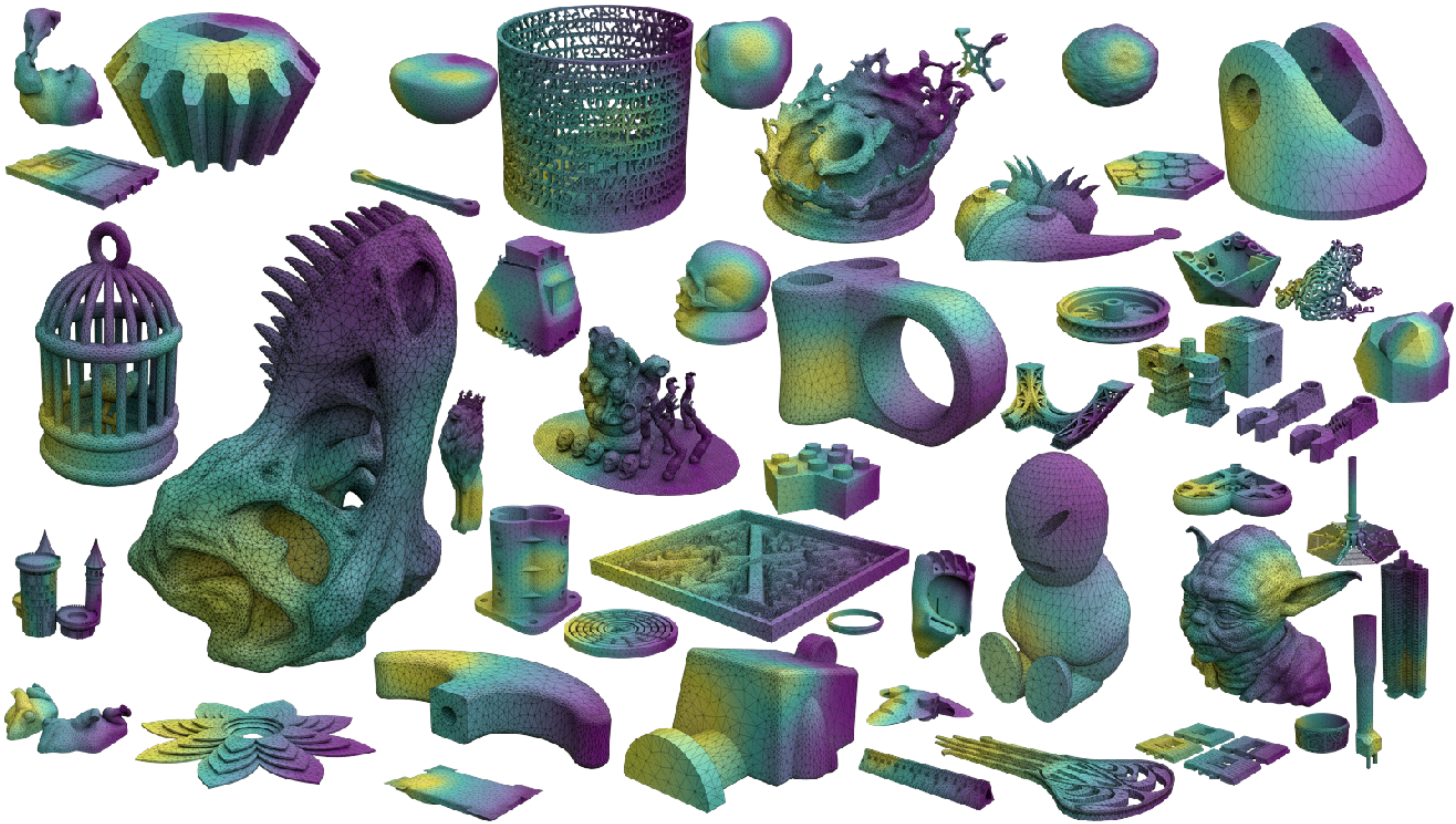


Neo-Hookean Elasticity



Large Dataset

- Thingi10k
  [Zhou 17]
- Tetwild
  [Hu 18]
- ~10k Optimized
- ~10k Not Optimized



How to Measure Errors?

$$e_h = \|u - u_h\|_0 \leq Ch^2 \|u\|_2$$

- Standard L_2 error estimate for linear elements

How to Measure Errors?

- Standard L_2 error estimate for linear elements

$$e_h = \boxed{\|u - u_h\|_0} \leq Ch^2 \|u\|_2$$

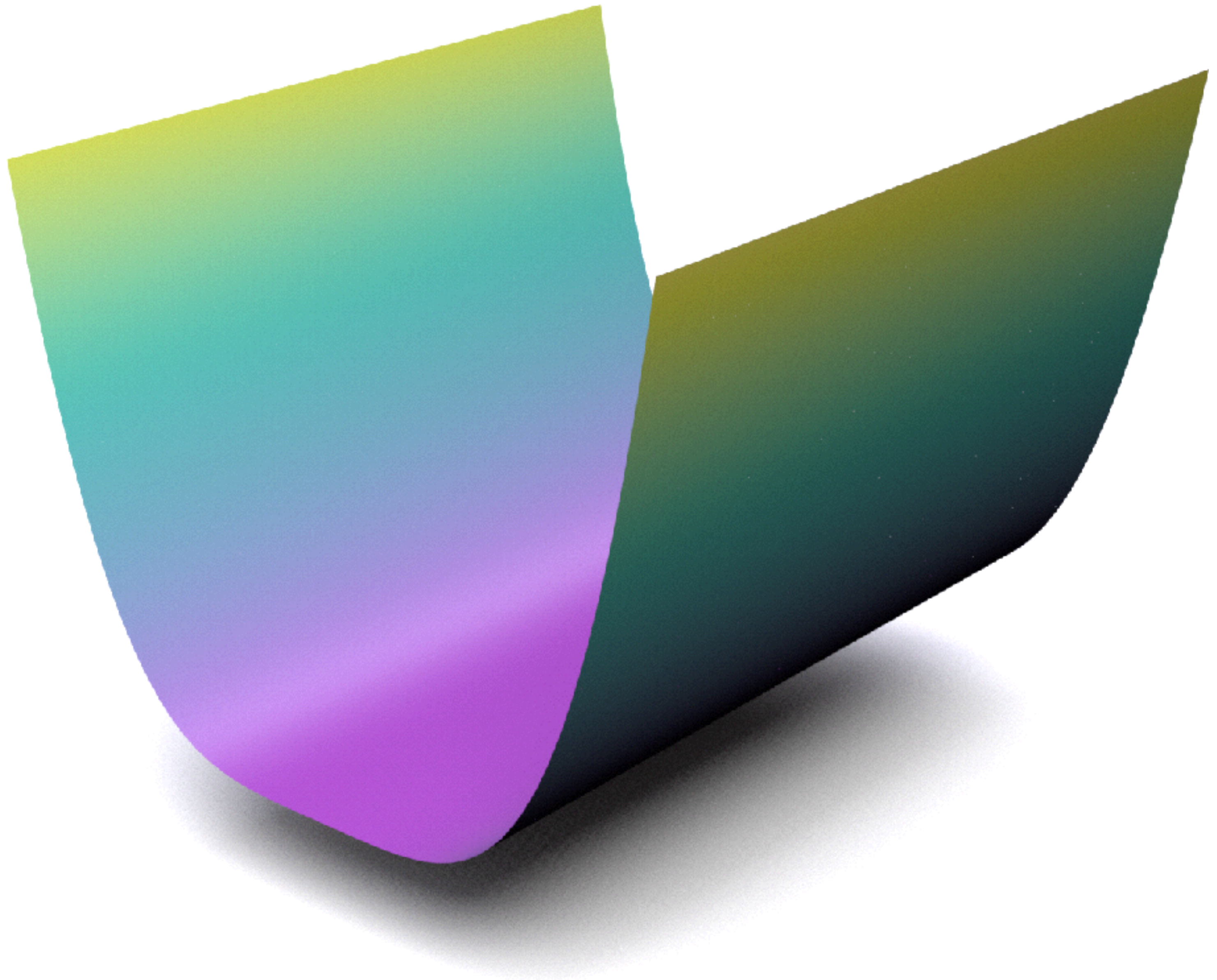
L_2 norm or average error

FEM Error Estimate

- Standard L_2 error estimate for linear elements

$$e_h = \|u - u_h\|_0 \leq Ch^2 \|u\|_2$$

Exact solution

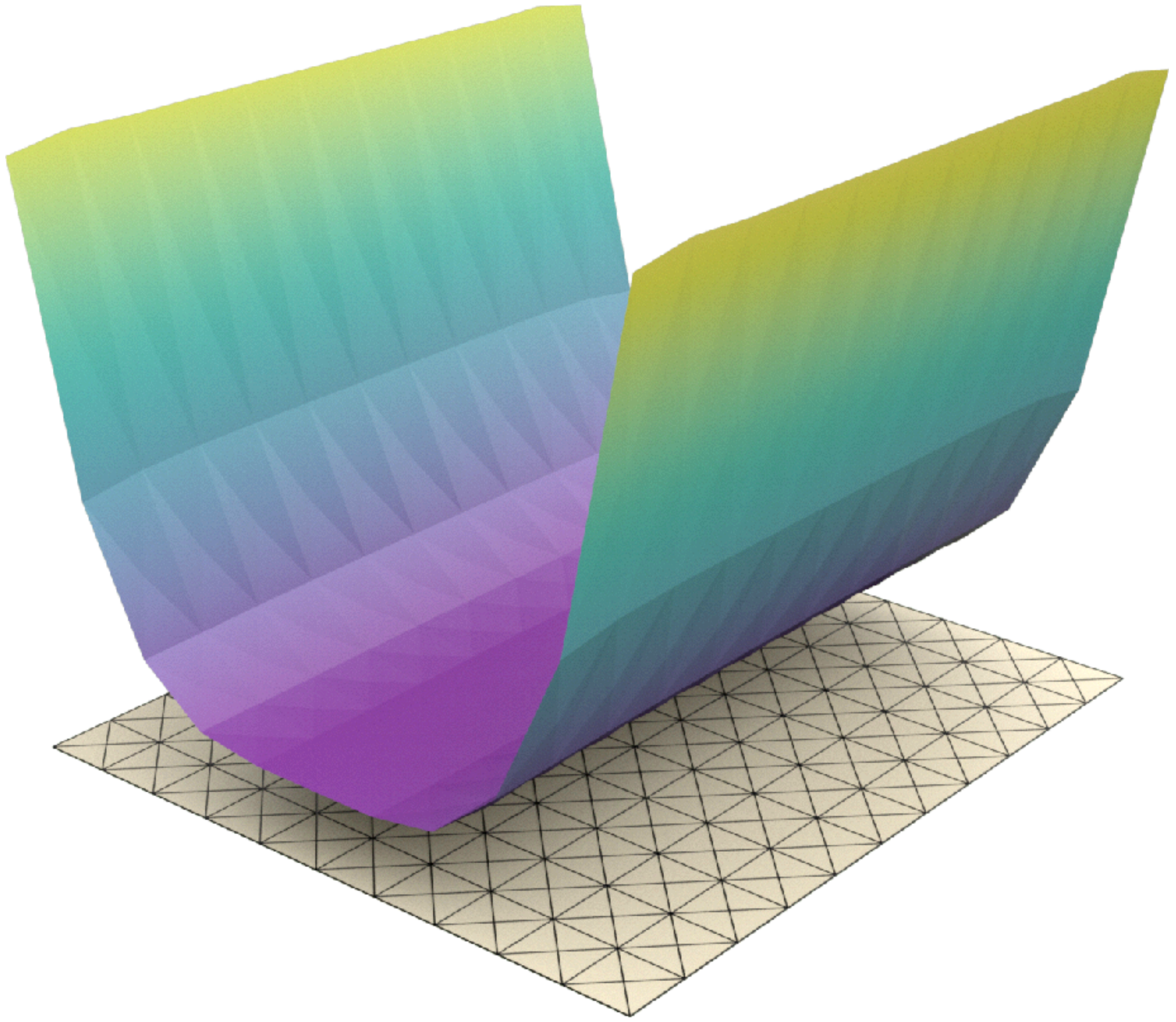


FEM Error Estimate

- Standard L_2 error estimate for linear elements

$$e_h = \|u - \boxed{u_h}\|_0 \leq Ch^2 \|u\|_2$$

Approximated solution



How to Measure Errors?

- Standard L_2 error estimate for linear elements

$$e_h = \|u - u_h\|_0 \leq C\boxed{h^2}\|u\|_2$$

- Different h for every model!

How to Measure Errors?

- Standard L_2 error estimate for linear elements

$$e_h = \|u - u_h\|_0 \leq Ch^2 \|u\|_2$$

  - Different h for every model!

- L_2 Efficiency

$$E_{L_2} = \frac{\|u - u_h\|_0}{h^2}$$

How to Measure Errors?

- Standard L_2 error estimate for linear elements

$$e_h = \|u - u_h\|_0 \leq Ch^2 \|u\|_2$$

  - Different h for every model!

- L_2 Efficiency

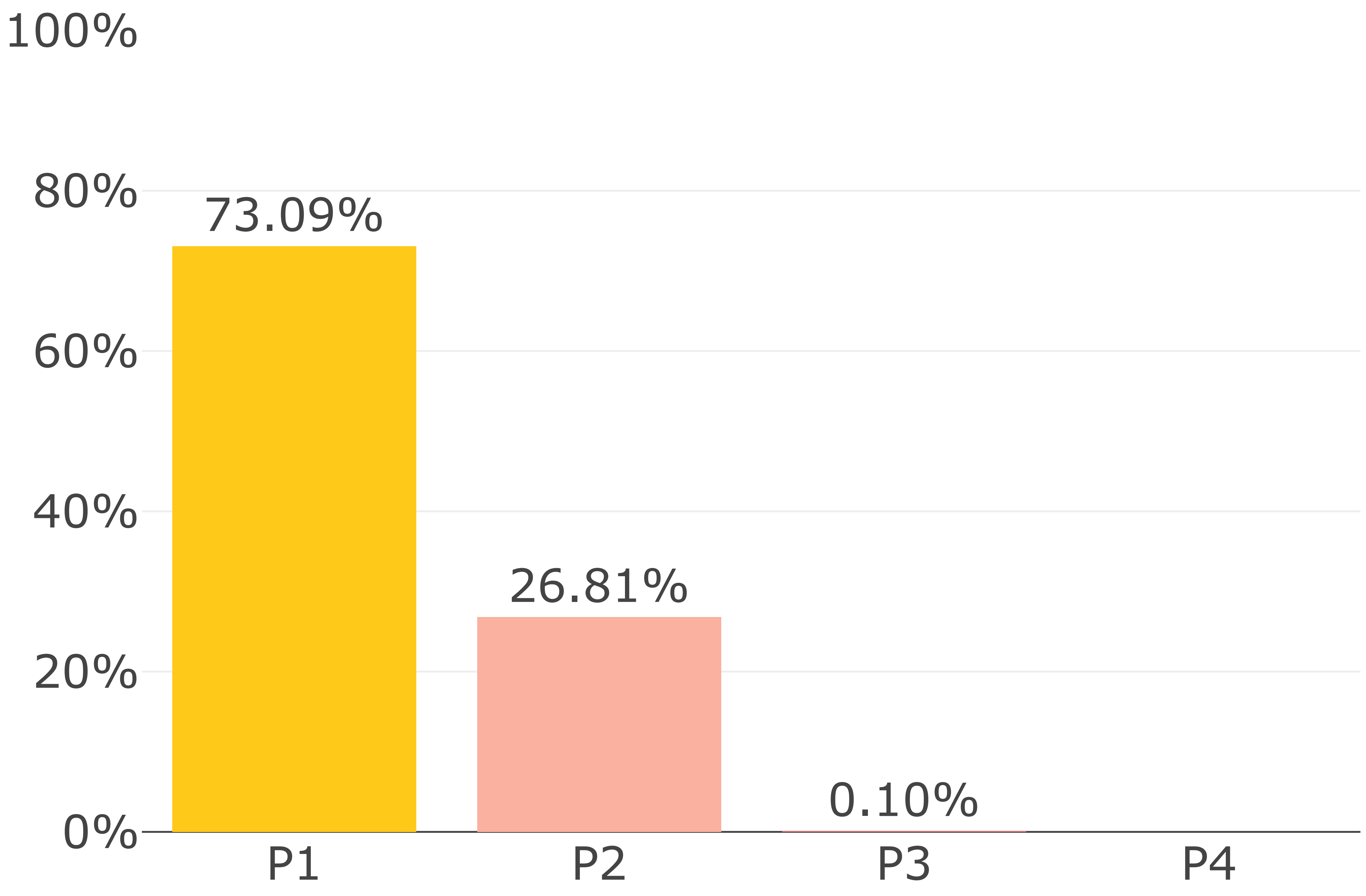
$$E_{L_2} = \frac{\|u - u_h\|_0}{h^2}$$

Small values are good!

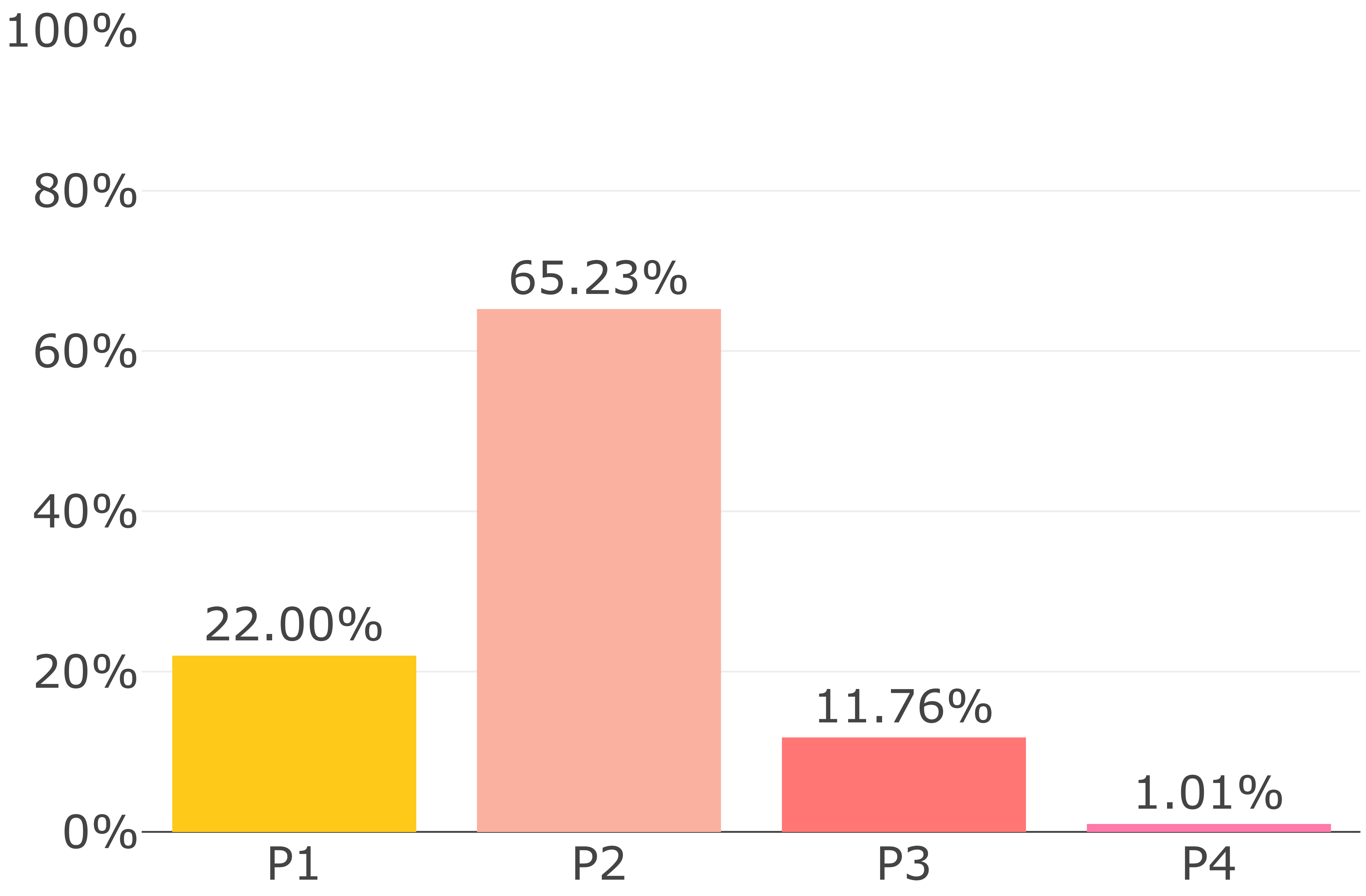
Efficiency


0.5
0.4
0.3
0.2
0.1
0
P1
Ours
0
1
2
3
4
5
6
7
L2 efficiency
Optimized
Not Optimized

Degree Distribution



Optimized



Not Optimized

Number of DOF

Timings

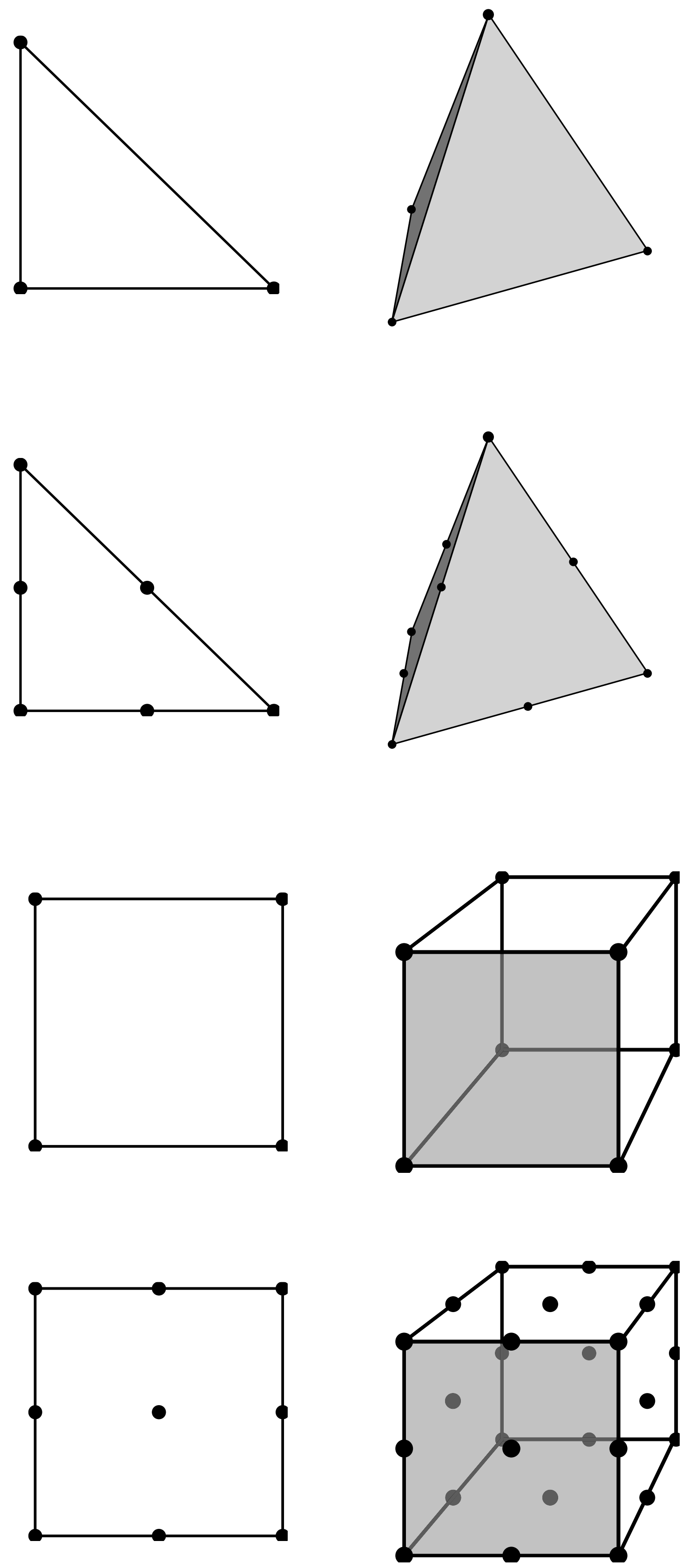
Optimized

Not Optimized

Overall Time (Meshing + Simulation)

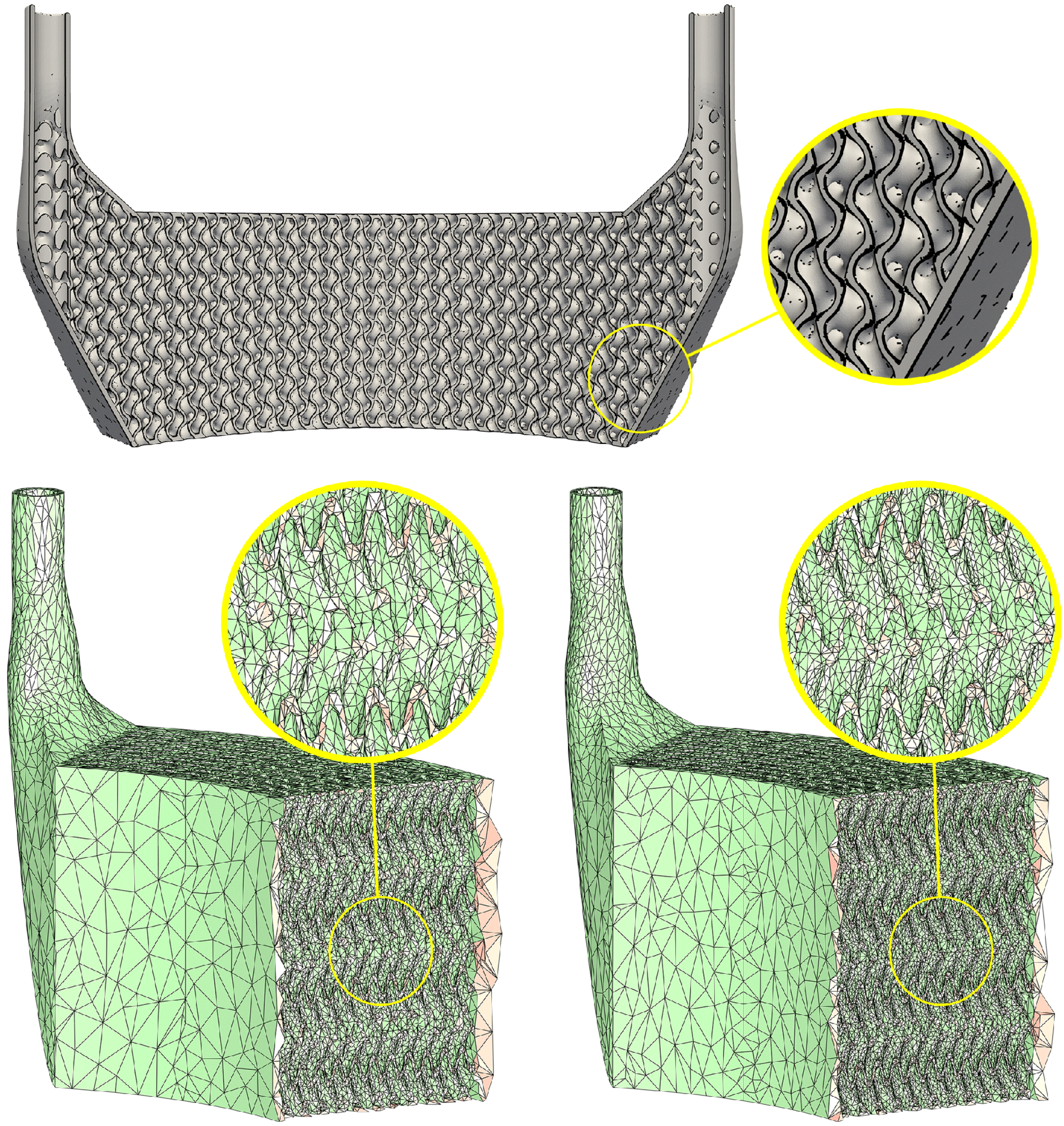
Overall Time (Meshing + Simulation)

Future Work



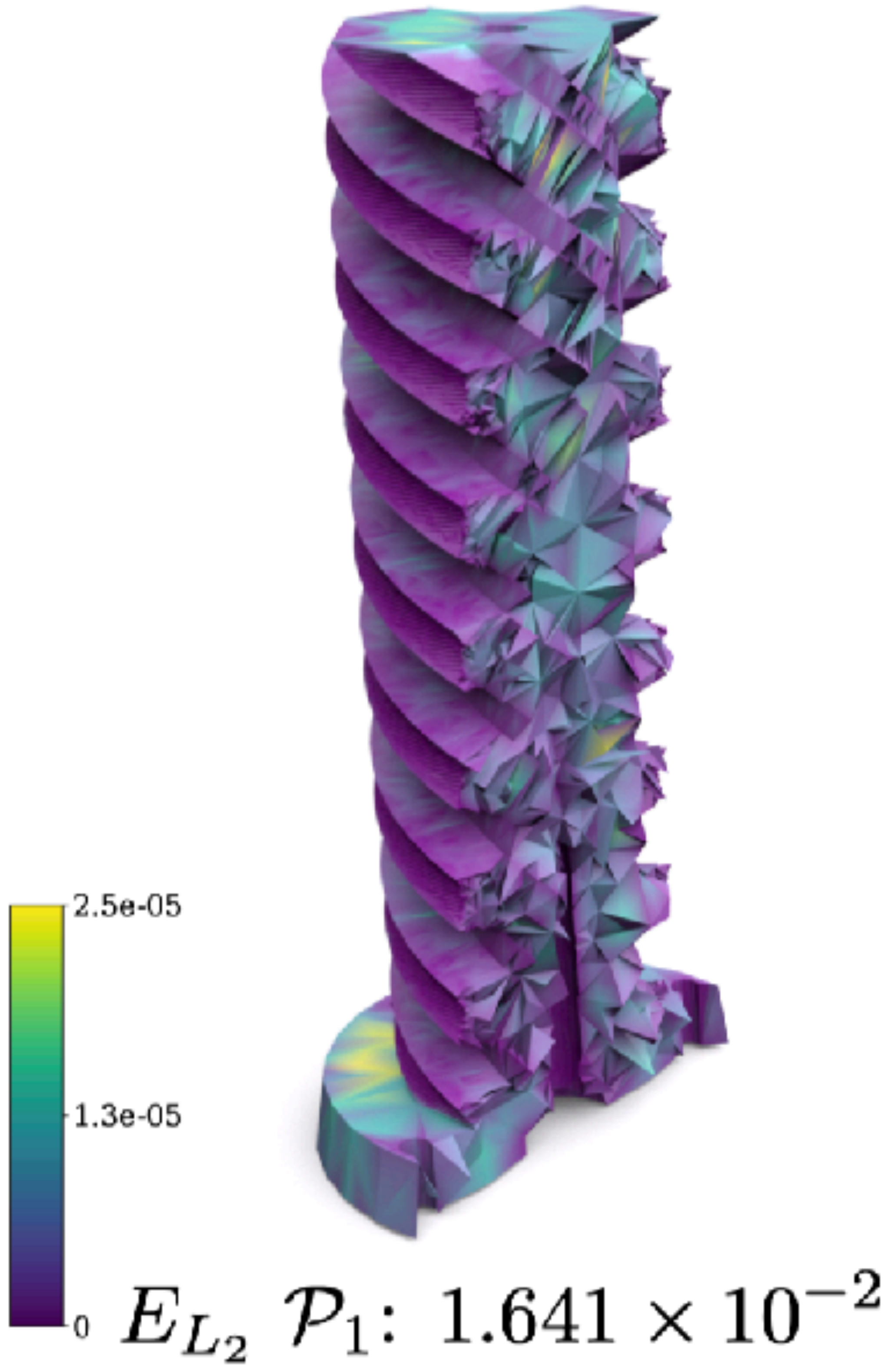
Analysis for elliptic PDEs only.
Does it make a difference for
Contacts or time-dependent problems?

Maybe



Meshing still takes way longer than
the FEM solve.
Can we make it real-time?

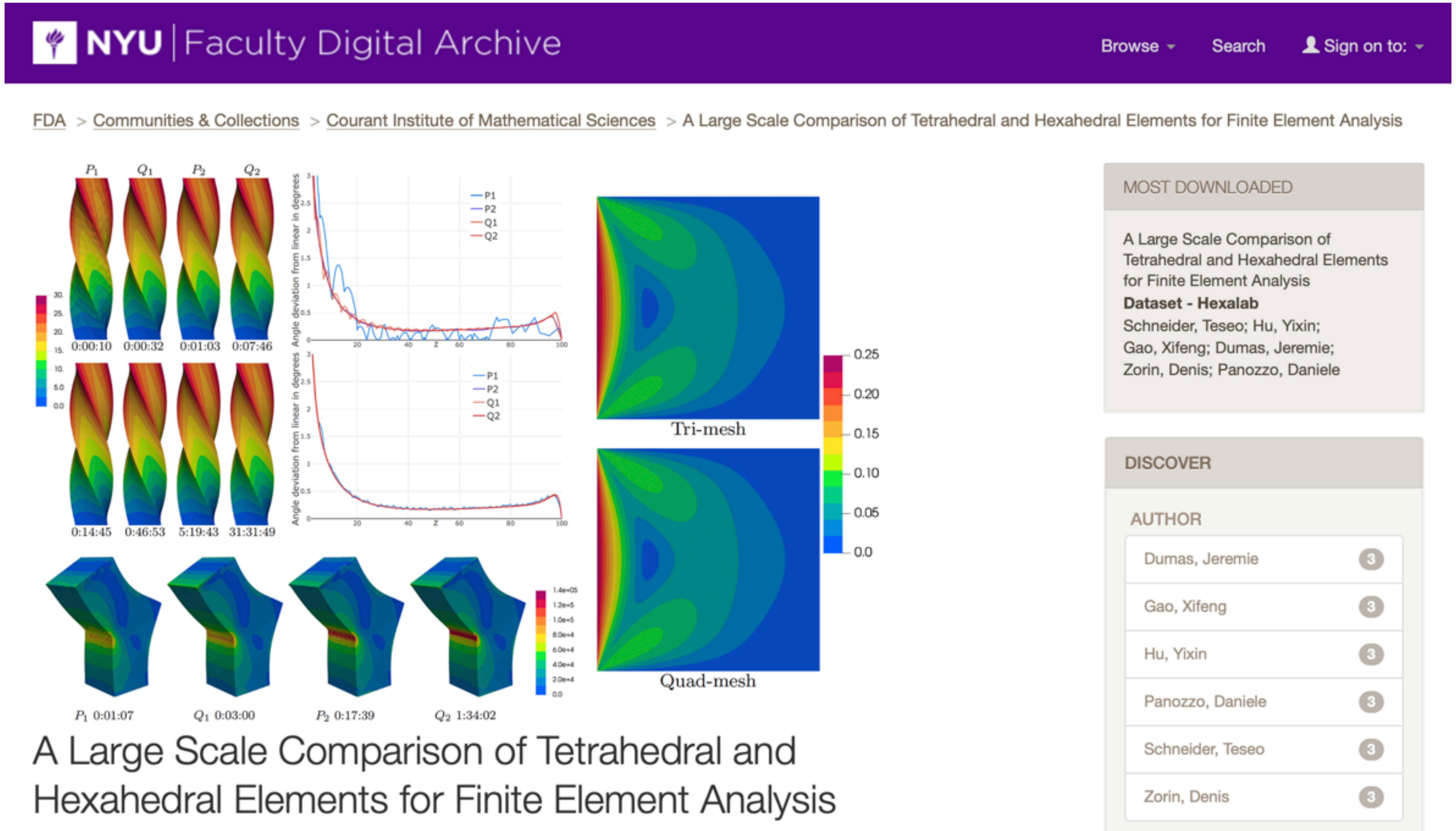
Maybe



Can we use a similar strategy
to limit/avoid remeshing in
dynamic simulations?

Why not?

Large Scale Comparison



https://archive.nyu.edu/handle/2451/44221