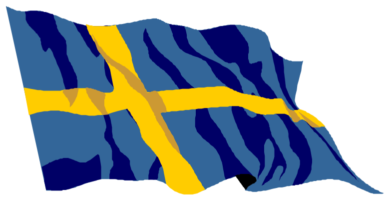
Rapsodi: Geometry preparation and grid generation

Anders Petersson
Center for Applied Scientific Computing
Lawrence Livermore National Laboratory
Livermore, California



Rapsodi team: David Brown, Kyle Chand, Bill Henshaw, Anders Petersson

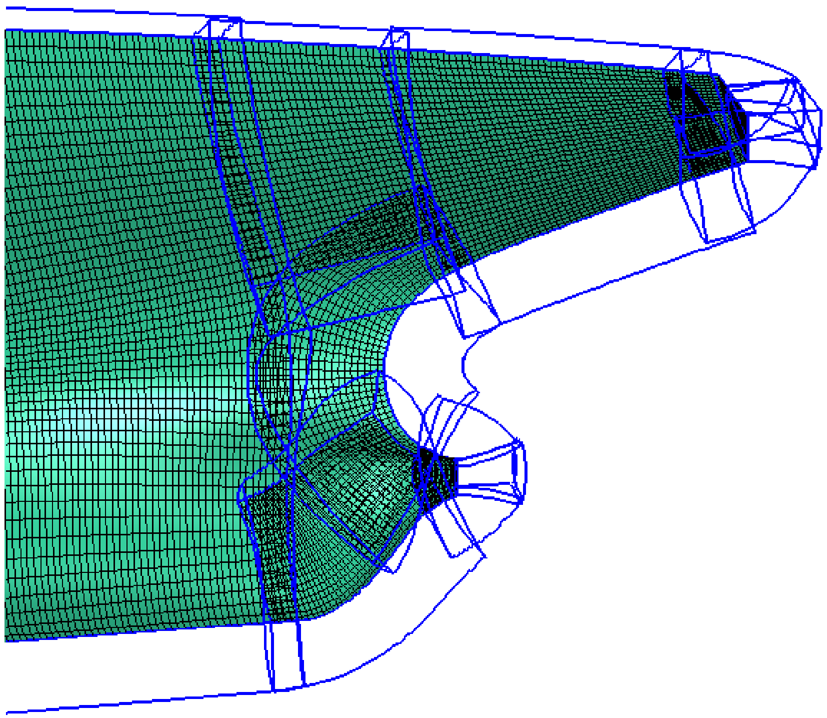
This work was performed under the auspices of the U.S. Department of Energy by University of California Lawrence Livermore National Laboratory under contract W-7405-Eng-48.

Our grid generation approach

- Import/cleanup or create surface description
- Establish projectable surface (details to follow)

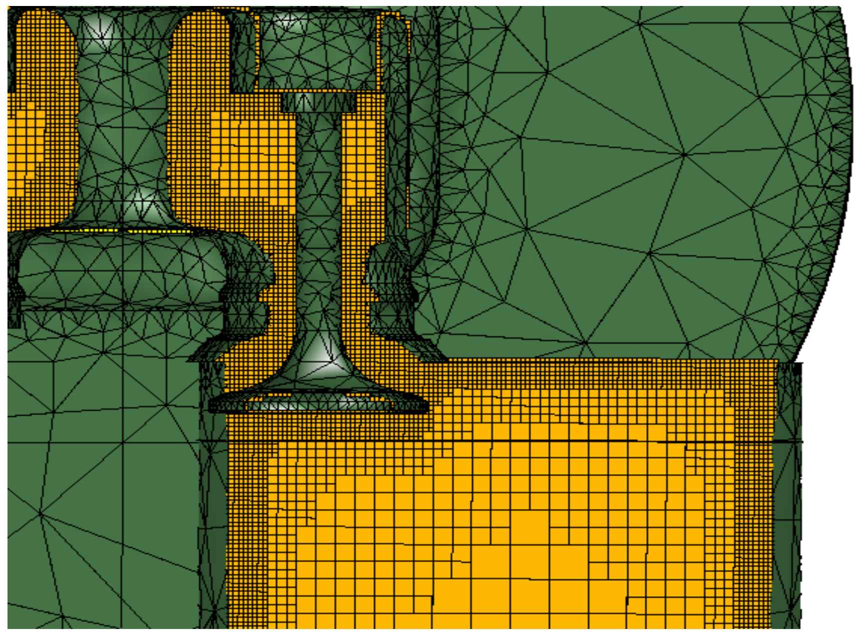
a) Body fitted (overlapping) grids

- Grow overlapping surface grids
- Grow volume grids into domain



b) Cartesian embedded boundary

- Refine global triangulation
- Use cubes from Cart3D



Our approach to CAD interaction

- Read geometry from neutral (IGES) file:
  - Modular approach: natural separation CAD <-> grid generation.
  - No CAD licence needed.
  - Generic to many (all) CAD programs. Legacy drawings use IGES.
  - Small overhead enables moving grid and AMR simulations.
  - Inconsistent geometry, no connectivity in IGES => some fixup/prep needed.
  - Hard to automate fixup! Needs good interactive editing capability.
- Competing alternative: Direct interface:
  - Link grid generator to geometry kernel through API.
  - Internal geometry description => Few errors.
  - Proprietary code, Interface depends on CAD program.
  - Licensing very expensive on massively parallel machines.
  - Efficiency? (memory / CPU)
- Common problems:
  - CAD drawings are often too detailed (made for production).
  - Editing required when only part of drawing is relevant to analysis.
  - Guesswork sometimes needed about designer's intent.

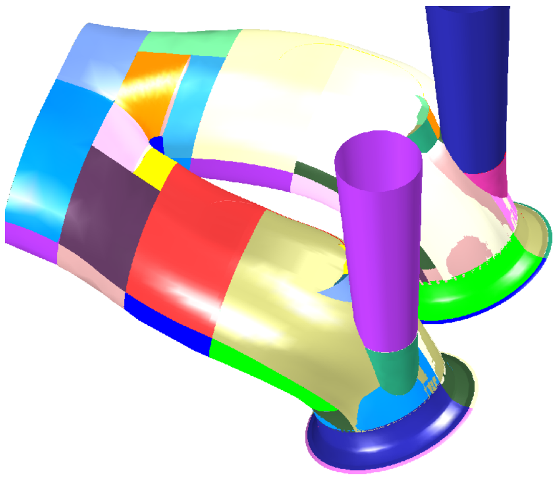
Overview of geometry process

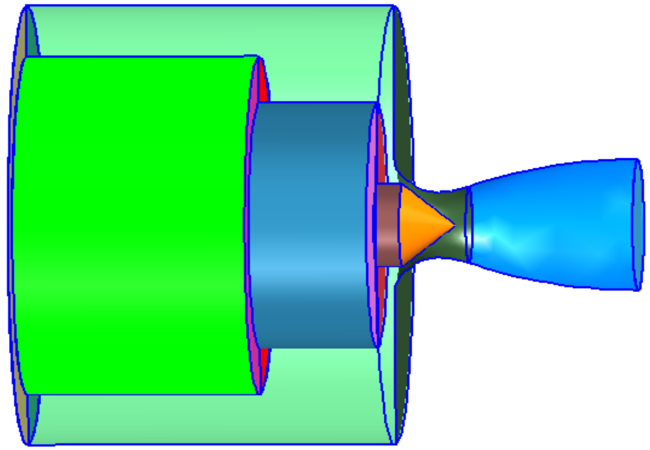
- Read IGES file, check trimming curves (no user interaction)
- Minimal repair of broken trimming curves -> enables evaluation
- Compute topology (no user interaction):
  - Map all trimming curves to 3-D edge curves.
  - Match neighboring edge curves, after splitting if necessary.
- Compute global surface triangulation (no user interaction):
  - Distribute grid points along boundary edges in 3-D
  - For each patch: map 3-D grid points to parameter space.
  - Delauney triangulate each patch in parameter space.
  - Map all grid points on all patches to 3-D.
  - Remove duplicate points along boundary edges.
- Result: a surface representation without gaps or overlaps,

allowing robust projection

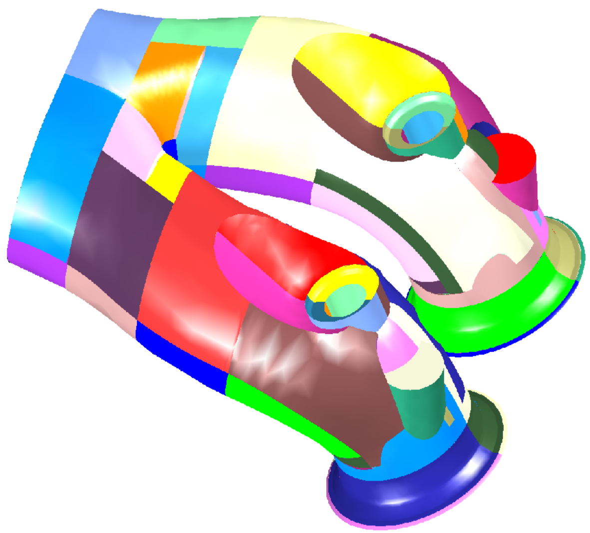
Workflow with rapsodi

1. Read / Make geometry



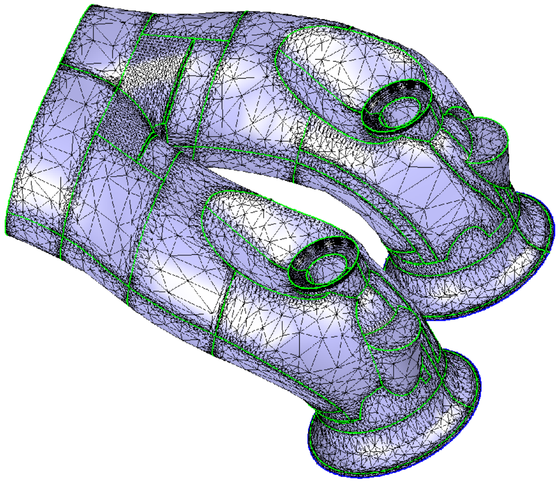


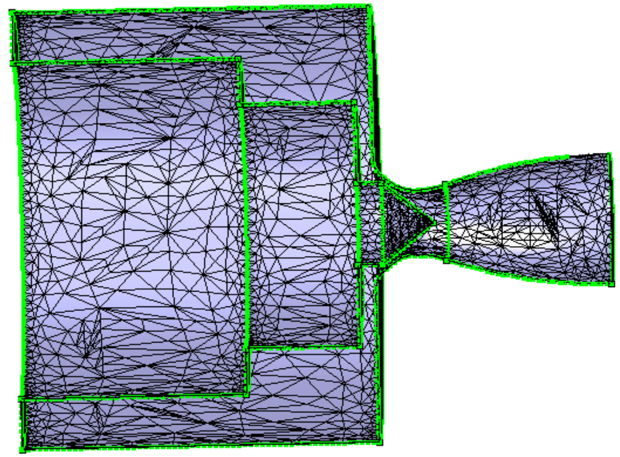
2. Correct surface patches/ remove irrelevant parts



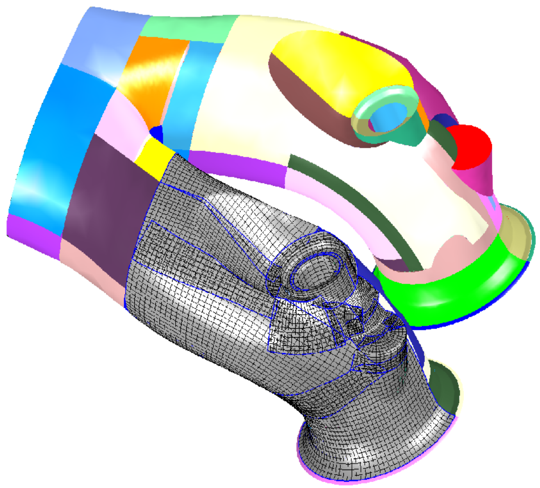
Workflow with rapsodi

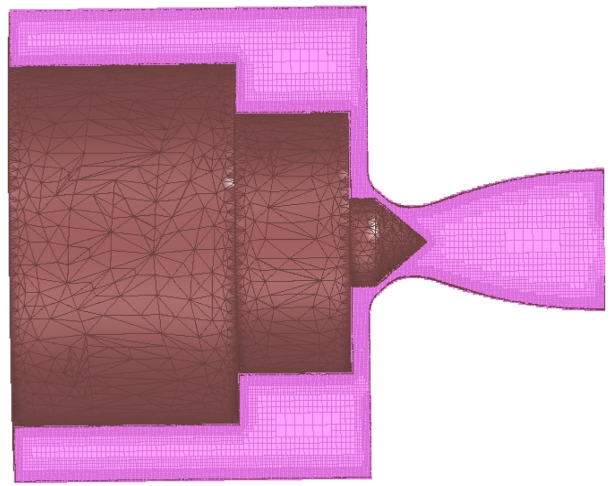
3. Connect surface patches & triangulate



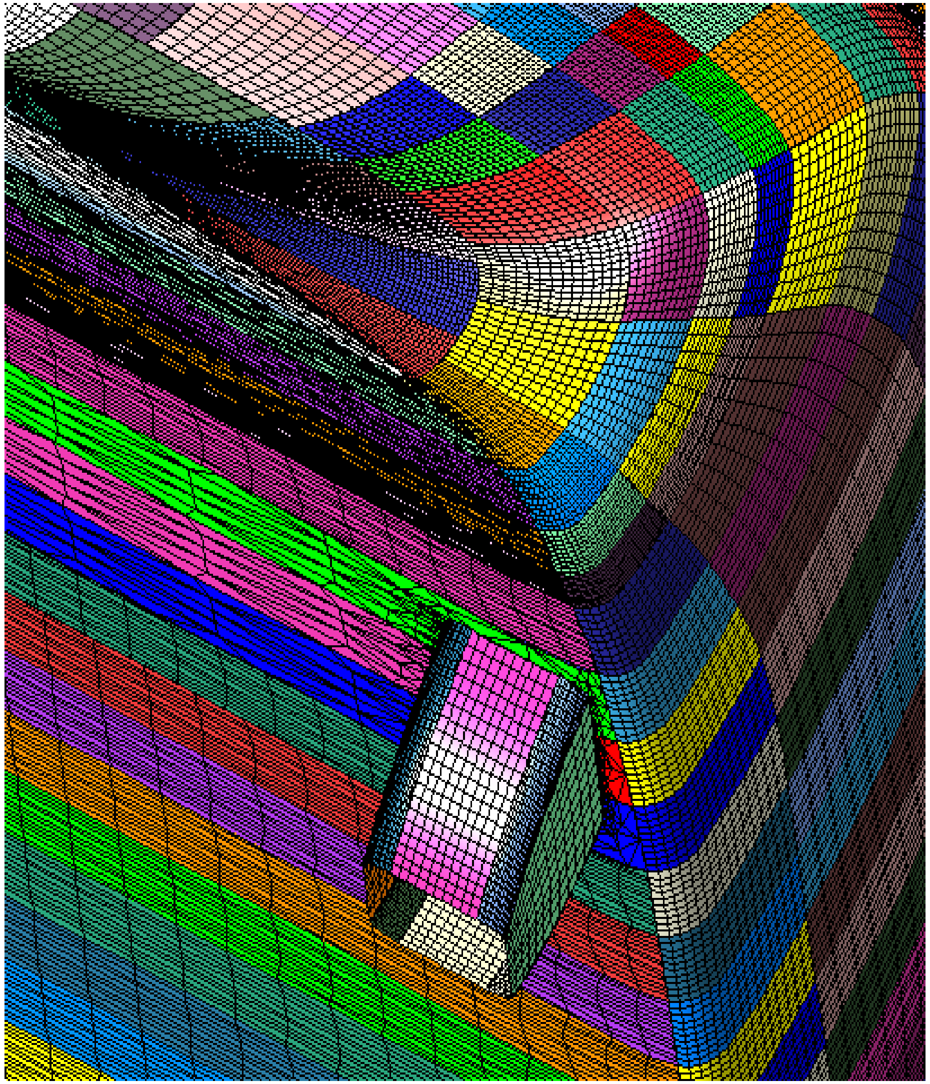


4. Build boundary fitted grids or Cartesian grids

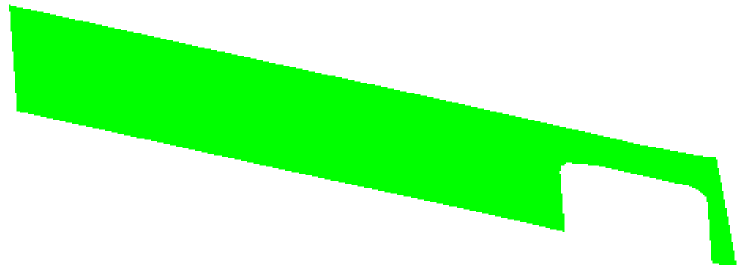




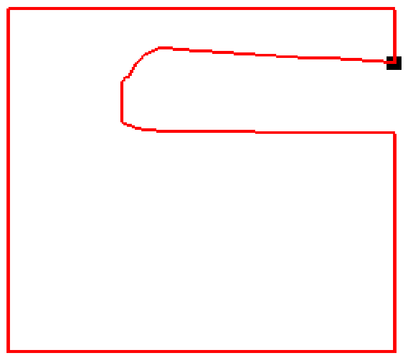
Surface: trimmed / untrimmed patchwork



Trimmed surface: Untrimmed surface plus trimming curve(s)

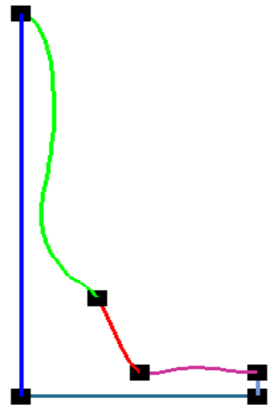


Trimming curve restricts surface in parameter space



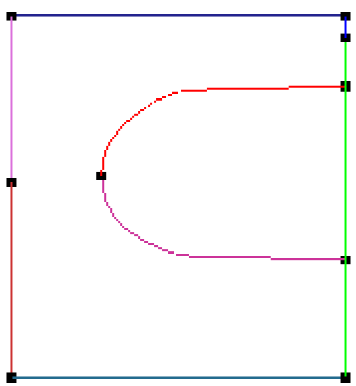
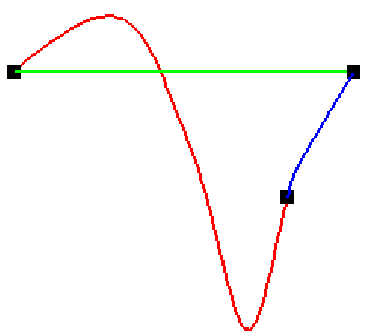
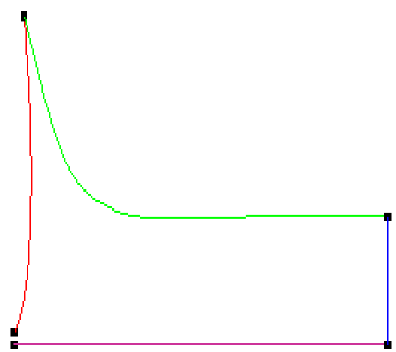
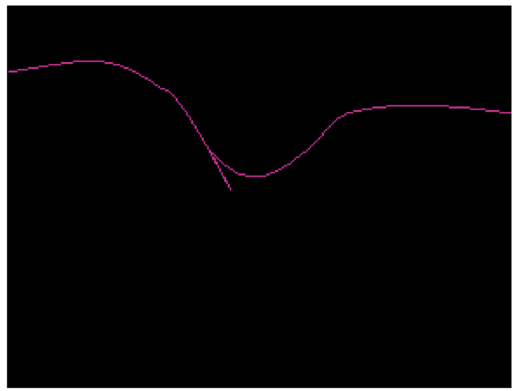
Trim curves

Trim curves need to be
1. Periodic
2. Correctly oriented
3. Inside unit square
4. Not self-intersecting



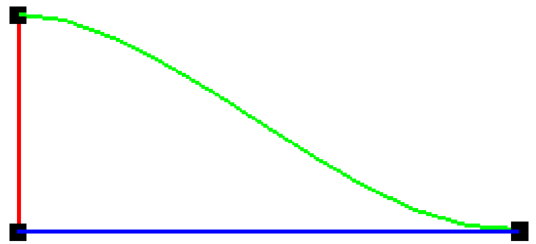
Trim curves are often collections of sub-curves:
1. Each sub-curve must occur exactly once
2. Each sub-curve must end where another sub-curve begins

Examples of inconsistent trim curves found in IGES files:

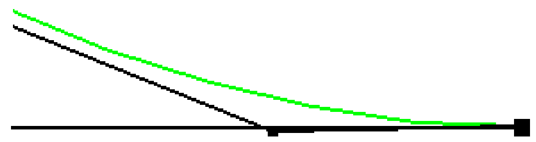





Almost valid trim curves

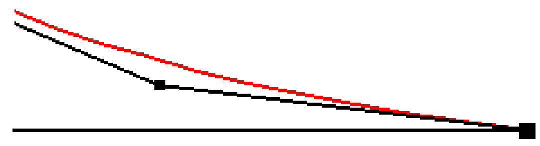
Self-intersection near cusp corner:



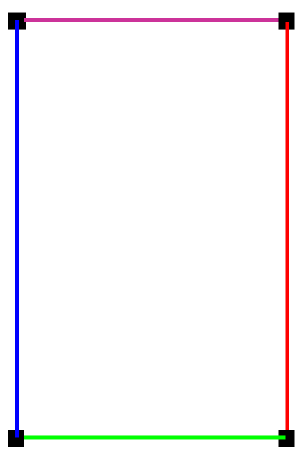
Control point polygon tangled:



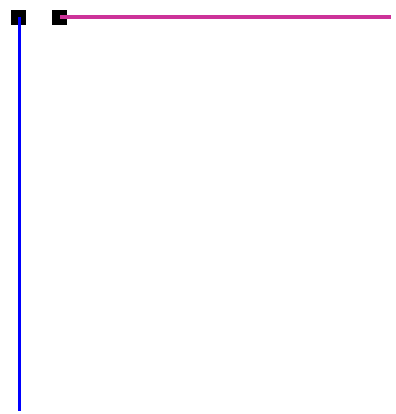
After untangling:



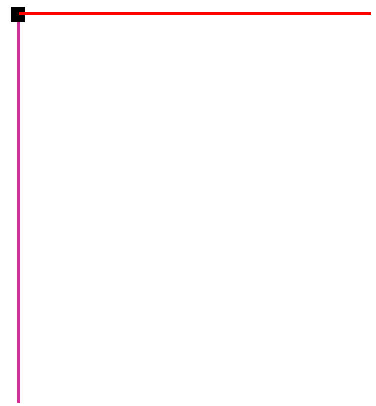
Gap at one corner:



Closeup:



Extrapolate curves to intersection point

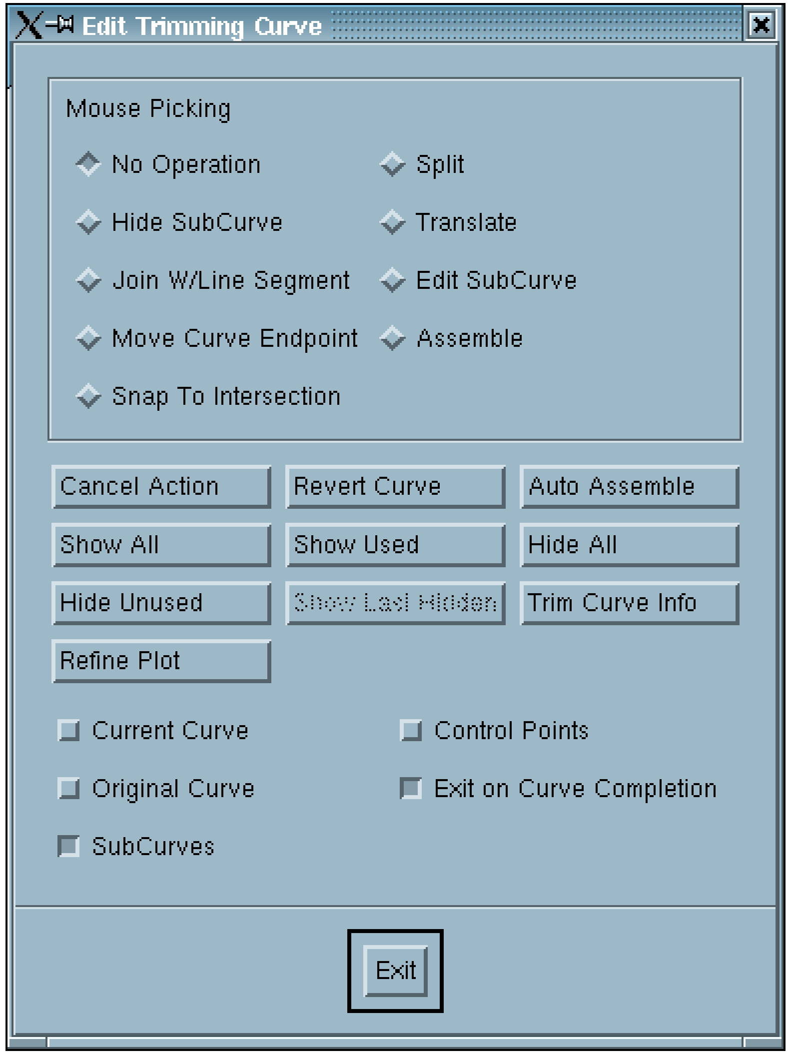


Correcting trim curves

NOTE: This procedure makes trimming curves formally valid. It doesn't aim at removing gaps, overlaps or discontinuities at patch boundaries.

1. If possible, make a valid trim-curve by modifying existing sub-curves:
   - Move end points
   - Intersect 2 sub-curves
   - Hide sub-curve
   - Join end points/w line segment
   - Split sub-curve

2. In difficult cases, recompute intersection curves between neighboring surfaces.
This adds new sub-curves to the trim curve.



3. The topology algorithm will reveal any remaining inconsistencies by edge-matching.

Topology (how does it fit together)

(a.k.a. making your computer put together a (cheap) jigsaw puzzle)

(Algorithm based on paper by Steinbrenner, Wyman & Chawner)

1. Construct boundary curves by mapping each trim curve to physical space. Also make boundary nodes from start/end points.

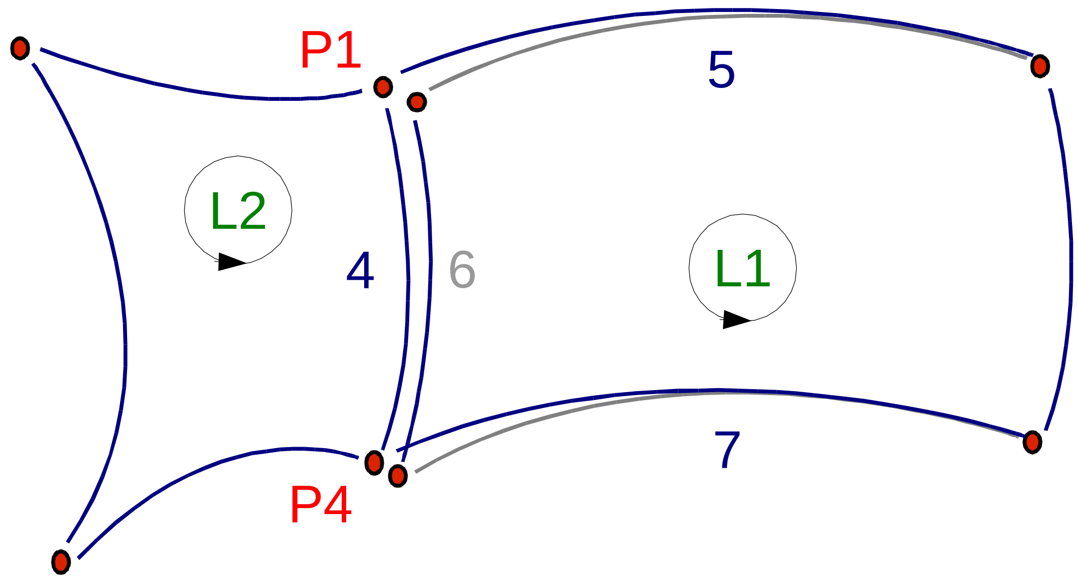
2. Search for a nearby matching boundary curves and merge curves by modifying end points and connectivity in the loop.

Maintain periodicity of each loop during the algorithm to ensure consistency

---

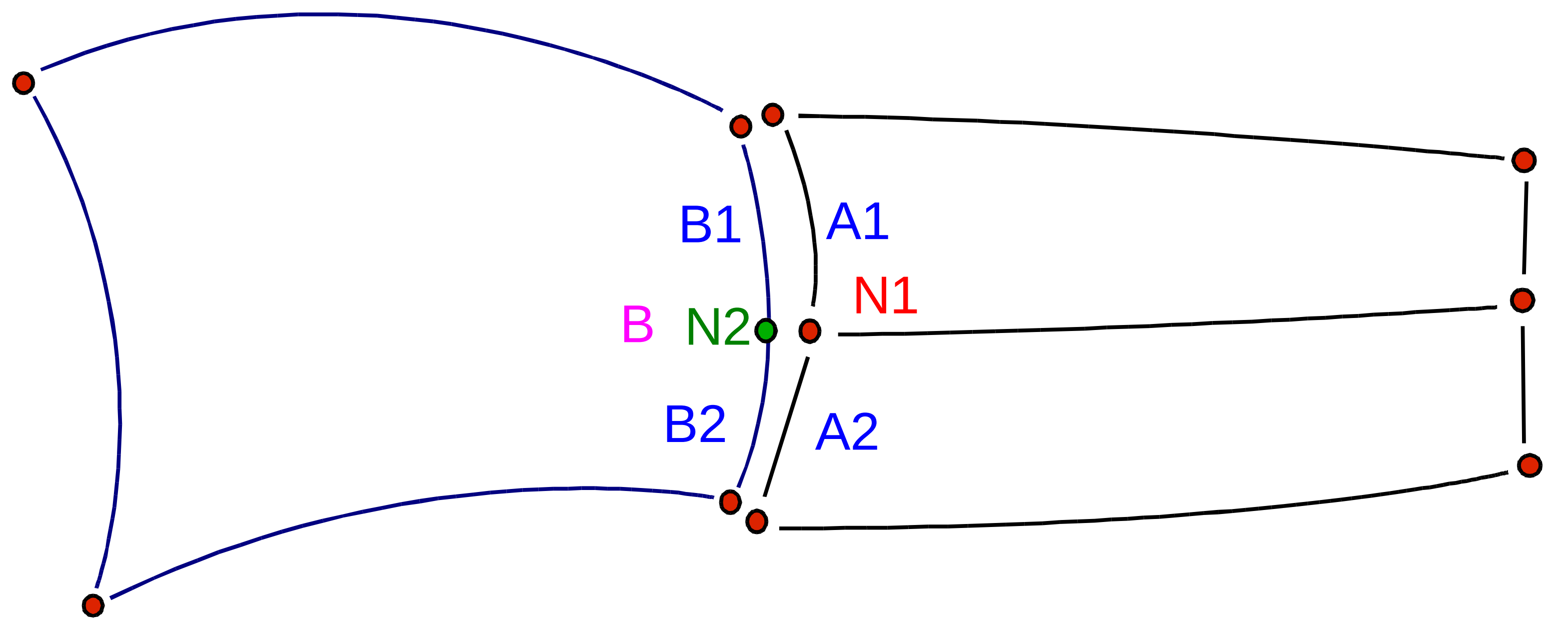
Example: Merge curves 6 and 4:

- Curve 6 replaced by curve 4 in loop L1
- Curve 5 modified to end at point P1
- Curve 7 modified to start at point P4
- Curve 6 not used after merge



Topology

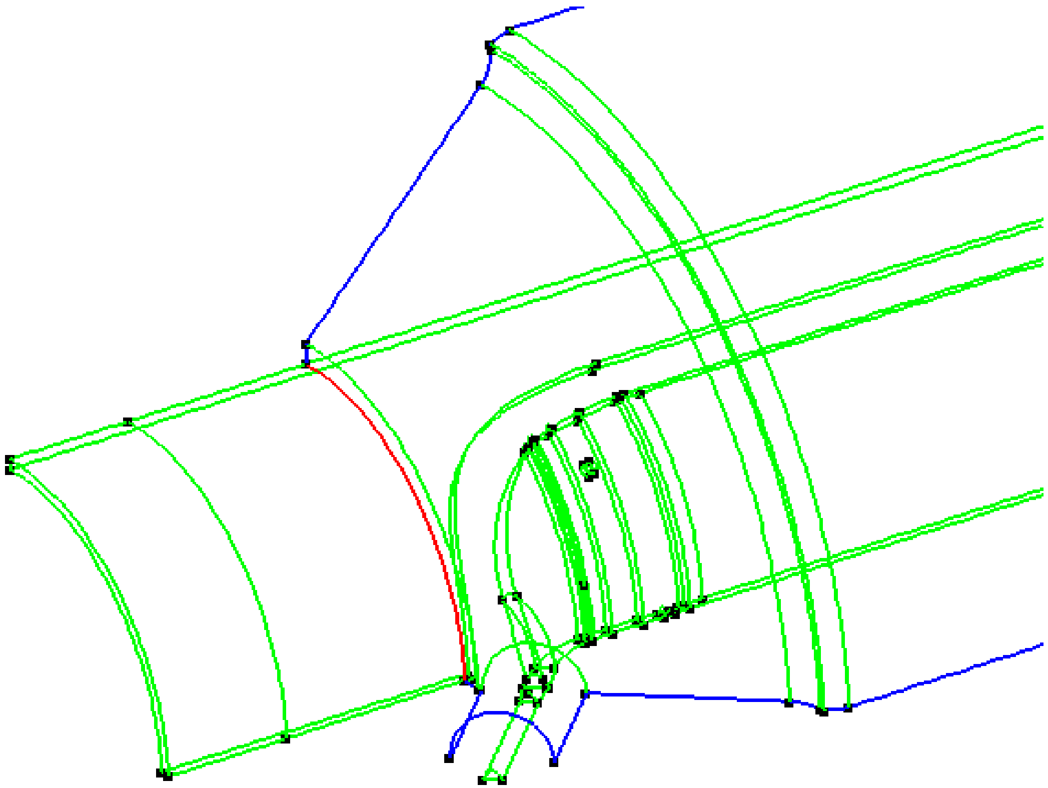
3. Split unmerged curves by nearby endpoints from other curves.
Then try to merge the split curves.



- Node N1 splits curve B, creating new curves B1 and B2 and new node N2.
- Curve B is replaced by B1 and B2 in the loop of the left surface.
- Curve A1 and B1 are merged, as well as A2 and B2.

Topology

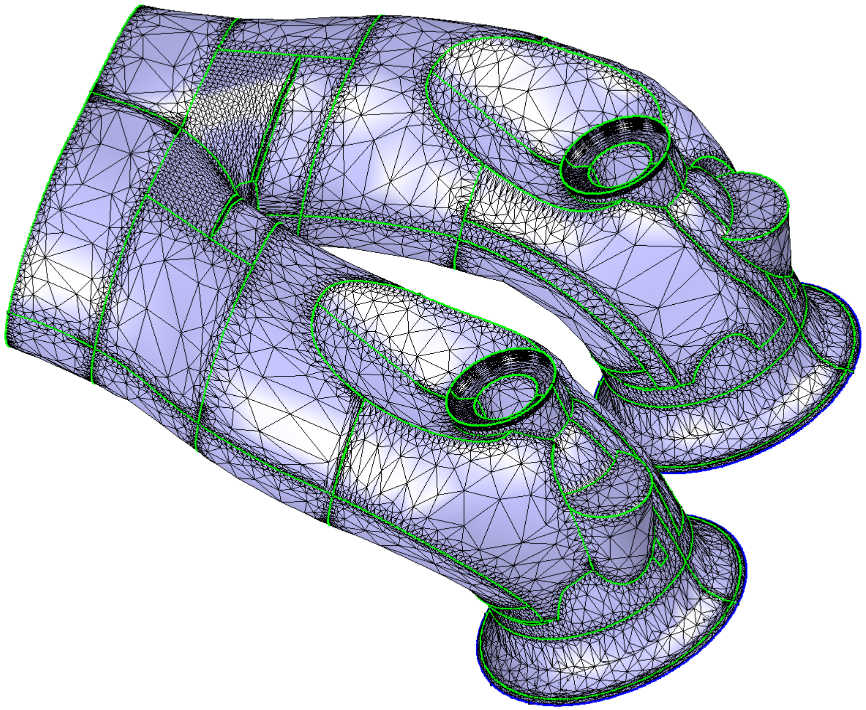
The curves are coloured based on how many surfaces they connect



A model is "watertight" when all curves are green. Blue are boundary curves.
Red curves, connected to more than two surfaces indicate a non-manifold geometry.

Surface Triangulation

- Distribute grid points along boundary curves
- Project grid points to parameter space => boundary nodes for triangulation.
- Triangulate each surface in parameter space using Delauney.
- Merge triangulations by removing duplicate nodes along boundary curves.



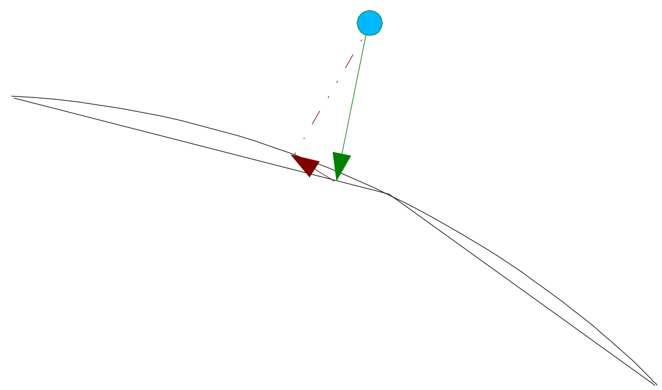
Features:

- The triangulation respects all surface patch boundaries.
- Each triangle lies on exactly one surface patch.

Using the triangulation

Surface grid generator needs to project (nearby) points onto the surface

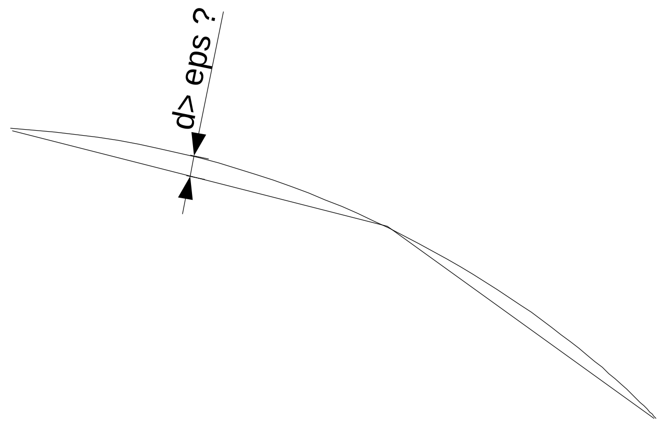
Project point onto global triangulation using:
a) ADT tree search (no initial guess)
b) Walking algorithm



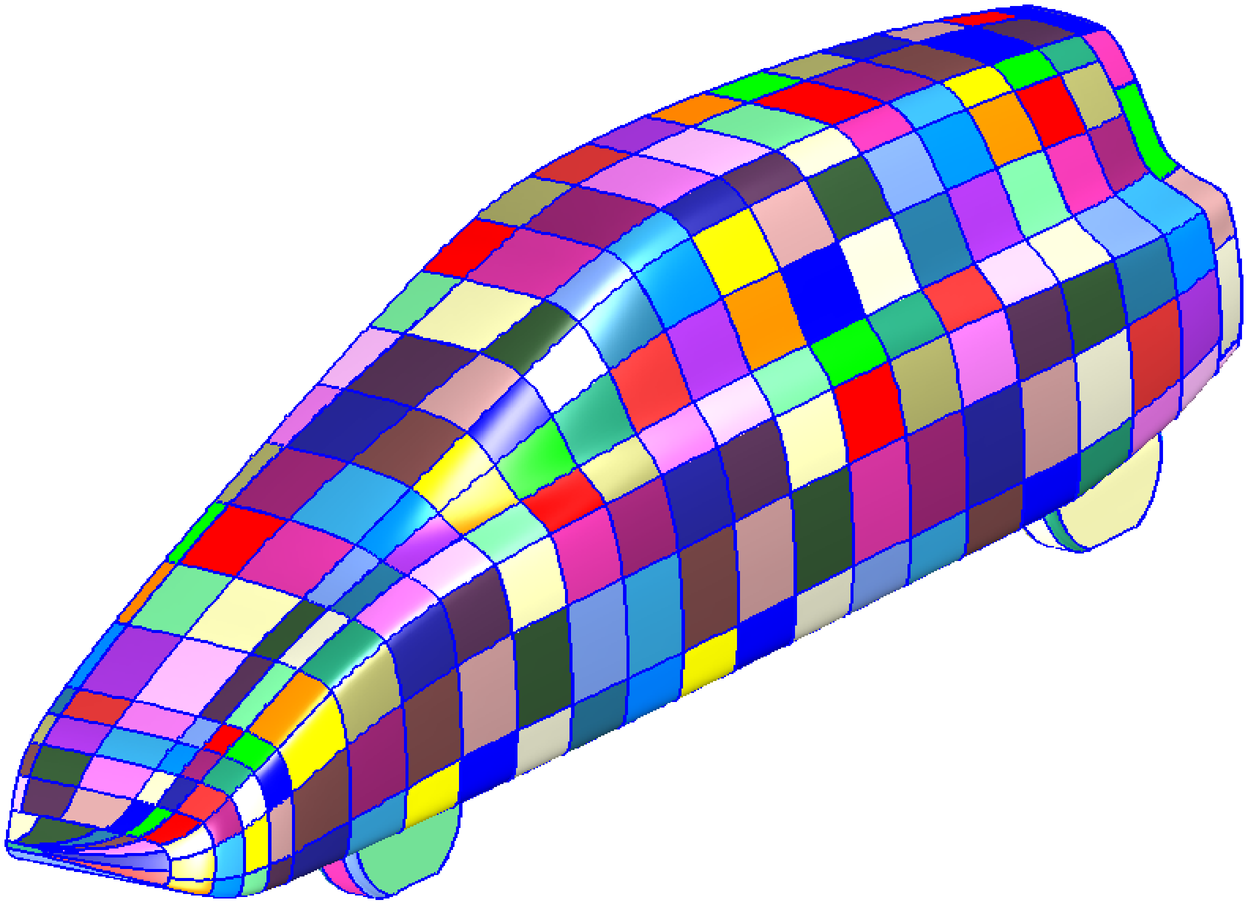
Closest triangle provides
a) surface patch number
b) initial guess for Newton's method

Triangulation needs to resolve geometry well for embedded bndry grid

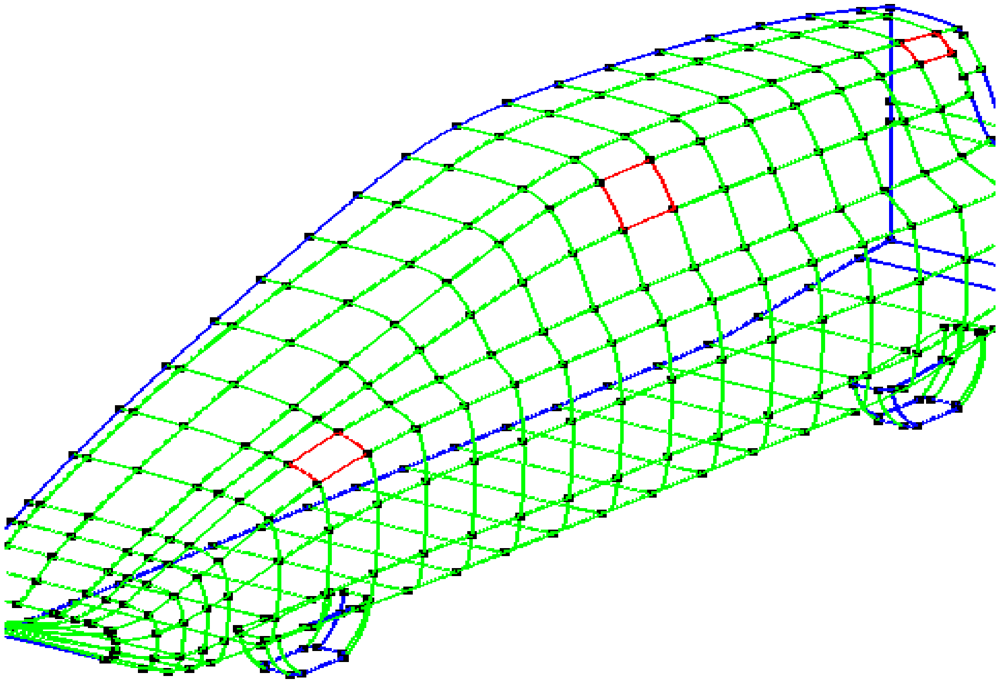
- Check midpoint distance to curved surface
- Add new node point if needed
- Re-triangulate all refined patches



Example 1: ASMO car model

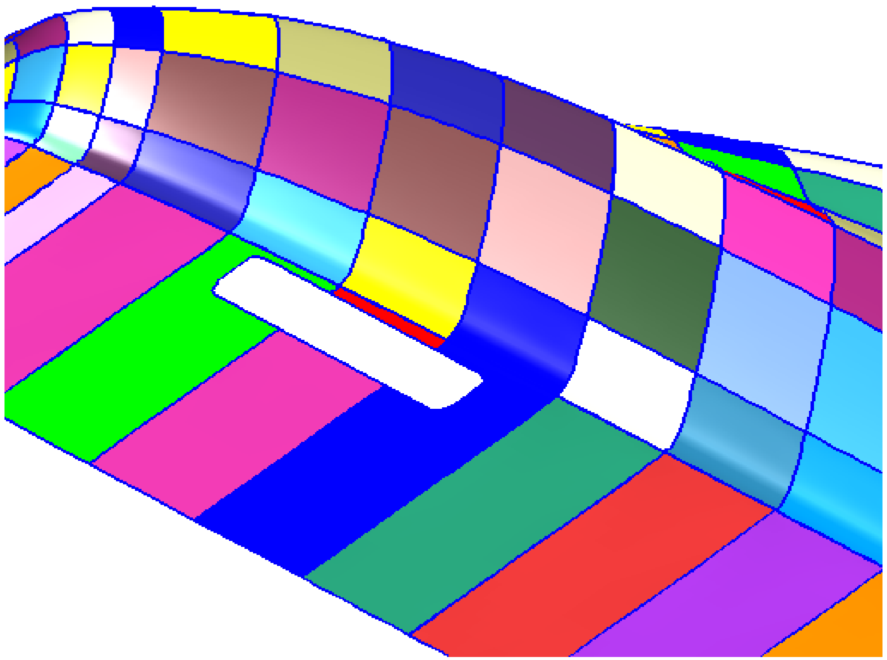


Checking the validity

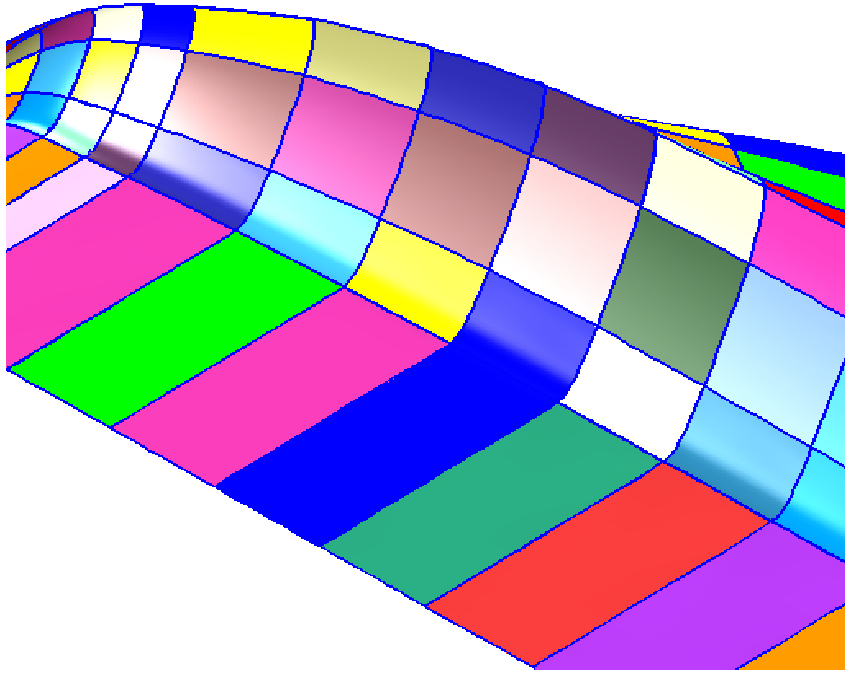


Topology algorithm reveals three duplicate surface patches, here outlined in red.

Localizing grid generation by simplifying/splitting the model

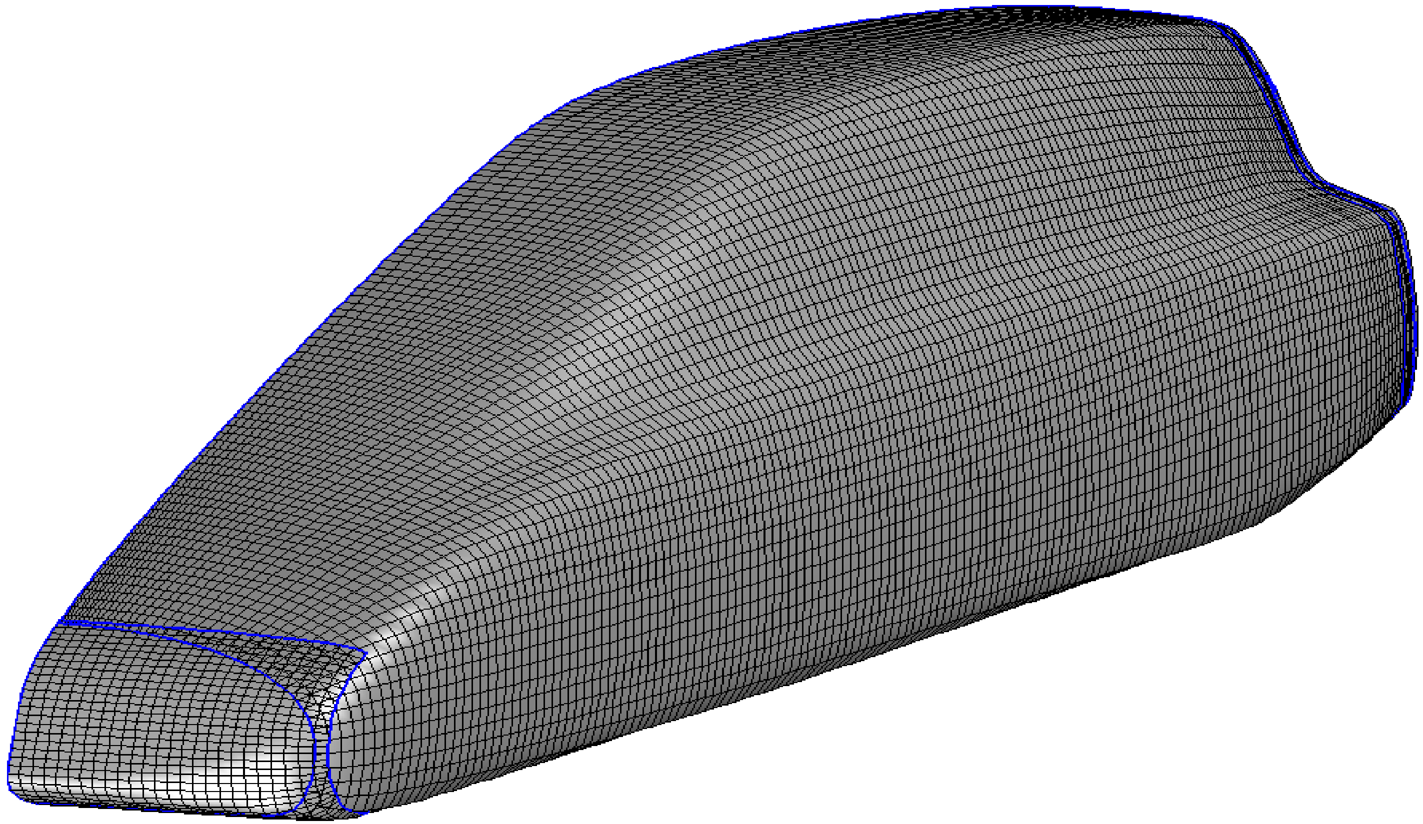


Deleting the surfaces around the front wheel leaves a hole



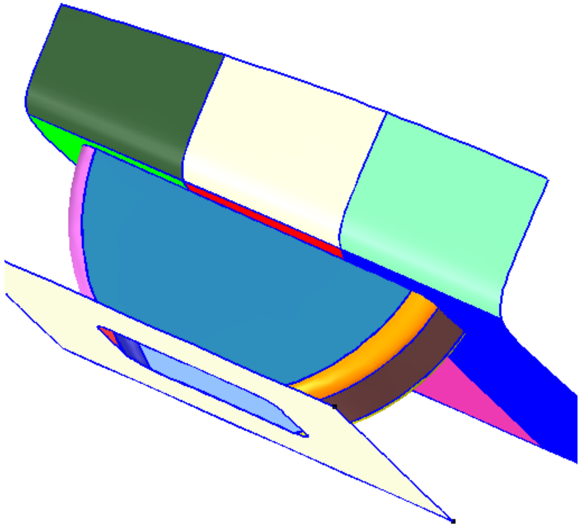
Deleting the trimming curves covers the hole

Surface grid generation on main body

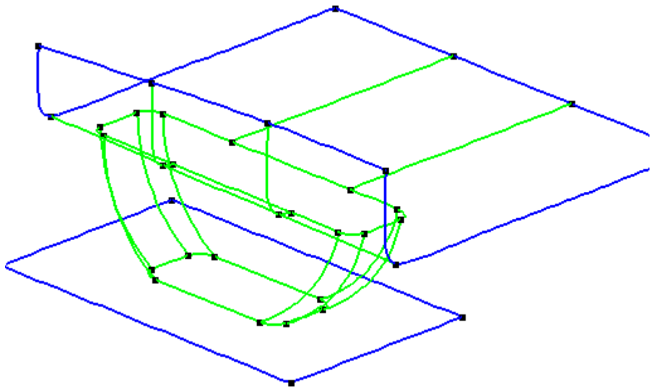


Five surface grids cover the main body

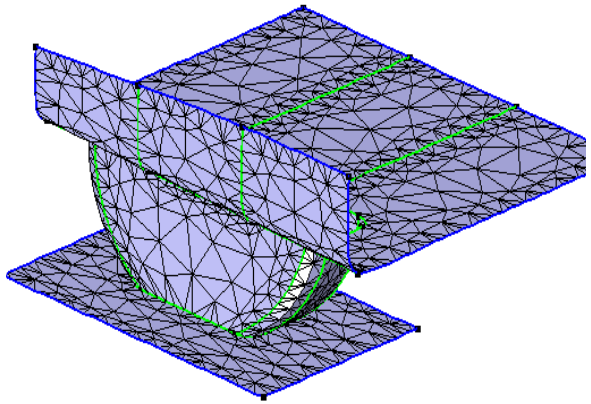
Front wheel



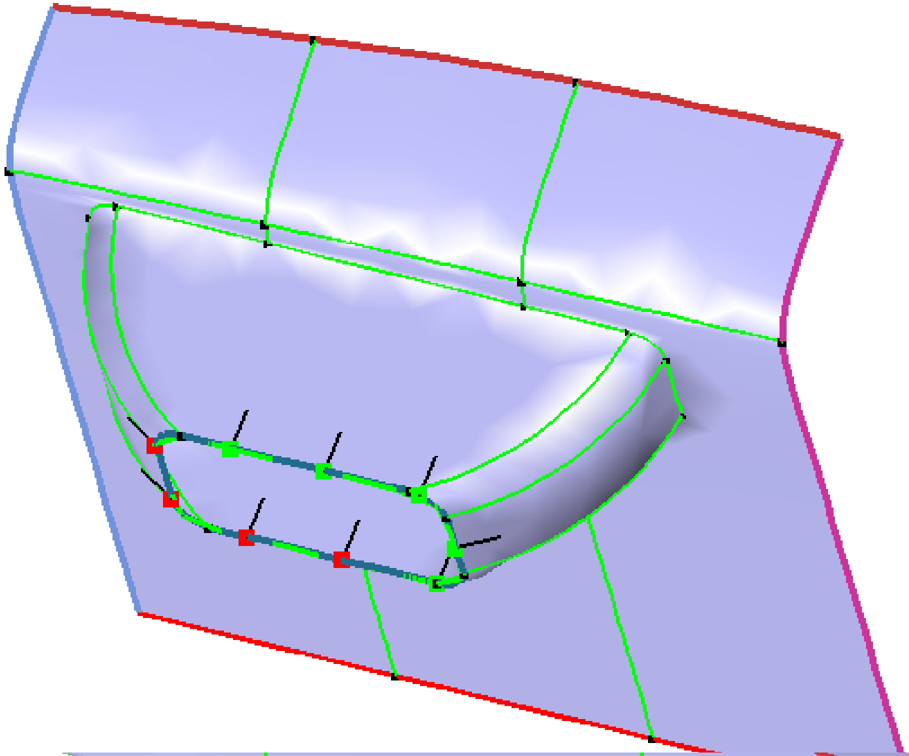
Surface patches near wheel

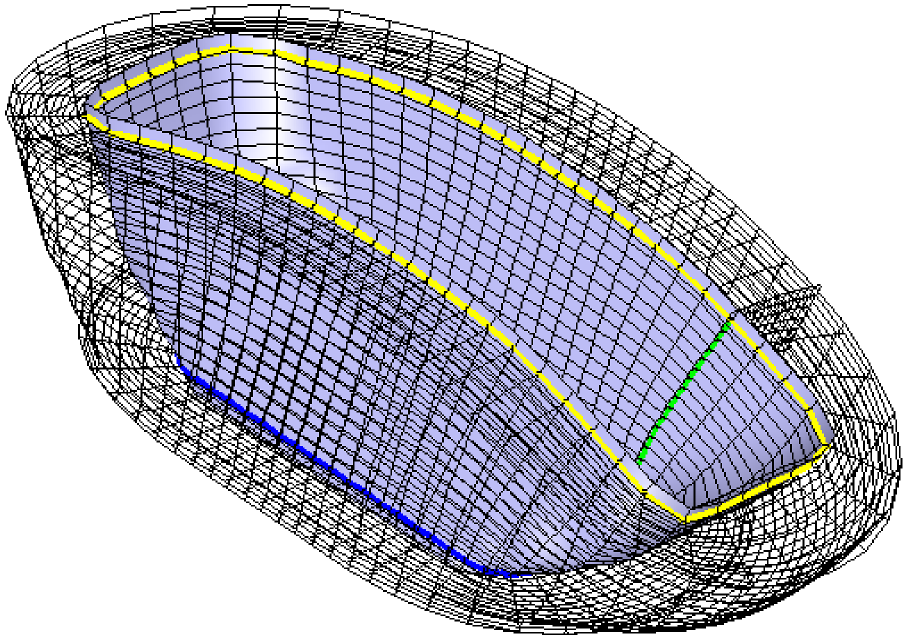


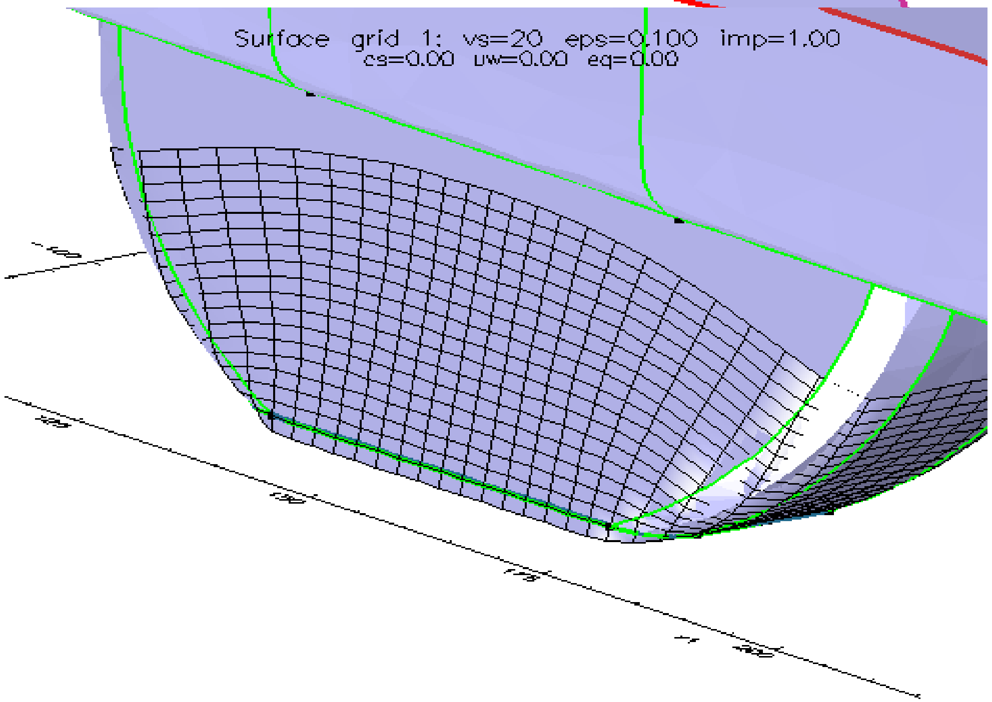
Topology

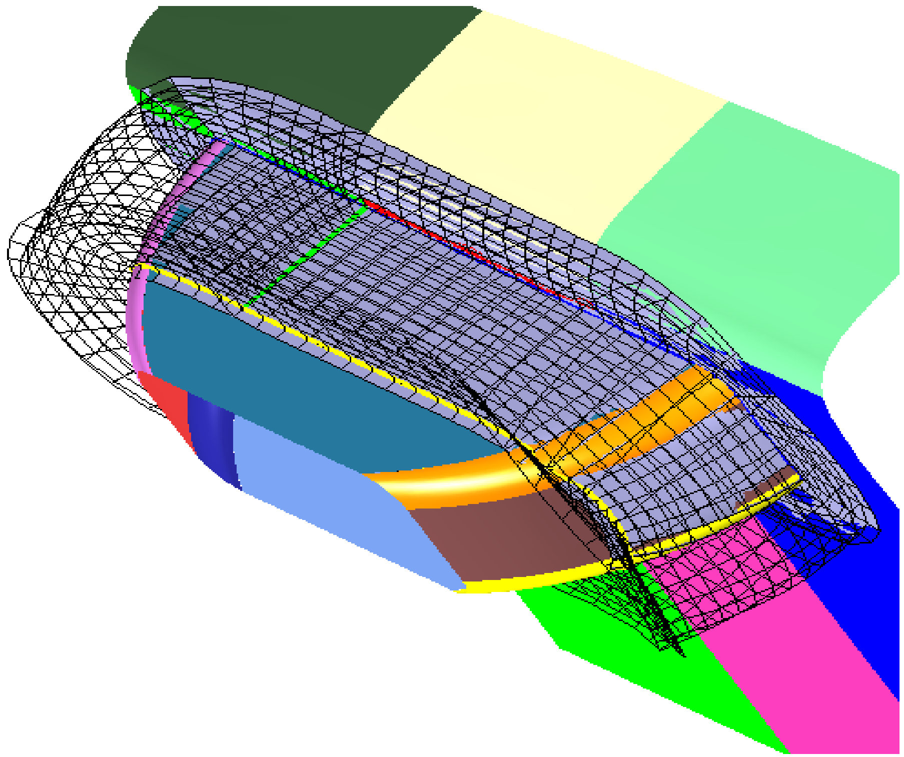


Surface triangulation

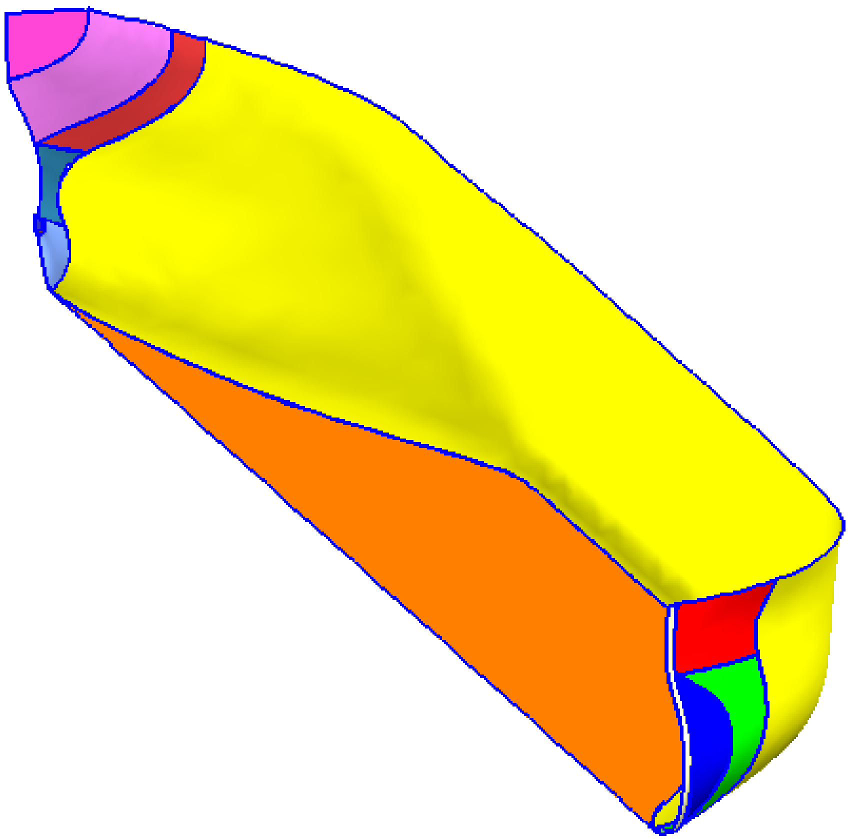




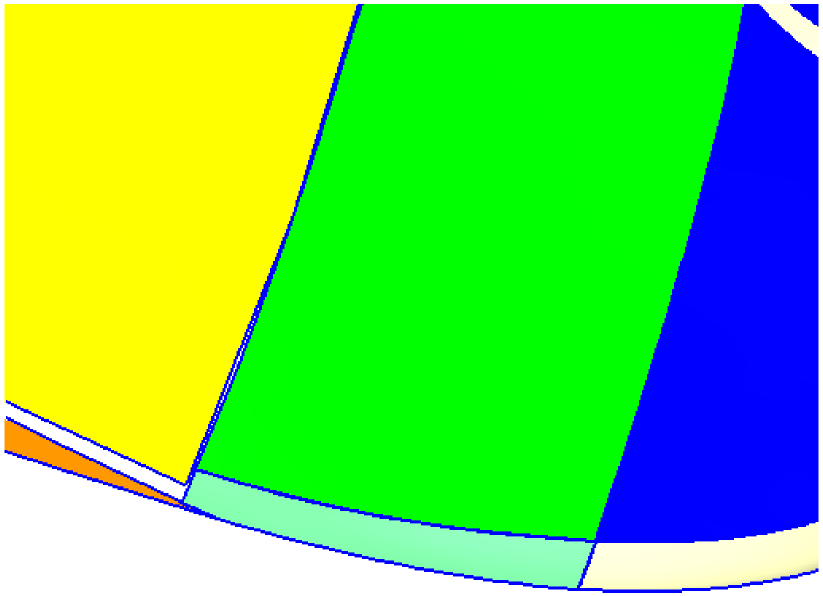

Surface grid 1: vs=20 eps=0.100 imp=1.00
cs=0.00 uw=0.00 eq=0.00



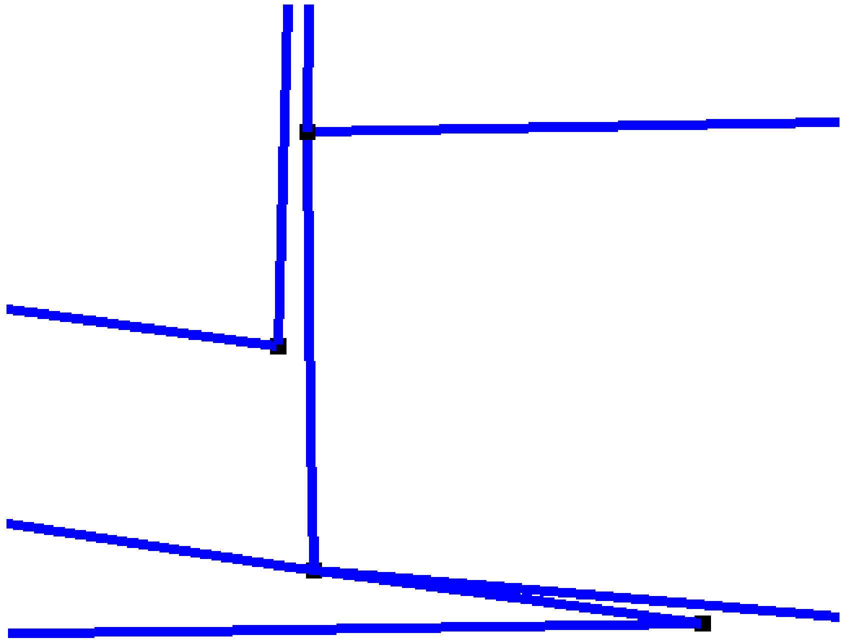
Example 2: KVLCC2 ship



Inconsistent patches => Leaky ship!

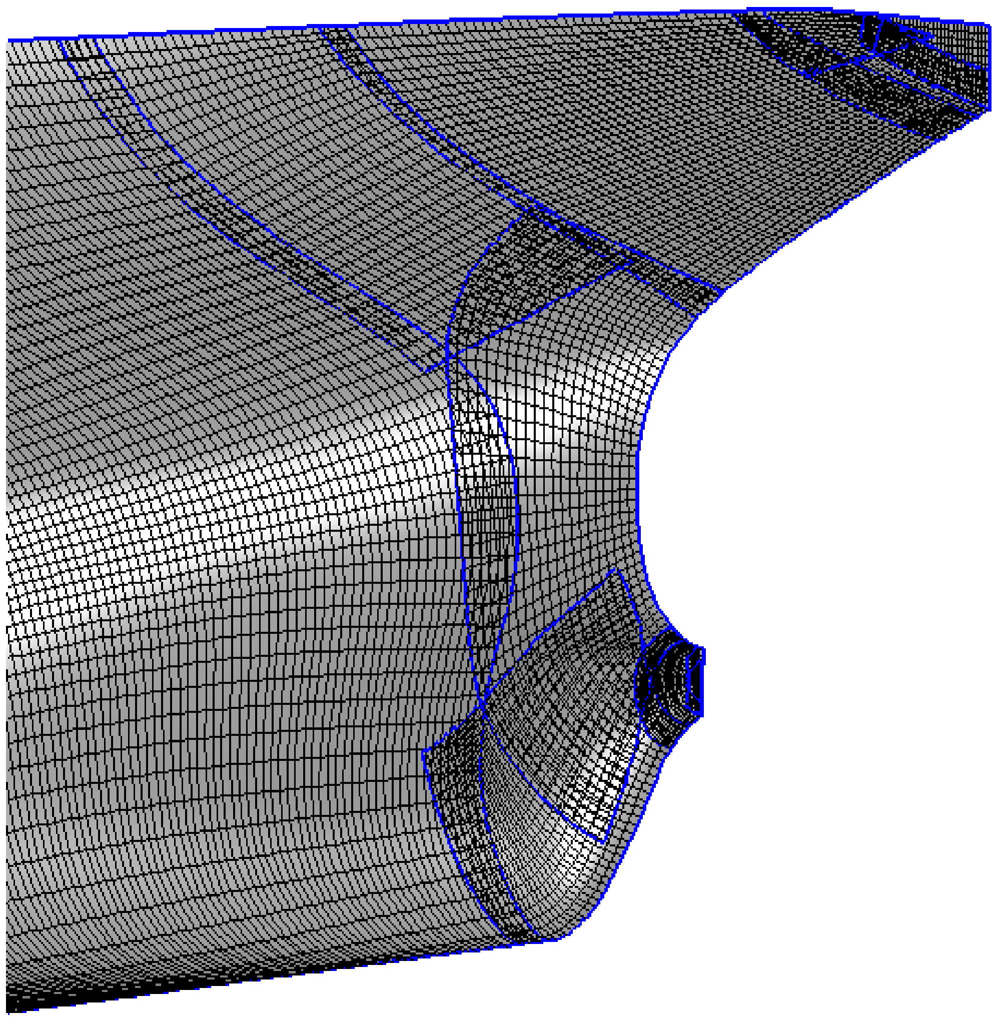


Gap between patches near bow

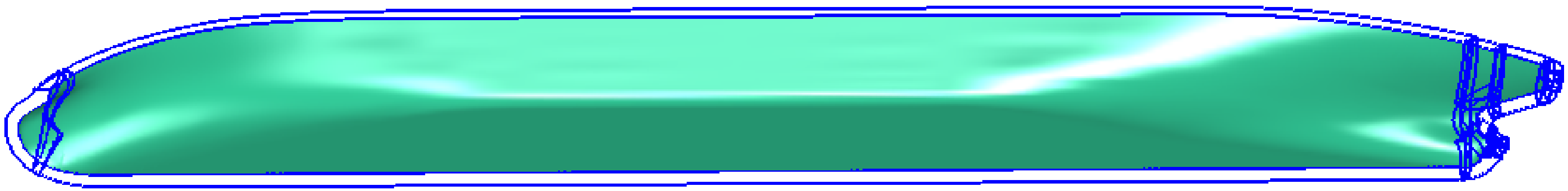


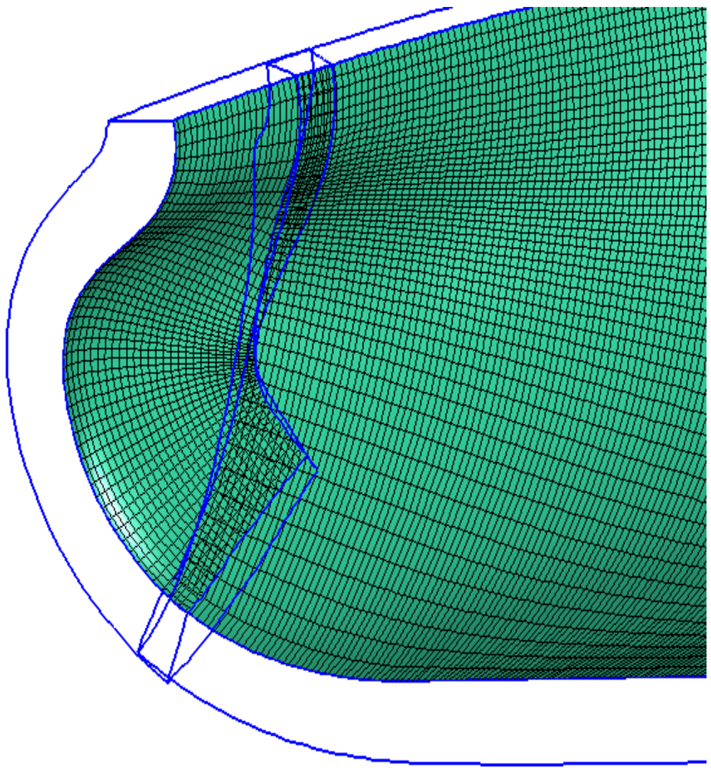
Manual correction of topology

Surface grid generation

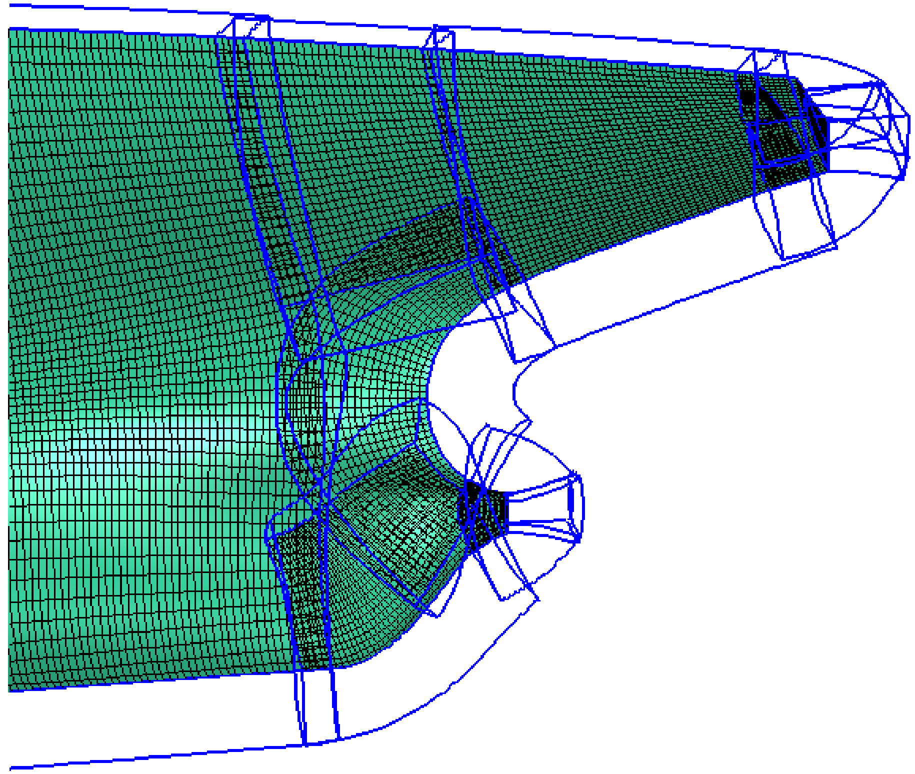


Volume grid generation



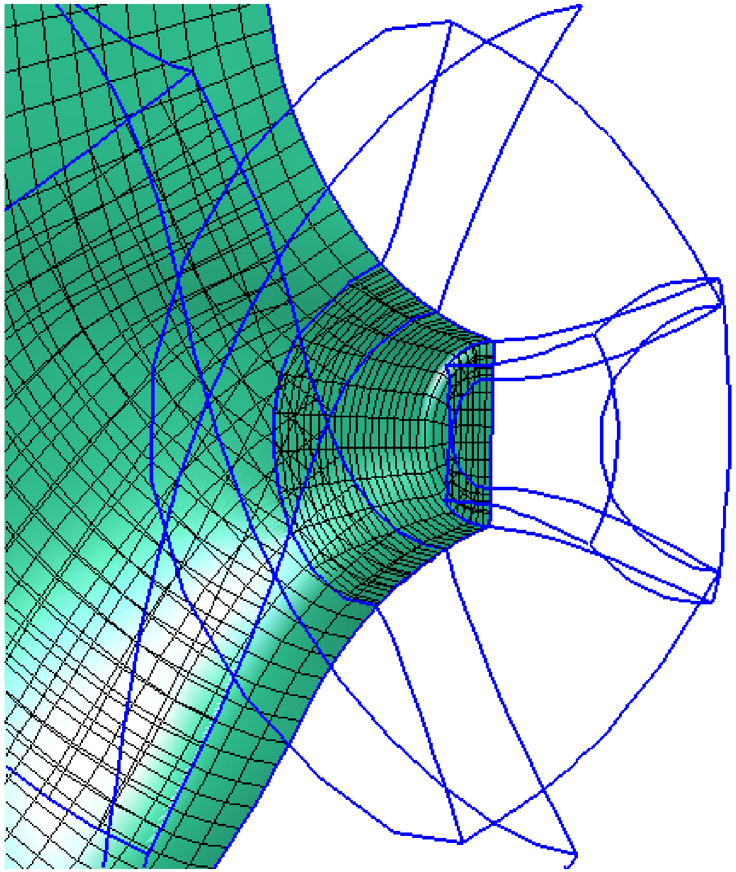


Bow

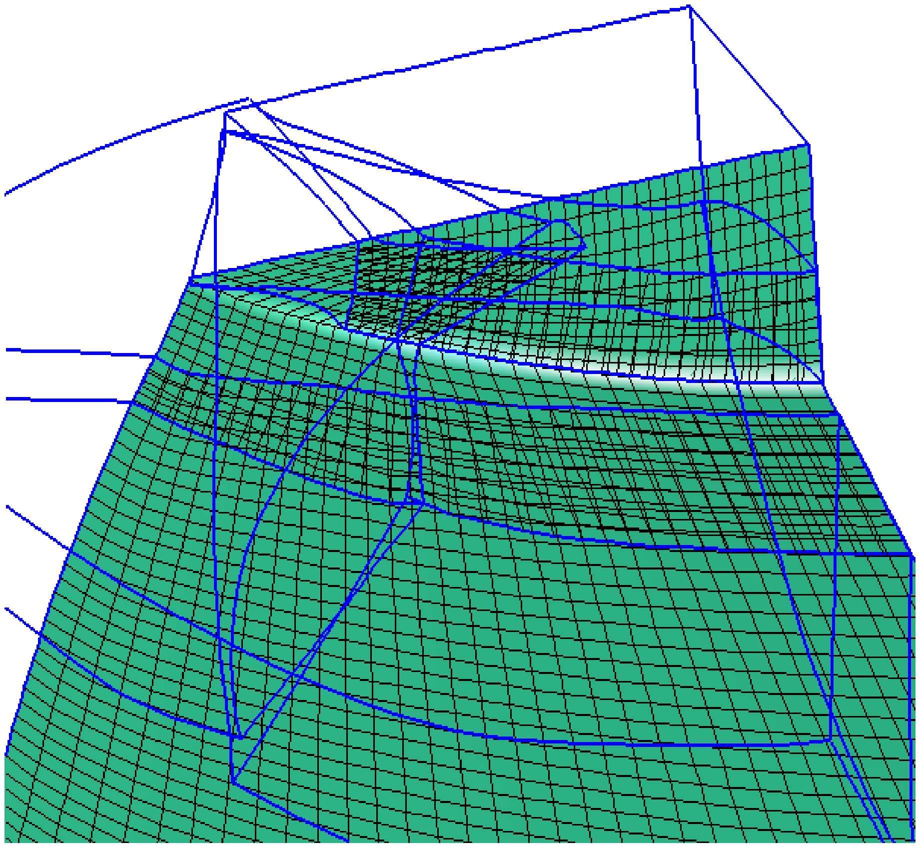


Stern

Volume grids



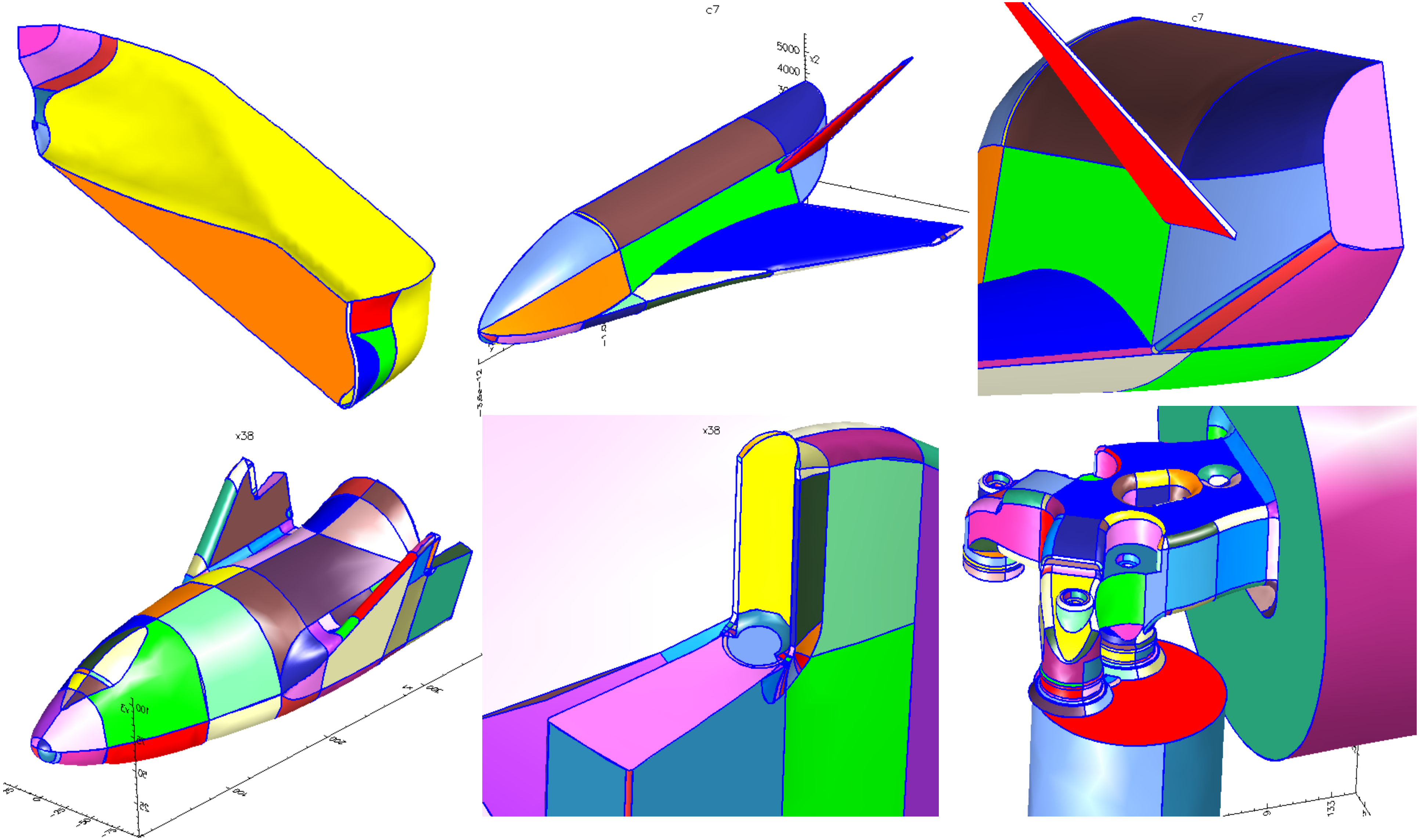
Propeller shaft



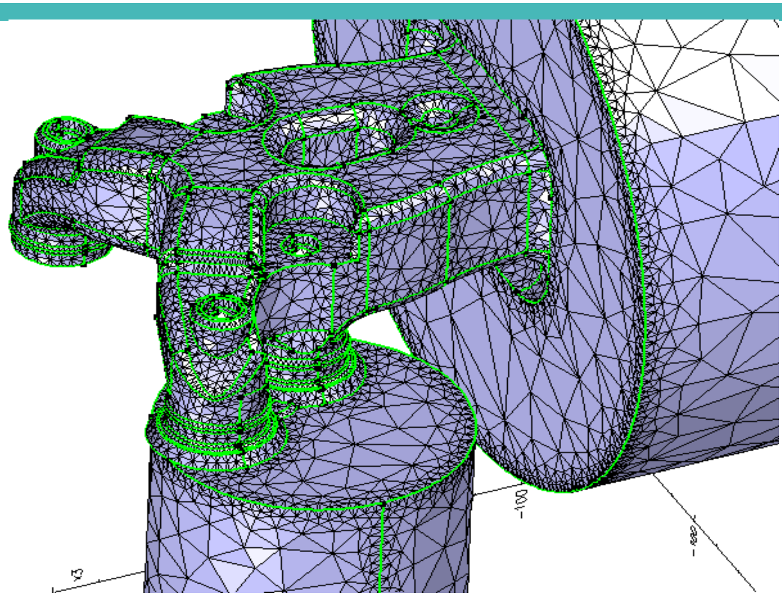
Transom stern

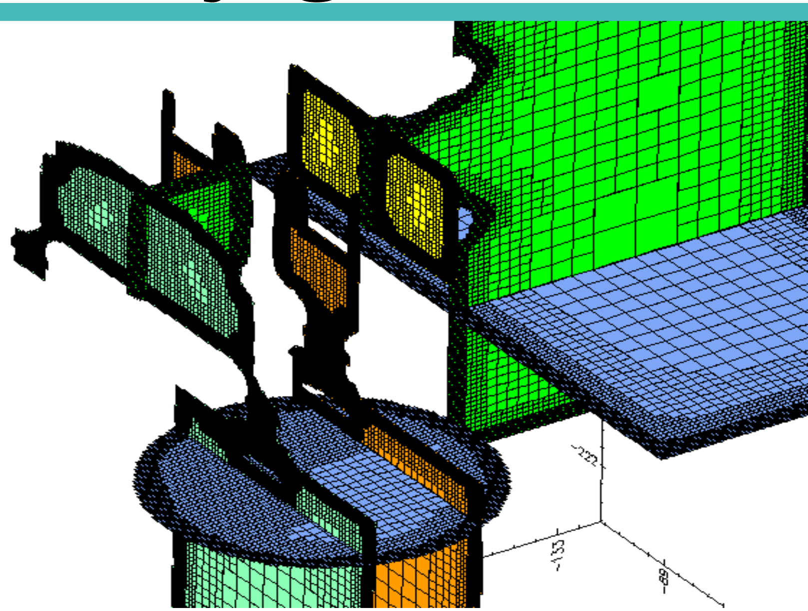
Geometrical complexity

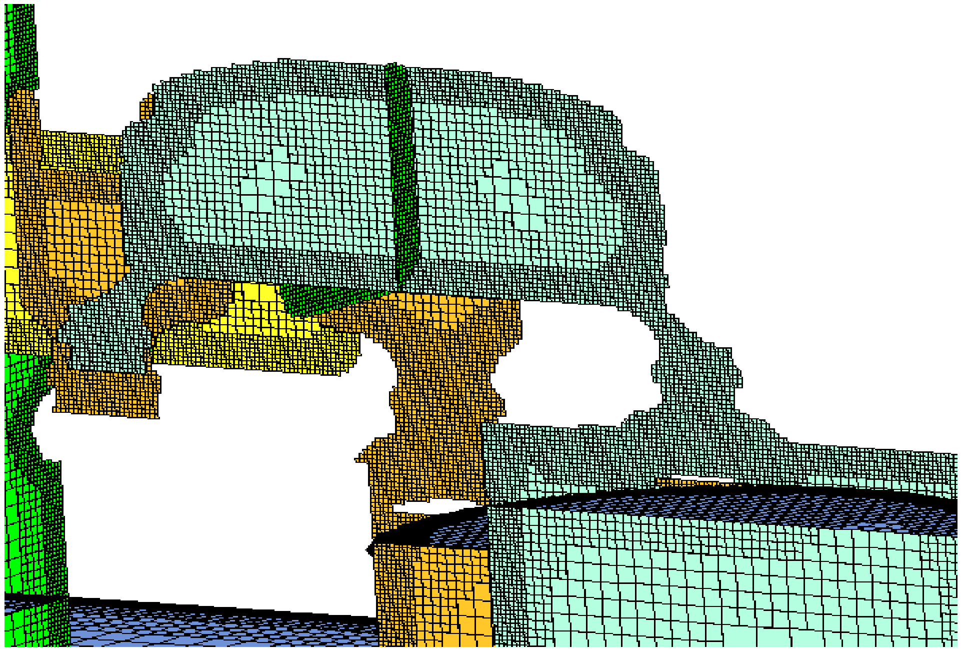
(As the complexity increases, it gets harder and harder to make a body fitted grid)

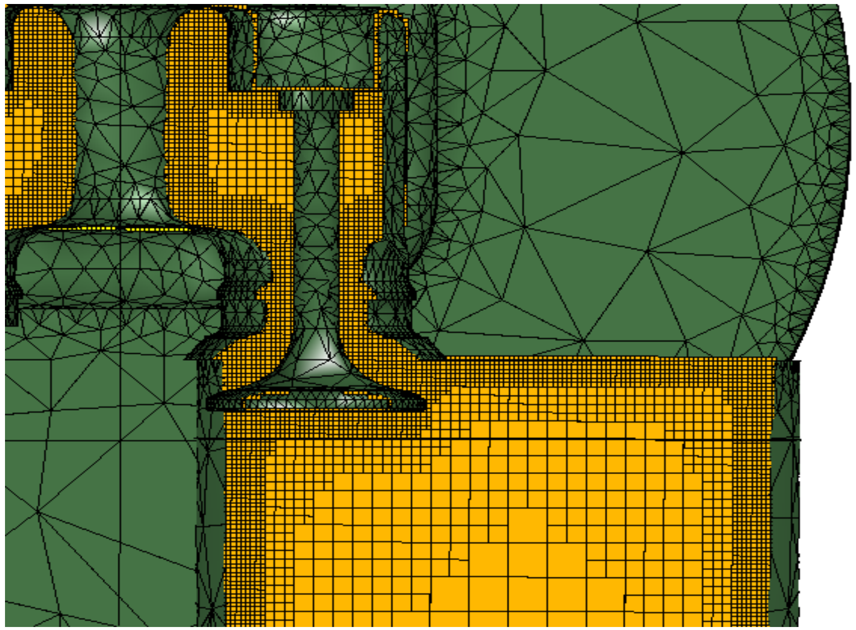


Embedded boundary grids









Concluding remarks

- Model read through neutral file format (IGES) or made internally.
- Tools are provided for correcting trimming curve errors
- Modifying geometry by adding/deleting/modifying patches
- Topology computed using edge matching algorithm
- Surface triangulation for fast projection onto CAD model
- Hyperbolic surface/volume grid generation
- Refine triangulation for more accurate representation
- Make embedded boundary grid with **cubes** from Cart3D
- Software freely available from www.llnl.gov/CASC/Overture