

Dual Sheet Meshing: An Interactive Approach to Robust Hexahedralization

Kenshi Takayama

takayama@nii.ac.jp

National Institute of Informatics

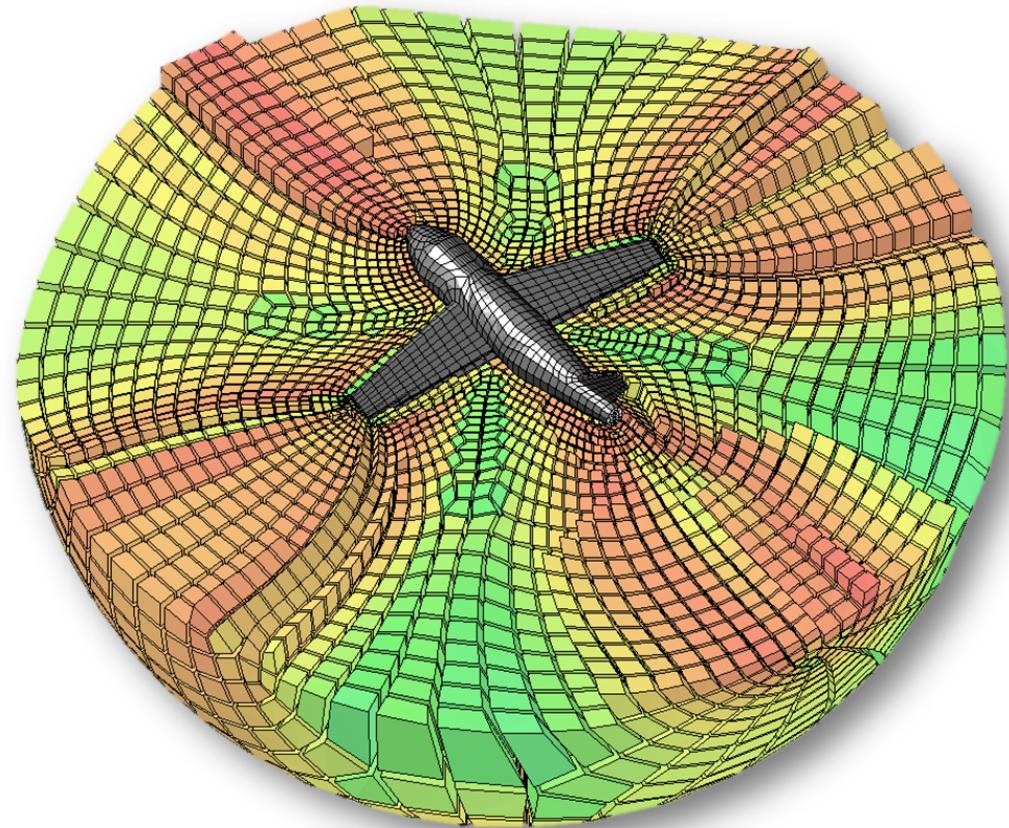
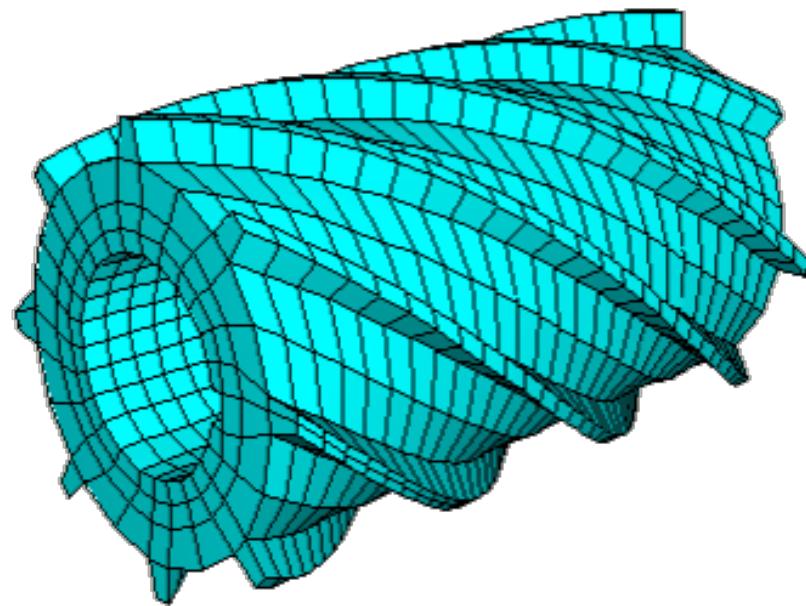
EG2019

The 40° Annual Conference of the European Association for Computer Graphics



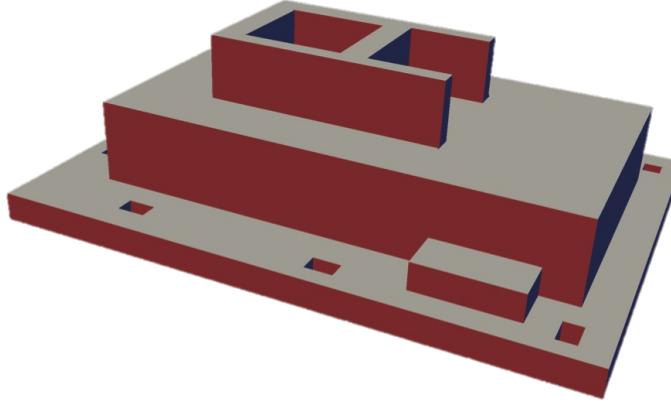
Hexahedral meshes

- Needed for running numerical simulations on 3D shapes



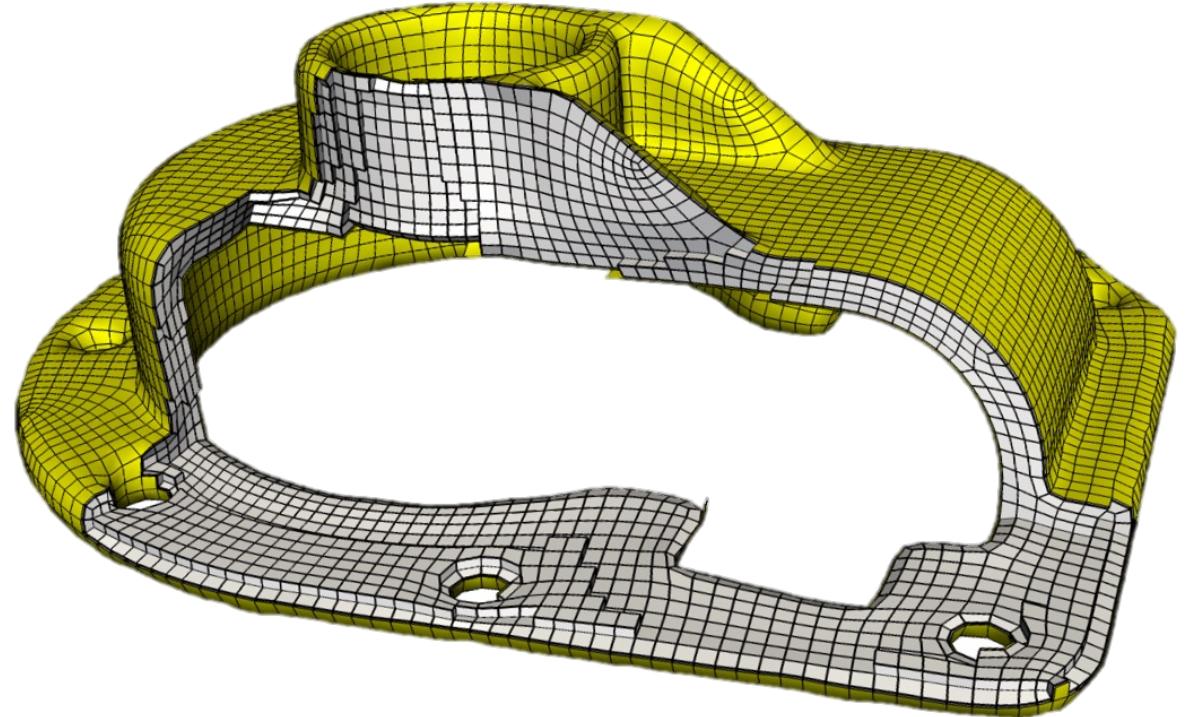
State-of-the-art (1/2): PolyCube mapping

[Gregson et al. 11, Huang et al. 14, Fang et al. 16]



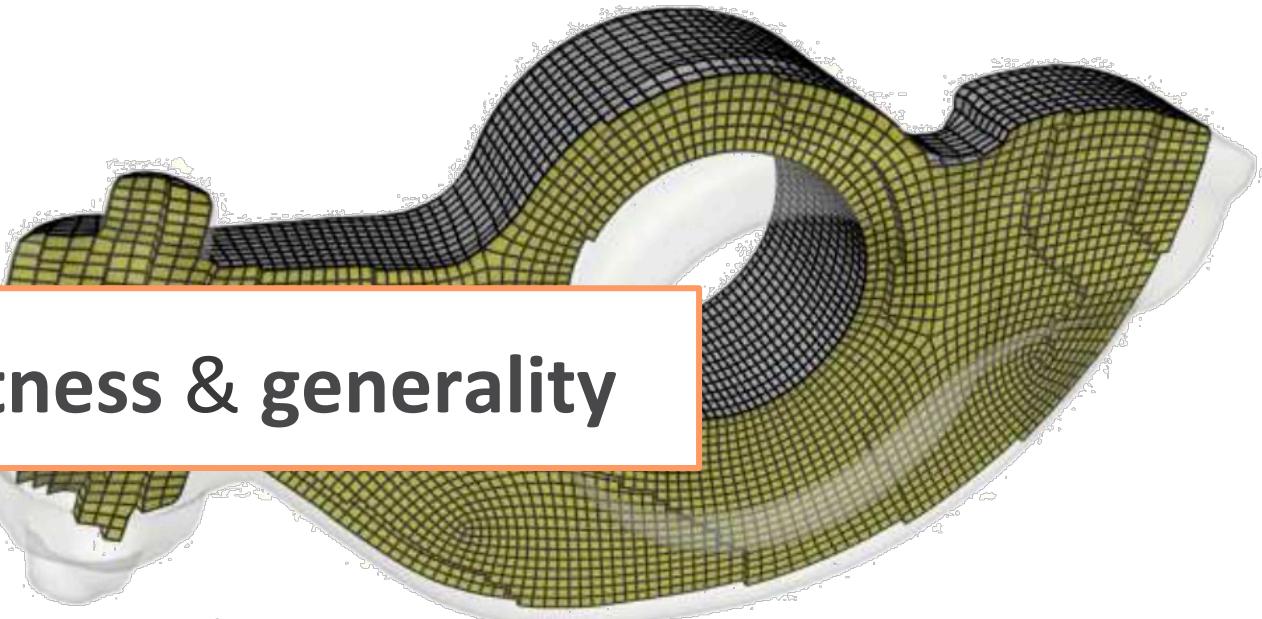
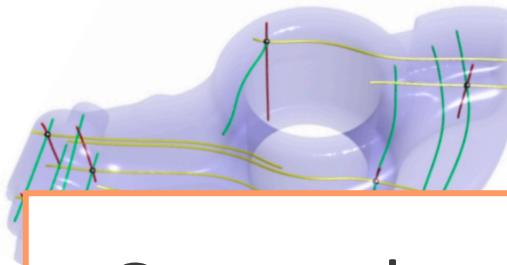
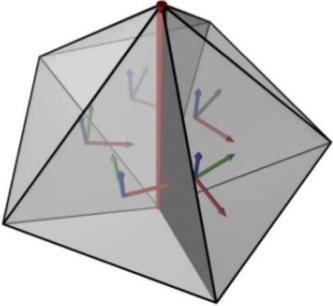
✓ Robust

✗ Not general enough
(Interior always forced to become regular grid)



State-of-the-art (2/2): Field-guided parameterization

[Nieser et al. 11, Huang et al. 11, Li et al. 12, Jiang et al. 14, Lyon et al. 16, Solomon et al. 17, Liu et al. 18]



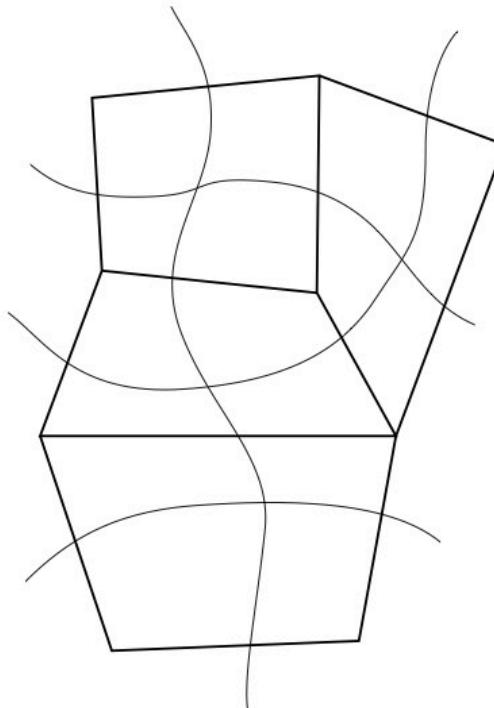
Our goal: **robustness & generality**

✓ General & high-quality
(Interior NOT forced to become regular grid)

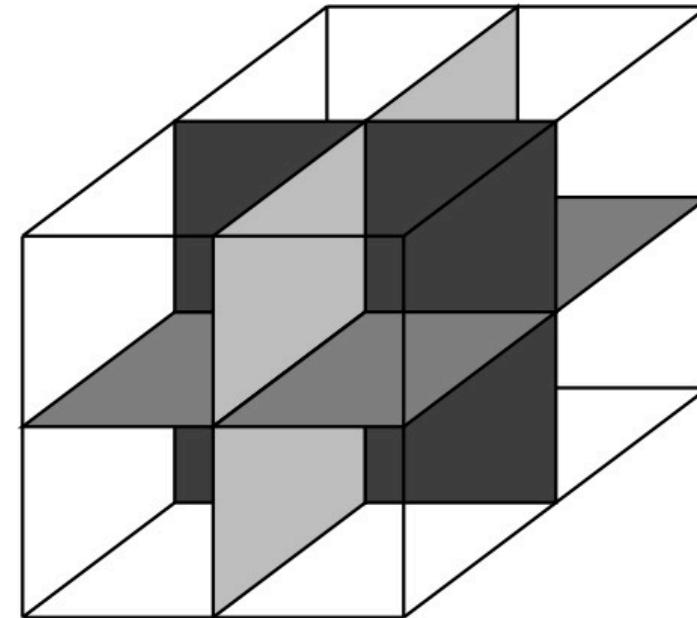
✗ Often fails
(Due to inadmissible types of field singularity)

Dual of quad/hex mesh

Images courtesy of [Eppstein 96]



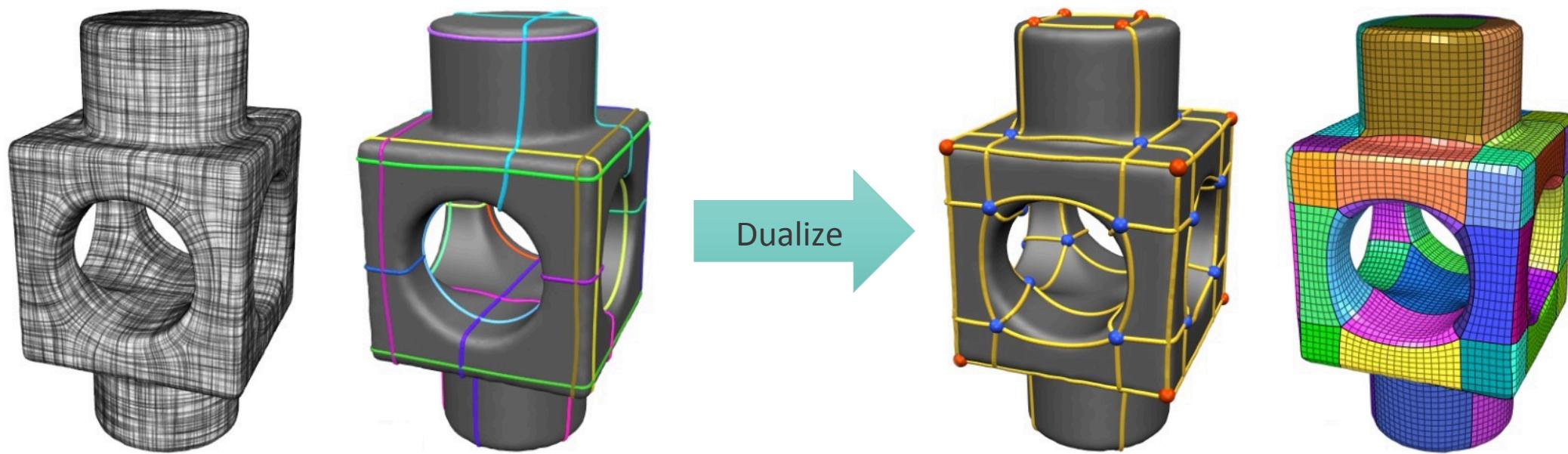
- Connect midpoints of opposite edges
→ Arrangement of curves
- Intersection of 2 curves == one quad



- Connect midpoints of “parallel” edges
→ Arrangement of surfaces
- Intersection of 3 surfaces == one hex

Inspiration: dual-based quad meshing

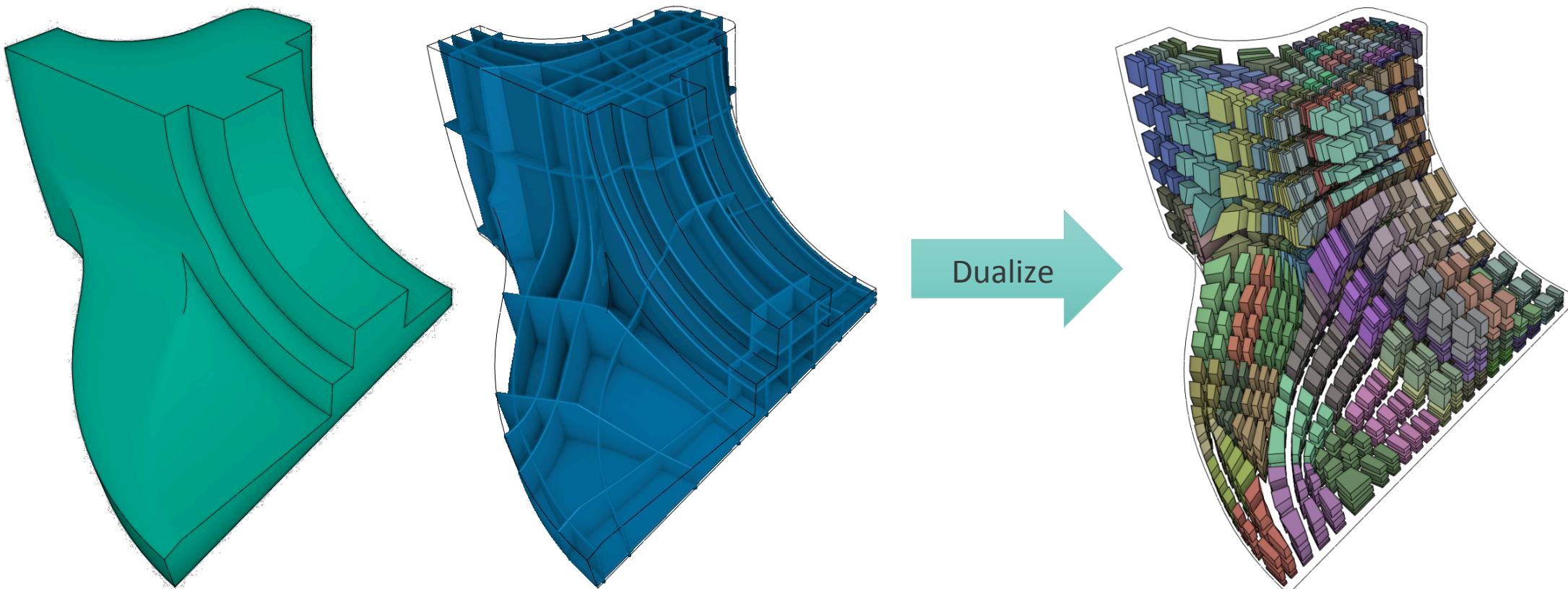
[Campen et al. 12, Campen & Kobbelt 14]



1. Generate enough loops (greedy [2012], controllable [2014])
2. Take dual → quad mesh

Let's take this to 3D!

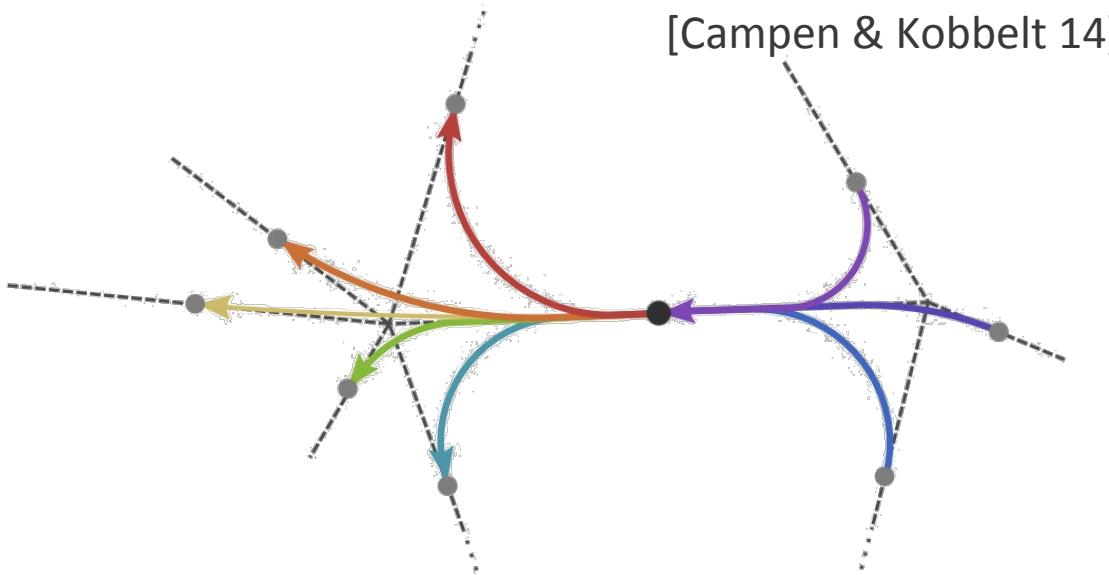
Our work: dual-based hex meshing



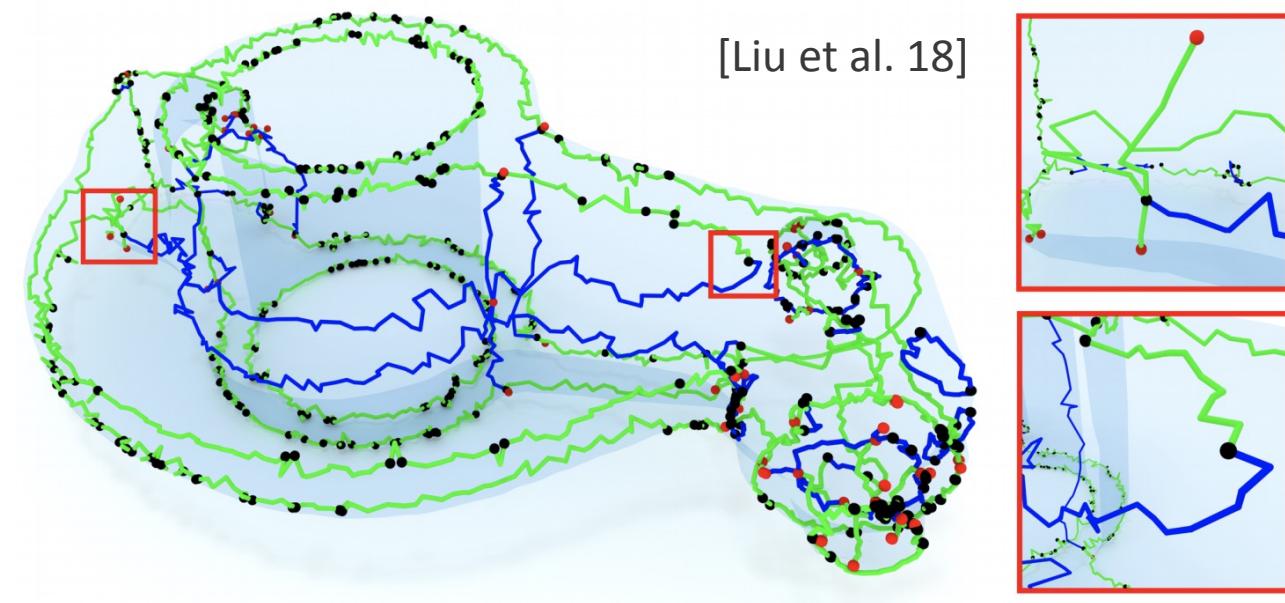
1. Generate enough sheets somewhat How?
2. Take dual → hex mesh

Automatically generating dual sheets seems hard!

- Campen et al.'s method not generalizable to 3D:

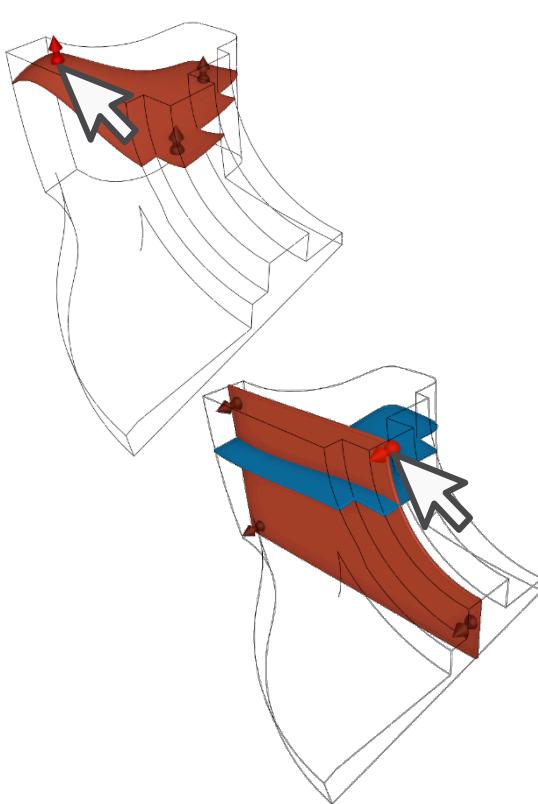


Anisotropic geodesic can find 1-manifolds only

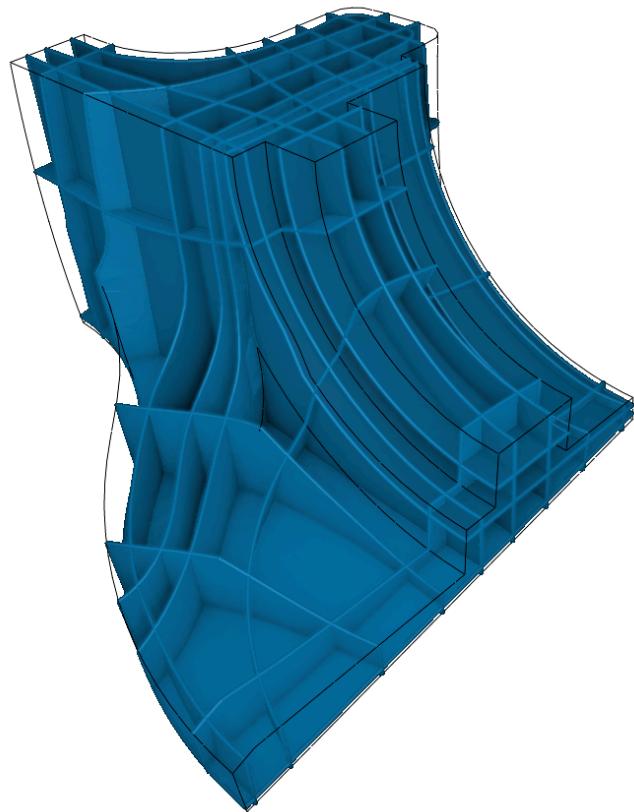


Field singularity is much more complex & noisy

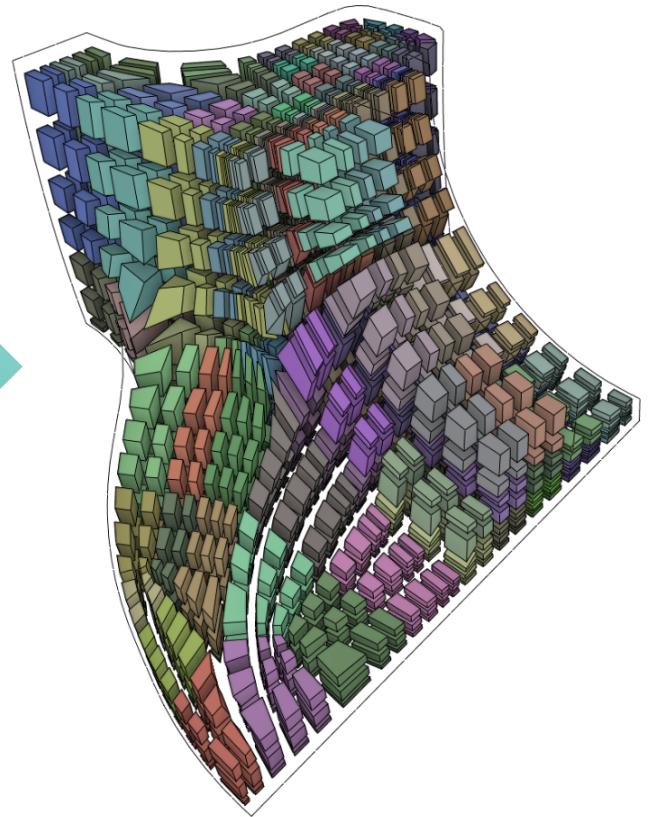
Our approach: manual 3D modeling



...

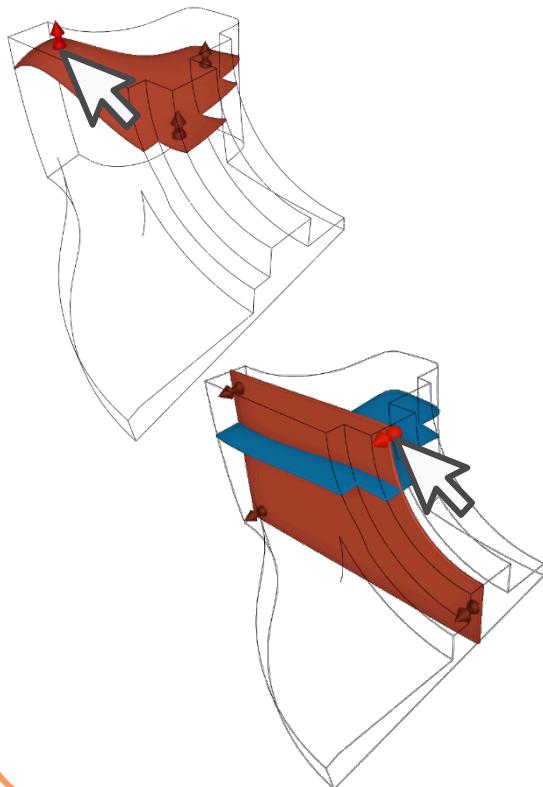


Dualize

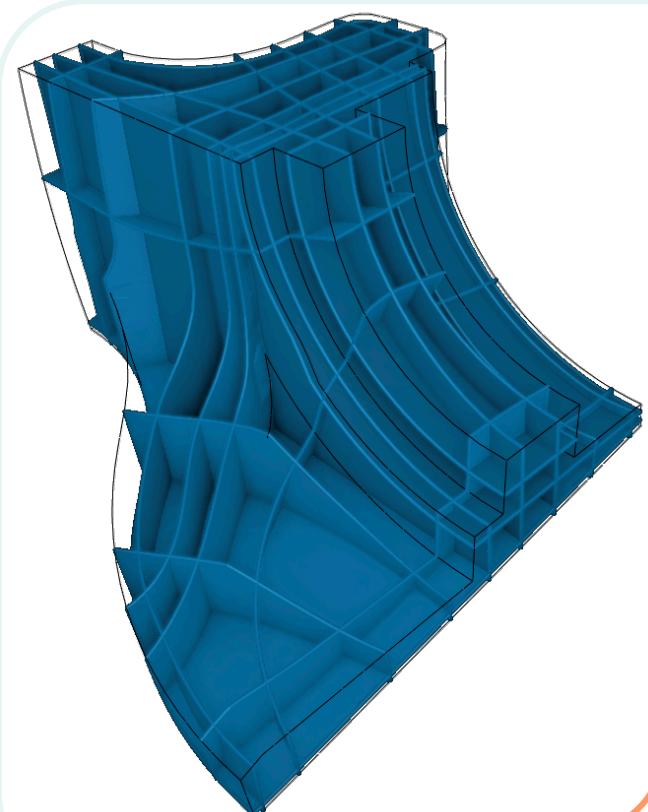


Talk outline

1. User interface

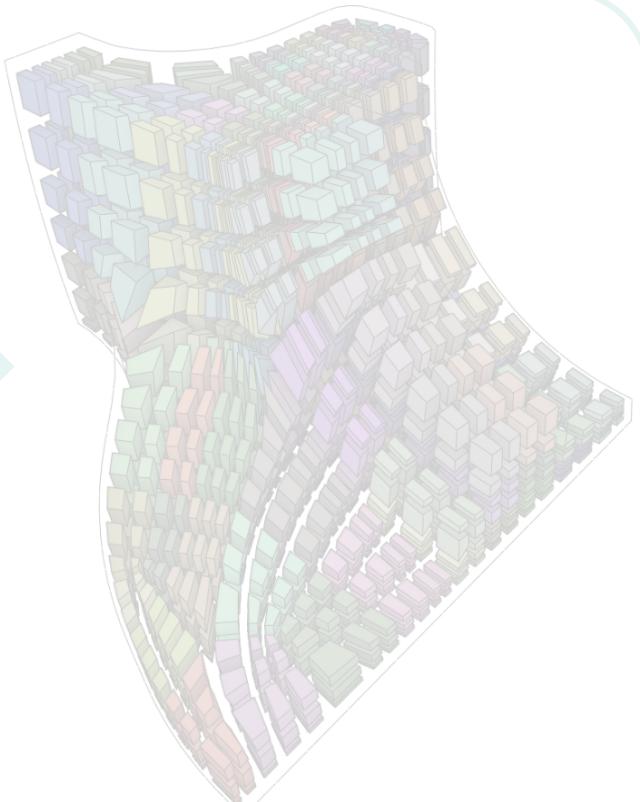


...

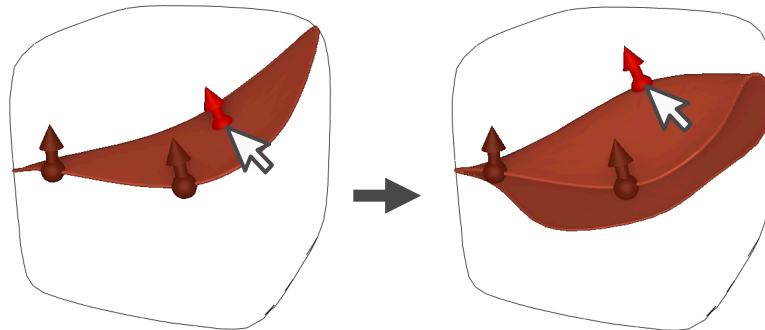


2. Algorithm

Dualize

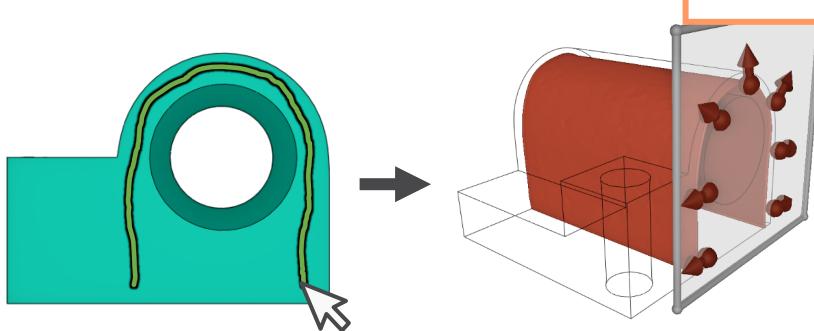


3 tools for creating sheets

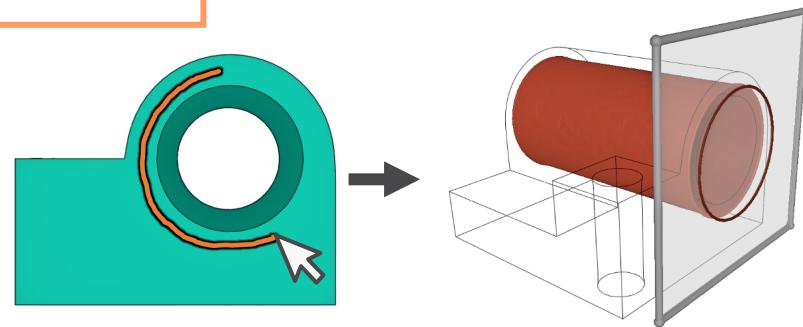


Freeform tool

Demo

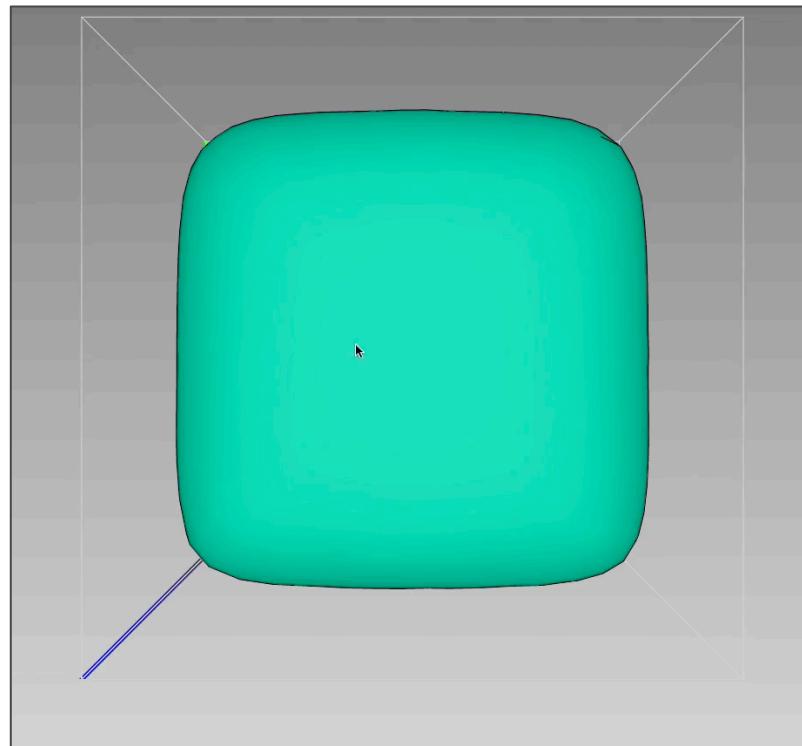


2D Sketch tool

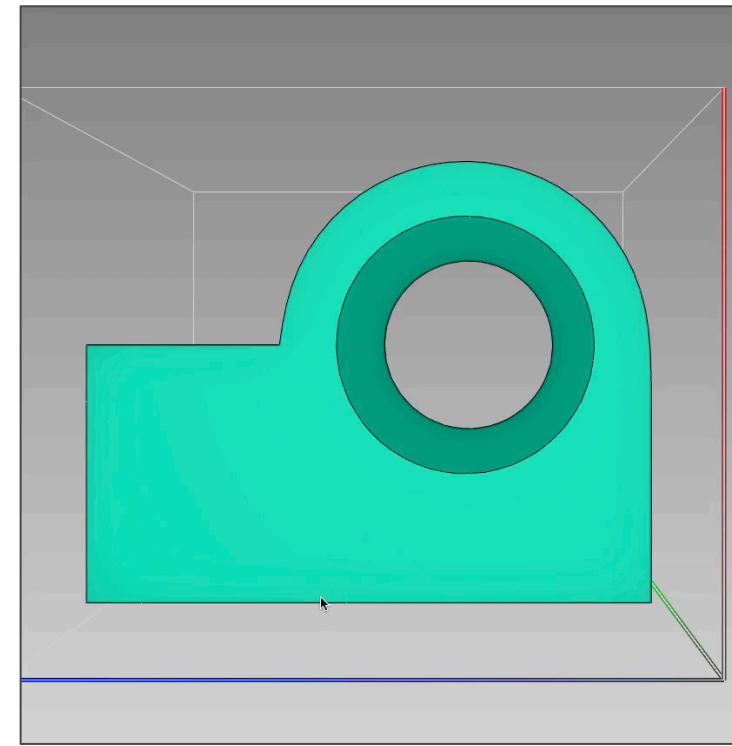


Cylinder tool

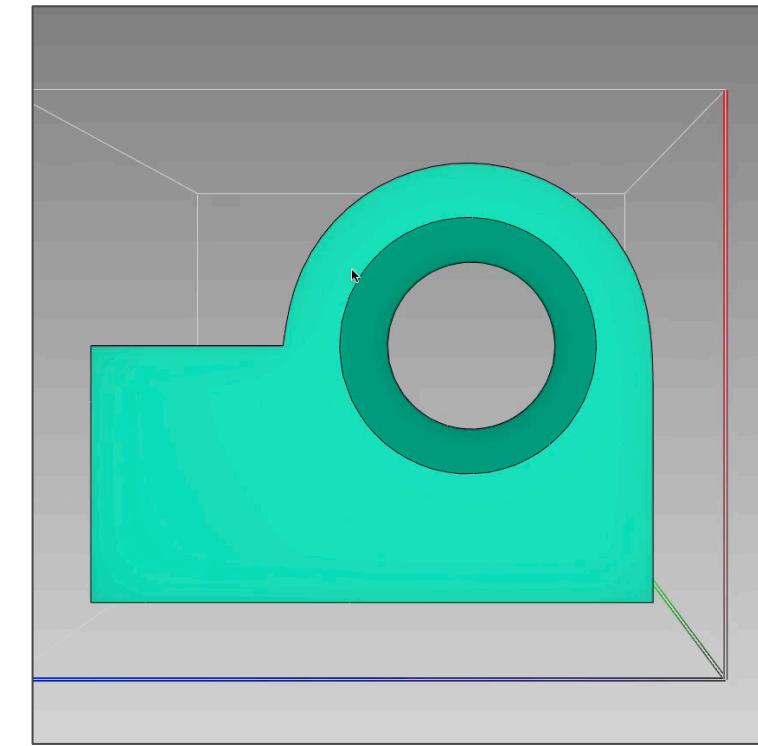
3 tools for creating sheets



Freeform tool

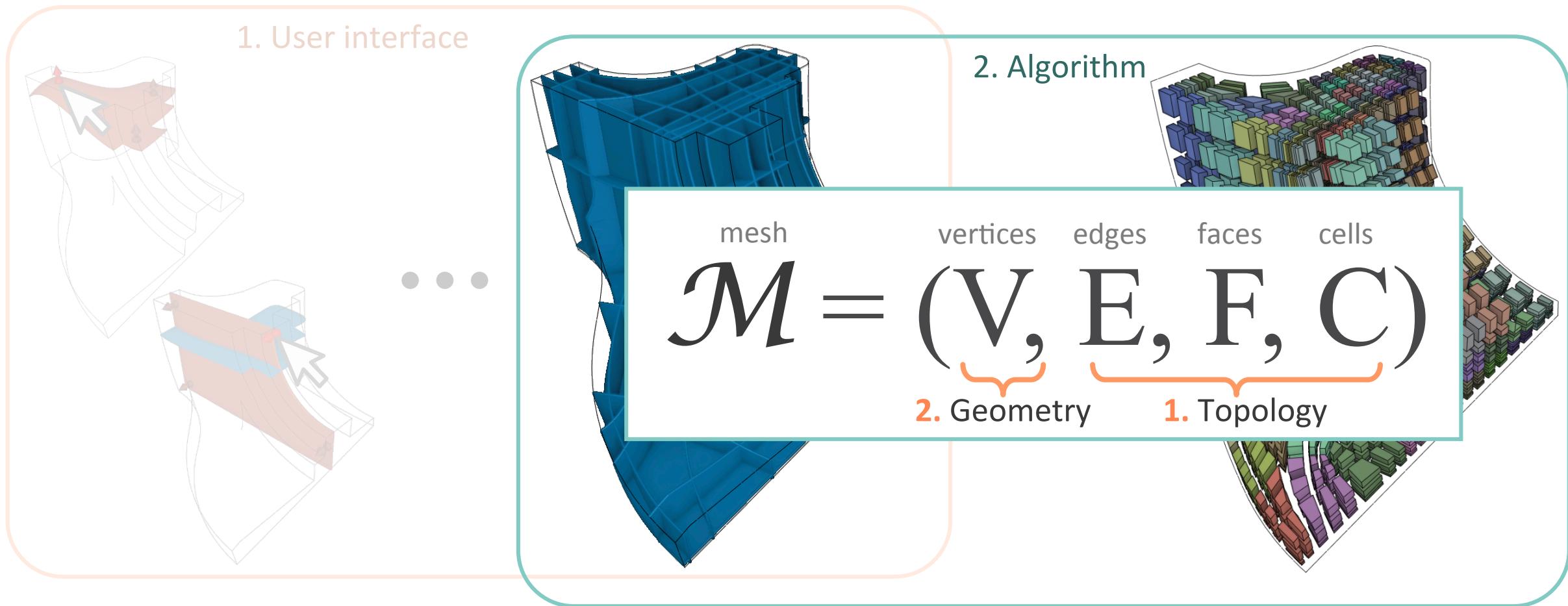


2D Sketch tool



Cylinder tool

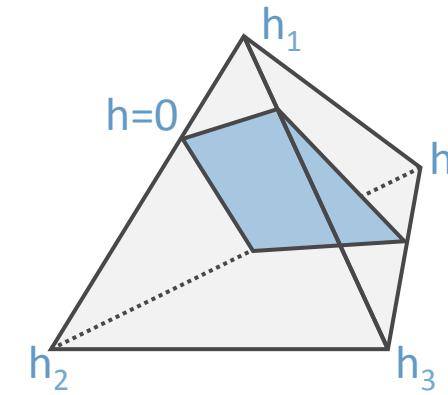
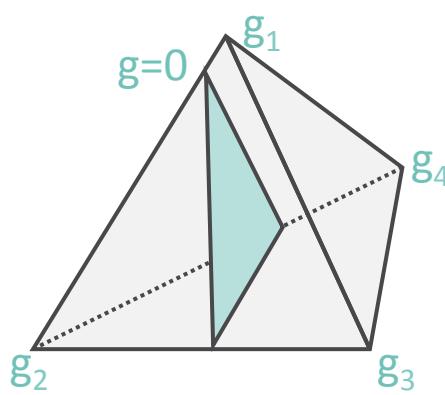
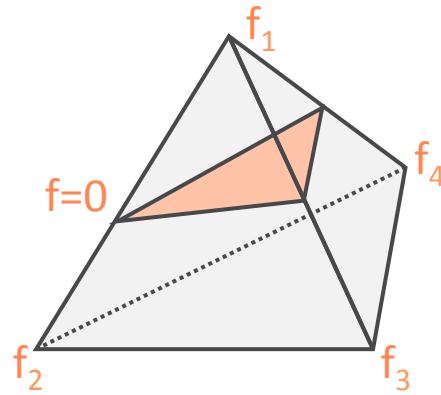
Talk outline



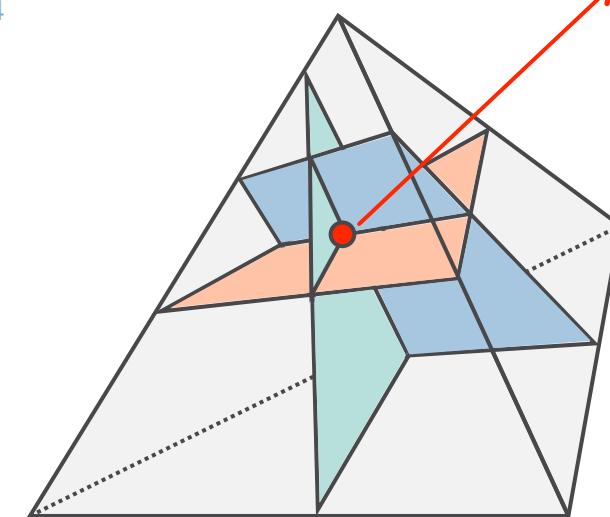
Generating topology (1/2): find sheet intersections

- Design decision: represent sheets as zero-isosurfaces
 - Piecewise-linear over tet mesh

→ makes finding sheet intersections very easy

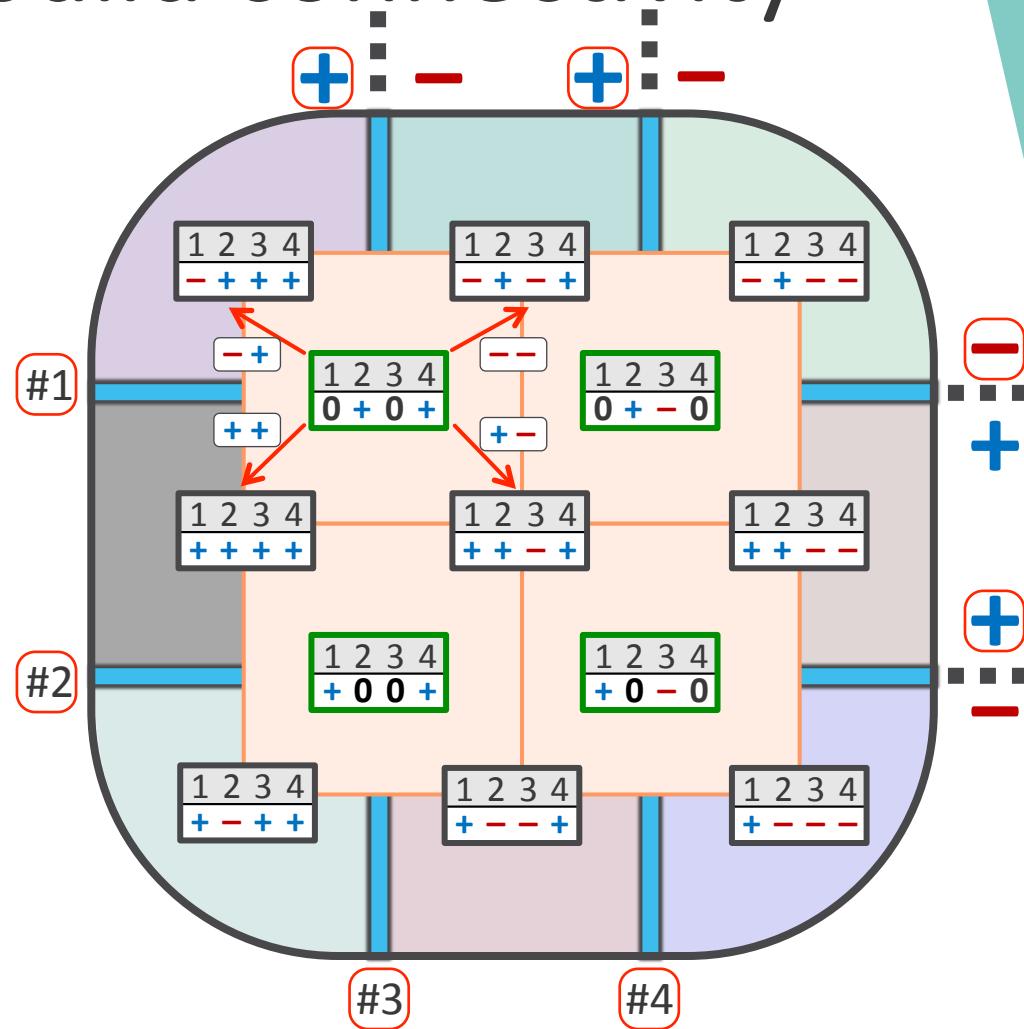


$$\begin{bmatrix} f_1 & f_2 & f_3 & f_4 \\ g_1 & g_2 & g_3 & g_4 \\ h_1 & h_2 & h_3 & h_4 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$



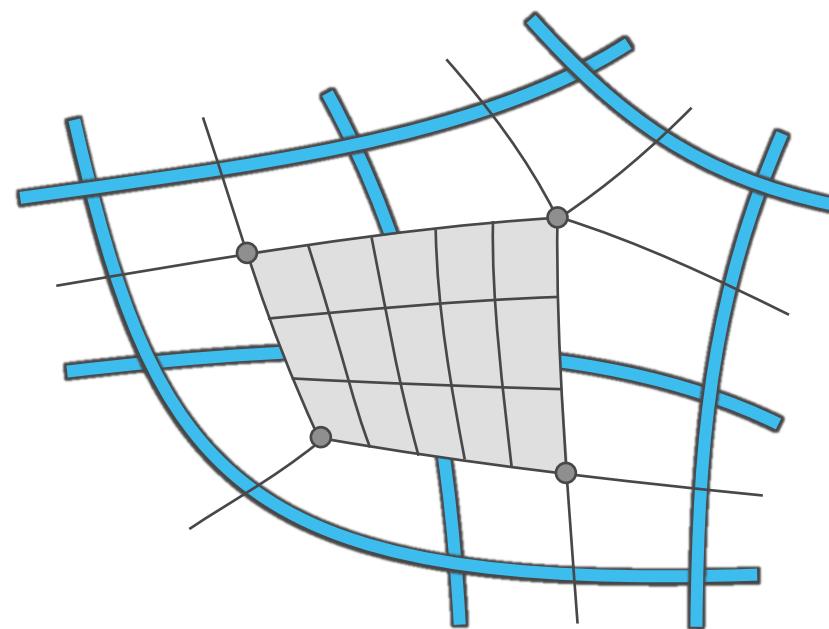
Generating topology (2/2): build connectivity

- *Vertex code*: array of signs
- *Cell code*: two digits set to 0
 - Derives 4 vertex codes by substituting all possible sign combinations:
 $\{ (++), (+-), (-+), (--) \}$
- Call `add_cell()` for each cell code



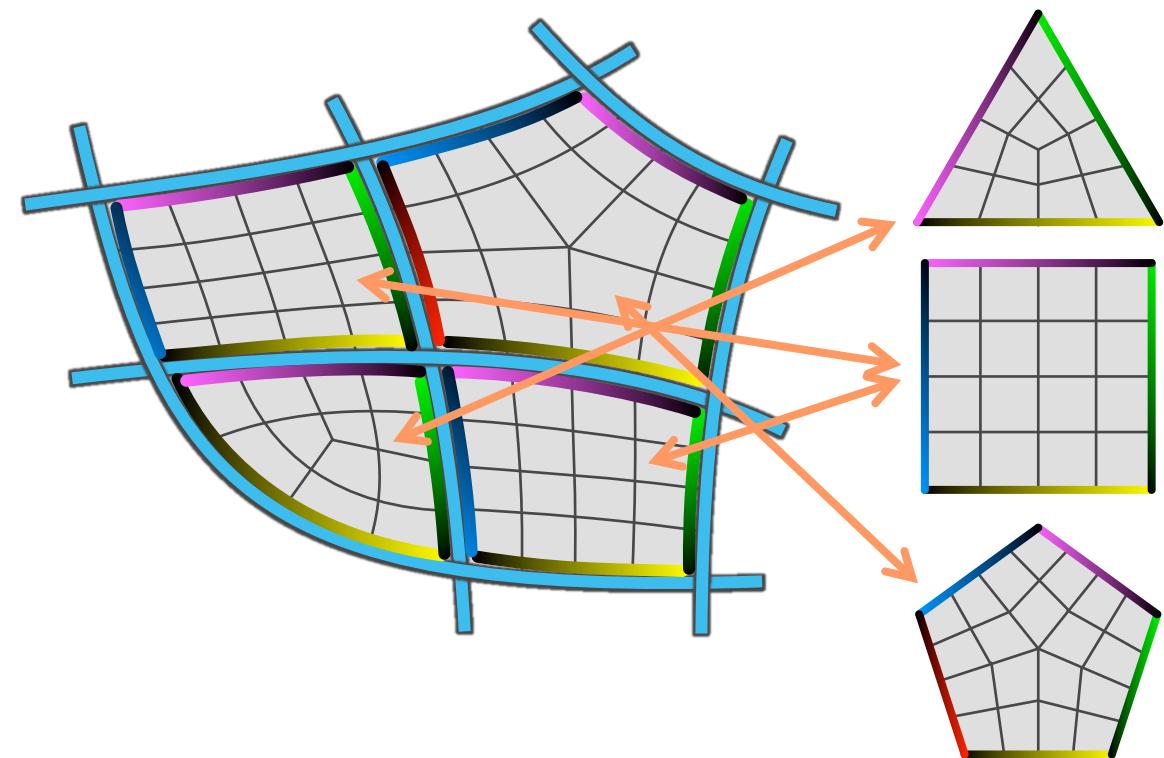
Generating geometry

- Campen et al.'s approach:
 1. Generate irregular vertex in each irregular dual face
 2. Trace separatrices using anisotropic geodesic
Not generalizable!
 3. Intersect separatrices as regular vertices
 4. Parameterize each quad domain



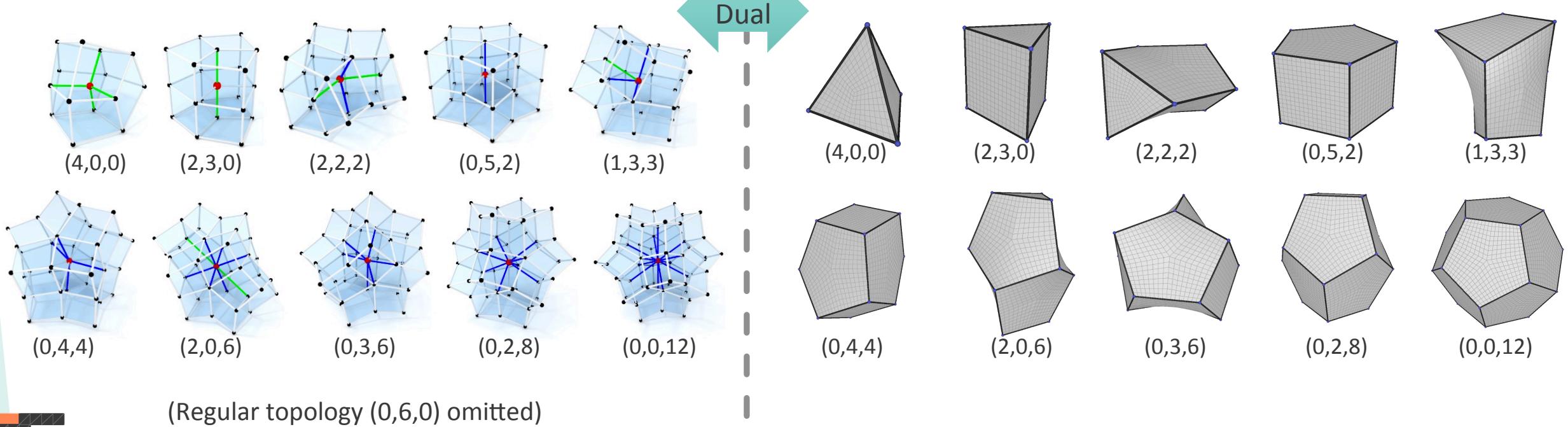
Generating geometry

- Our approach:
 - Map each dual face onto a regular polygon using harmonic parameterization
- Generalizable to 3D
 - In 2D, map to regular N-gon
 - In 3D, map to ?



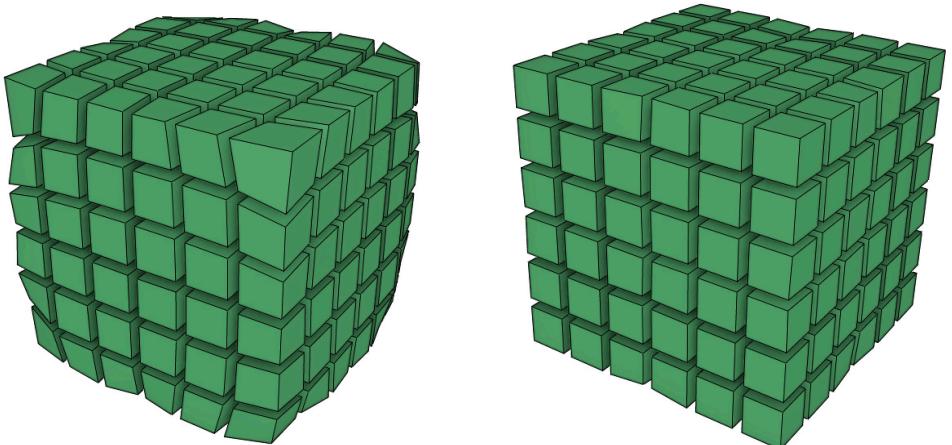
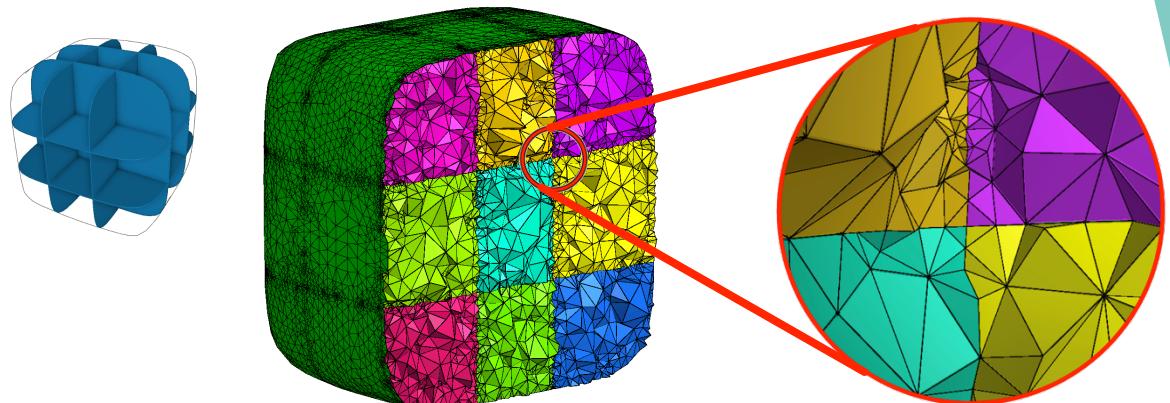
Useful observation in [Liu et al. 18]

- Assume:
 - “Every edge valence is {3, 4, 5}”
- All vertex topologies fall into only 11 types!
- Assume:
 - “Every dual face is {3, 4, 5}-gon”
 - All dual cell topologies fall into only 11 types! → *reference polyhedra*



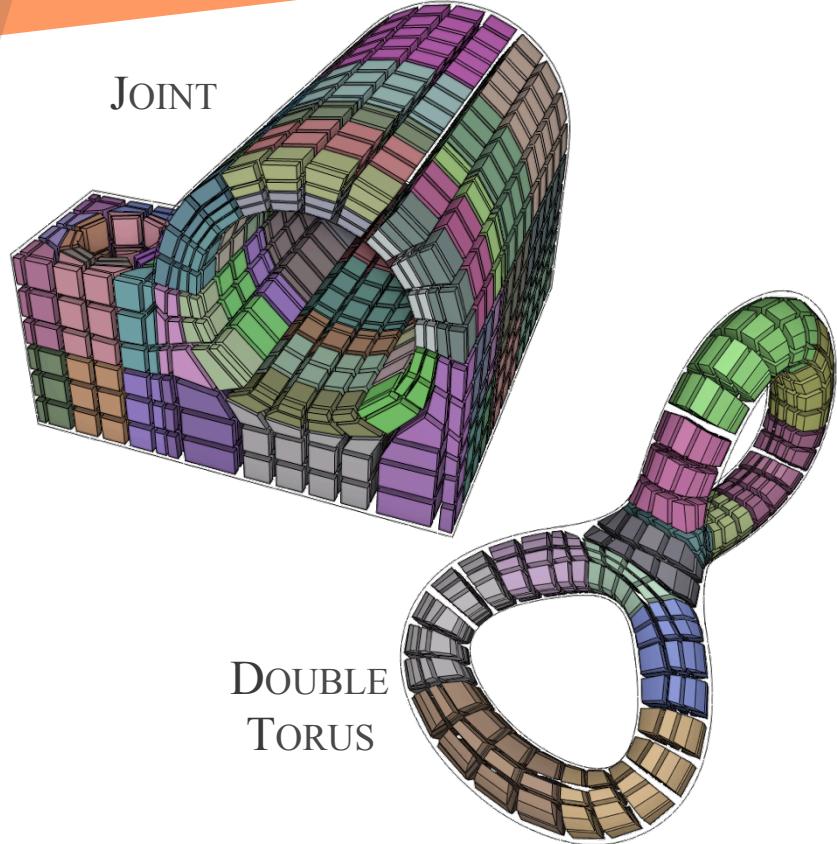
Implementation details

- Volume discretized by tet mesh
 - Conforming to sheet intersections
- Preservation of sharp features
 - By modifying boundary conditions

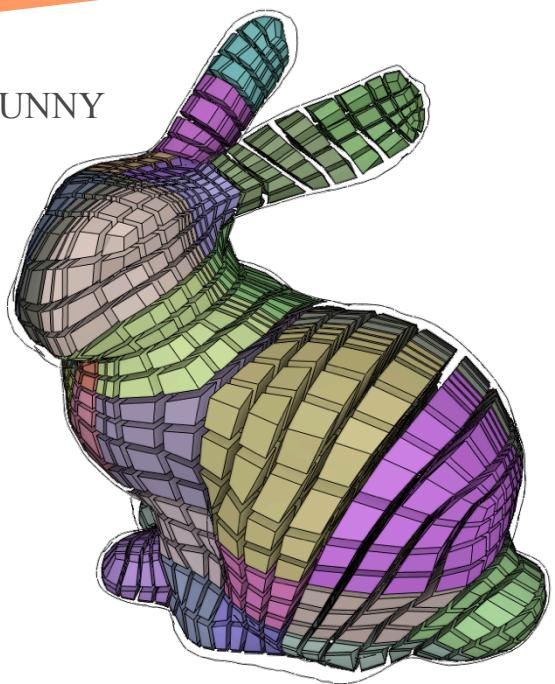


Results

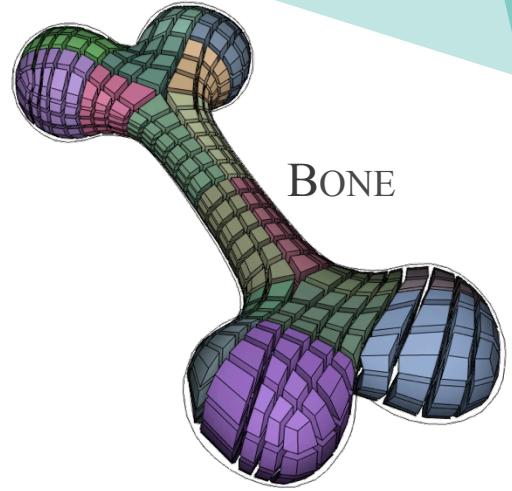
JOINT



BUNNY



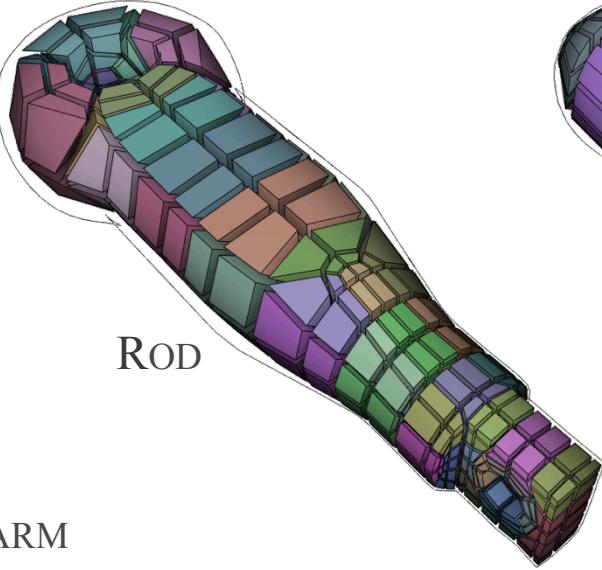
BONE



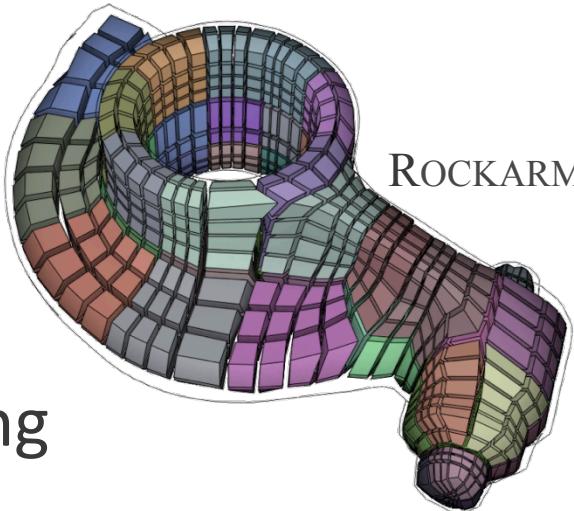
DOUBLE
TORUS



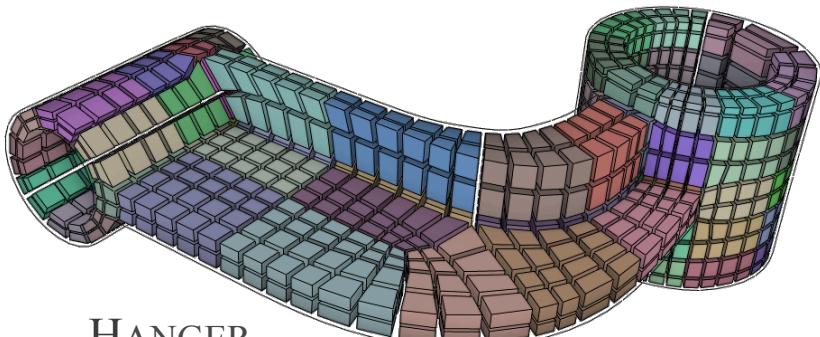
ROD



ROCKARM

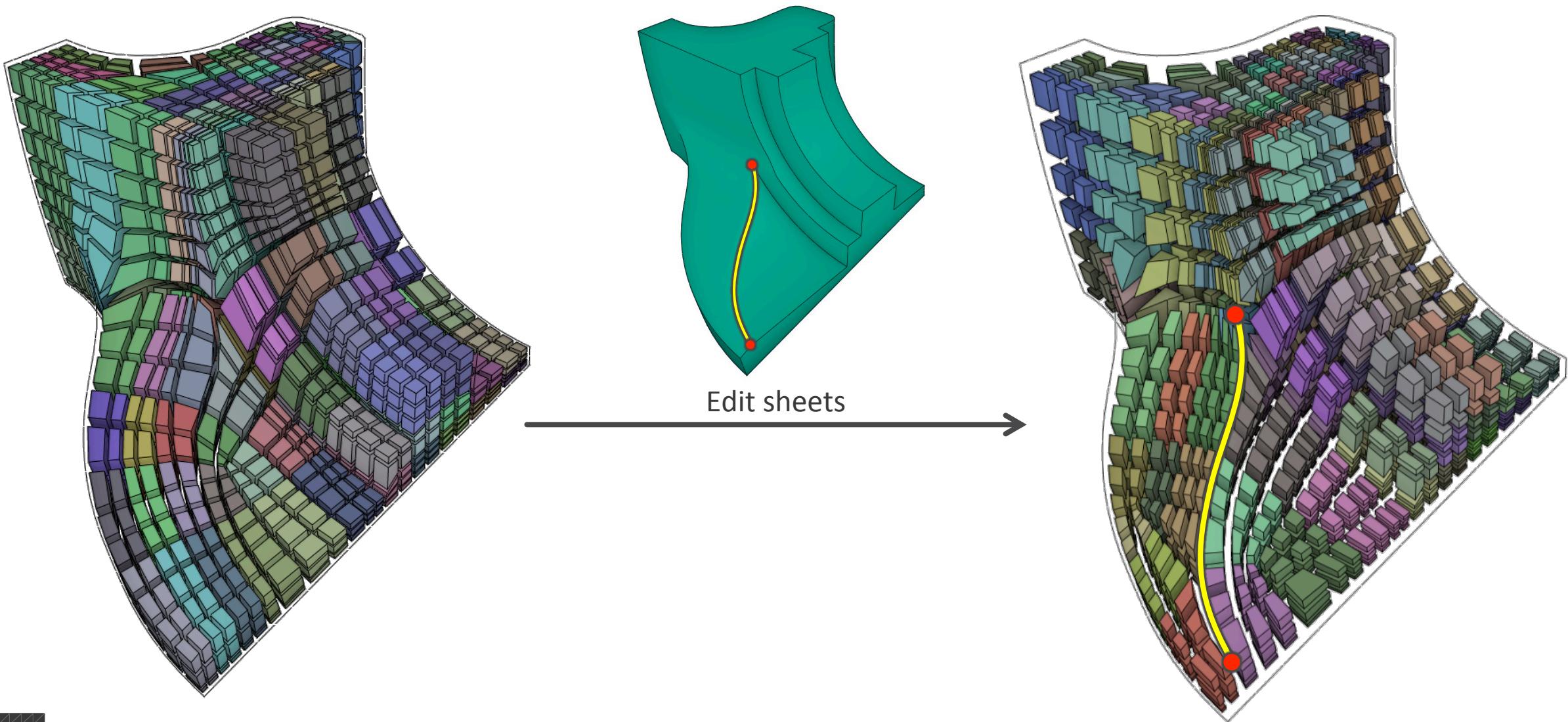


HANGER



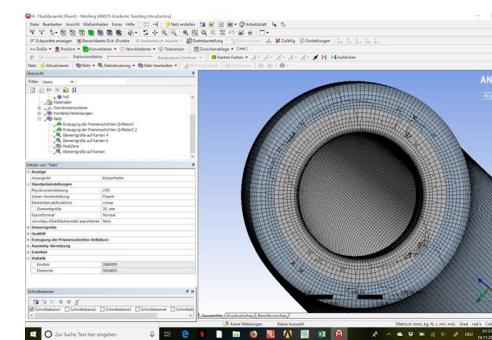
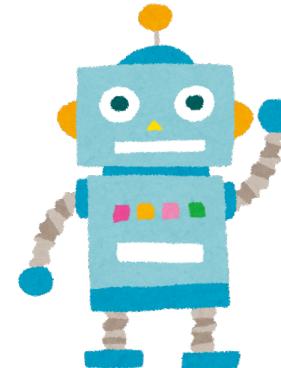
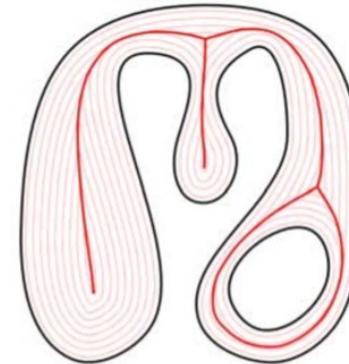
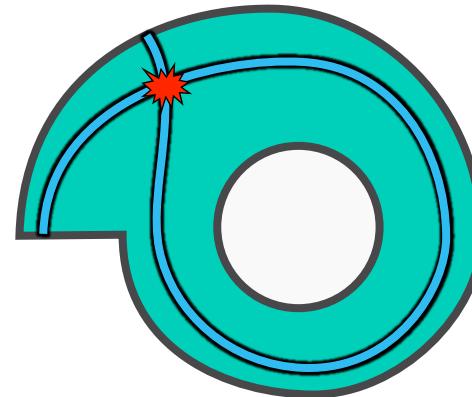
- All created by me
- A few hours for 3D modeling
- 3~5 min for dualization

Use case: edit existing topology



Limitations & future work

- Self-intersecting sheets
 - Currently not possible due to implicit representation
- More intelligent sheet creation
 - Geometric cues (e.g., skeletons)
 - Partial automation (machine learning...?)
- User study
 - Comparison against commercial tools



ANSYS®

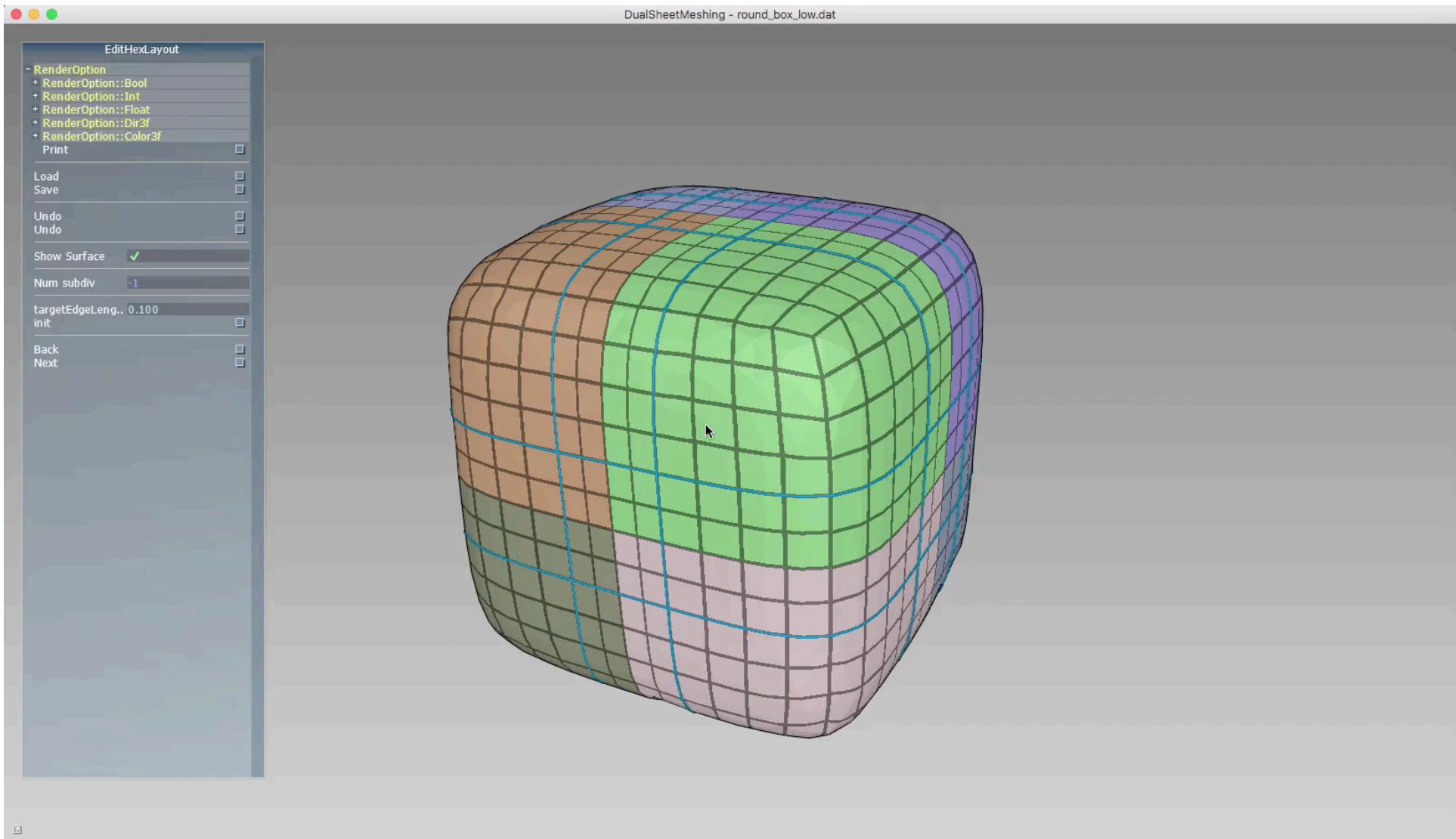
Acknowledgements:

- Olga Sorkine-Hornung
- Daniele Panozzo
- Anonymous reviewers
- JSPS KAKENHI (15K15999)



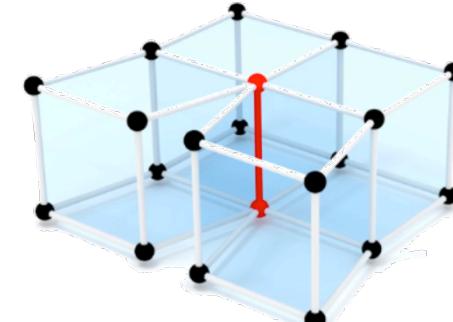
<https://bitbucket.org/kenshi84/dual-sheet-meshing>

Layout previewing



Future work

- We exclude boundary edge of valence 4
- [Liu et al. 18] supports much wider range of boundary vertex topologies by enumerating significantly more (237) types
- We'd need to generate reference polyhedra for those 237 types



[Liu et al. 18]