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

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
Situation

Target → Physical

F
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

\[ \vec{u}_0 \xrightarrow{S} \vec{u}_1 \xrightarrow{S^{-1}} \vec{u}_2 \xrightarrow{T} \vec{u}_3 \]

\[ \vec{x}(\vec{u}) \]

Reference

Target

Physical
Reference Space and Formulas

\[ S \]

\[ S^{-1} \]

\[ \vec{x}(\vec{u}) \]

\[ \vec{x}_0 \]

\[ \vec{x}_1 \]

\[ \vec{x}_2 \]

\[ \vec{x}_3 \]

\[ \vec{u}_0 \]

\[ \vec{u}_1 \]

\[ \vec{u}_2 \]

\[ \vec{u}_3 \]

Reference

Physical
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

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

\[ f_\epsilon(J) := \frac{tr J^T J}{(\chi(\det J, \epsilon))^\frac{2}{3}} \]
Epsilon

\[ \chi(\det J, \epsilon) := \frac{\det J + \sqrt{\epsilon^2 + \det^2 J}}{2}, \quad \epsilon \geq 0 \]
Algorithm 1 58 Tetrahedra

Input: X, List of Unit Tetrahedra

Output: X

validity ← check mesh validity(X, Tetrahedra)
if not validity then
    \( \varepsilon^0 \leftarrow 1 \)
while \( \det_{\text{min}} \leq 0 \) do
    \( F_{\text{prev}} \leftarrow \text{energy}(X^k, \varepsilon^k, \text{Tetrahedra}) \)
    \( X^{k+1} \leftarrow X^k + \Delta X \)
    \( F \leftarrow \text{energy}(X^{k+1}, \varepsilon^k, \text{Tetrahedra}) \)
    \( \varepsilon^{k+1} \leftarrow \text{update epsilon}(F_{\text{prev}}, F, \varepsilon^k) \)
end while
end if
Algorithm 1 58 Tetrahedra

Input: X, List of Unit Tetrahedra

Output: X

validity ← check mesh validity($X, Tetrahedra$)
if not validity then
    $e^0 ← 1$
    while $\text{det}_{\text{min}} \leq 0$ do
        $F_{\text{prev}} ← \text{energy($X^k, e^k, Tetrahedra$)}$
        $X^{k+1} ← X^k + \Delta X$
        $F ← \text{energy($X^{k+1}, e^k, Tetrahedra$)}$
        $e^{k+1} ← \text{update epsilon($F_{\text{prev}}, F, e^k$)}$
    end while
end if
Algorithm 1 58 Tetrahedra

Input: X, List of Unit Tetrahedra
Output: X

validity ← check mesh validity(X, Tetrahedra)
if not validity then
    \( e^0 \leftarrow 1 \)
    while \( \text{det}_{\text{min}} \leq 0 \) do
        \( F_{\text{prev}} \leftarrow \text{energy}(X^k, e^k, \text{Tetrahedra}) \)
        \( X^{k+1} \leftarrow X^k + \Delta X \)
        \( F \leftarrow \text{energy}(X^{k+1}, e^k, \text{Tetrahedra}) \)
        \( e^{k+1} \leftarrow \text{update epsilon}(F_{\text{prev}}, F, e^k) \)
    end while
end if
Algorithm 1: Tetrahedra

Input: X, List of Unit Tetrahedra
Output: X

validity ← check mesh validity($X, Tetrahedra$)
if not validity then
    $\epsilon_0 \leftarrow 1$
    while $\text{det}_{\min} \leq 0$ do
        $F_{\text{prev}} \leftarrow \text{energy}(X^k, \epsilon^k, Tetrahedra)$
        $X^{k+1} \leftarrow X^k + \Delta X$
        $F \leftarrow \text{energy}(X^{k+1}, \epsilon^k, Tetrahedra)$
        $\epsilon^{k+1} \leftarrow \text{update epsilon}(F_{\text{prev}}, F, \epsilon^k)$
    end while
end if
Algorithm 1 58 Tetrahedra

Input: $X$, List of Unit Tetrahedra

Output: $X$

validity ← check mesh validity($X$, Tetrahedra)
if not validity then
    $e^0 ← 1$
    while $det_{min} ≤ 0$ do
        $F_{prev} ← energy(X^k, e^k, Tetrahedra)$
        $X^{k+1} ← X^k + \Delta X$
        $F ← energy(X^{k+1}, e^k, Tetrahedra)$
        $e^{k+1} ← update \text{ epsilon}(F_{prev}, F, e^k)$
    end while
end if
Algorithm 1 58 Tetrahedra

**Input:** X, List of Unit Tetrahedra

**Output:** X

validity ← 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

$$f_\epsilon(J) := \frac{tr J^T J}{(\chi(\det J, \epsilon))^\frac{2}{3}}$$
**Algorithm 1** 58 Tetrahedra

**Input:** X, List of Unit Tetrahedra

**Output:** X

validity ← check mesh validity($X$, Tetrahedra)

if not validity then

$e^0 ← 1$

while $\det_{\text{min}} \leq 0$ do

$F_{\text{prev}} ← \text{energy}(X^k, e^k, Tetrahedra)$

$X^{k+1} ← X^k + \Delta X$

$F ← \text{energy}(X^{k+1}, e^k, Tetrahedra)$

$e^{k+1} ← \text{update epsilon}(F_{\text{prev}}, F, e^k)$

end while

end if
Algorithm 1 58 Tetrahedra

Input: X, List of Unit Tetrahedra

Output: X

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

$$f_\epsilon(J) := \frac{\text{tr} J^T J}{(\chi(\det J, \epsilon))^{\frac{2}{3}}}$$
\[ \epsilon^k = \begin{cases} 2 \sqrt{\mu^k (\mu^k - D_{\text{min}}^{k+1})}, & D_{\text{min}}^{k+1} < \mu^k \\ 0, & D_{\text{min}}^{k+1} \geq \mu^k \end{cases} \]

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

\[ \sigma^k = \max \left( \frac{1}{10}, 1 - \frac{F(X^{k+1}, \epsilon^k)}{F(X^k, \epsilon^k)} \right) \]
There is room for improvement!
Amount of Tetrahedra

Reduction by Factor of $58 / 8 = 7.25$

| # Hexahedra | # Tetrahedra |
|-------------|--------------|
| 2832        | $2832 \times 58 = 164'256$  
               | $2832 \times 8 = 22'656$ |
Blobs

Use of exact Validity Condition

Johnen & al 2017

156 invalid

45 invalid
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

\[ E = F + F_{\text{penalty}} \]

\[ F_{\text{penalty}} = (pos_{\text{current}} - pos_{\text{start}})^2 \cdot \text{factor} \]
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
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 ← check mesh validity(X, Tetrahedra)
if not validity then
    \( \varepsilon^0 \leftarrow 1 \)
    while \( \det_{\min} \leq 0 \) do
        \( F_{\text{prev}} \leftarrow \text{energy}(X^k, \varepsilon^k, \text{Tetrahedra}) \)
        \( X^{k+1} \leftarrow X^k + \Delta X \)
        \( F \leftarrow \text{energy}(X^{k+1}, \varepsilon^k, \text{Tetrahedra}) \)
        \( \varepsilon^{k+1} \leftarrow \text{update epsilon}(F_{\text{prev}}, F, \varepsilon^k) \)
    end while
end if
Conclusion

• 58 Tets / Hex

• Foldover Free Maps

• Basic Algorithm

• Improvements
Thank you for your attention!

Any Questions?
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)