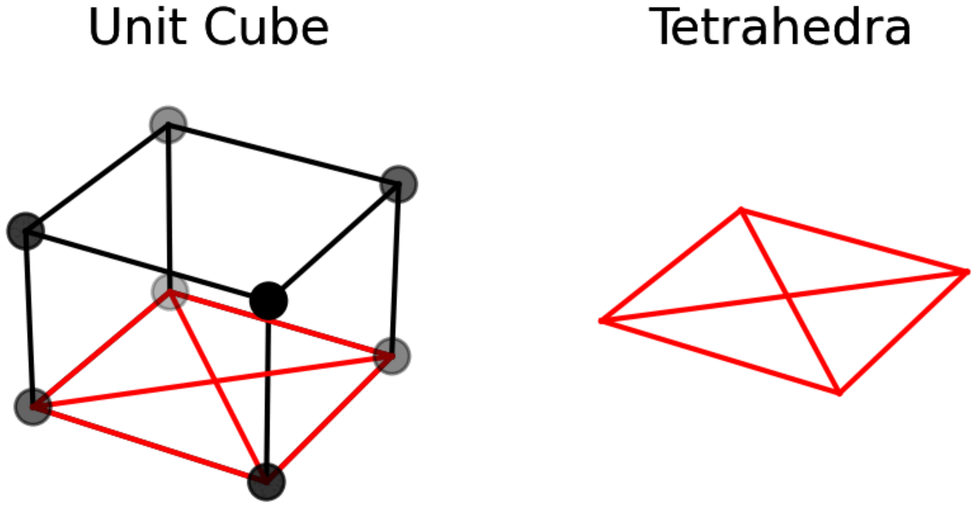
Up to 58 Tets/Hex to untangle Hex meshes

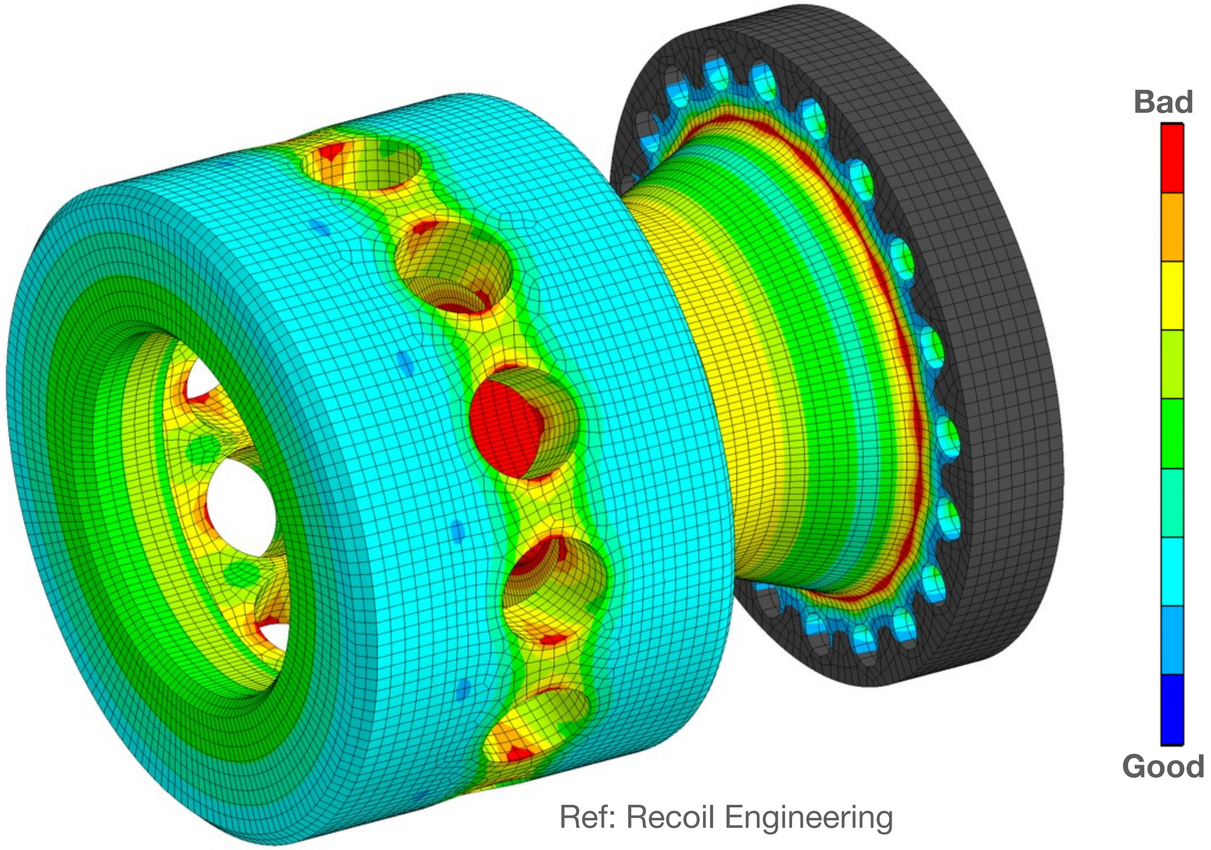
Bachelor Thesis



Luca Schaller, 3.11.22

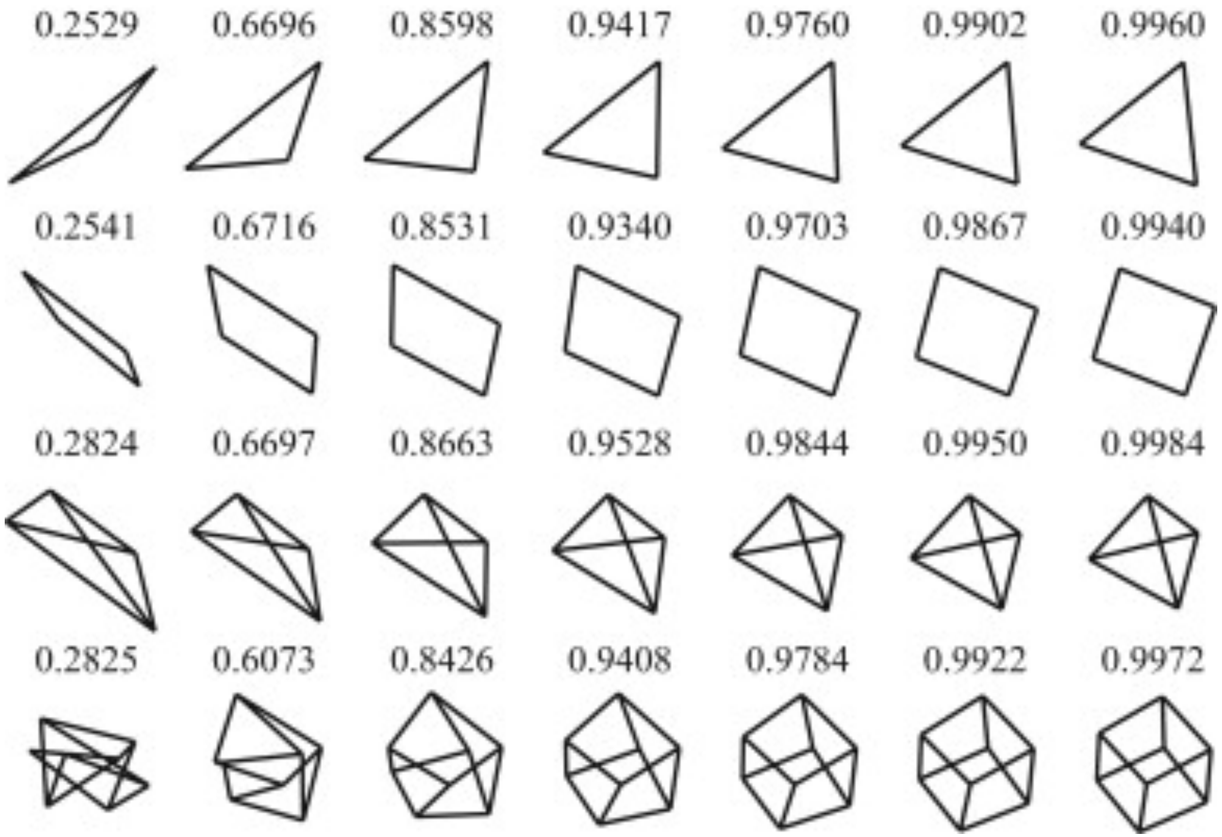
Motivation

Example: Finite Element Analysis



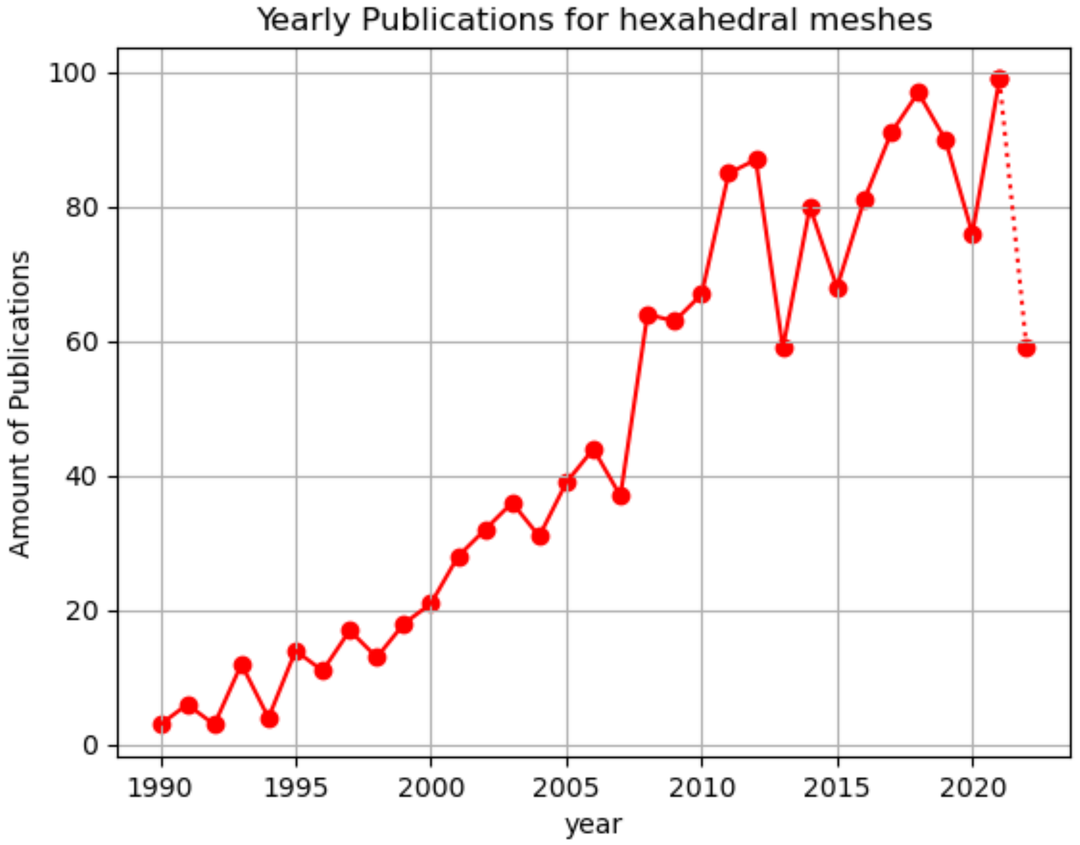
Ref: Recoil Engineering

Motivation



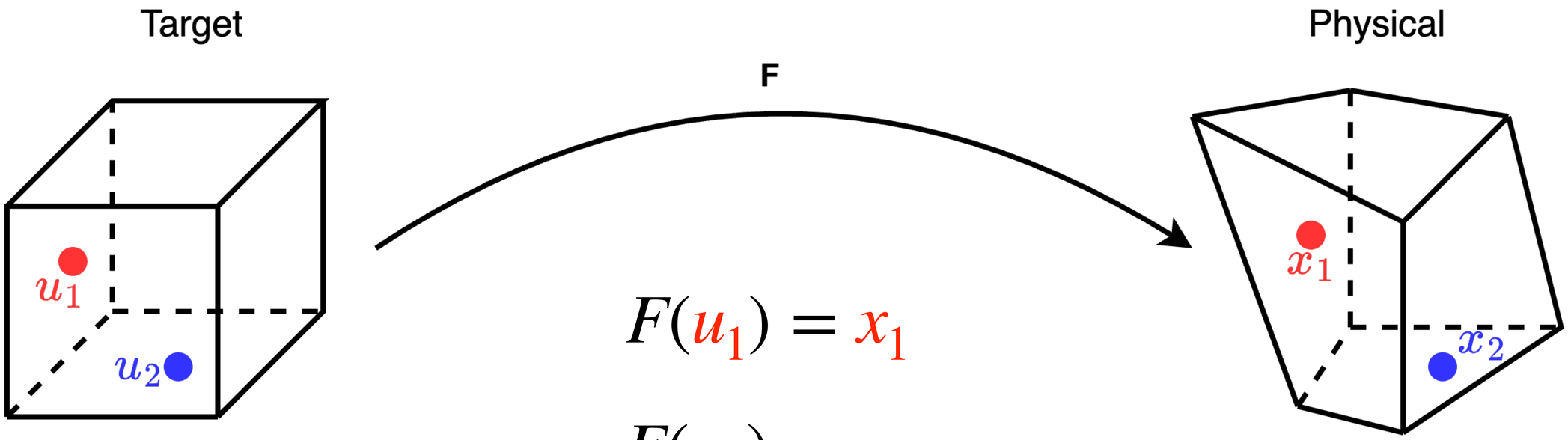
Ref: Dimitris Varziotis et al

Motivation


Yearly Publications for hexahedral meshes
Amount of Publications
year
0
20
40
60
80
100
1990
1995
2000
2005
2010
2015
2020

Trilinear Map

Injectivity

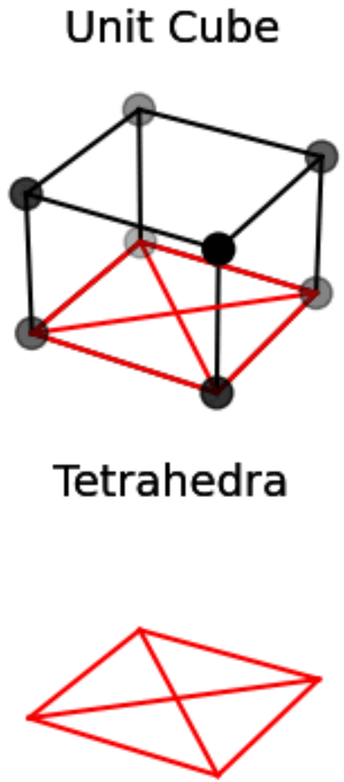


$$F(u_1) = x_1$$

$$F(u_2) = x_2$$

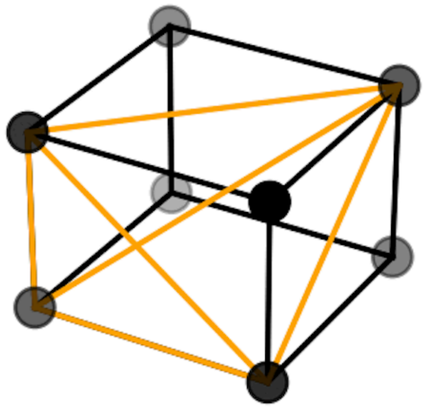
$$x_1 = x_2 \iff u_1 = u_2$$

58 Tets / Hex

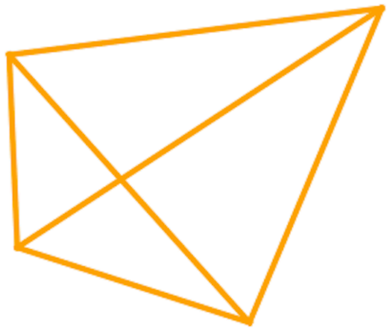


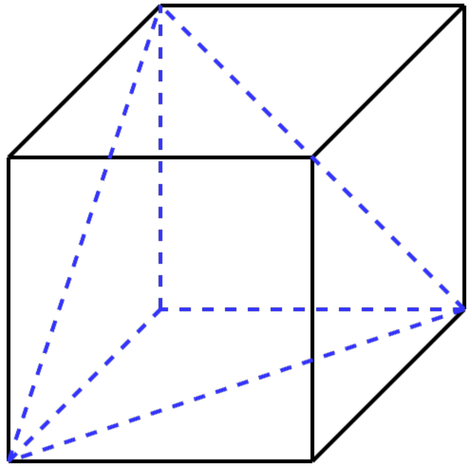
Tetrahedra Generation

Unit Cube



Tetrahedra





Validity

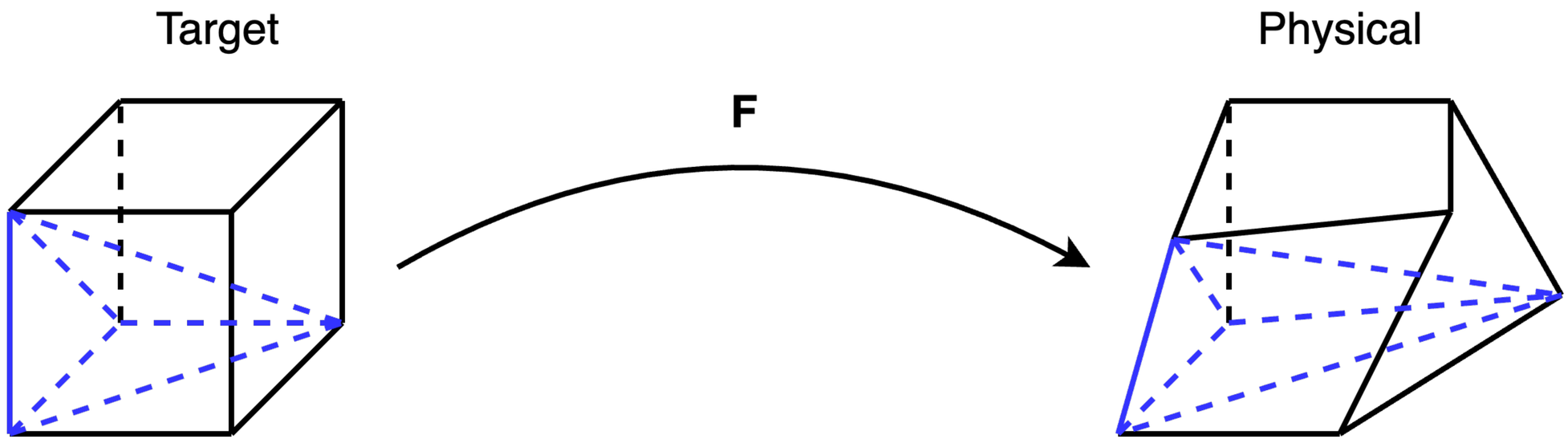
Necessary: 8 Corner Tetrahedra
Sufficient: all 58 Tetrahedra

Ushakova 2011

Unit Cube

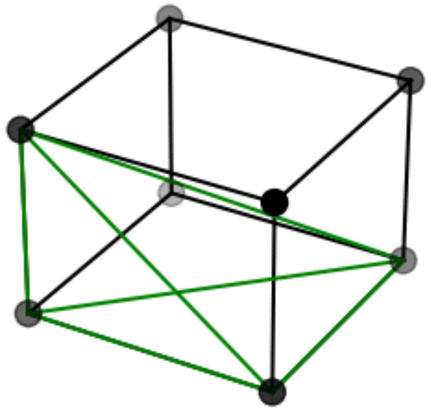
Tetrahedra

Situation

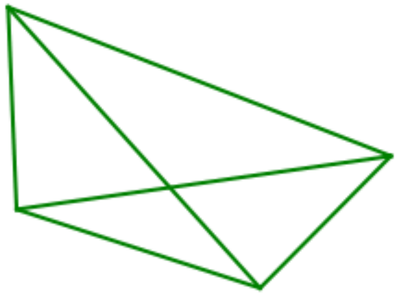


Target vs. Physical Space

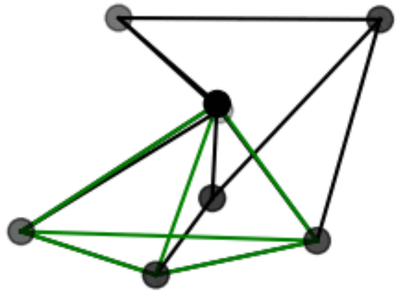
Target Hexahedron



Target Tetrahedron



Physical Hexahedron



Physical Tetrahedron

Untangling using Foldover-Free Maps

Garanzha & al 2021

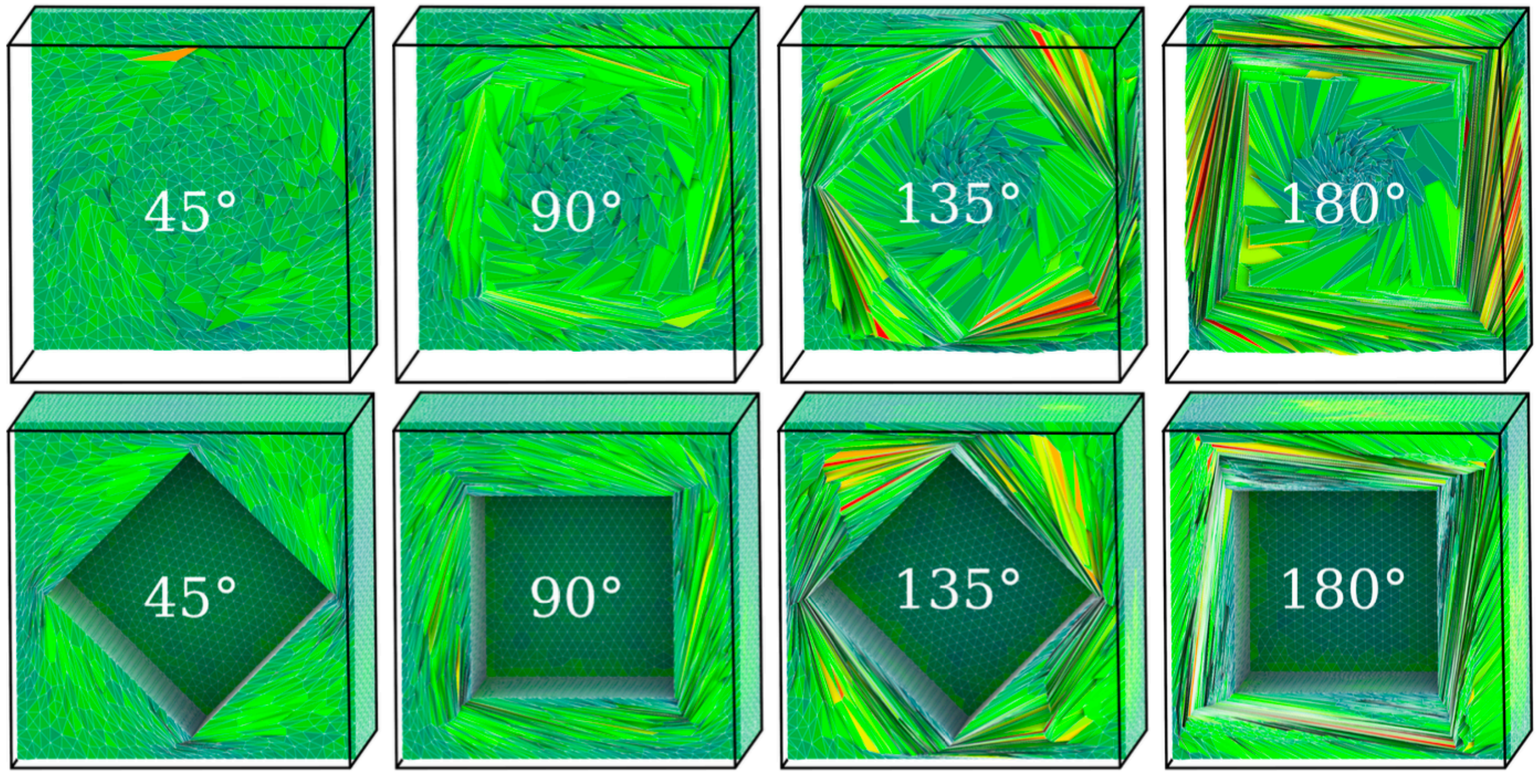
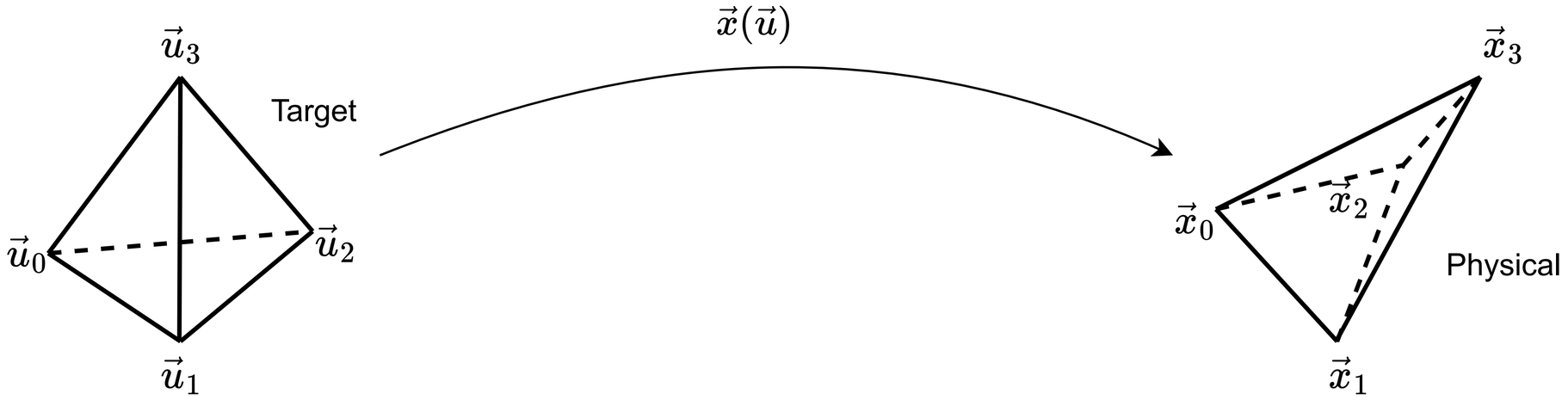
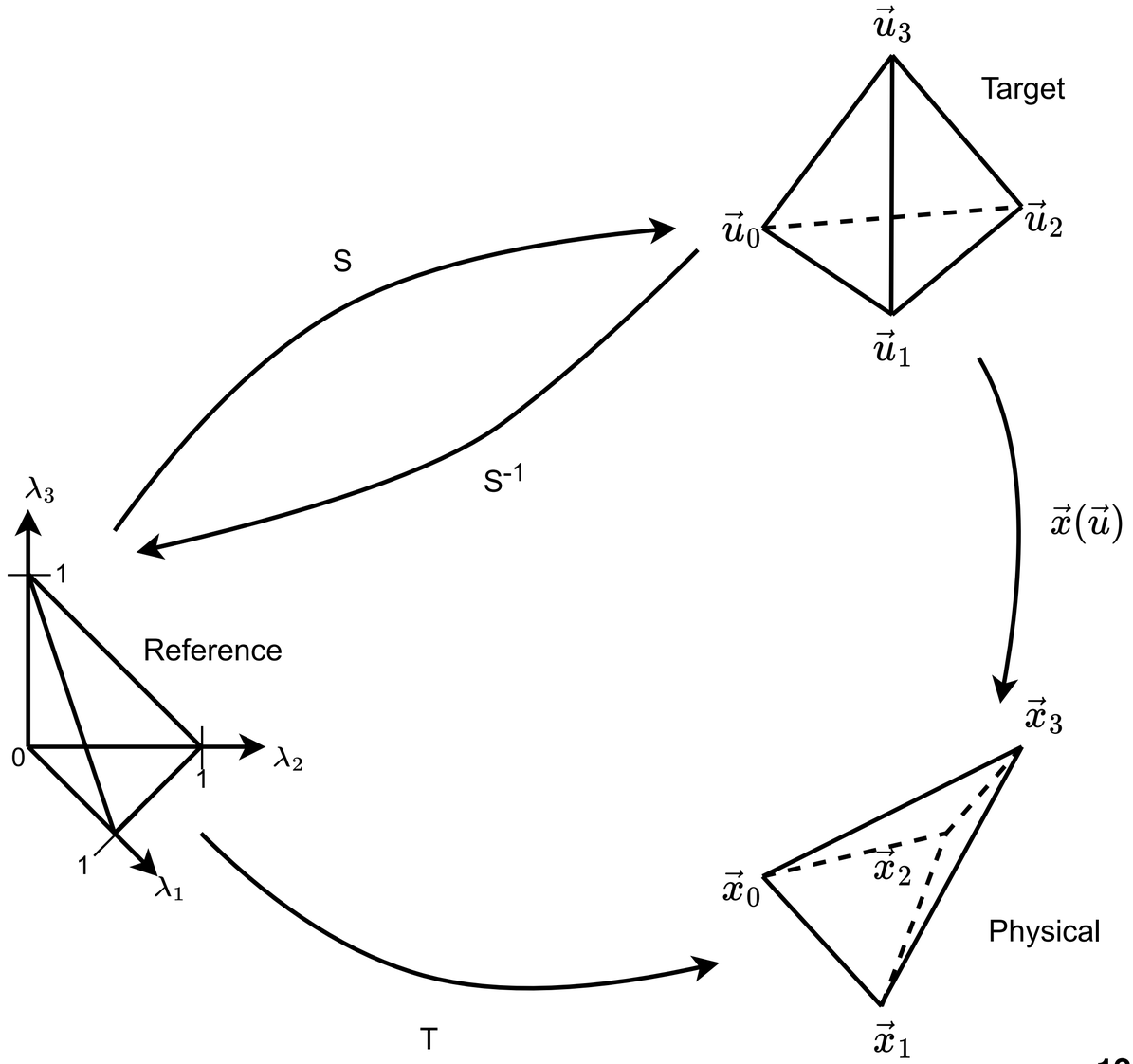


Fig. 6 in Foldover Free Maps

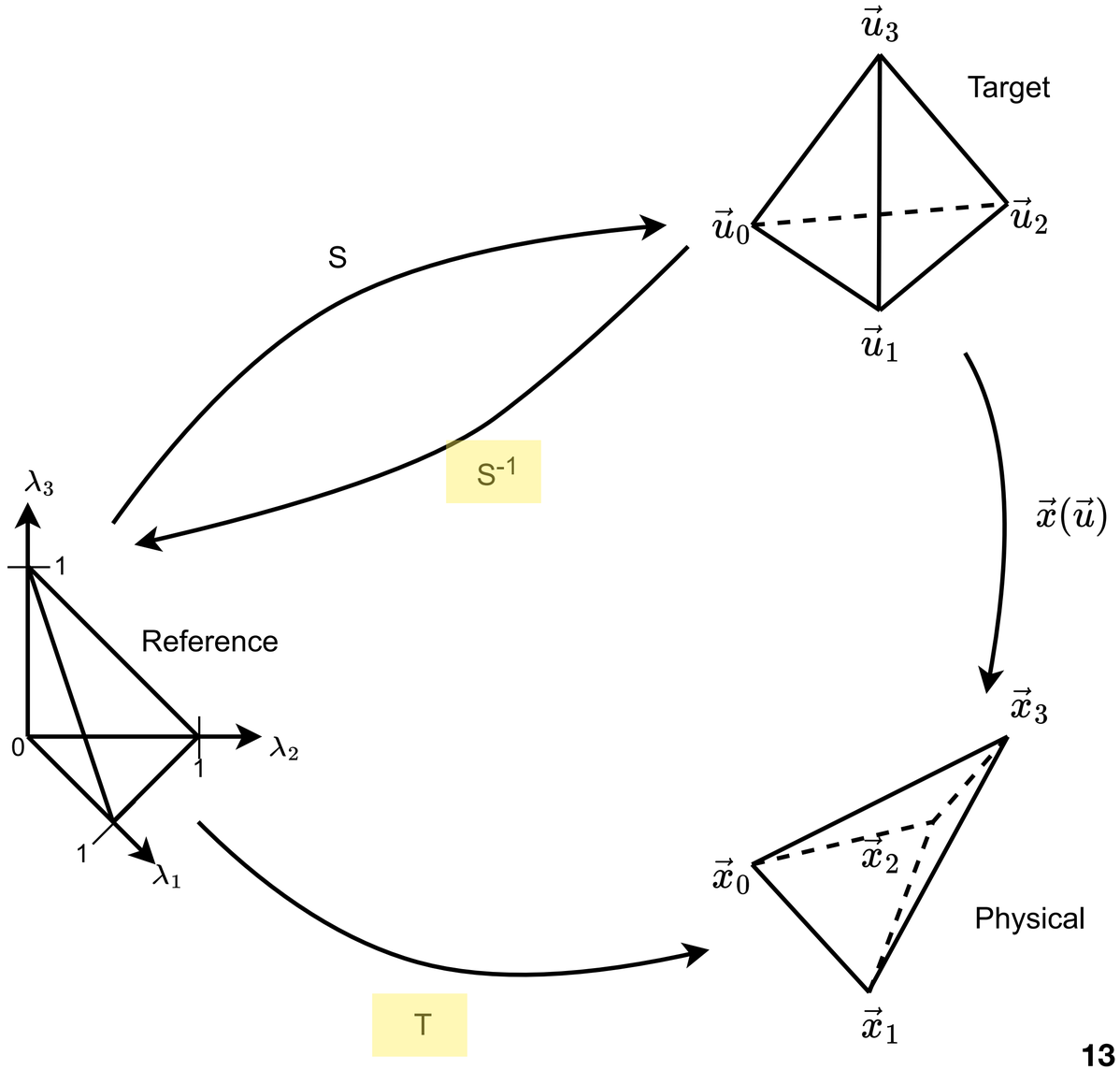
New Situation



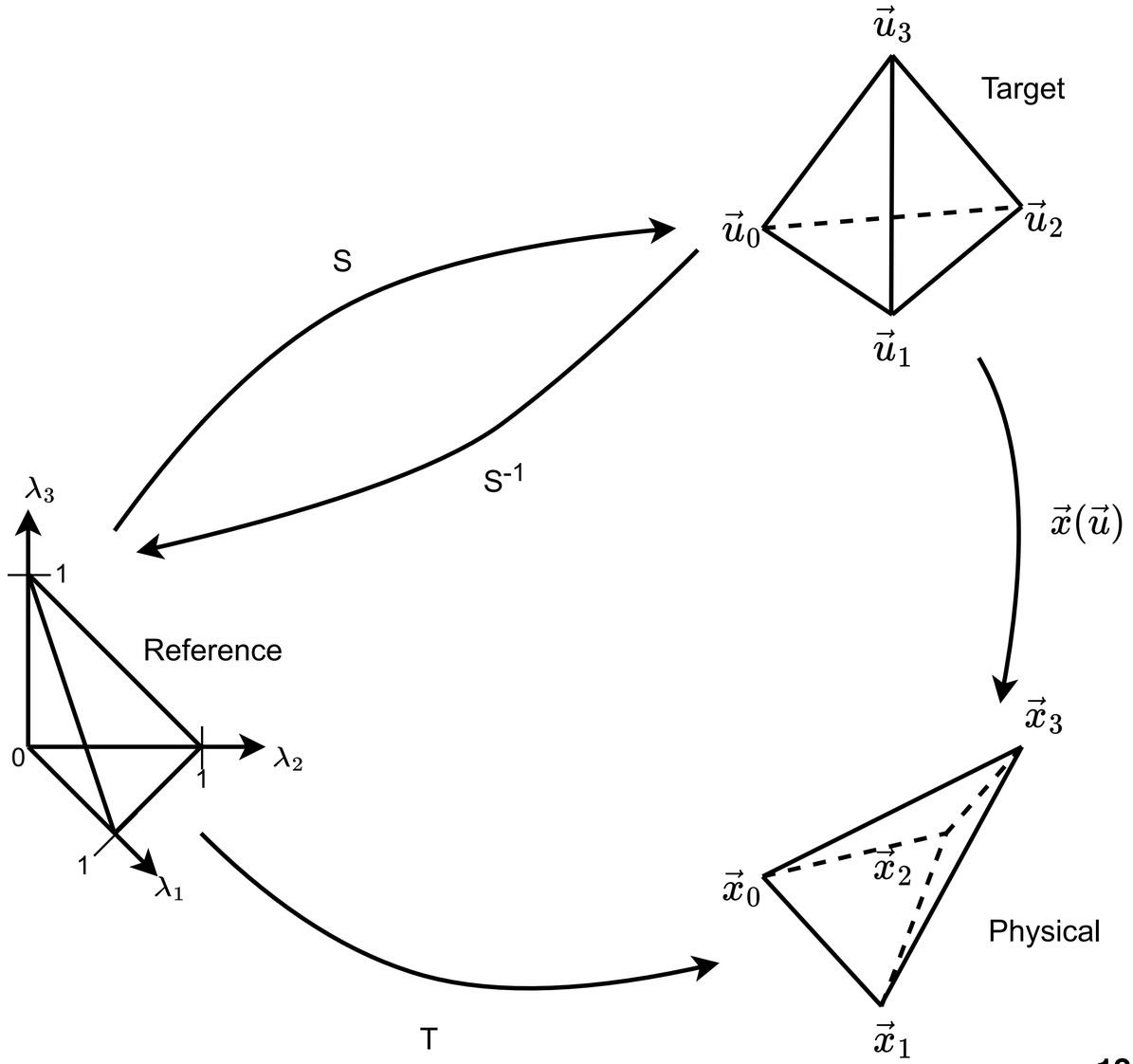
Reference Space and Formulas



Reference Space and Formulas

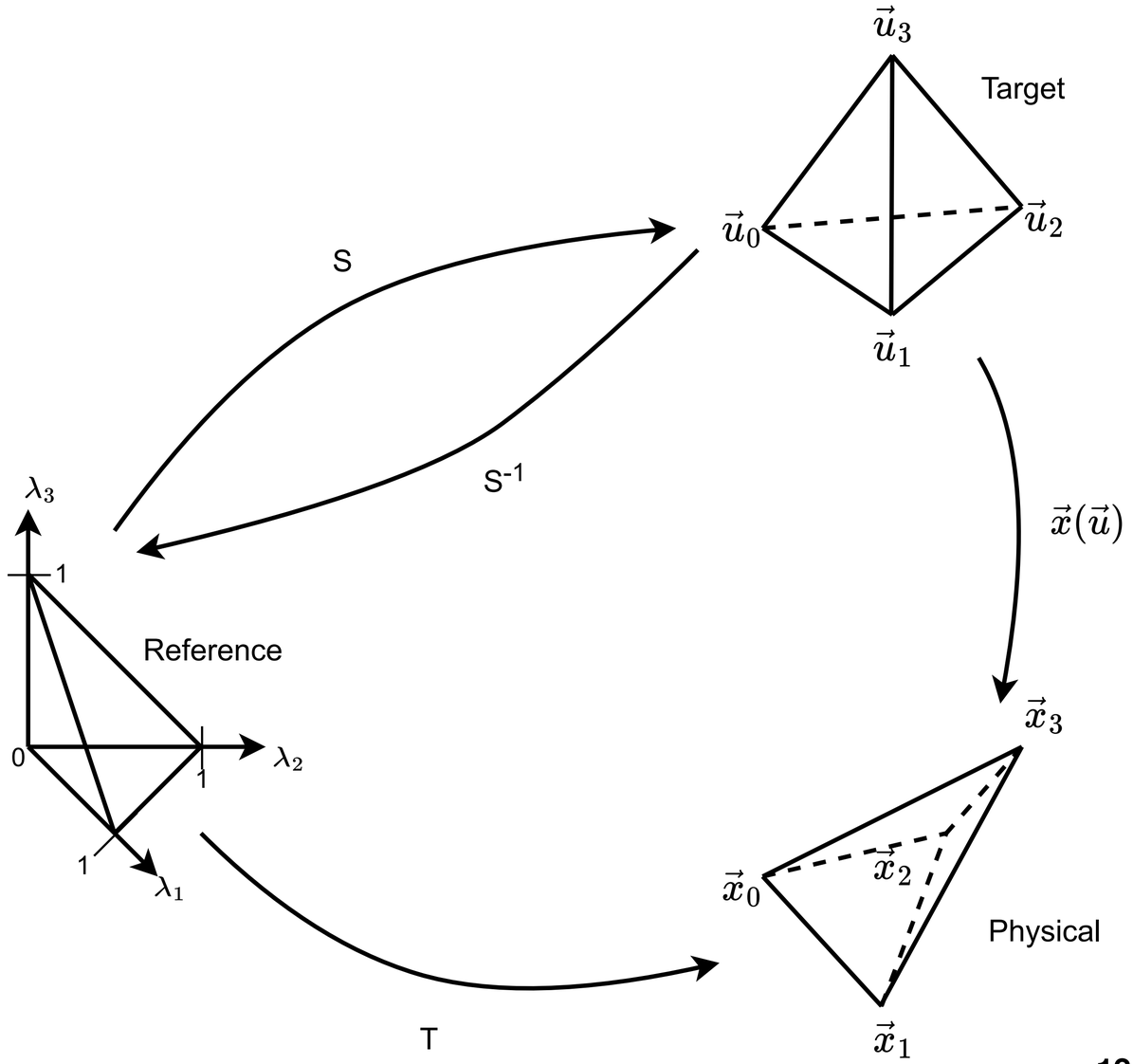


Reference Space and Formulas



$$J(T) = \begin{pmatrix} x_1 - x_0 & x_2 - x_0 & x_3 - x_0 \\ y_1 - y_0 & y_2 - y_0 & y_3 - y_0 \\ z_1 - z_0 & z_2 - z_0 & z_3 - z_0 \end{pmatrix}$$

Reference Space and Formulas



$$J(T) = \begin{pmatrix} x_1 - x_0 & x_2 - x_0 & x_3 - x_0 \\ y_1 - y_0 & y_2 - y_0 & y_3 - y_0 \\ z_1 - z_0 & z_2 - z_0 & z_3 - z_0 \end{pmatrix}$$

$$f_\epsilon(J) := \frac{trJ^T J}{(\chi(\det J, \epsilon))^{\frac{2}{3}}}$$

Epsilon

$$\chi(\det J, \epsilon) := \frac{\det J + \sqrt{\epsilon^2 + \det^2 J}}{2}, \epsilon \geq 0$$

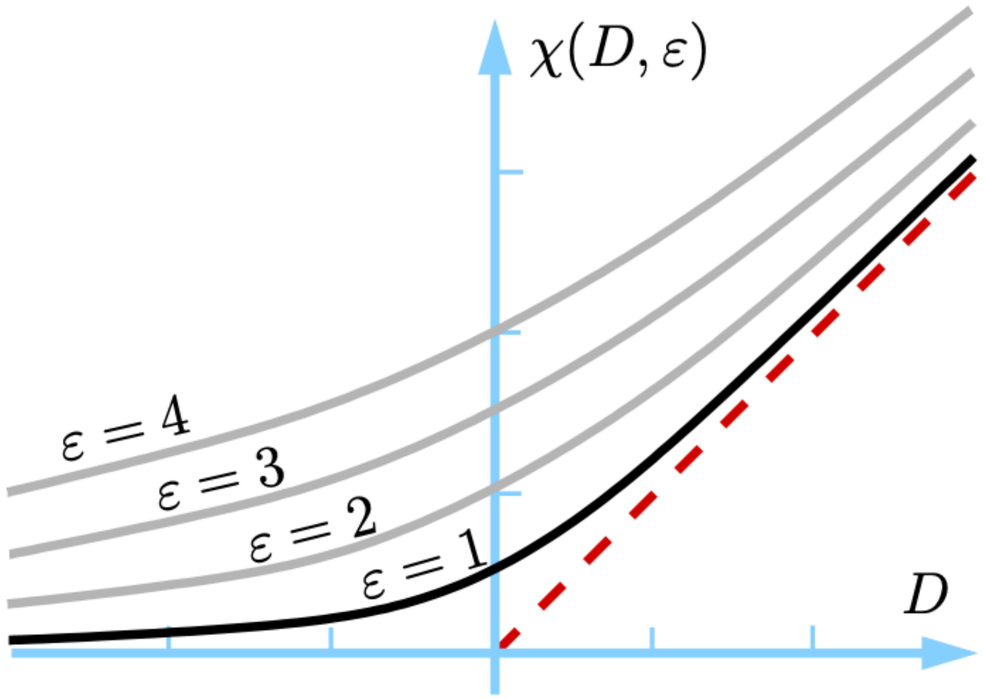


Fig. 2 in Foldover Free Maps

Basic Algorithm

Algorithm 1 58 Tetrahedra

Input: X, List of Unit Tetrahedra
Output: X

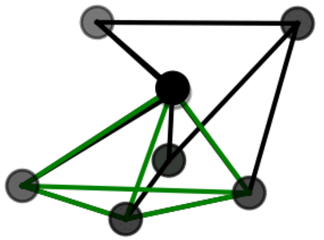
validity $\leftarrow$ check mesh validity$(X, Tetrahedra)$
if not validity **then**
  $\epsilon^0 \leftarrow 1$
  while $\det_{\min} \leq 0$ **do**
    $F_{prev} \leftarrow \text{energy}(X^k, \epsilon^k, Tetrahedra)$
    $X^{k+1} \leftarrow X^k + \Delta X$
    $F \leftarrow \text{energy}(X^{k+1}, \epsilon^k, Tetrahedra)$
    $\epsilon^{k+1} \leftarrow \text{update epsilon}(F_{prev}, F, \epsilon^k)$
  end while
end if

Algorithm 1 58 Tetrahedra

Input: X, List of Unit Tetrahedra
Output: X

validity $\leftarrow$ check mesh validity$(X, Tetrahedra)$
if not validity **then**
  $\epsilon^0 \leftarrow 1$
  while $\det_{\min} \leq 0$ **do**
    $F_{prev} \leftarrow \text{energy}(X^k, \epsilon^k, Tetrahedra)$
    $X^{k+1} \leftarrow X^k + \Delta X$
    $F \leftarrow \text{energy}(X^{k+1}, \epsilon^k, Tetrahedra)$
    $\epsilon^{k+1} \leftarrow \text{update epsilon}(F_{prev}, F, \epsilon^k)$
  end while
end if

Physical Hexahedron



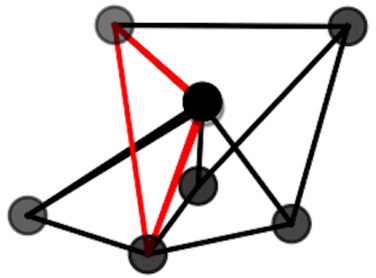
Physical Tetrahedron

Algorithm 1 58 Tetrahedra

Input: X, List of Unit Tetrahedra
Output: X

validity $\leftarrow$ check mesh validity$(X, Tetrahedra)$
if not validity **then**
  $\epsilon^0 \leftarrow 1$
  while $\det_{\min} \leq 0$ **do**
    $F_{prev} \leftarrow \text{energy}(X^k, \epsilon^k, Tetrahedra)$
    $X^{k+1} \leftarrow X^k + \Delta X$
    $F \leftarrow \text{energy}(X^{k+1}, \epsilon^k, Tetrahedra)$
    $\epsilon^{k+1} \leftarrow \text{update epsilon}(F_{prev}, F, \epsilon^k)$
  end while
end if

Physical Hexahedron

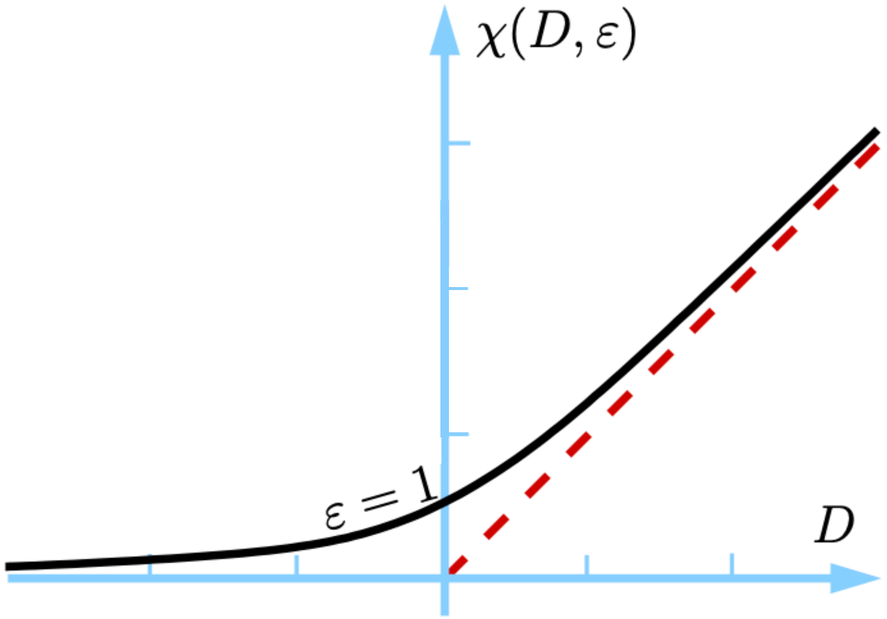


Physical Tetrahedron

Algorithm 1 58 Tetrahedra

Input: X, List of Unit Tetrahedra
Output: X

validity $\leftarrow$ check mesh validity$(X, Tetrahedra)$
if not validity **then**
  $\epsilon^0 \leftarrow 1$
  while $\det_{\min} \leq 0$ **do**
    $F_{prev} \leftarrow$ energy$(X^k, \epsilon^k, Tetrahedra)$
    $X^{k+1} \leftarrow X^k + \Delta X$
    $F \leftarrow$ energy$(X^{k+1}, \epsilon^k, Tetrahedra)$
    $\epsilon^{k+1} \leftarrow$ update epsilon$(F_{prev}, F, \epsilon^k)$
  end while
end if



Algorithm 1 58 Tetrahedra

Input: X, List of Unit Tetrahedra
Output: X

validity $\leftarrow$ check mesh validity$(X, Tetrahedra)$
if not validity **then**
  $\epsilon^0 \leftarrow 1$
  while $\det_{\min} \leq 0$ **do**
    $F_{prev} \leftarrow \text{energy}(X^k, \epsilon^k, Tetrahedra)$
    $X^{k+1} \leftarrow X^k + \Delta X$
    $F \leftarrow \text{energy}(X^{k+1}, \epsilon^k, Tetrahedra)$
    $\epsilon^{k+1} \leftarrow$ update epsilon$(F_{prev}, F, \epsilon^k)$
  end while
end if

Algorithm 1 58 Tetrahedra

Input: X, List of Unit Tetrahedra
Output: X

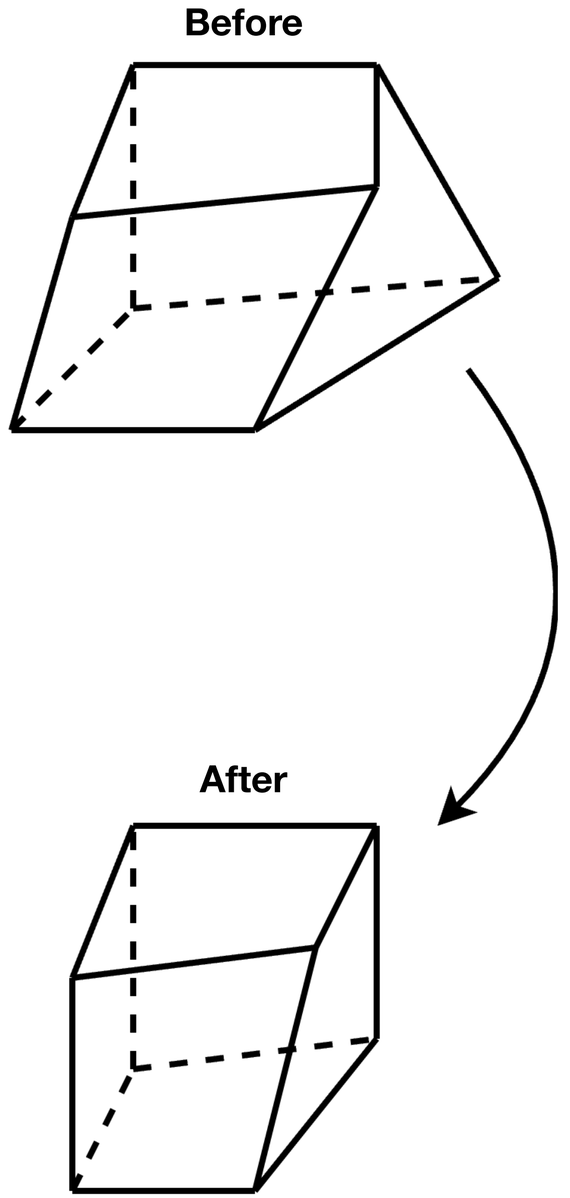
validity $\leftarrow$ check mesh validity$(X, Tetrahedra)$
if not validity **then**
  $\epsilon^0 \leftarrow 1$
  while $\det_{\min} \leq 0$ **do**
    $F_{prev} \leftarrow \text{energy}(X^k, \epsilon^k, Tetrahedra)$
    $X^{k+1} \leftarrow X^k + \Delta X$
    $F \leftarrow \text{energy}(X^{k+1}, \epsilon^k, Tetrahedra)$
    $\epsilon^{k+1} \leftarrow$ update epsilon$(F_{prev}, F, \epsilon^k)$
  end while
end if

$$f_\epsilon(J) := \frac{trJ^TJ}{(\chi(\det J, \epsilon))^{\frac{2}{3}}}$$

Algorithm 1 58 Tetrahedra

Input: X, List of Unit Tetrahedra
Output: X

validity $\leftarrow$ check mesh validity$(X, Tetrahedra)$
if not validity **then**
  $\epsilon^0 \leftarrow 1$
  while $\det_{\min} \leq 0$ **do**
    $F_{prev} \leftarrow \text{energy}(X^k, \epsilon^k, Tetrahedra)$
    $X^{k+1} \leftarrow X^k + \Delta X$
    $F \leftarrow \text{energy}(X^{k+1}, \epsilon^k, Tetrahedra)$
    $\epsilon^{k+1} \leftarrow \text{update epsilon}(F_{prev}, F, \epsilon^k)$
  end while
end if



Algorithm 1 58 Tetrahedra

Input: X, List of Unit Tetrahedra
Output: X

validity $\leftarrow$ check mesh validity$(X, Tetrahedra)$
if not validity **then**
  $\epsilon^0 \leftarrow 1$
  while $\det_{\min} \leq 0$ **do**
    $F_{prev} \leftarrow \text{energy}(X^k, \epsilon^k, Tetrahedra)$
    $X^{k+1} \leftarrow X^k + \Delta X$
    $F \leftarrow \text{energy}(X^{k+1}, \epsilon^k, Tetrahedra)$
    $\epsilon^{k+1} \leftarrow$ update epsilon$(F_{prev}, F, \epsilon^k)$
  end while
end if

$$f_\epsilon(J) := \frac{trJ^TJ}{(\chi(\det J, \epsilon))^{\frac{2}{3}}}$$

Algorithm 1 58 Tetrahedra

Input: X, List of Unit Tetrahedra
Output: X

validity $\leftarrow$ check mesh validity$(X, Tetrahedra)$
if not validity **then**
  $\epsilon^0 \leftarrow 1$
  while $\det_{\min} \leq 0$ **do**
    $F_{prev} \leftarrow \text{energy}(X^k, \epsilon^k, Tetrahedra)$
    $X^{k+1} \leftarrow X^k + \Delta X$
    $F \leftarrow \text{energy}(X^{k+1}, \epsilon^k, Tetrahedra)$
    $\epsilon^{k+1} \leftarrow \text{update epsilon}(F_{prev}, F, \epsilon^k)$
  end while
end if

$$\epsilon^k = \begin{cases} 2\sqrt{\mu^k(\mu^k - D_{min}^{k+1})}, & D_{min}^{k+1} < \mu^k \\ 0, & D_{min}^{k+1} \geq \mu^k \end{cases}$$

$$\mu^k := (1 - \sigma^k)\chi(D_{min}^{k+1}, \epsilon^k)$$

$$\sigma^k = \max\left(\frac{1}{10}, 1 - \frac{F(X^{k+1}, \epsilon^k)}{F(X^k, \epsilon^k)}\right)$$

There is room for improvement!

Amount of Tetrahedra

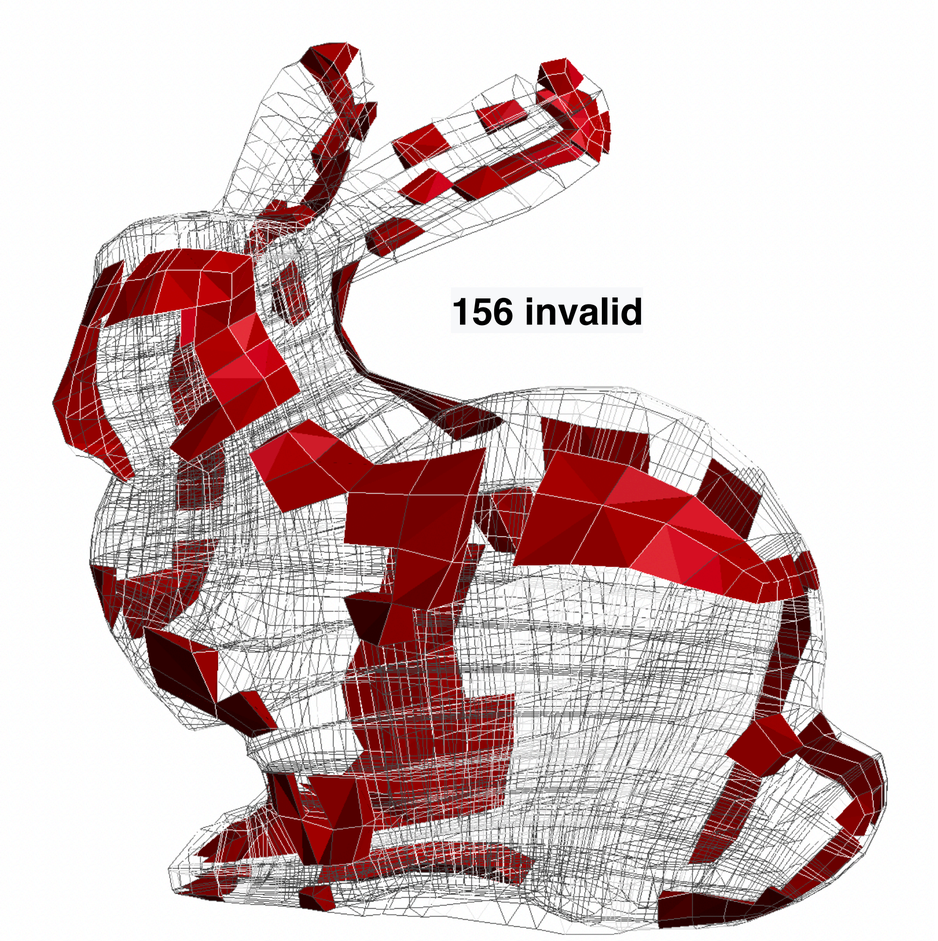
Reduction by Factor of
58 / 8 = 7.25

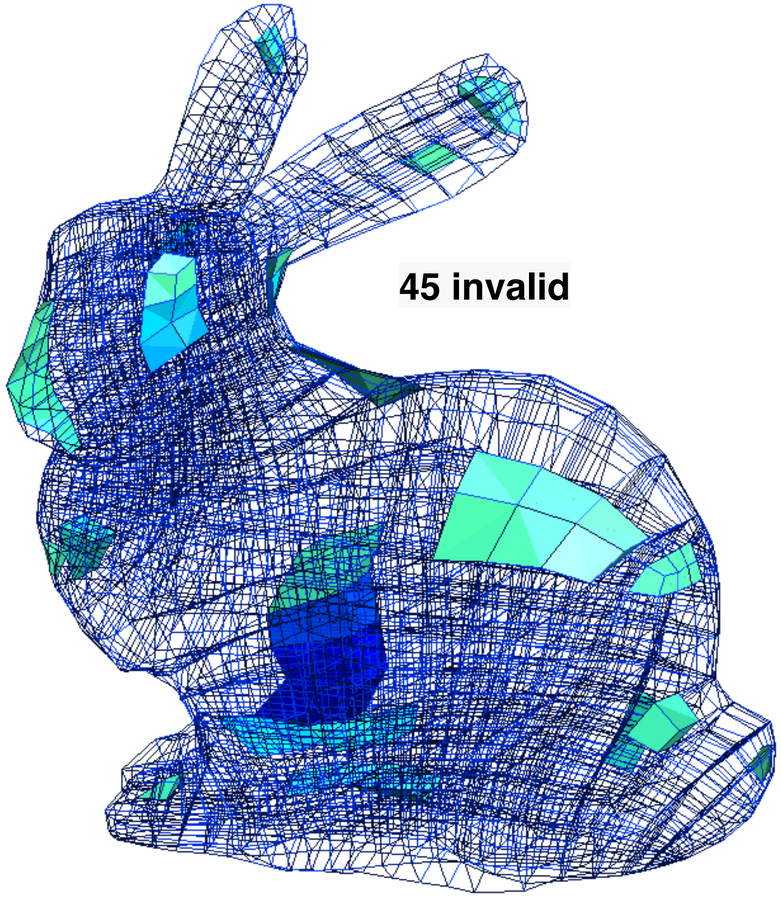
# Hexahedra	# Tetrahedra
2832	2832 * 58 = 164’256 2832 * 8 = 22’656

Blobs

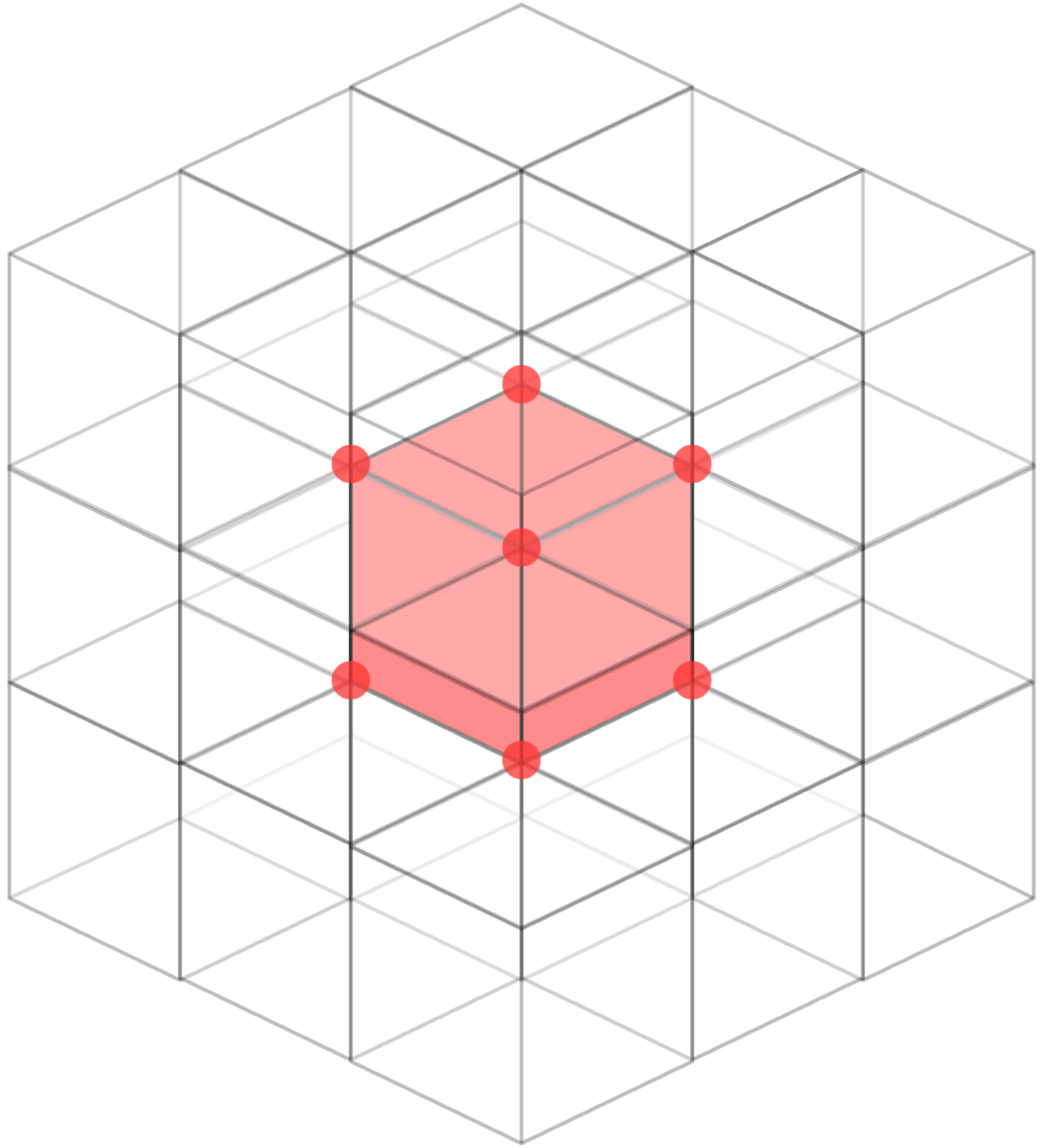
Use of exact Validity Condition

Johnen & al 2017



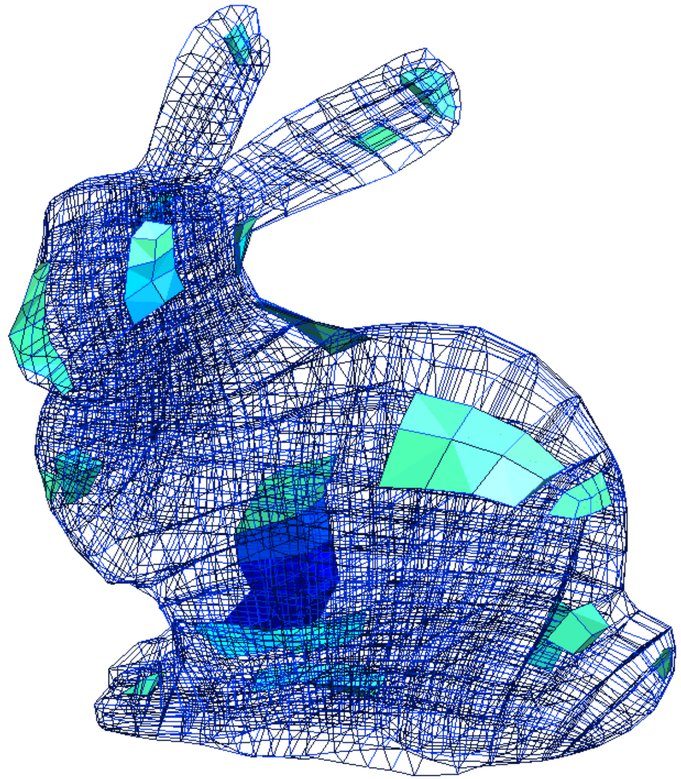


Blob Construction



Blobs

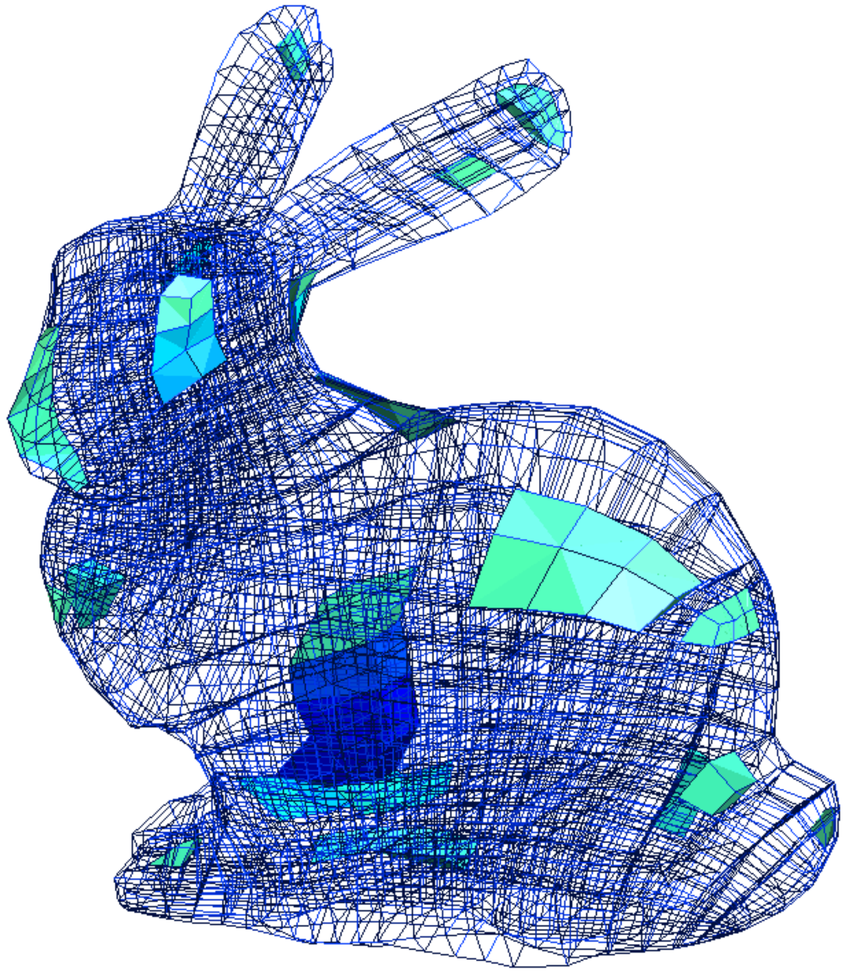
Reduction by Factor of 22’656 / 2’536 = 8.93



# Hexahedra	# Invalid Hexahedra	# Blob Hexahedra	# Tetrahedra
2832	45	317	2832 * 8 = 22’656 317 * 8 = 2’536

Boundary Relaxation

45 invalid Boundary Hexahedra



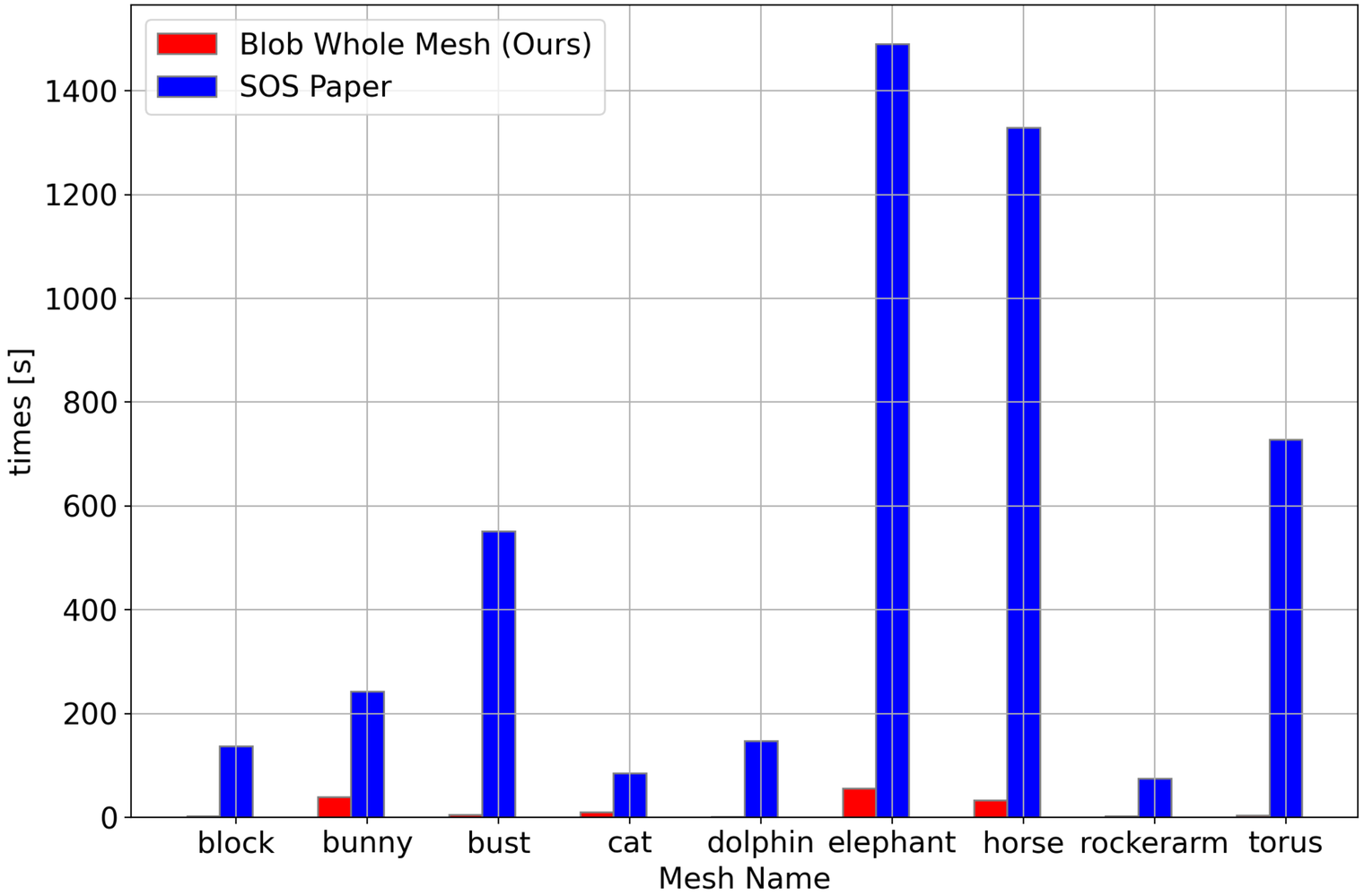
Boundary Penalty

$$E = F + F_{penalty}$$

$$F_{penalty} = (pos_{current} - pos_{start})^2 \cdot factor$$

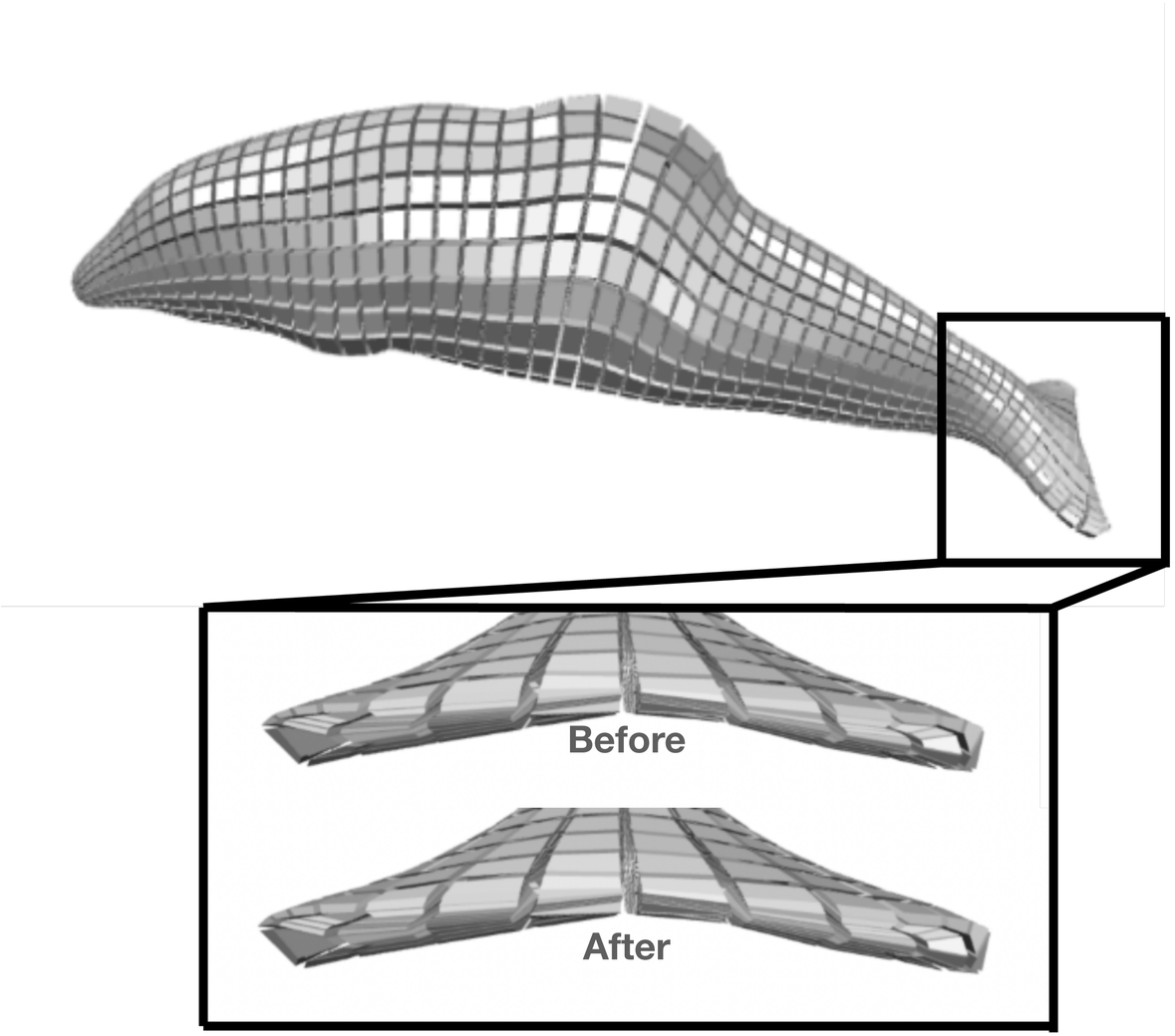
Results

Marschner & al 2020



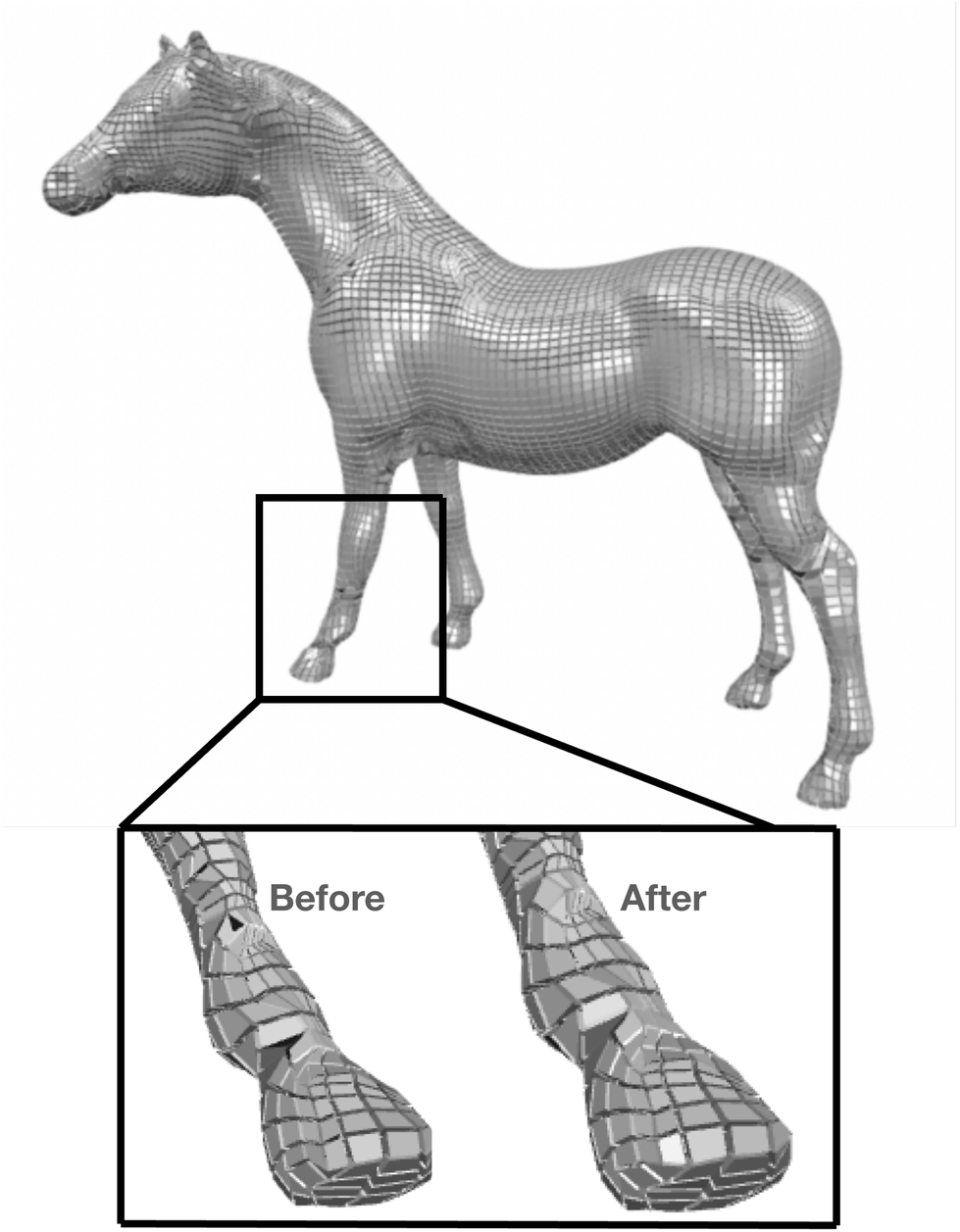
Dolphin

- Times:
  - Ours: 1.24474 **s**
  - SOS: 2.45 **min**
- Boundary Displacement (Scaled)
  - avg: 1.171649E-05
  - max: 4.158615E-03



Horse

- Times:
  - Ours: 29.7245 **s**
  - SOS: 22.15 **min**
- Boundary Displacement (Scaled)
  - avg: 8.840065E-03
  - max: 7.445004E-01



Conclusion

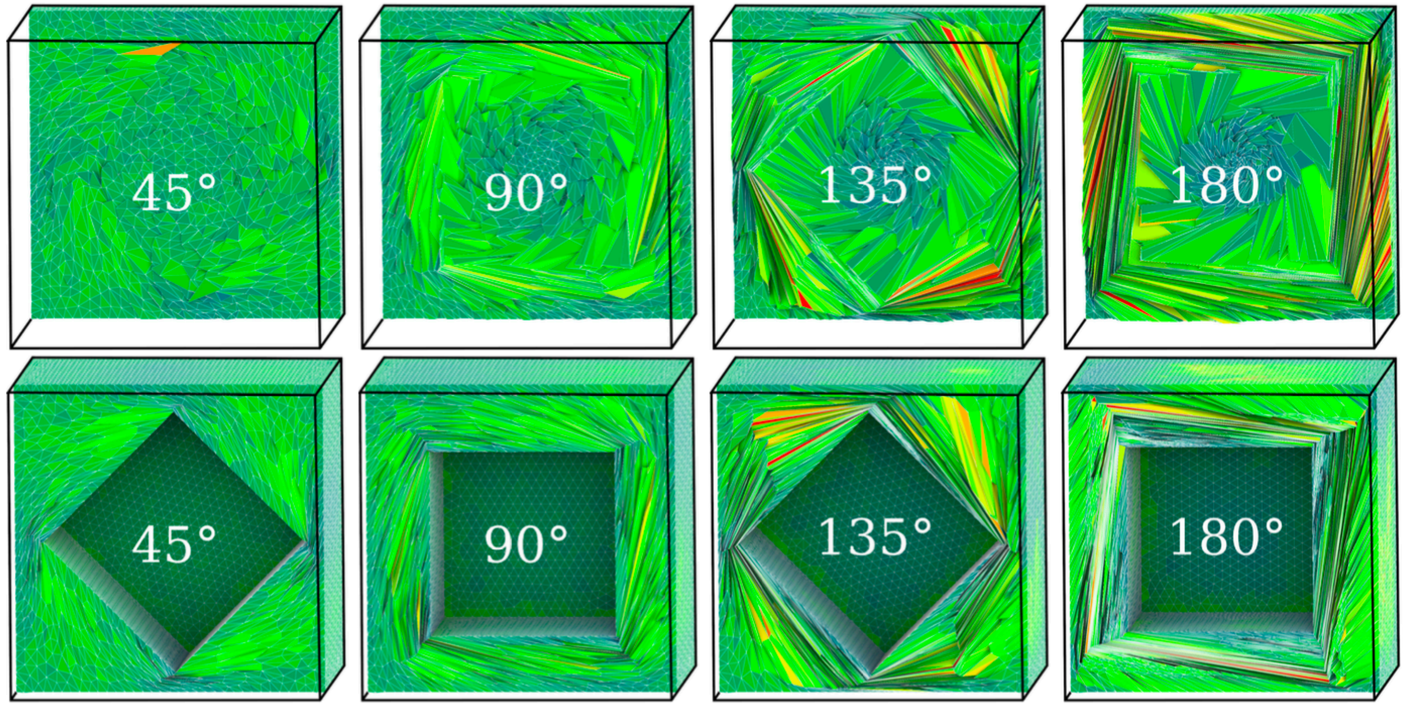
- 58 Tets / Hex

Unit Cube

Tetrahedra

Conclusion

- 58 Tets / Hex
- Foldover Free Maps



Conclusion

- 58 Tets / Hex
- Foldover Free Maps
- Basic Algorithm

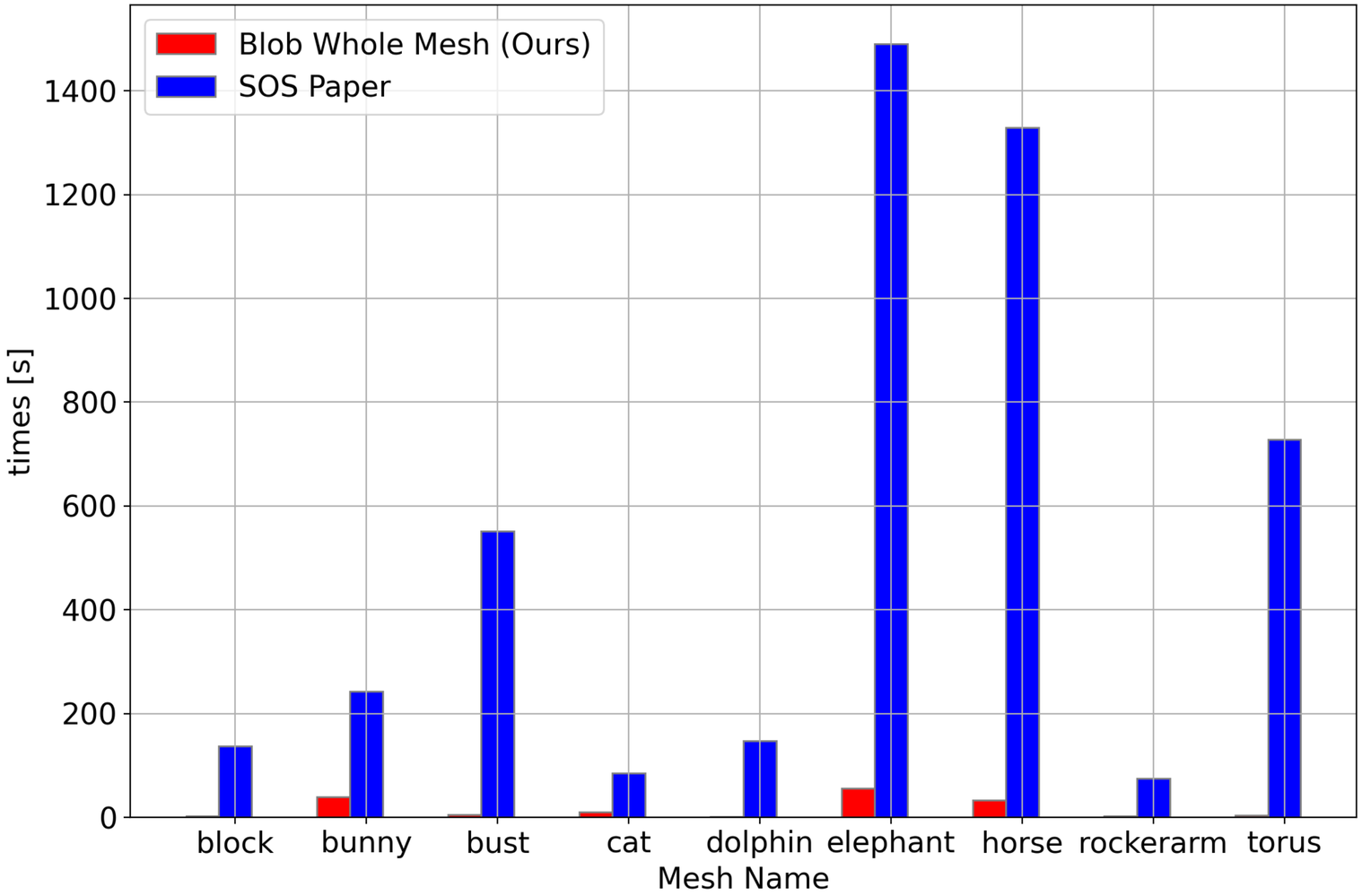
Algorithm 1 58 Tetrahedra

Input: X, List of Unit Tetrahedra
Output: X

validity $\leftarrow$ check mesh validity$(X, Tetrahedra)$
if not validity **then**
  $\epsilon^0 \leftarrow 1$
  while $\det_{\min} \leq 0$ **do**
    $F_{prev} \leftarrow$ energy$(X^k, \epsilon^k, Tetrahedra)$
    $X^{k+1} \leftarrow X^k + \Delta X$
    $F \leftarrow$ energy$(X^{k+1}, \epsilon^k, Tetrahedra)$
    $\epsilon^{k+1} \leftarrow$ update epsilon$(F_{prev}, F, \epsilon^k)$
  end while
end if

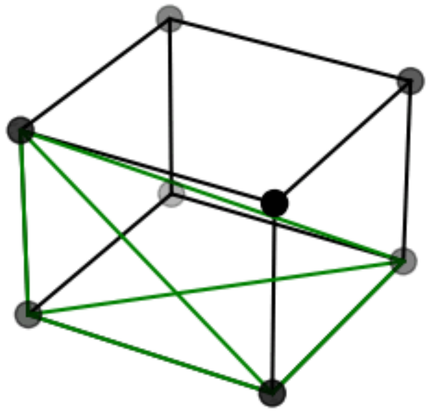
Conclusion

- 58 Tets / Hex
- Foldover Free Maps
- Basic Algorithm
- Improvements

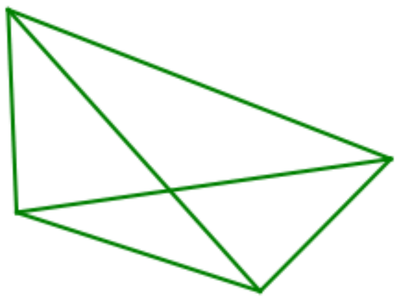


Thank you for your attention!
Any Questions?

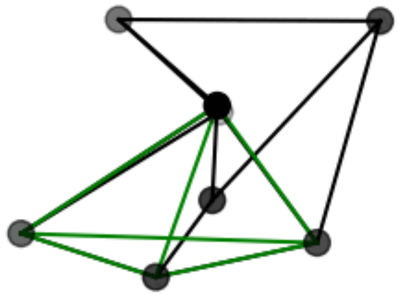
Target Hexahedron



Target Tetrahedron



Physical Hexahedron



Physical Tetrahedron

References

- Recoil Engineering: https://www.recoilengineering.com/fea
- Dimitris Varziotis et al: https://www.sciencedirect.com/science/article/abs/pii/S0168874X12002077
- Olga V.Ushakova: Nondegeneracy tests for hexahedral cells. Comput. Methods Appl. Mech. Engrg. 200: 1649–1658 (2011)
- Amaury Johnen, Jean-Christophe Weill, Jean-François Remacle: Robust and efficient validation of the linear hexahedral element. CoRR abs/1706.01613 (2017)
- Vladimir A. Garanzha, Igor E. Kaporin, Liudmila N. Kudryavtseva, François Protais, Nicolas Ray, Dmitry Sokolov: Foldover-free maps in 50 lines of code. ACM Trans. Graph. 40(4): 102:1-102:16 (2021)
- Zoë Marschner, David R. Palmer, Paul Zhang, Justin Solomon: Hexahedral Mesh Repair via Sum-of-Squares Relaxation. Comput. Graph. Forum 39(5): 133-147 (2020)