# LA-UR-23-20857

**Approved for public release; distribution is unlimited.**

**Title:** CrossLink - General Overview

**Author(s):** Drayna, Travis William
Lang, Laura M.
Long, Louie Joseph
Stam, Henry Russell

**Intended for:** Public release of recent software advancements.

**Issued:** 2023-01-30

# CrossLink

## General Overview

Travis Drayna
Laura Lang
Louie Long
Henry Stam

January 19, 2023

# **CrossLink:** An Overview

# Problem & Solution

## Problem

Traditional mesh generation approaches are labor intensive and have limited robustness when applied to parametric design exploration and optimization of complex geometries. While automatic mesh generation approaches exist, they tend to generate tetrahedral or mixed-hybrid meshes which are generally unsuitable for physics applications with strong shock waves, thin boundary layers, and strong gradients. In addition, simulations sizes in the billions of cells are becoming more common with traditional mesh generation methods quickly reaching scalability limits.

## Solution

CrossLink offers a topology-based mesh generation approach with unstructured block-filling methods and a scalable mesh generation engine. In addition, CrossLink incorporates a python-based API for seamless workflow integration and robust repeatability of the geometry handling and mesh generation process. This makes it ideal for parametric design study and optimization of complex geometries. Finally, future versions of CrossLink will offer a parametric mesh capability that optimizes a high-order mesh and enables reconstruction of the final mesh in memory by the physics solver.

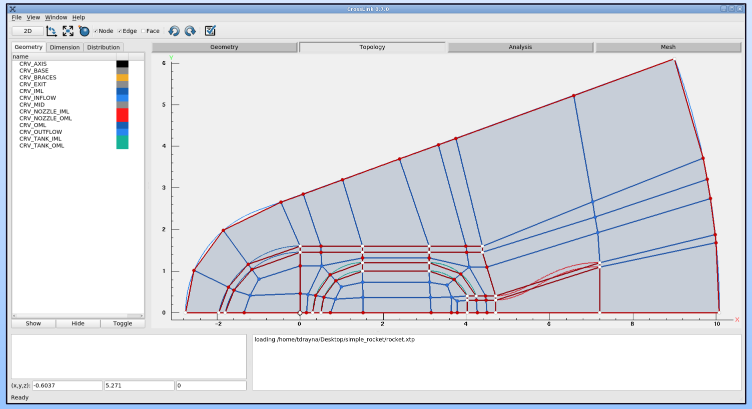
# CrossLink – Scalable 2D/3D Topology-based Meshing

## Geometry

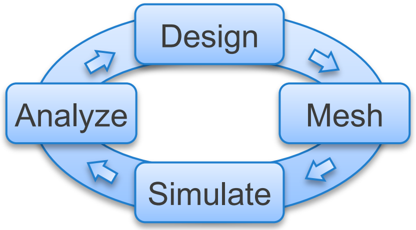
### xGeom
- Python API
- NURBS model
- 2D/3D geometry
- Flexible I/O

### xCAD
- Python API
- Native CAD driver
- Creo Parametric
- C++ toolkit

## CrossLink – GUI



### xMesh – Mesh Engine
- Topology-based mesh engine
- Parallel mesh optimization




## Multi-physics Code

### Mesh File

### Mesh Optimizer
- Improved simulation robustness
- Improved mesh quality control

### ParMesh API
- C++ library
- Reconstruct mesh in memory
- Scalable mesh size
- Simplified startup and partitioning
- Supports linear and curved elements
- Allows for structured memory layout

# CrossLink – Main Components

## Geometry

**XGEOM**
- → Python scripting API
- → NURBS curves & surfaces
- → Conceptual design

**XCAD**
- → Python API
- → CAD model driver
- → NURBS geometry access
- → Headless (no GUI)

**XGUI**
- → Graphical user interface
- → Organized into workbenches
- → Load geometry
- → Build topology
- → Define meshing
- → Specify tags
- → Inspect mesh

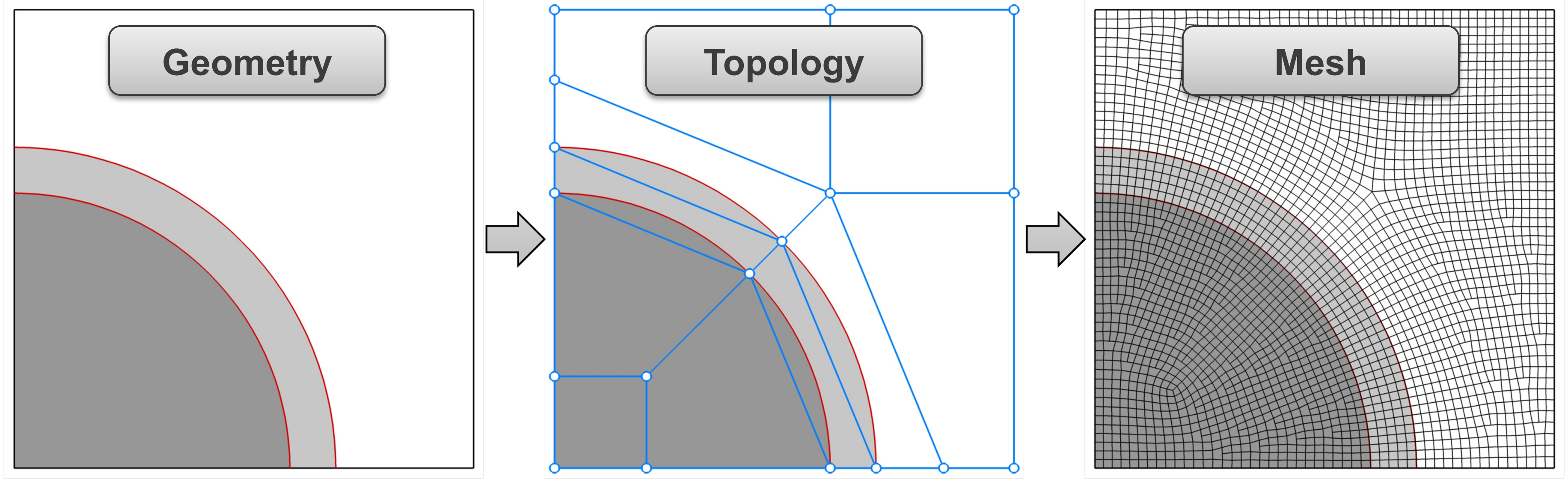
**XMESH**
- → Python API
- → Mesh generation engine
- → OpenMP parallelism
- → Loads topology/geometry
- → Initializes mesh
- → Optimizes mesh
- → Exports mesh

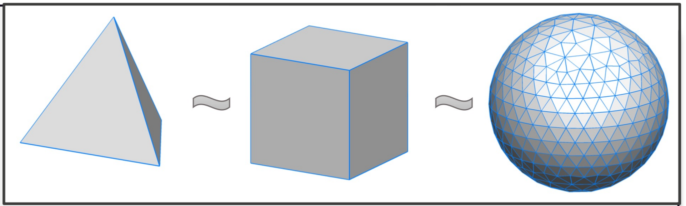
## Physics Codes

**Mesh Optimizer**
- → Target-matrix optimizer
- → Supports smooth mesh deformations
- → Preserves spacings of initial mesh

**XPAR**
- → Mesh reconstruction library
- → Scalable mesh size
- → Low-order or high-order elements

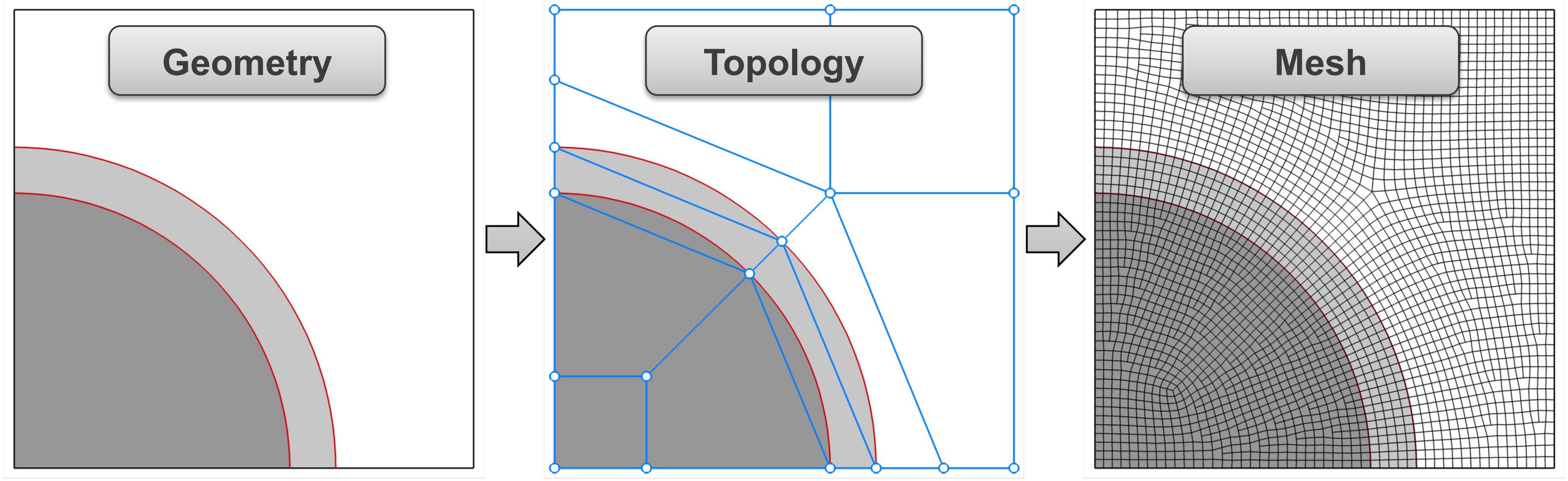
# What is topology-based meshing?



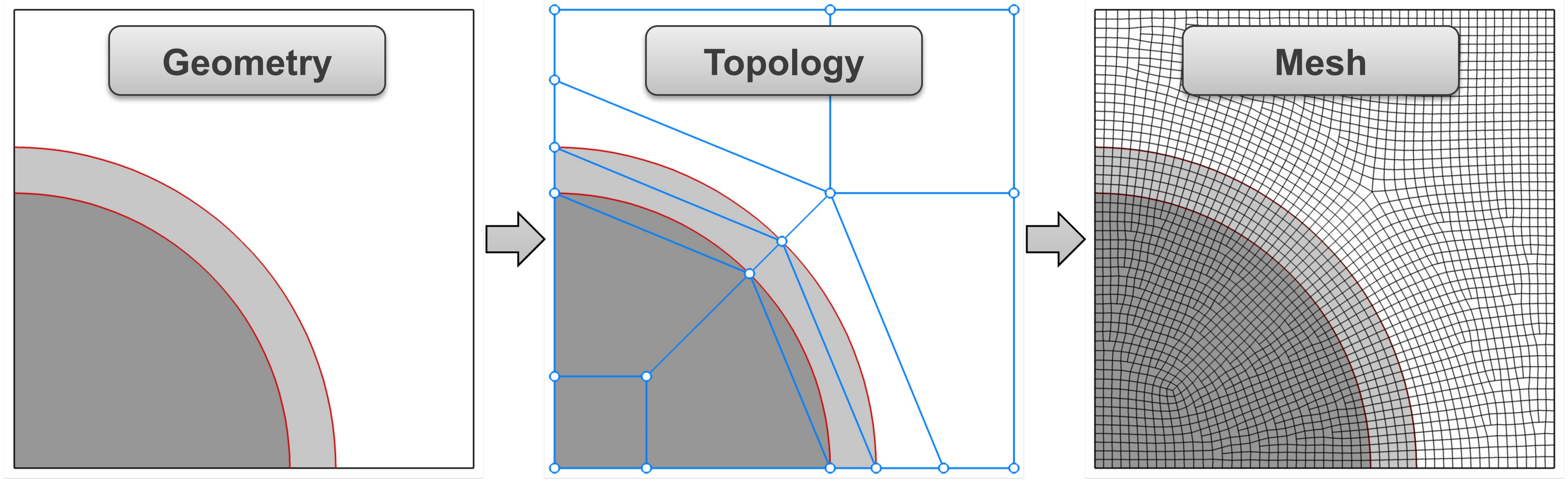
- A computational mesh is essential for many design and analysis workflows.
- Many design and analysis workflows focus on *families* of similar geometries.
- These families are often topologically similar (i.e., same Euler characteristic).
- Topologically similar geometries can often be spatially decomposed in the same way.
- A mesh topology (nodes, edges, faces, and blocks) can be used to define the spatial decomposition.
- Geometry constraints and meshing rules can be added to the mesh topology to control the final mesh.
- A mesh generation algorithm performs the initialization and optimization of the final mesh.

# What is the meshing process?

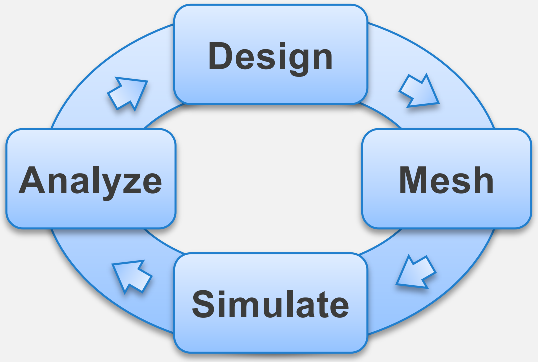



1. Starting with geometry, geometry groups are used to define hard internal and external boundaries.
2. A mesh topology (nodes, edges, faces, and blocks) is created to decompose the spatial domain into regular entities.
3. Geometric constraints are imposed by assigning topological entities to geometry groups.
4. Mesh rules (e.g., dimension, distribution, spacing, feathering) are defined to control mesh quality.
5. Mesh tags (e.g., boundary, volume) are defined for the desired simulation (e.g., CFD, FEA).
6. The mesh is initialized, optimized, and exported to a mesh file.

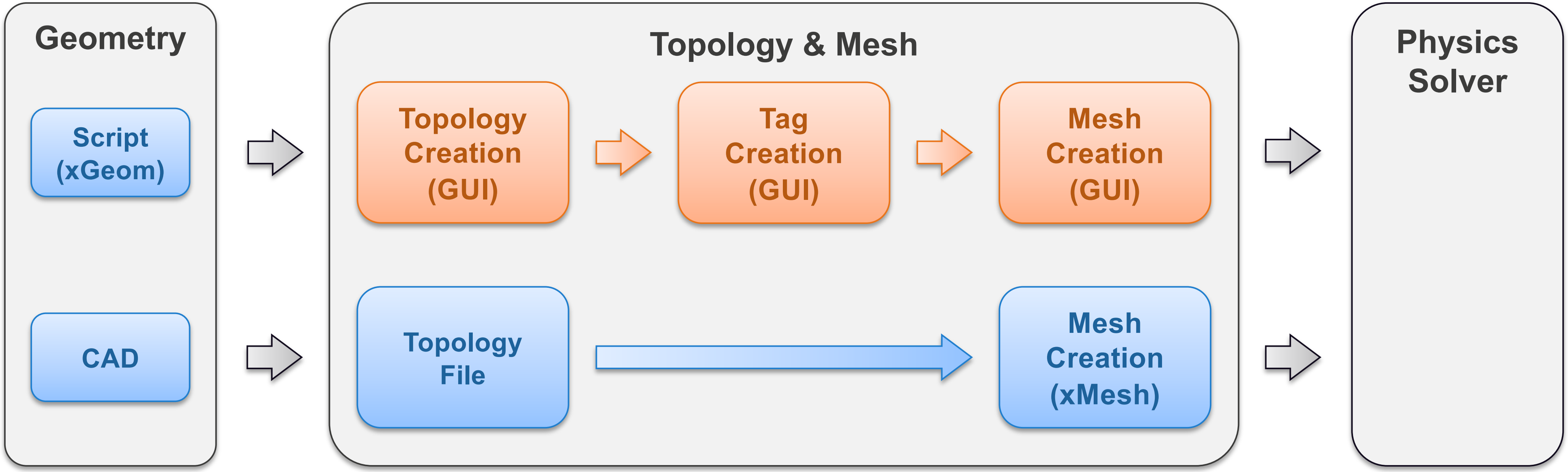
# What are the benefits of this approach?




- ✓ Simplifies the meshing process
- ✓ Reduces overall turnaround time
- ✓ Reduces user effort
- ✓ Supports legacy and modern geometry
- ✓ Produces high quality meshes
- ✓ Handles parametric changes

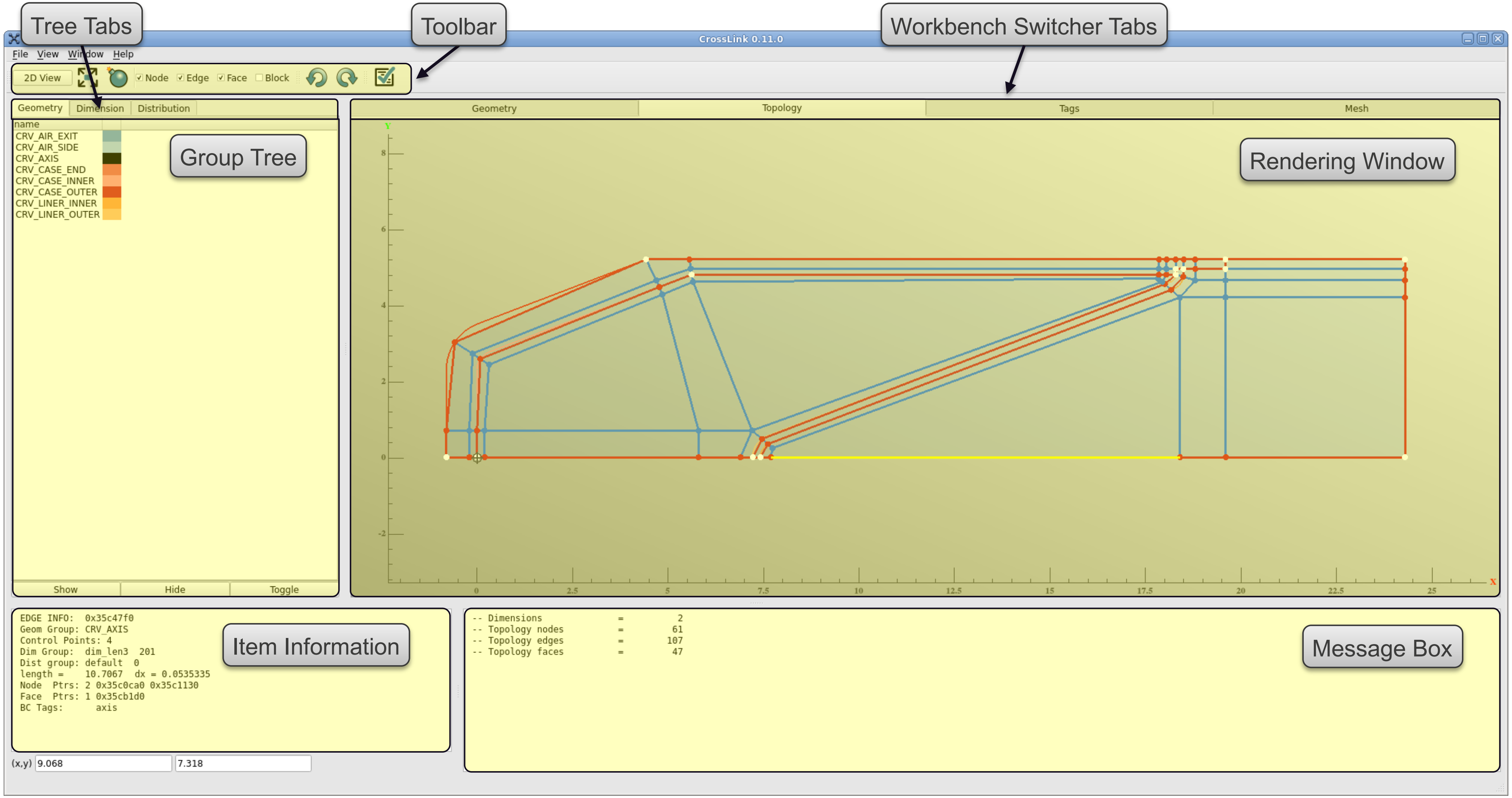


# General Workflow

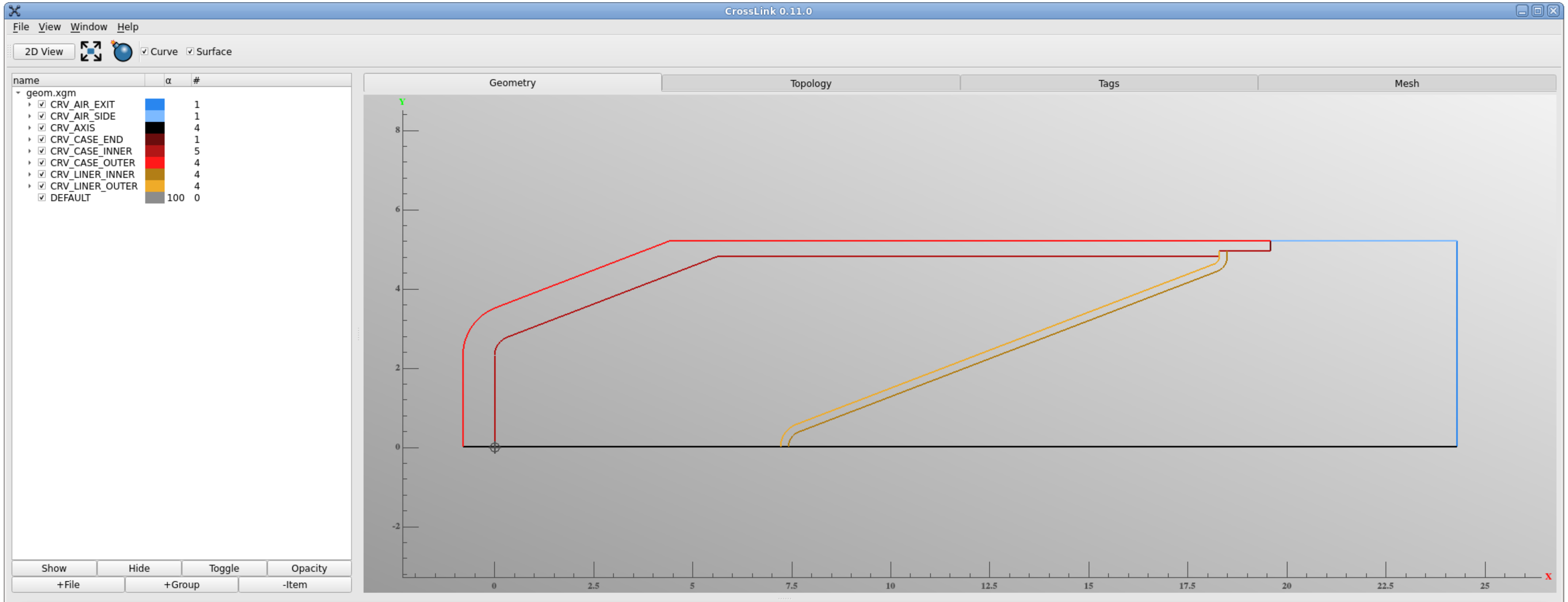



- Initial geometry is created via a geometry script or a CAD software package.
- A mesh topology is created interactively using CrossLink's Graphical User Interface (GUI).
- Geometry constraints, meshing rules, and mesh tags are defined and stored in the topology file.
- The xGeom and xMesh python API libraries are used to automate the mesh generation process.
- The workflow handles geometric parameter changes and robustly (re)meshes each design.

# Graphical User Interface

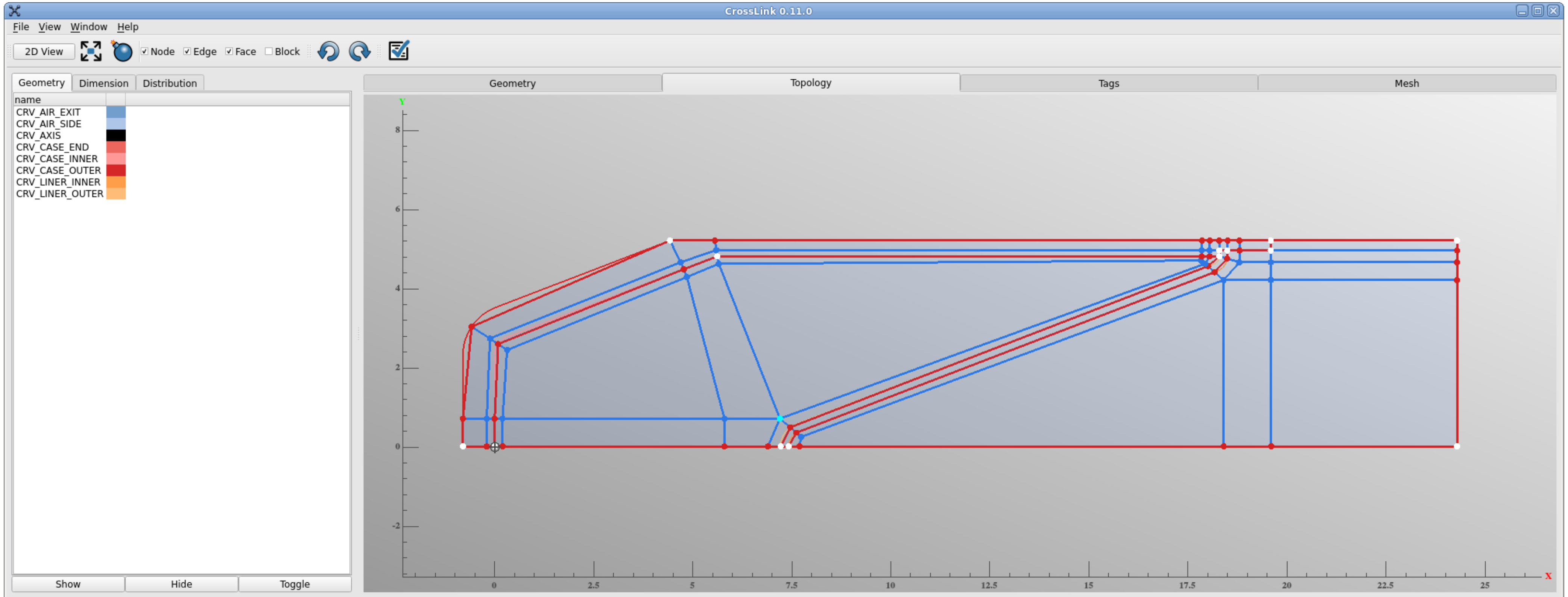


# Geometry Workbench



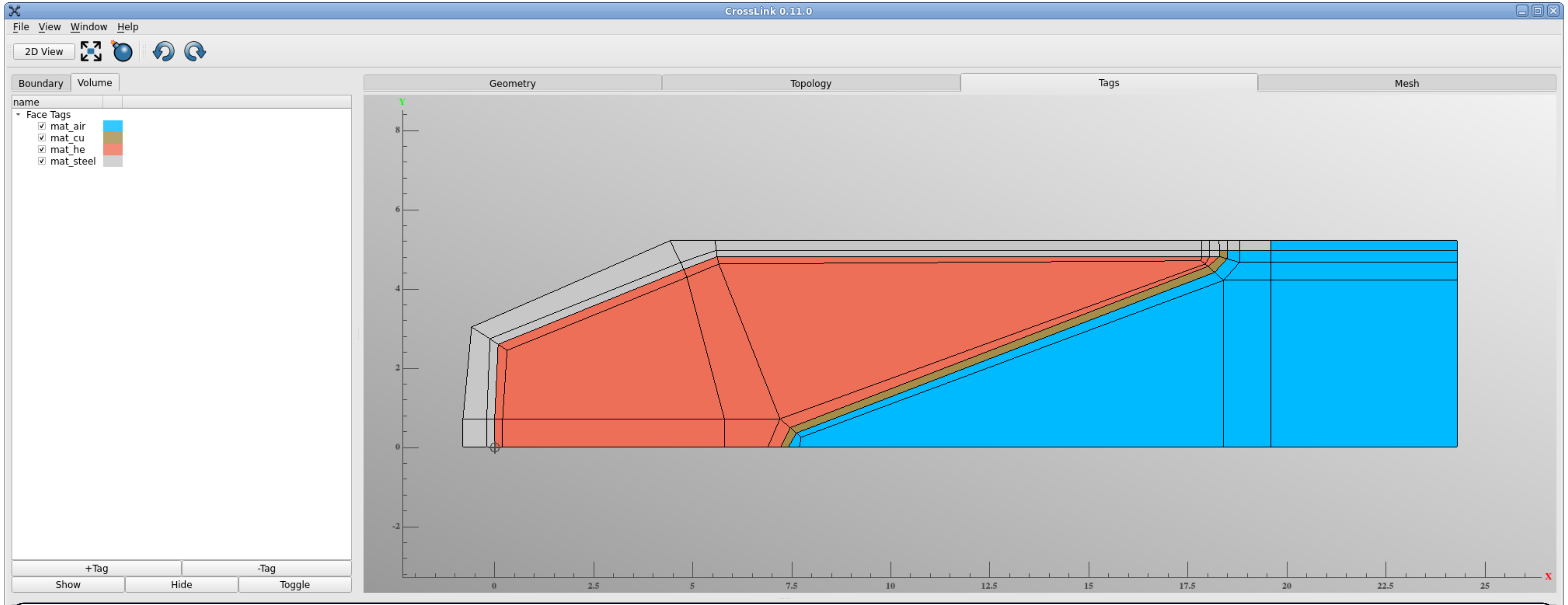

- The *Geometry* workbench is used to inspect the geometry.
- This workbench also supports geometry group modification.

# Topology Workbench



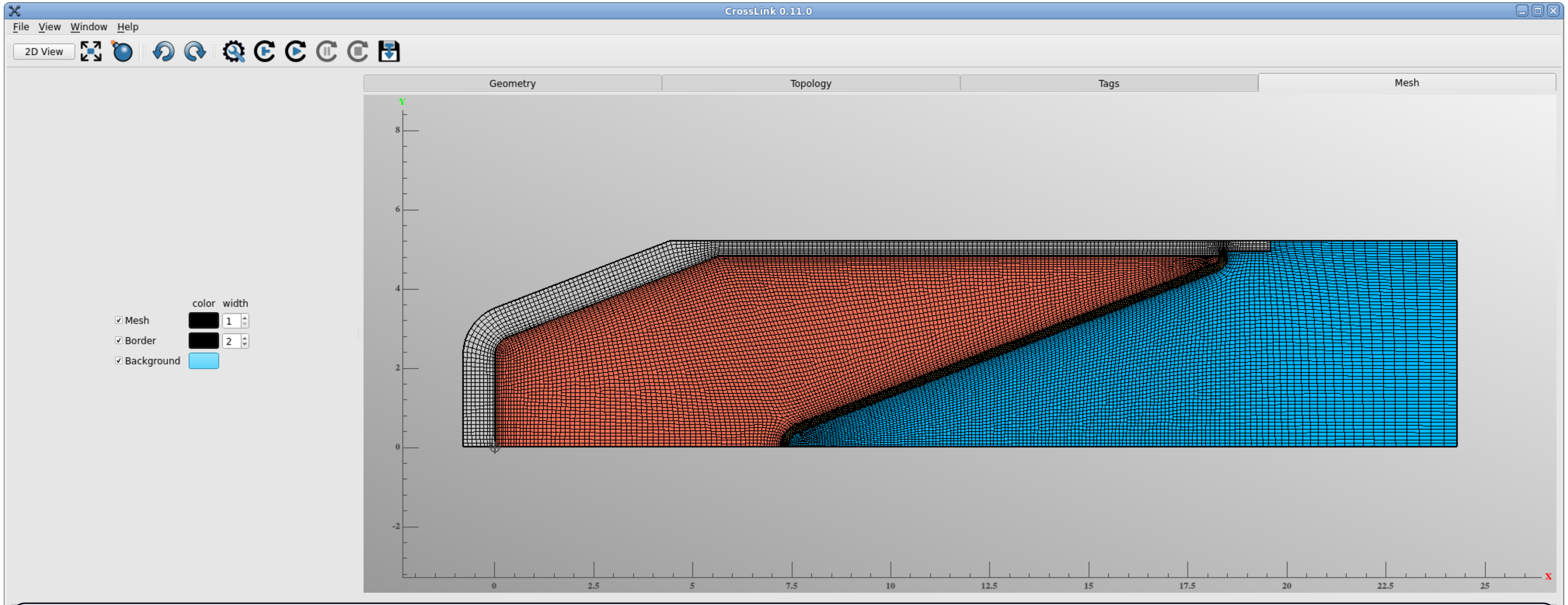

- The *Topology* workbench is used to create and modify the mesh topology.
- Geometry constraints and meshing rules are defined using this workbench.
- A topology validity check helps diagnose errors and issues with the topology.

# Tags Workbench




- The *Tags* workbench is used to create and modify mesh tags.
- Mesh tags are sets of points or cells that can be used by the analysis software.
- Mesh tags can be either boundary tags or volume tags.

# Mesh Workbench

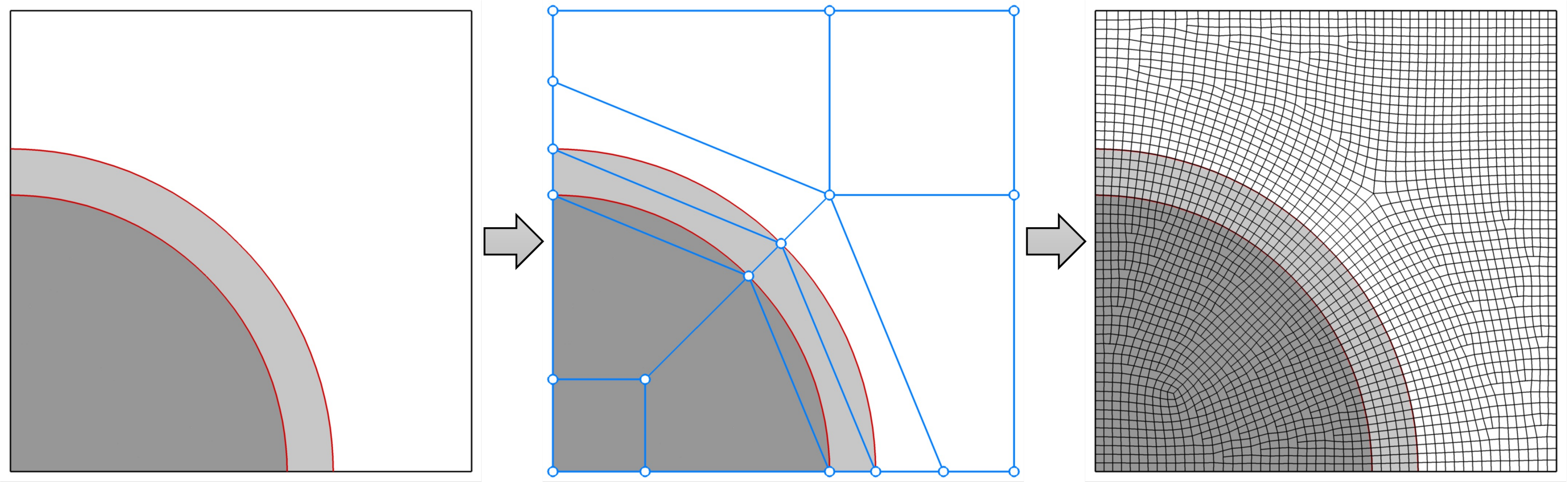



- The *Mesh* workbench is used to initialize, optimize, and export the computational mesh.
- The workbench allows users to inspect 2D and 3D meshes.
- The workbench offers interactive control of the mesh optimizer.

# **CrossLink:** Mesh Generation

# Mesh Generation Steps

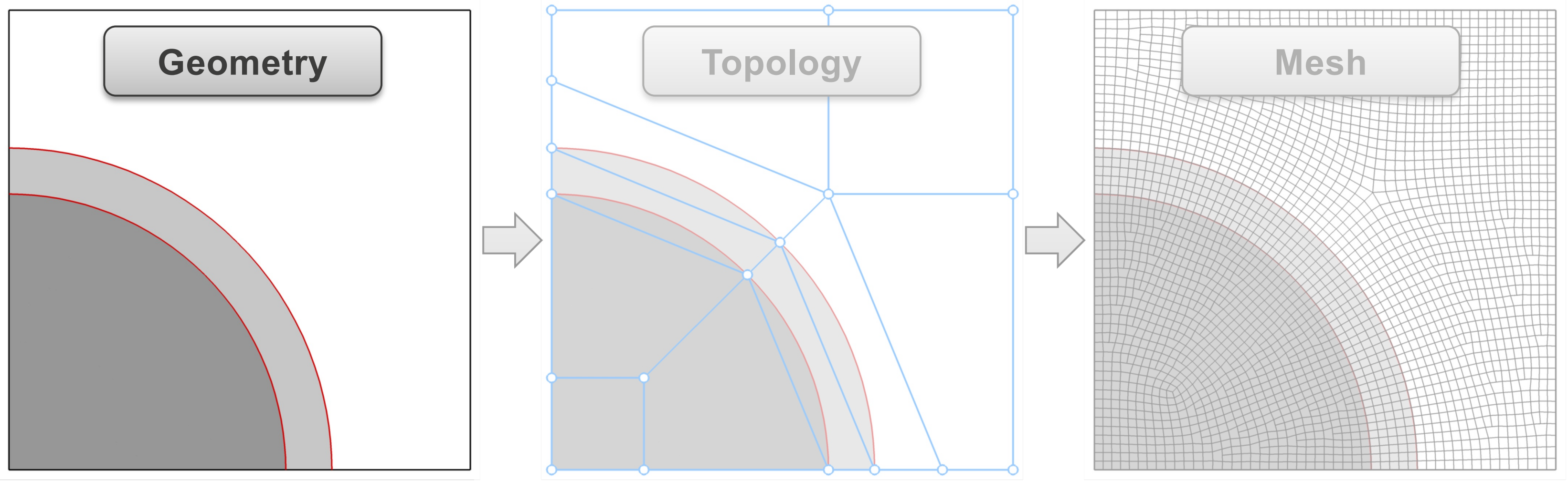

Interactive topology creation



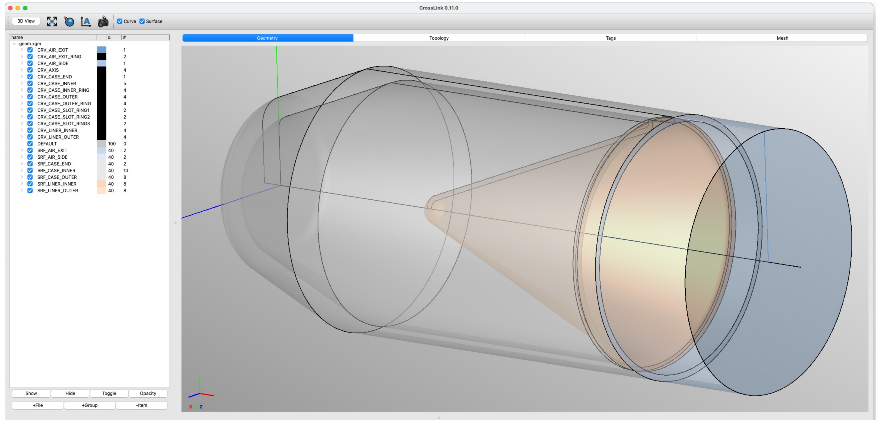
Meshing workflow

# Geometry Sources

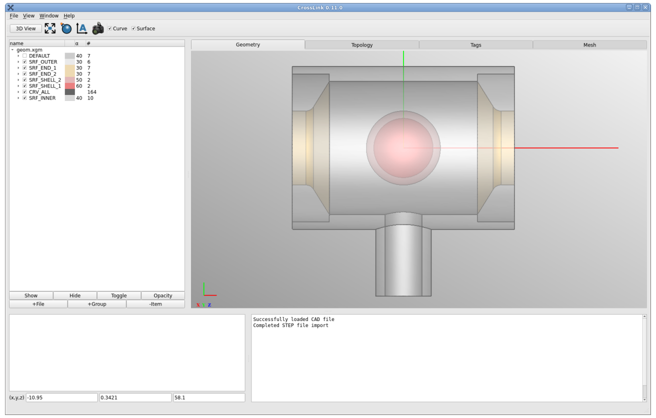
- CrossLink requires a geometric model to constrain the final mesh.



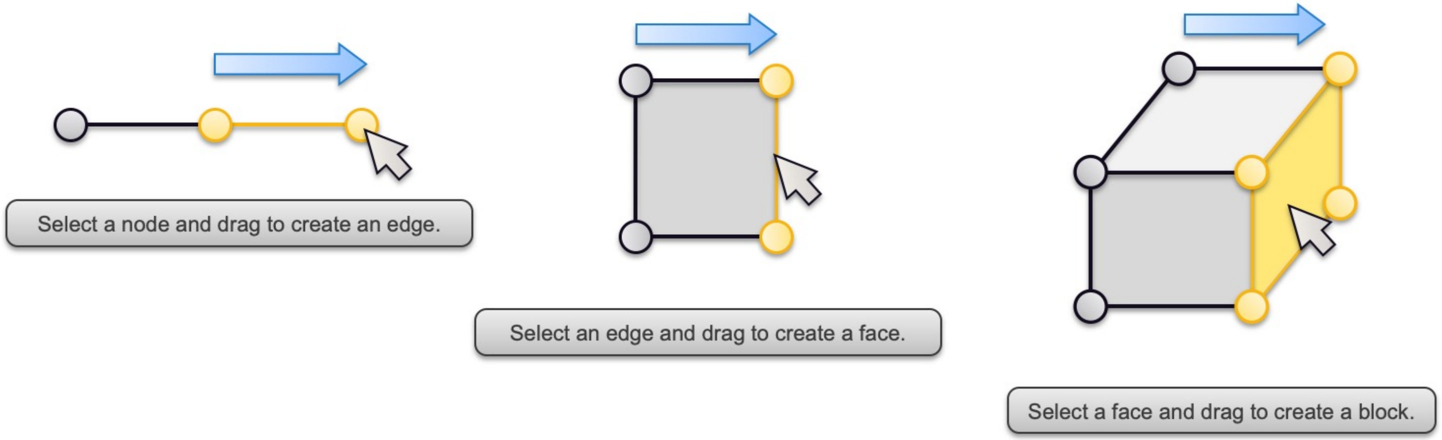

- Supported geometry sources/types include:



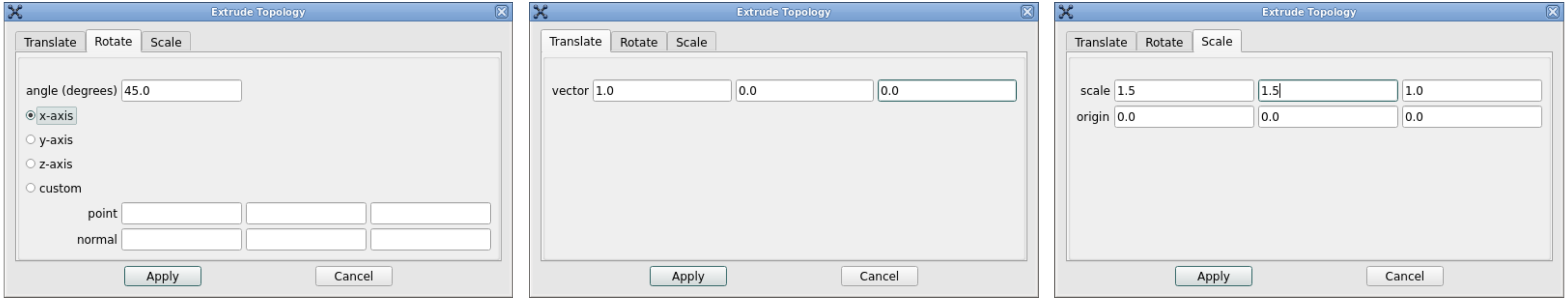
| Source | Type |
| --- | --- |
| xGeom, CAD | NURBS curve |
| xGeom, CAD | NURBS surface |
| User Code | Tabular curve |
| User Code | Tabular surface |



# Topology Creation Modes

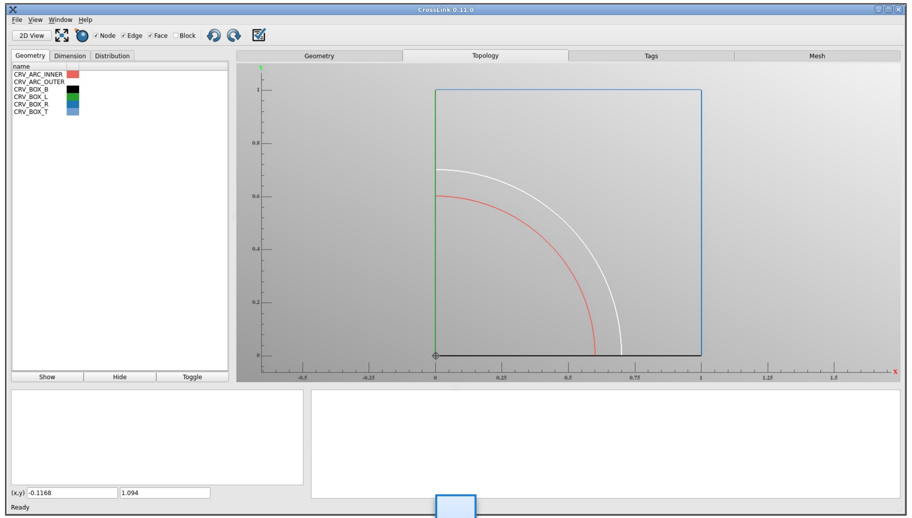



Mouse-driven topology creation

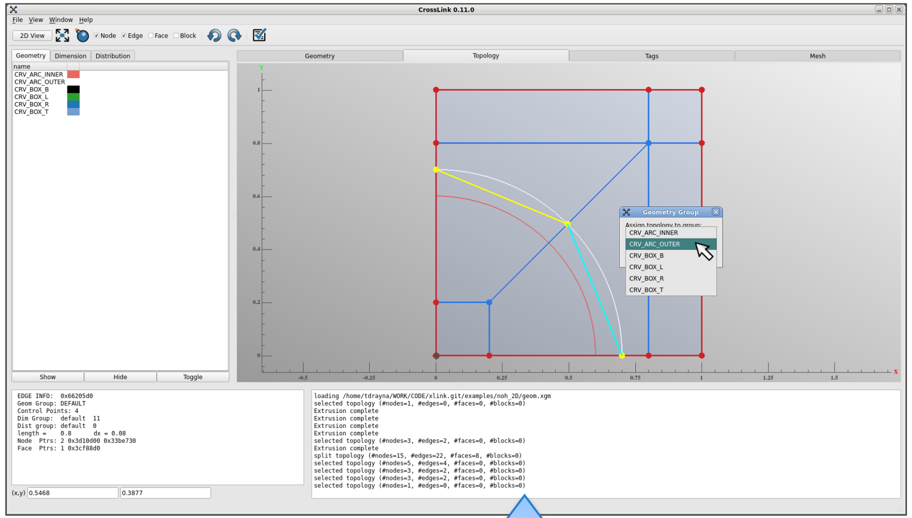



Menu-driven topology creation

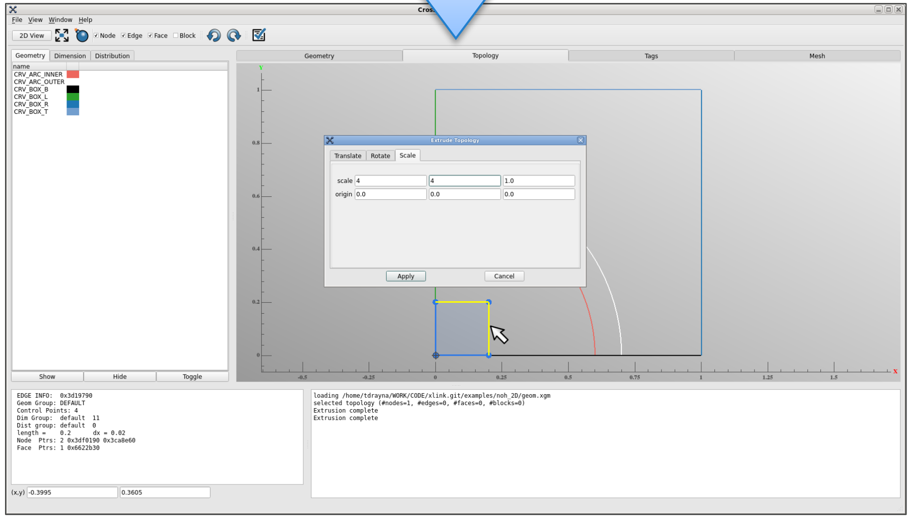
# Building Topology



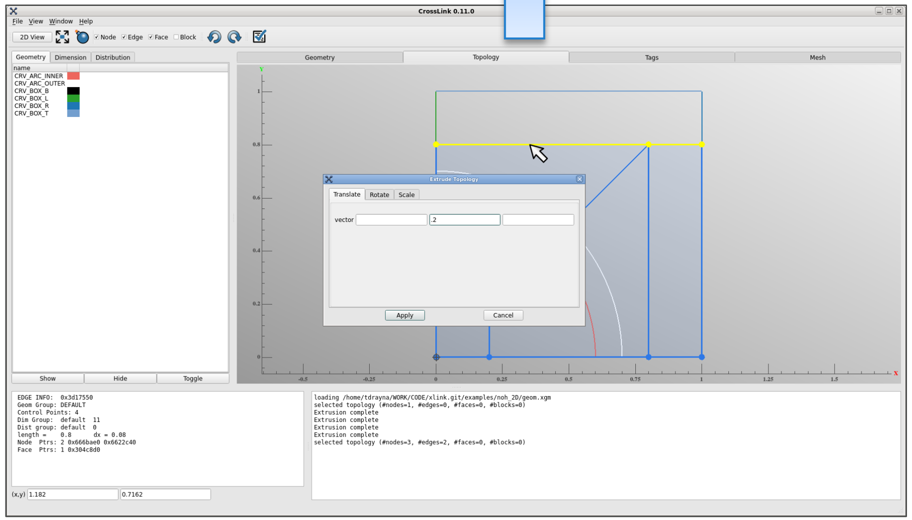
Import geometry



Set constraints

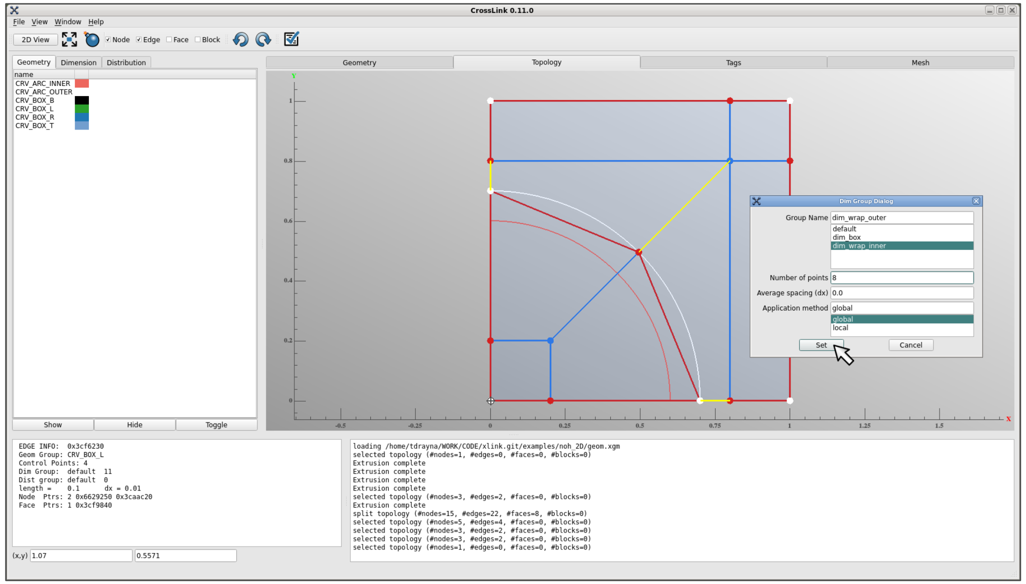


Extrude topology (scale)

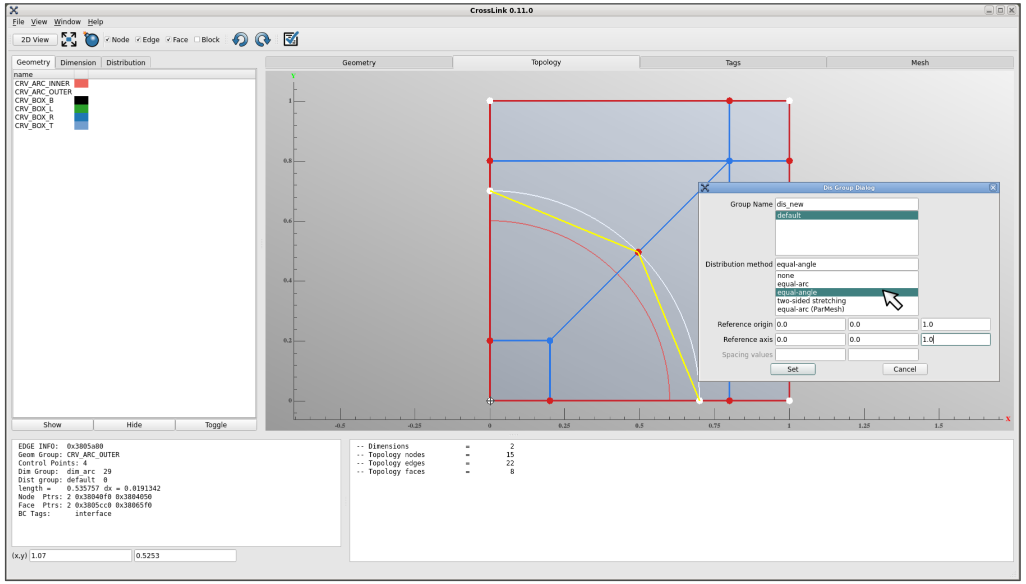


Extrude topology (translate)

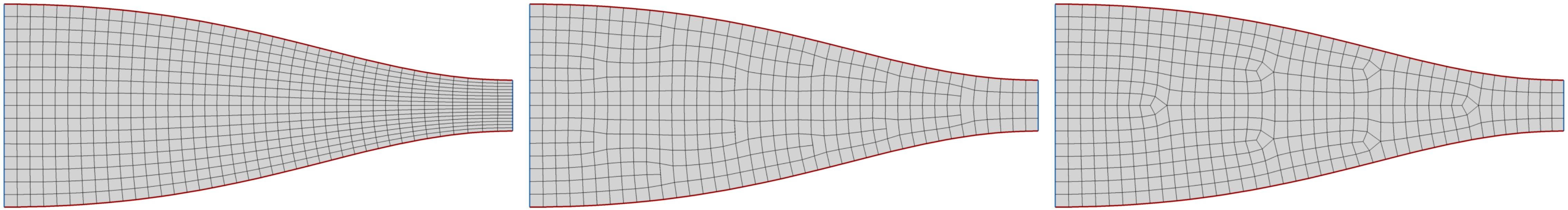
# Mesh Control



Dimension Groups

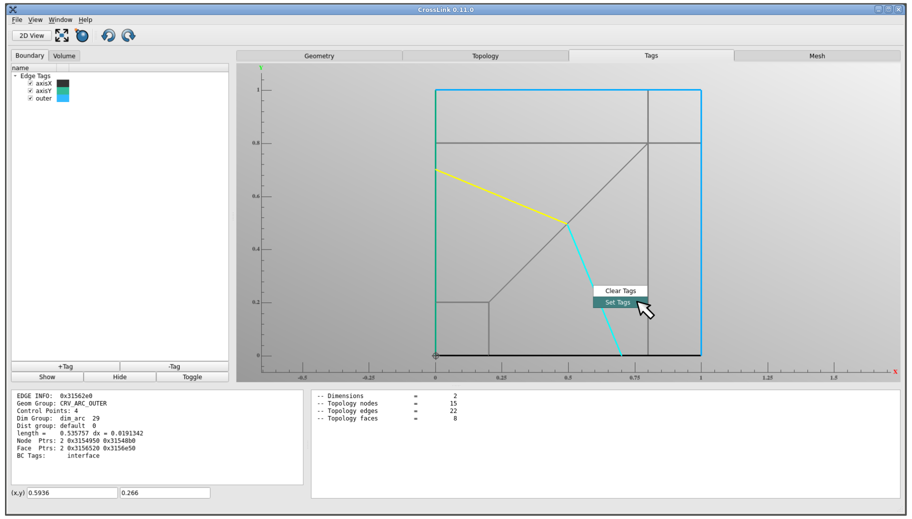


Distribution Groups

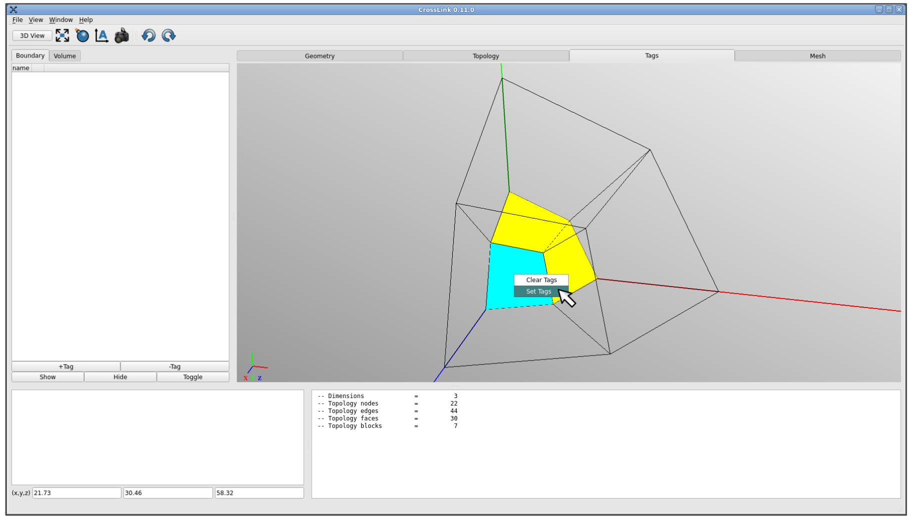


Refinement Patterns

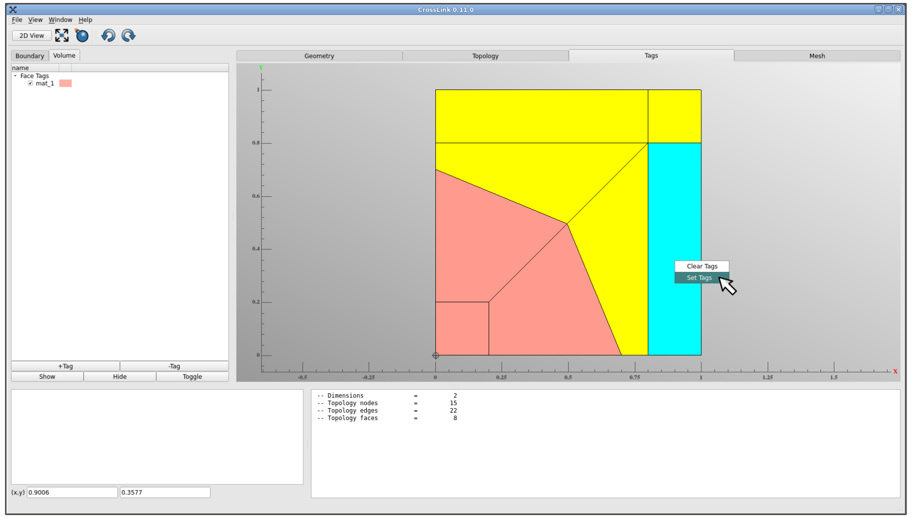
# Setting Mesh Tags



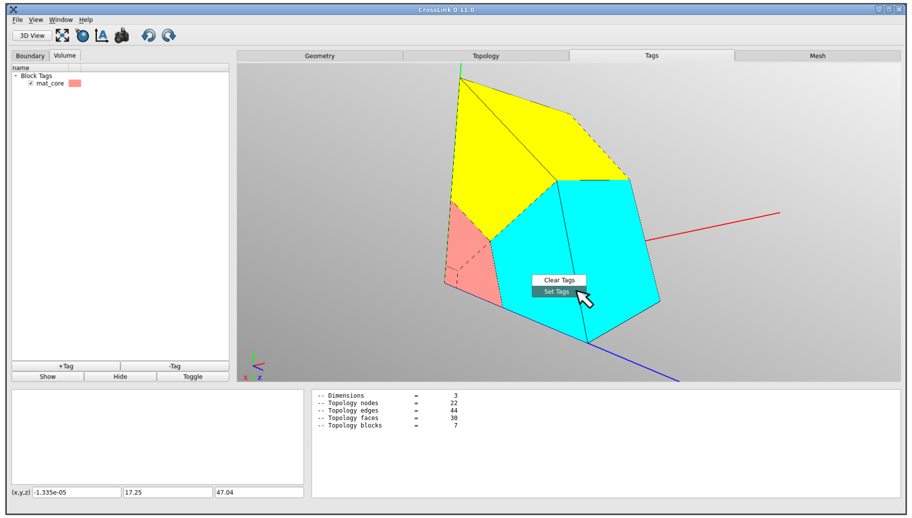
Boundary Tag (2D)



Boundary Tag (3D)

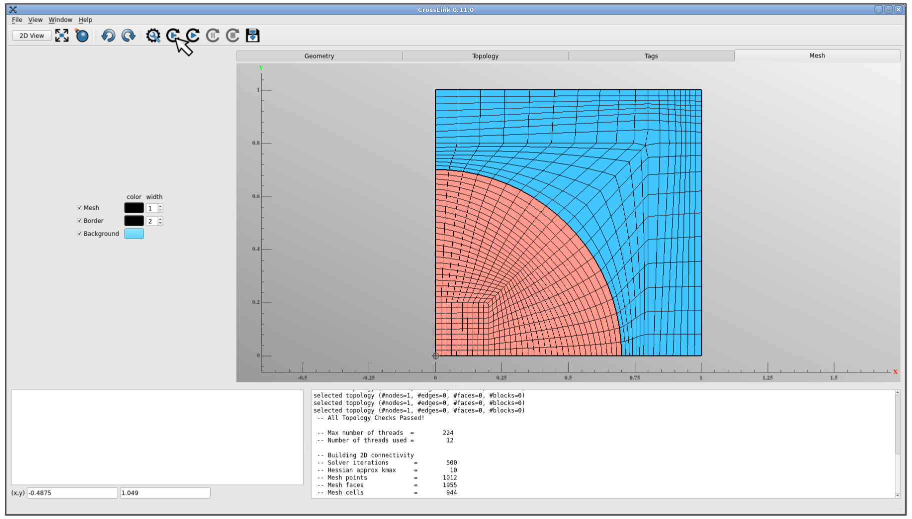


Volume Tag (2D)

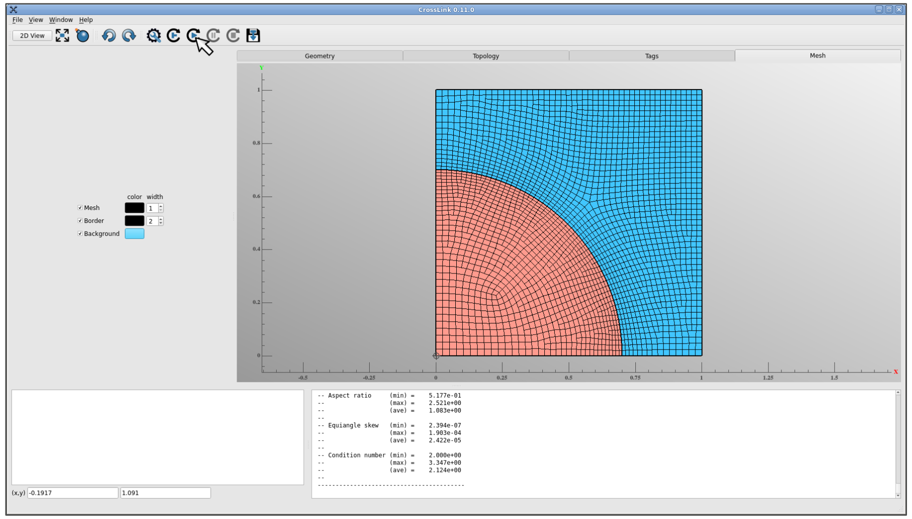


Volume Tag (3D)

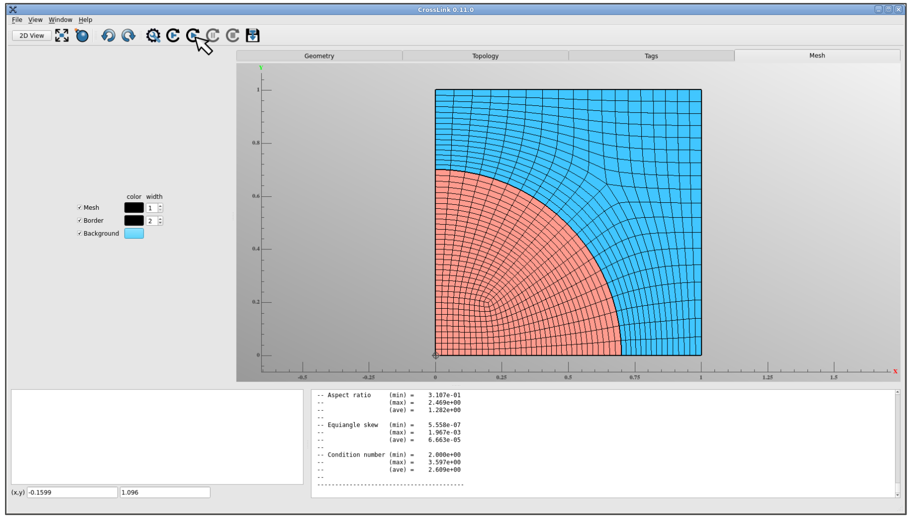
# Generating the Mesh



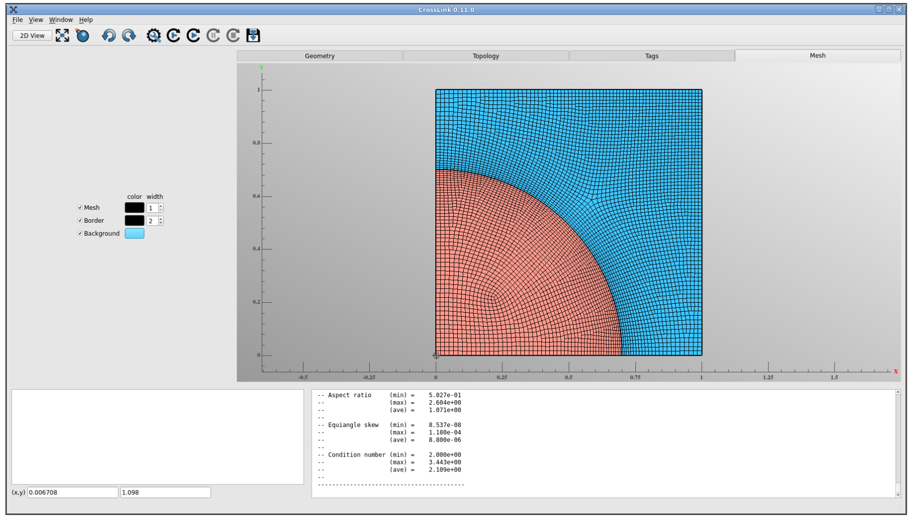
Initial Mesh



Optimized with feathering



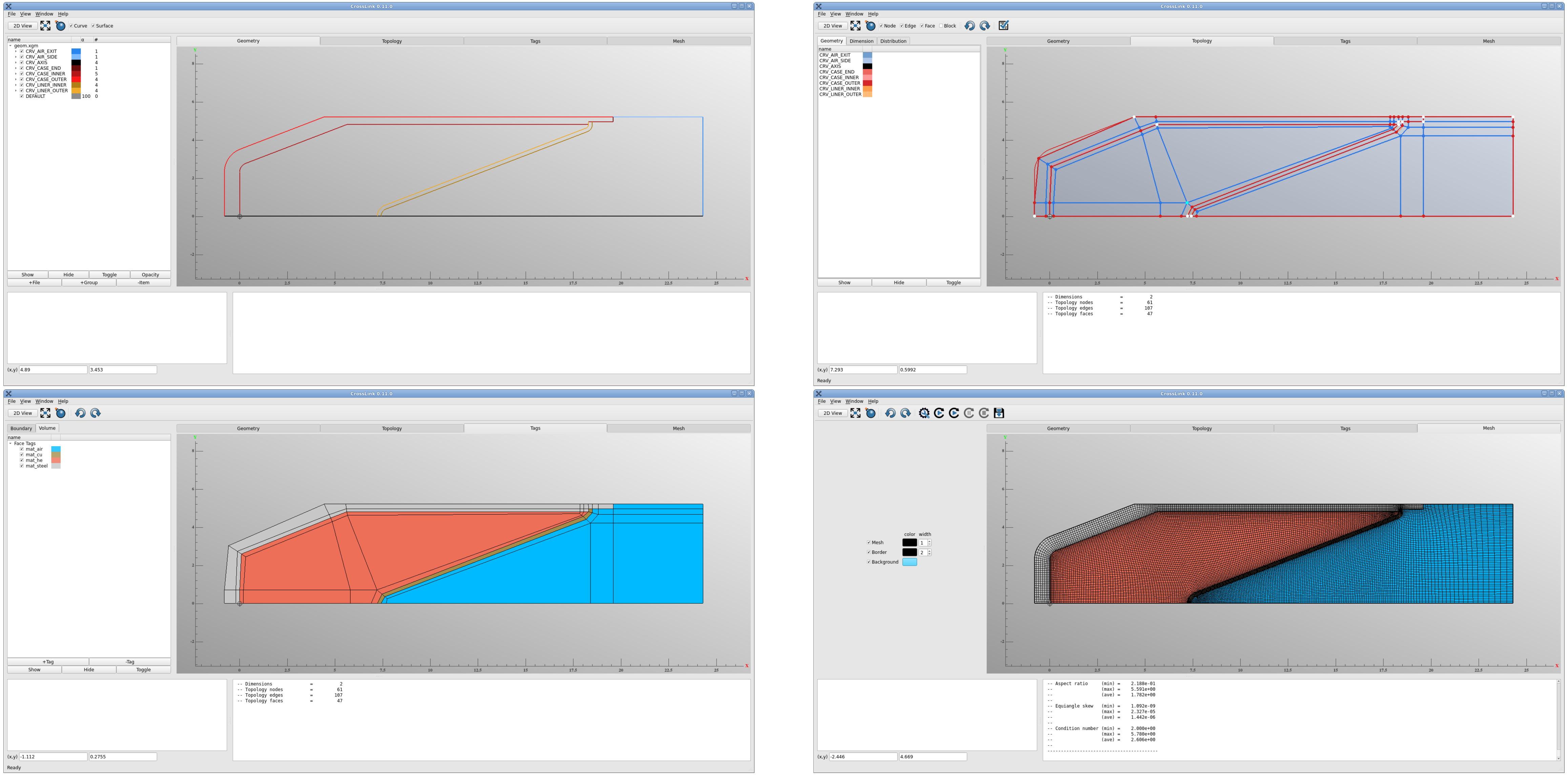
Optimized Mesh



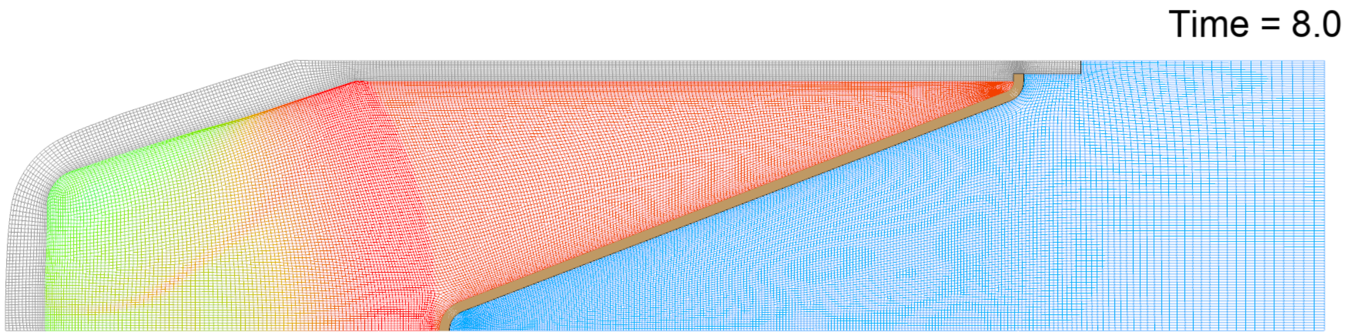
Optimized with feathering and 1.5x resolution

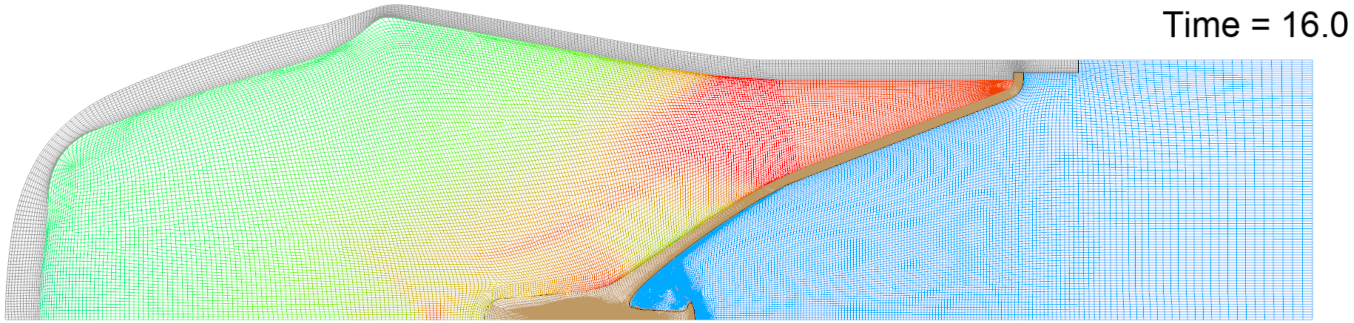
**Example:** 2D Shaped Charge

# Shaped Charge (2D)



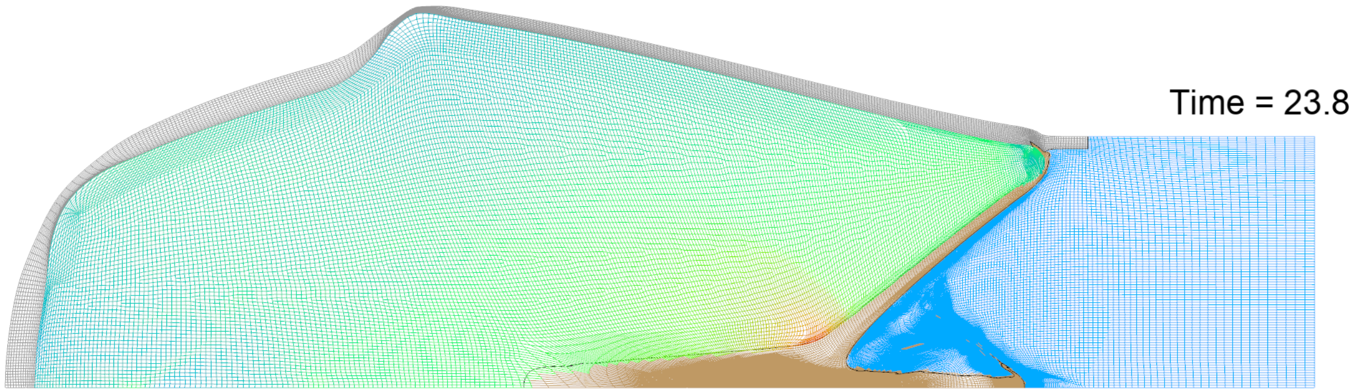
# Shaped Charge (2D)



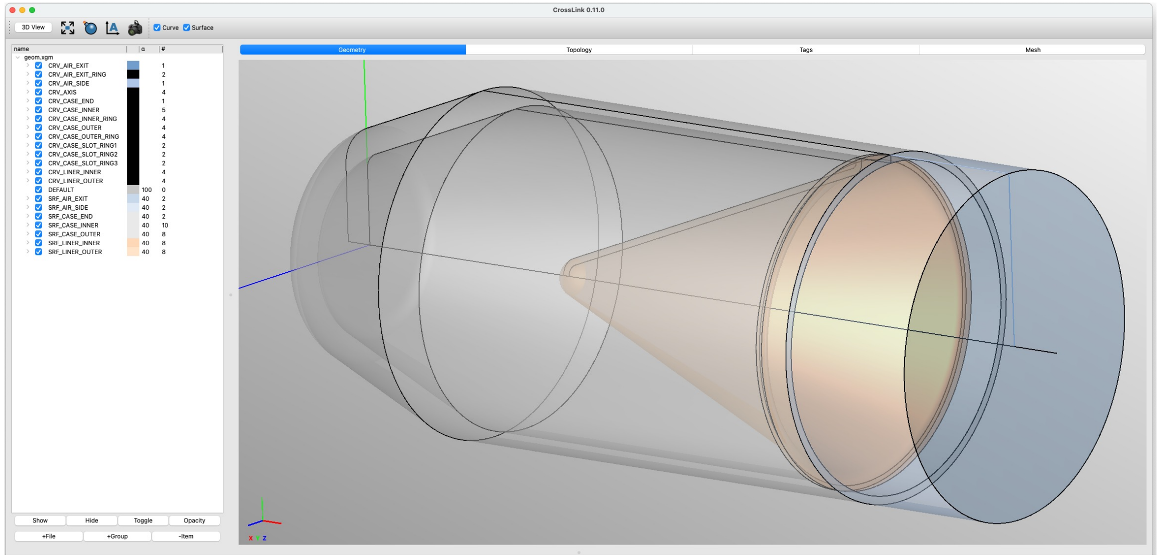

**FLAG simulation**

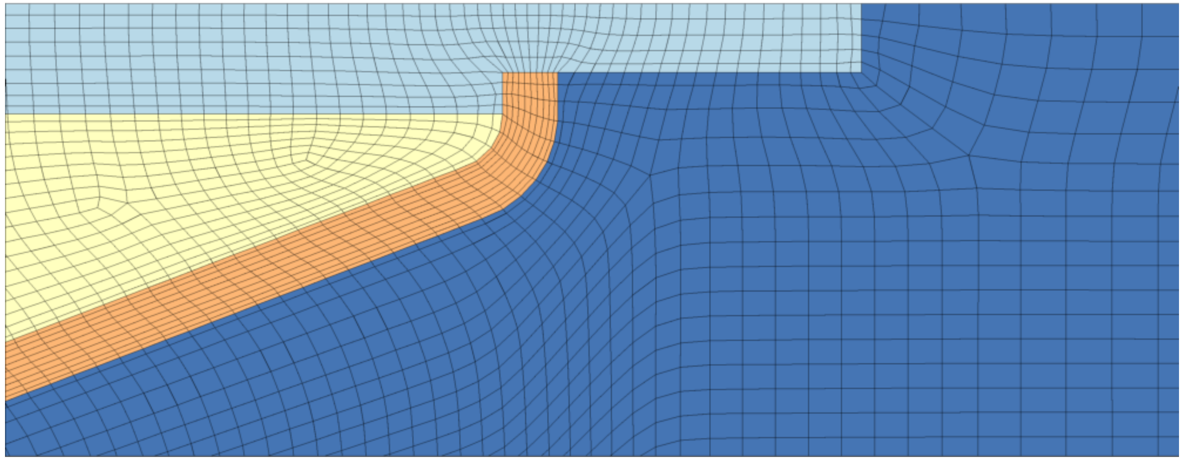
→ Axisymmetric mesh
→ 43k cells
→ Steel case, copper liner, HE, air
→ ALE activated at t = 8.0

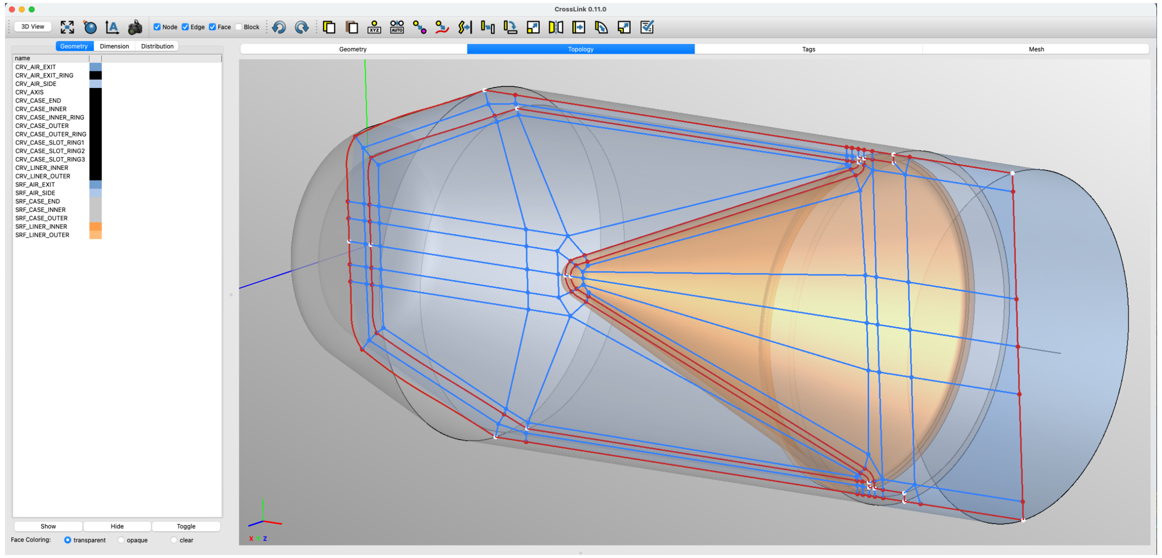


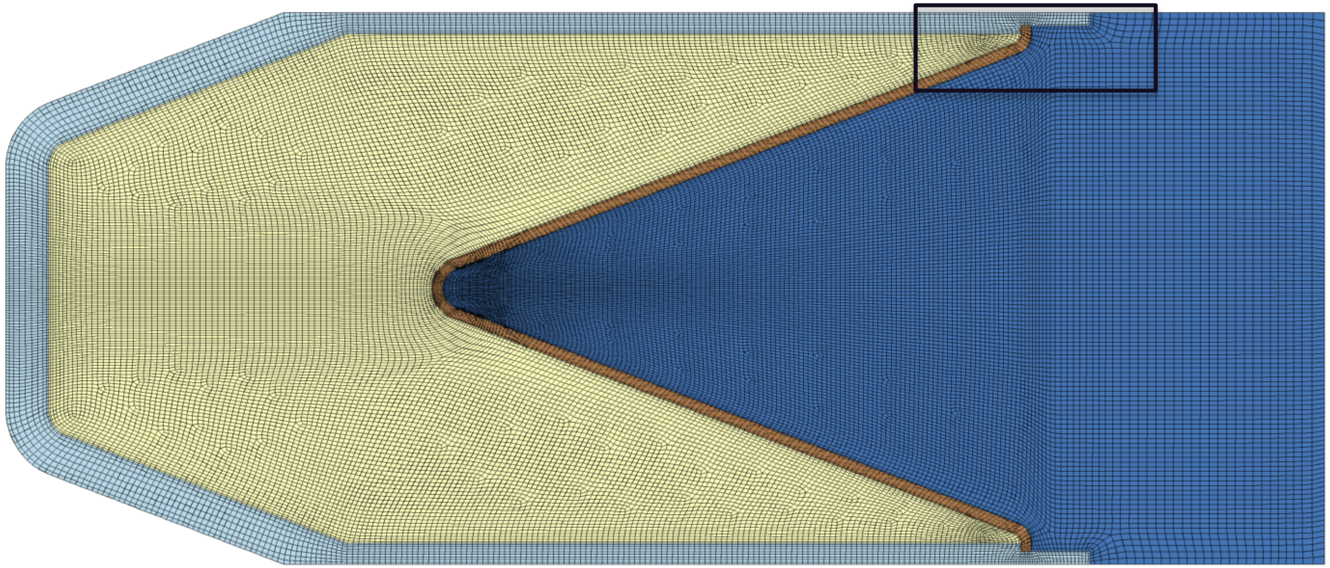
**Example:** 3D Shaped Charge

# Shaped Charge 3D – Geometry and Mesh

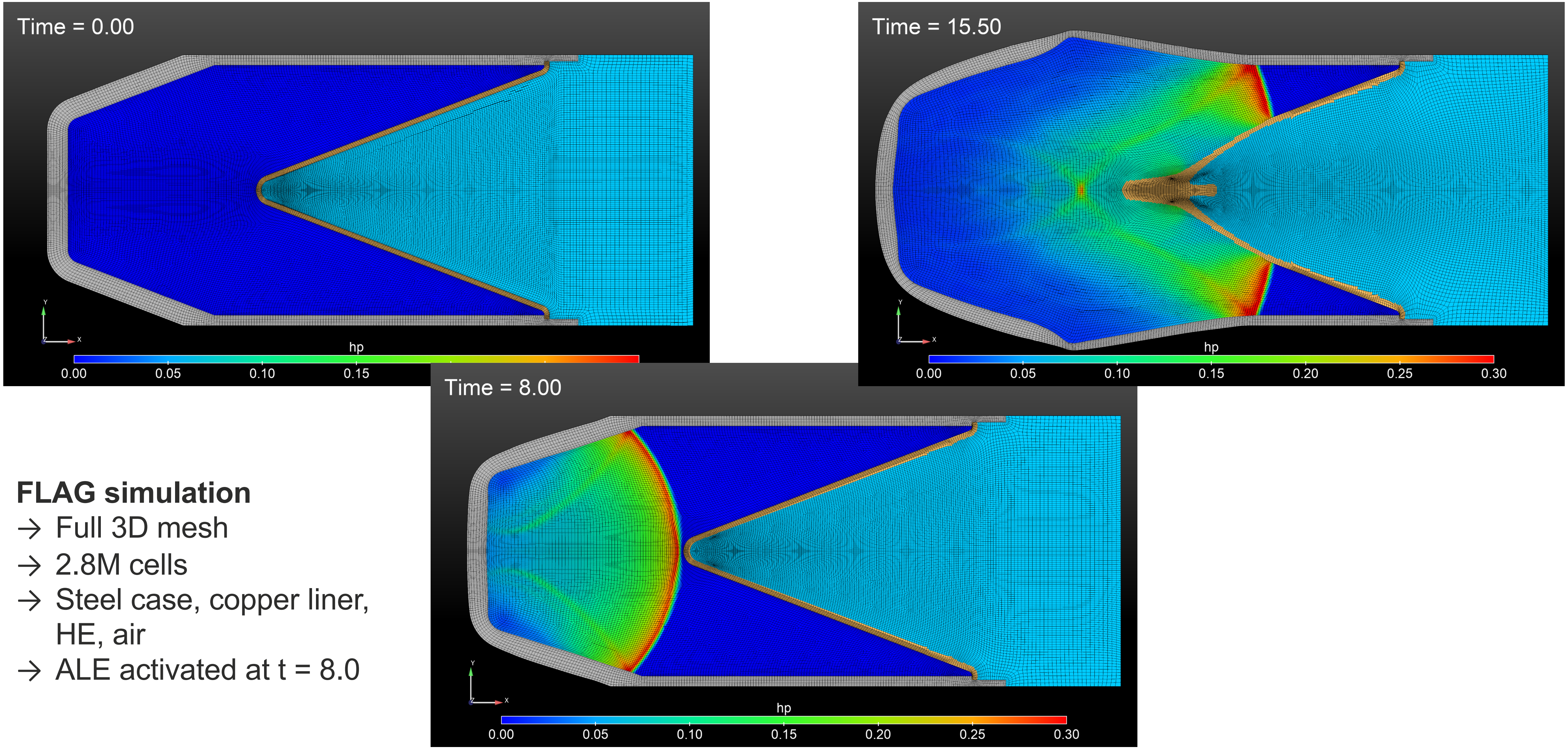




# Shaped Charge 3D – FLAG Simulation




**FLAG simulation**

→ Full 3D mesh

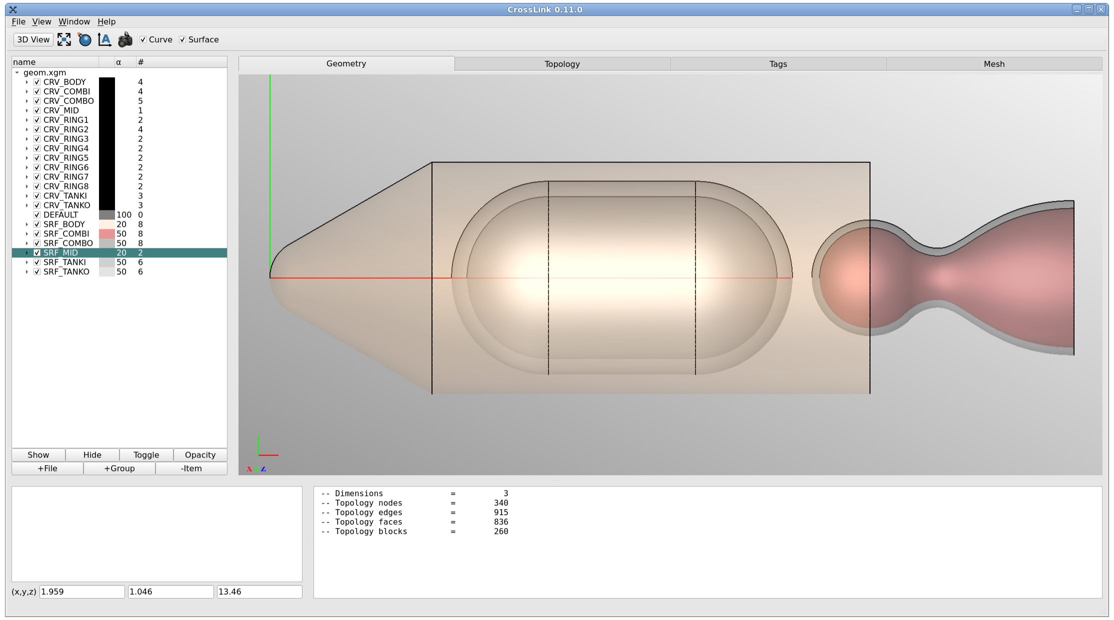
→ 2.8M cells

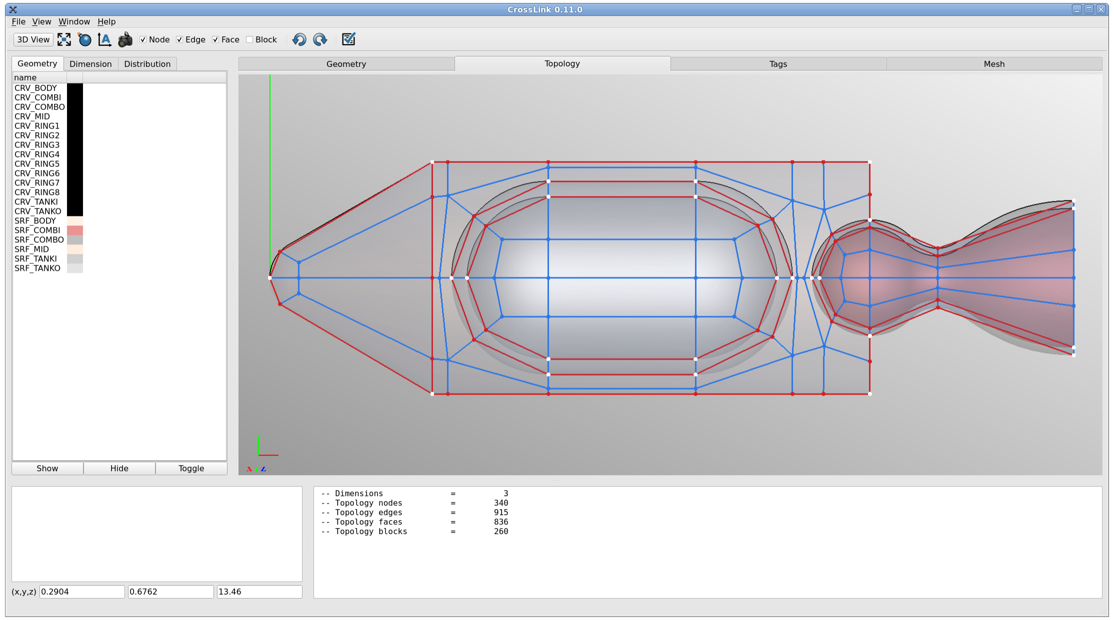
→ Steel case, copper liner, HE, air

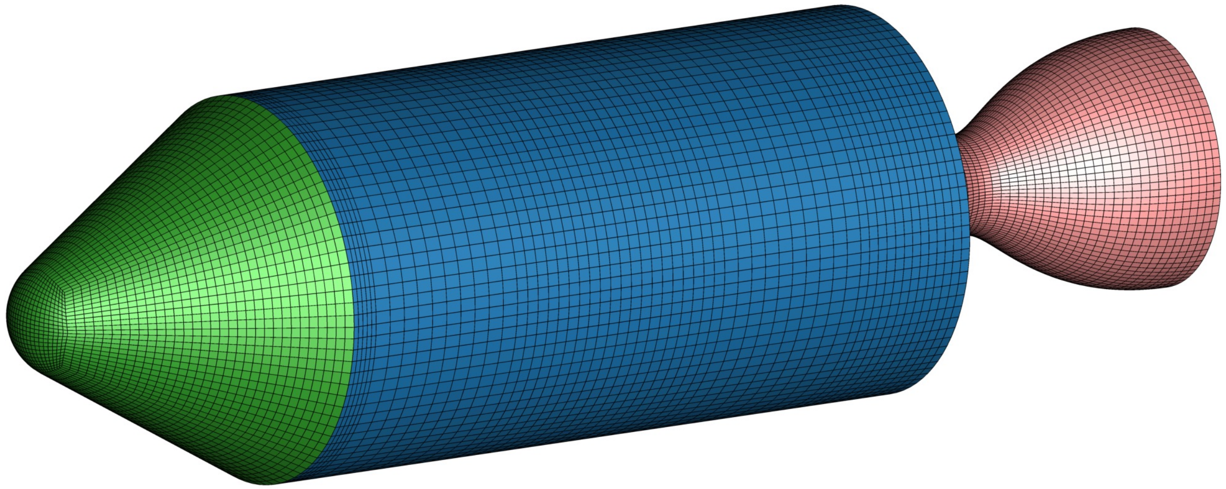
→ ALE activated at t = 8.0

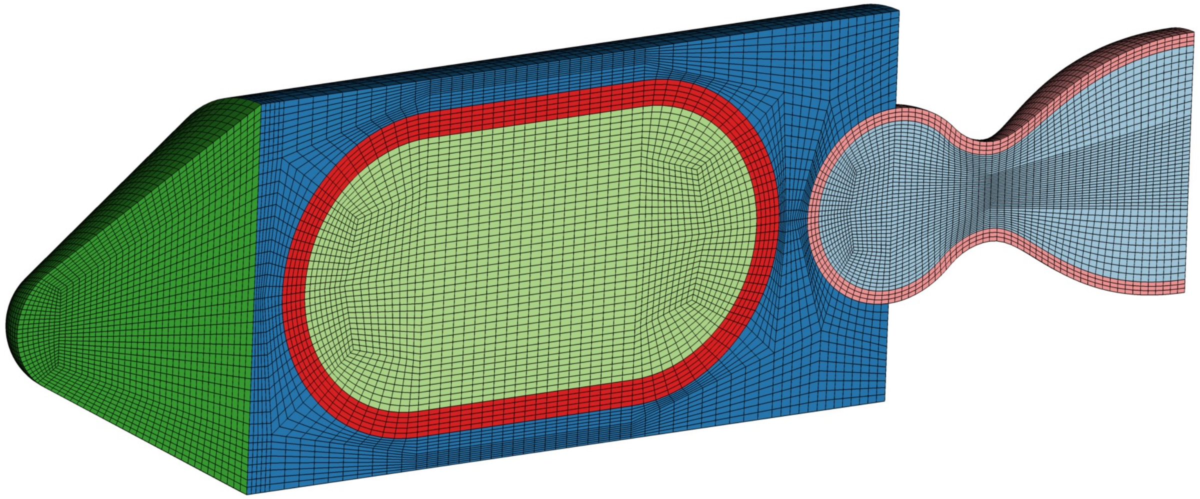
**Example:** 3D Simple Rocket

# 3D Rocket Geometry and Topology

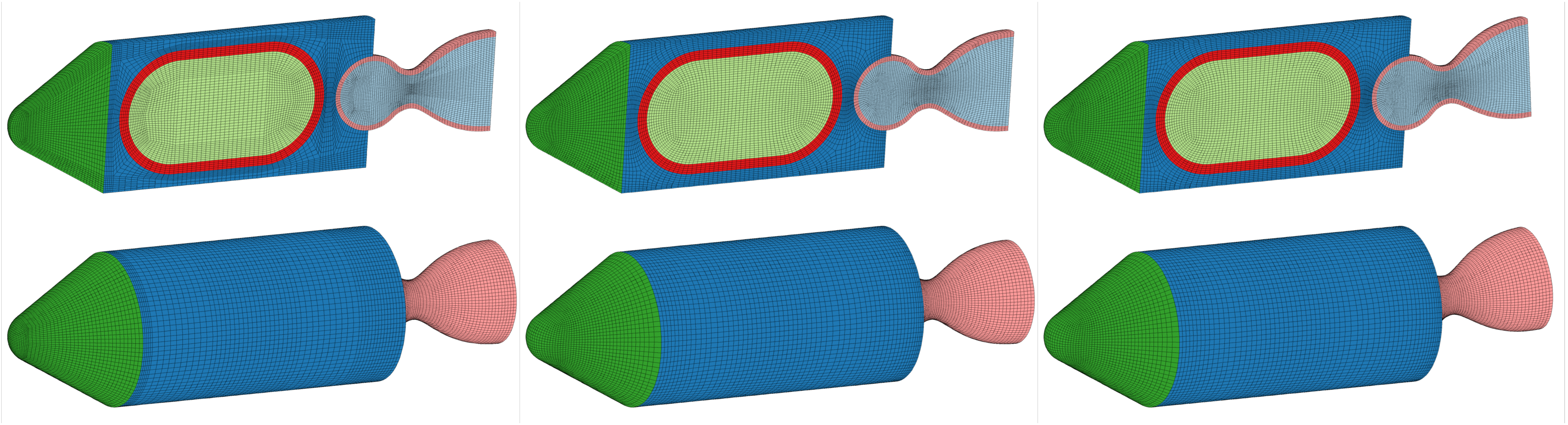


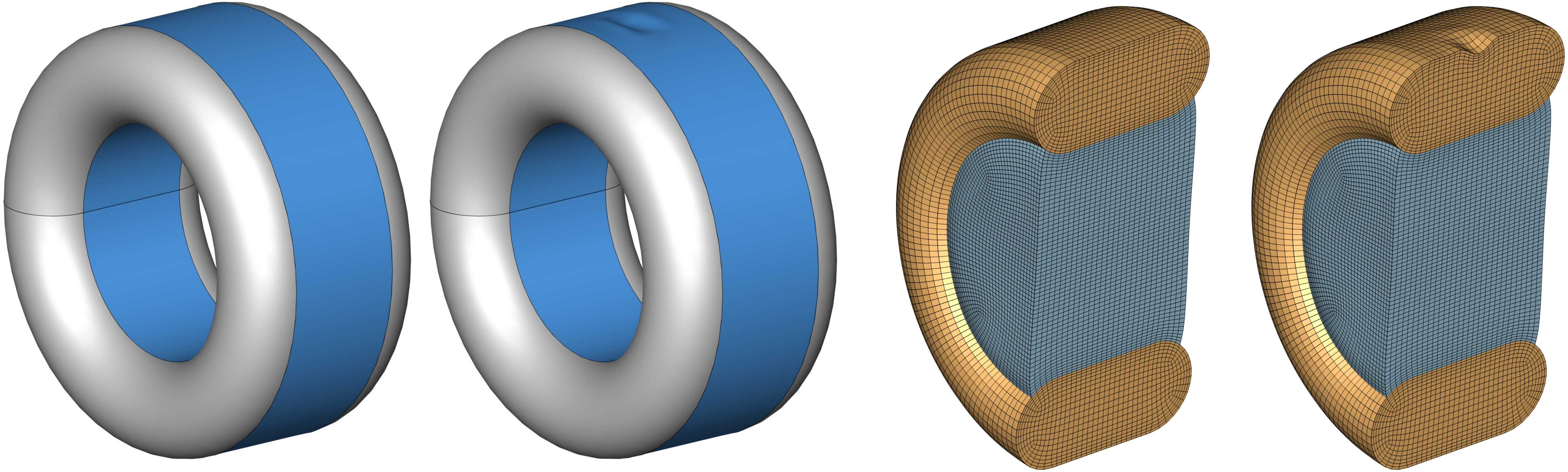


# 3D Rocket Geometry and Topology



- Geometry defined using xGeom python script (NURBS curves and surfaces)
- Nozzle positioned at 0º and 10º
- Mesh regenerated using scripted xMesh script

# Damage Scenario Modeling



- Geometry defined using xGeom python script (NURBS curves and surfaces)
- Damage introduced via control point perturbation (can also interpolate data)
- Mesh regenerated using scripted xMesh script

# **Future:** Advanced Capability R & D

# R & D – Advanced Capabilities

**Develop xCAD model driver**

- Creo Parametric CAD model driver (headless)
- Develop associated Python API

**Parametric Mesh Optimization (ParMesh)**

- Convert mesh engine to operate on high-order cells
- Design high-order mesh description for storage
- Develop mesh builder library for host physics codes

**Curved Topology**

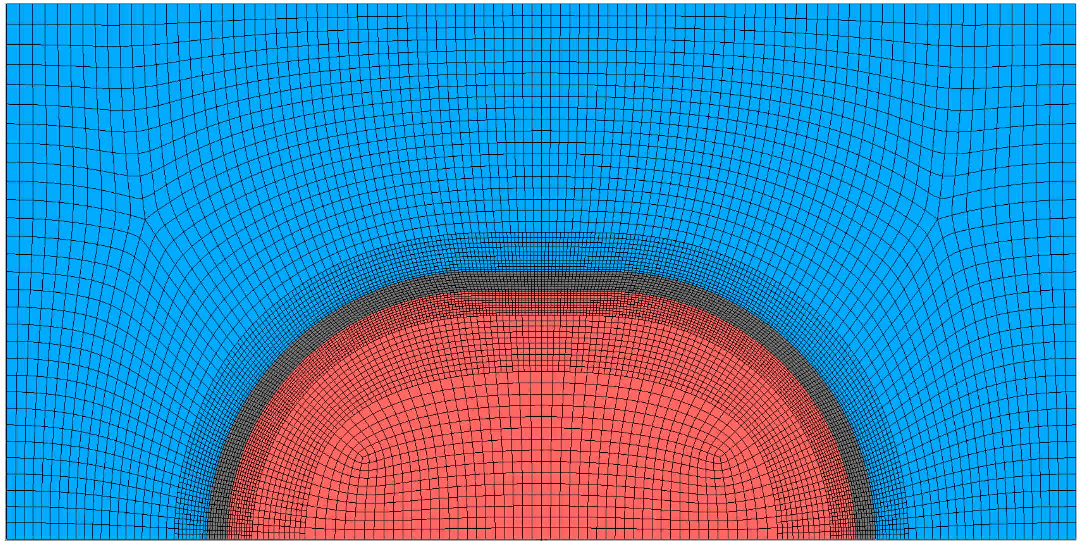
- Implement curved edge and face rendering and selection
- Enable shading on Topology, Tags, and Mesh workbenches

**Compound Topology**

- Enable creation and handling of compound faces and blocks

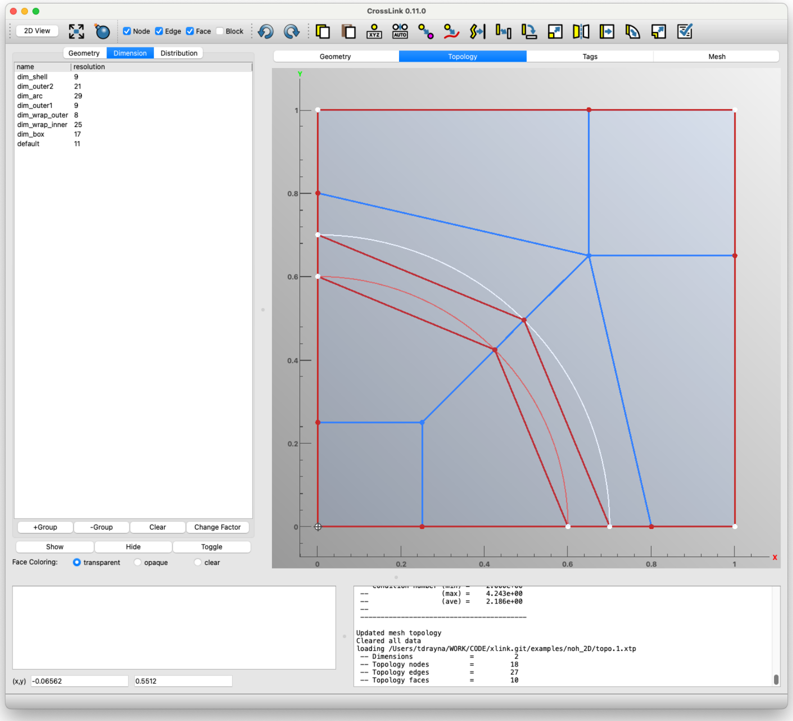
**Crossfield Helper Tools**

- Implement crossfield-based topology helper tools

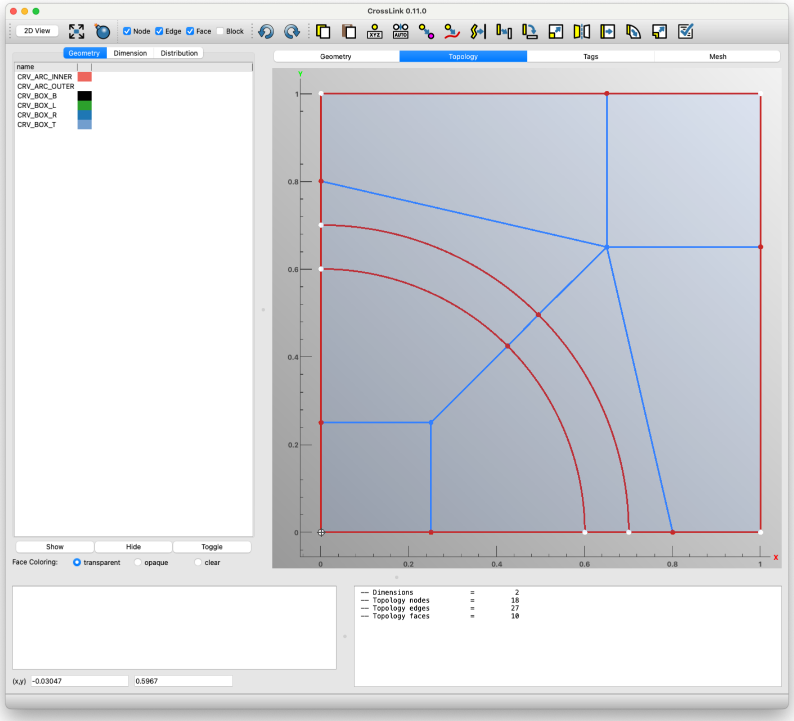


ParMesh with block AMR.

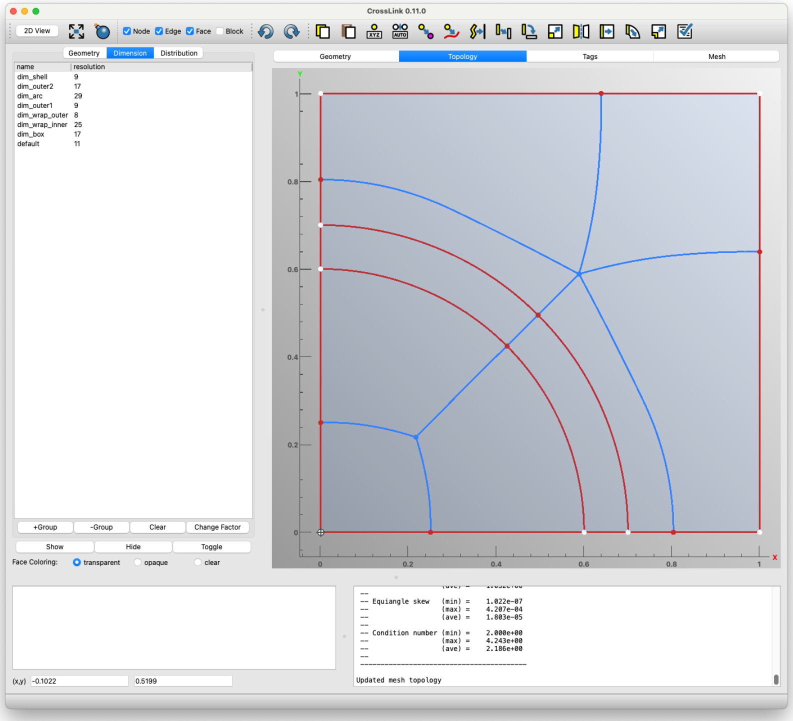
# R & D – Curved Topology



Original topology



Curved topology



Curved topology refit to mesh

- Cubic NURBS curves represent topology edges; additional data stored on edges
- Enables interactive user modification, mesh refitting, on-geometry fitting
- Improves visual experience, improves initial mesh, reduces mesh optimization cost

# R & D – Crossfield-based Helper Tool

- Topology helper can be executed on any closed domain
- Solver computes singularities and resulting topology (red)
- Field streamlines can be created to guide topology assembly (blue)
- Can be used as a helper for both 2D and 3D domains