ERC Seminar series

Automated CFD Meshing with focus on Internal Combustion Engines

Federico Perini, Ph.D.

UW-Madison, Madison, WI

June 14th, 2024

About me

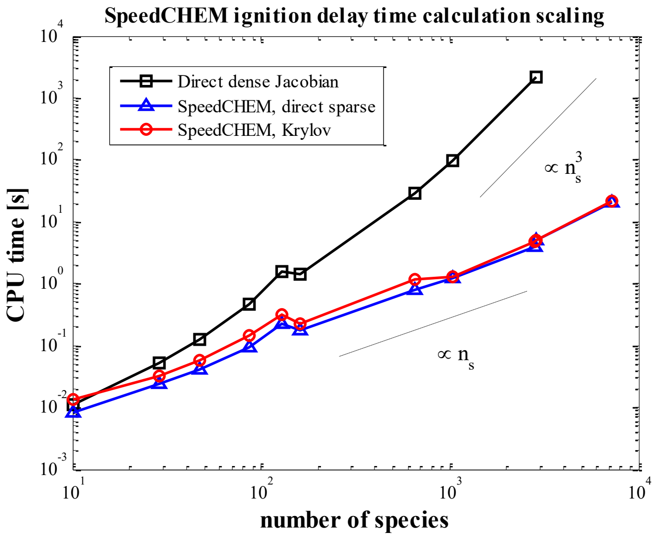
- MSc, Ph.D. Engines Modena, Italy
- Post-Doc UW
- Consultant at WERC
- Working on FRESCO, my combustion CFD code

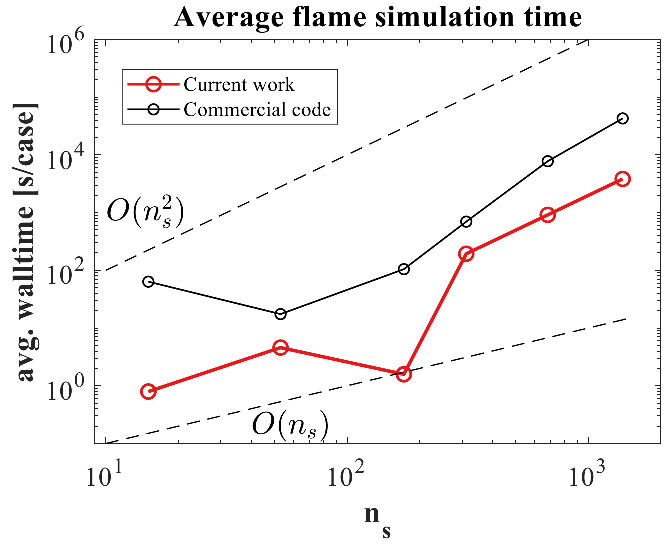
About FRESCO

- **<u>FRESCO</u>: Unstructured, parallel CFD solver with body-fitted mesh**
- **fast detailed chemistry thanks to Sparse Analytical Jacobians**

FRESCO setup	
Mesh	Body-fitted sector
Time accuracy	O(Δt) implicit (diffusion) / explicit (advection, CFL=0.2)
Finite Volumes	O(Δx²) – diffusion w/ exact Jac., O(Δx) – advection (Upwind, minmod)
Parallelism	MPI (cpu), OpenCL (gpu)
Chemistry	direct, Sparse Analytical Jacobian (SpeedCHEM-II)
Combustion	G-Equation with detailed chemistry
Turbulence	Generalized re-normalization group (GRNG) k-ε
Spray	Dynamic Blob, KH-RT, gas-jet SGS, extended collisions, 1D multicomponent vaporization

$$\begin{cases} \dfrac{\partial Y_i}{\partial t} = \dfrac{\dot{\omega}_i W_i}{\rho} - \dfrac{1}{\tau}(Y_i - Y_i^*), \qquad i = 1, \dots, n_s \\ \dfrac{\partial T}{\partial t} = \dfrac{1}{c_v}\left[\dfrac{\dot{Q}}{V} - \sum_{i=1}^{n_s} \dfrac{U_i \dot{Y}_i}{W_i} + \dfrac{1}{\tau}\sum_{i=1}^{n_s} Y_i^*(U_i^* - U_i)\right]. \end{cases}$$

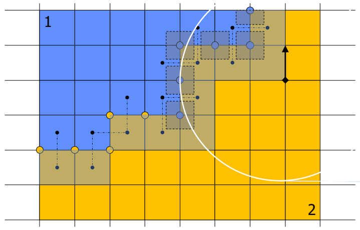




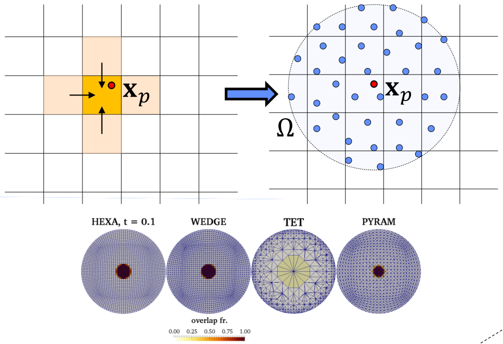
About FRESCO

CFD/Numerics

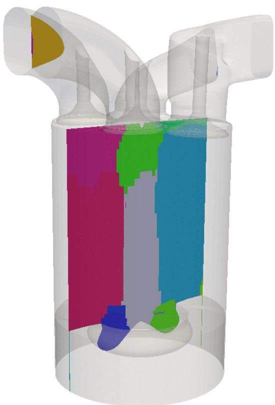
Parallel/MPI



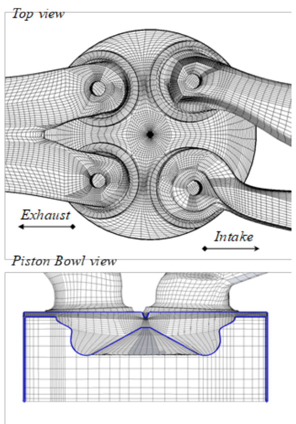
Lagrangian



Decomposition

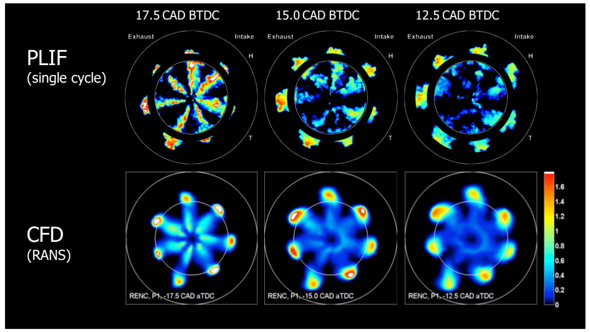


Meshing

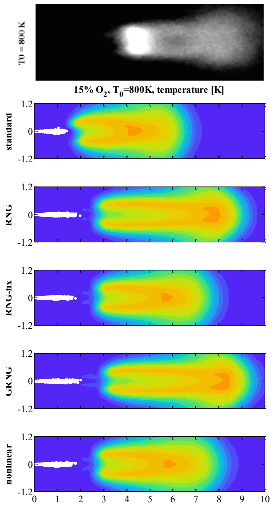


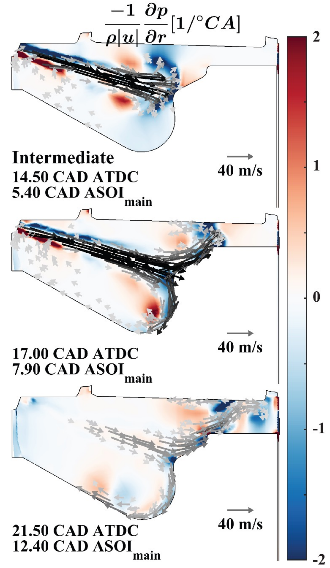
Engine Studies

Mixing

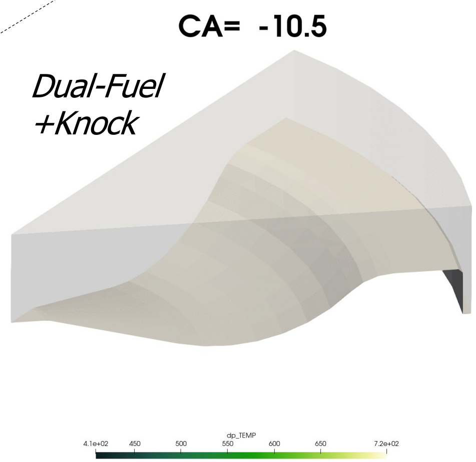


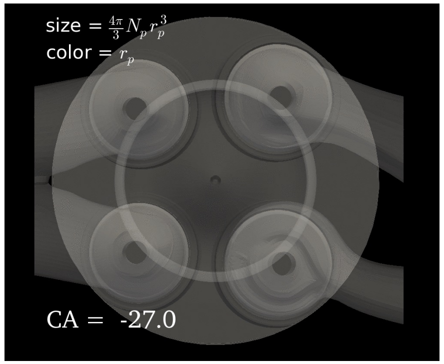
Sprays / Turbulence

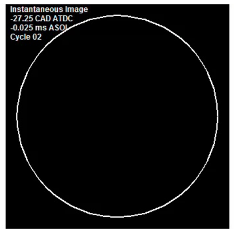




Dual-Fuel +Knock



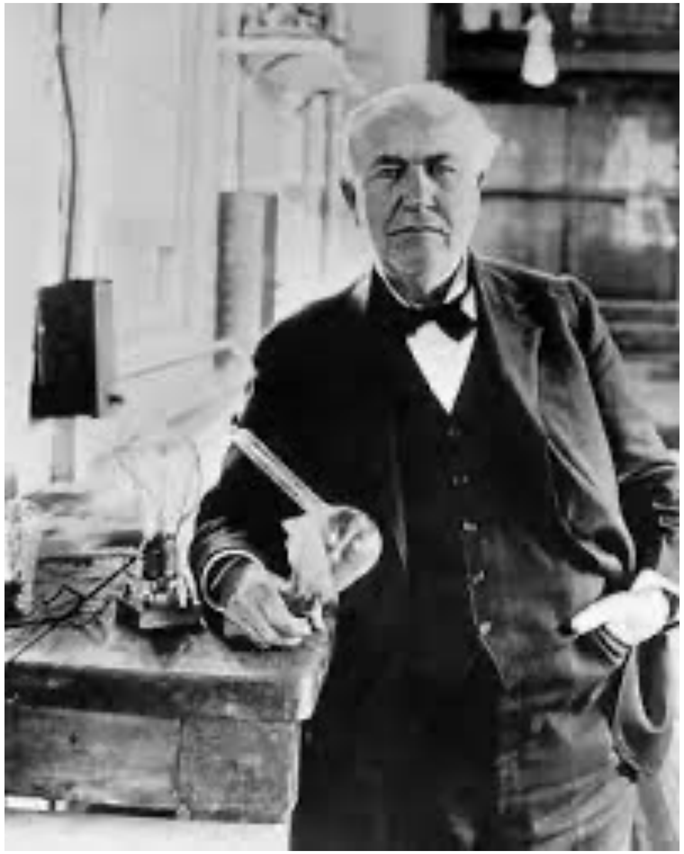




Friday wisdom

**"I didn't fail 1000 times.
The light bulb was an
invention with 1000 steps."**

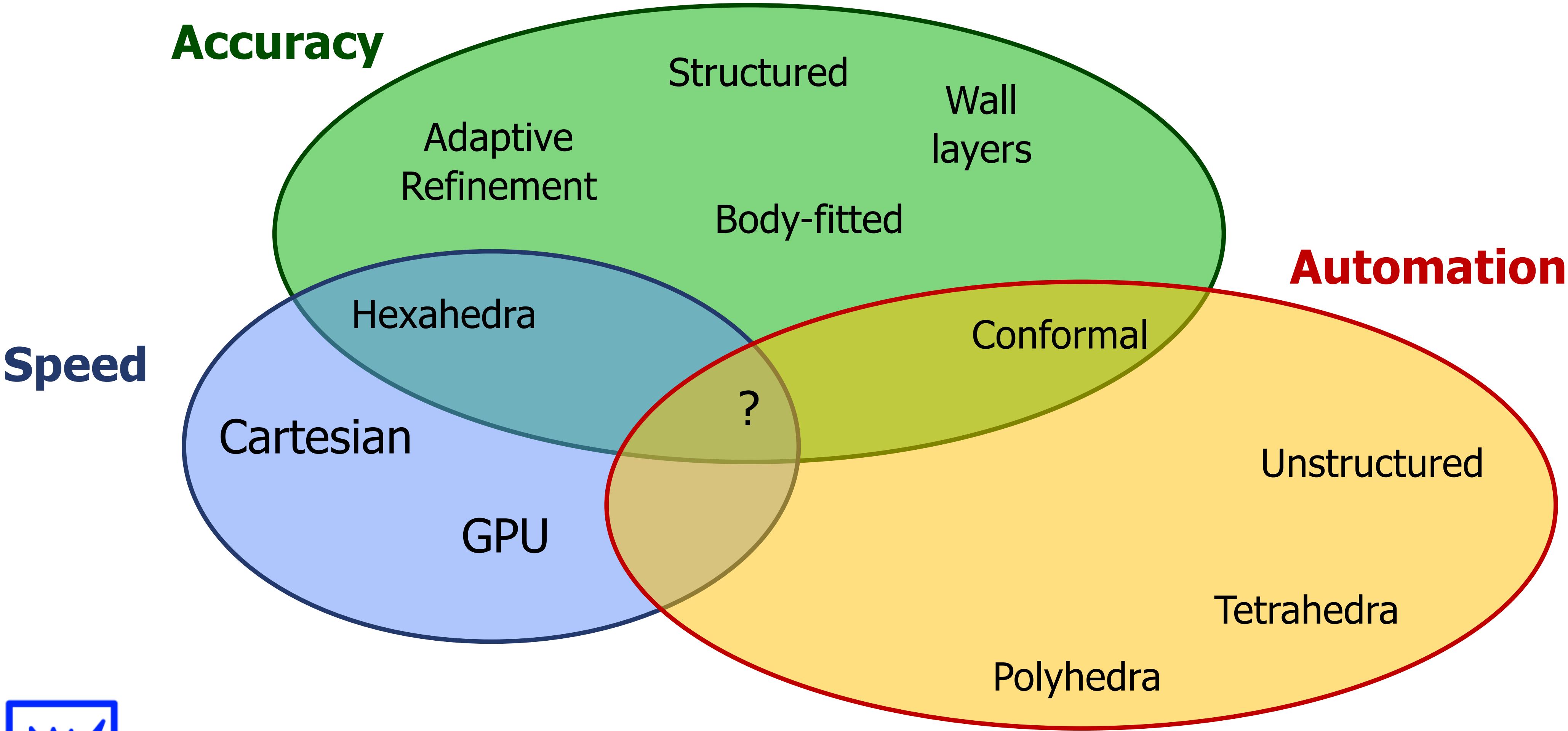
T.A. Edison



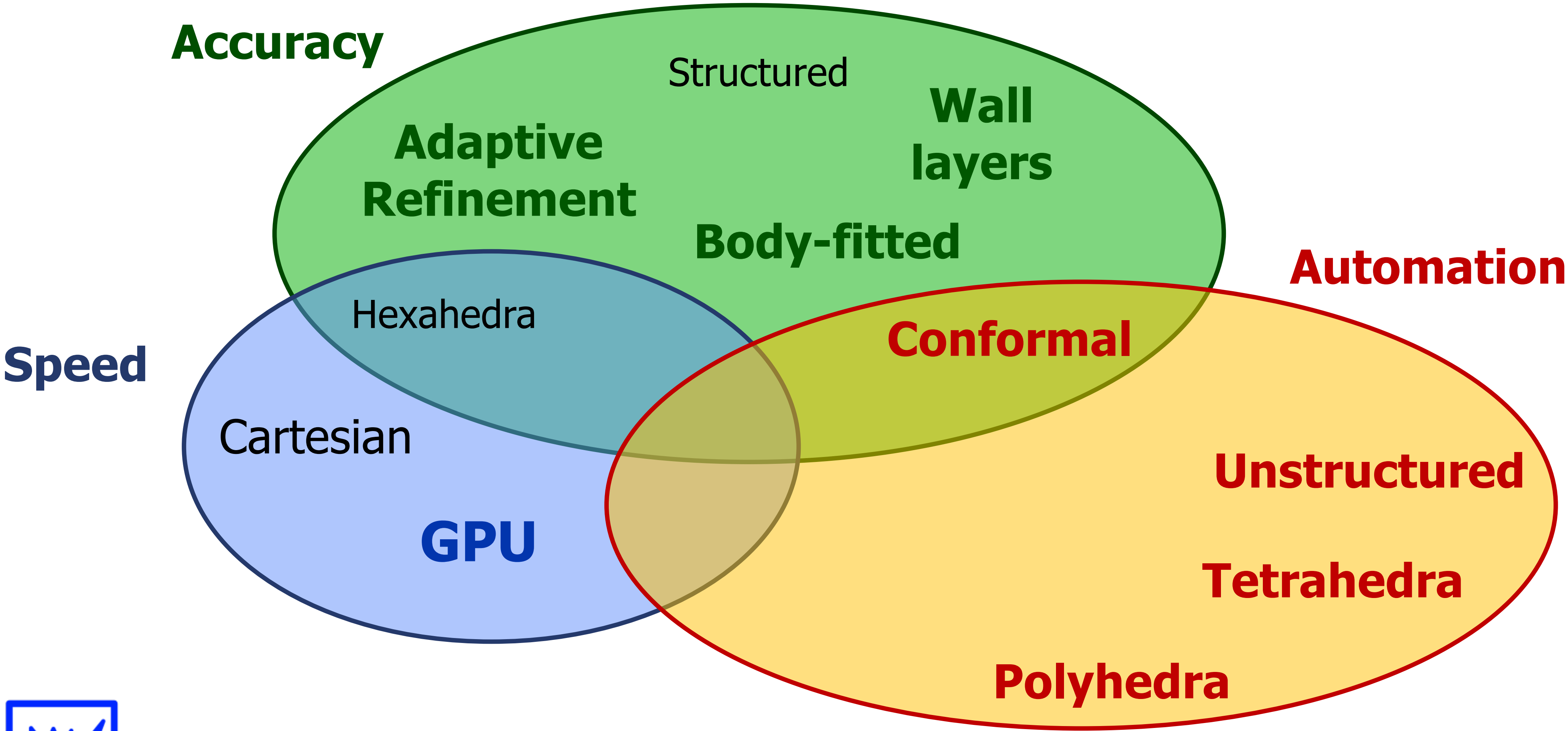
What makes for a great mesh?

- → Body-Fitted → Boundary Layer accuracy
- → Prismatic wall layers → Boundary Layer accuracy
- → Mostly Hexahedra → Solution stability, accuracy
- → Solution-Adaptive → Optimal turnaround time
- → Real-Time → Reduce user time
- → Automated → Reduce user time
- → Good cell aspect ratio → Solution stability, accuracy
- → Moving → Real-world cases

Mesh requirements are competing



Let's aim for the best

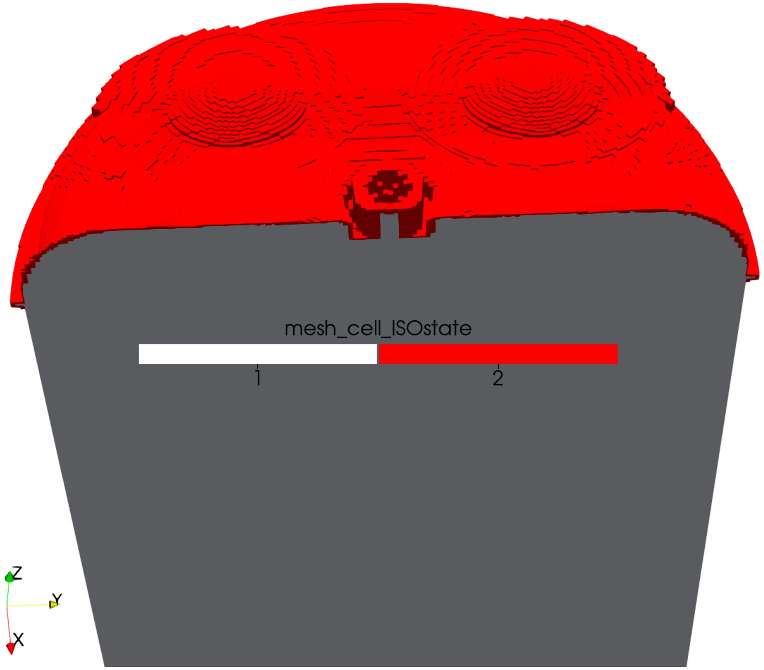


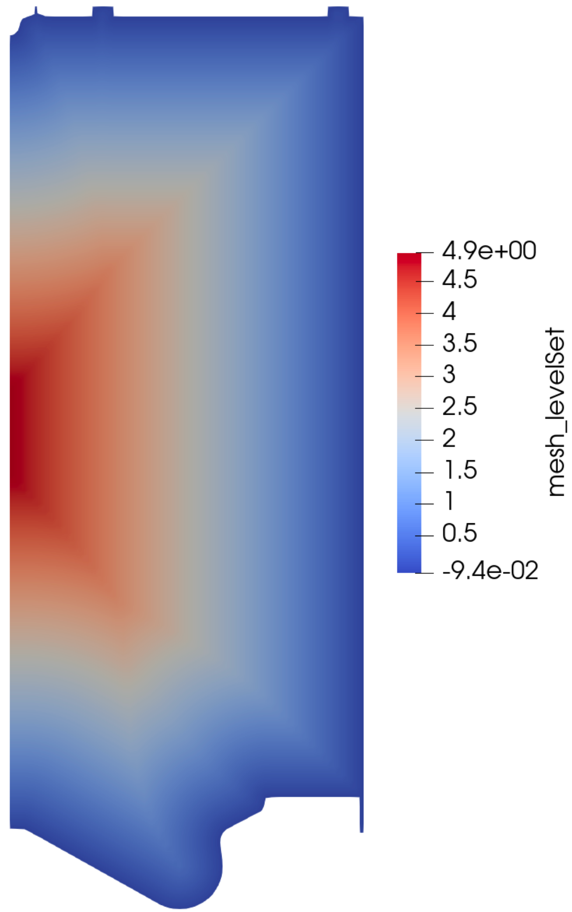
Mesh Level-set

Finite-volume representation of the distance from the wall:

$$G(\boldsymbol{x}) = \begin{cases} \mathrm{G}(\boldsymbol{x}) = \mathbf{0} & \textit{at walls} \\ |\nabla G(\boldsymbol{x})| = \mathbf{1} & \forall \boldsymbol{x} \end{cases}$$

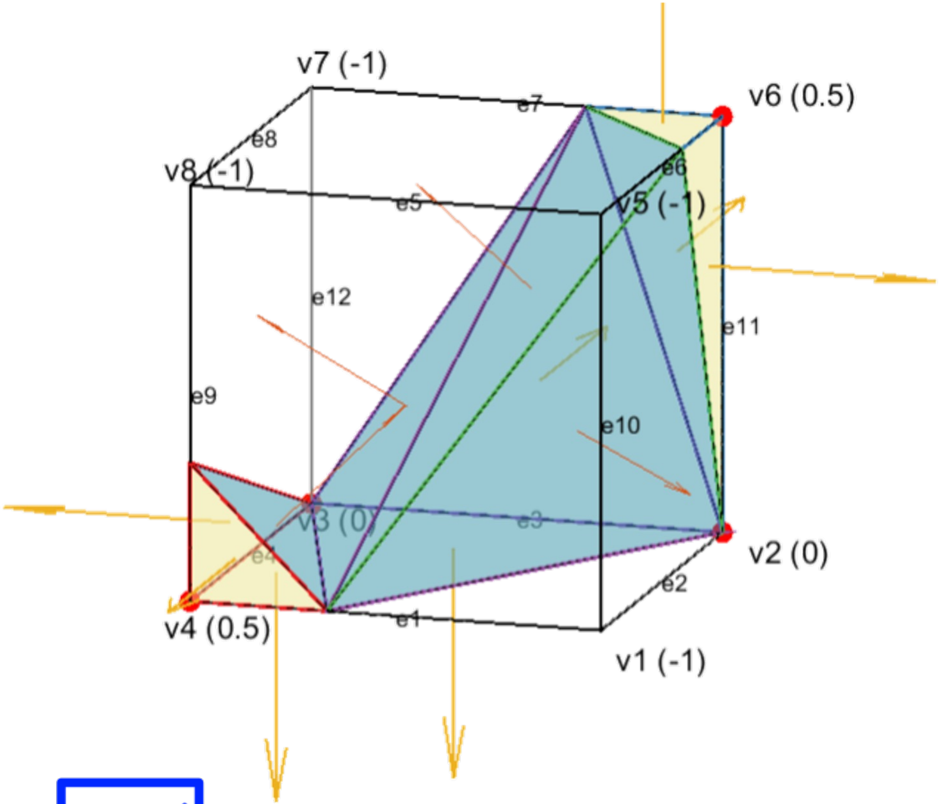
- Classify cells that contain a surface
- Works with any background mesh
- For a surface mesh, the Signed Distance Function (SDF) is used





Real-time mesh with the SDF

- **Create a "Marching Cubes*" method for volume meshing**



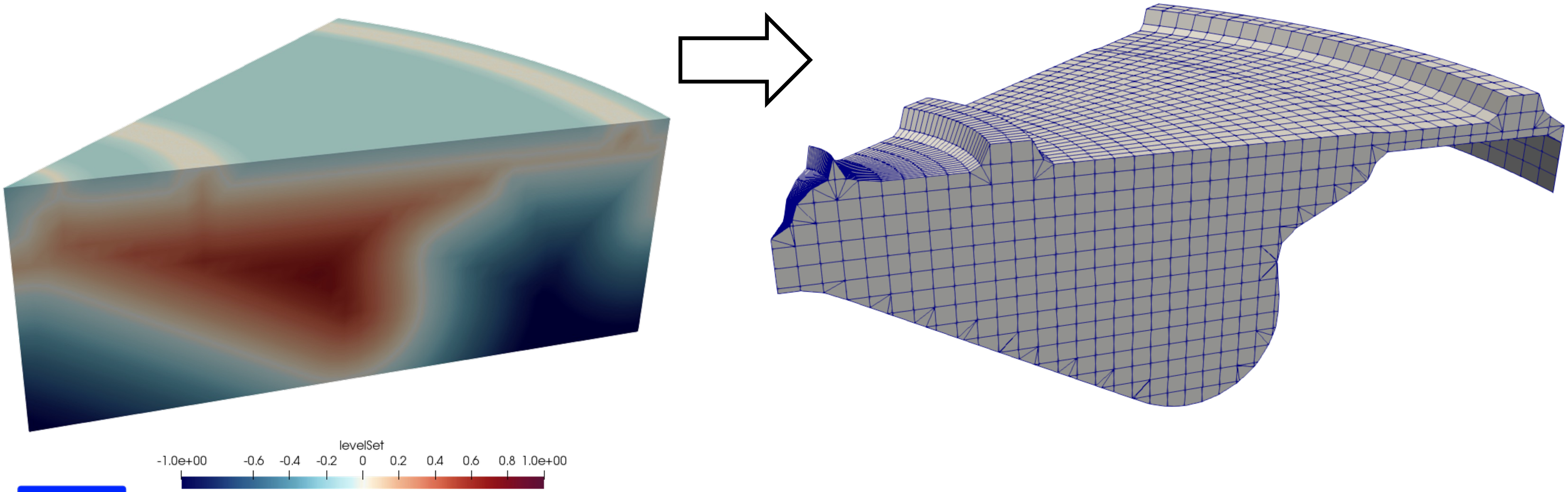
- 3 node states:
  - 00 → G<0 ("low")
  - 01 → G=0 ("iso")
  - 11 → G>0 ("high")

- LUT configurations:
  - $3^8 = 6561$ Hexa
  - $3^6 = 729$ Prism
  - $3^5 = 243$ Pyramid
  - $3^4 = 81$ Tet

*Lorensen, Cline 1987

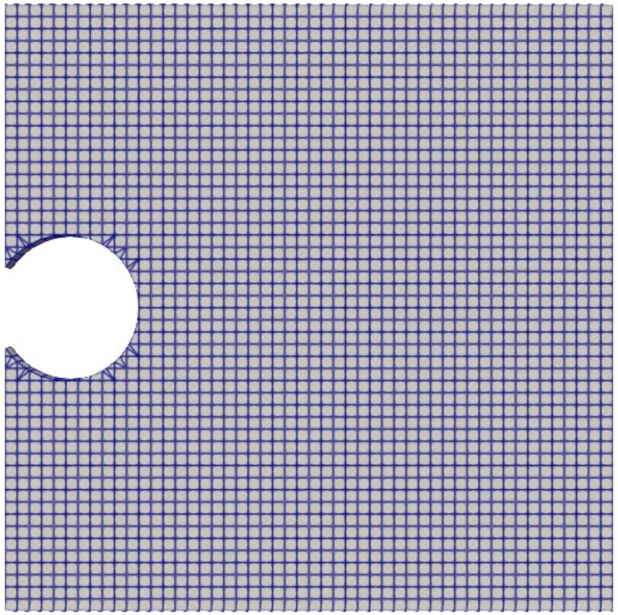
SDF-Trimmed Mesh: advantages

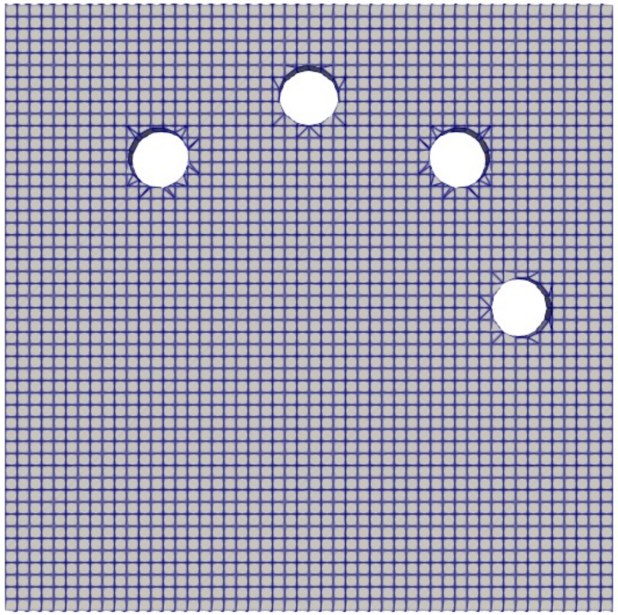
- ✓ **Real-time**
- ✓ **Conformal unstructured**



SDF-Trimmed Mesh: advantages

✓ **Handle motion easily**



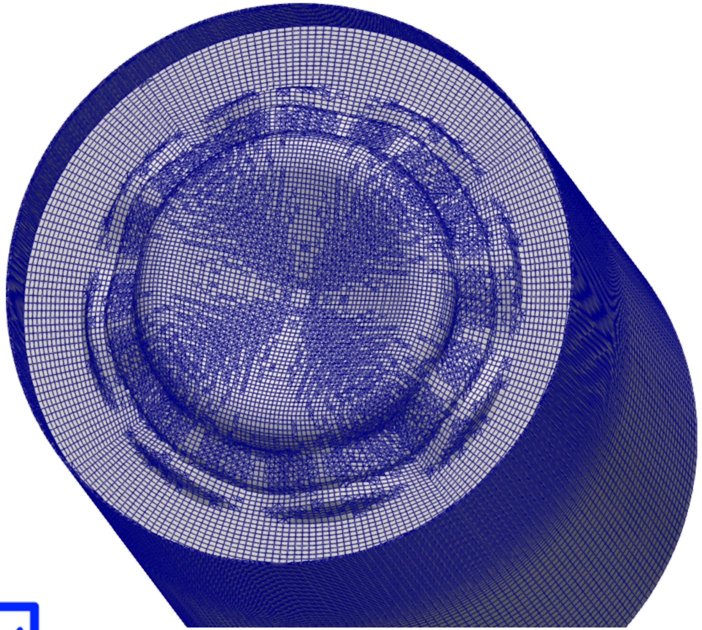


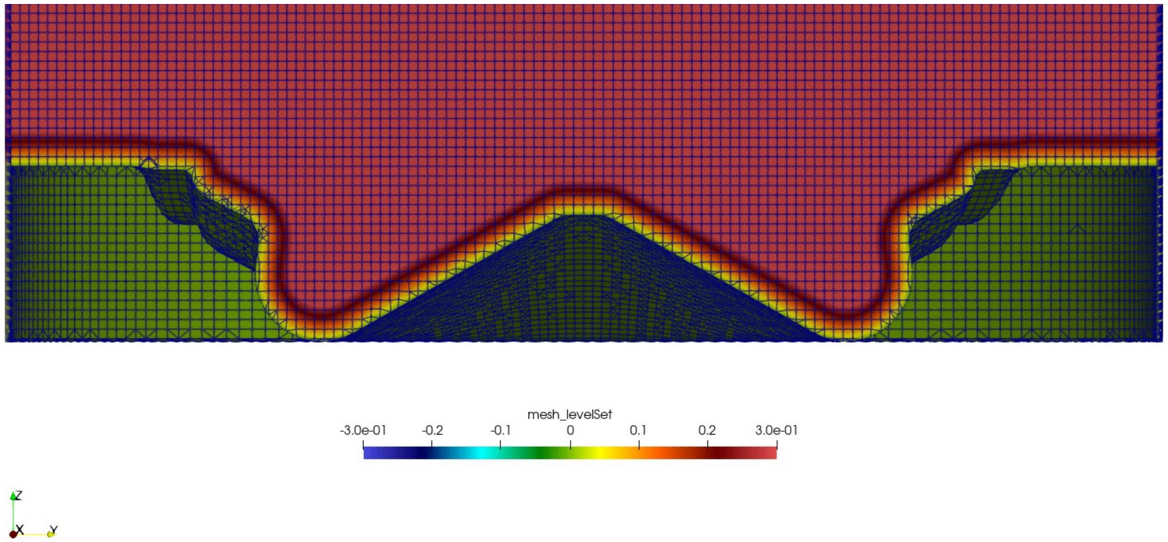
SDF-Trimmed Mesh: advantages

✓ **Smooth surface optimization / piston design**

✓ **Parameterized surface extraction**

✗ **But…**

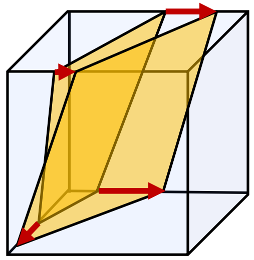




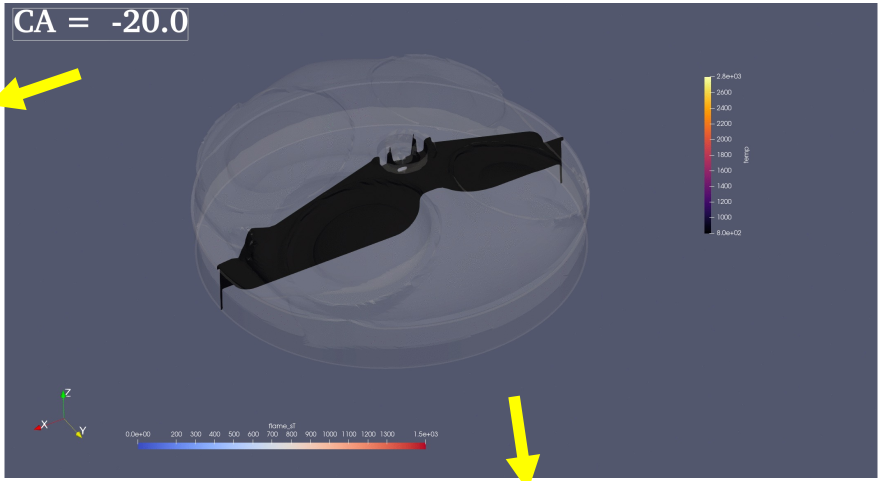
SDF-Trimmed Mesh for SI combustion

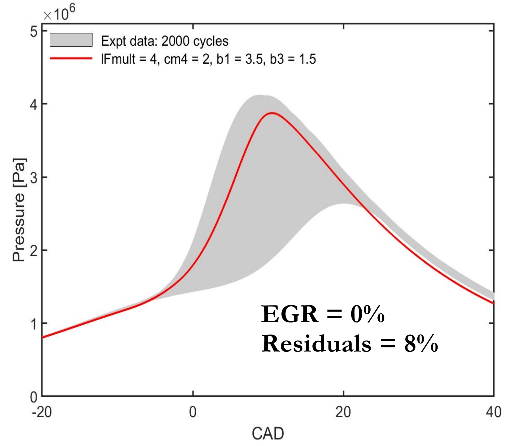
- ORNL LPG platform

Mesh-insensitive G-Equation source terms:

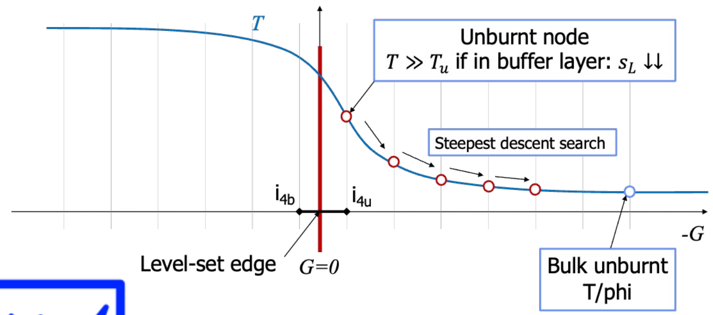


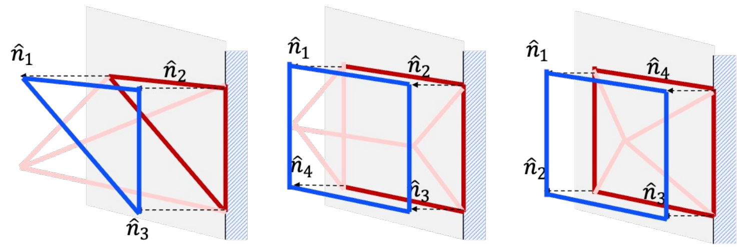
$$\dot{\omega}\Delta V$$

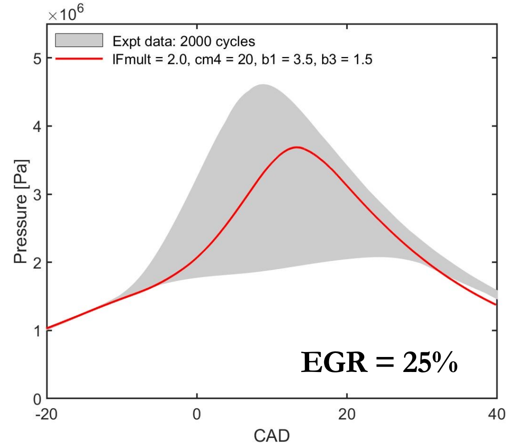




"Virtual Prismatic Layer": less grid-sensitive law of the wall source terms



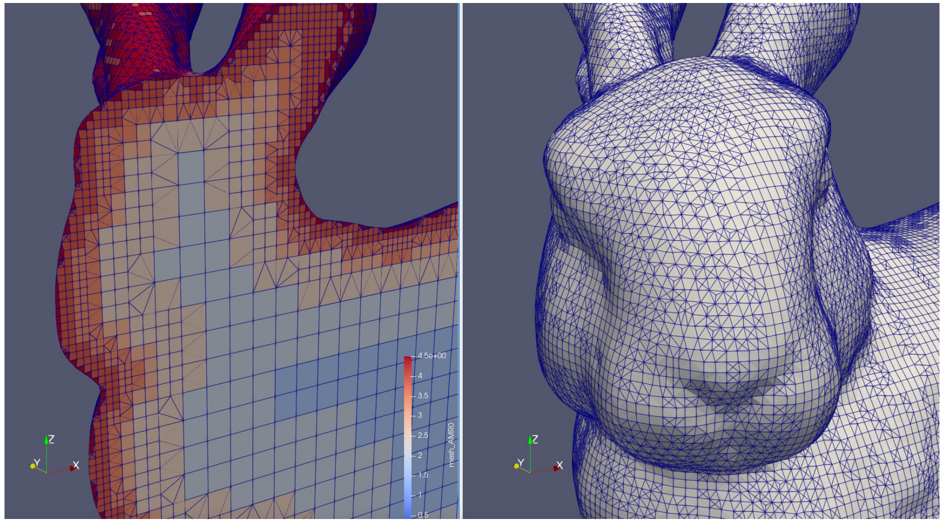




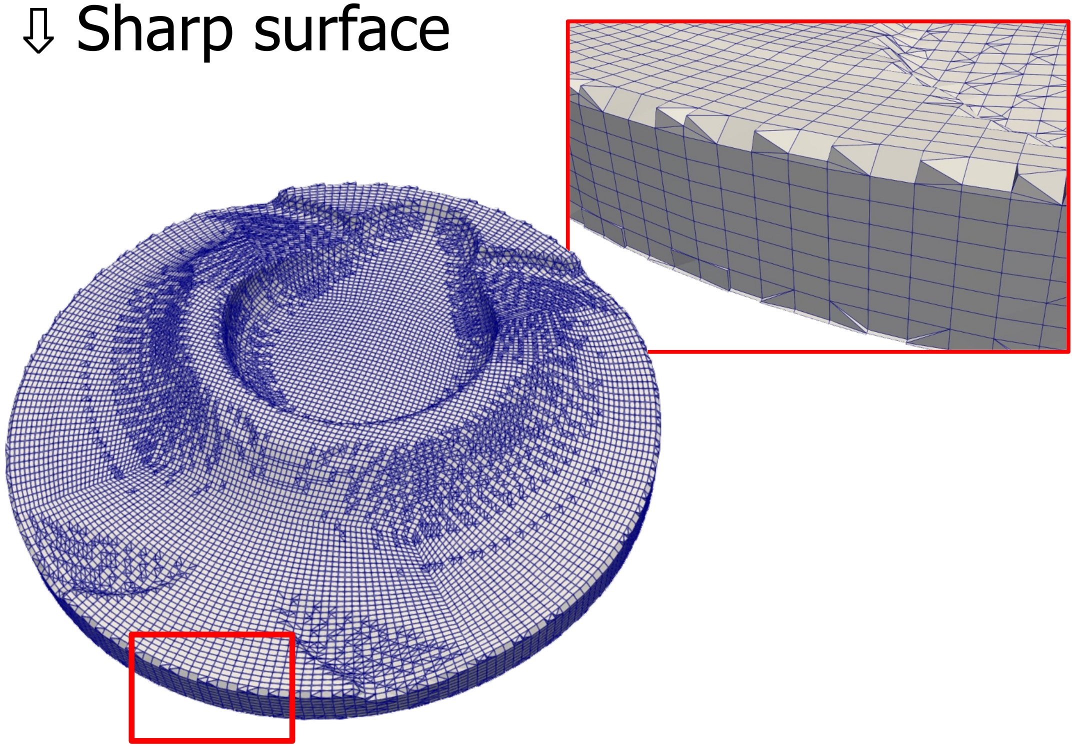
SDF-Trimmed Mesh: ....issues

- Sharp features are not well resolved
- Unless insanely large numbers of cells are used

⇩ Smooth surface

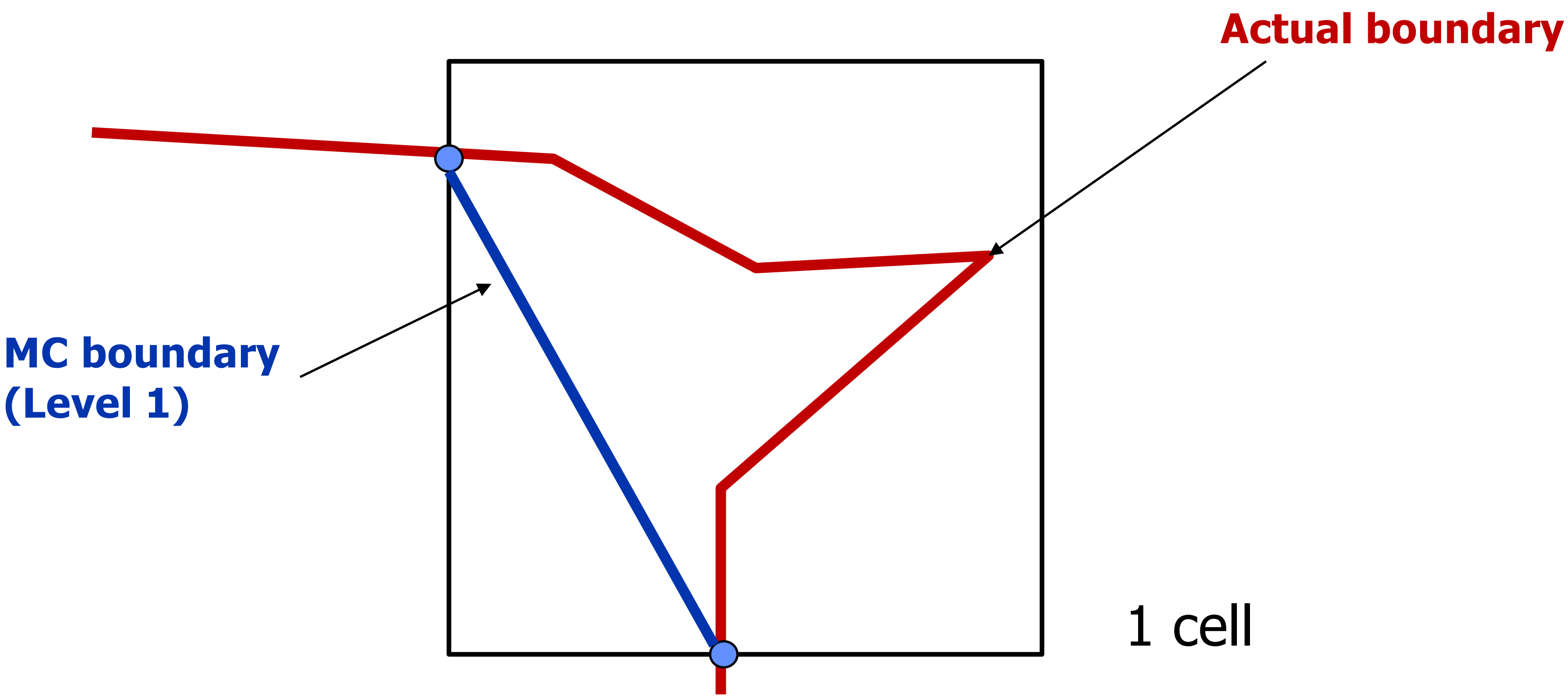


⇩ Sharp surface



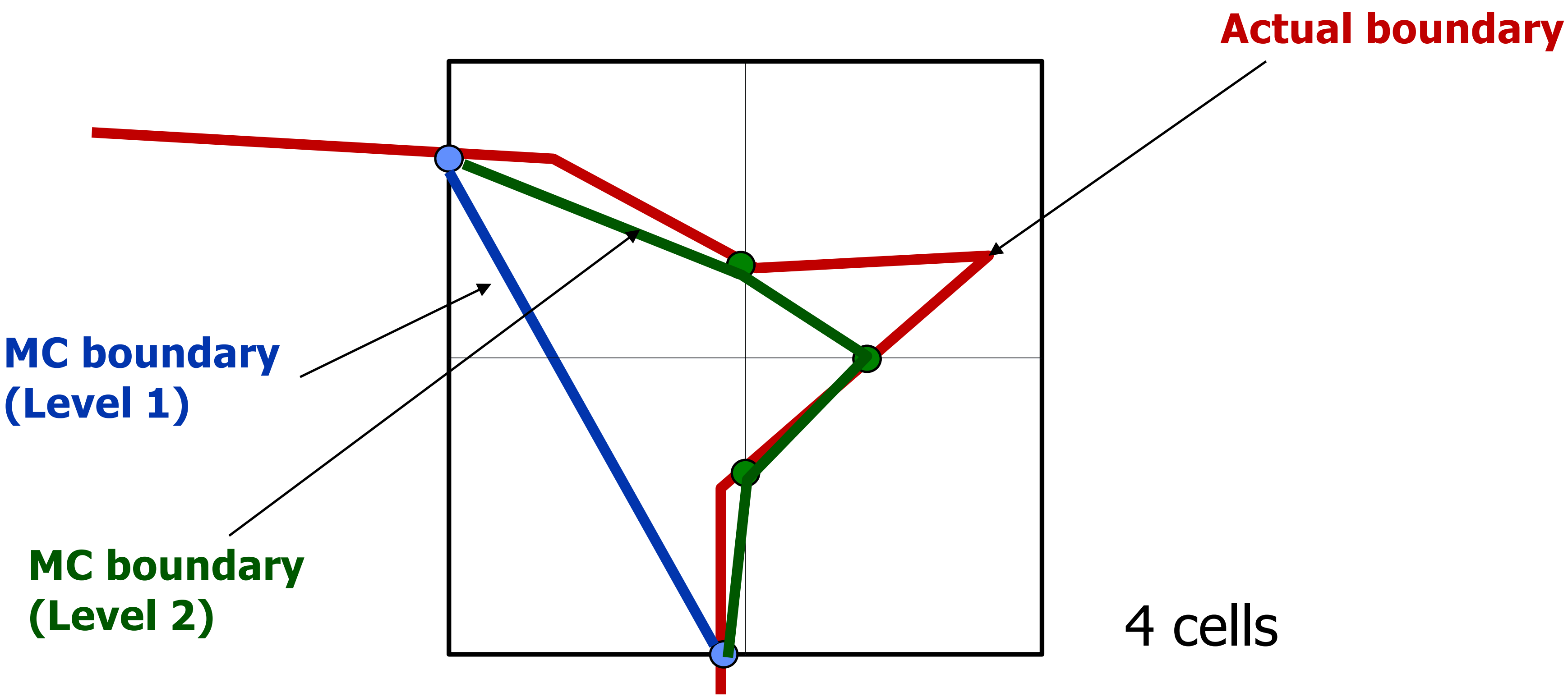
SDF-Trimmed Mesh: issues

- MC uses a <u>linear</u> field approximation inside the cell
- Impossible to capture sub-grid features without additional refinement



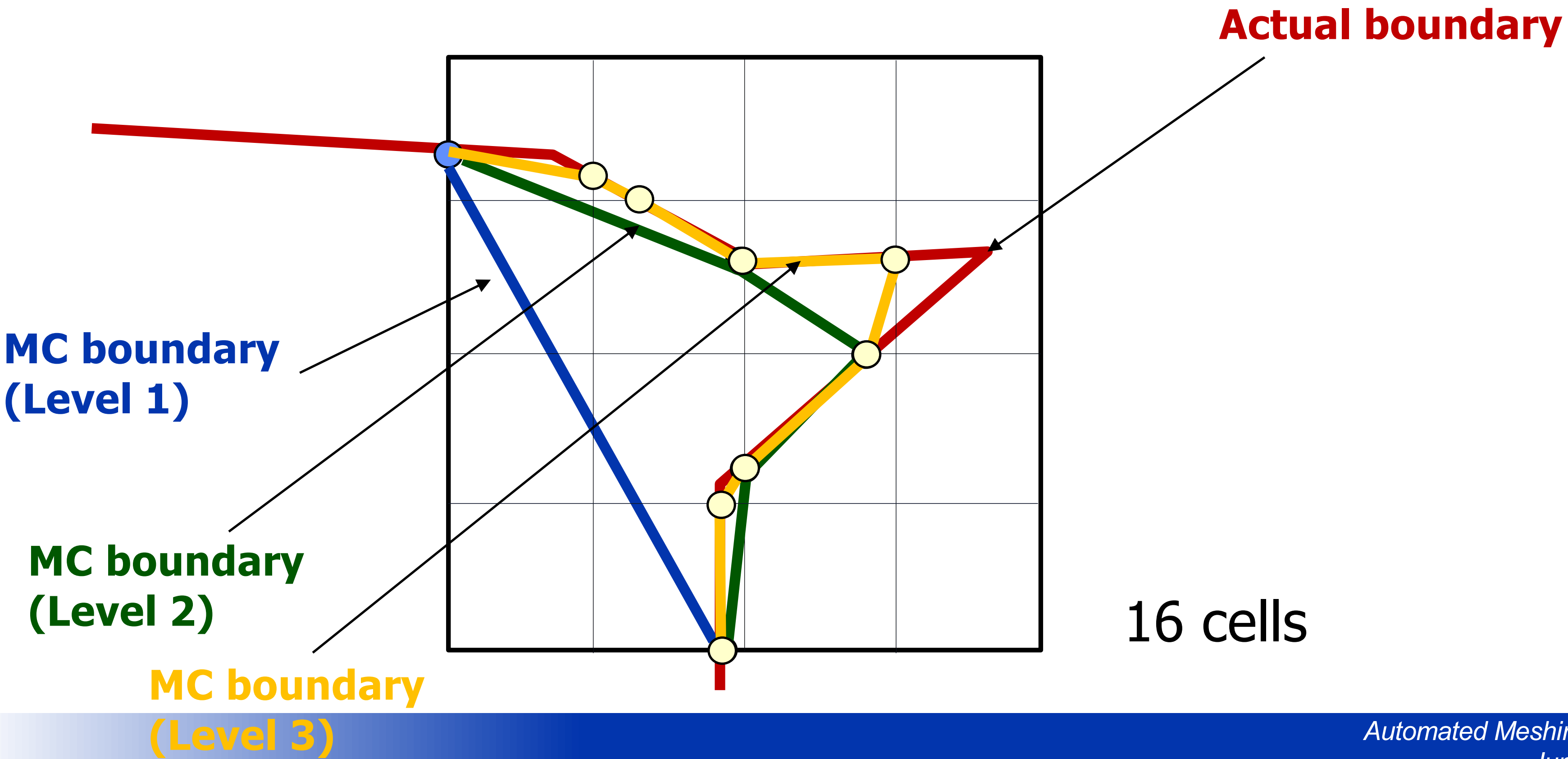
SDF-Trimmed Mesh: issues

- MC uses a <u>linear</u> field approximation inside the cell
- Impossible to capture sub-grid features without additional refinement



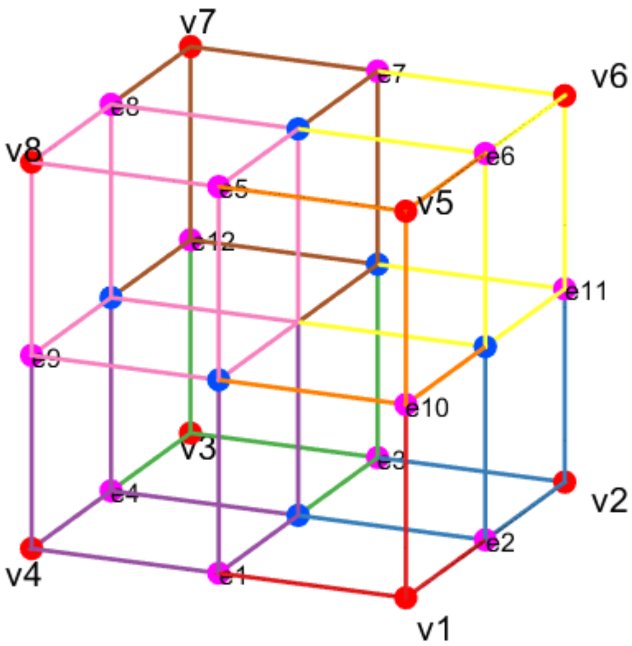
SDF-Trimmed Mesh: issues

- MC uses a <u>linear</u> field approximation inside the cell
- Impossible to capture sub-grid features without additional refinement



Split table approach for AMR

- **Marching Cubes but for AMR**



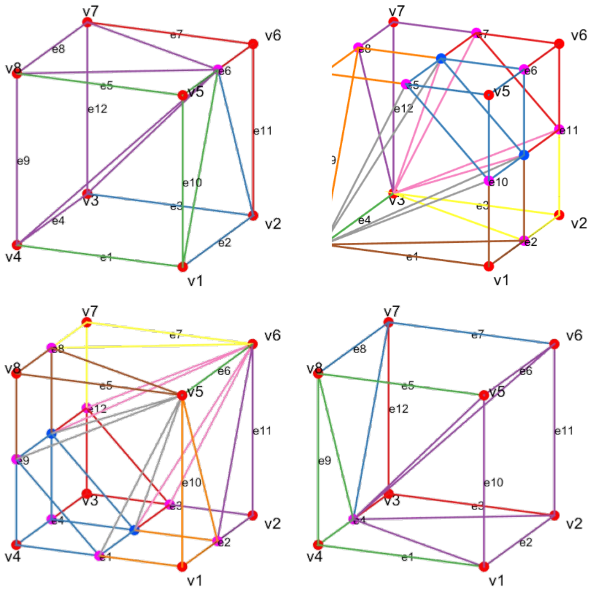
- Now split:
  - 12 edges
  - 6 faces
  - 1 cell center

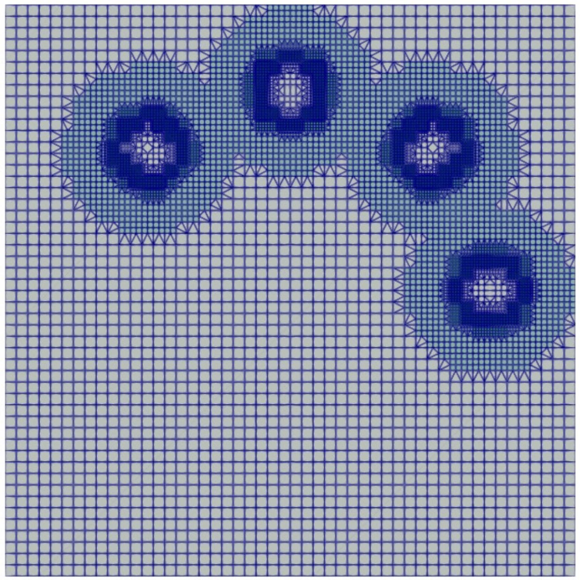
- LUT configurations:
  - 2^{18} = 262,143
  - Hexa only

Jaillet, Lobos 2021

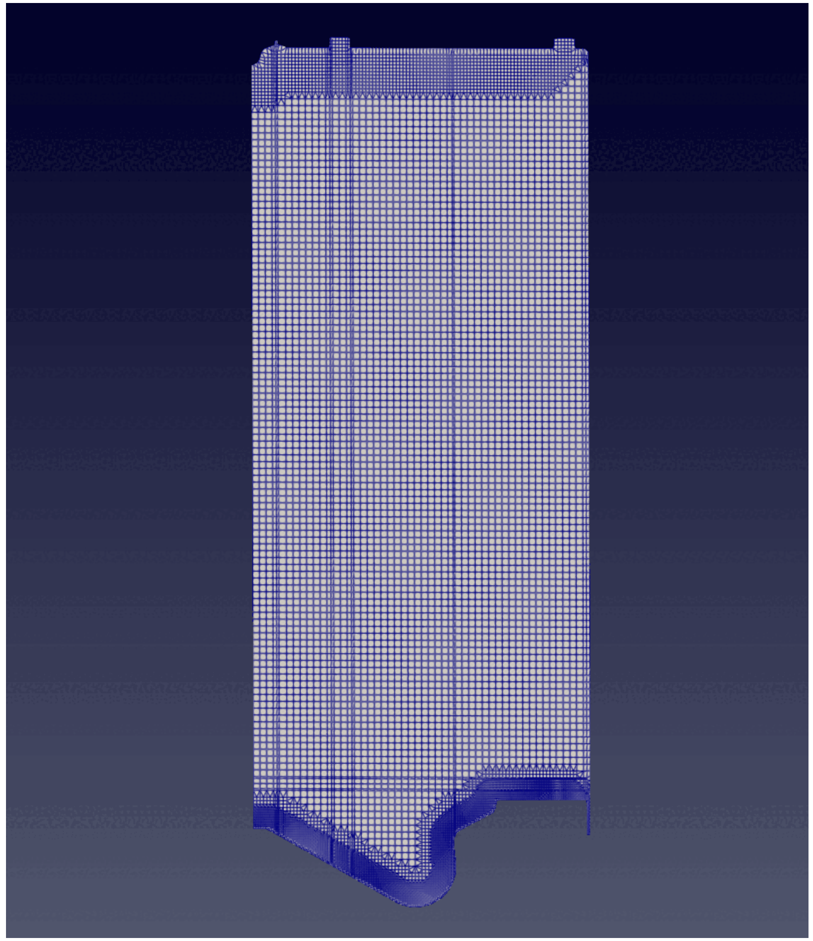
SDF- or surface-based refinement

- A conformal, unstructured mesh at all times

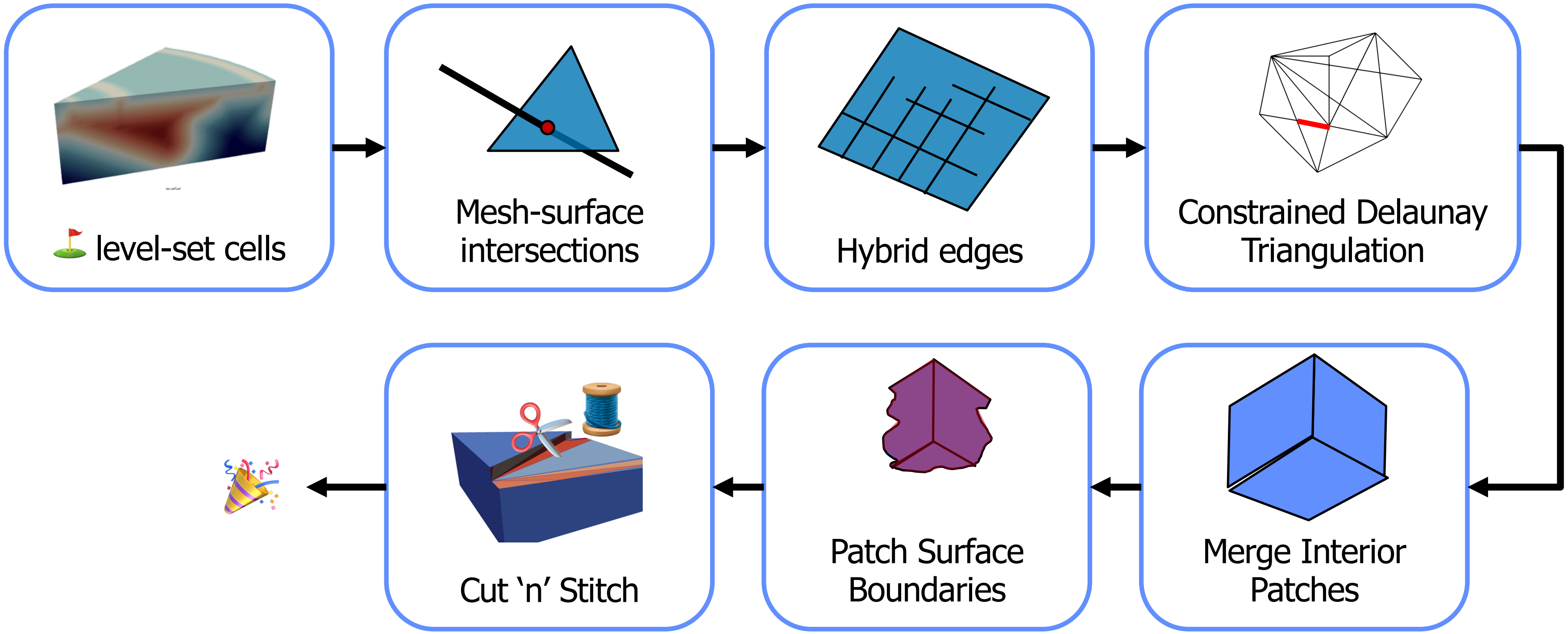




- Sinusoidal AMR region sizing

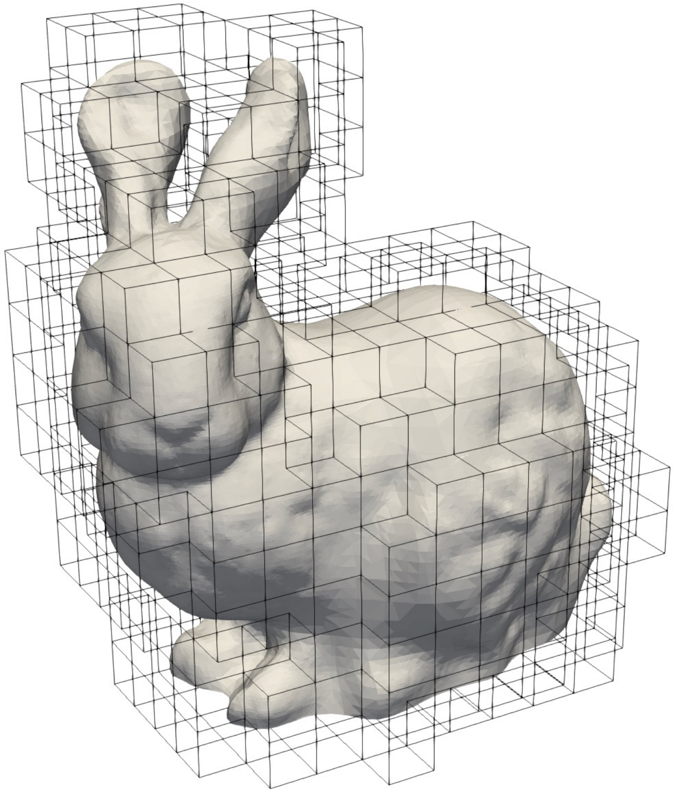


Cut-cell generation

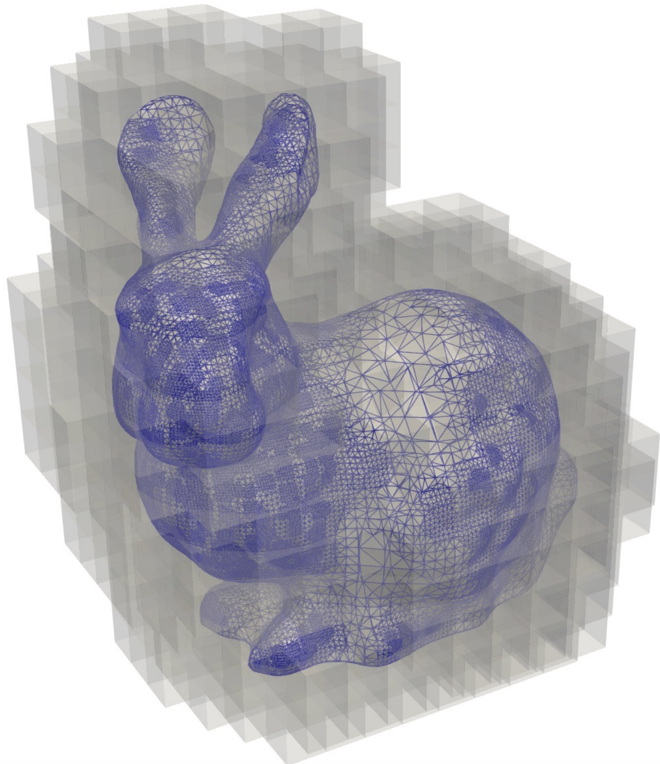


Hashed Mesh-Surface Intersections

Mesh edges → BC faces

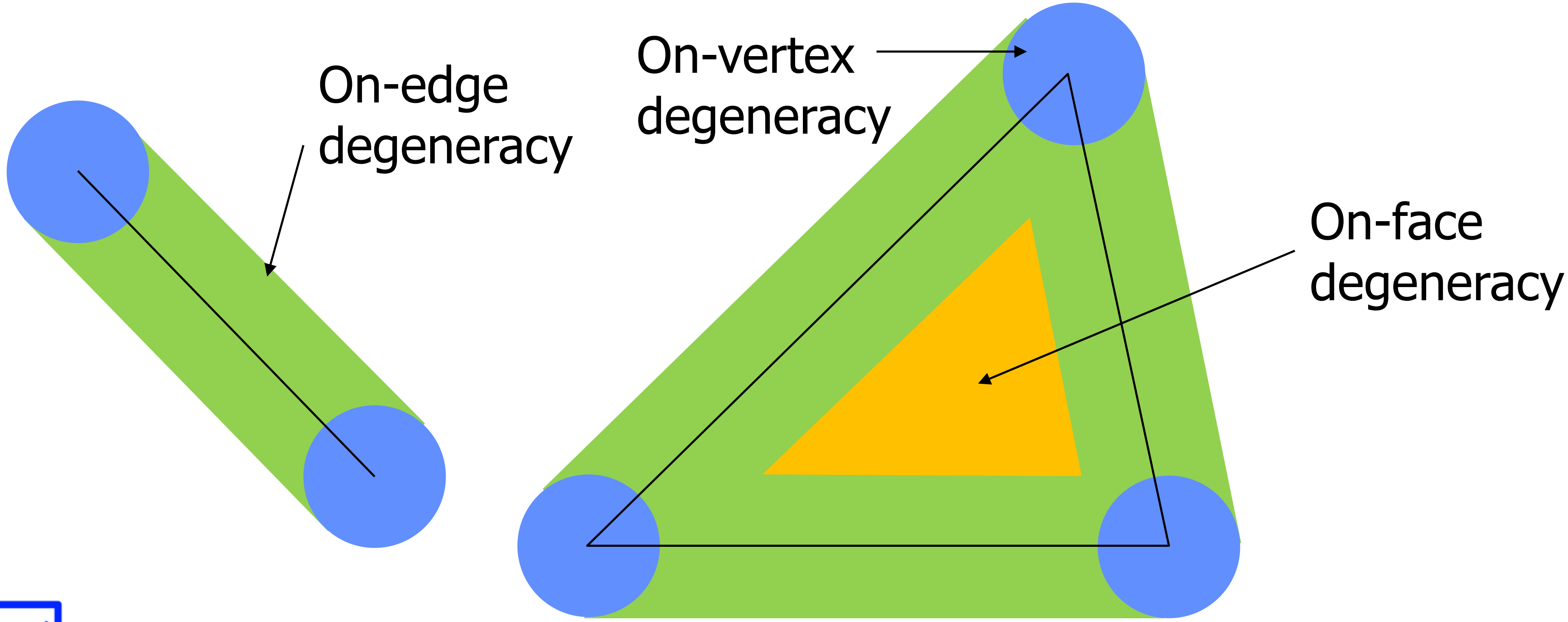


BC edges → Mesh faces

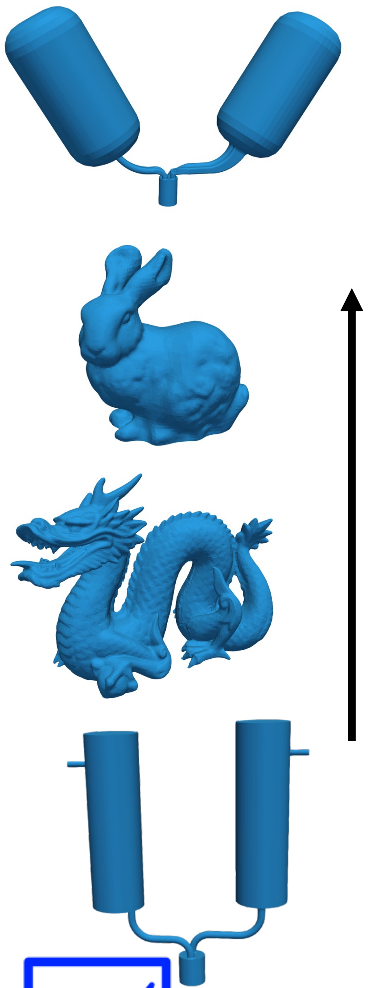


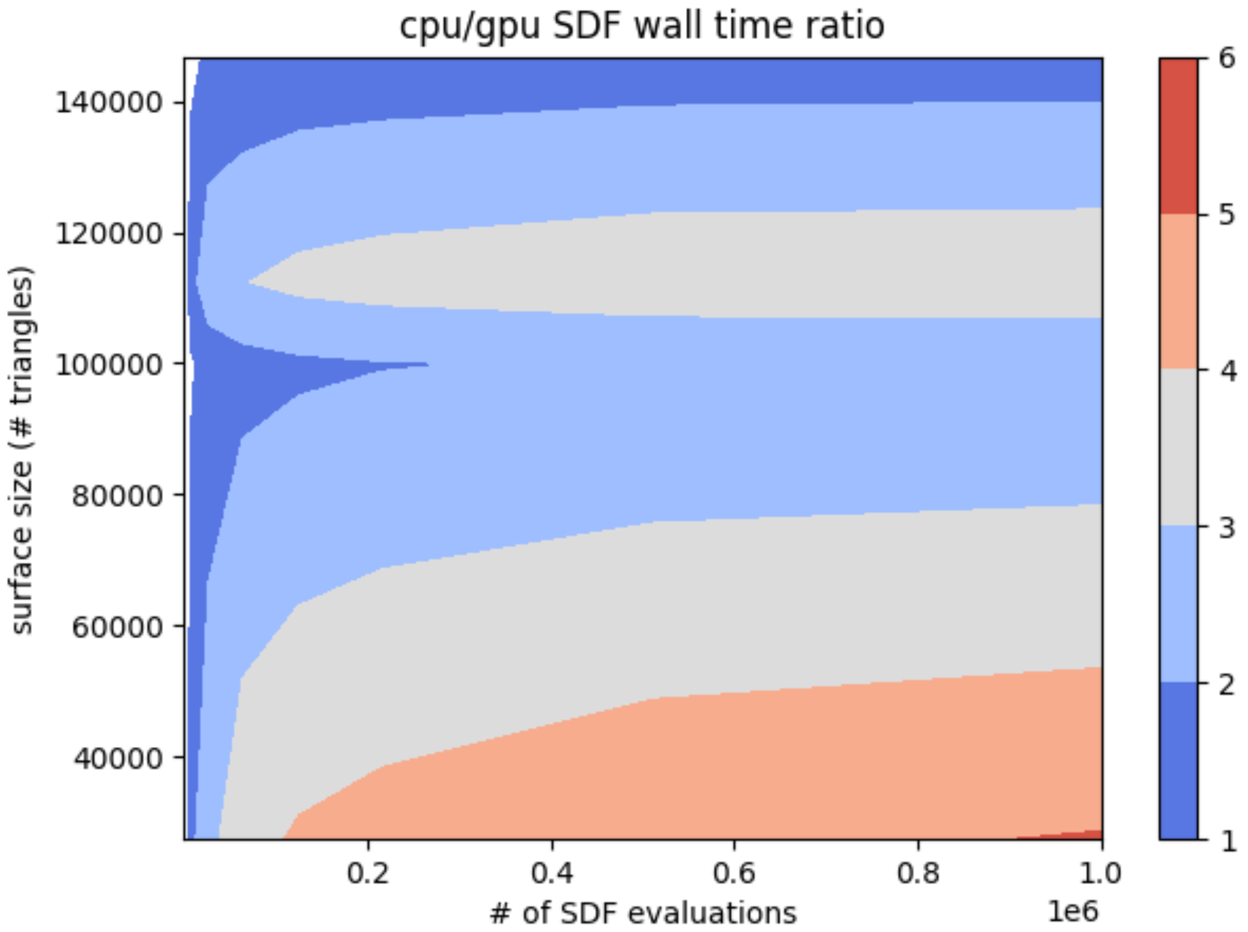
Floating point intersections

- **Cartesian mesh → Easy: Use Integer Arithmetic and x, y, z planes**
- **General mesh → Hard problem! fp math is not exact → <u>Tolerances</u>***



GPUs are useful, but not perfect





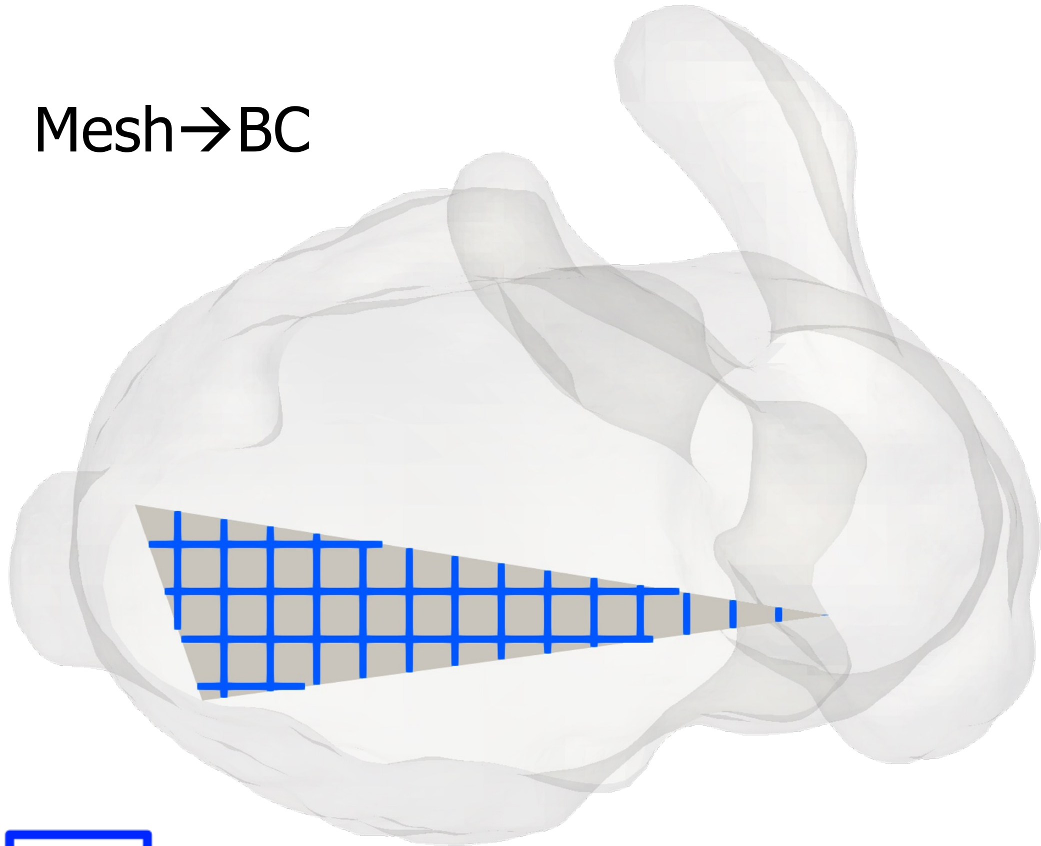
✓ Noticeable speedup

✓ Dependency on triangle tree footprint

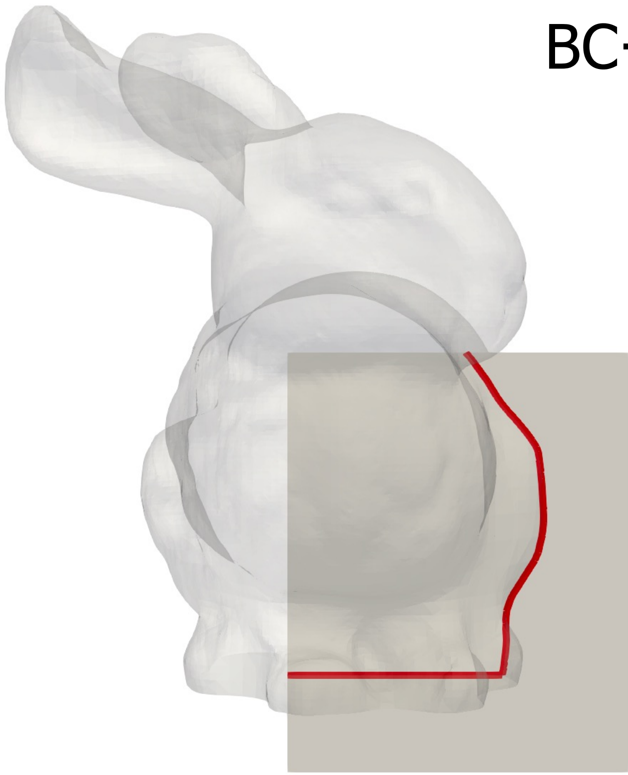
Generate hybrid edges

- Intersection points from the same face generate constraint edges…

Mesh→BC



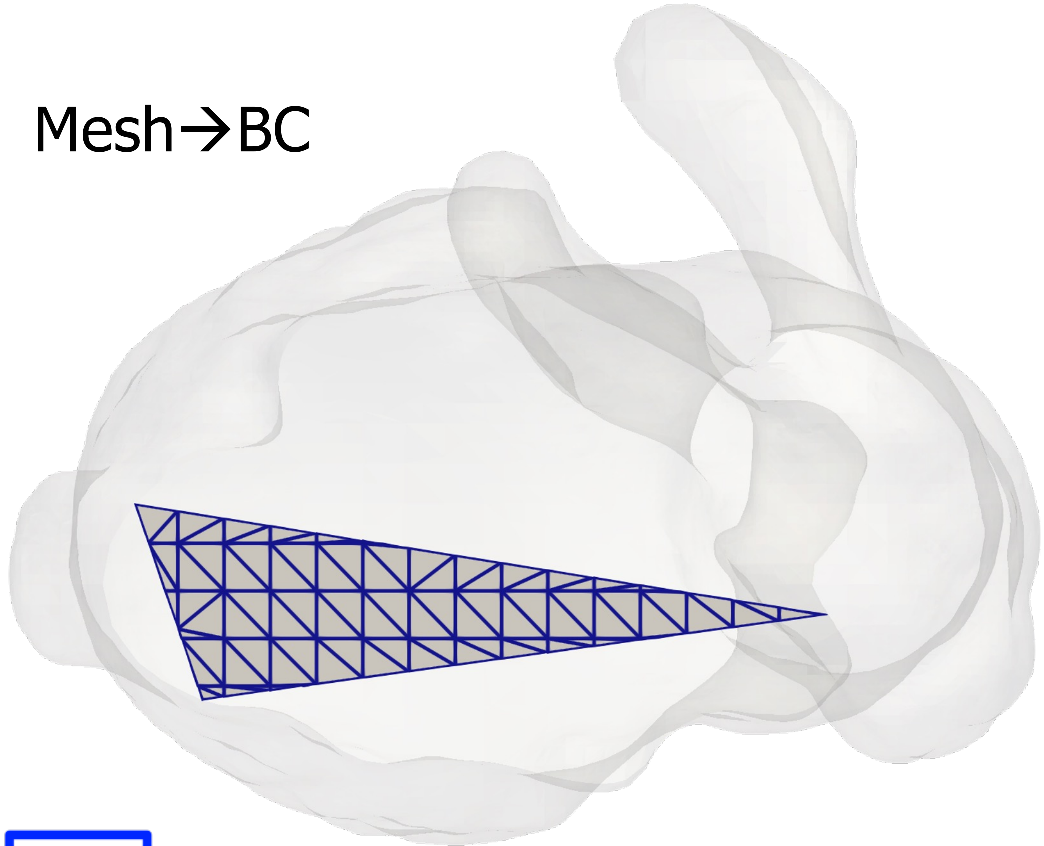
BC→Mesh



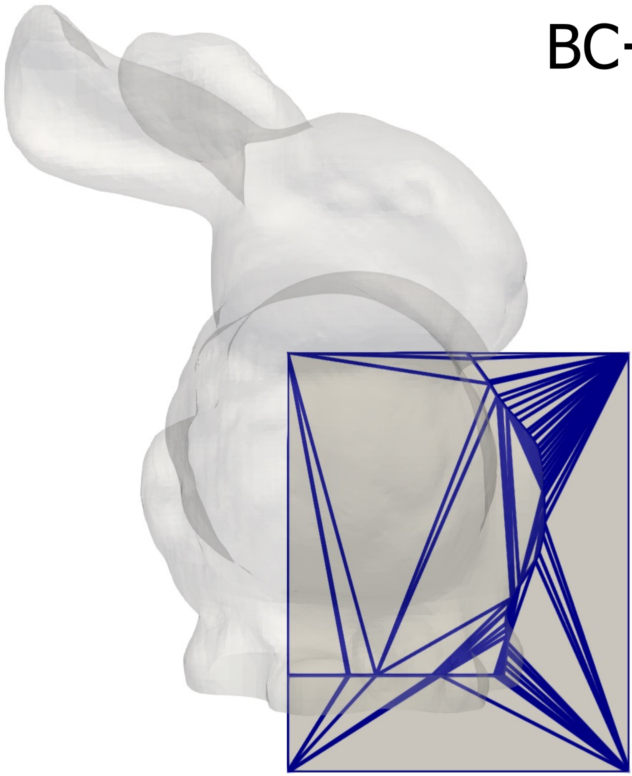
Constrained Delaunay Triangulation

- …that are used to triangulate the face

Mesh→BC

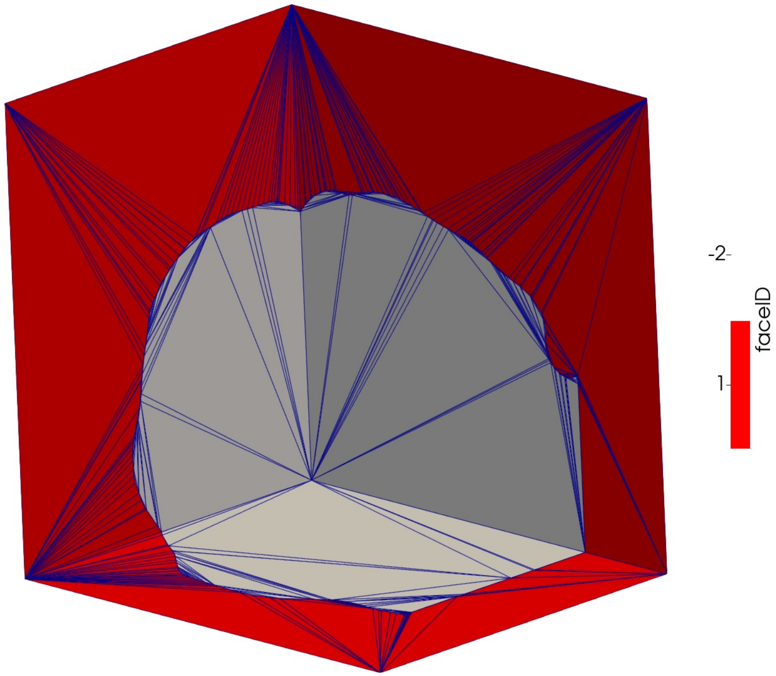


BC→Mesh



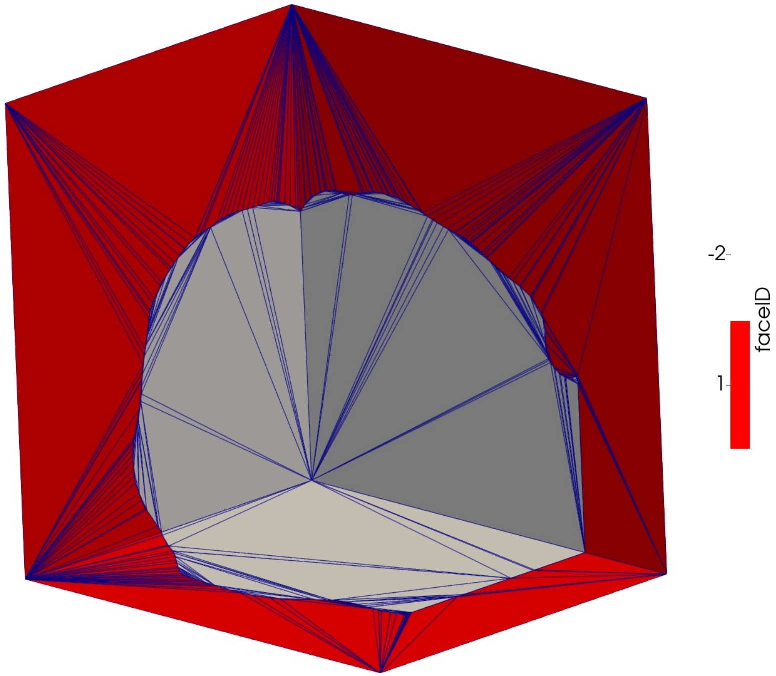
Merge Interior Patches

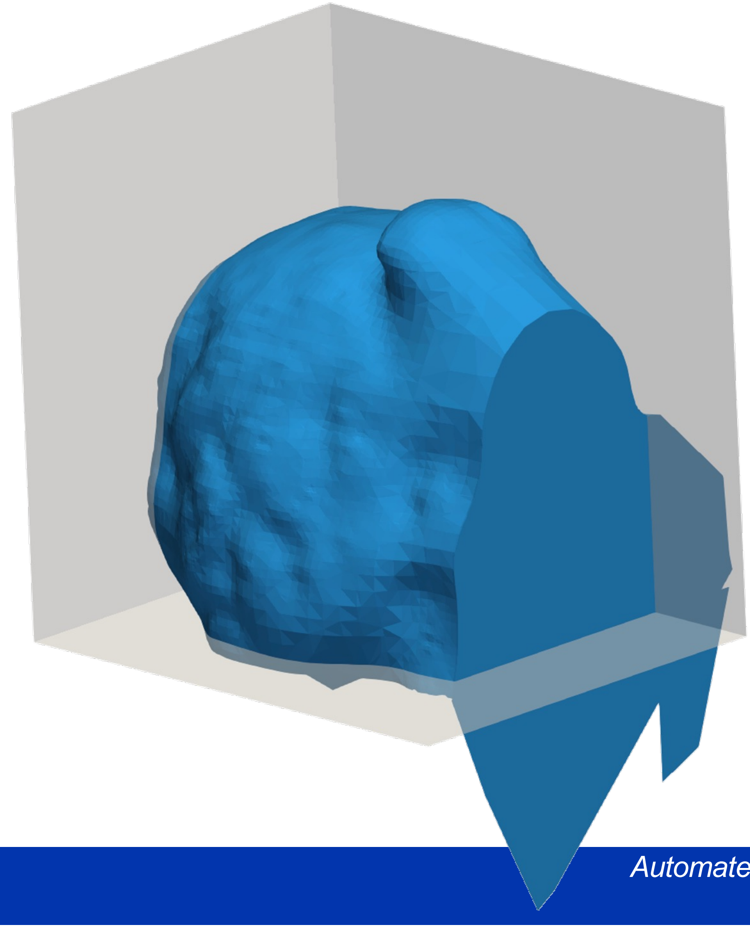
- For each cut-cell, collect face CDTs
- Flag Mesh-Inside-Surface: Connectivity? Or SDF



Patch Surface Boundary

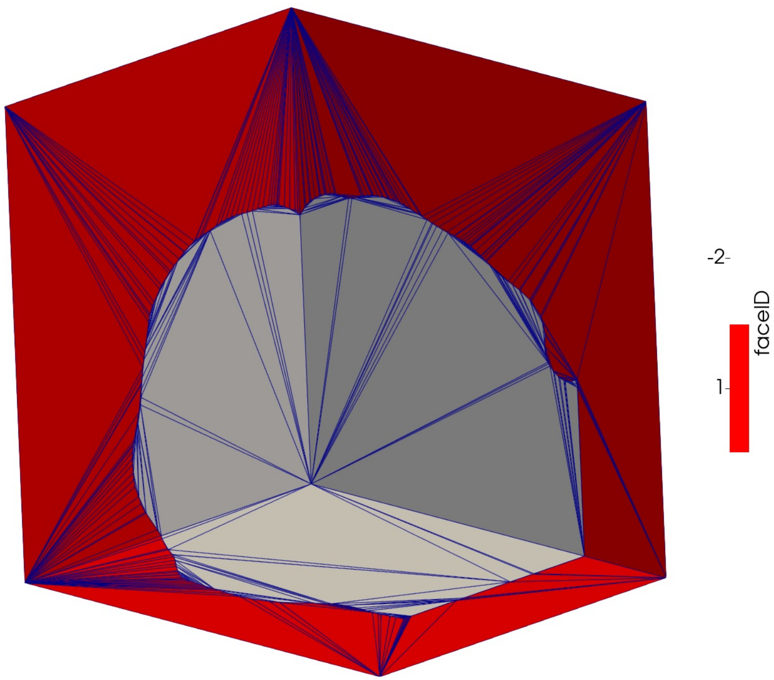
- **Extract BC mesh subset**

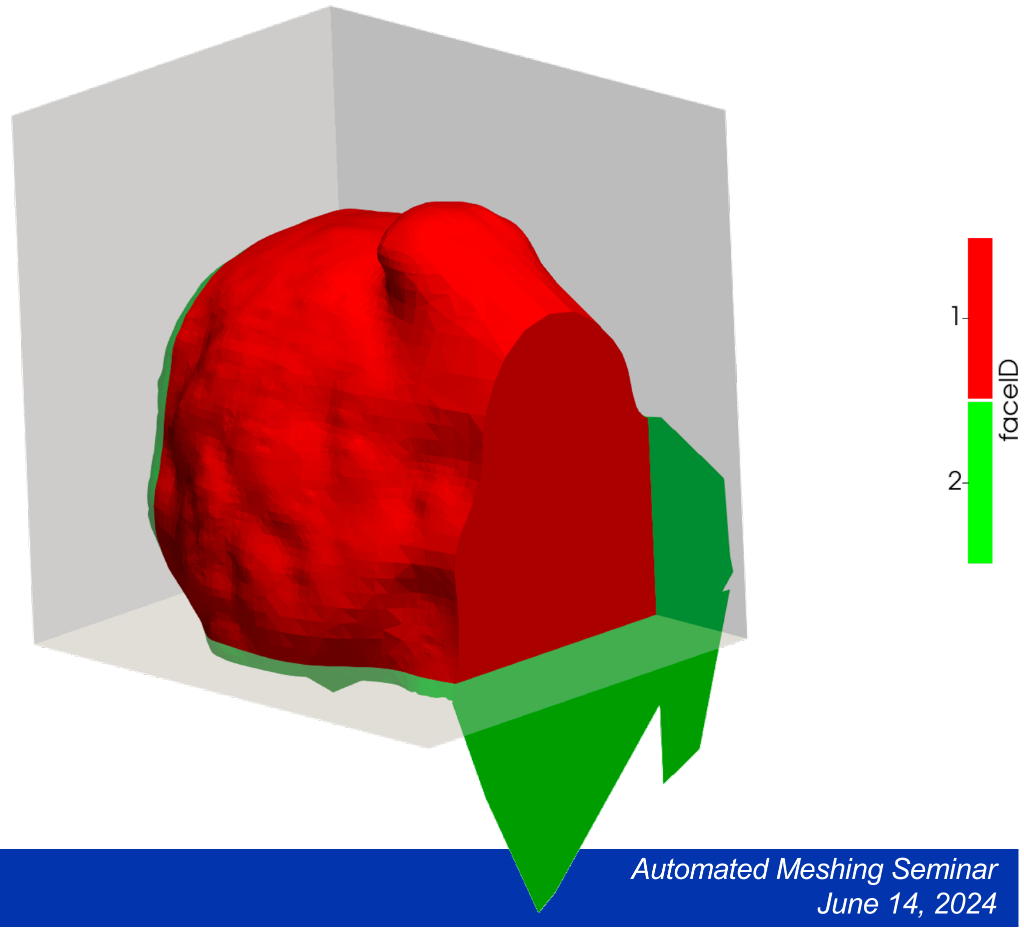




Patch Surface Boundary

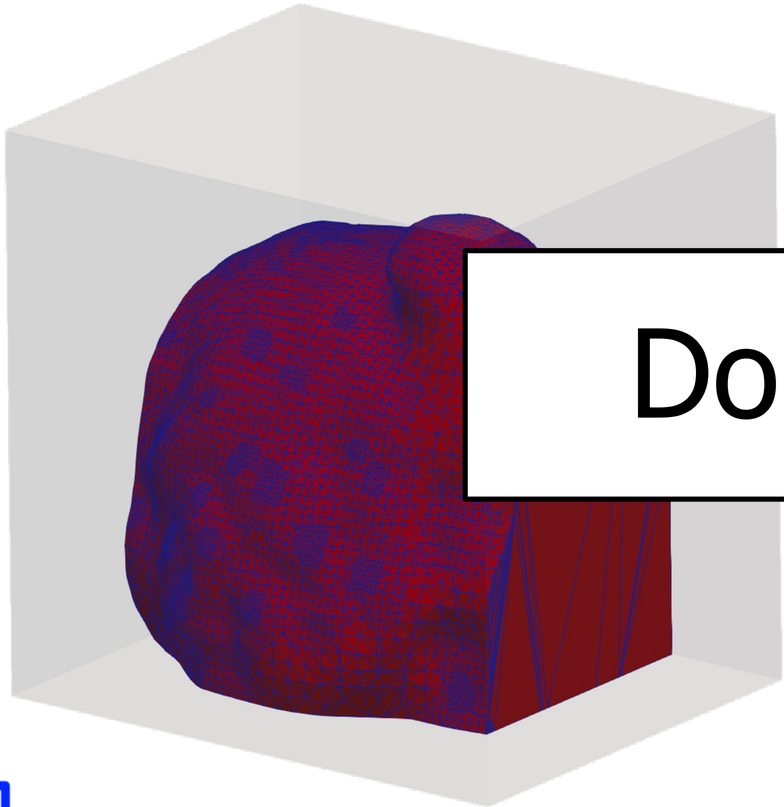
- **Flag Surface-Inside-Cell: Connectivity? Or SDF**



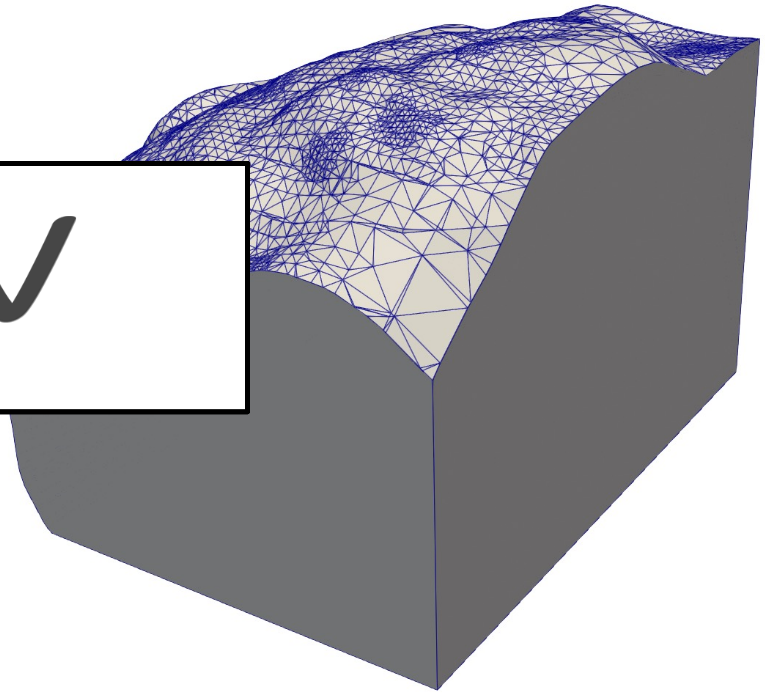


Cut-cell assembly

- **Assemble faces** **Merge face triangles → Polygons**



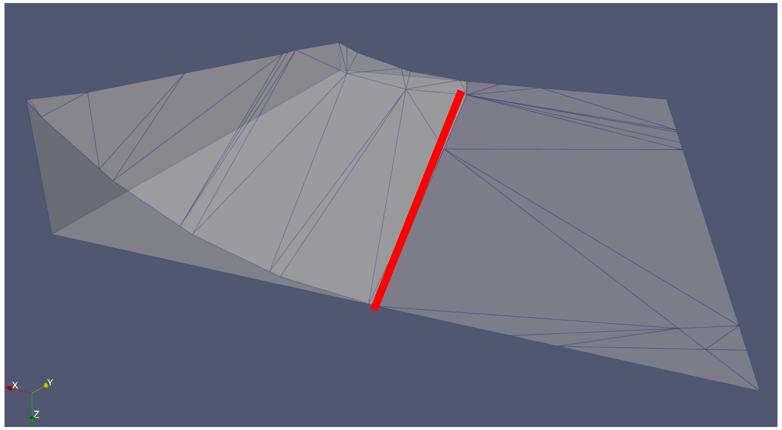
Done ✓



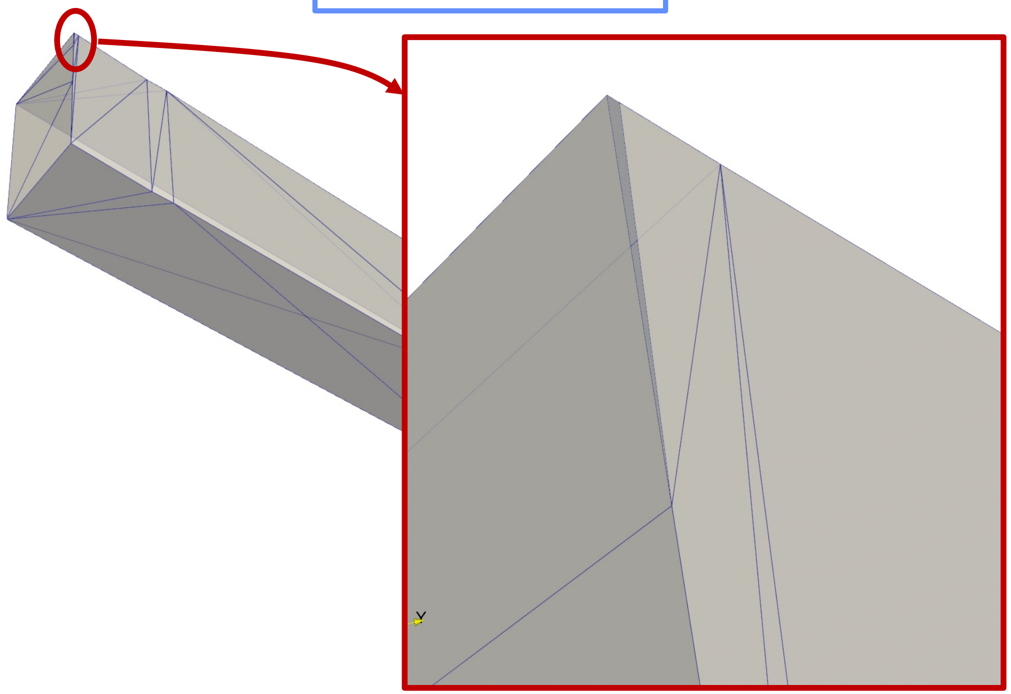
Cut 'n' stitch: what can possibly go wrong?

- Surface is manifold if no free boundaries
- 2 types of degeneracies through edge connectivity

3+ faces: non-manifold triangles

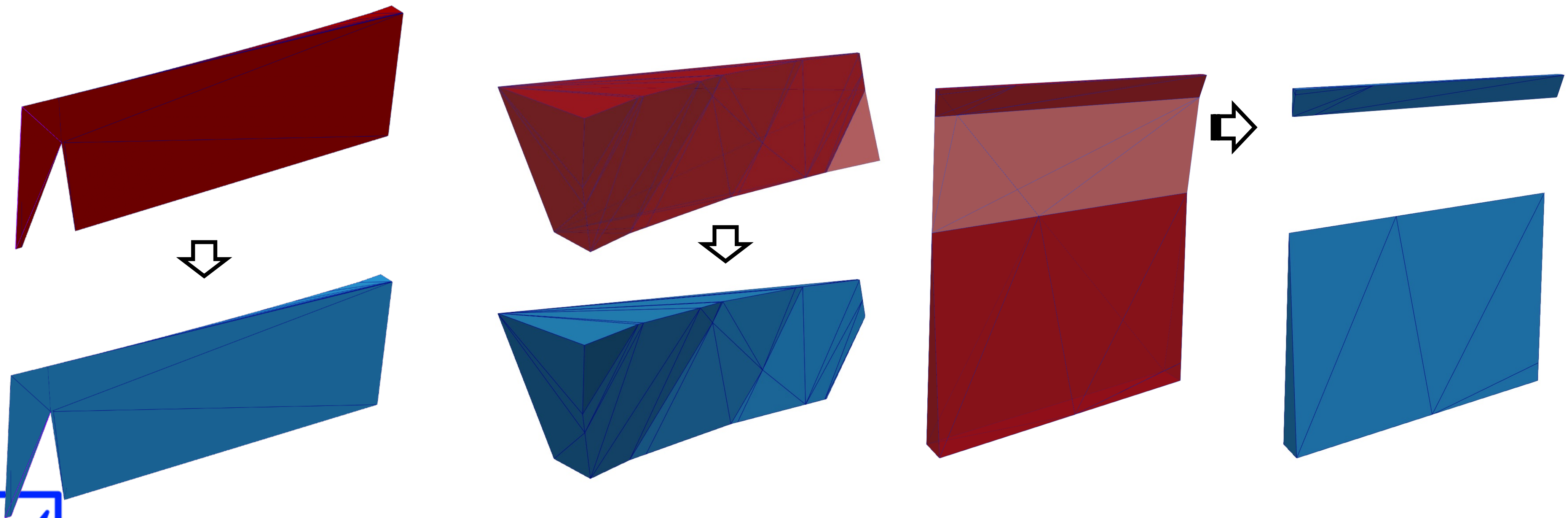


1 face: holes



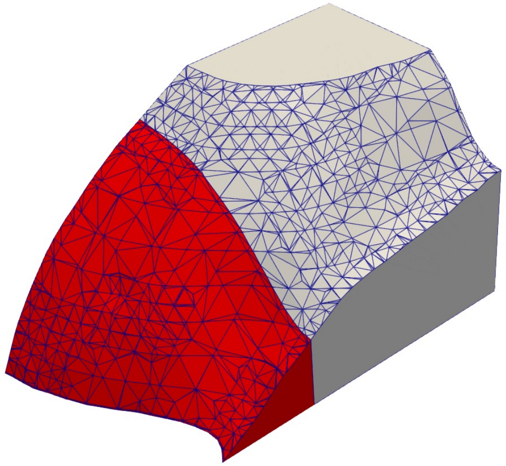
Cut 'n' stitch: examples

- "Cut" – duplicate all non-manifold edges → create holes
- "Stitch" – Merge them back, then remove leftovers
- "Fill Holes" – close remaining open boundaries



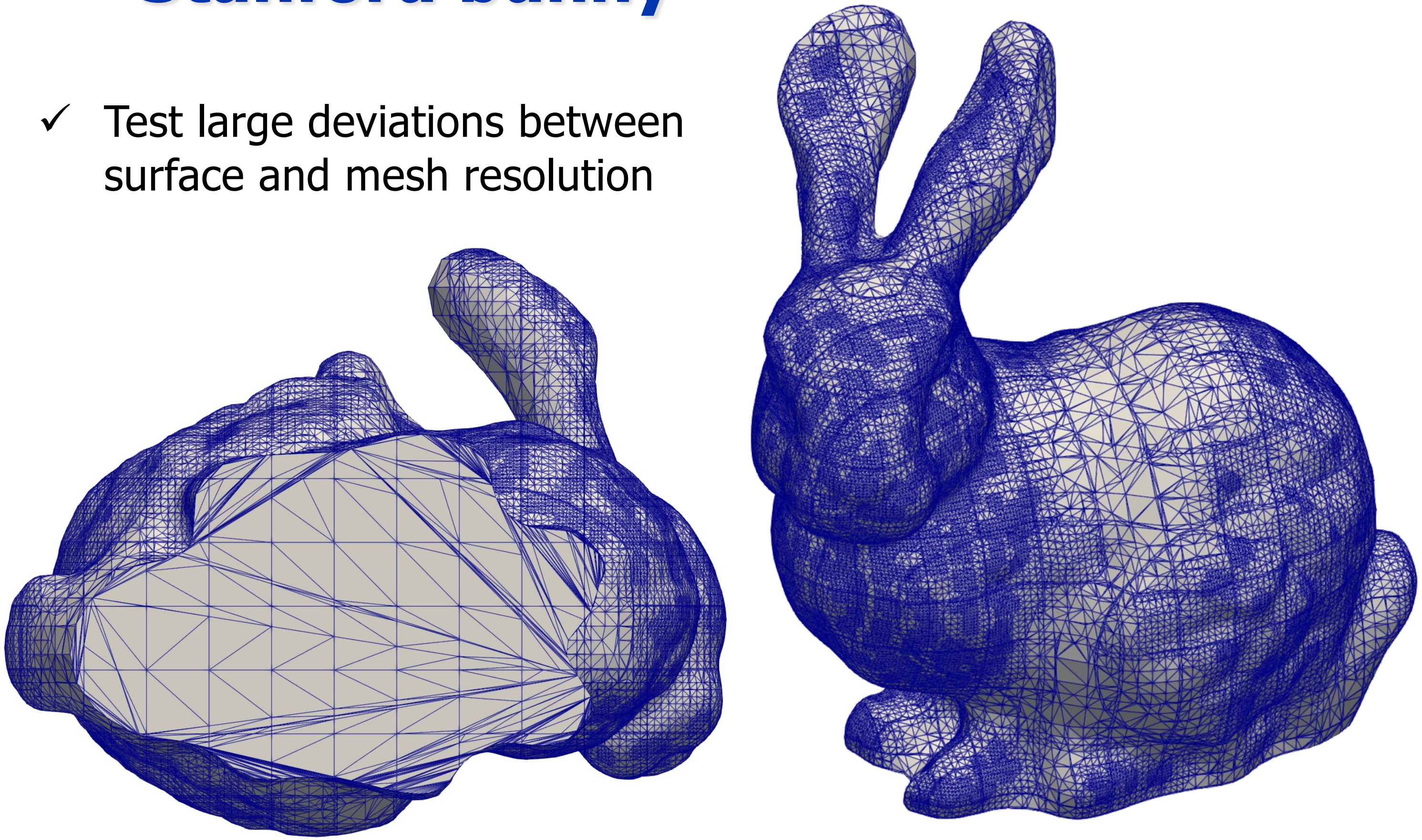
Stanford bunny

Domain	Stanford bunny
# cells	112,400
# tri	100,000
Δx	5.0mm



- 2 cells
- 922 points

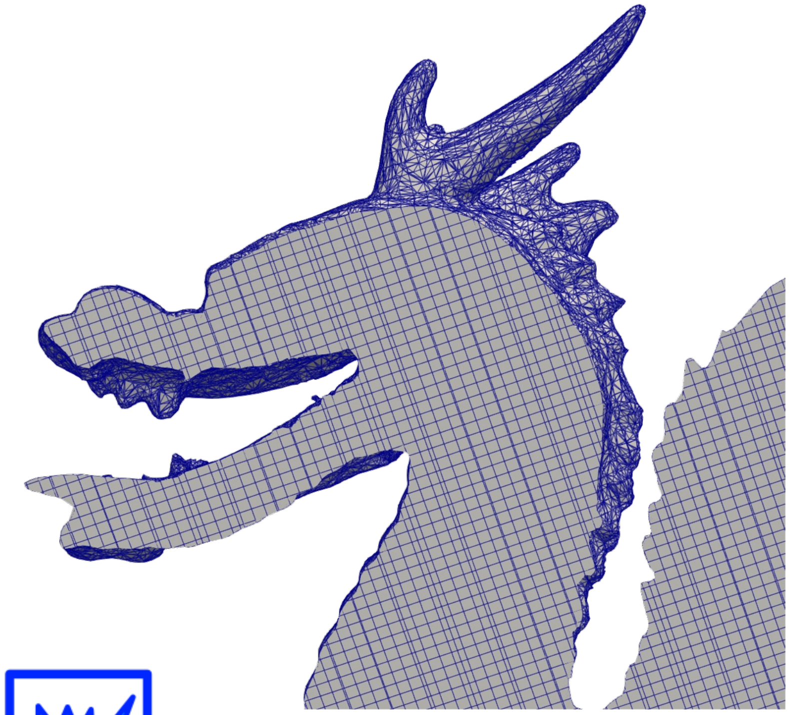
✓ Test large deviations between surface and mesh resolution

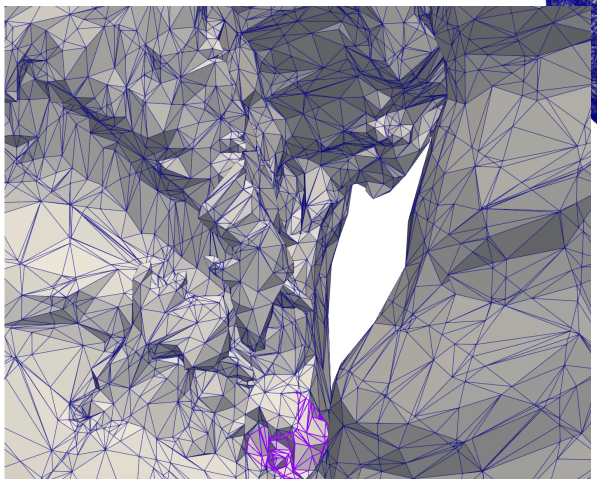


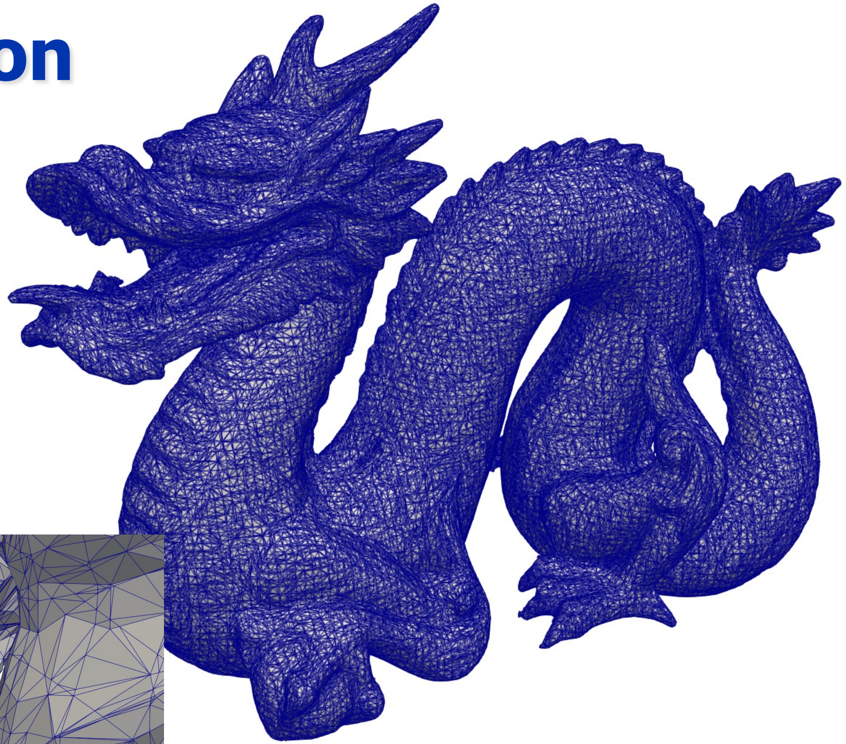
Stanford Dragon

Domain	Stanford Dragon
# cells	98,174
# tri	100,000
Δx	0.5mm

✓ Robust w.r.t. self-intersecting triangles



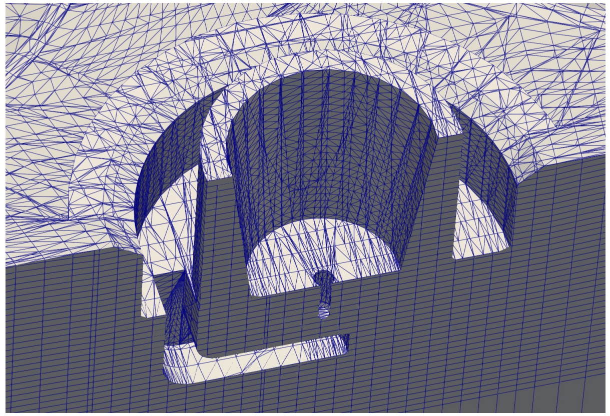


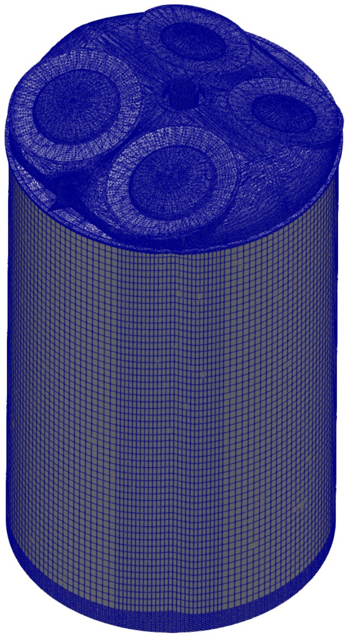


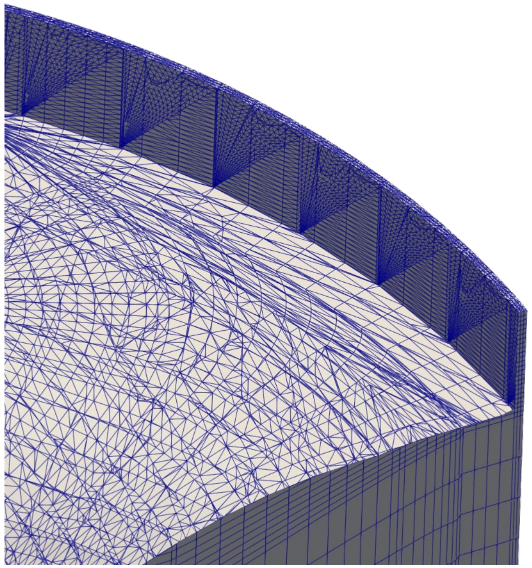
Domain	ORNL LNF
# cells	1,344,304
# tri	24,230
Δx	0.7mm

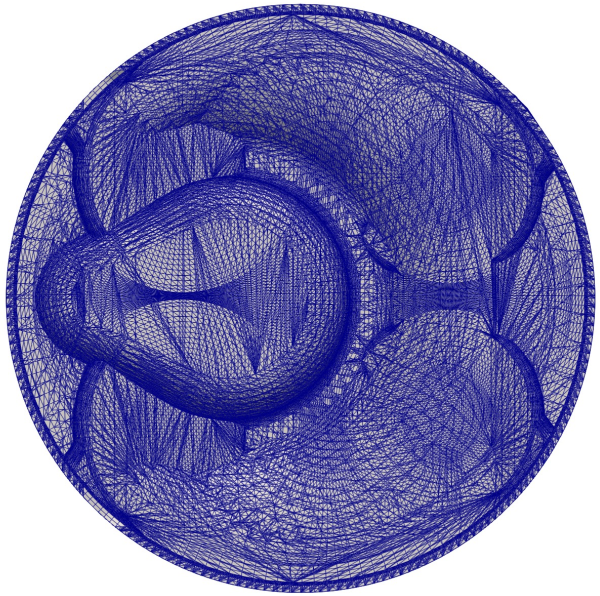
GDI cylinder

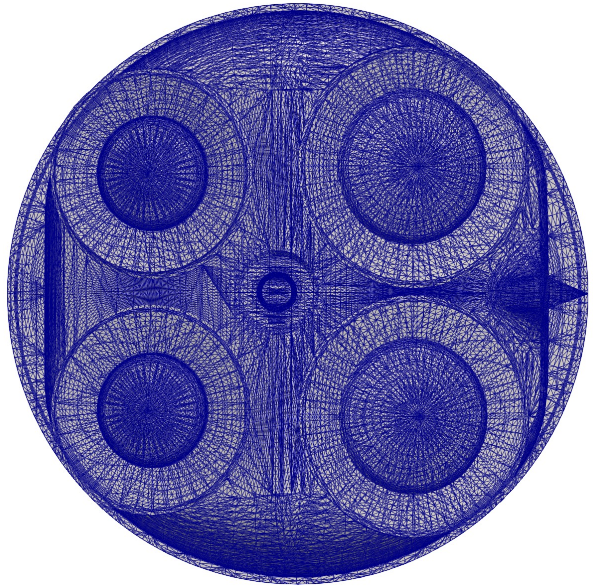
- ✓ Background O-grid mesh
- ✓ O-grid valve regions
- ✓ Head and crevice refinement







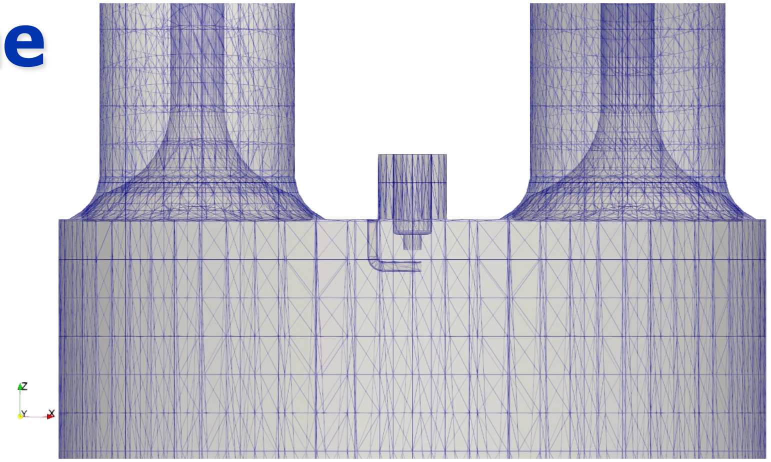


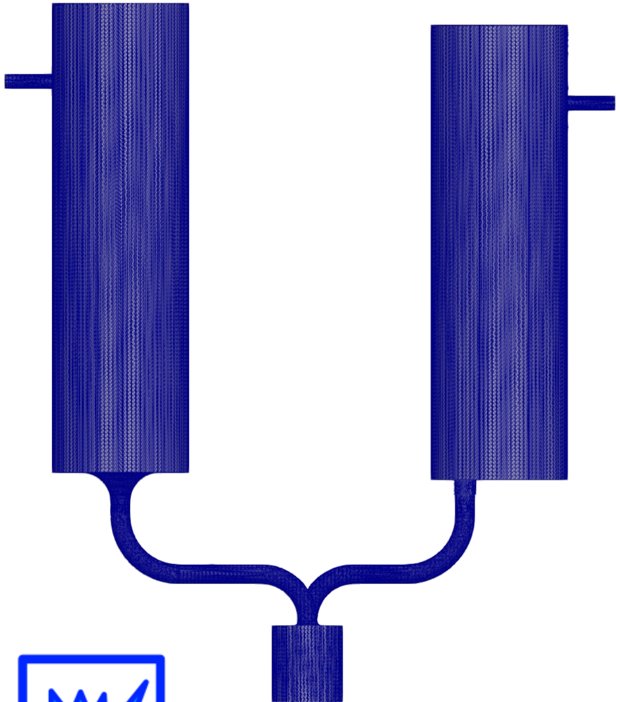


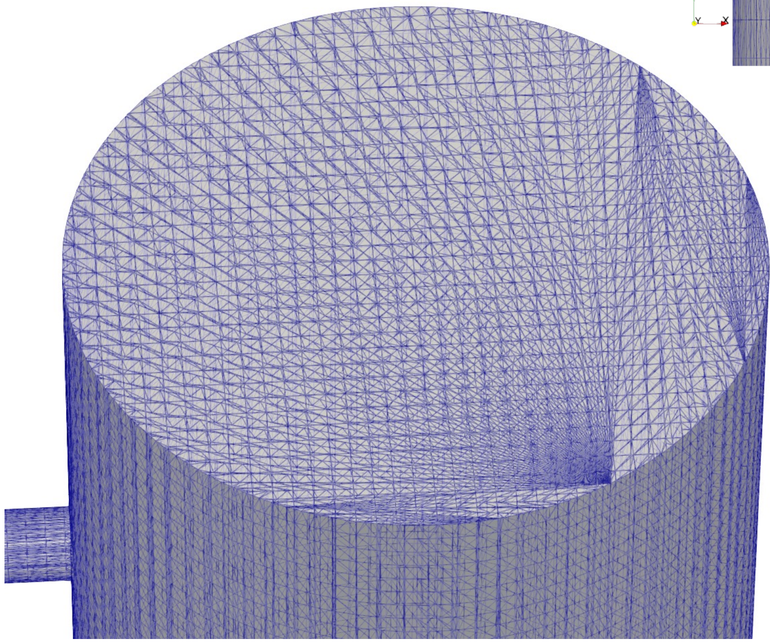
UM TCC Engine

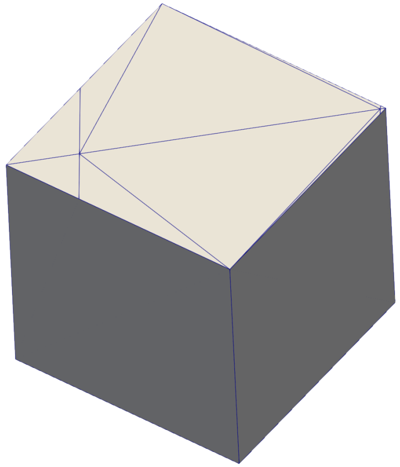
Domain	UM TCC
# cells	271,848
# tri	27,342
Δx	3.0mm

✓ Test for co-planar triangle intersections



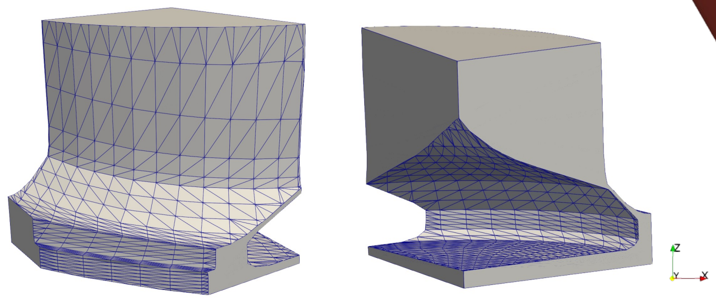


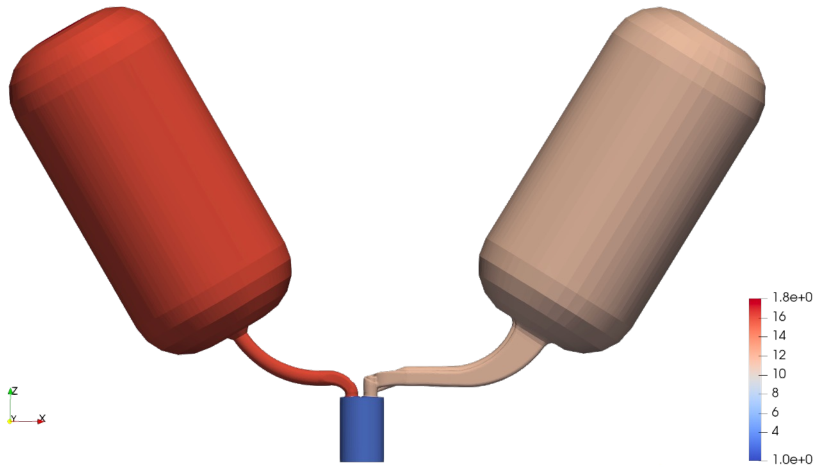




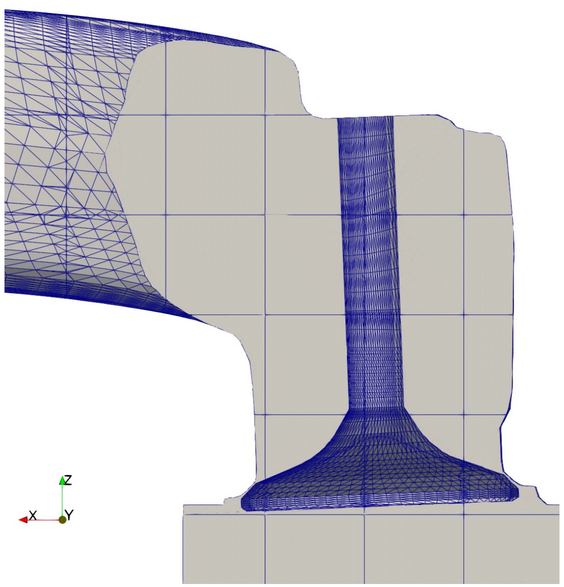
Domain	SNL Light-duty
# cells	112,855
# tri	146,844
Δx	1.0 cm

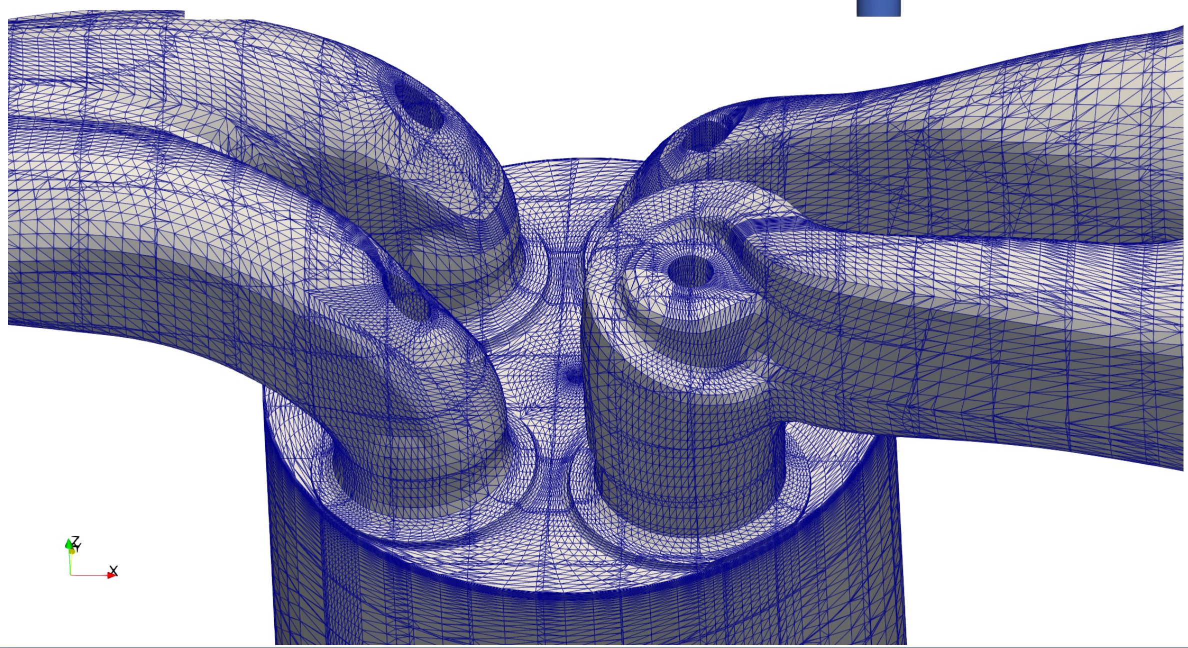
...full engine





✓ 18 surfaces

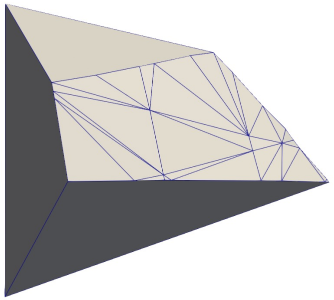


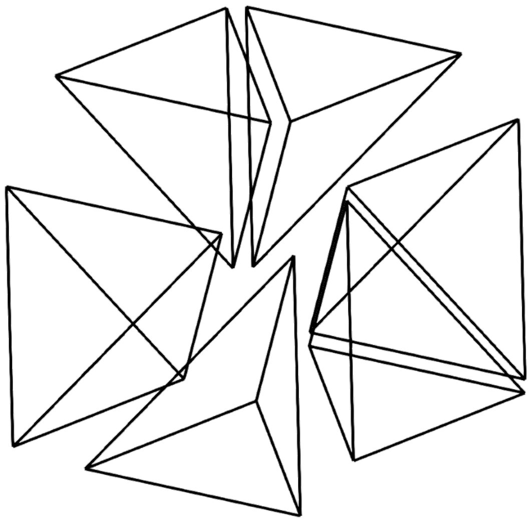


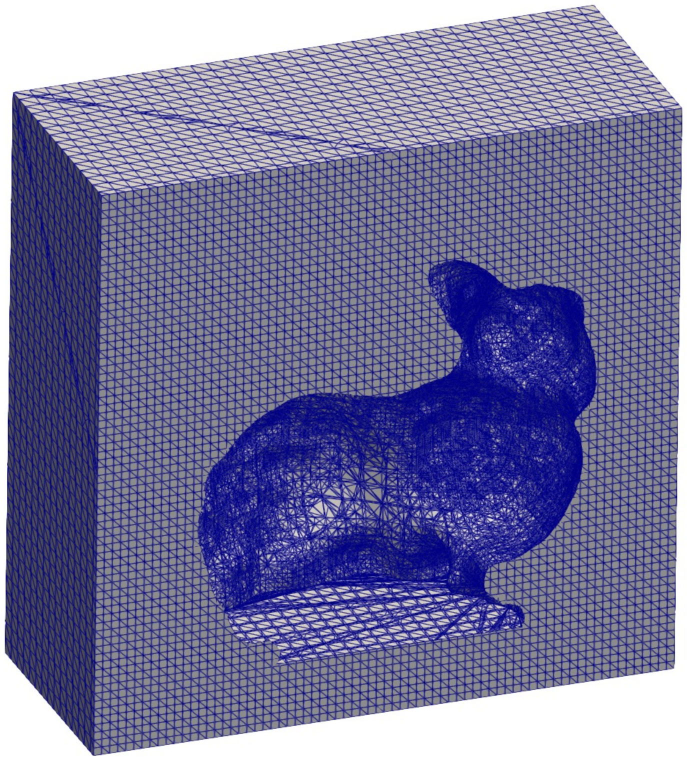
unstructured tet mesh

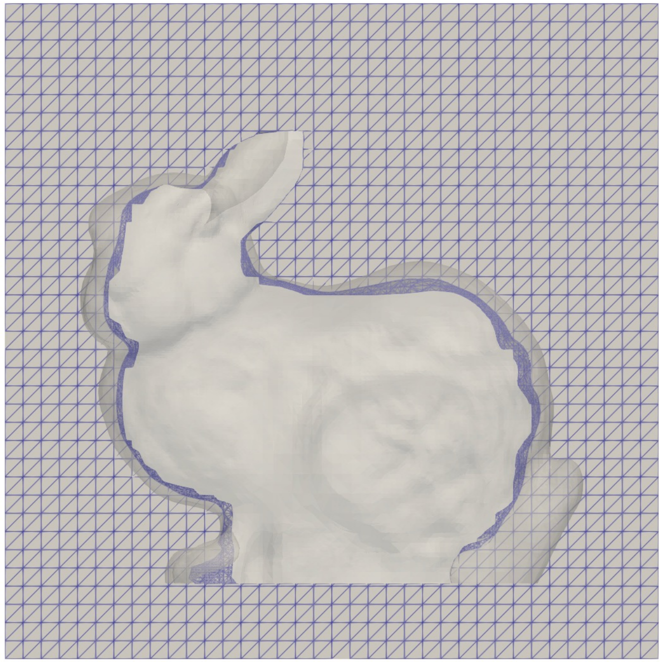
Domain	Stanford bunny
# cells	203,956
# tri	112,412
Δx	4.0 mm

✓ 6 tets per hexahedron





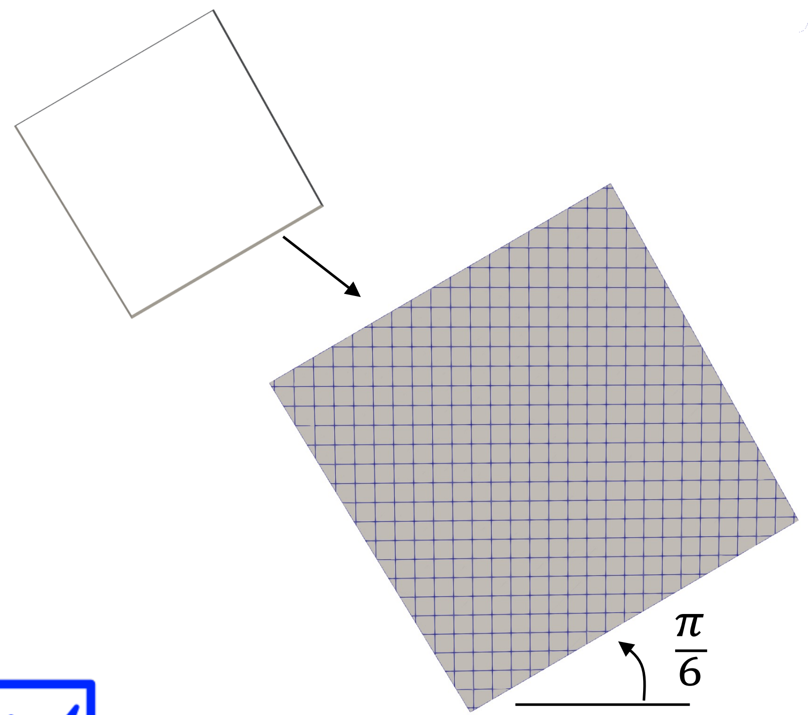


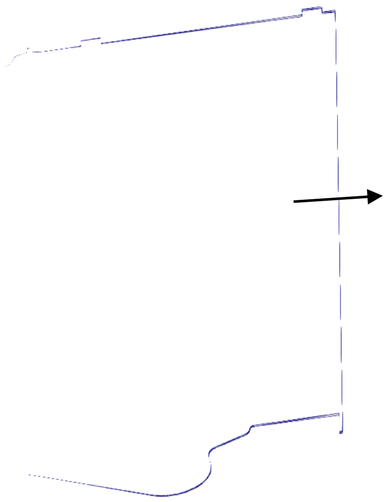


Domain	SNL Medium-duty
# cells	6,848
# tri	98
Δx	1.0 mm

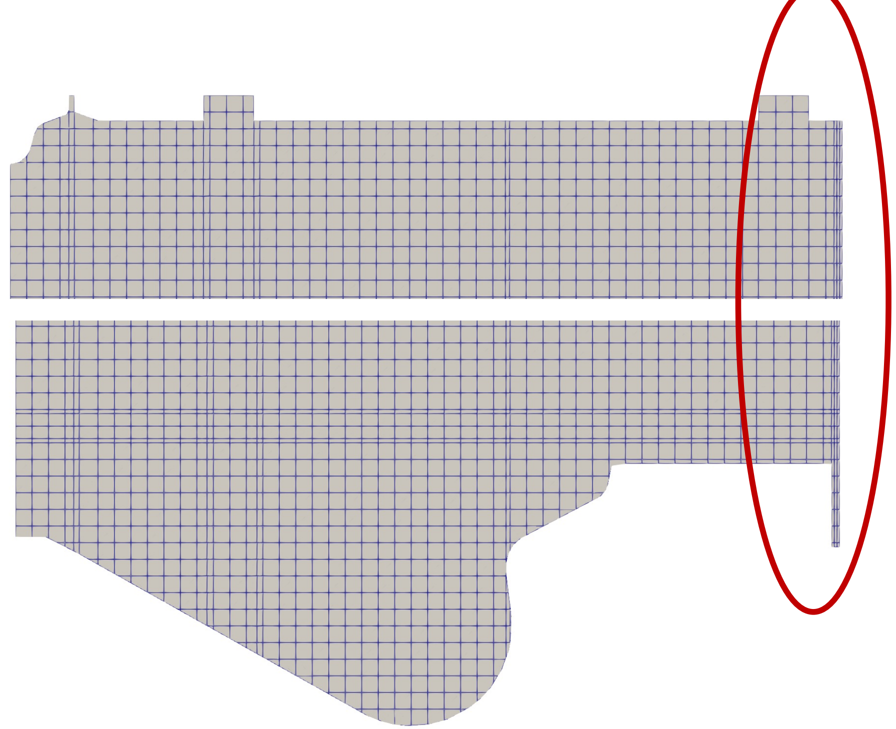
2D planar/sector cases

- ✓ From 2D polylines -> triangles





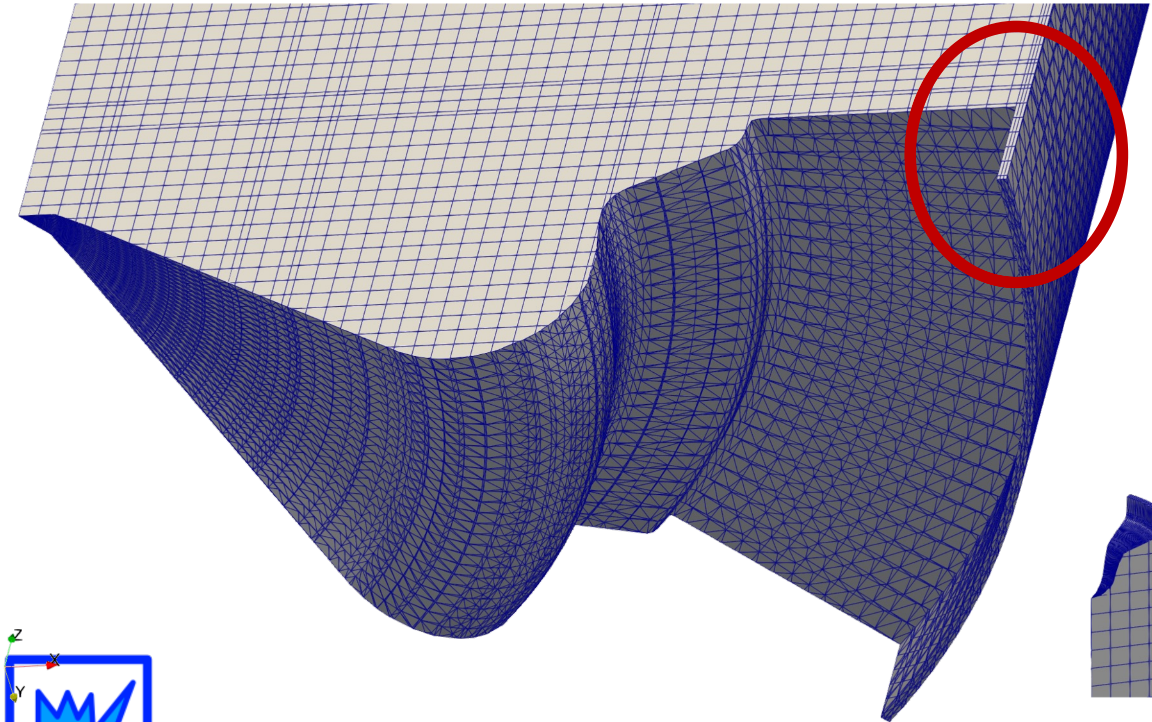
- ✓ Custom refinement

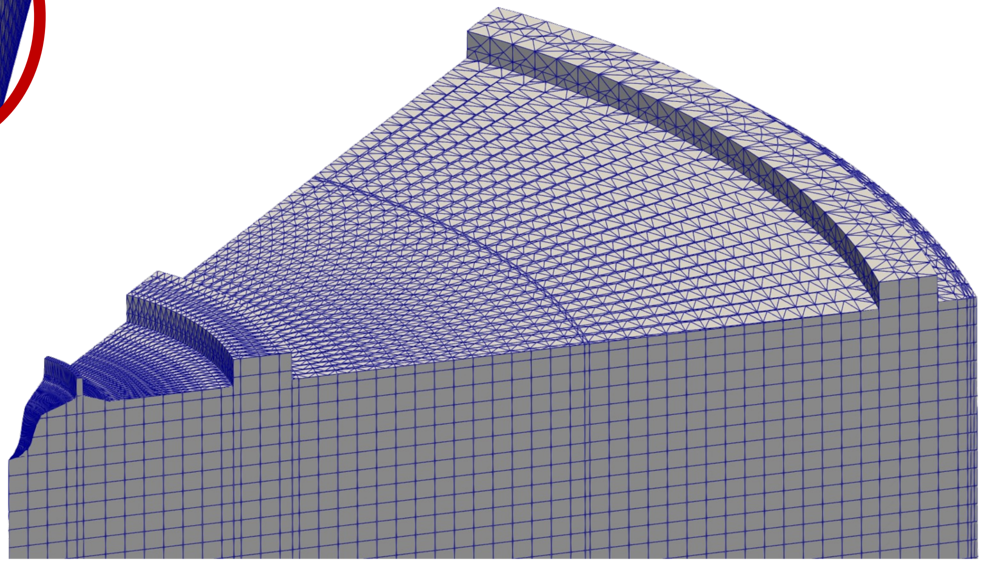


...2.5D cases

Domain	SNL Medium-duty
# cells	178,048
# tri	2,548
Δx	1.0 mm

- ✓ Prismatic layers at the liner
- ✓ Arbitrary feature lines on the cylinder head





CPU times comparison (Apple M1, 8CPU, on battery)

[s]	# cells	Intersect	Mesh faces	BC faces	Assemble	Total
Bunny #1	112k	1.11	0.13	0.26	0.23	**1.73**
LNF cylinder	1.3M	4.80	1.34	1.26	2.80	**10.2**
Dragon	98k	2.49	0.28	0.24	0.25	**3.26**
TCC Engine	272k	2.09	0.57	0.78	0.85	**4.29**
SNL LD	112k	3.16	0.57	0.70	0.59	**5.02**
Bunny+Tets	204k	2.09	0.27	0.60	0.53	**3.49**
Engine Sector	178k	0.32	0.07	0.03	0.18	**0.6**

Onward

- Enable combined AMR + Cut-Cell operation
- Cut-cell update on moving boundary
- Extend FV solver to cut-cells

Thank you!