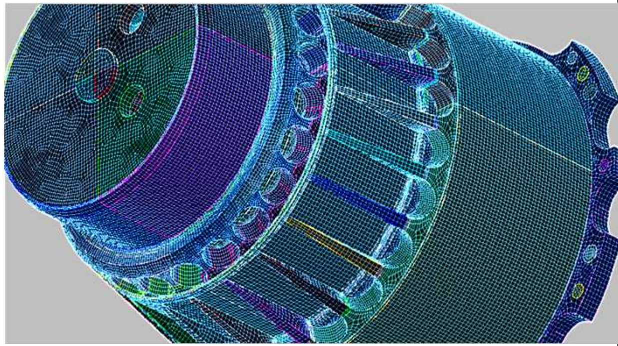
SAND2016-9195C

Exceptional service in the national interest

Sandia National Laboratories



An Introduction to Automatic Mesh Generation Algorithms

Short Course, September 26, 2016

Washington, DC

Steven Owen

Sandia National Laboratories

U.S. DEPARTMENT OF ENERGY

NNSA National Nuclear Security Administration

Agenda

Sandia National Laboratories

Part I 8:00-9:30AM

- The Simulation Process
- Geometry Basics
- Mesh Representations
- Mesh Generation Methods
- Tet/Tri Meshing Methods
- Surface Meshing Basics
- Smoothing

Part II 10:00-11:30AM

- Tet vs. Hex Meshing
- Structured vs. Unstructured
- Structured Hex Methods
- Unstructured Hex Methods
- Hex Dual Representations
- Overlay Grid
- Automatic Block Decomposition
- Hybrid Methods

Hex Meshing Software

A Representative Sample

Unstructured

- Cubit/Trelis, Sandia National Laboratories, Csimsoft
- Hexotic, INRIA, France, Distene
- Harpoon, Sharc
- Kubrix, Simulation Works
- Hexpress, Numeca
- Hypermesh, Altair
- Patran, MSC Software

Structured/Multiblock

- TrueGrid, XYZ Scientific Applications, Inc.
- GridPro, Program Development Company
- ICEM CFD, Ansys, Inc.
- Gridgen, Pointwise, Inc.

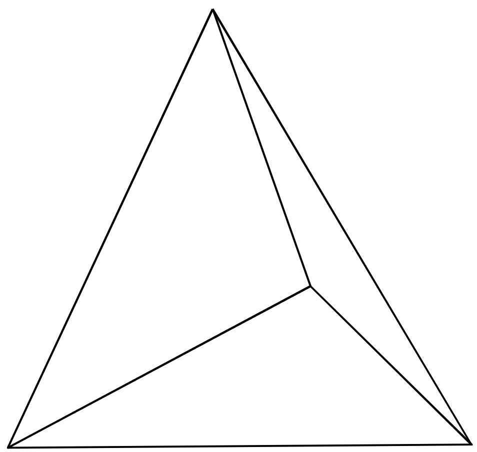
Survey Paper on Hex Mesh Generation

I recommend this survey paper for people entering the field of hex mesh generation:

J. Sarrate, E. Ruiz-Gironés and X. Roca, "Unstructured and Semi-Structured Hexahedral Mesh Generation Methods," Computational Technology Reviews, Volume 10, 2014, http://www.ctresources.info/ctr/paper.html?id=60

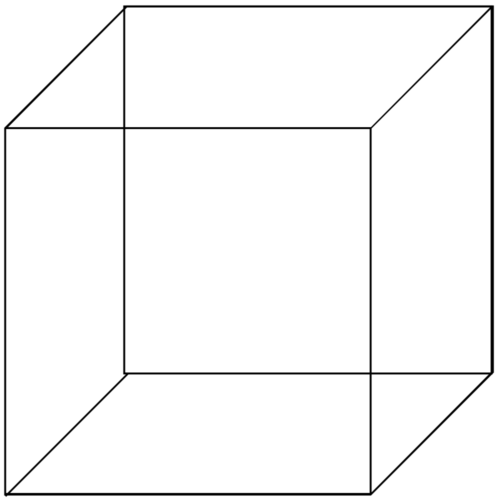
Hexahedra vs. Tetrahedra

Finite element meshes can be generated with either tetrahedral or hexahedral elements. Automatic tetrahedral meshing is generally considered a solved problem, while hexahedral meshing is still an open problem.



Tetrahedra (simplex)
- 4 nodes
- 6 edges
- 4 faces

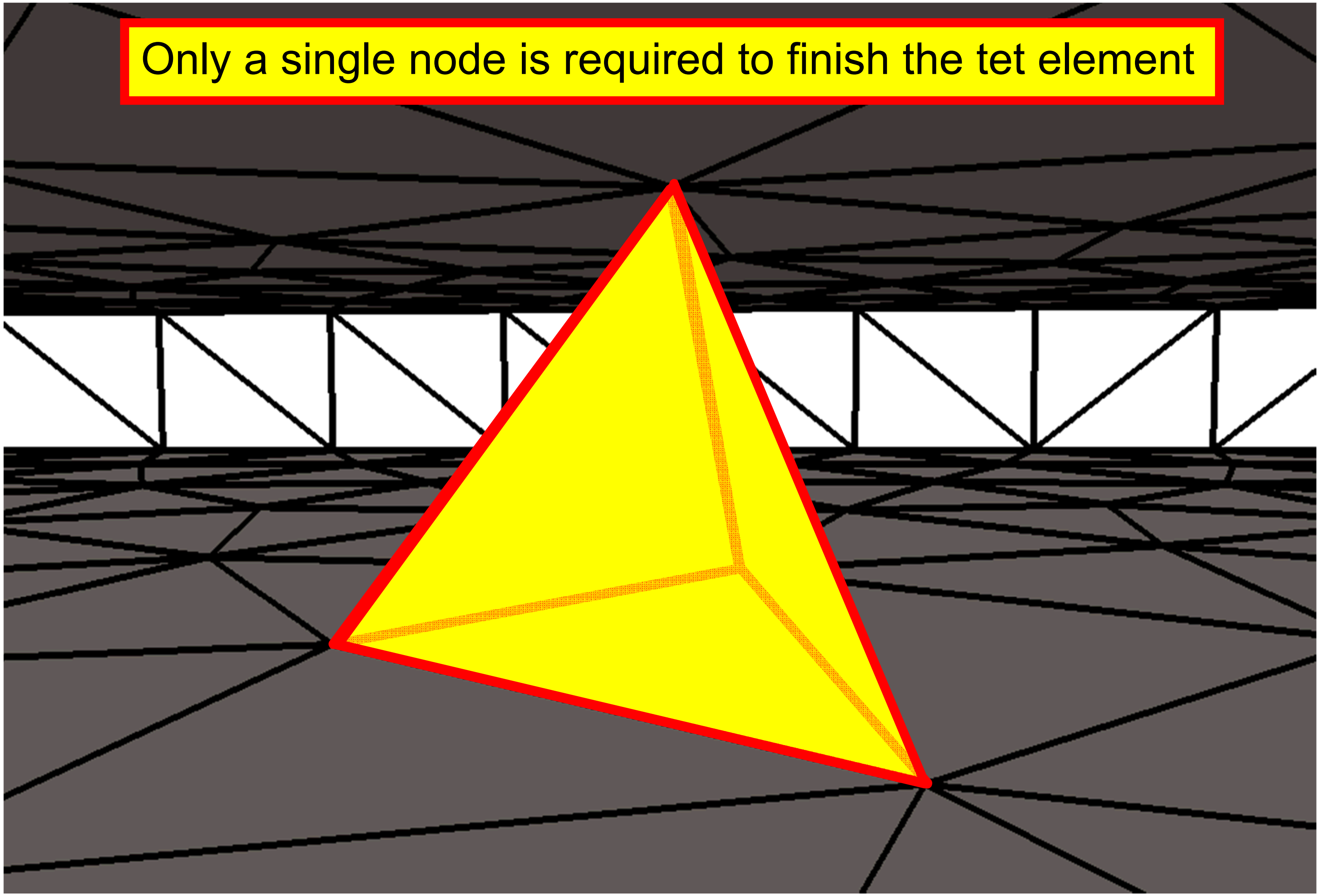
Automated Generation
Locally Modifiable



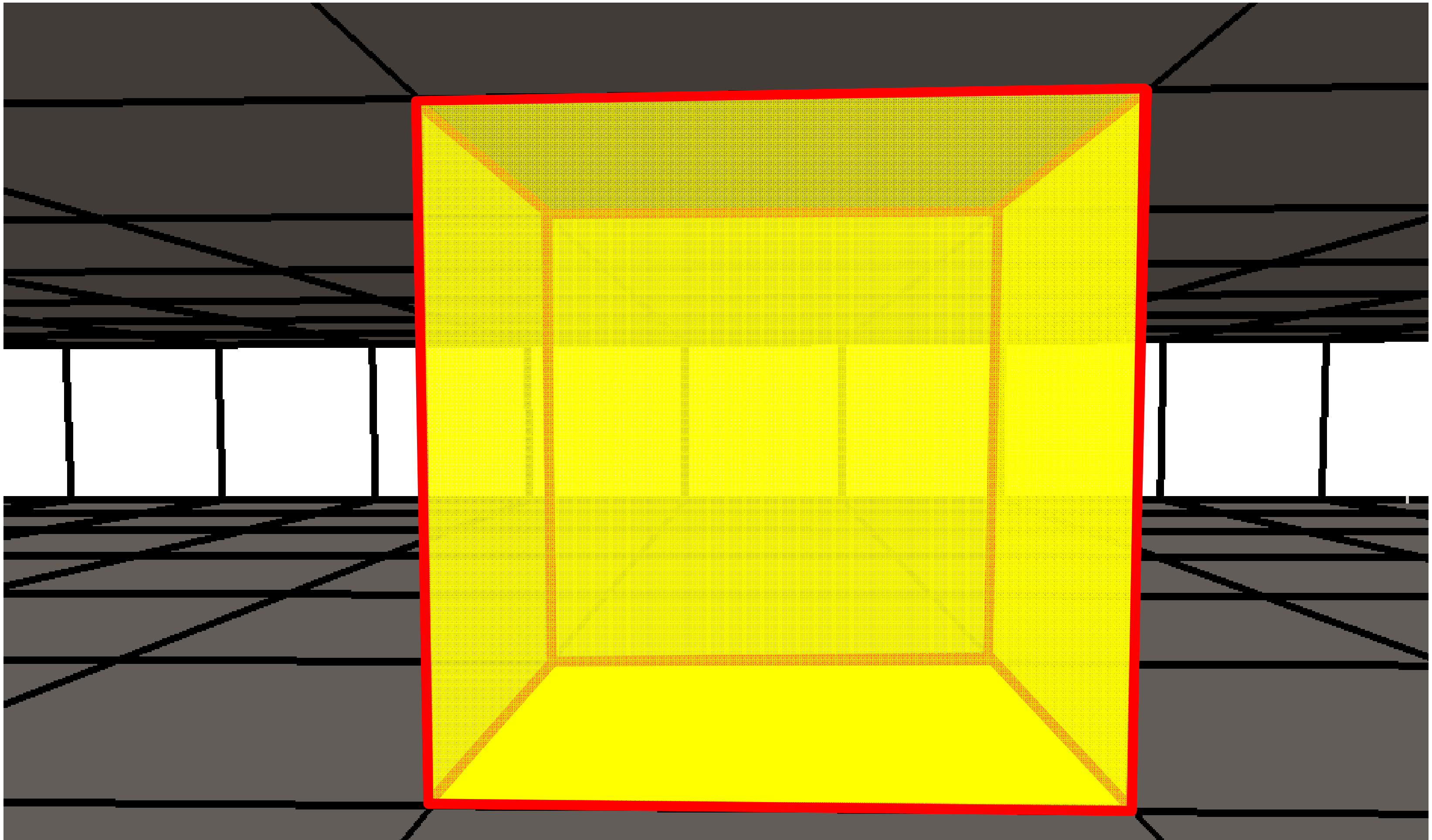
Hexahedra
- 8 nodes
- 12 edges
- 6 faces

Semi-Automatic & Manual Generation
Constrained Modifications

Advancing Front Element Creation

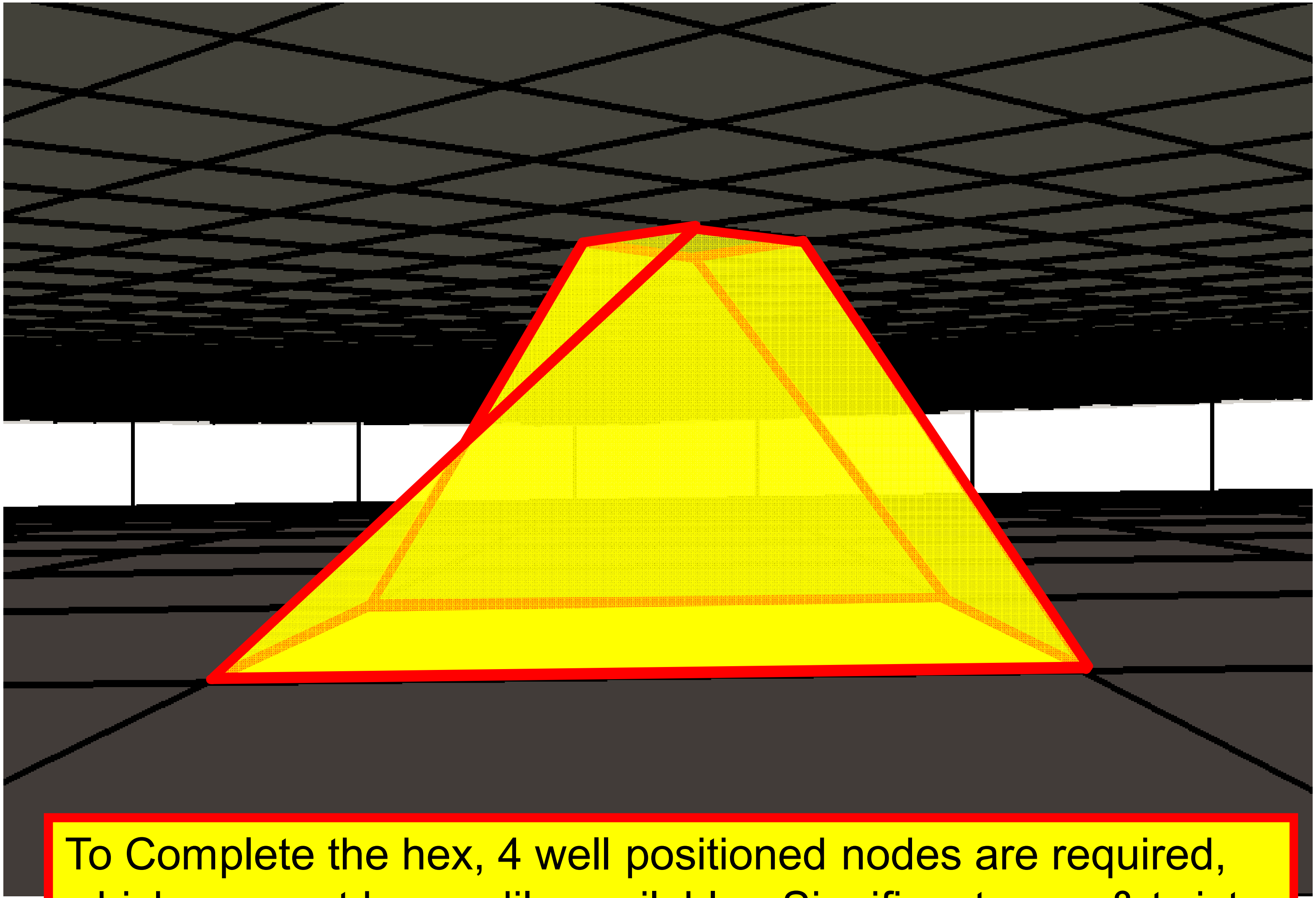


Advancing Front Element Creation



To Complete the hex, 4 well positioned nodes are required, which may not be readily available.

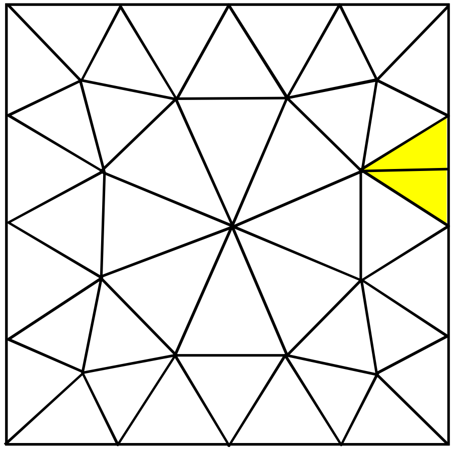
Advancing Front Element Creation



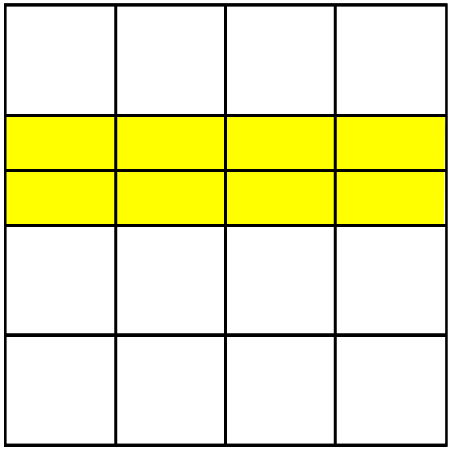
To Complete the hex, 4 well positioned nodes are required, which may not be readily available. Significant warp & twist are often required to build hex elements.

Local Modifications to Meshes

Triangle and tetrahedral meshes can be easily modified locally.



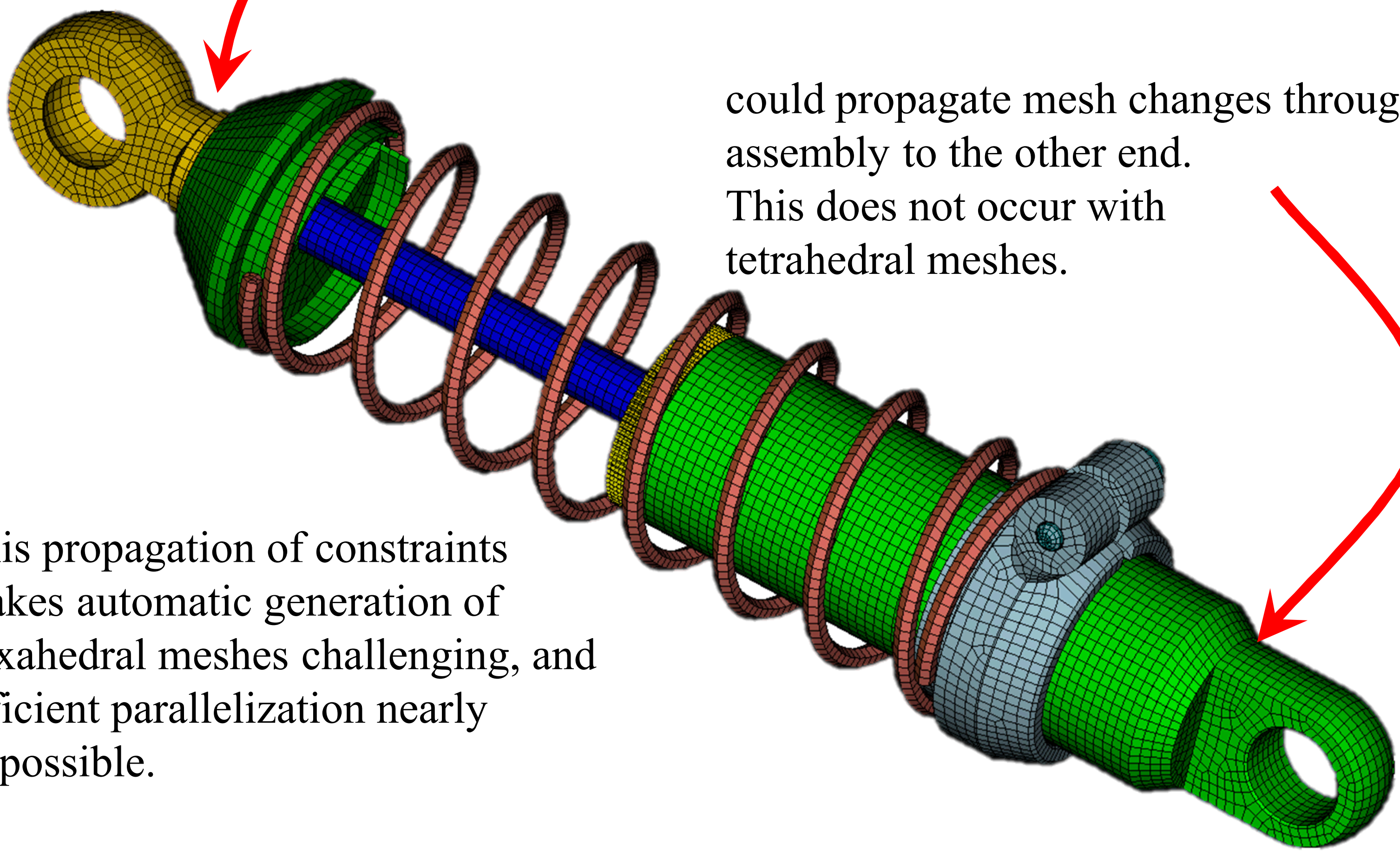
In contrast, to maintain a conforming non-hybrid mesh, small changes to quadrilateral and hexahedral meshes propagate through the mesh due to topology constraints.



Global Propagation of Hexahedral Constraints on Assembly Models

When generating all-hexahedral conforming mesh through interfaces in an assembly,

a geometric feature here



could propagate mesh changes through the assembly to the other end.
This does not occur with tetrahedral meshes.

This propagation of constraints makes automatic generation of hexahedral meshes challenging, and efficient parallelization nearly impossible.

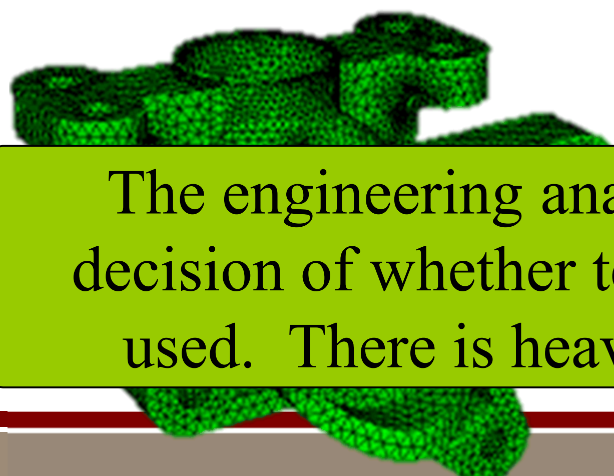
Tet Meshing Vs. Hex Meshing

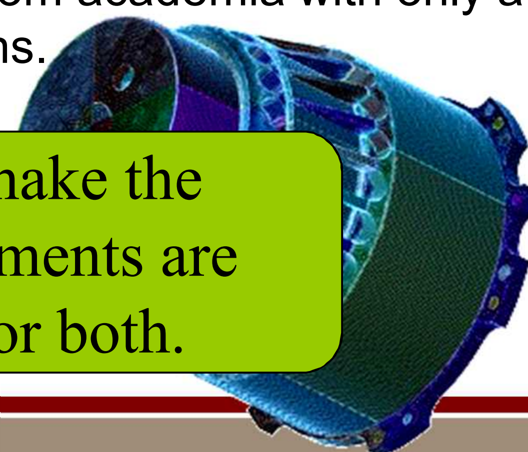
Tet Meshing

1. Fully Automated, mostly push-button
2. Generate millions of elements in minutes/seconds
3. User time generally minutes/hours
4. Can require 4-10X number of elements to achieve same accuracy as all-hex mesh
5. Tet-Locking phenomenon for linear tet results in stiffer physics
6. Preferred by many academics for mathematic properties in generation

Hex Meshing

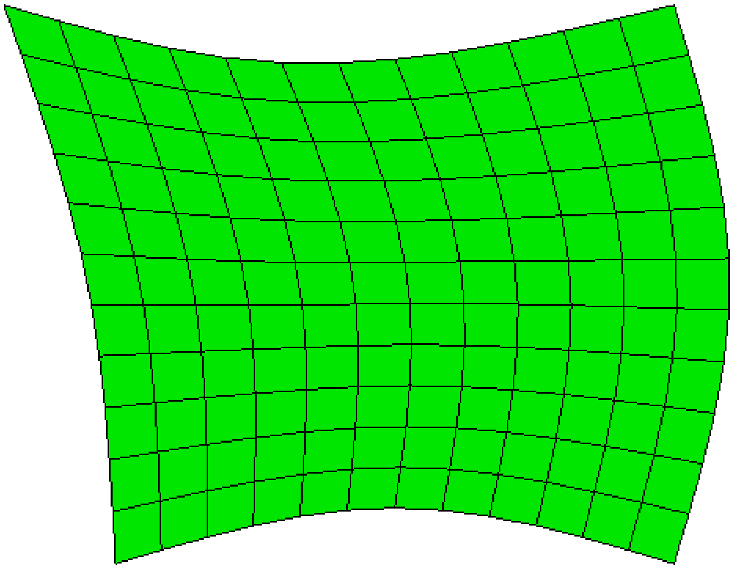
1. Partially automated, some manual
2. Can require major user effort/expertise to prepare geometry to accept a hex mesh
3. User time to generate mesh may be typically days/weeks/months
4. Computational methods may prefer or require hex element
5. Preferred by many analysts for solution accuracy
6. Heavily used in industry & national labs. Absent from academia with only a few exceptions.

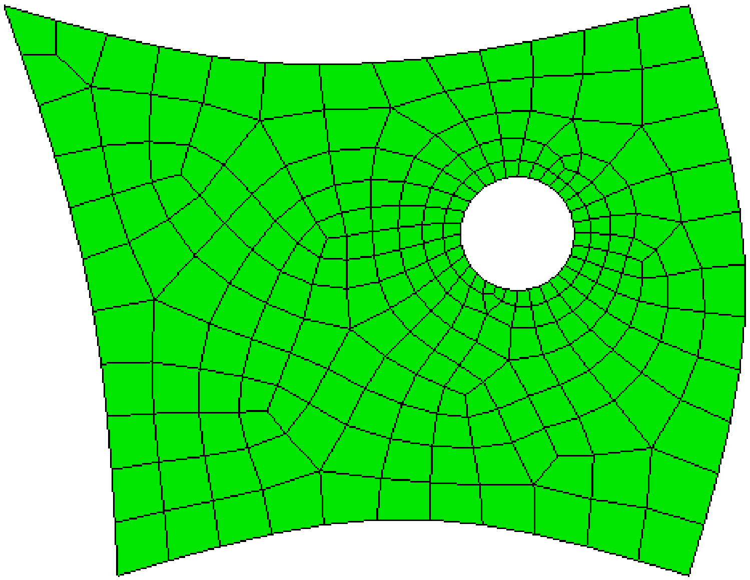




The engineering analysts must make the decision of whether tet or hex elements are used. There is heavy demand for both.

Structured vs. Unstructured





Structured

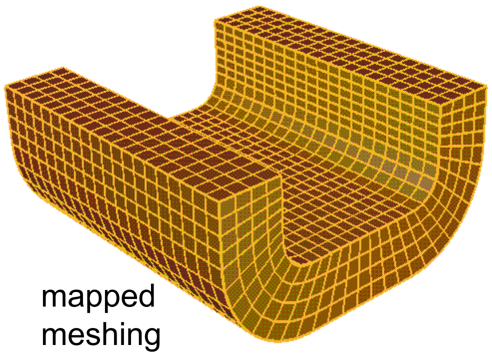
1. Interior node valence is constant.
   ie. number of elements at each interior node=4
2. Meshing algorithm relies on specific topology constraints.
   ie. number of sides=4

Unstructured

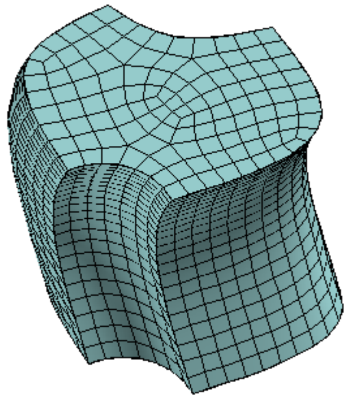
1. Interior node valence varies.
   ie. number of elements at each node=3,4,5…
2. Meshing algorithm applies to arbitrary topology
   ie. number of sides is arbitrary

Structured vs. Unstructured

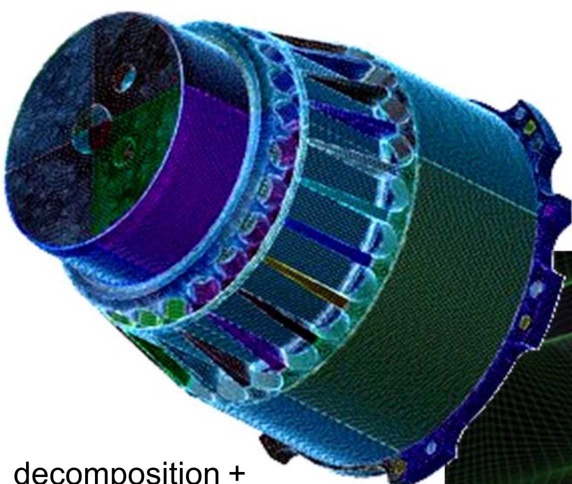
Structured



mapped meshing

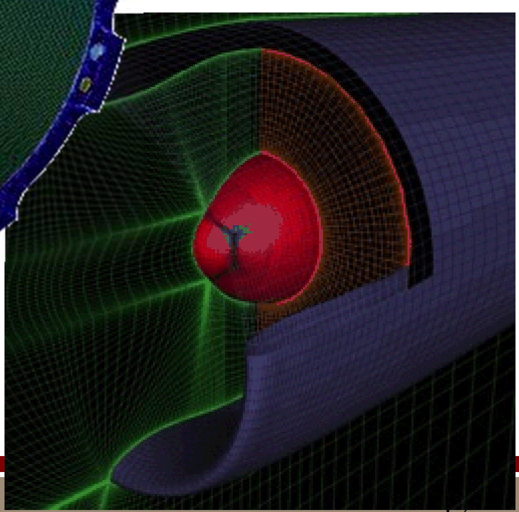


sweeping

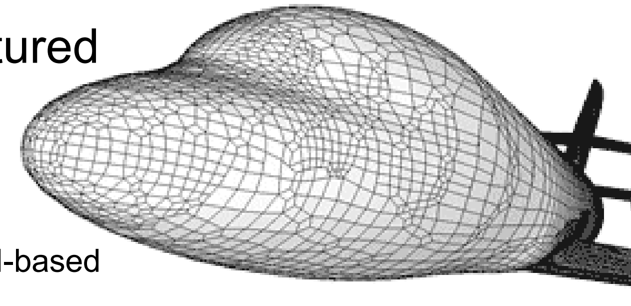


decomposition + sweeping

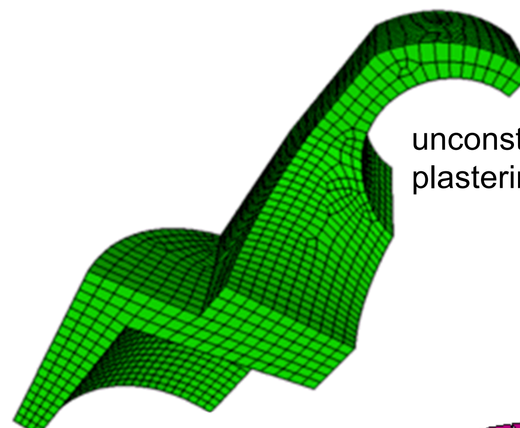
block structured



Unstructured

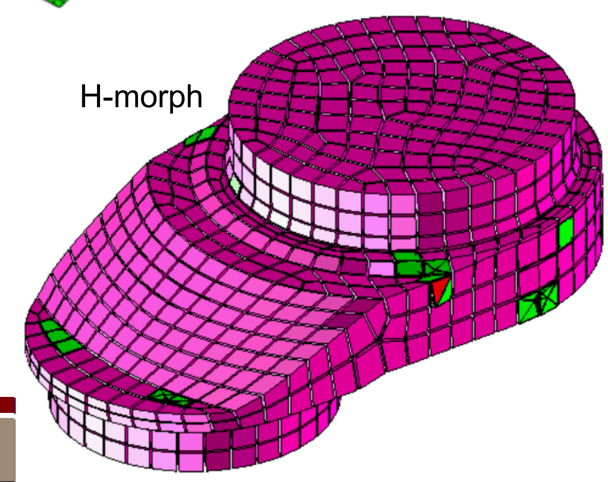


grid-based



unconstrained plastering

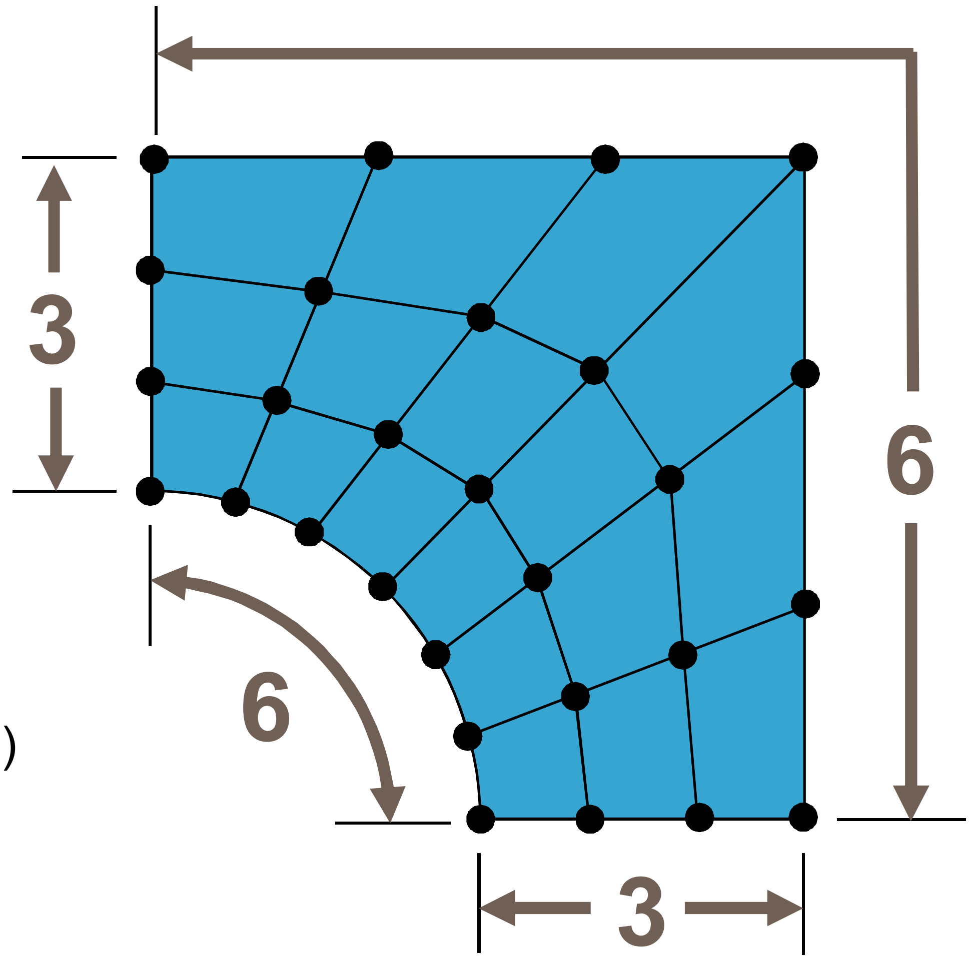
H-morph



Mapped Meshing

Algorithm

- Trans-finite Interpolation (TFI)
- maps a regular lattice of quads onto polygon (Thompson,88;99) (Cook,82)

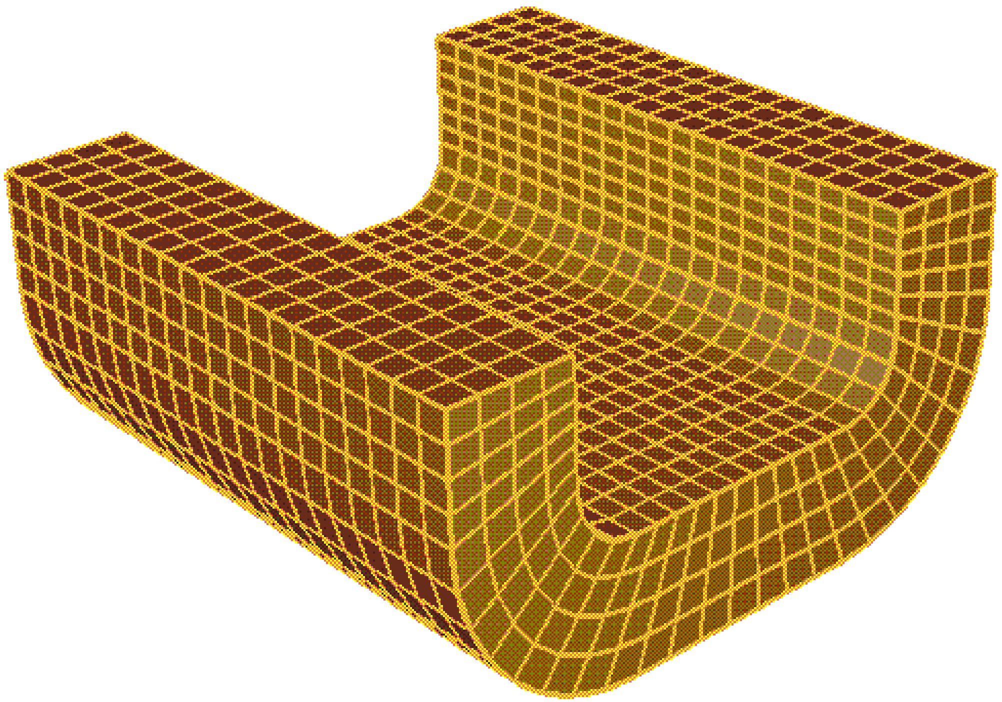


Geometry Requirements

- 4 topological sides
- opposite sides must have similar intervals

Mapped Meshing

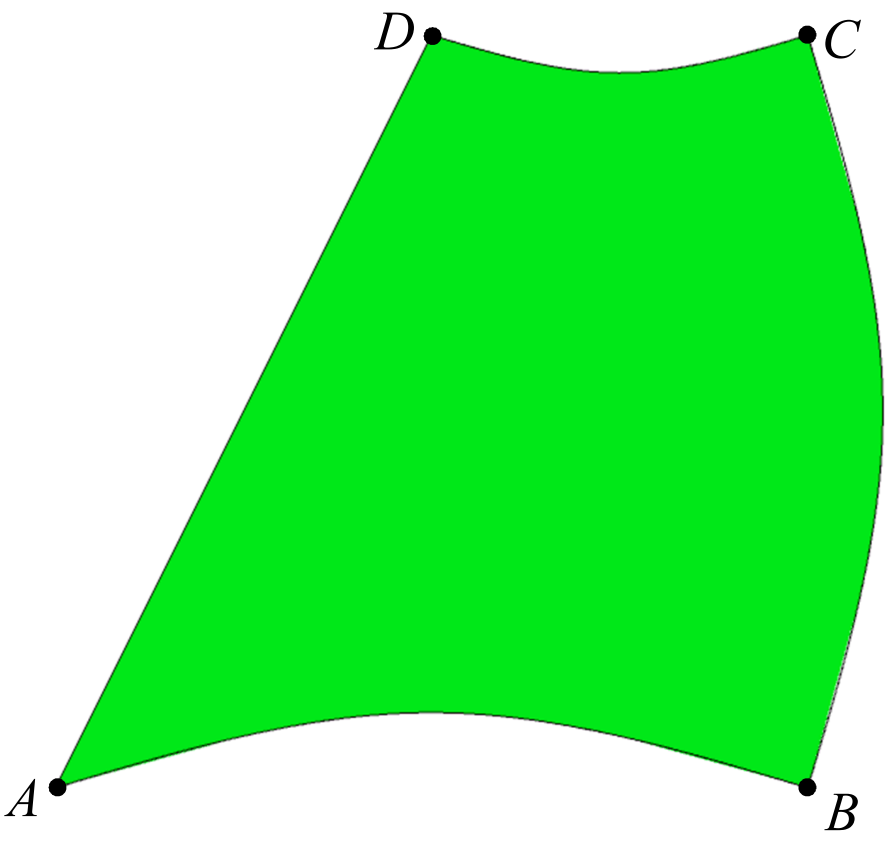
3D Mapped Meshing



Geometry Requirements

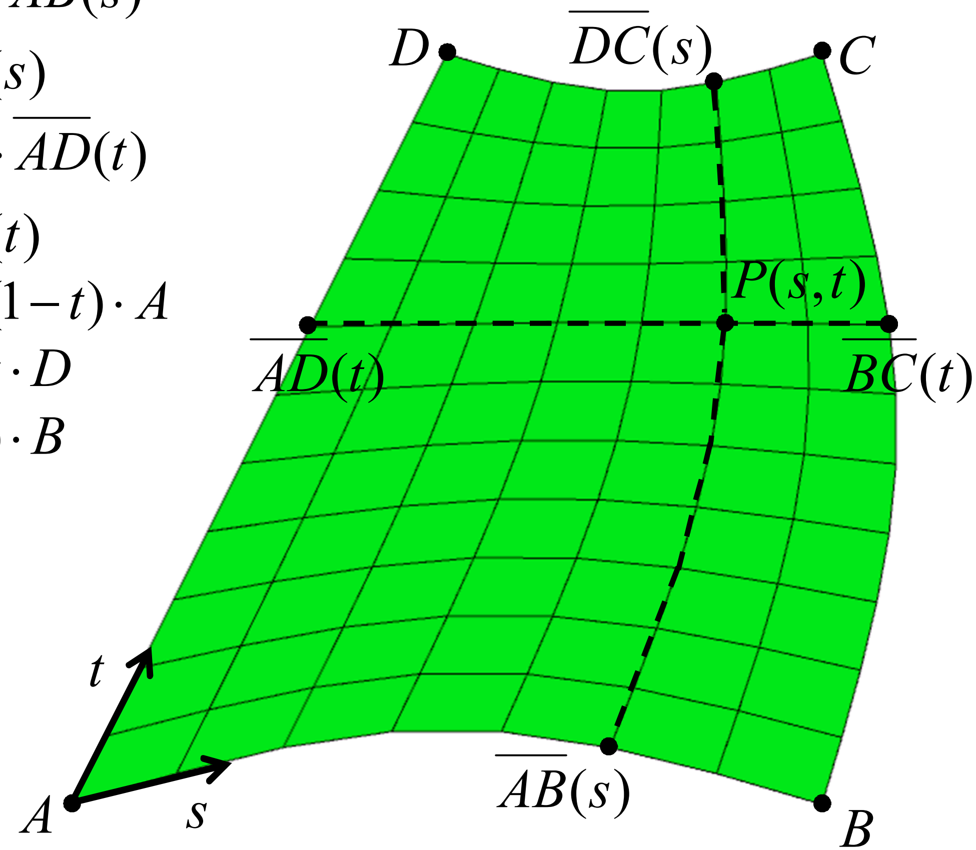
- 6 topological surfaces
- opposite surfaces must have similar mapped meshes

Transfinite Interpolation



Transfinite Interpolation

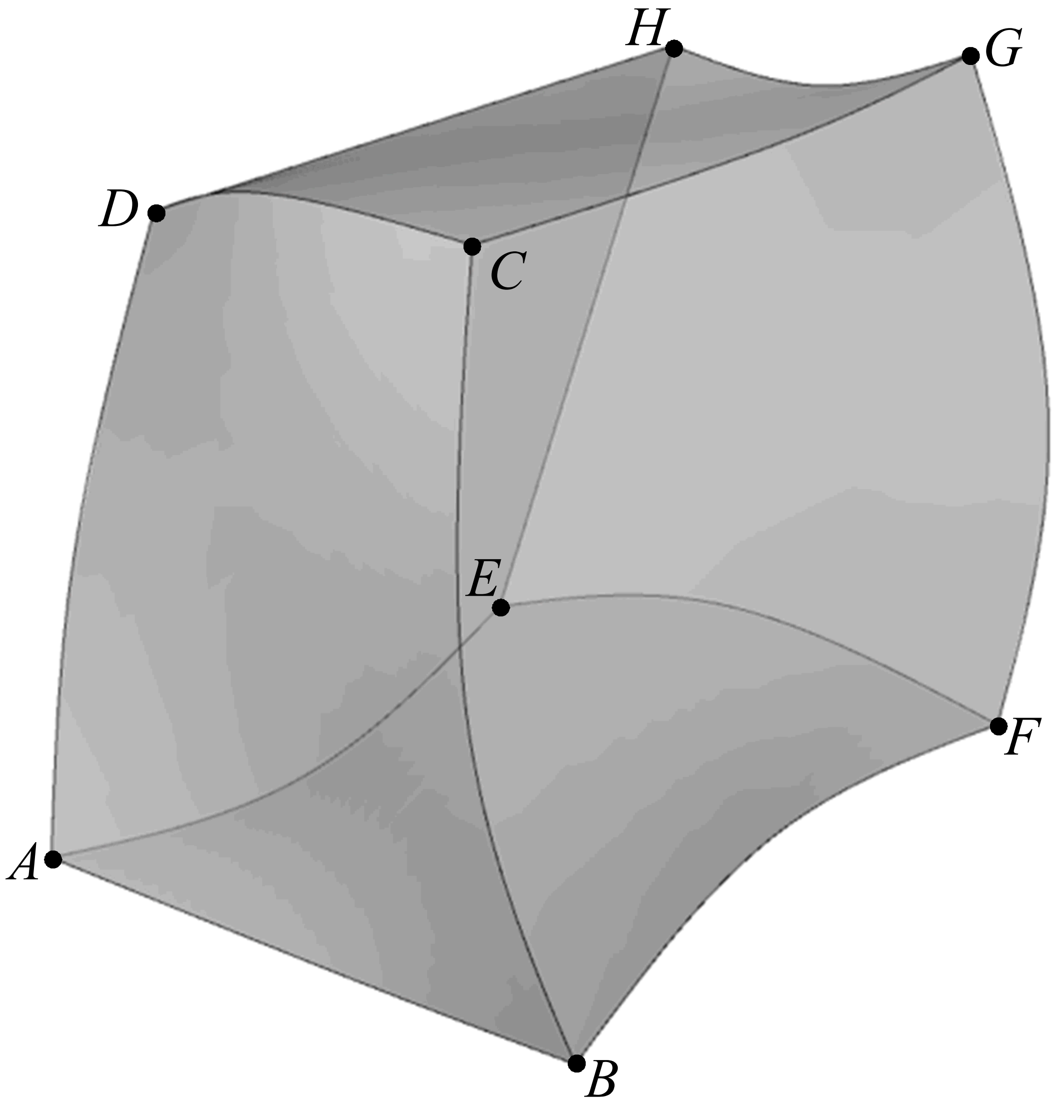
$$P(s,t) = (1-t)\cdot \overline{AB}(s) + t\cdot \overline{DC}(s) + (1-s)\cdot \overline{AD}(t) + s\cdot \overline{BC}(t) - (1-s)(1-t)\cdot A - (1-s)t\cdot D - s(1-t)\cdot B - st\cdot C$$



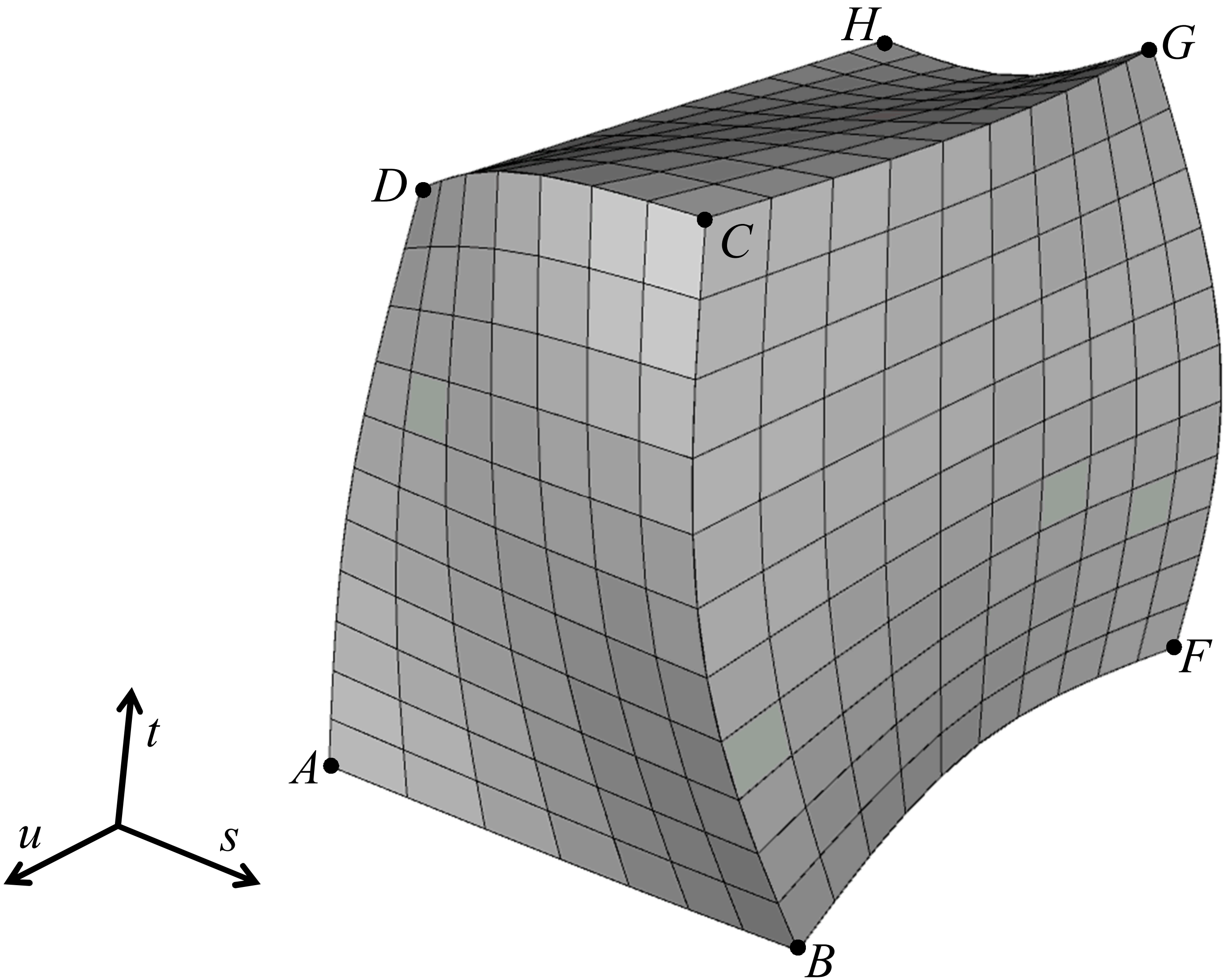
$$\overline{AD}(t) = A + t(D - A)$$

straight segment

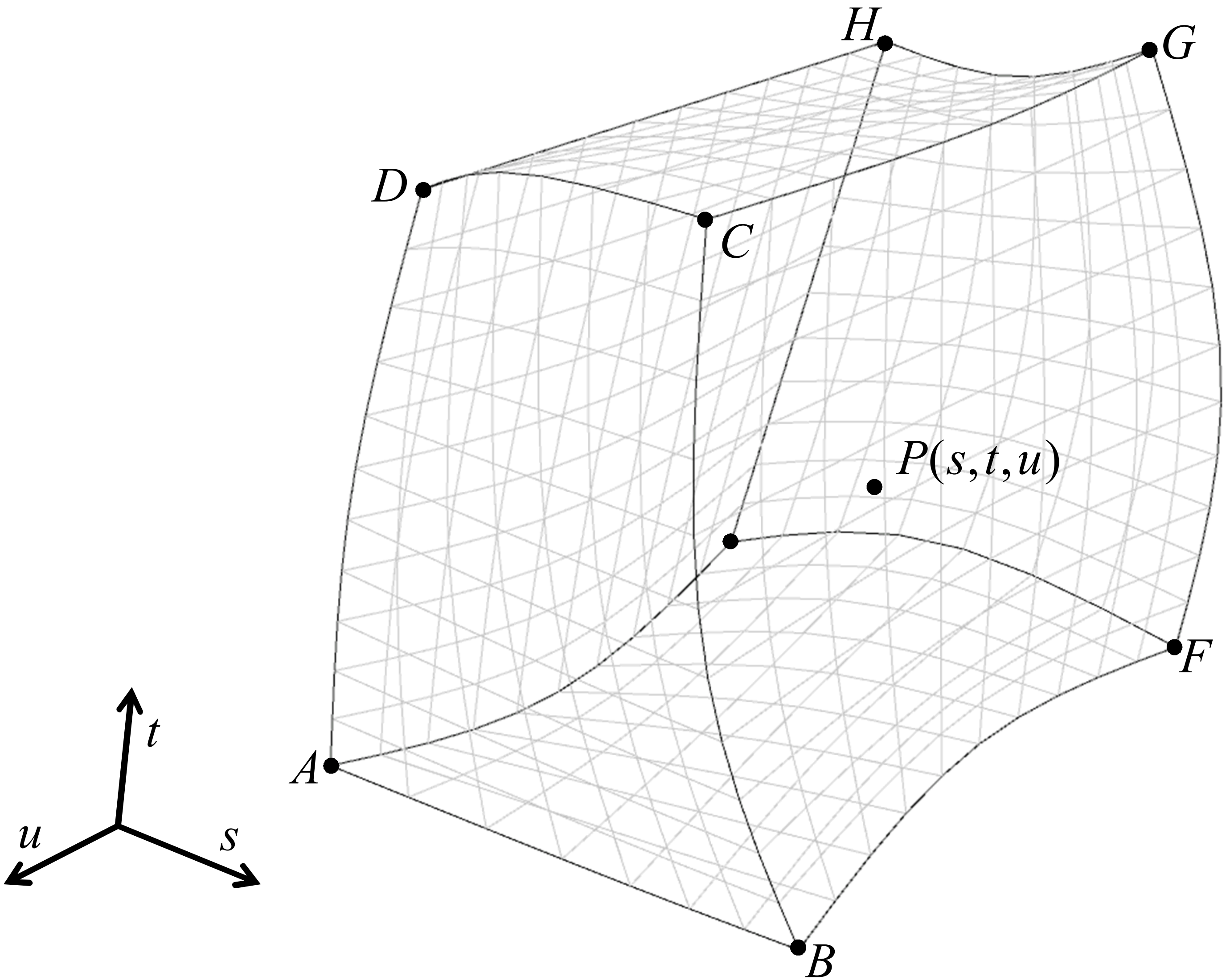
Transfinite Interpolation



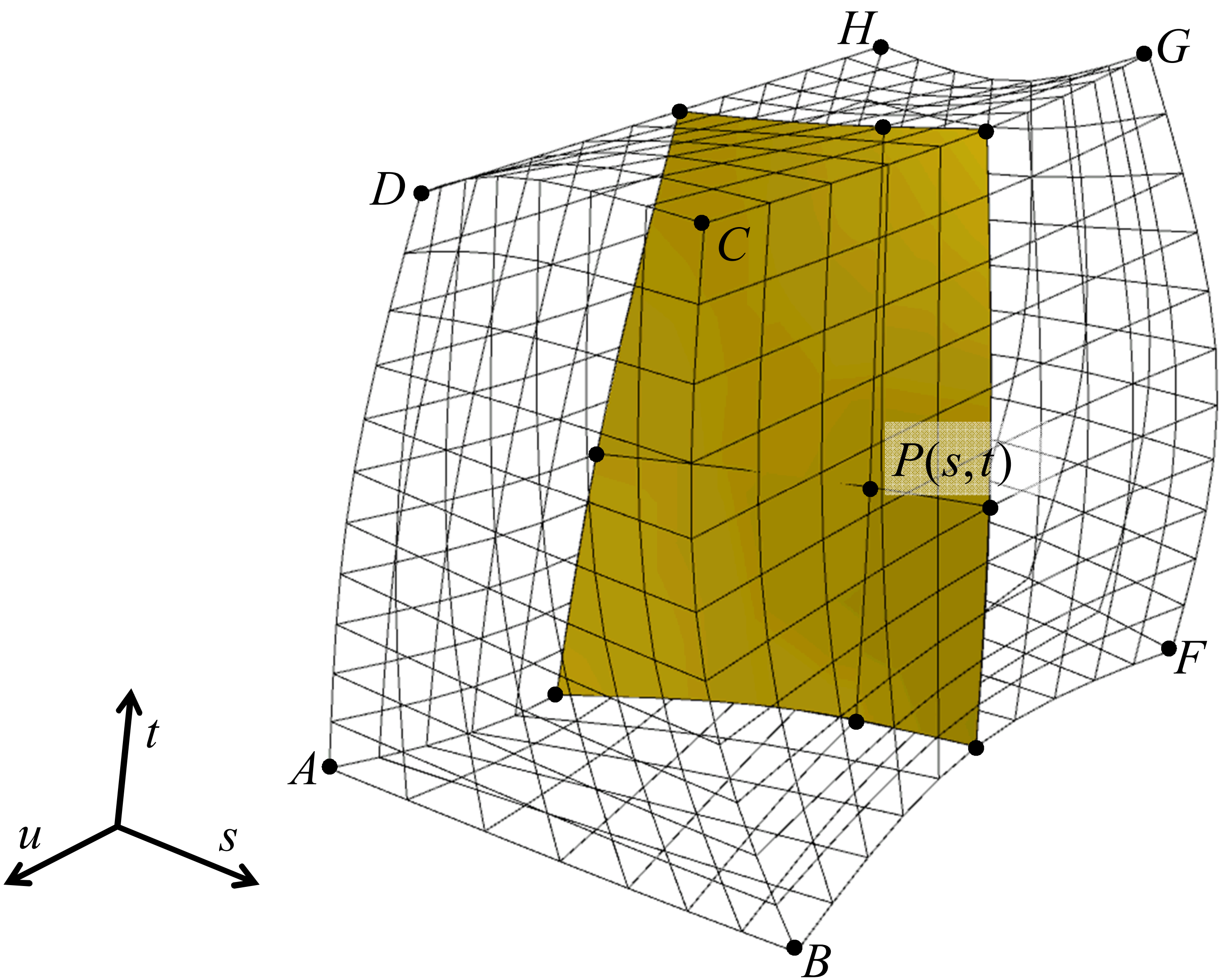
Transfinite Interpolation



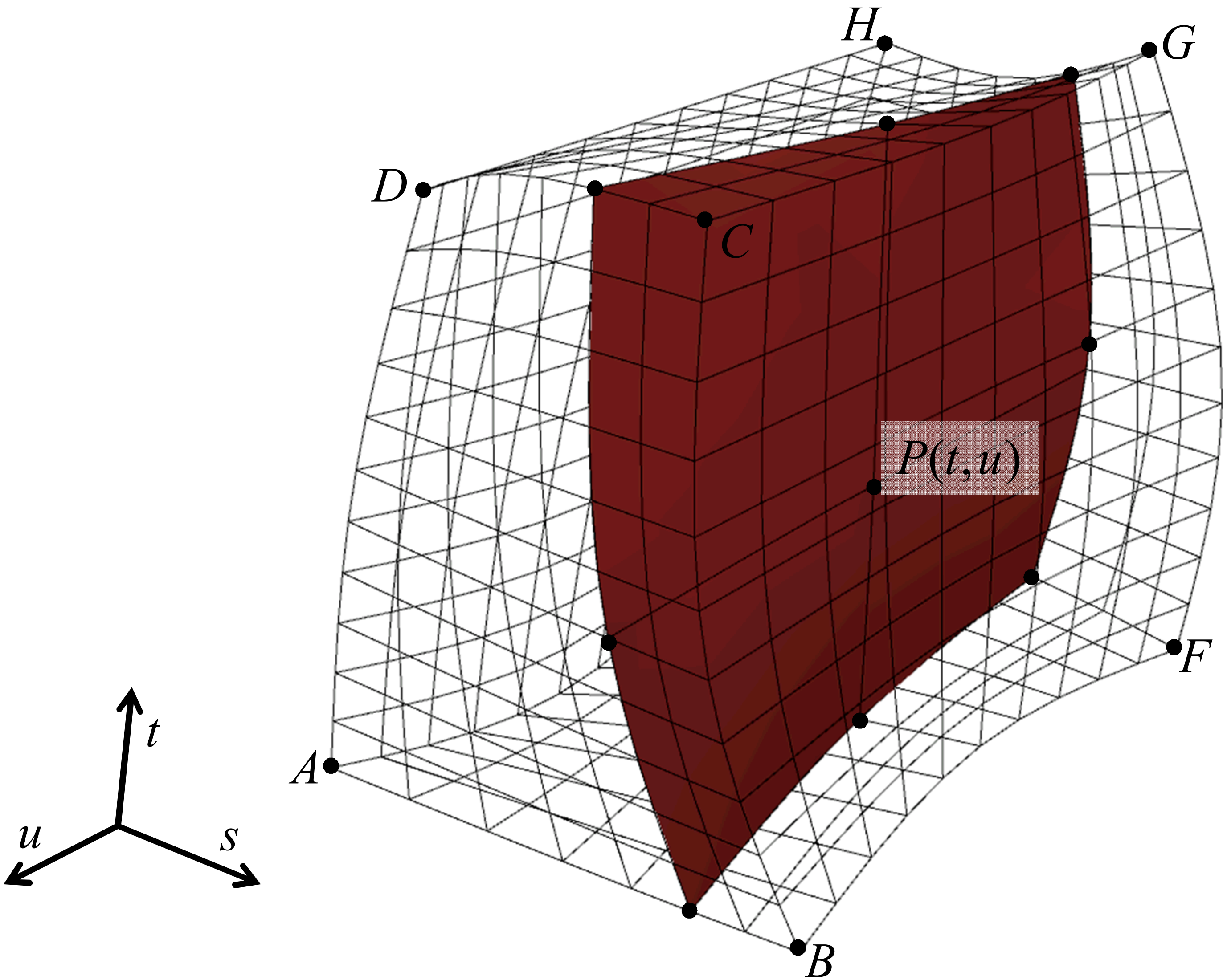
Transfinite Interpolation



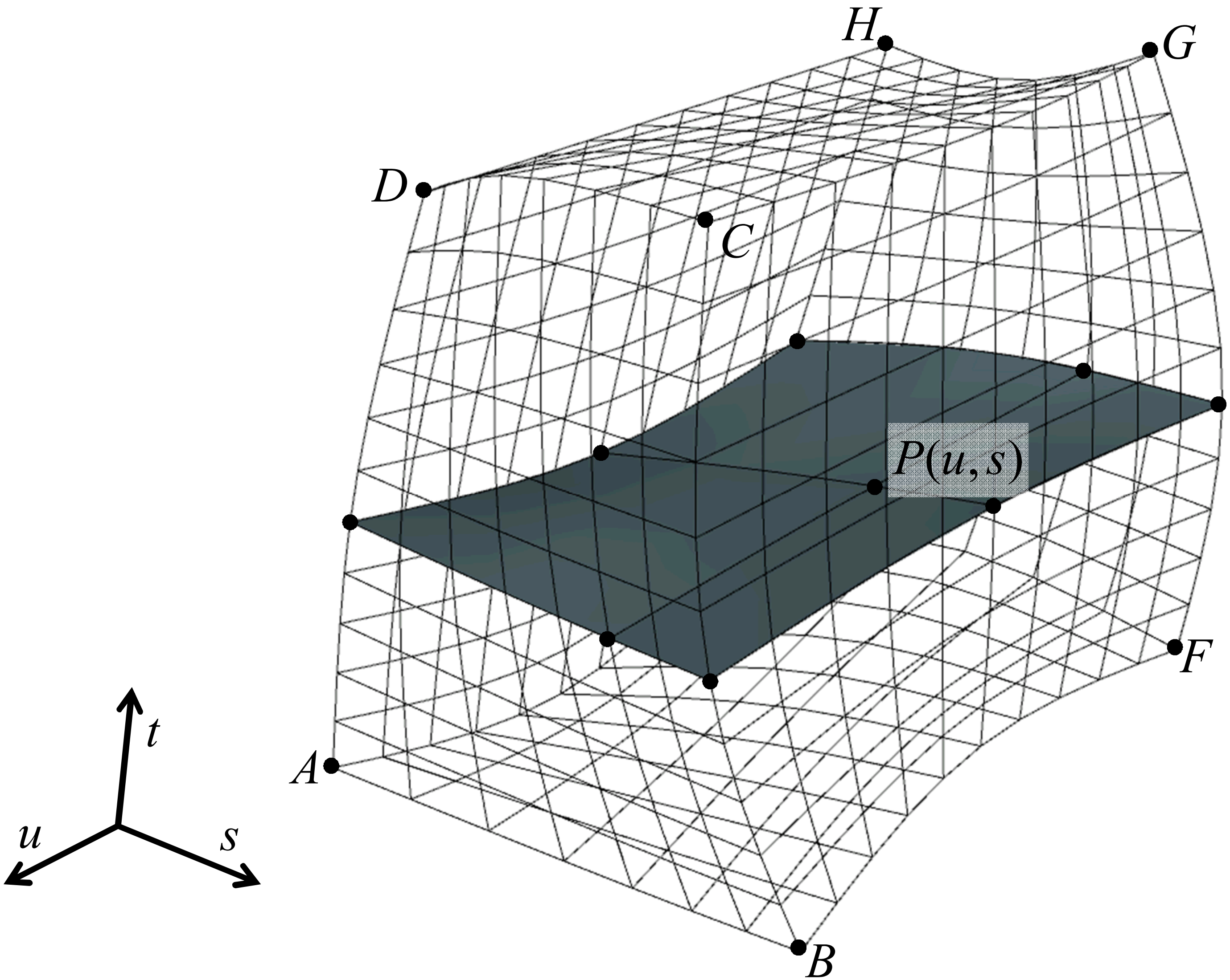
Transfinite Interpolation



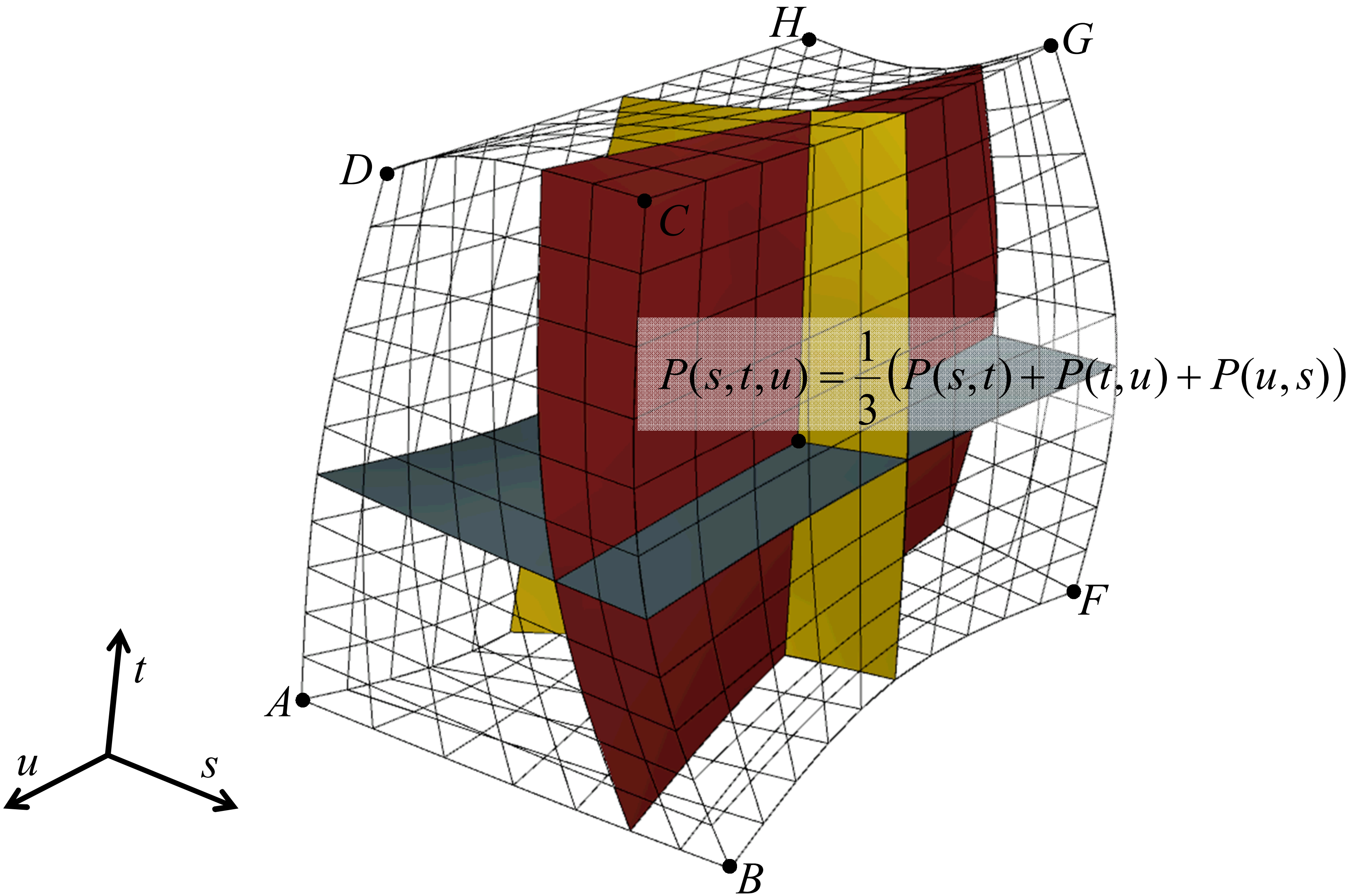
Transfinite Interpolation



Transfinite Interpolation

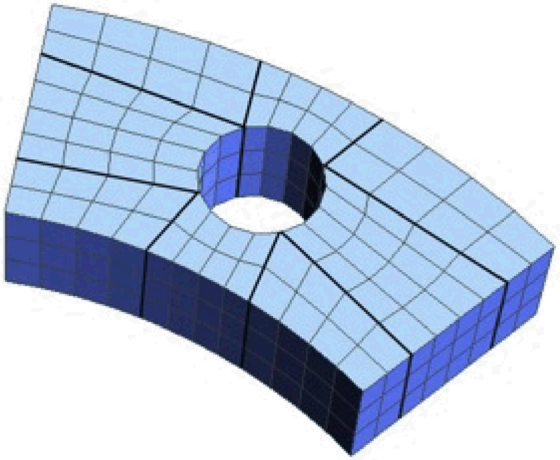


Transfinite Interpolation



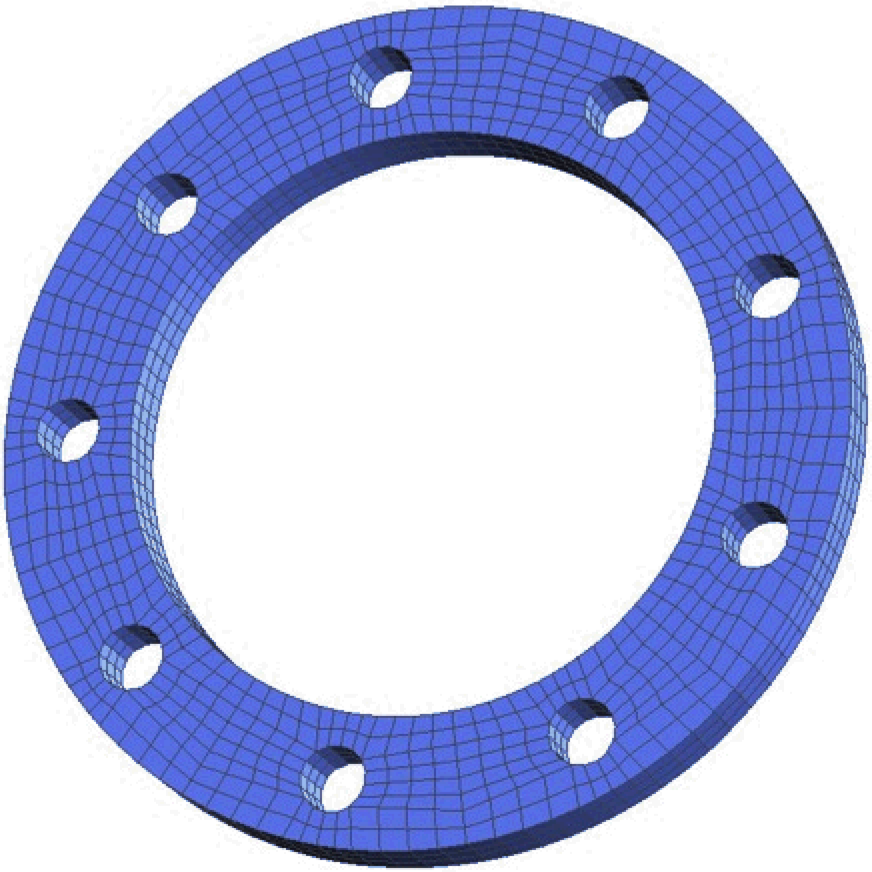
Block Structured Meshing

TrueGrid
http://www.truegrid.com



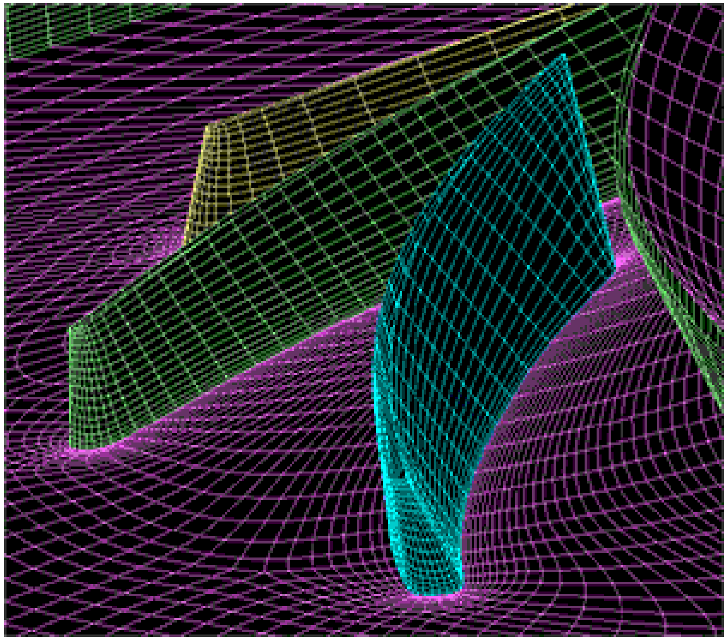
Geometry manually decomposed into 8 blocks (hexahedral regions)

Map mesh generated in each region

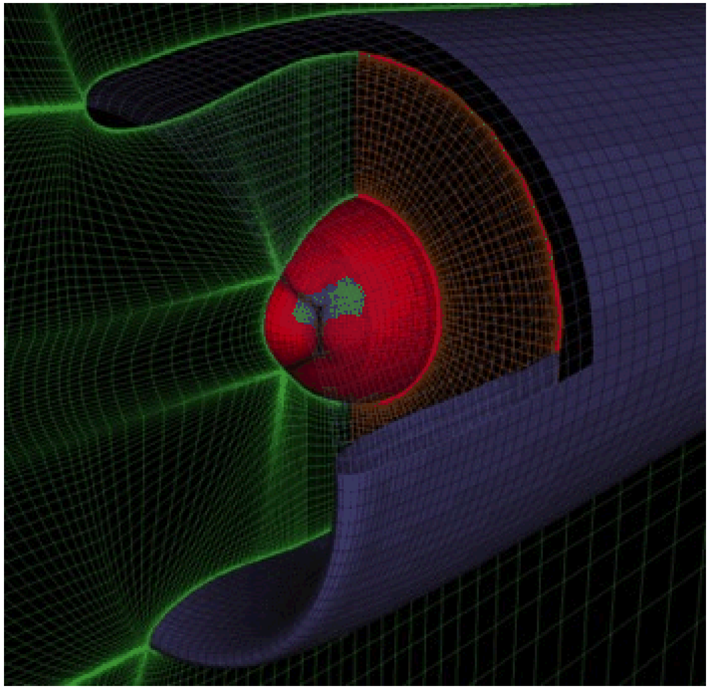


Total 72 blocks for this example

Block Structured Meshing



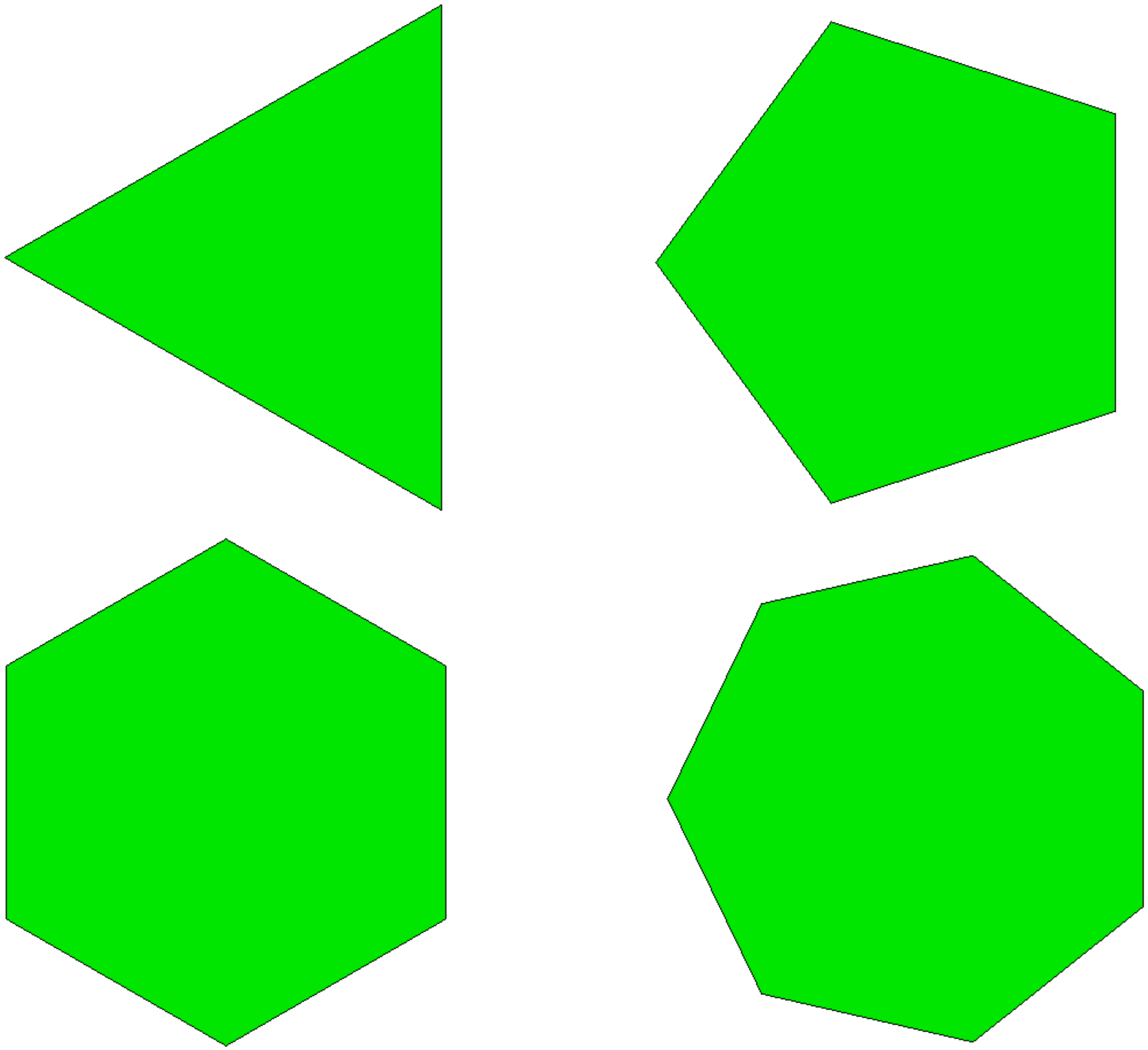
http://www.gridpro.com/gridgallery/tmachinery.html



http://www.pointwise.com/case/747.htm

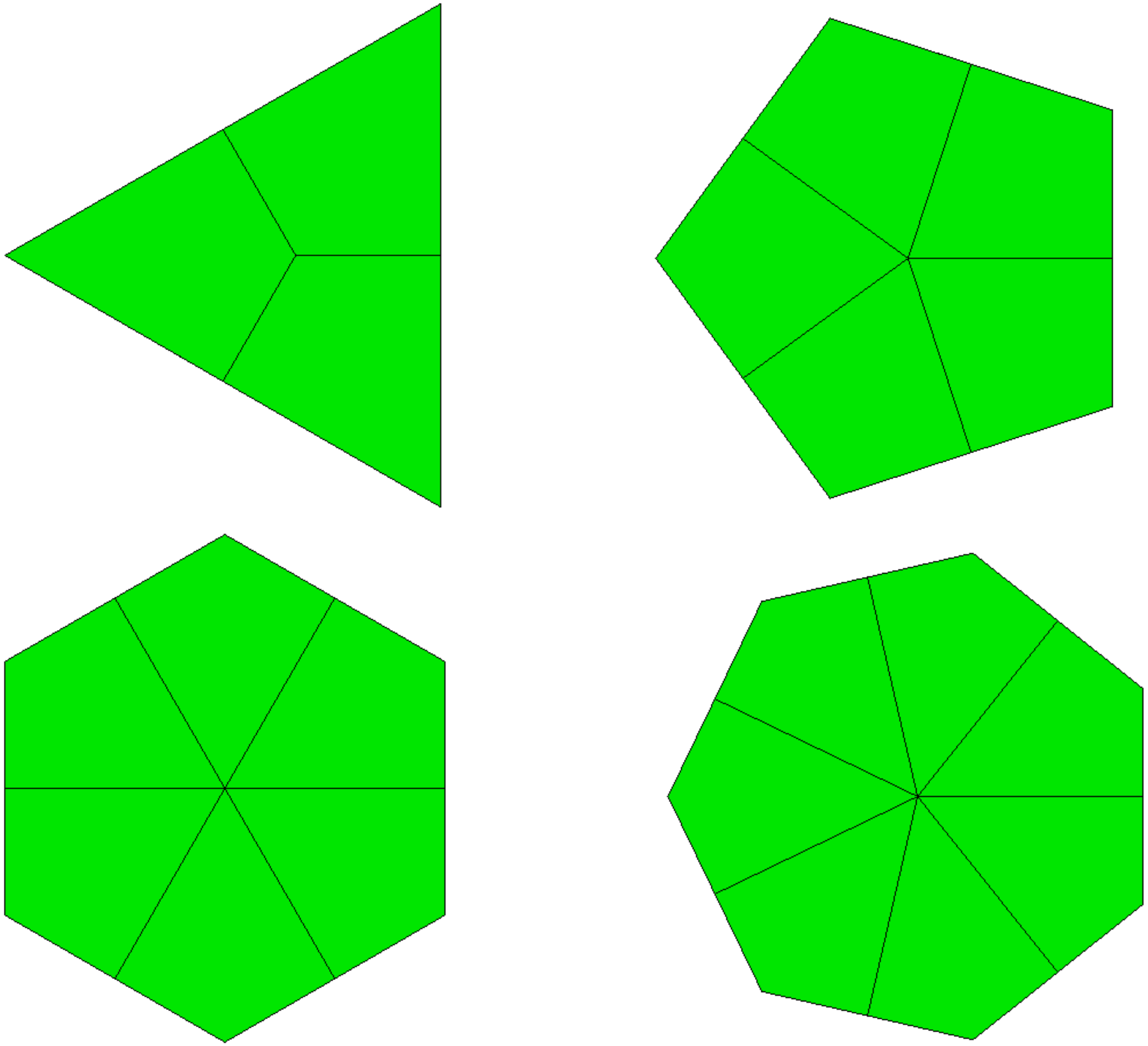
Many sophisticated tools available for interactive decomposition of geometry into blocks

Midpoint subdivision



Regular convex polygons

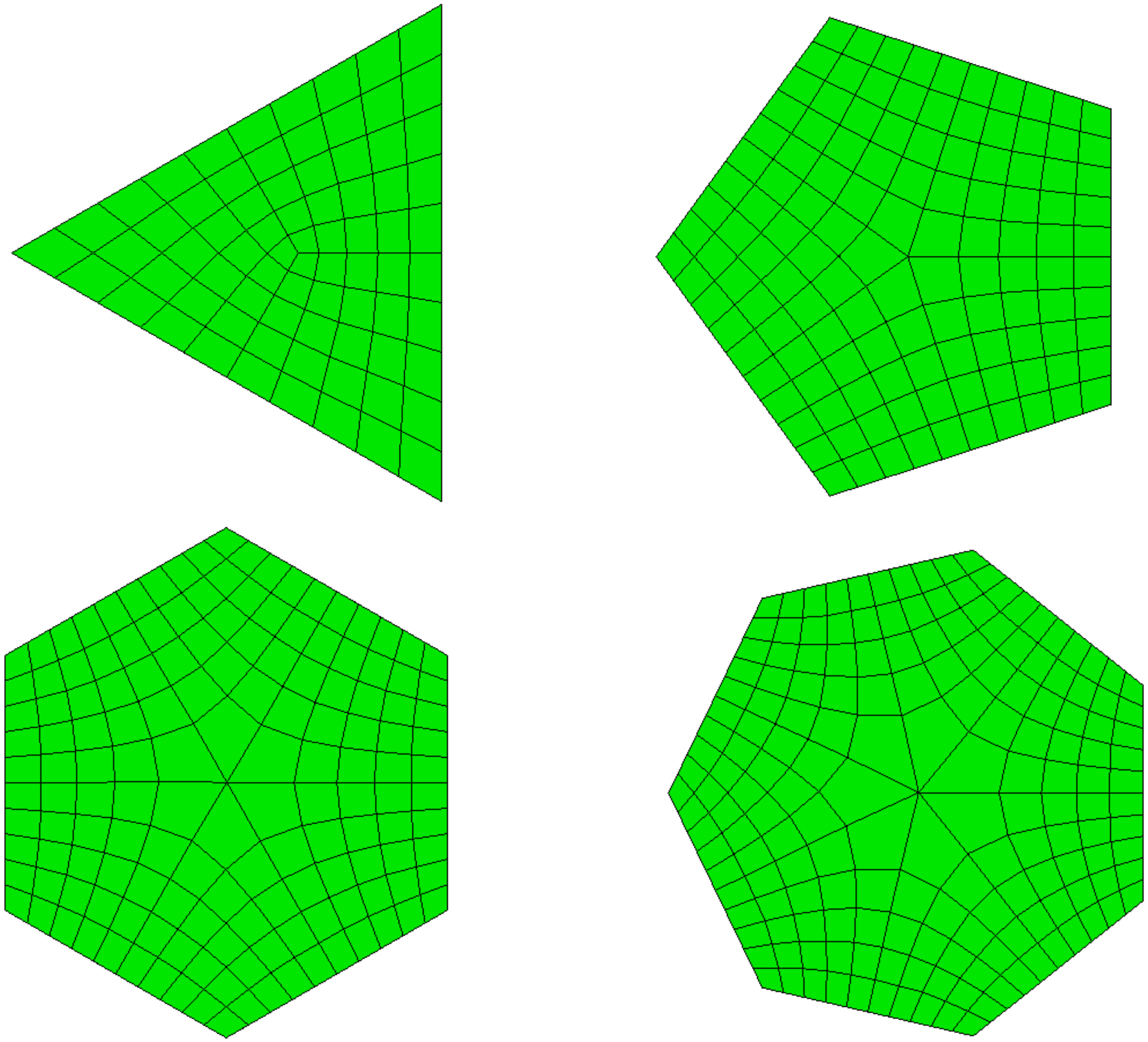
Midpoint subdivision



Regular convex polygons

Each side is subdivided and one node placed at the interior to create quadrilaterals

Midpoint subdivision



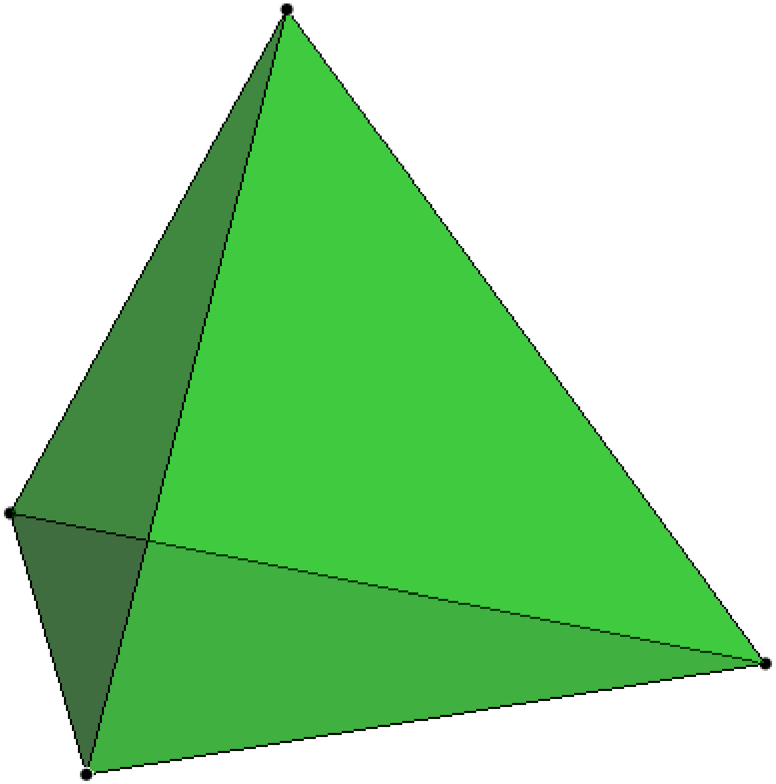
Regular convex polygons

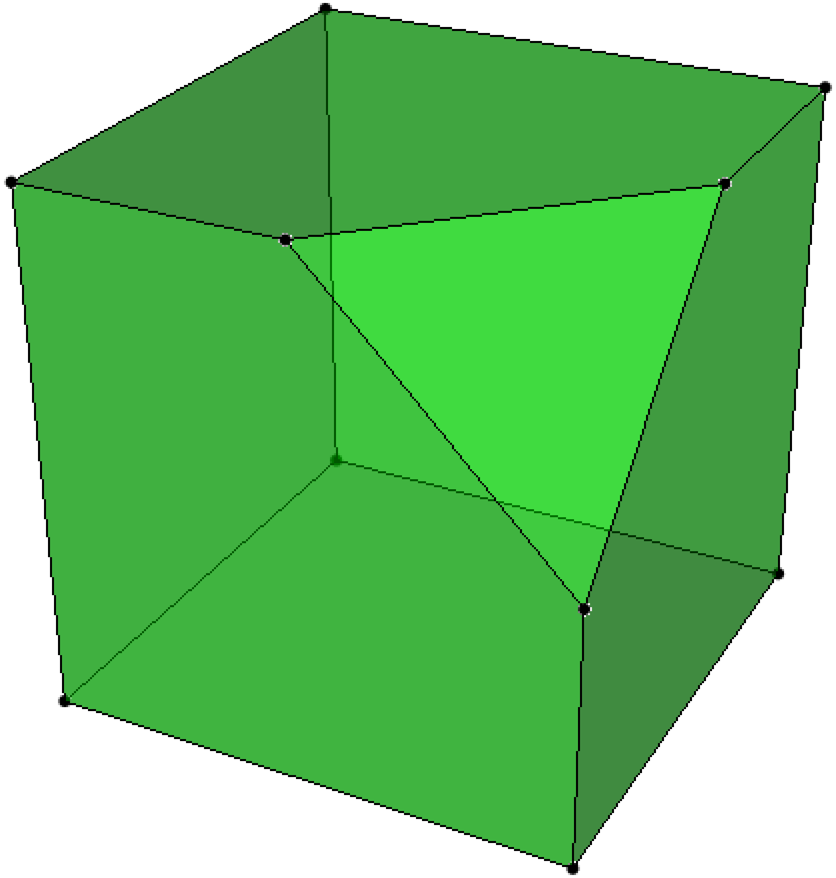
Each side is subdivided and one node placed at the interior to create quadrilaterals

Each individual quadrilateral is map meshed

Each side must have matching intervals

Midpoint subdivision

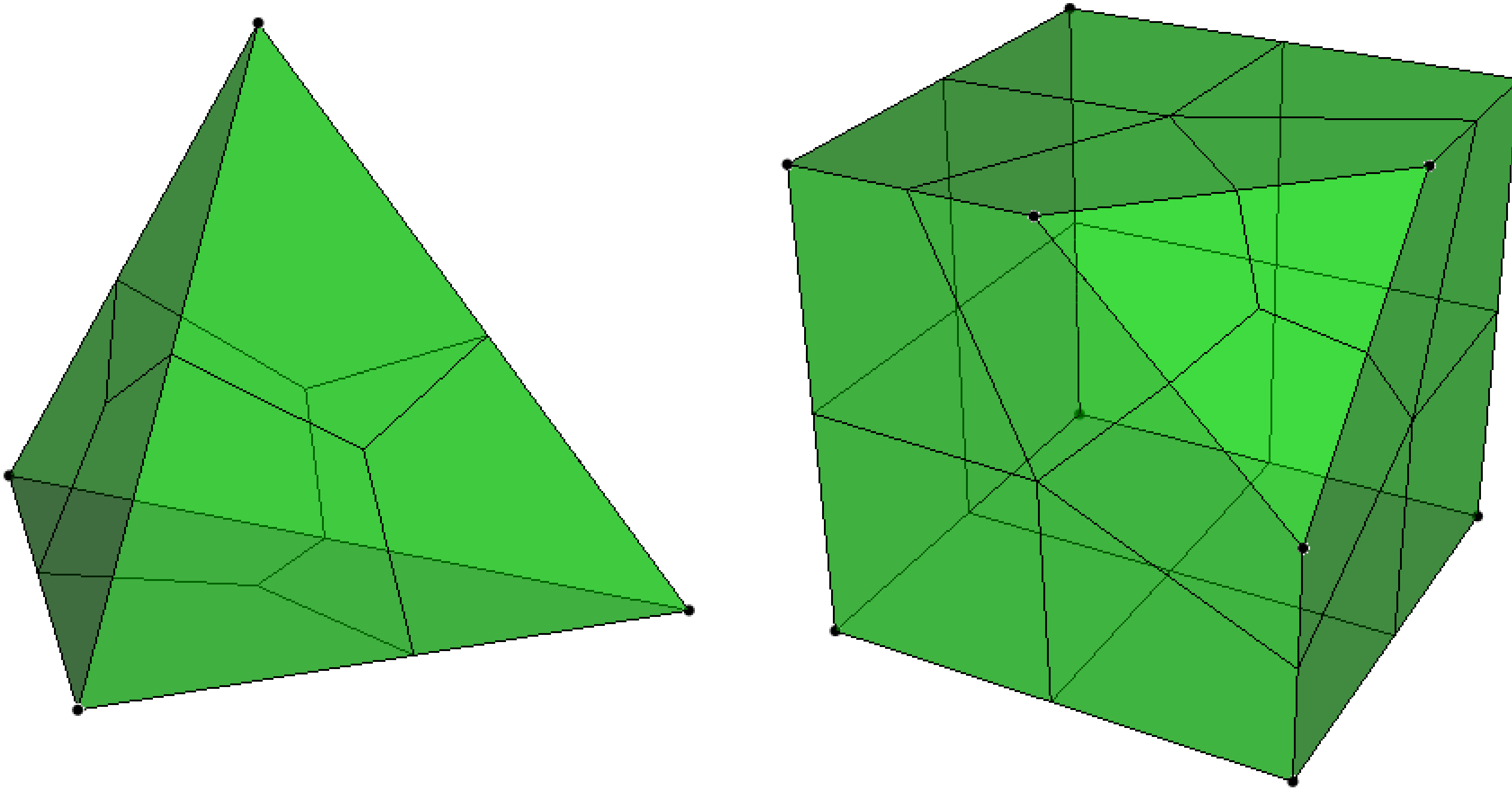




3D Midpoint subdision

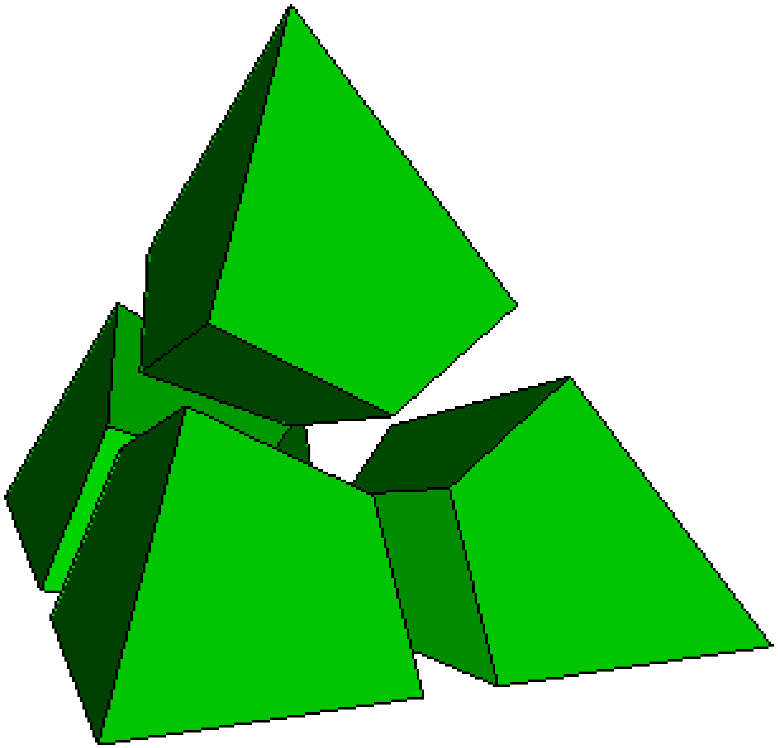
Requires 3-valent vertices and convex polyhedron

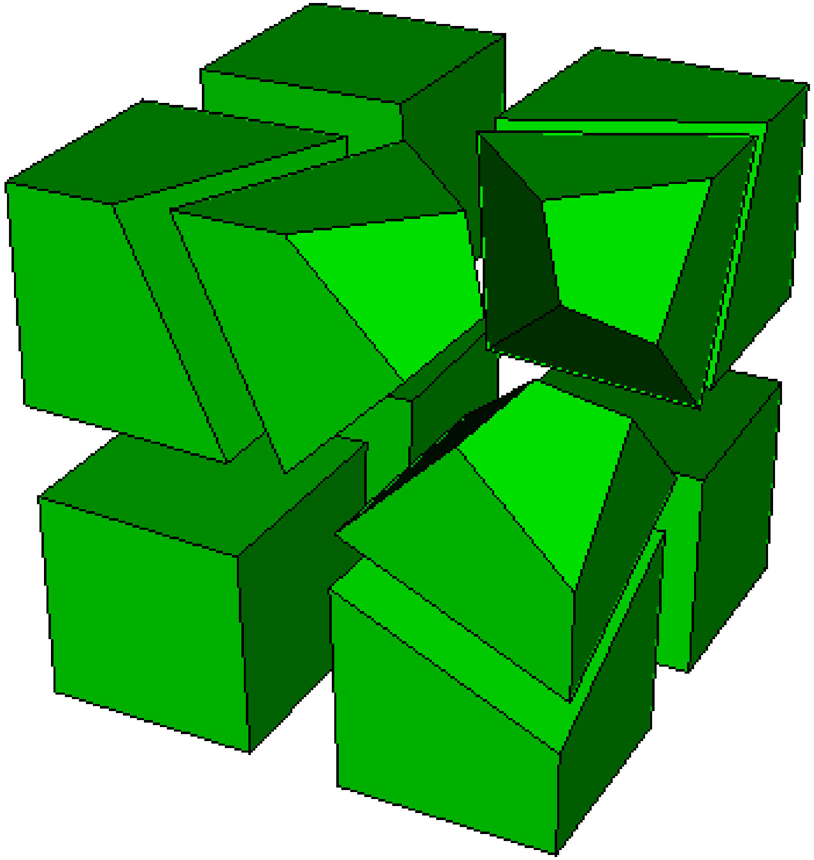
Midpoint subdivision



Sibdivide each surface into
quadrilaterals using 2D subdivision

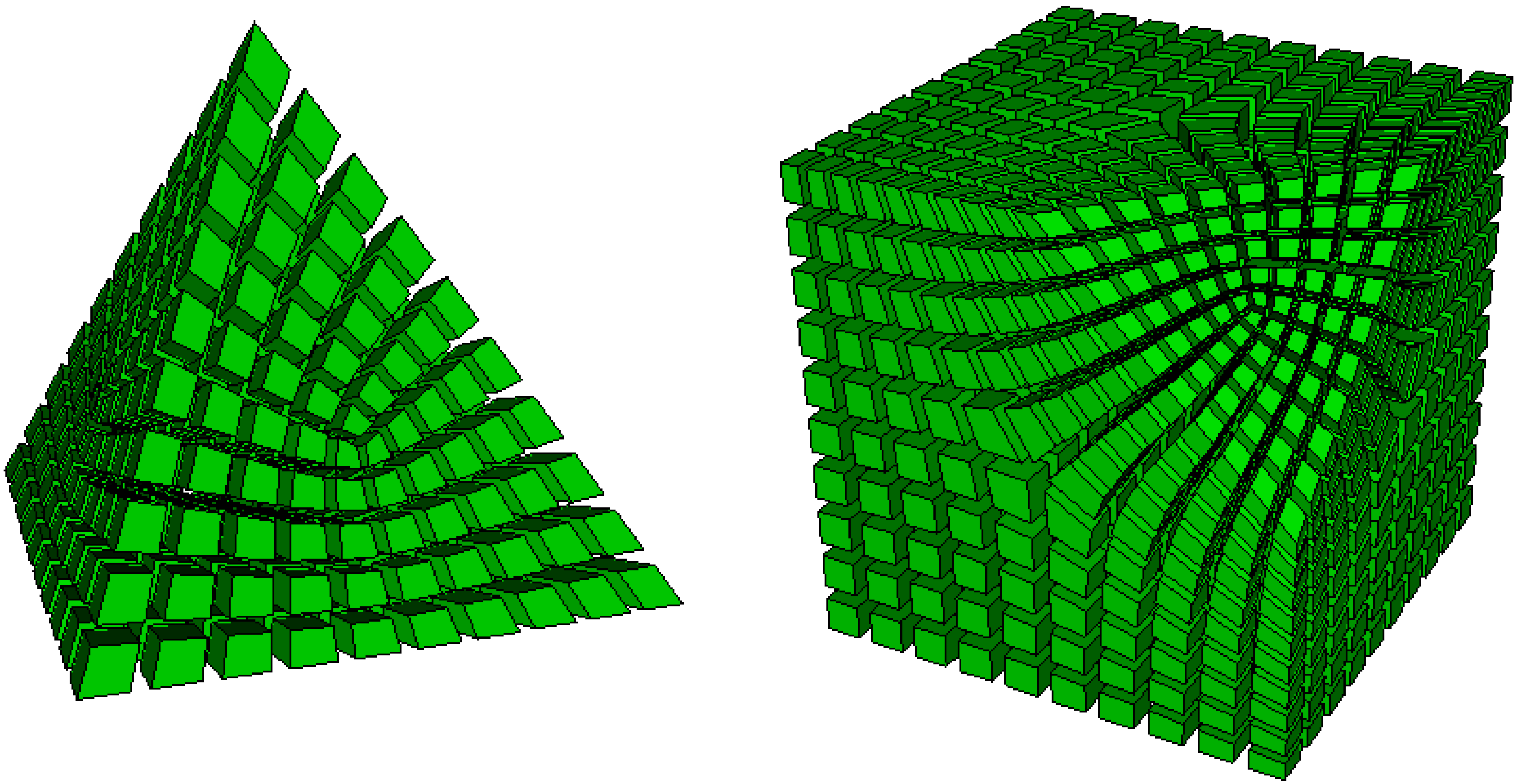
Midpoint subdivision





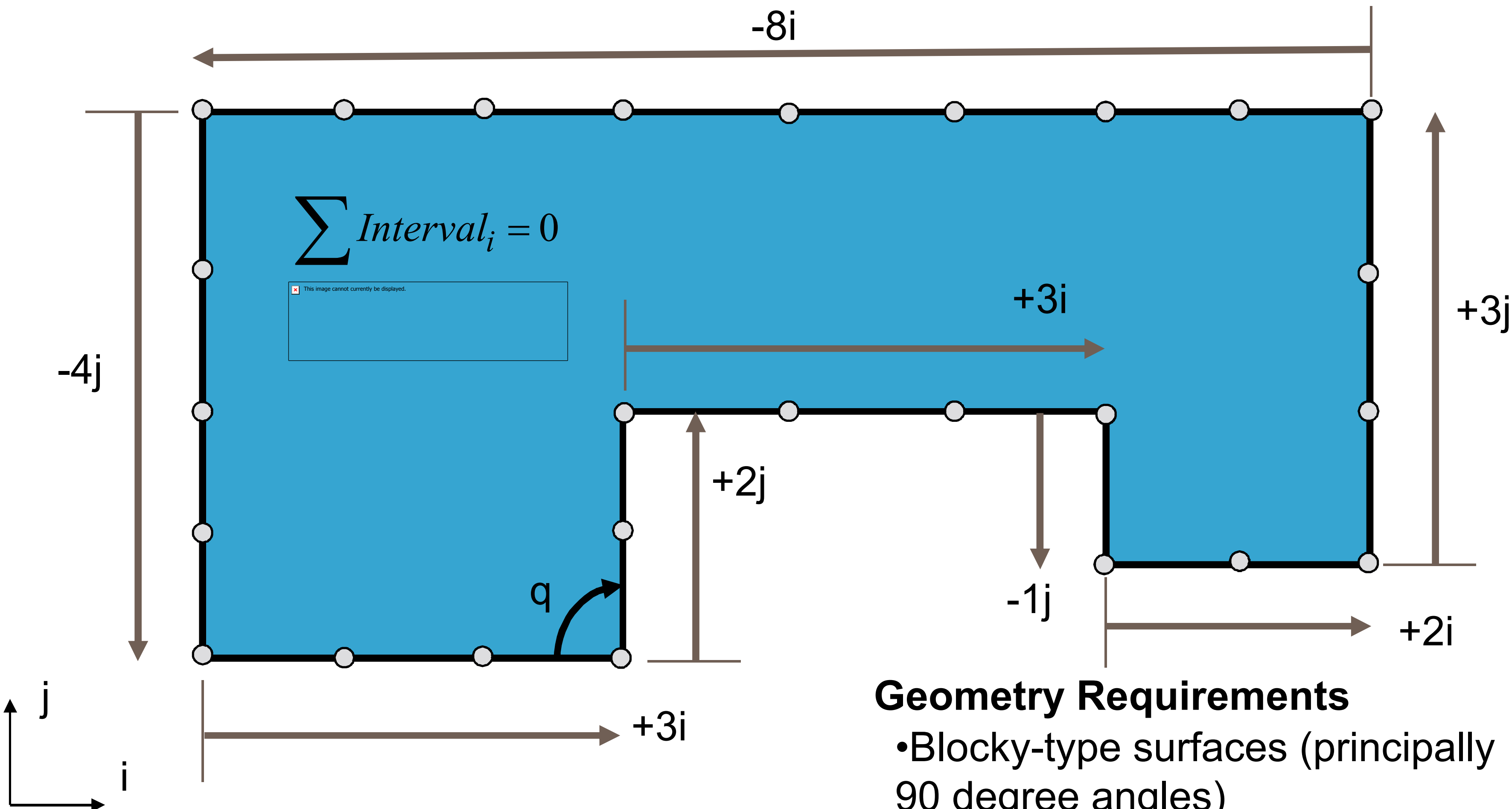
Add interior node and generate hexahedral regions.
3 hex faces surrounding boundary vertices and 3 faces at the interior node

Midpoint subdivision



Map Mesh each of the hexahedral regions
Ensure curve and surface intervals match between regions

Sub-mapping

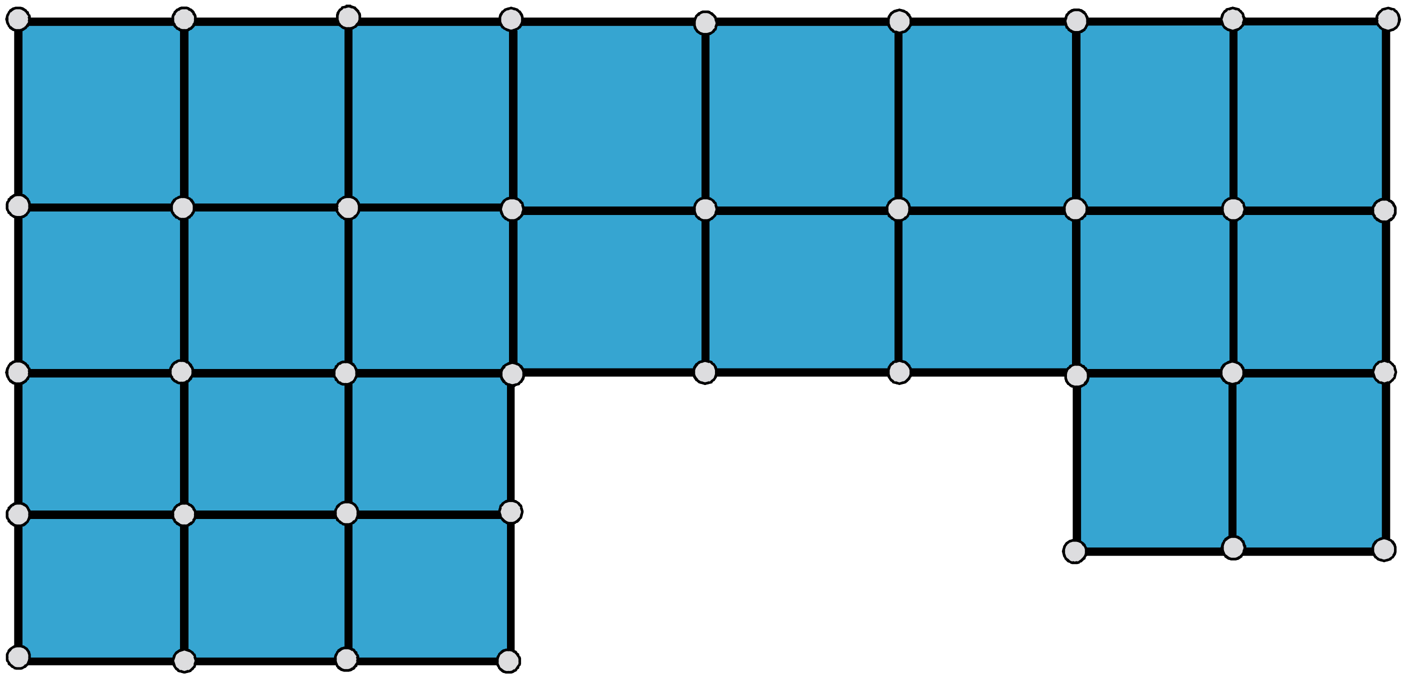


Geometry Requirements

- Blocky-type surfaces (principally 90 degree angles)

(White,95)

Sub-mapping



•Automatically decomposes surface into mappable regions based on assigned intervals

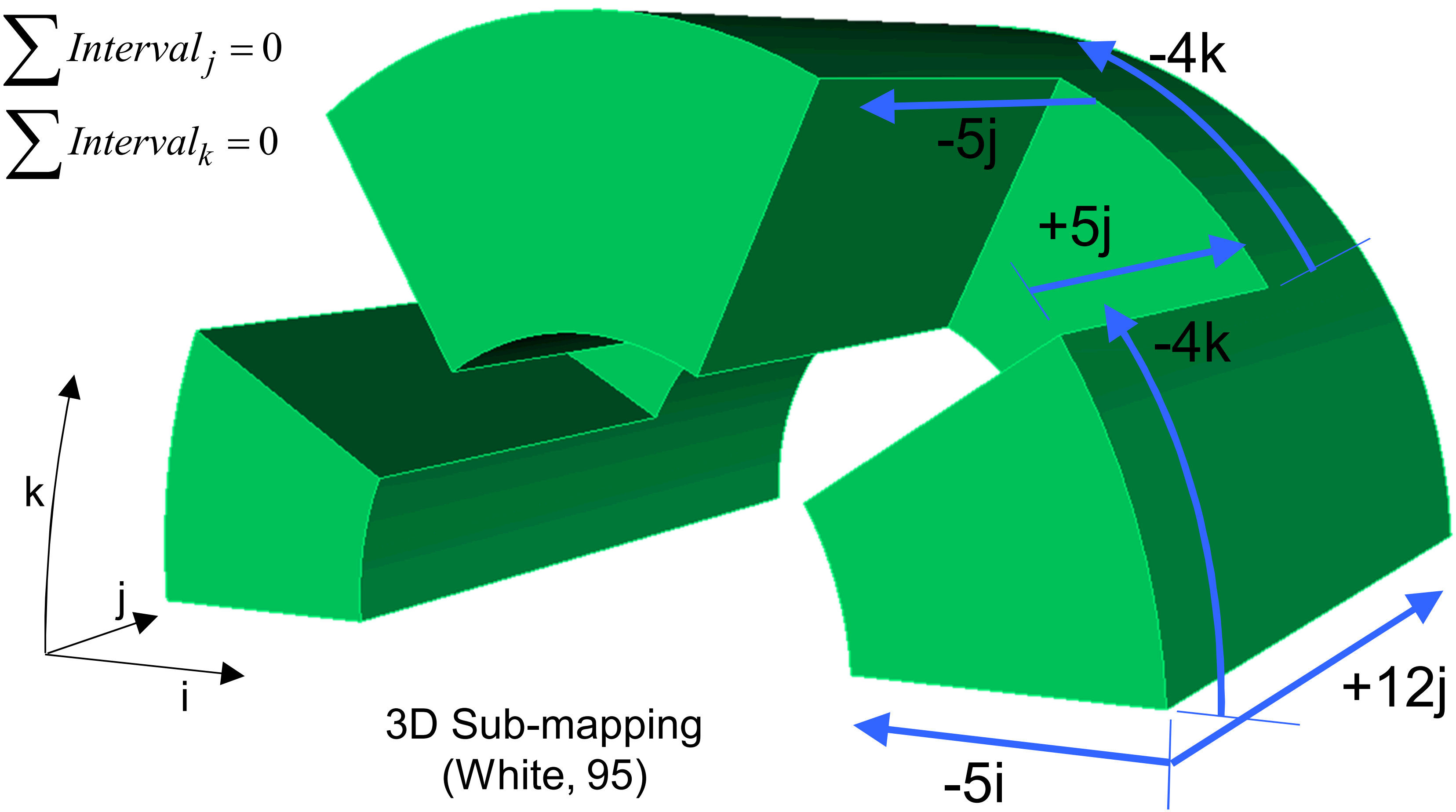
(White,95)

Sub-mapping

$$\sum Interval_i = 0$$

$$\sum Interval_j = 0$$

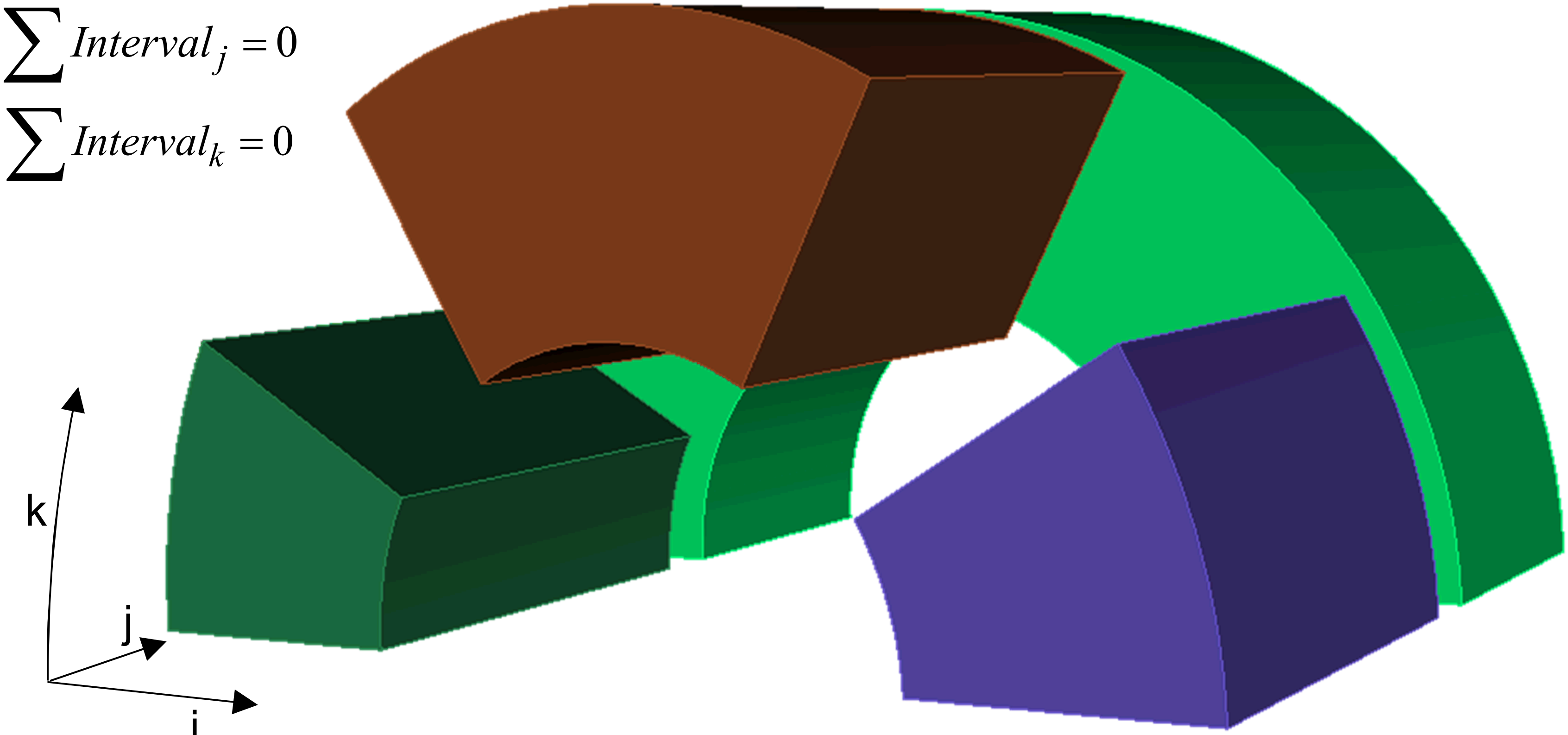
$$\sum Interval_k = 0$$



3D Sub-mapping
(White, 95)

Sub-mapping

$$\sum Interval_i = 0$$
$$\sum Interval_j = 0$$
$$\sum Interval_k = 0$$



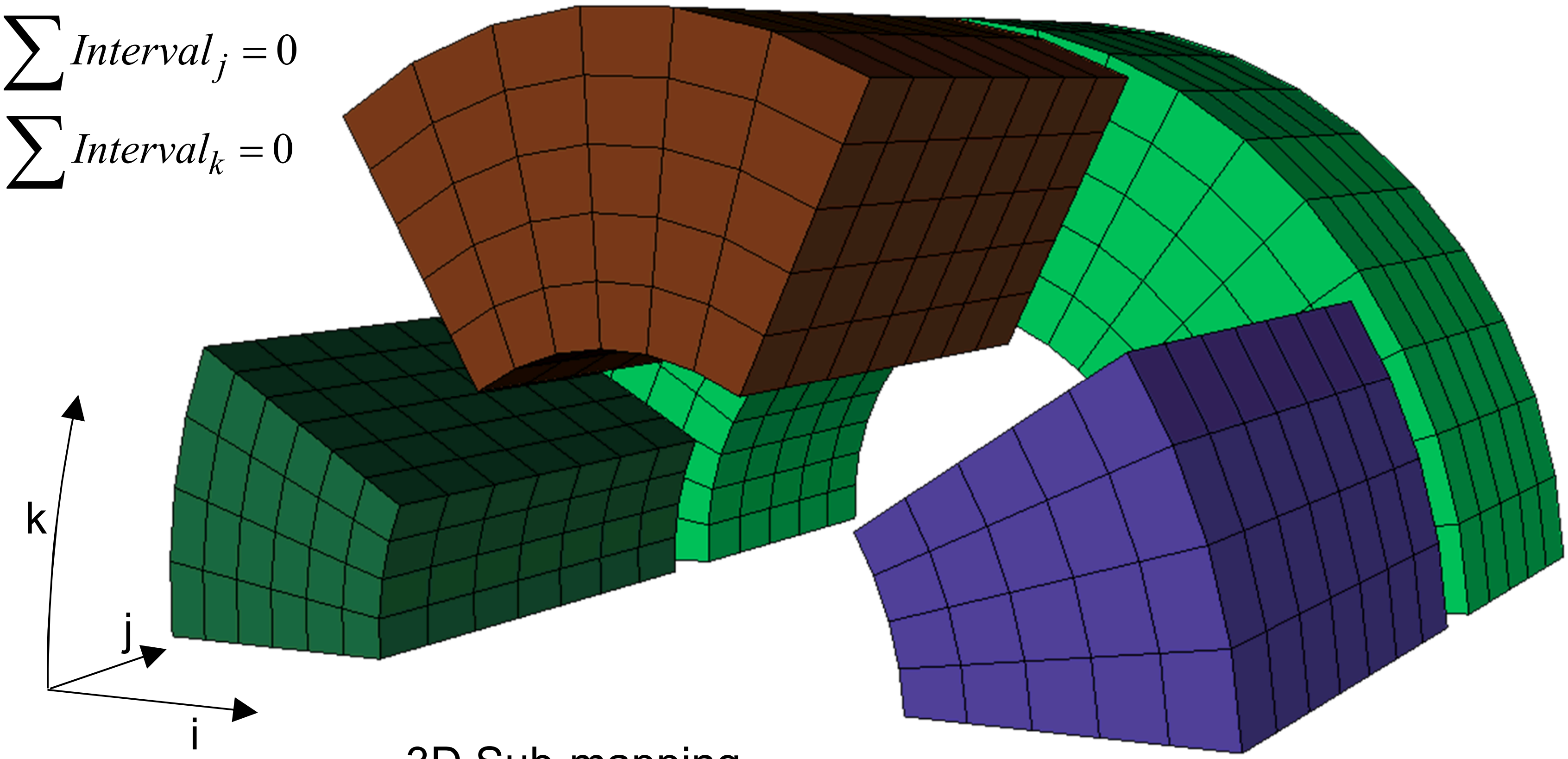
3D Sub-mapping
(White, 95)

Sub-mapping

$$\sum Interval_i = 0$$

$$\sum Interval_j = 0$$

$$\sum Interval_k = 0$$



3D Sub-mapping
(White, 95)

Sub-mapping

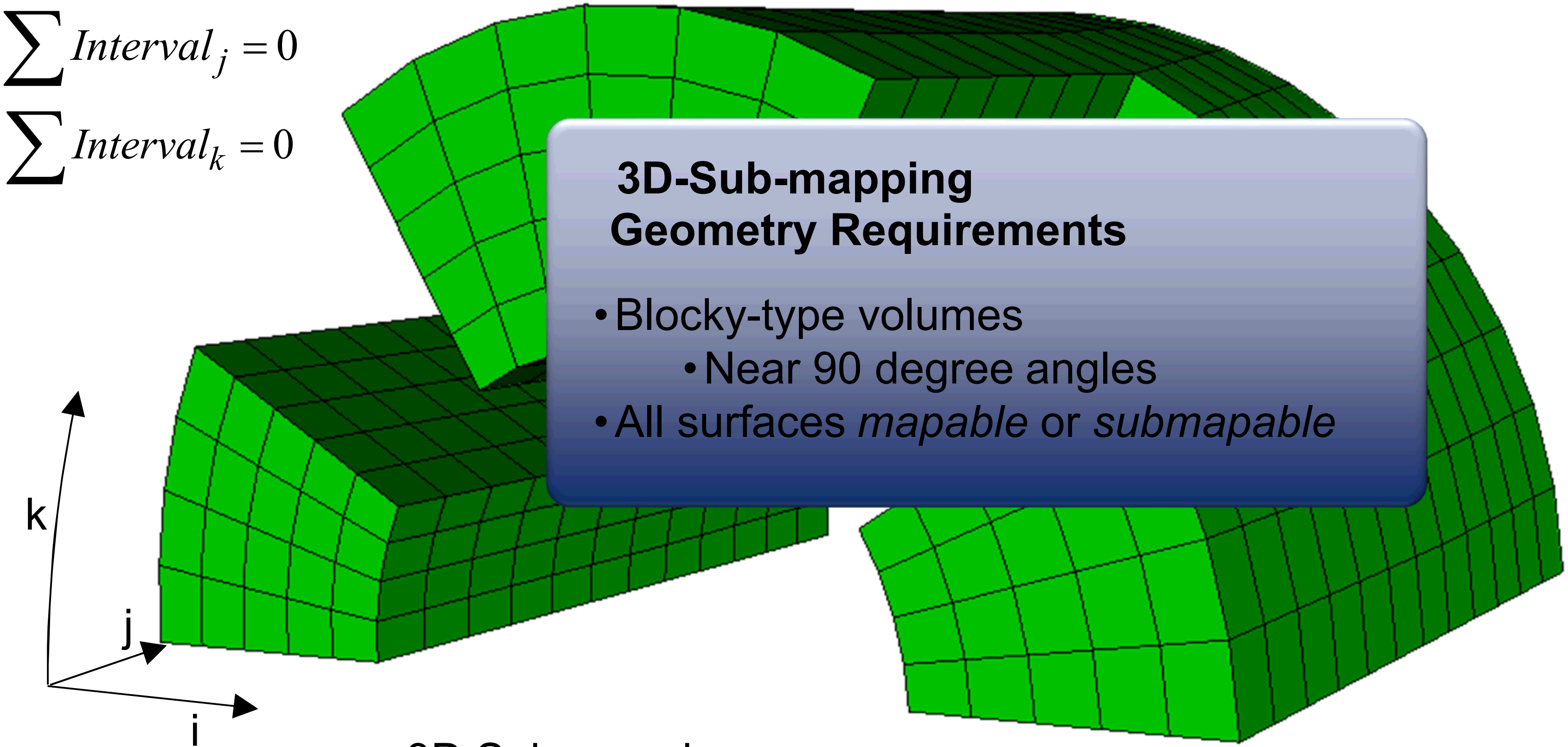
$$\sum Interval_i = 0$$

$$\sum Interval_j = 0$$

$$\sum Interval_k = 0$$

3D-Sub-mapping Geometry Requirements

- Blocky-type volumes
    - Near 90 degree angles
- All surfaces *mapable* or *submapable*

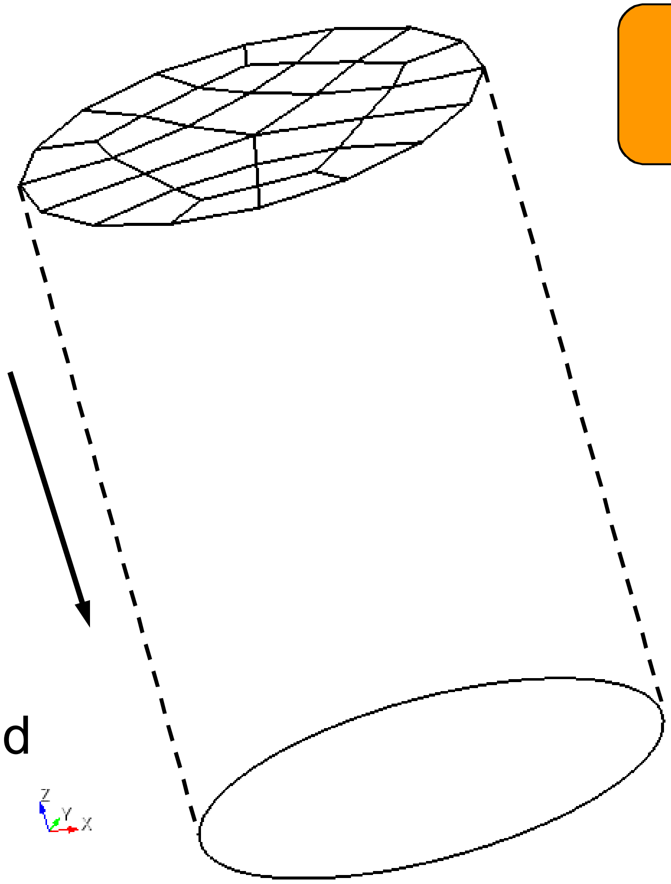


3D Sub-mapping
(White, 95)

Sweeping

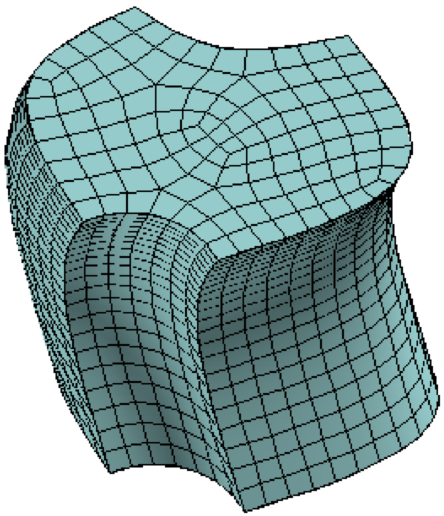
Current CUBIT Capability

Sweep Direction



Source surface is meshed with all quad mesh

1-to-1 sweepable



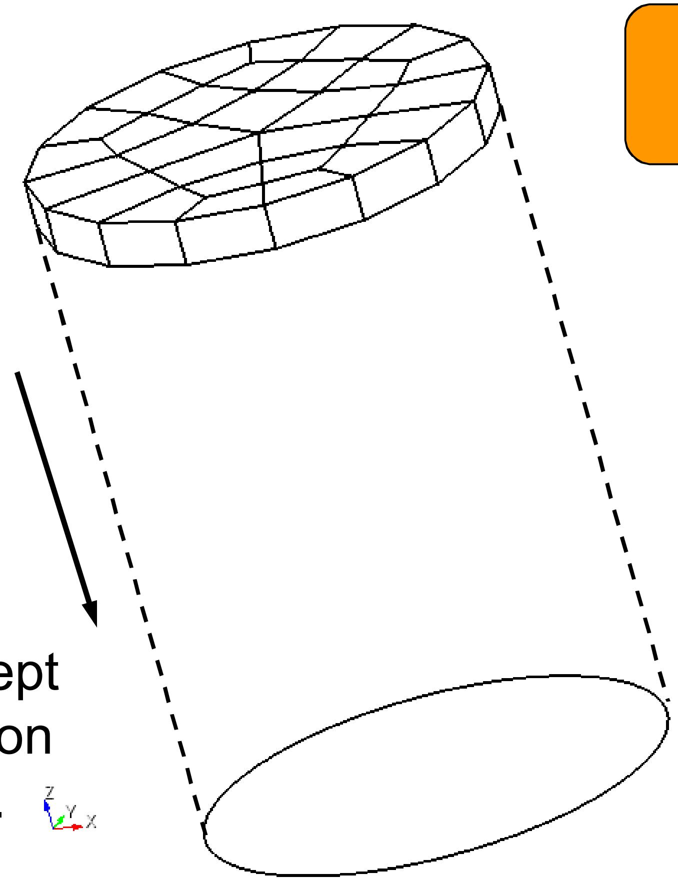
Matt Staten

Sweeping

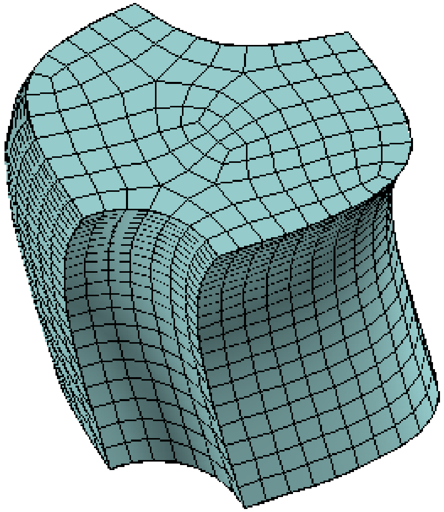
Current CUBIT Capability

Sweep Direction

Source mesh is swept along sweep direction towards the targets.



1-to-1 sweepable



Matt Staten

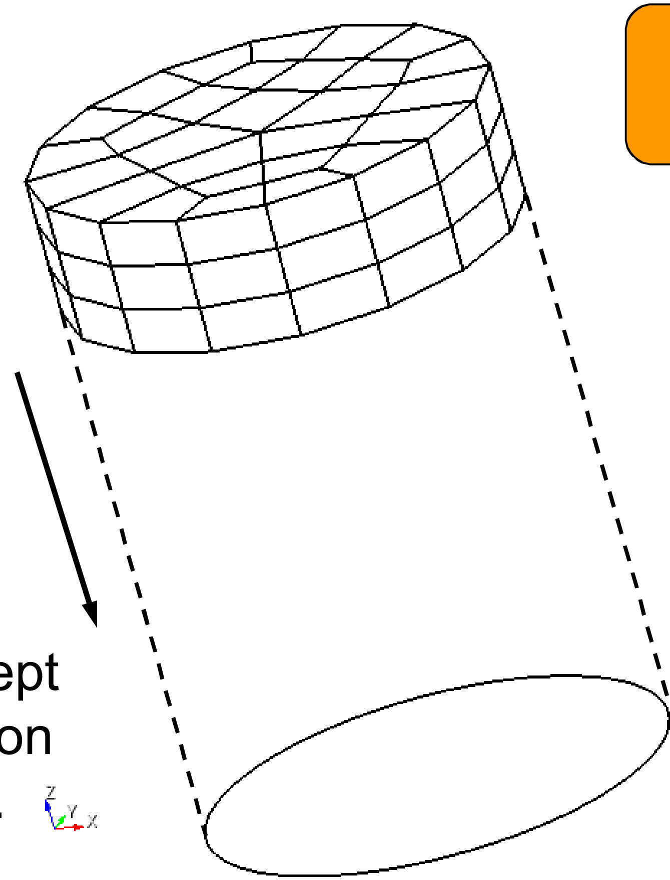
Sweeping

Sandia National Laboratories

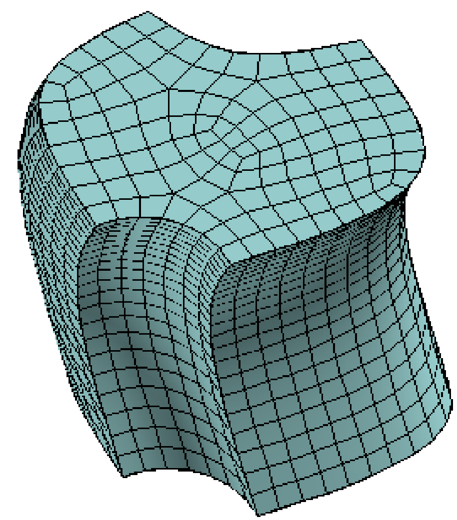
Current CUBIT Capability

1-to-1 sweepable

Sweep Direction



Source mesh is swept along sweep direction towards the targets.



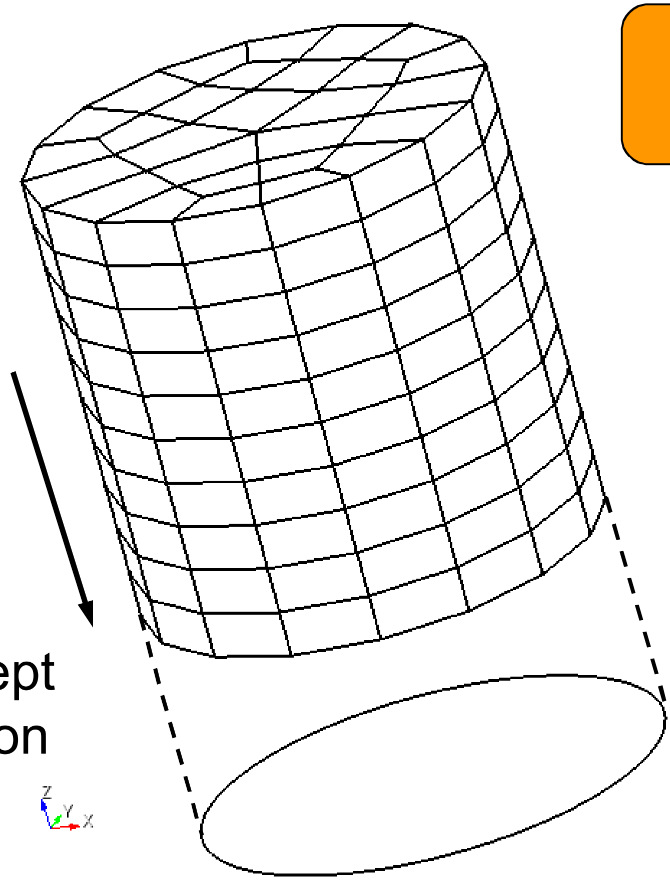
Matt Staten

Sweeping

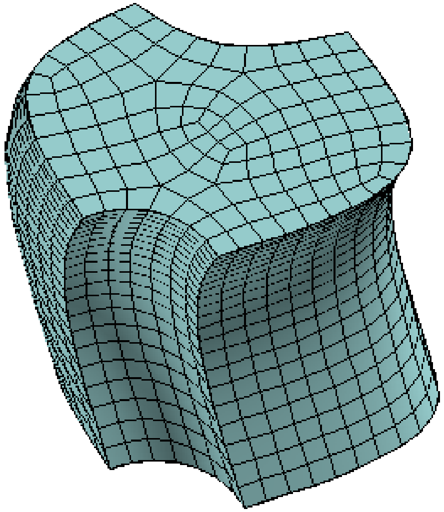
Current CUBIT Capability

1-to-1 sweepable

Sweep Direction



Source mesh is swept along sweep direction towards the targets.



Matt Staten

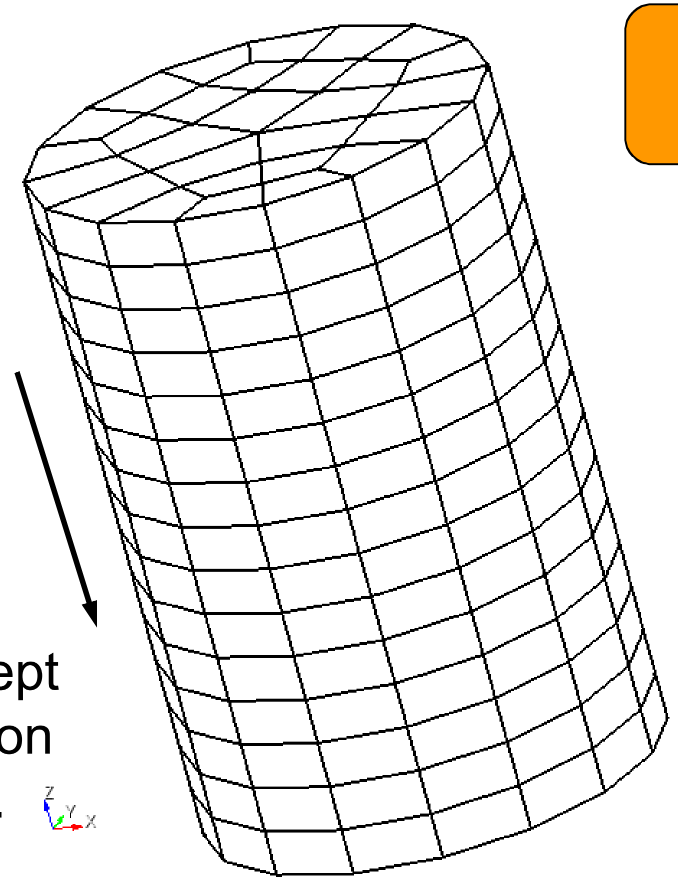
Sweeping

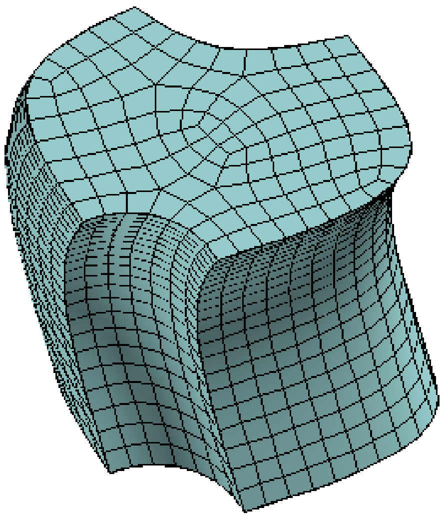
Current CUBIT Capability

1-to-1 sweepable

Sweep Direction

Source mesh is swept along sweep direction towards the targets.



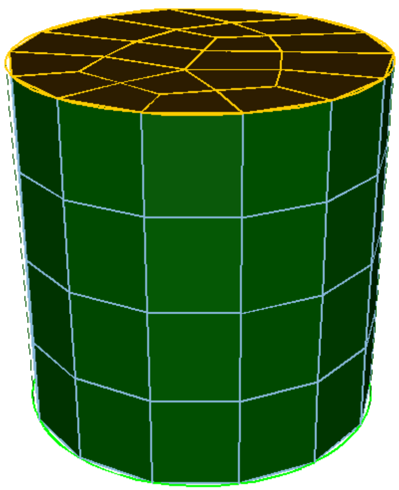


Matt Staten

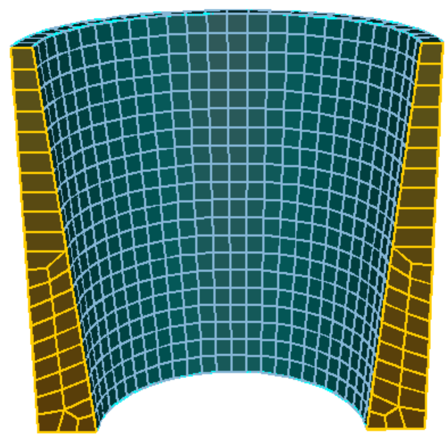
Sweeping

Typical one-to-one sweeps

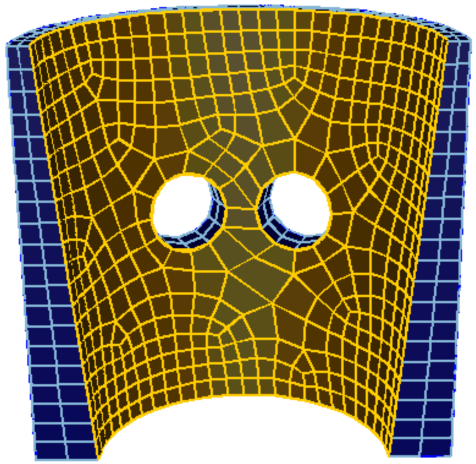
translation

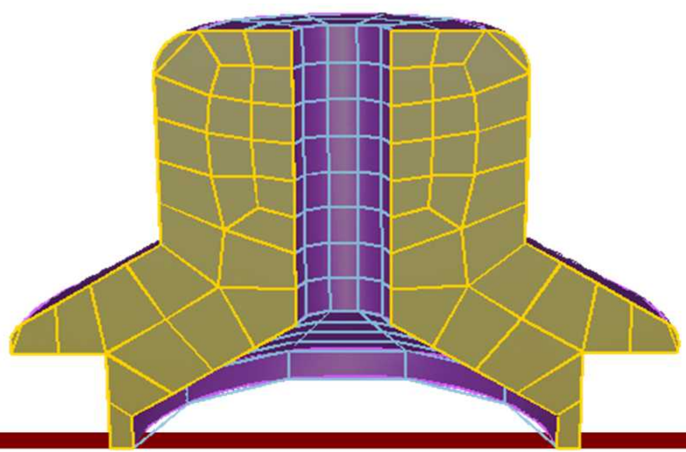


rotation



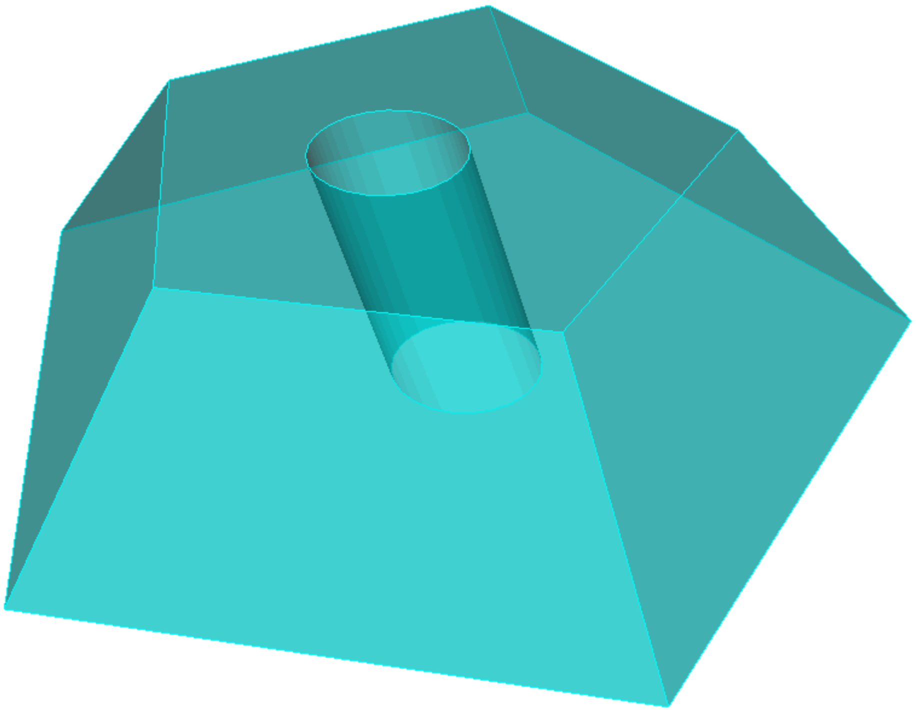
inside-out

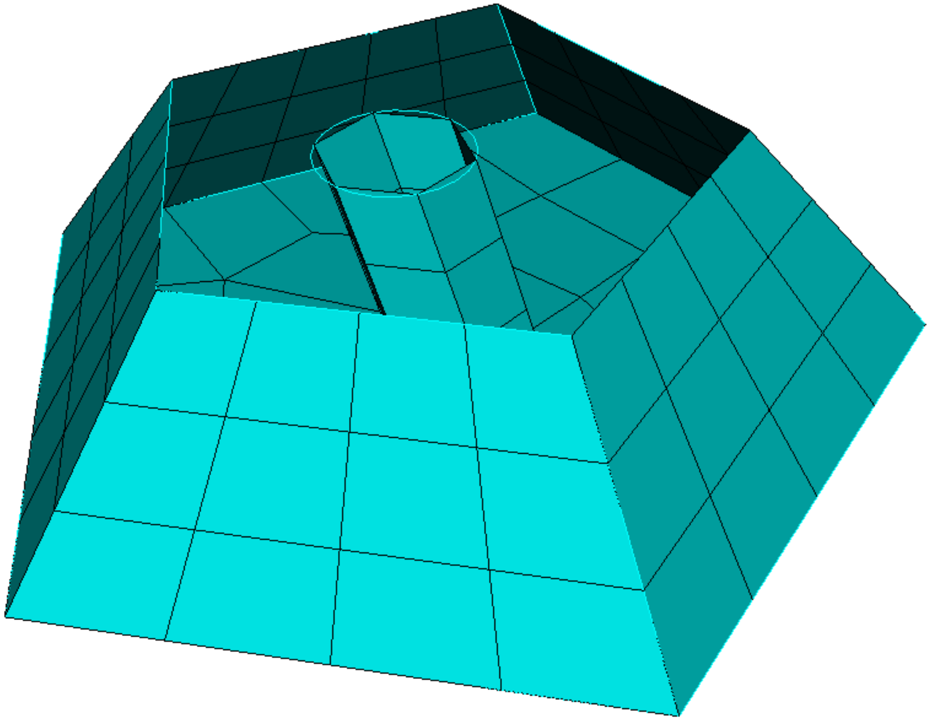




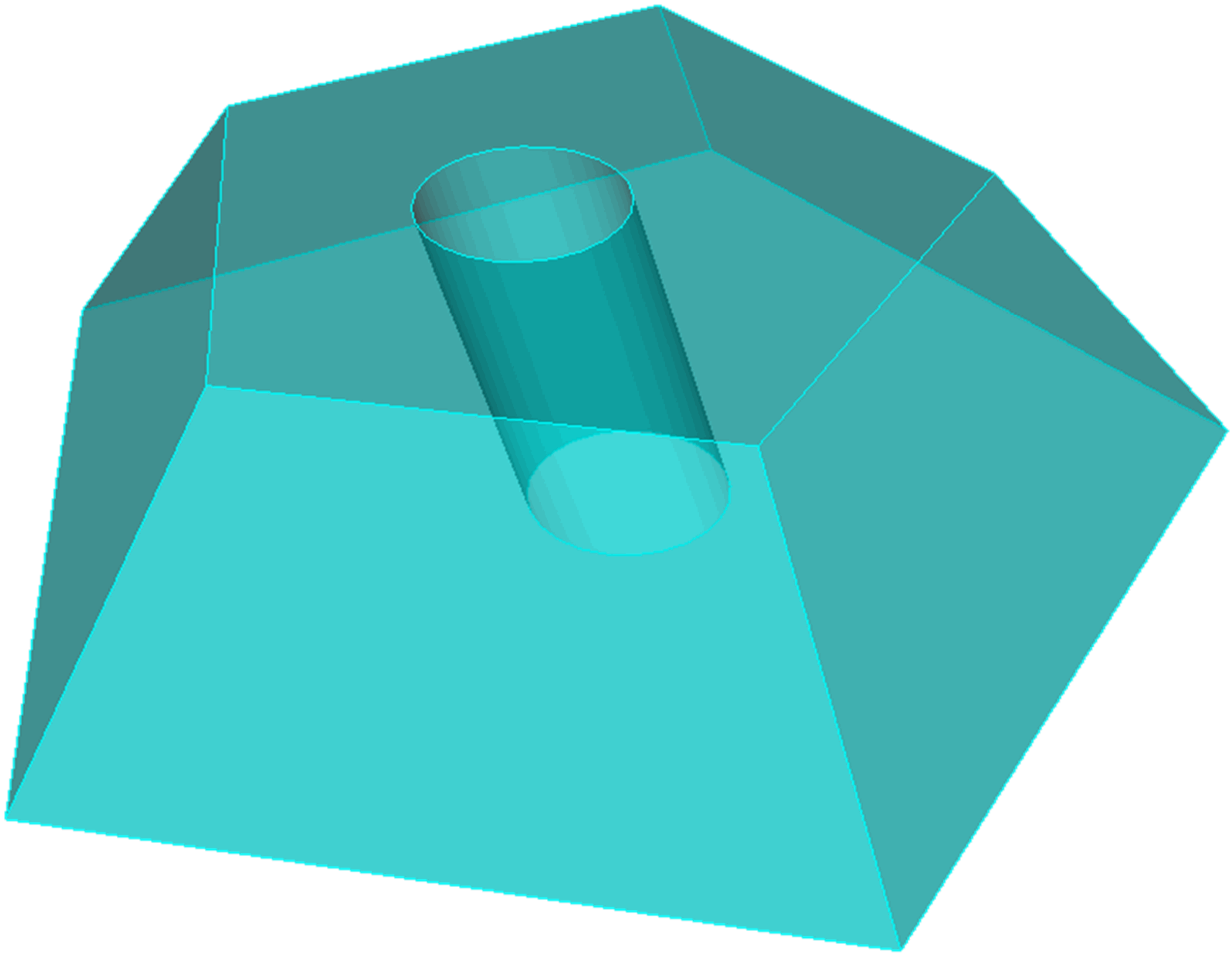
Sweeping

An input volume that is 2.5D, with source and all wall faces meshed.

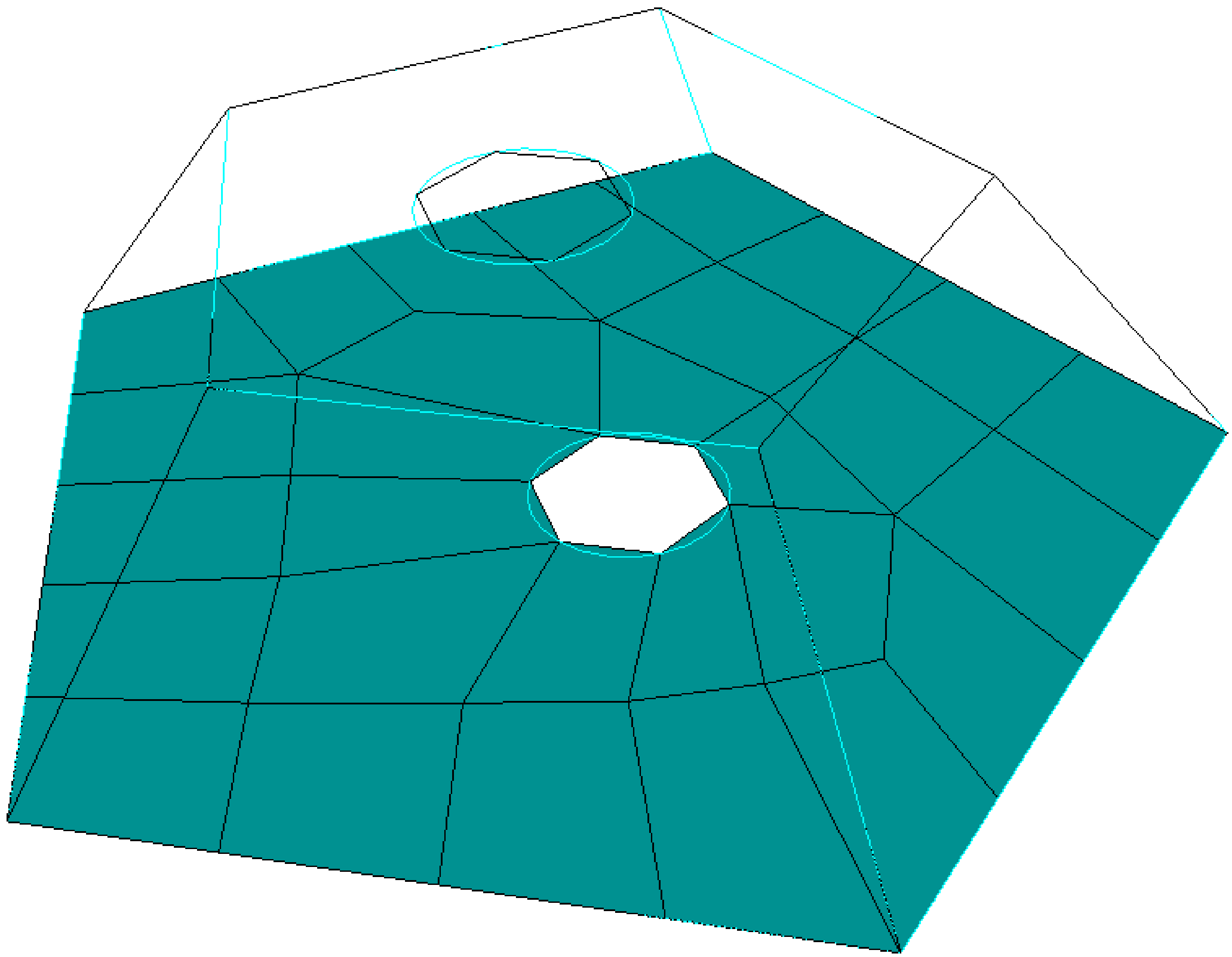




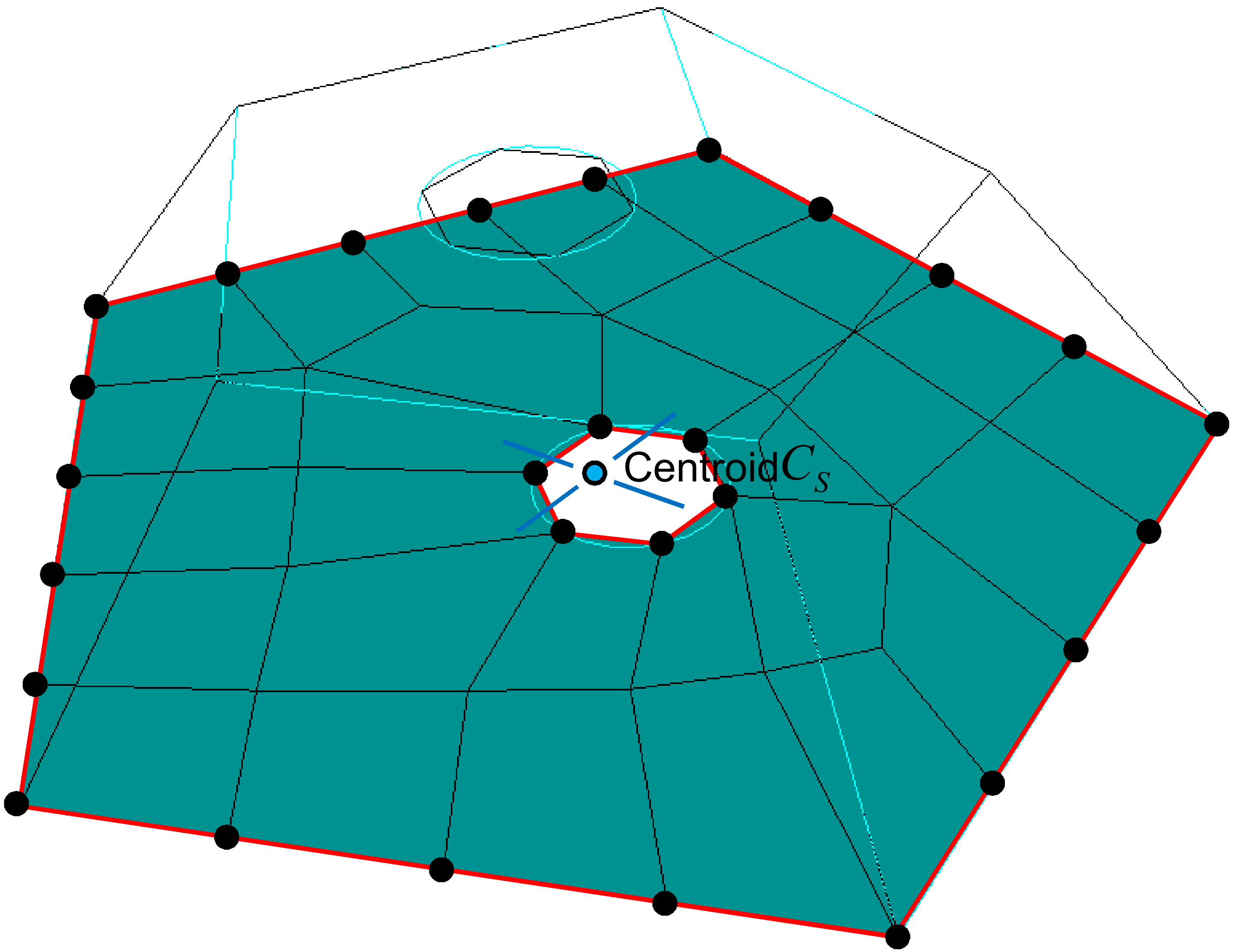
Sweeping



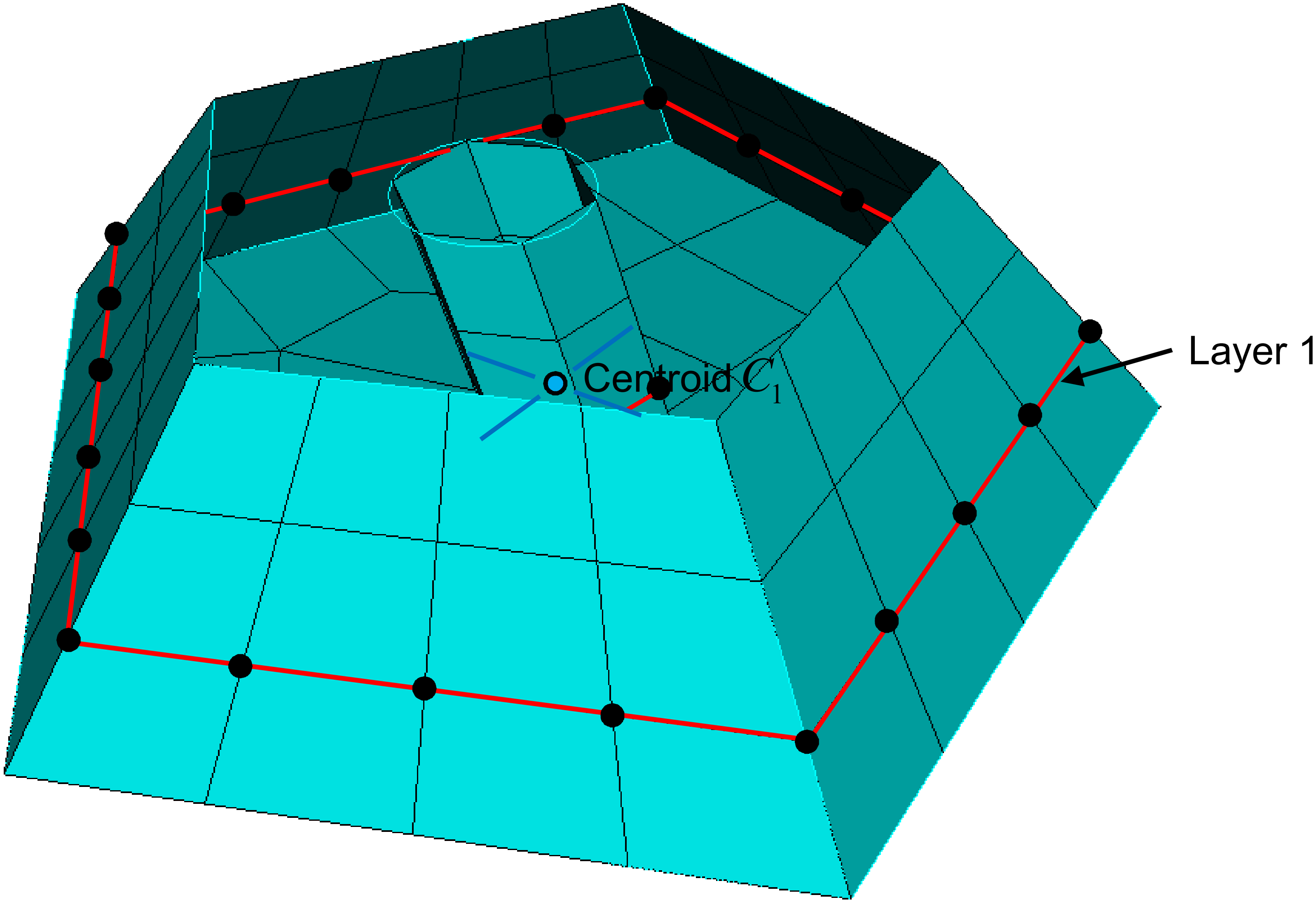
Sweeping



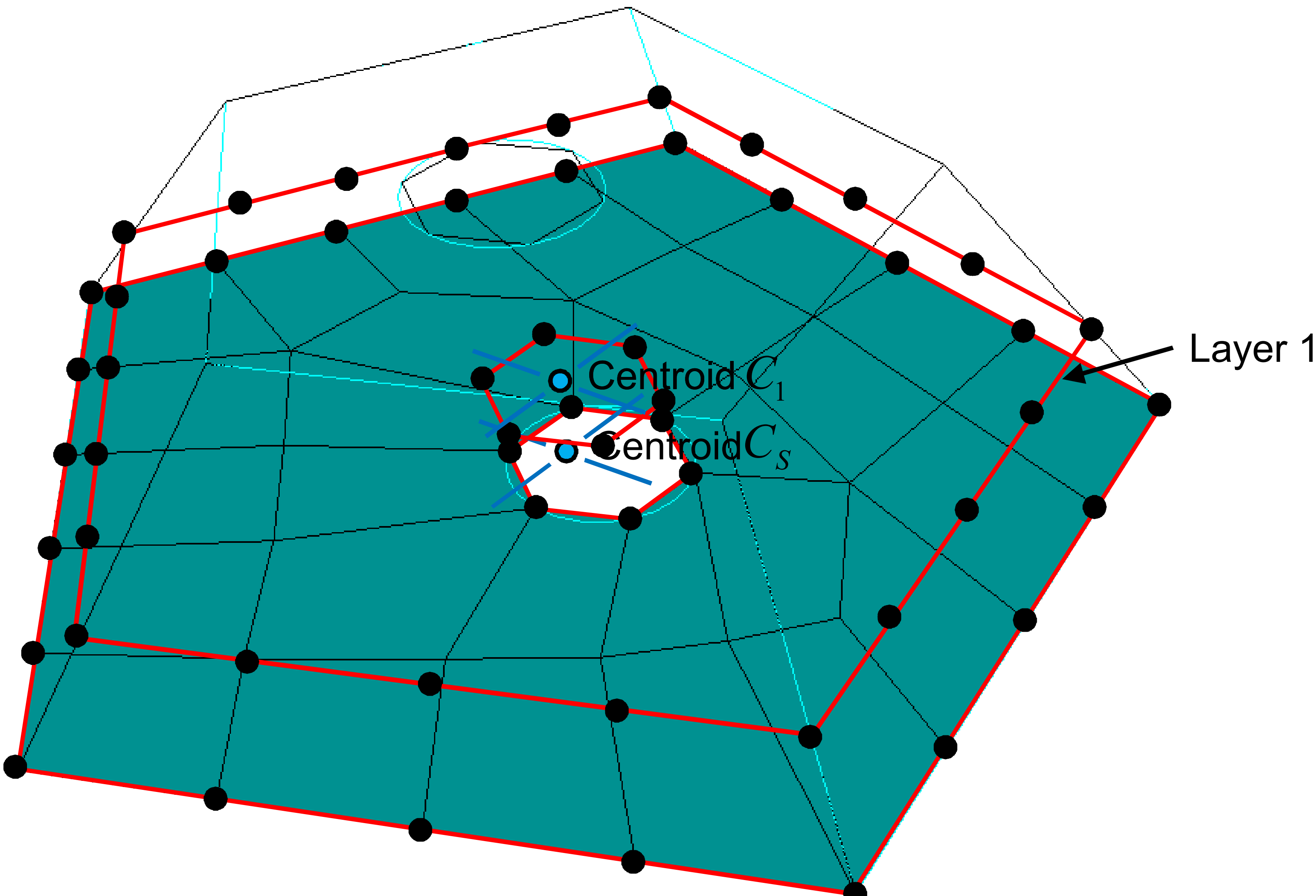
Weighted Residual Method



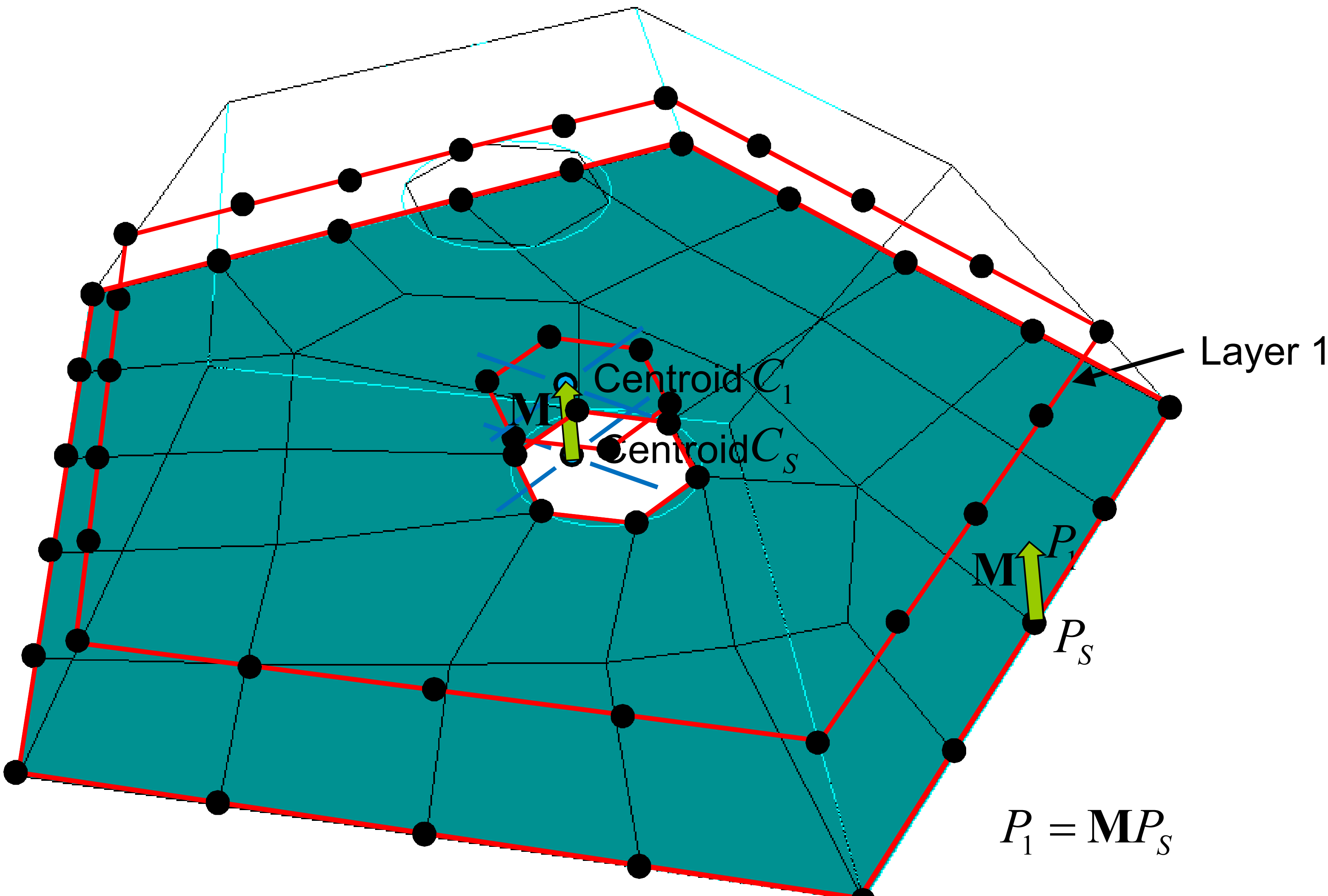
Weighted Residual Method



Weighted Residual Method

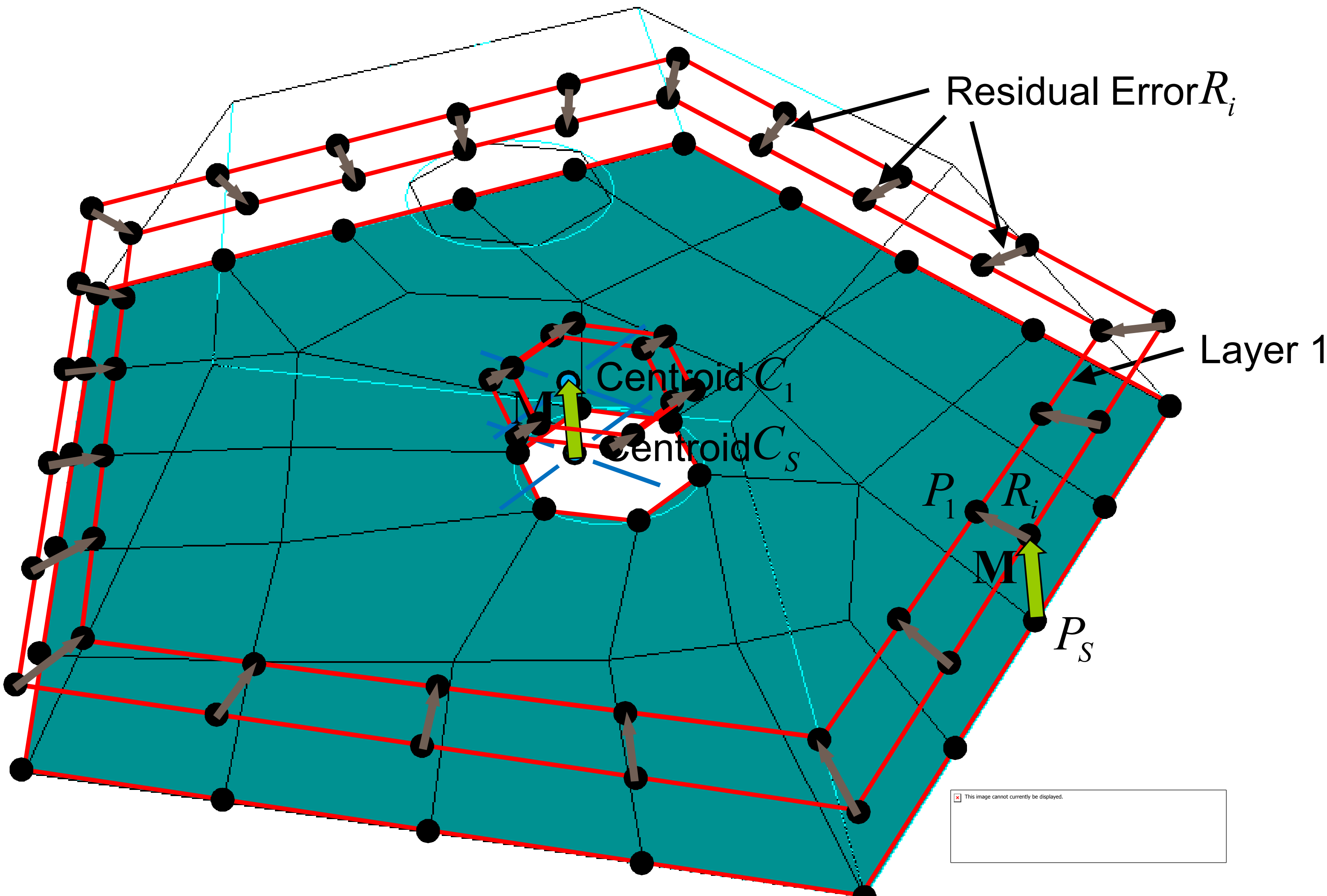


Weighted Residual Method

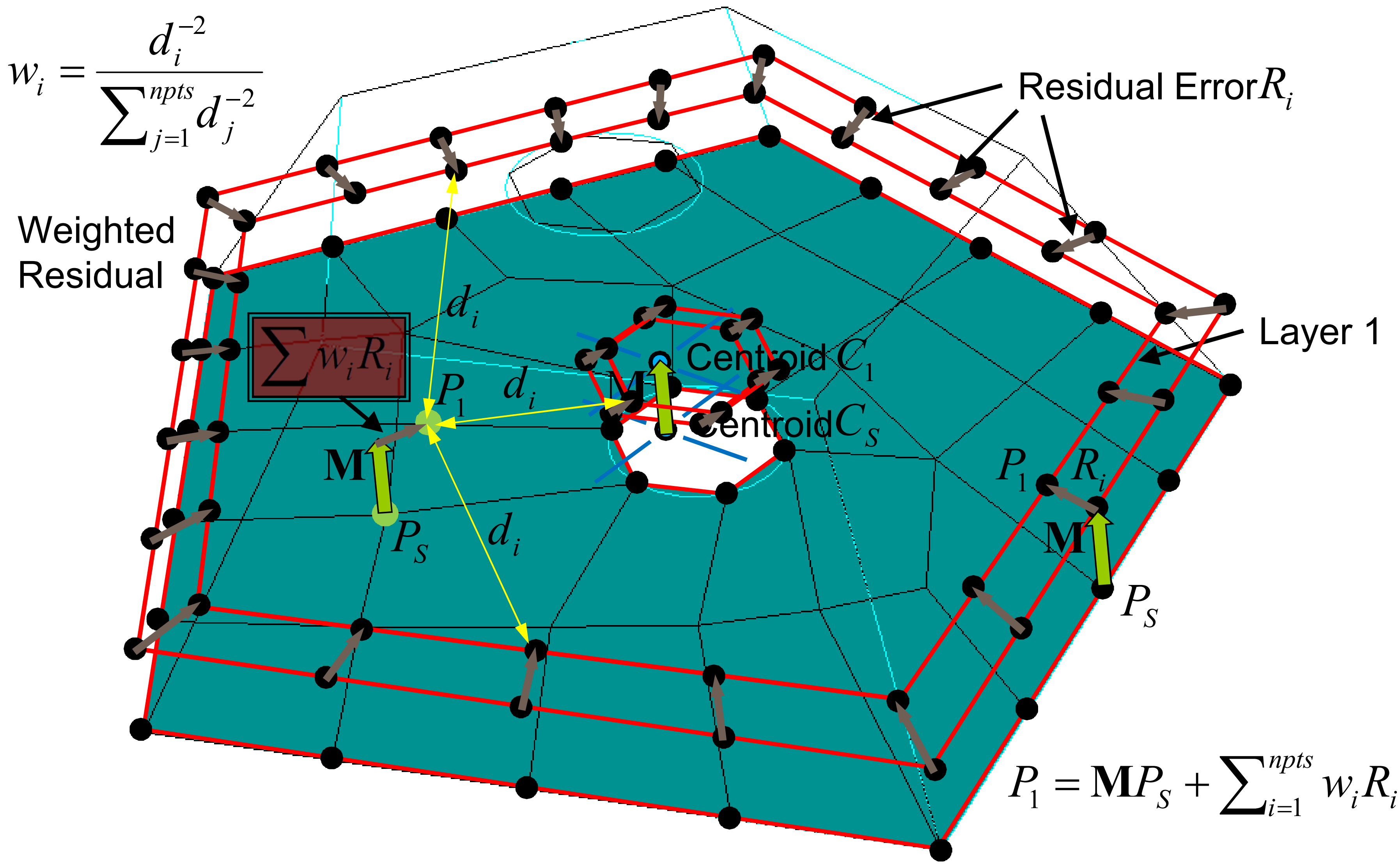


$$P_1 = \mathbf{M}P_S$$

Weighted Residual Method

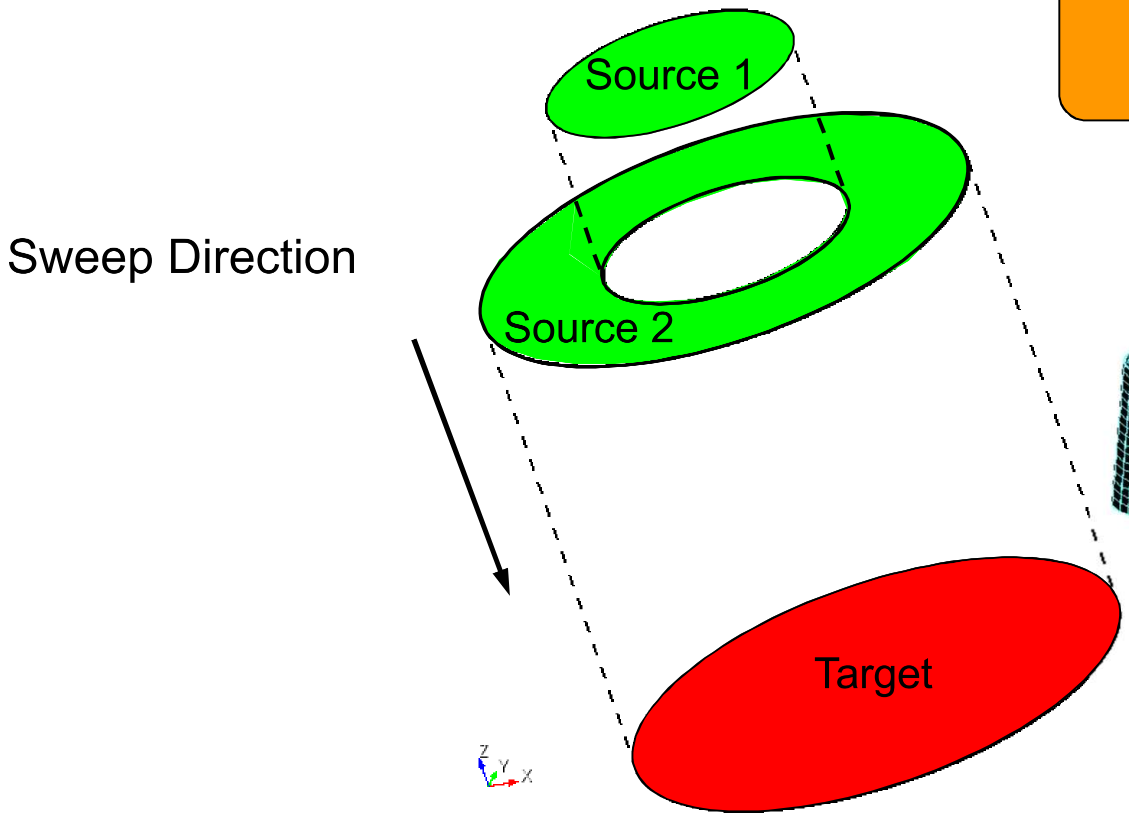


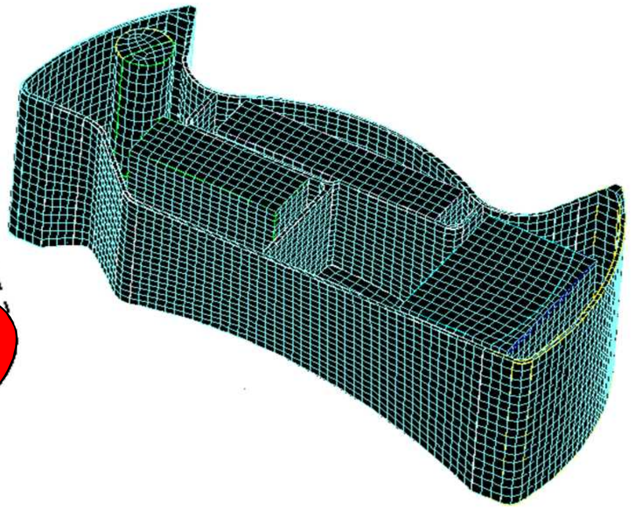
Weighted Residual Method



One-to-many Sweeping

n-to-1 sweepable



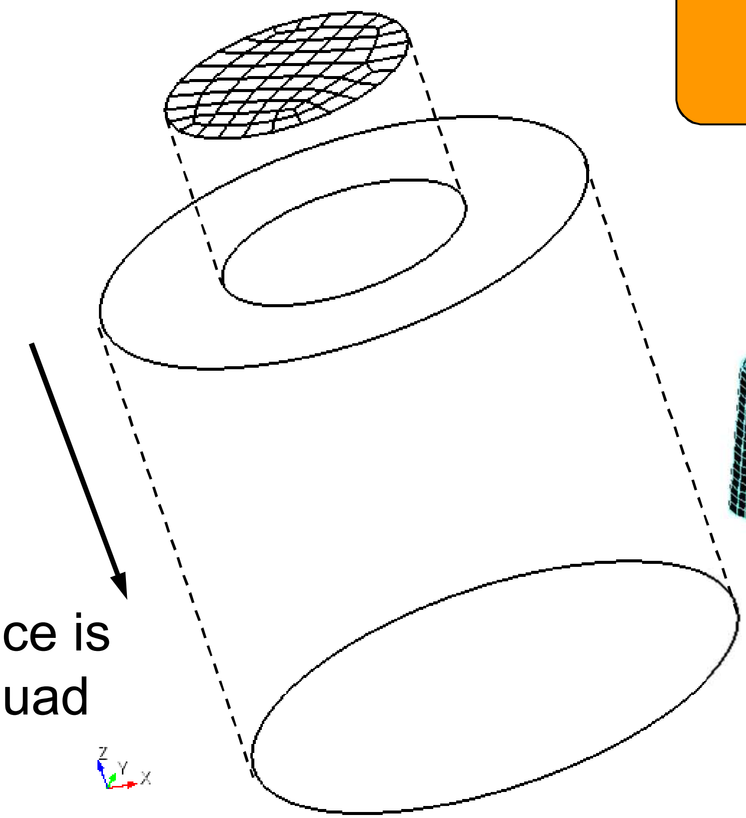


Matt Staten

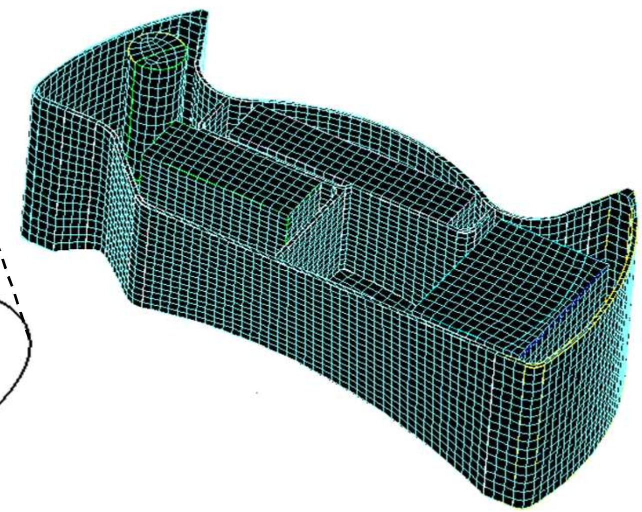
One-to-many Sweeping

n-to-1 sweepable

Sweep Direction



First source surface is meshed with all quad mesh

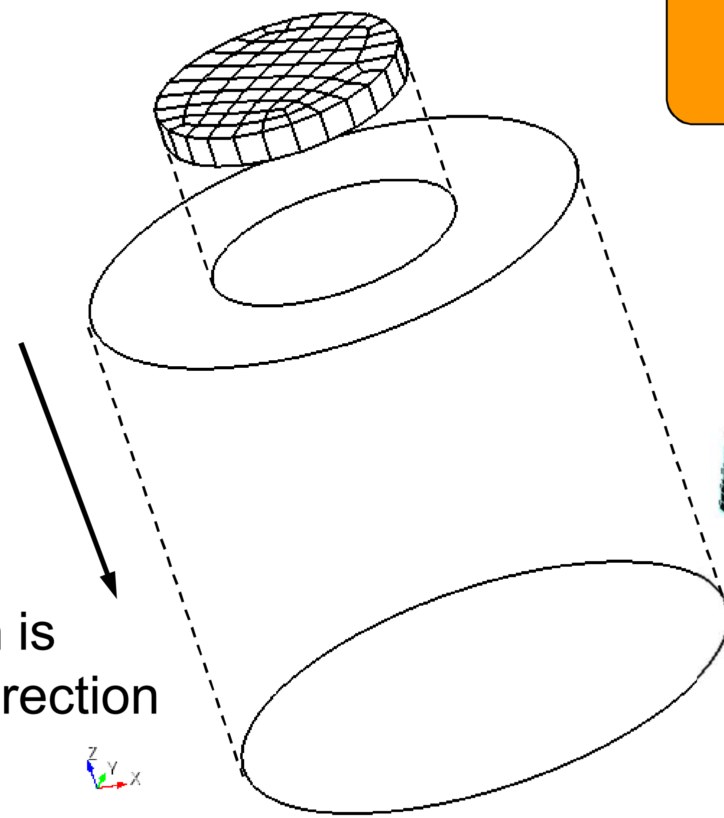


Matt Staten

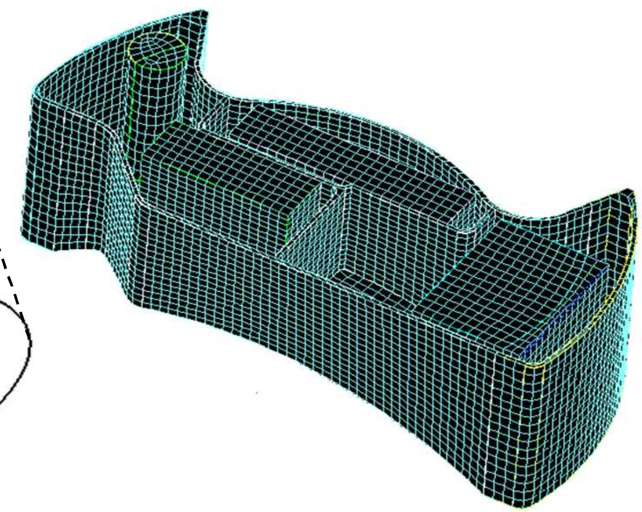
One-to-many Sweeping

n-to-1 sweepable

Sweep Direction



First source mesh is swept in sweep direction until it meets with another source

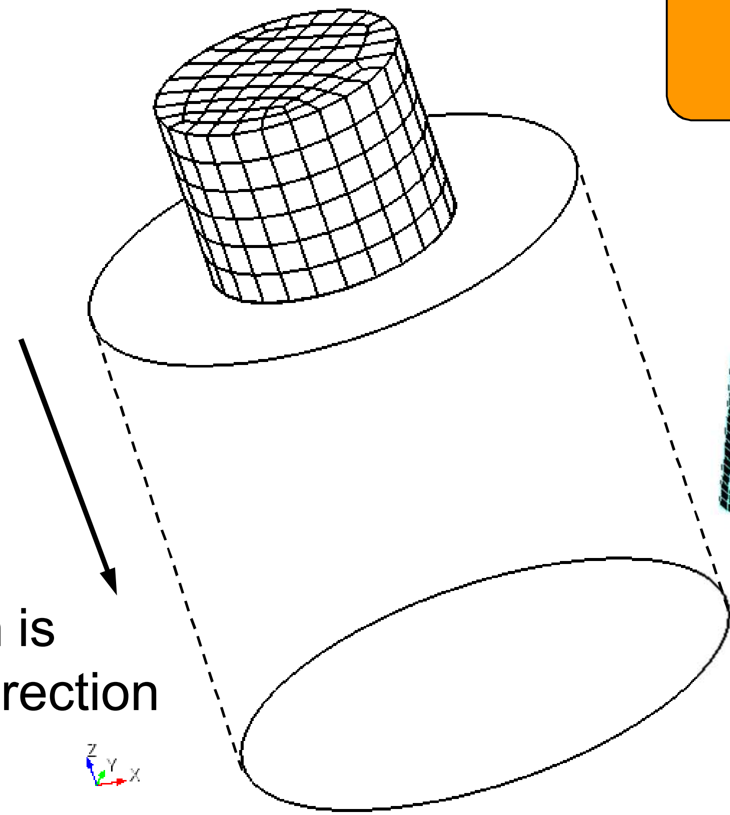


Matt Staten

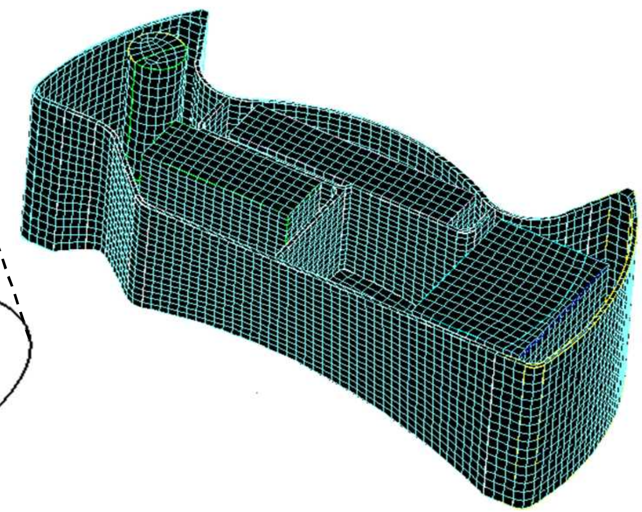
One-to-many Sweeping

n-to-1 sweepable

Sweep Direction



First source mesh is swept in sweep direction until it meets with another source

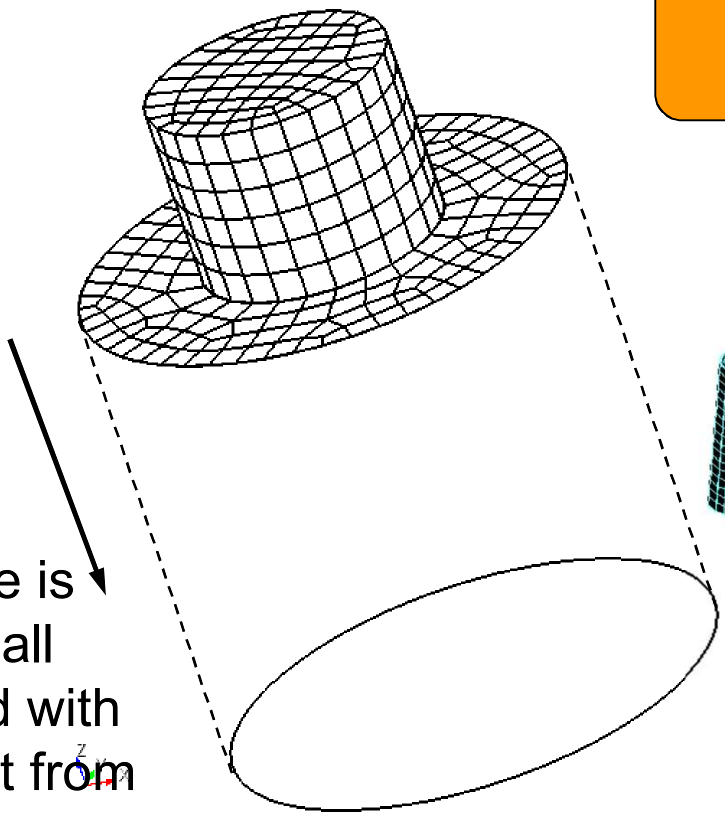


Matt Staten

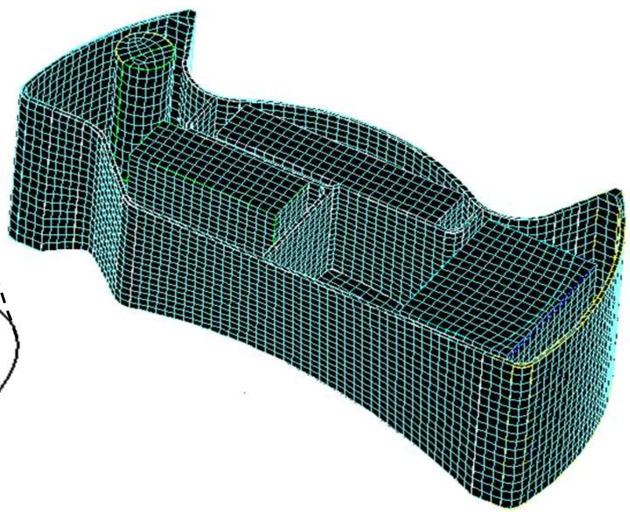
One-to-many Sweeping

n-to-1 sweepable

Sweep Direction



The second source is then meshed with all quads and merged with quads being swept from the first source.

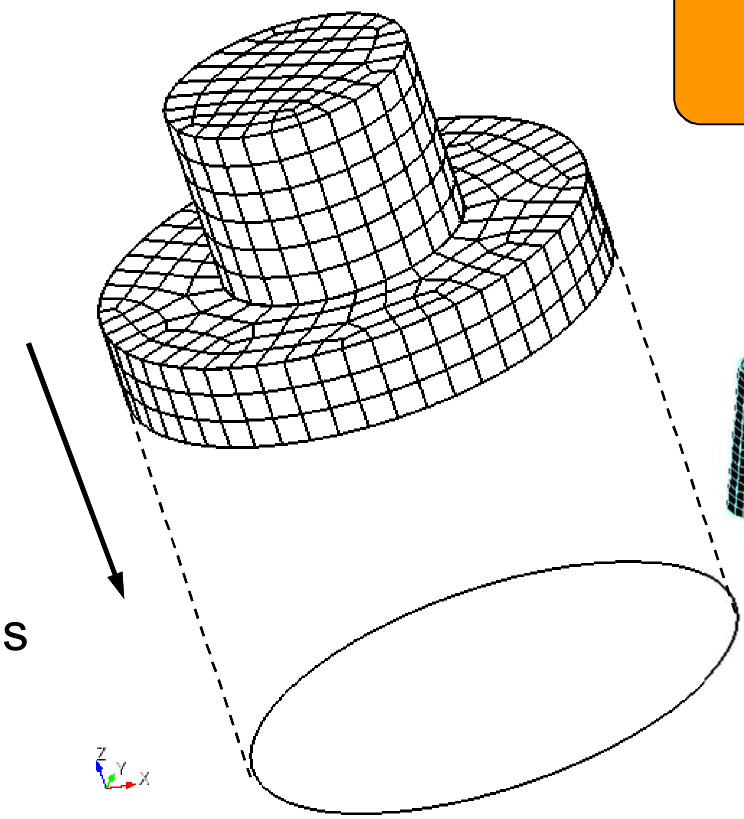


Matt Staten

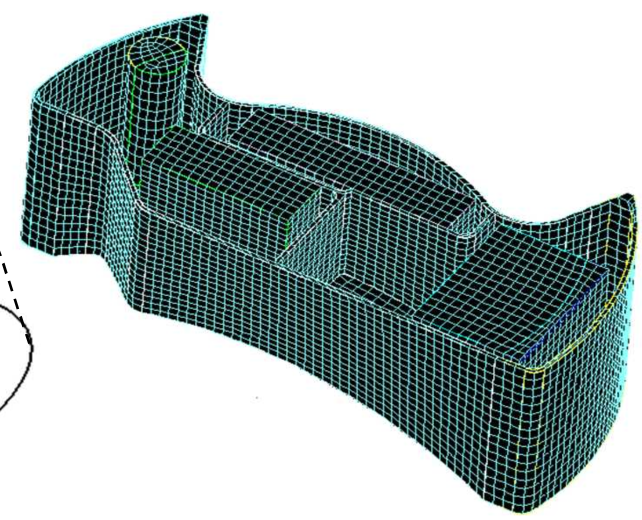
One-to-many Sweeping

n-to-1 sweepable

Sweep Direction



Combined mesh is swept towards a single target.



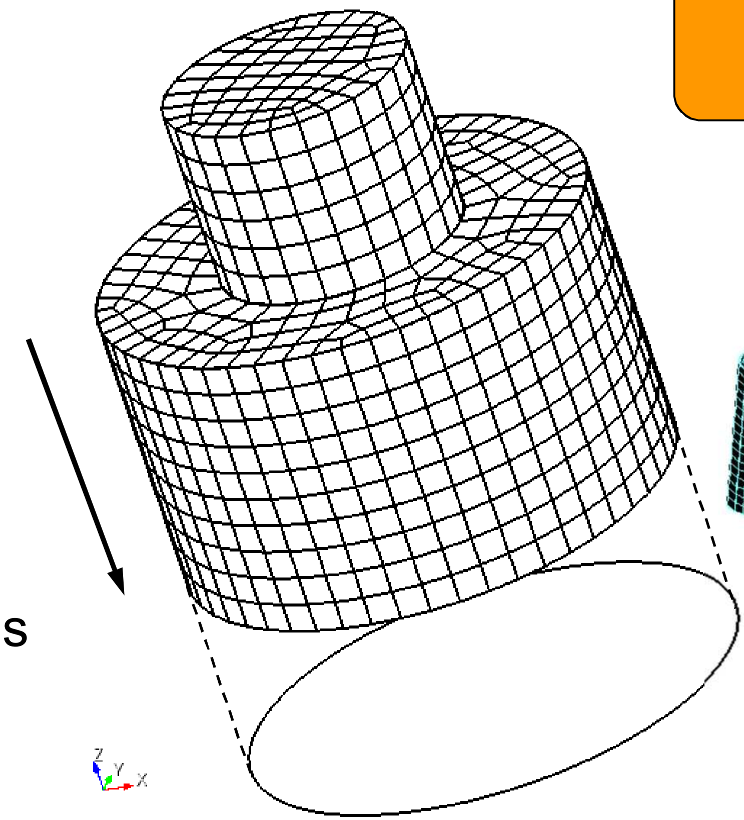
Matt Staten

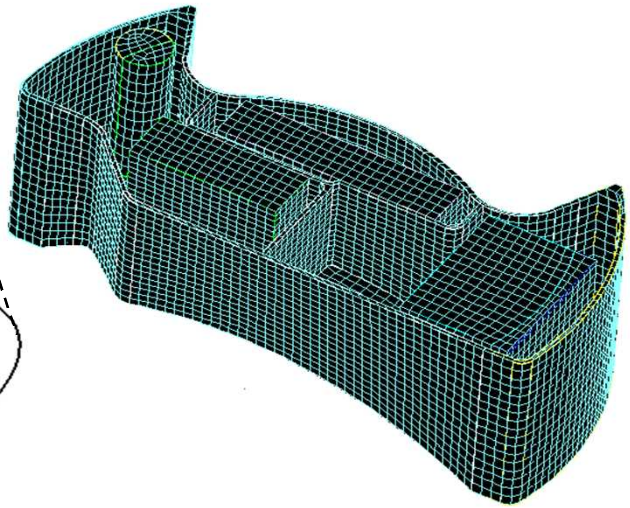
One-to-many Sweeping

n-to-1 sweepable

Sweep Direction

Combined mesh is swept towards a single target.



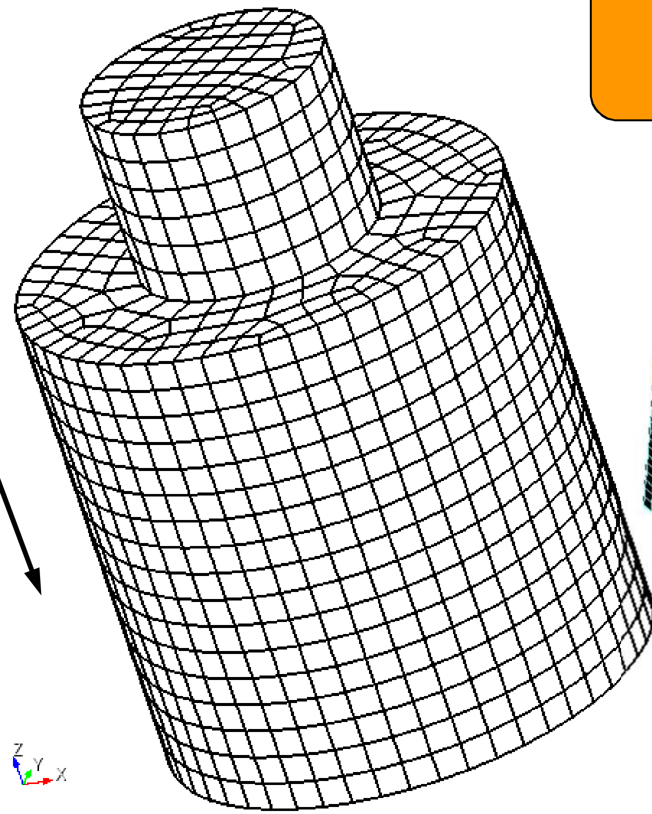


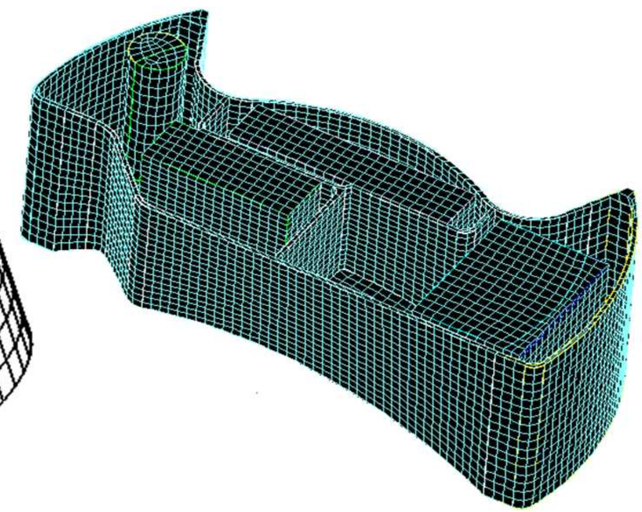
Matt Staten

One-to-many Sweeping

n-to-1 sweepable

Sweep Direction



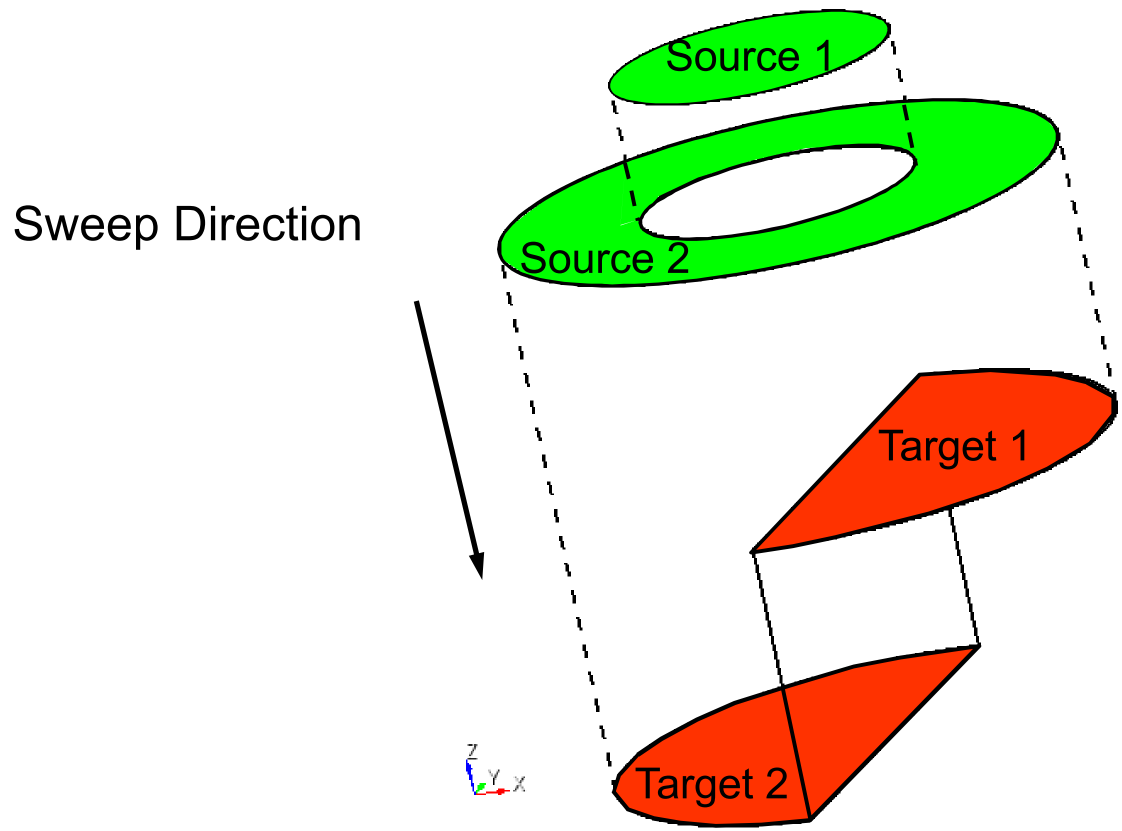


Combined mesh is swept towards a single target.

Matt Staten

Many-to-many Sweeping

n-to-m sweepable
Multi-Sweep

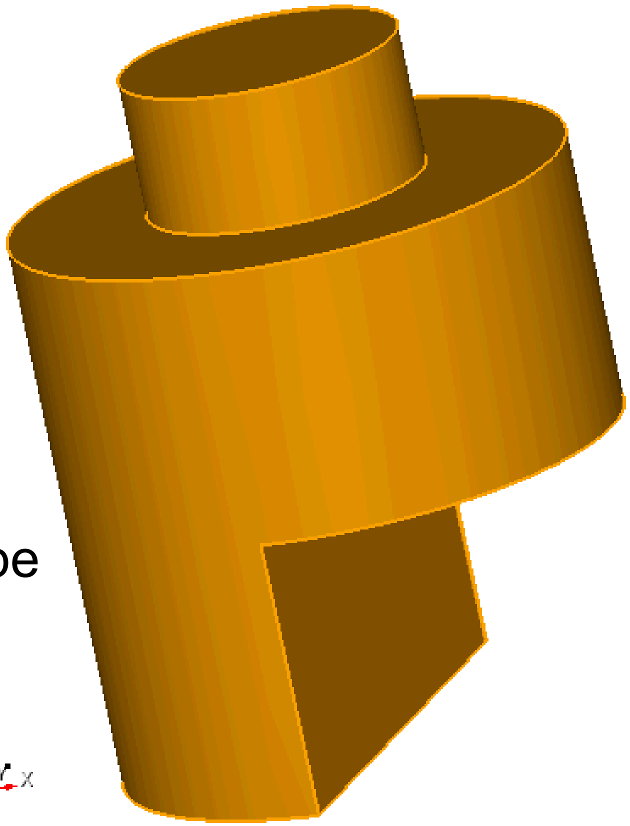


Matt Staten

Many-to-many Sweeping

n-to-m sweepable
Multi-Sweep

Sweep Direction



The footprints of the multiple targets must be imprinted onto the sources. This is done with virtual partitioning.

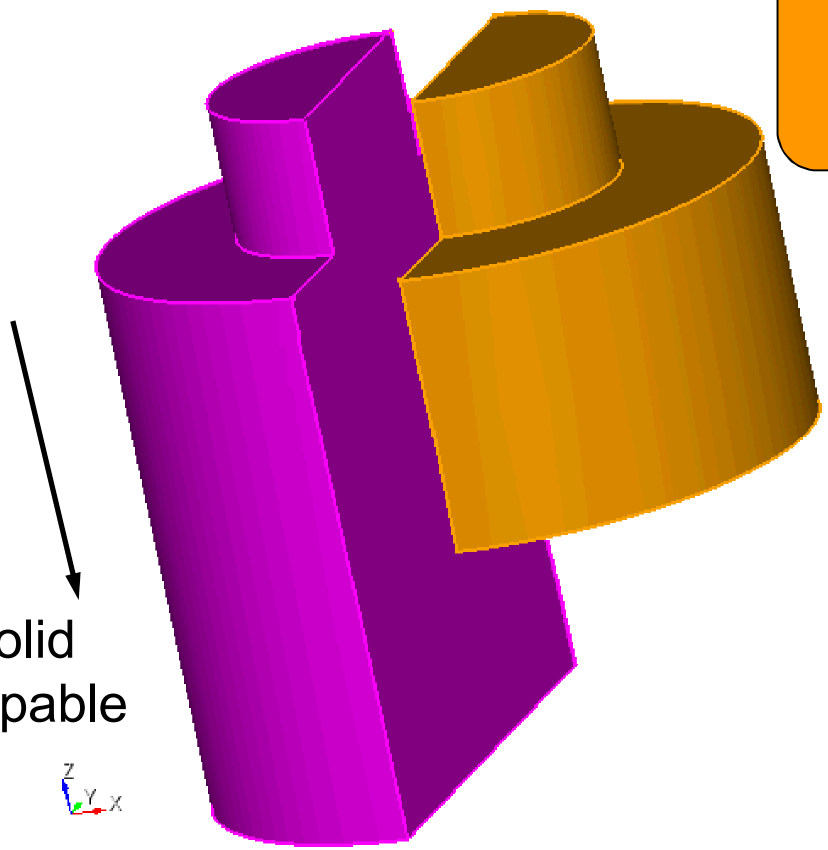
Matt Staten

Many-to-many Sweeping

Sandia National Laboratories

n-to-m sweepable
Multi-Sweep

Sweep Direction



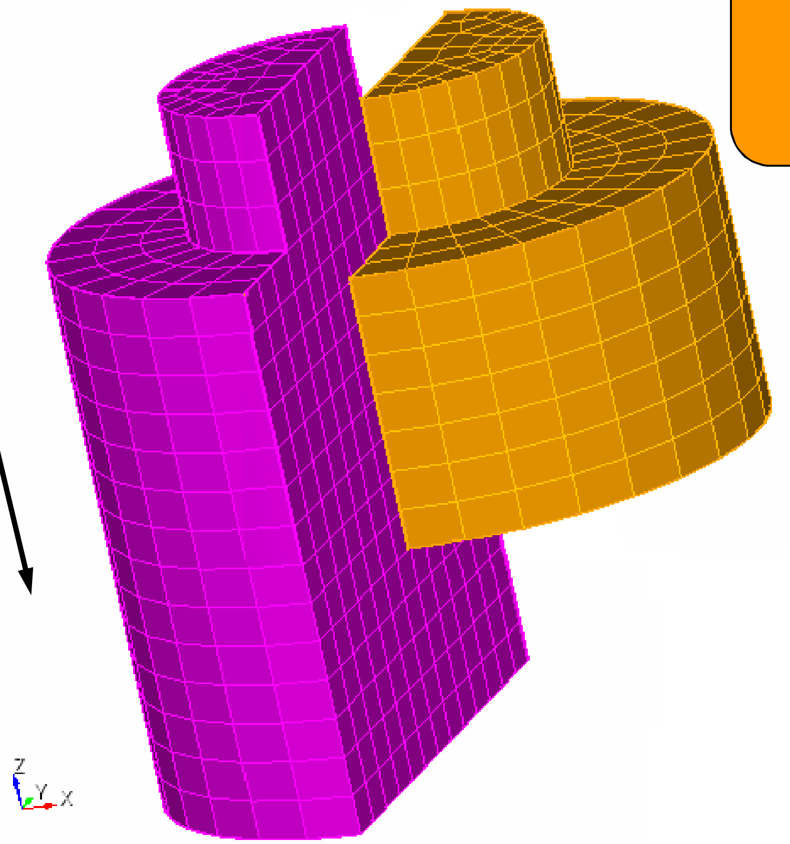
Virtual partitioning decomposes the solid into n-to-one sweepable sub-volumes.

Matt Staten

Many-to-many Sweeping

n-to-m sweepable
Multi-Sweep

Sweep Direction



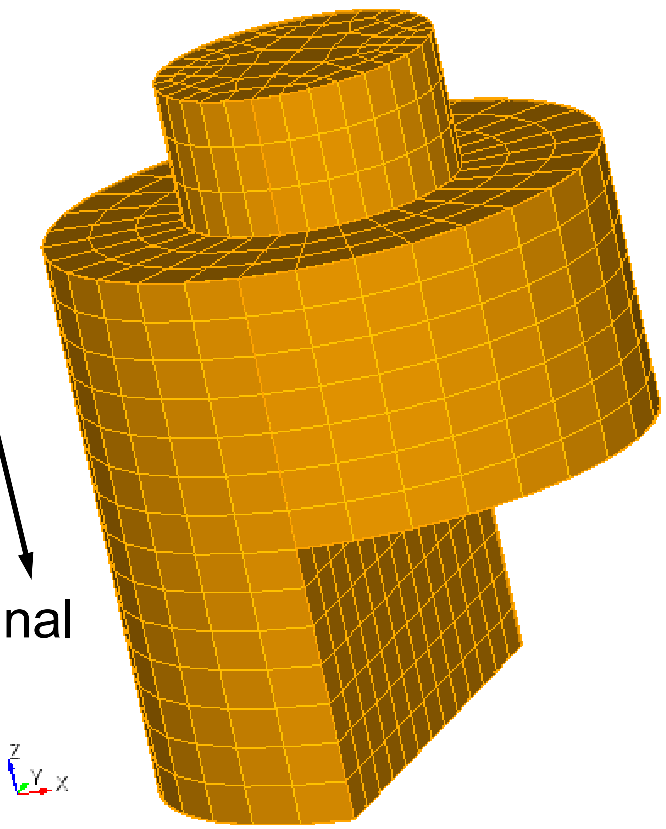
Each partition is meshed individually.

Matt Staten

Many-to-many Sweeping

n-to-m sweepable
Multi-Sweep

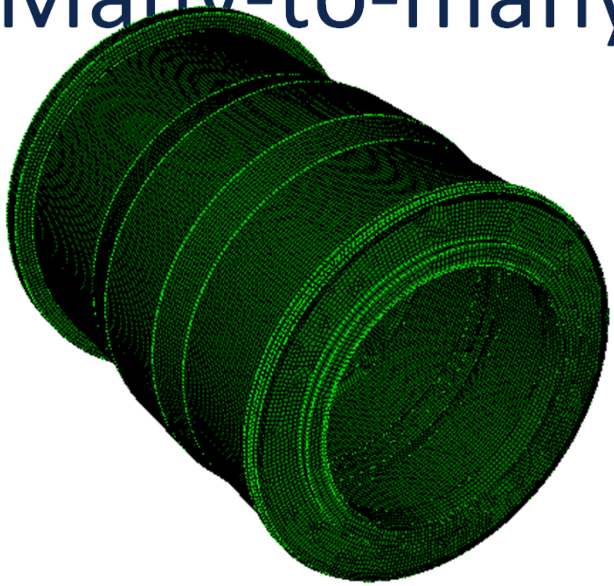
Sweep Direction

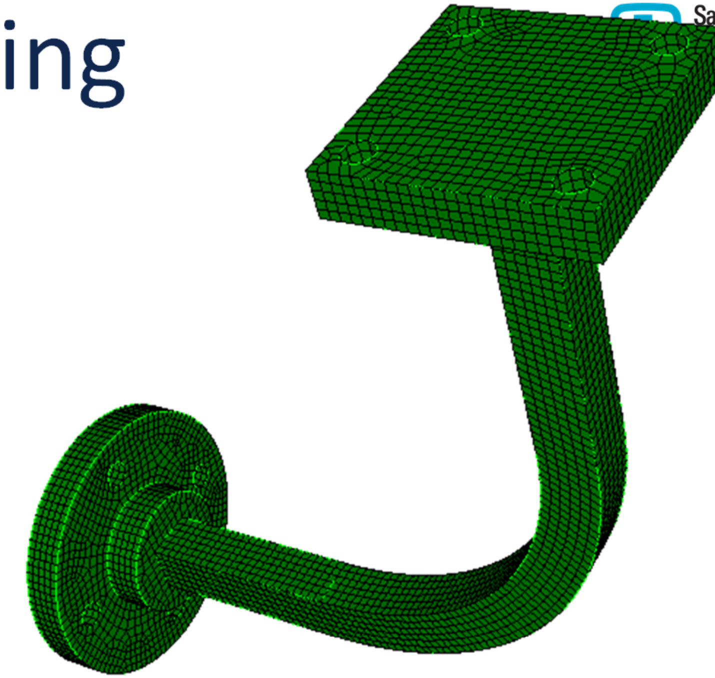


Virtual partitions are deleted leaving the final mesh.

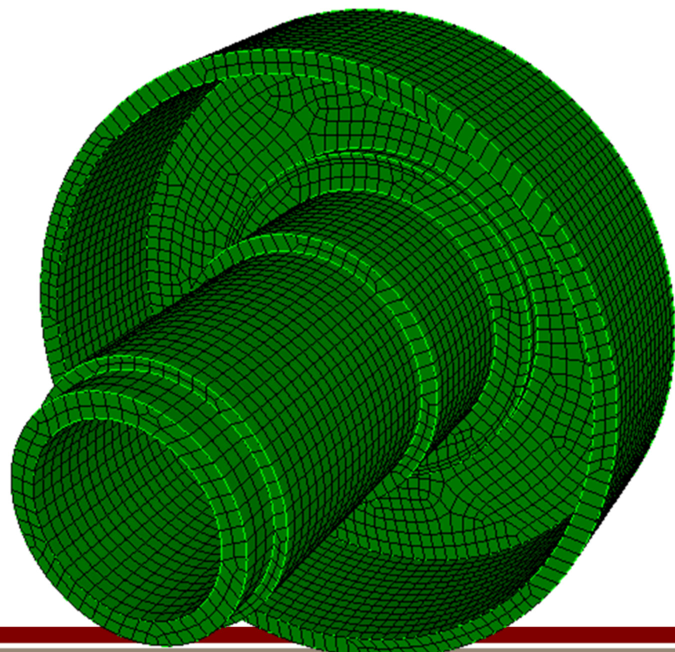
Matt Staten

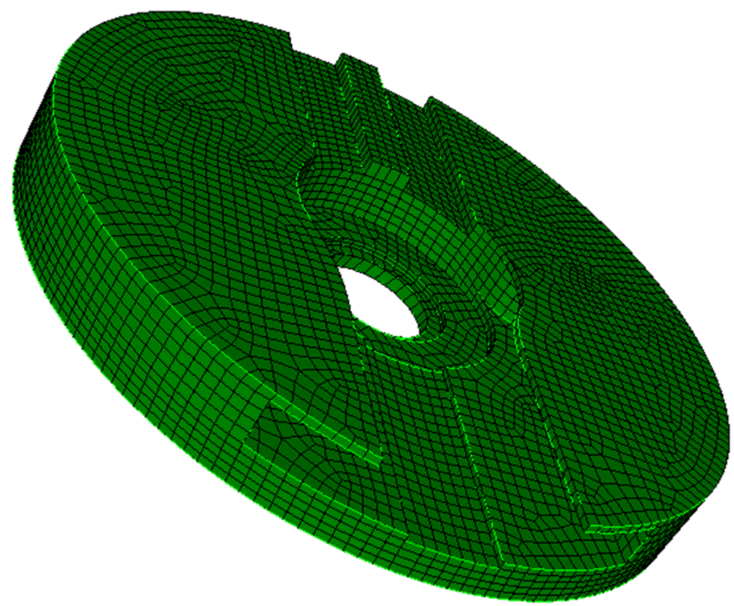
Many-to-many Sweeping





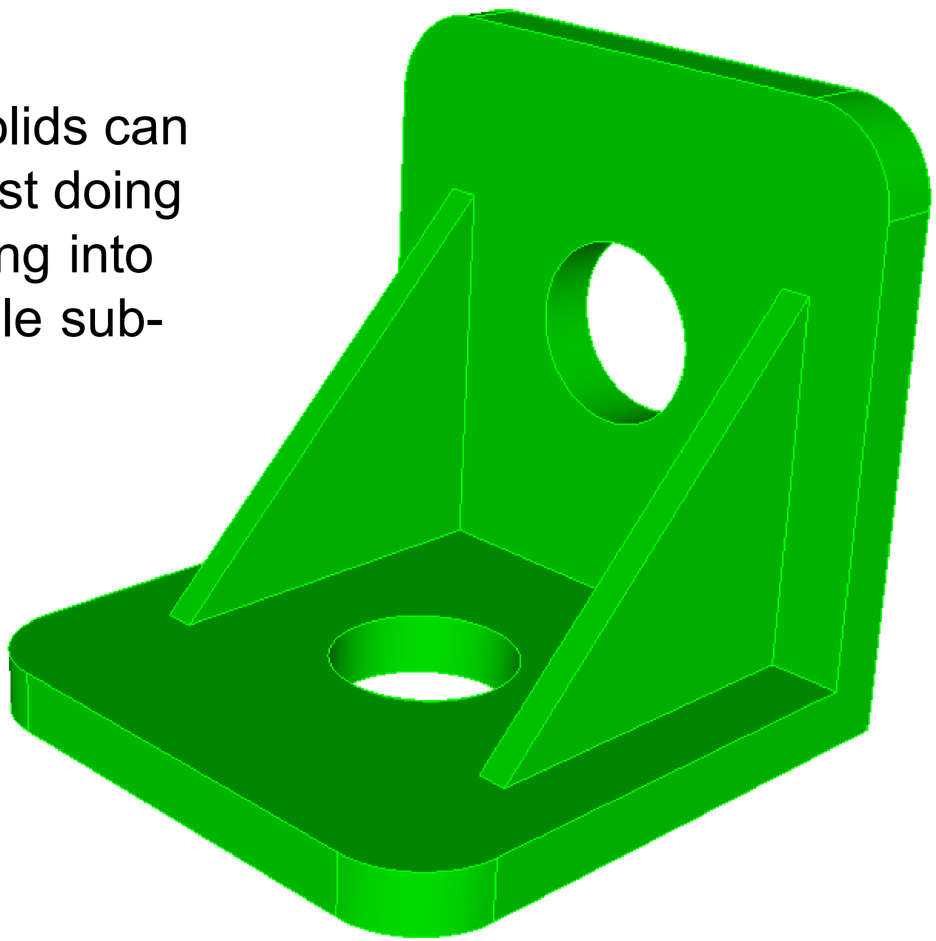
Examples of Many-to-many sweeping with CUBIT





Partition & Sweeping

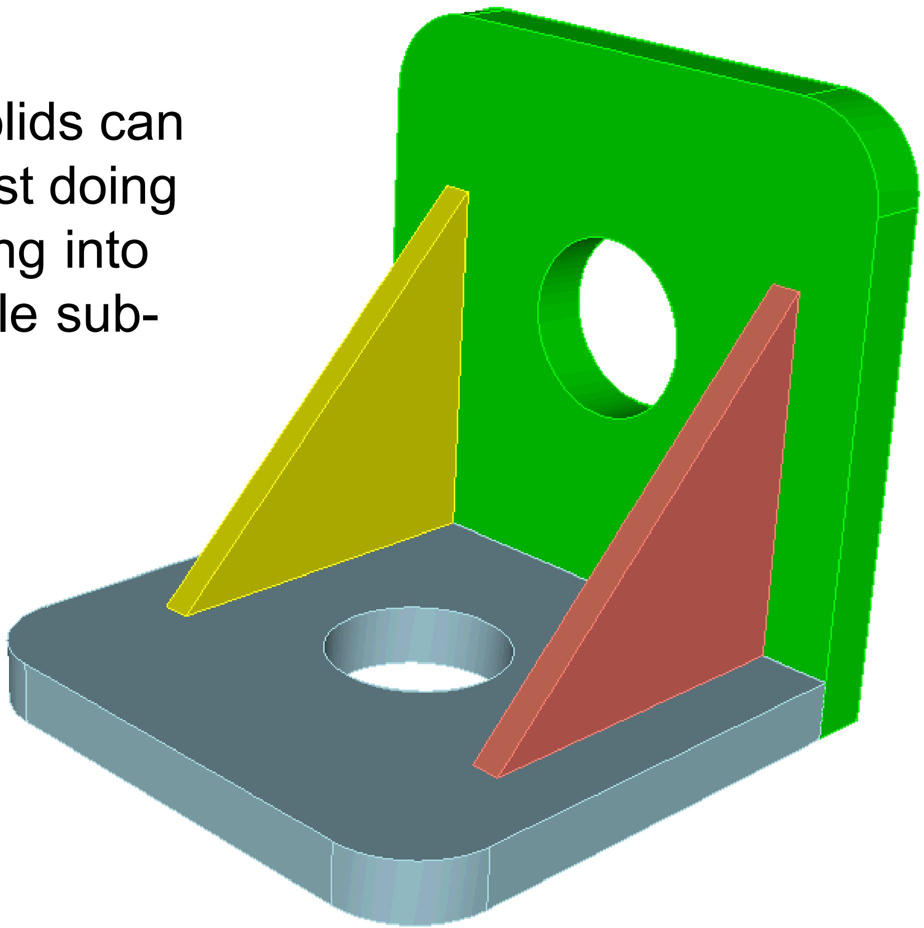
More complex solids can be meshed by first doing manual partitioning into several sweepable sub-solids.



Matt Staten

Partition & Sweeping

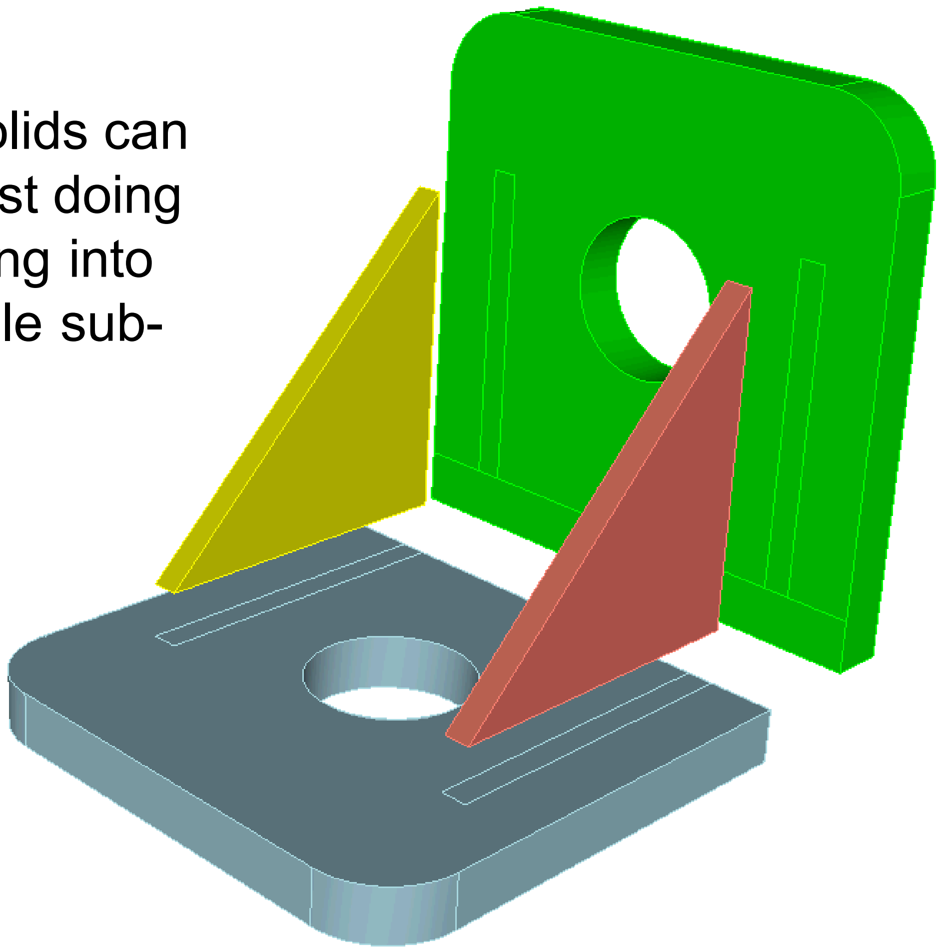
More complex solids can be meshed by first doing manual partitioning into several sweepable sub-solids.



Matt Staten

Partition & Sweeping

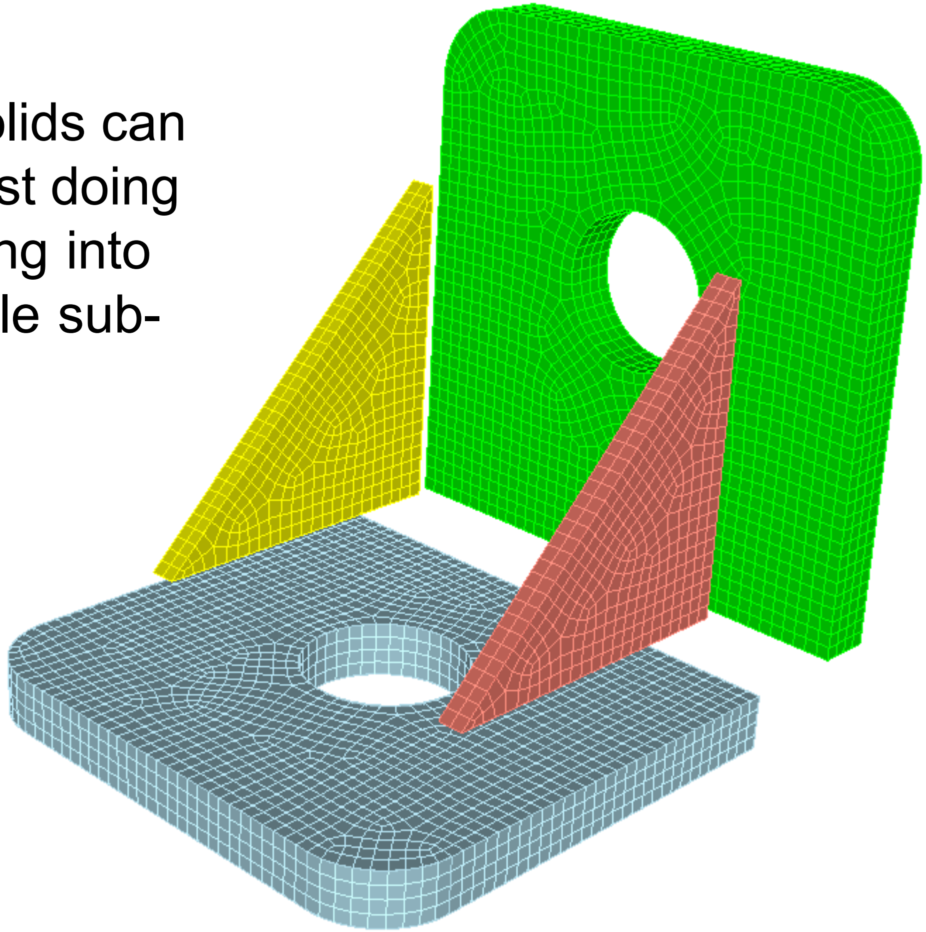
More complex solids can be meshed by first doing manual partitioning into several sweepable sub-solids.



Matt Staten

Partition & Sweeping

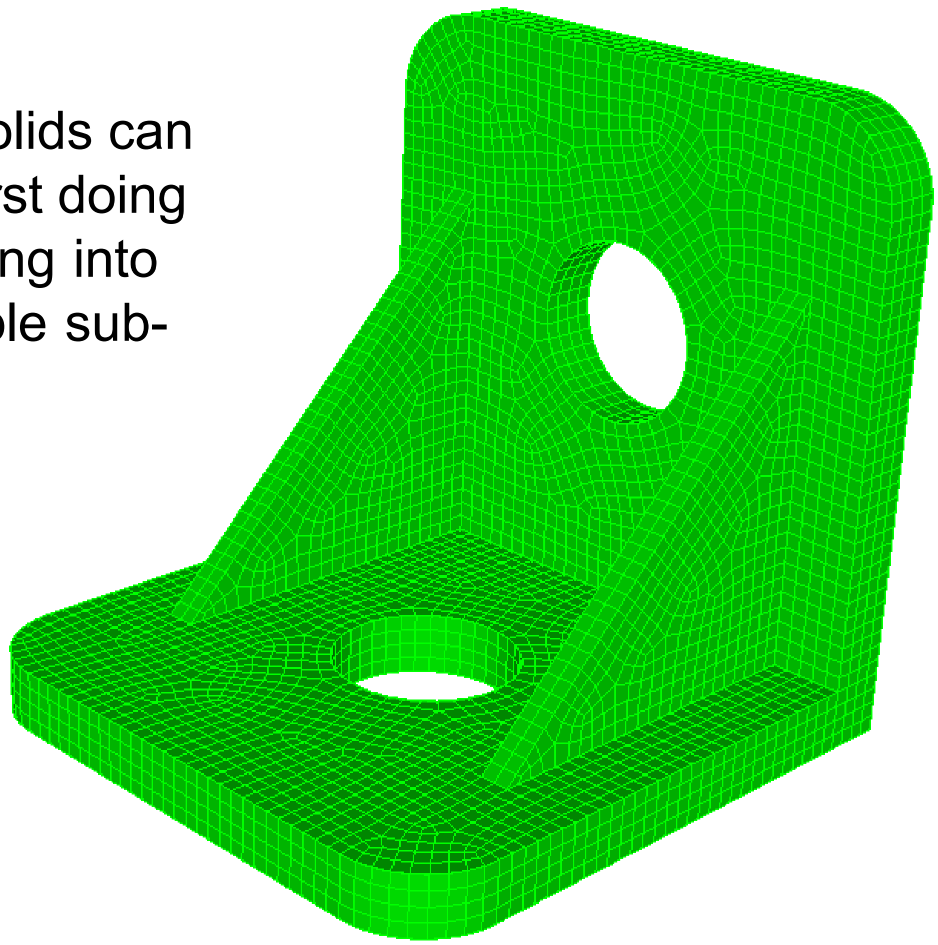
More complex solids can be meshed by first doing manual partitioning into several sweepable sub-solids.



Matt Staten

Partition & Sweeping

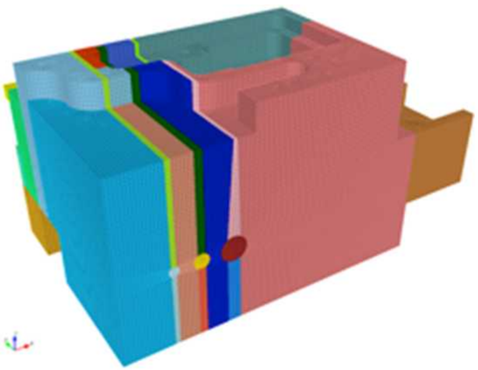
More complex solids can be meshed by first doing manual partitioning into several sweepable sub-solids.



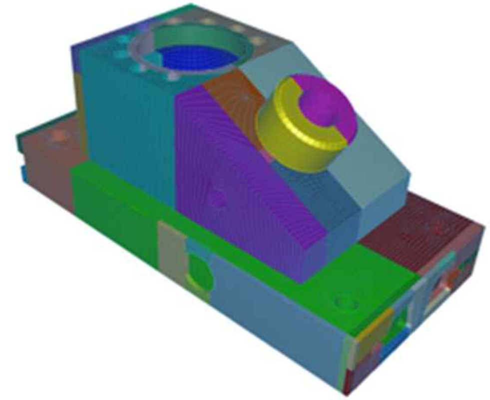
Matt Staten

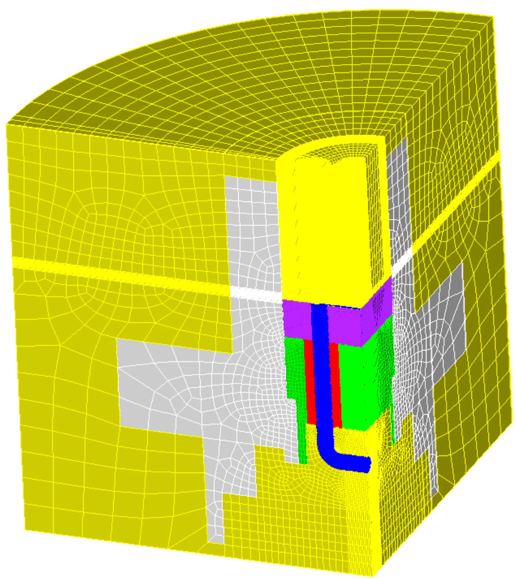
Sandia National Laboratories

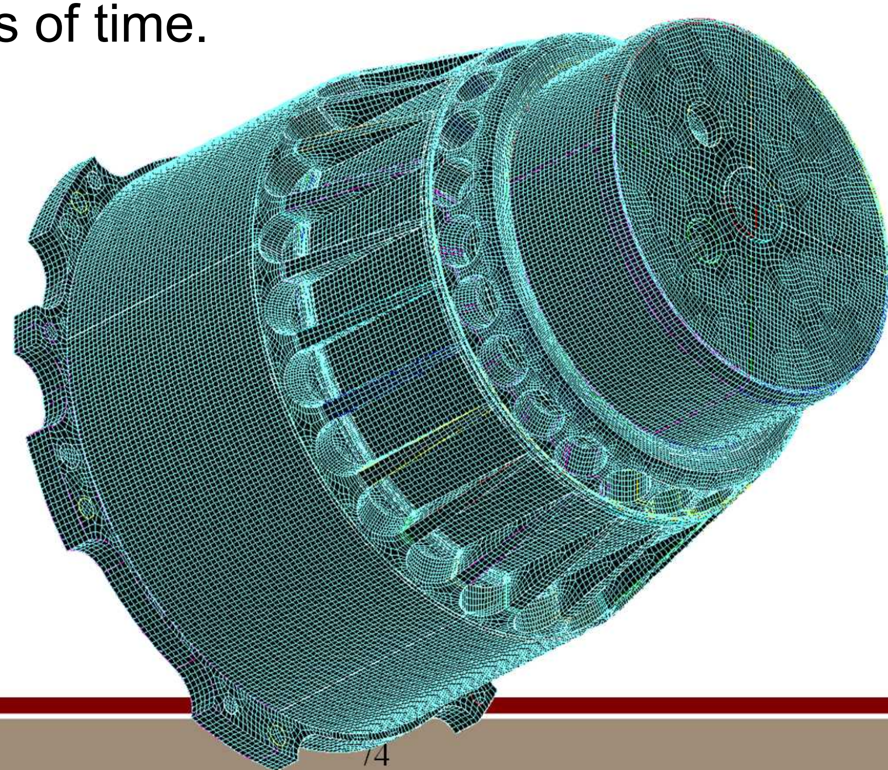
Partitioning & Sweeping Very Complex Solids

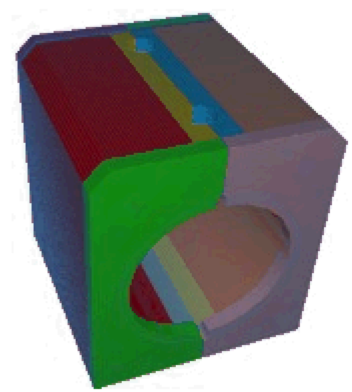


"Any" geometry, regardless of complexity, can be meshed by first decomposing it into sweepable sub-solids. Decomposition step of complex solids requires tedium, experience, and creativity and often lots of time.



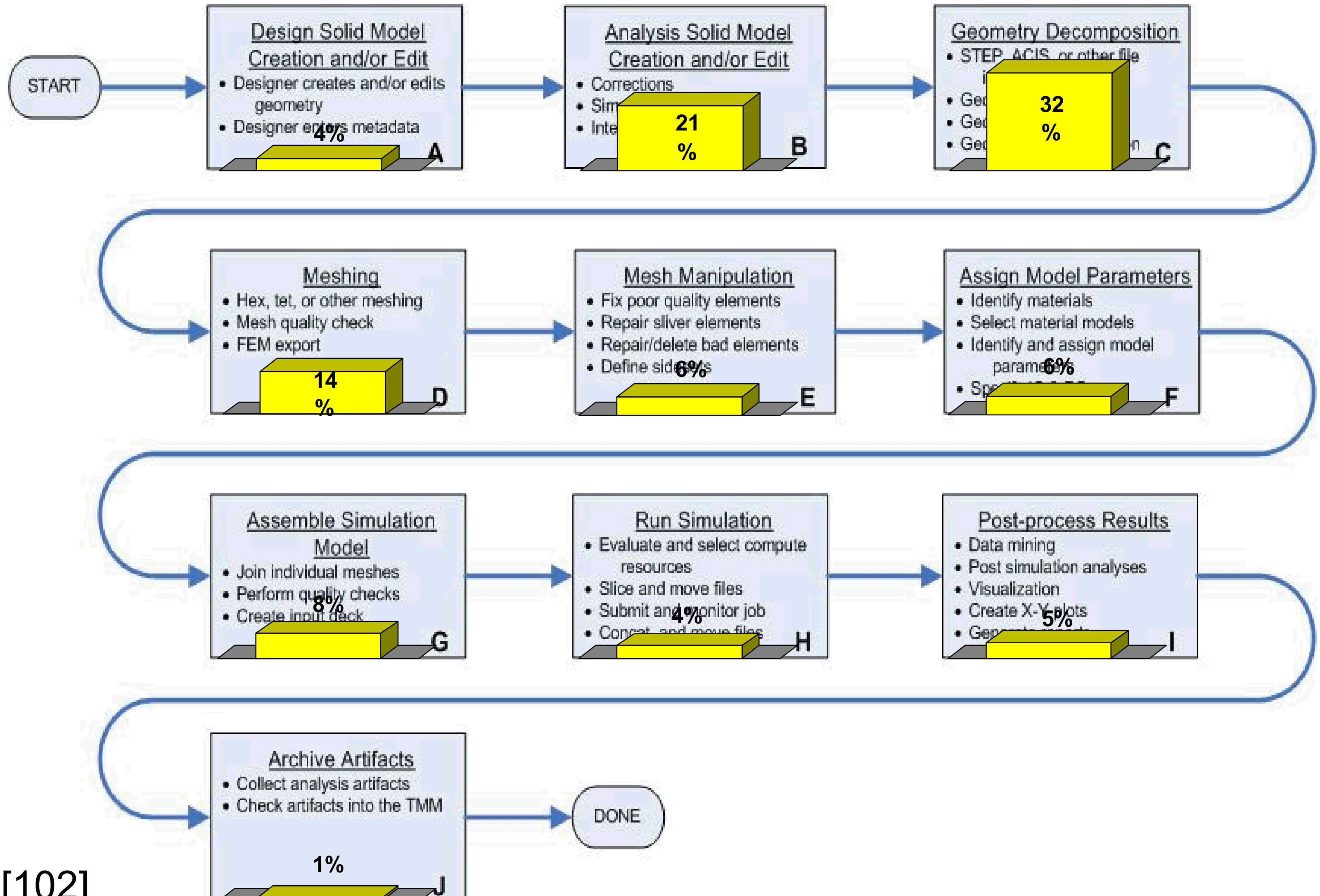






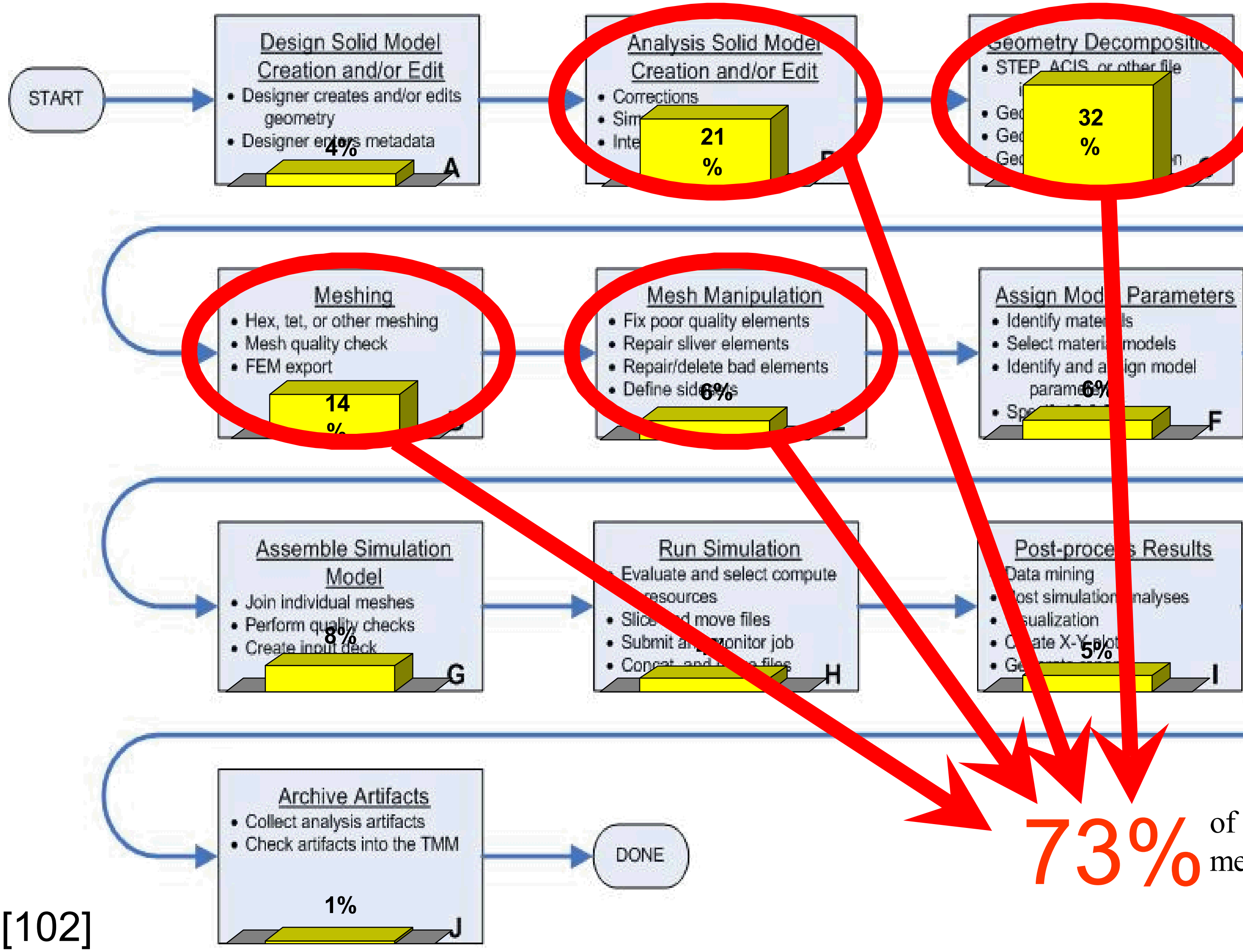
Matt Staten

Engineering Analysis with Geometry Conforming Hexahedral Elements



Owen [102]

Engineering Analysis with Geometry Conforming Hexahedral Elements



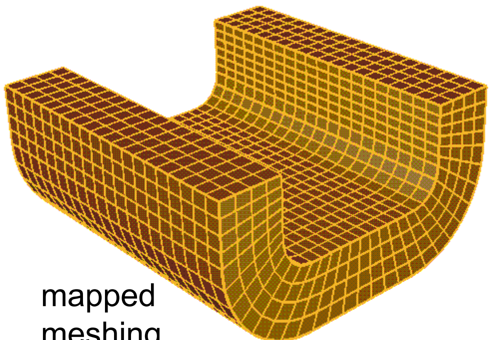
of user time spent in mesh preparation

Owen [102]

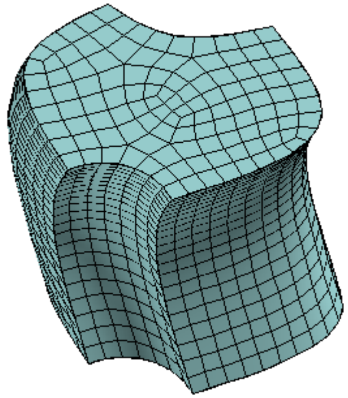
Structured vs. Unstructured

Sandia National Laboratories

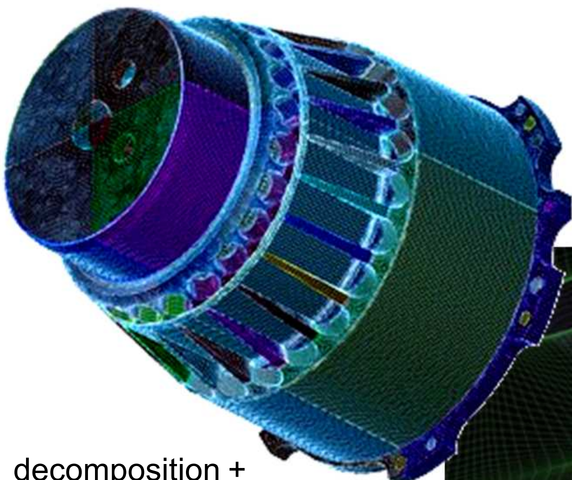
Structured



mapped meshing

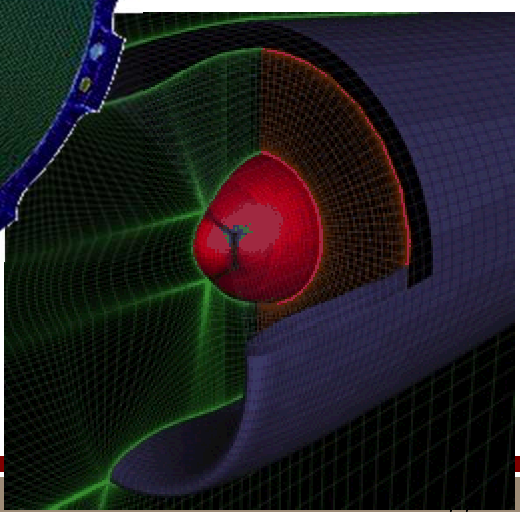


sweeping

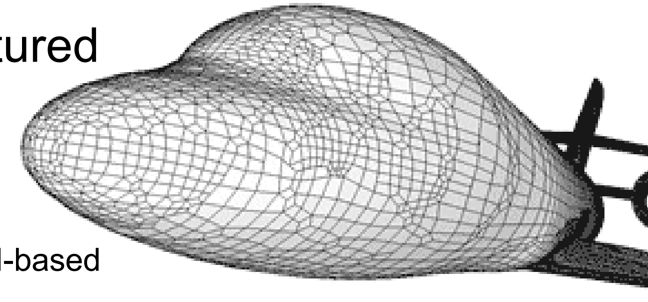


decomposition + sweeping

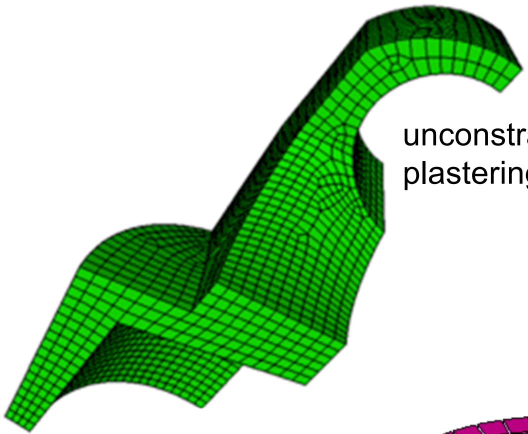
block structured



Unstructured

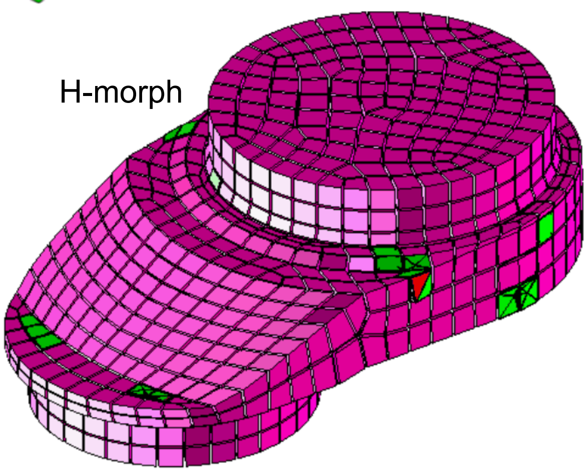


grid-based

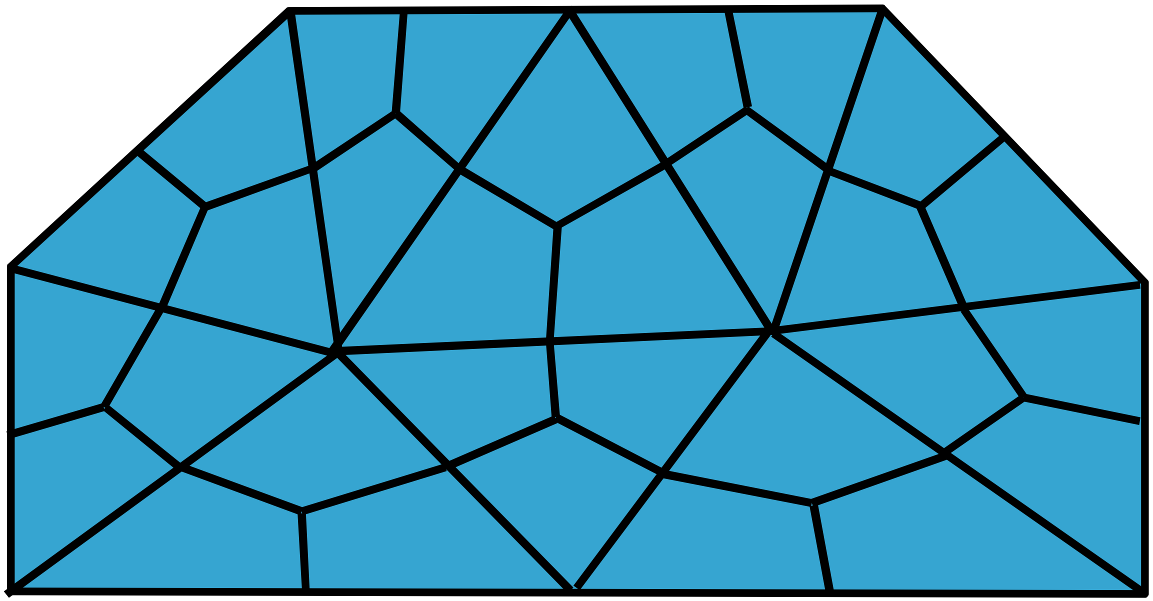


unconstrained plastering

H-morph

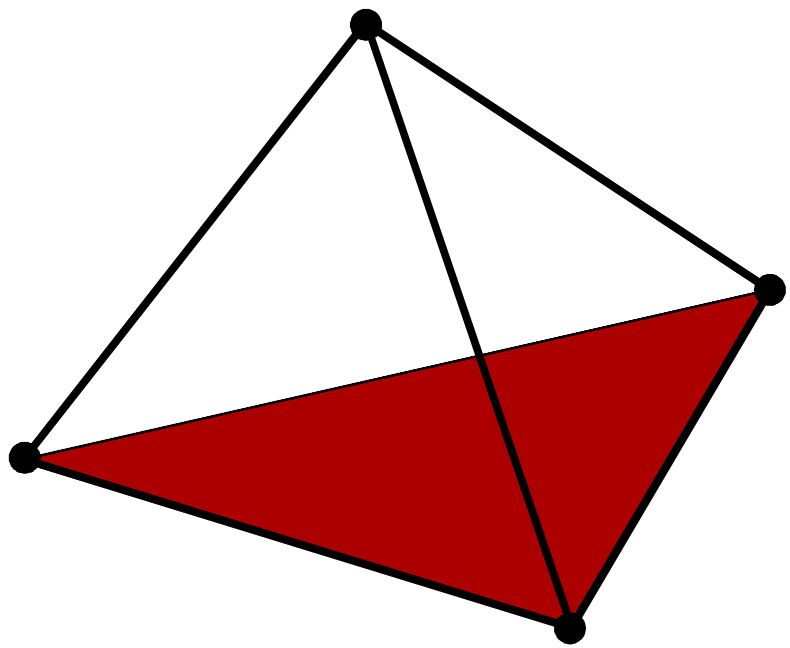


Triangle splitting



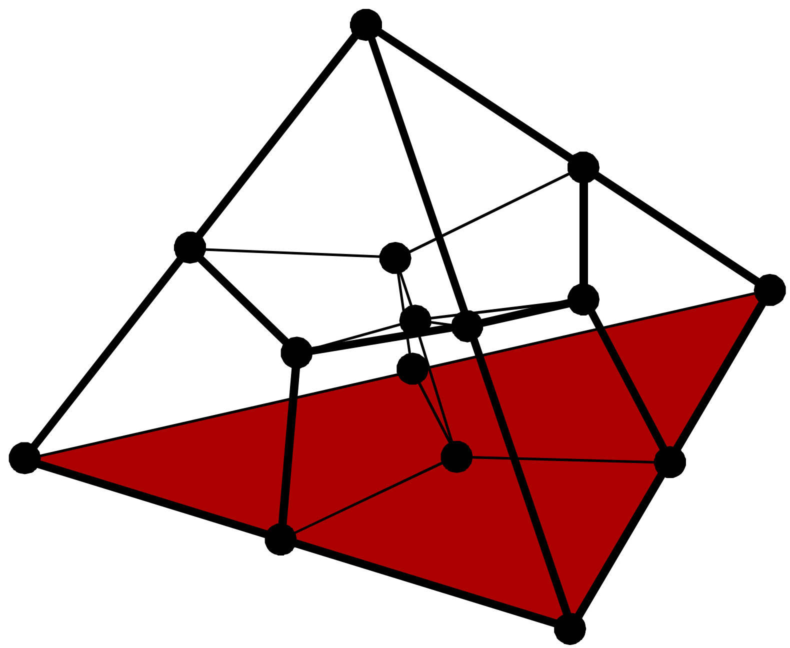
•Each triangle split into 3 quads
•Typically results in poor angles

Indirect Hex



- Each tetrahedra split into 4 hexahedra
- Typically results in poor angles

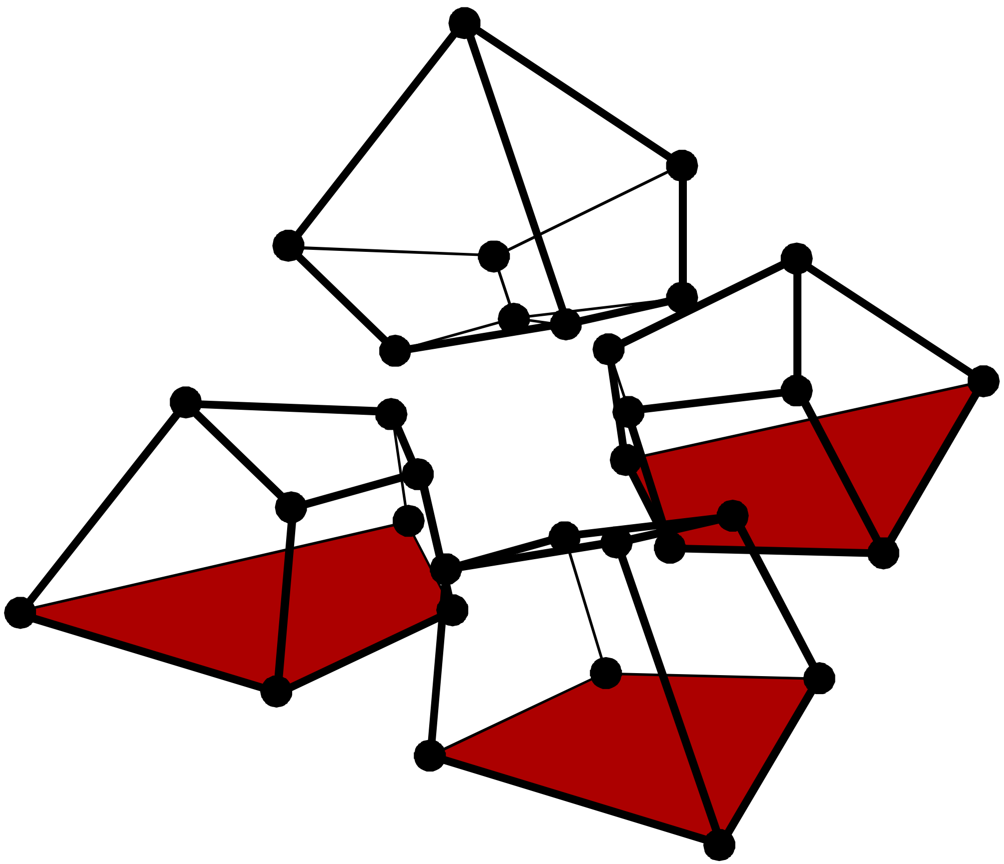
Tetrahedra splitting



- Each tetrahedra split into 4 hexahedra
- Typically results in poor angles

(Taniguchi, 96)

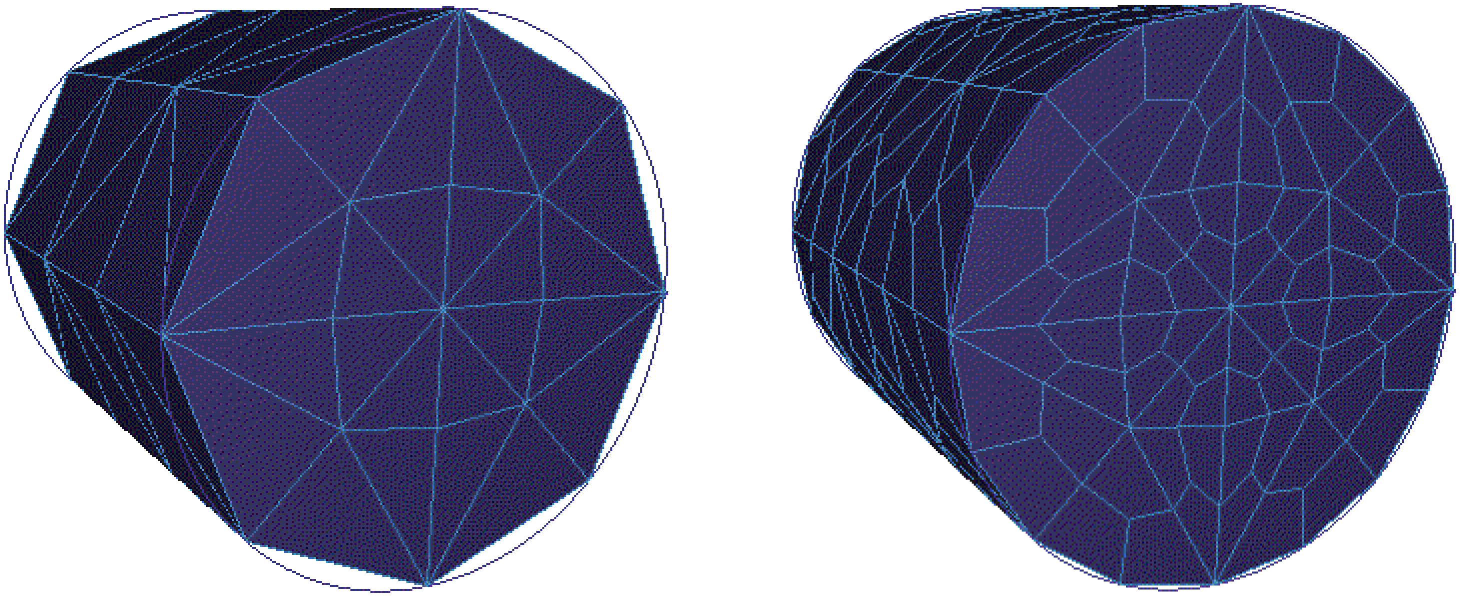
Tetrahedra splitting



- Each tetrahedra split into 4 hexahedra
- Typically results in poor angles

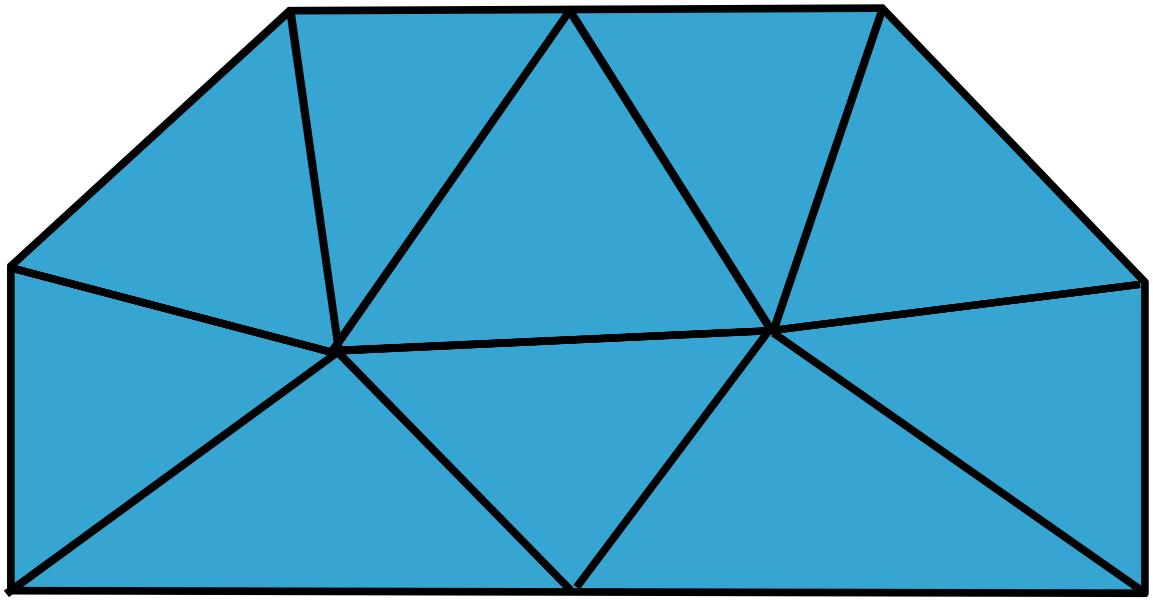
(Taniguchi, 96)

Tetrahedra splitting



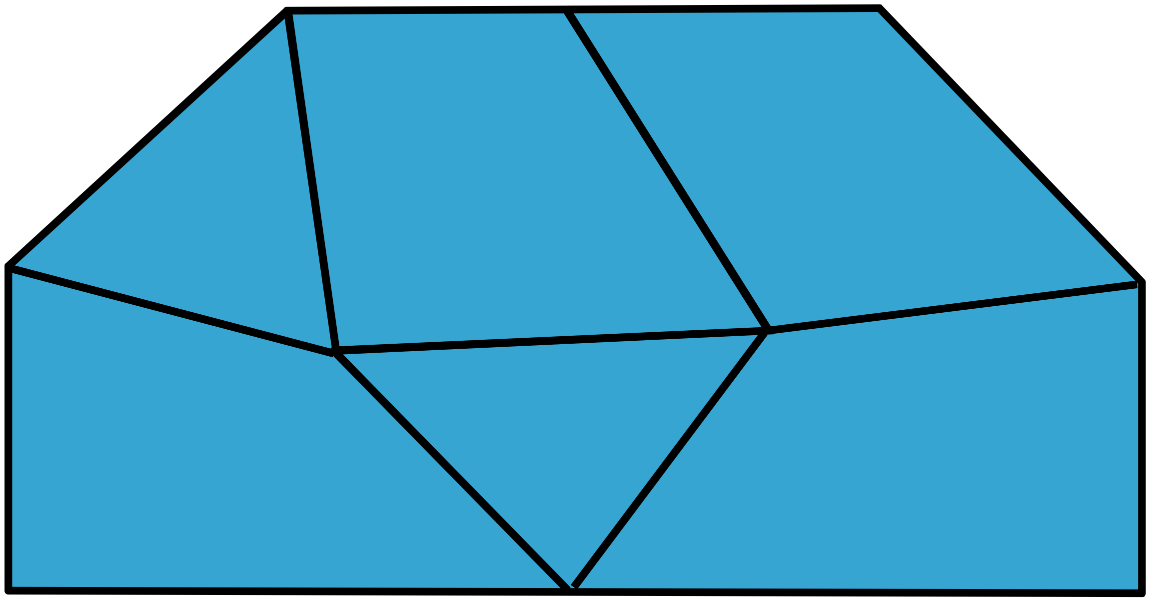
- Example of geometry meshed by tetrahedrasplitting
- Cubit's *T-Hex* algorithm
- Quality is rarely sufficient for FEA

Triangle Merging



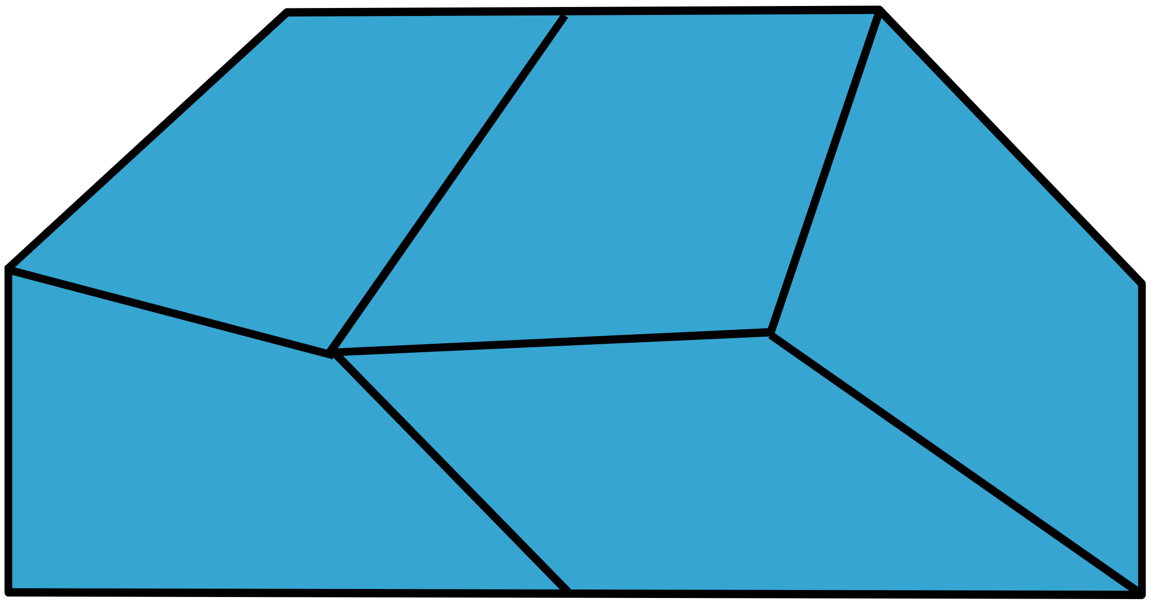
- Two adjacent triangles combined into a single quad
- Test for best local choice for combination
- Triangles can remain if attention is not paid to order of combination

Triangle Merging



- Two adjacent triangles combined into a single quad
- Test for best local choice for combination
- Triangles can remain if attention is not paid to order of combination

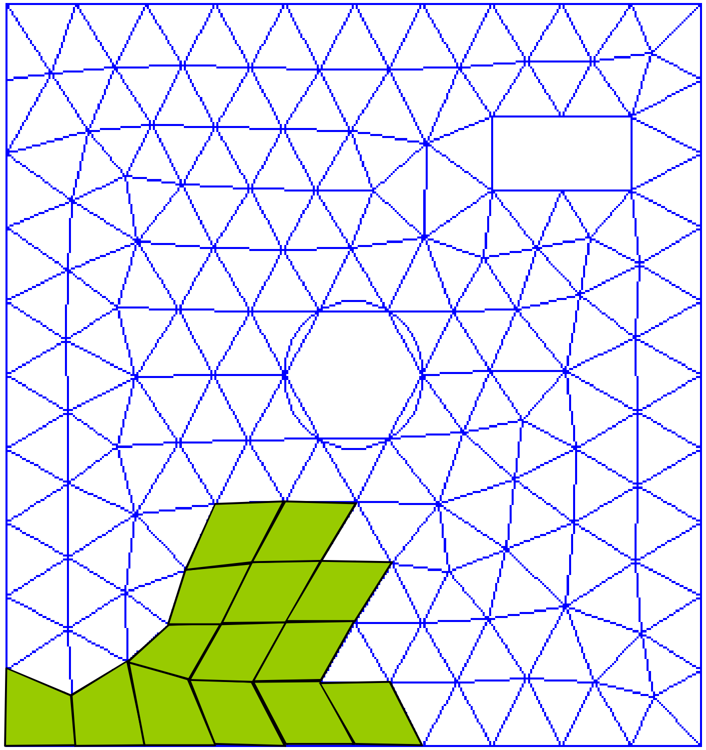
Triangle Merging



- Two adjacent triangles combined into a single quad
- Test for best local choice for combination
- Triangles can remain if attention is not paid to order of combination

Directed Triangle Merging

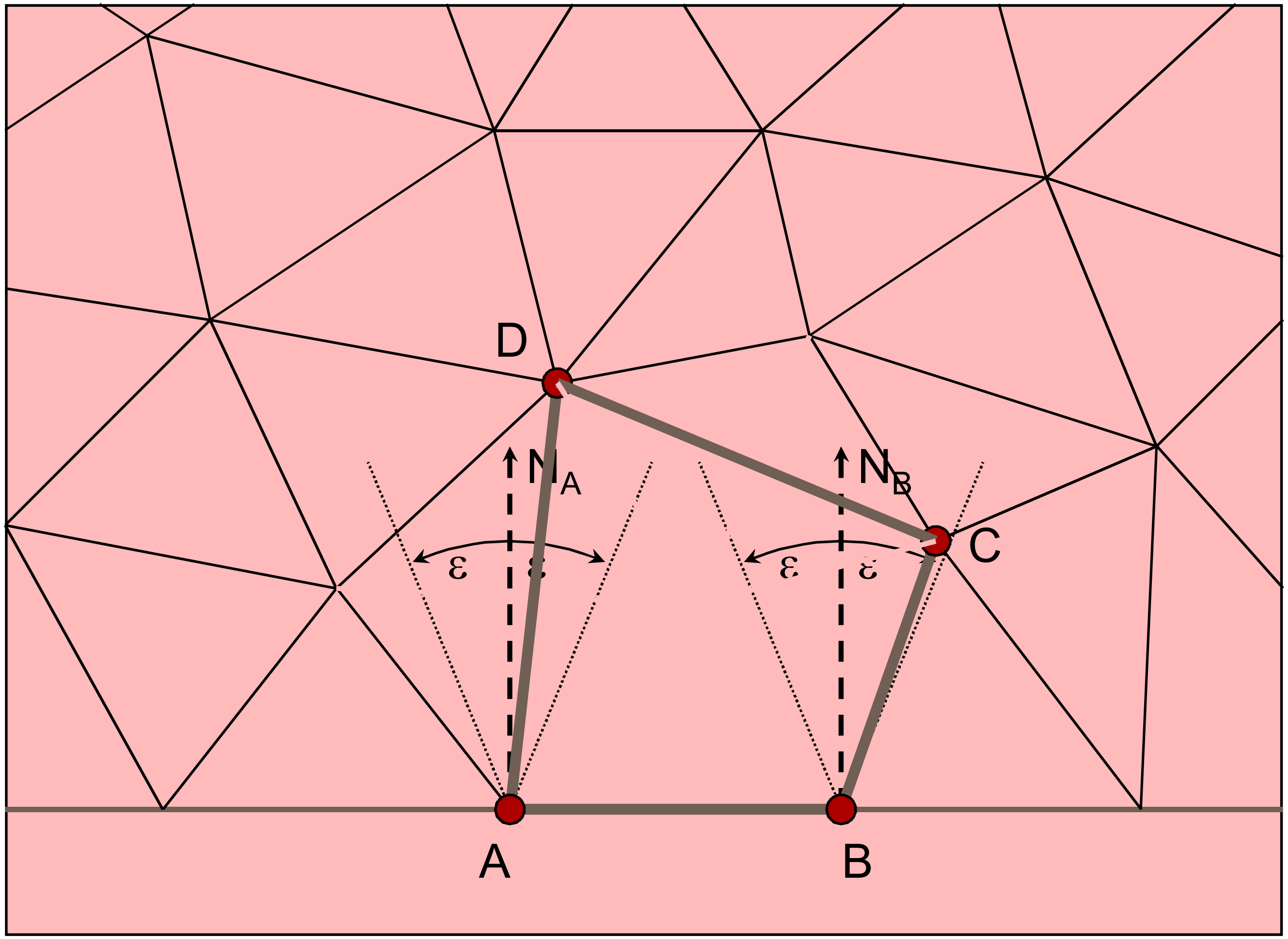
- Merging begins at a boundary
- Advances from one set of triangles to the next
- Attempts to maintain even number of intervals on any loop
- Can produce all-quad mesh
- Can also incorporate triangle splitting
- (Lee and Lo, 94)



Q-Morph

Triangle Merge with local transformations

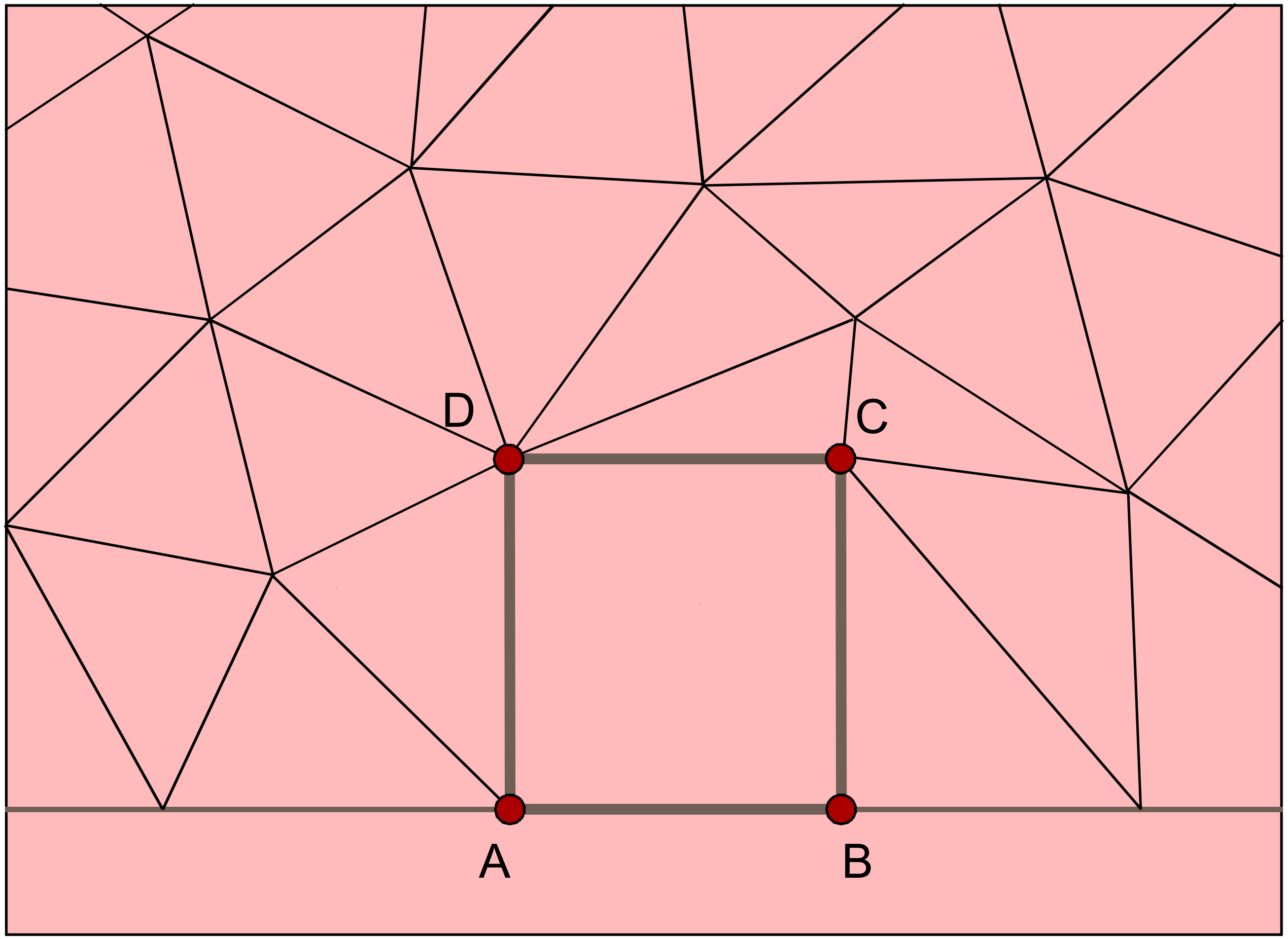
- Uses an advancing front approach
- Local swaps applied to improve resulting quad
- Any number of triangles merged to create a quad
- Attempts to maintain even number of intervals on any loop
- Produces all-quad mesh from even intervals
- (Owen, 99)



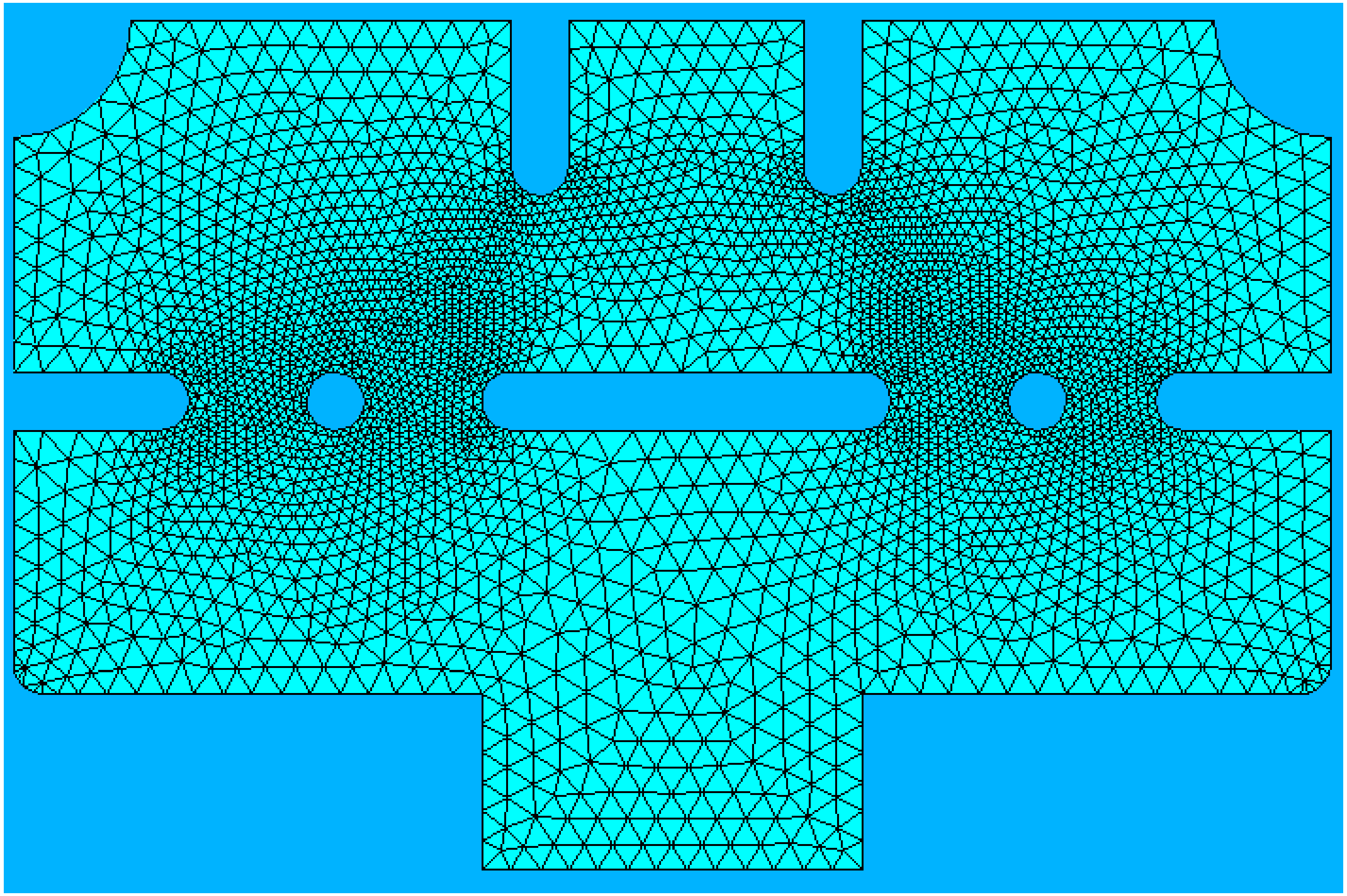
Q-Morph

Triangle Merge with local transformations

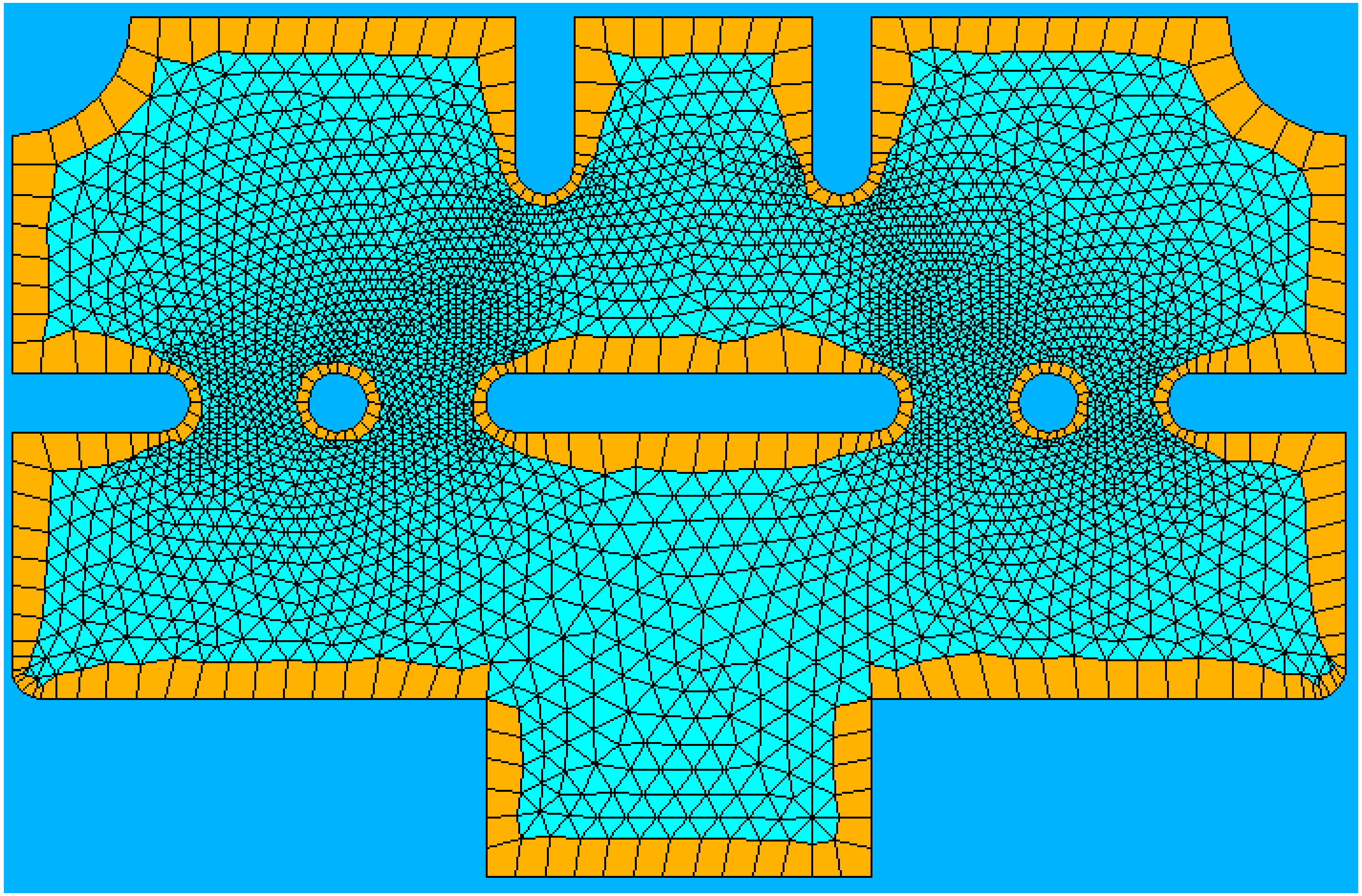
- Uses an advancing front approach
- Local swaps applied to improve resulting quad
- Any number of triangles merged to create a quad
- Attempts to maintain even number of intervals on any loop
- Produces all-quad mesh from even intervals
- (Owen, 99)



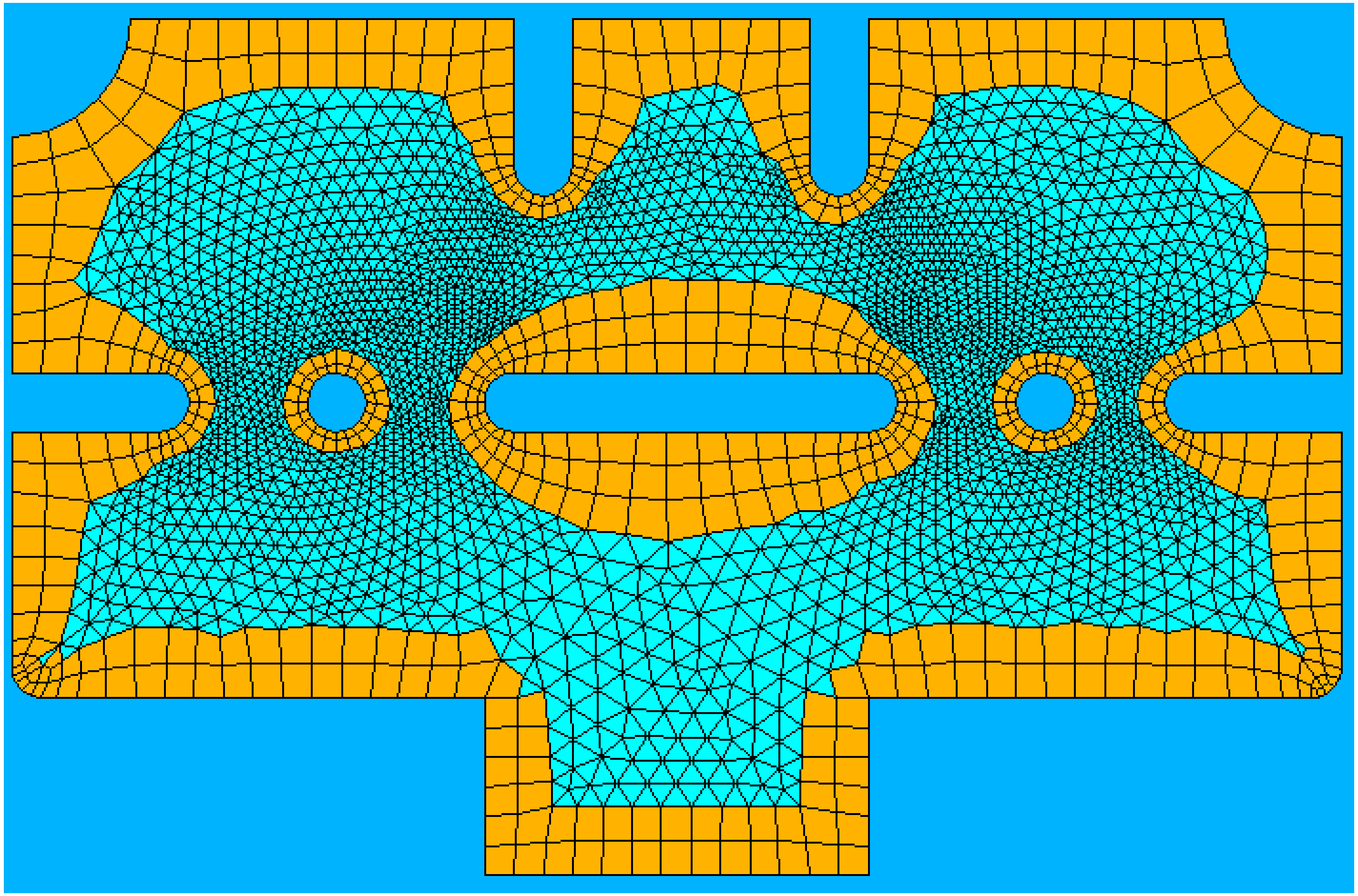
Q-Morph



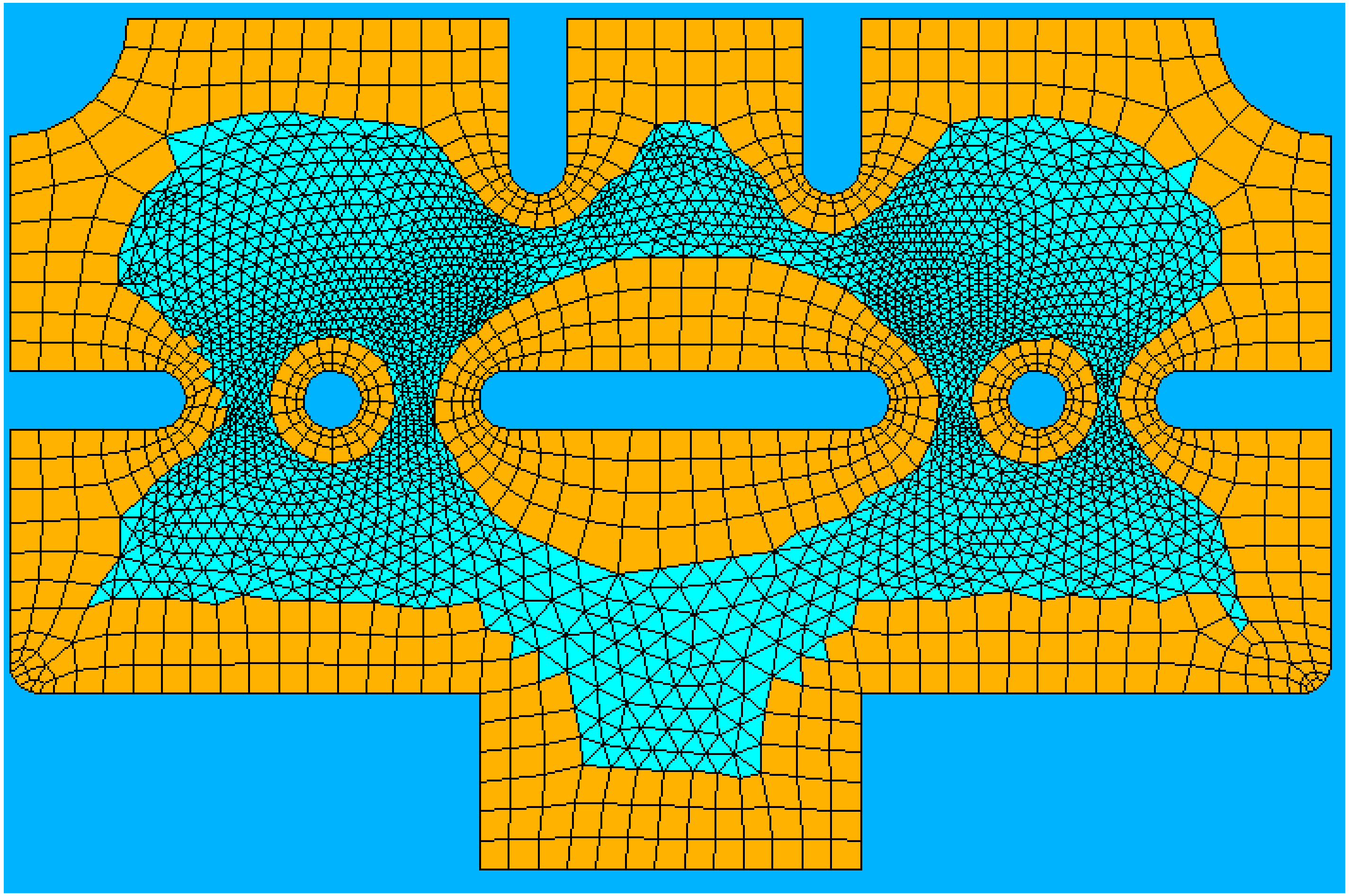
Q-Morph



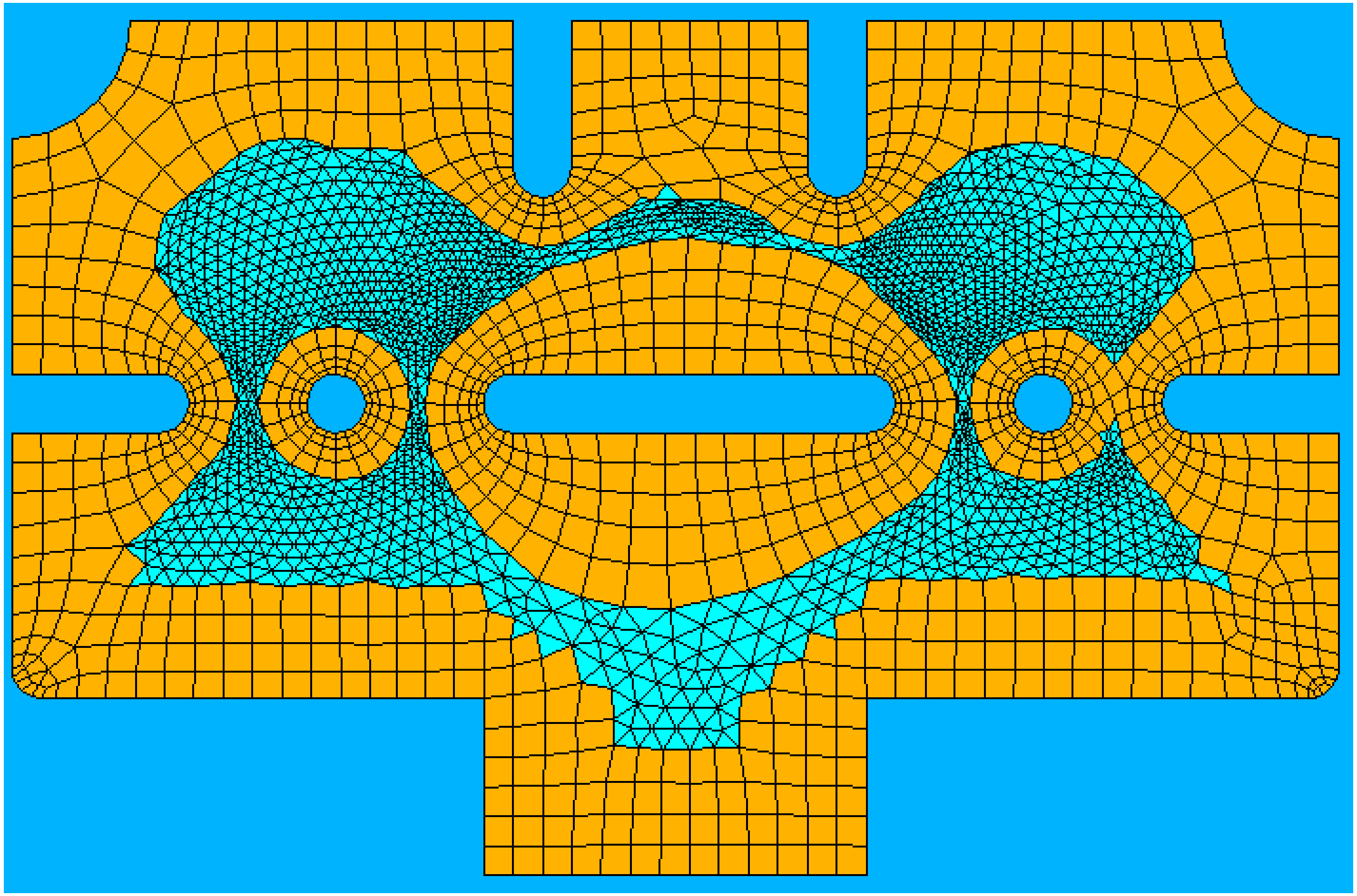
Q-Morph



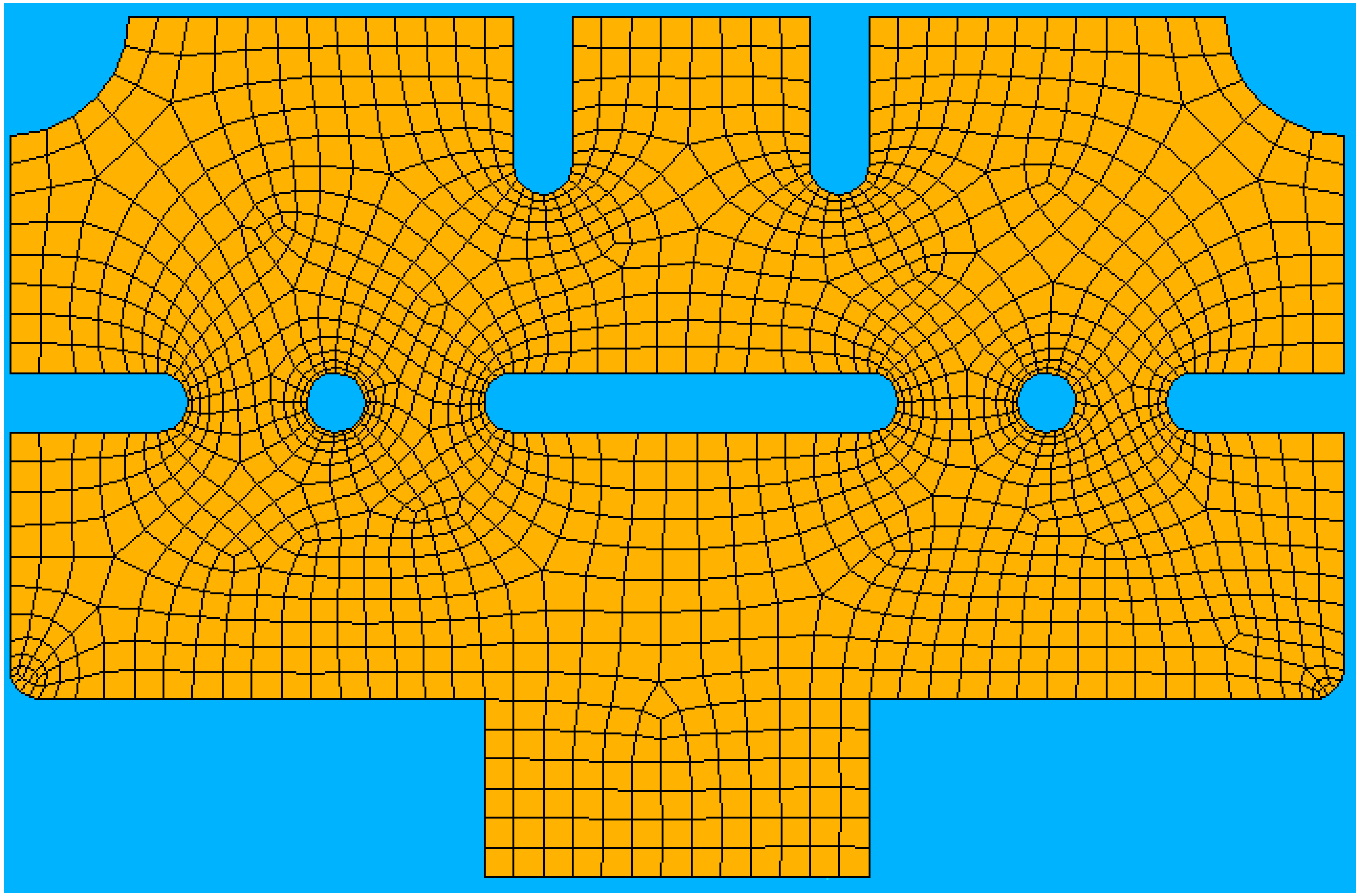
Q-Morph



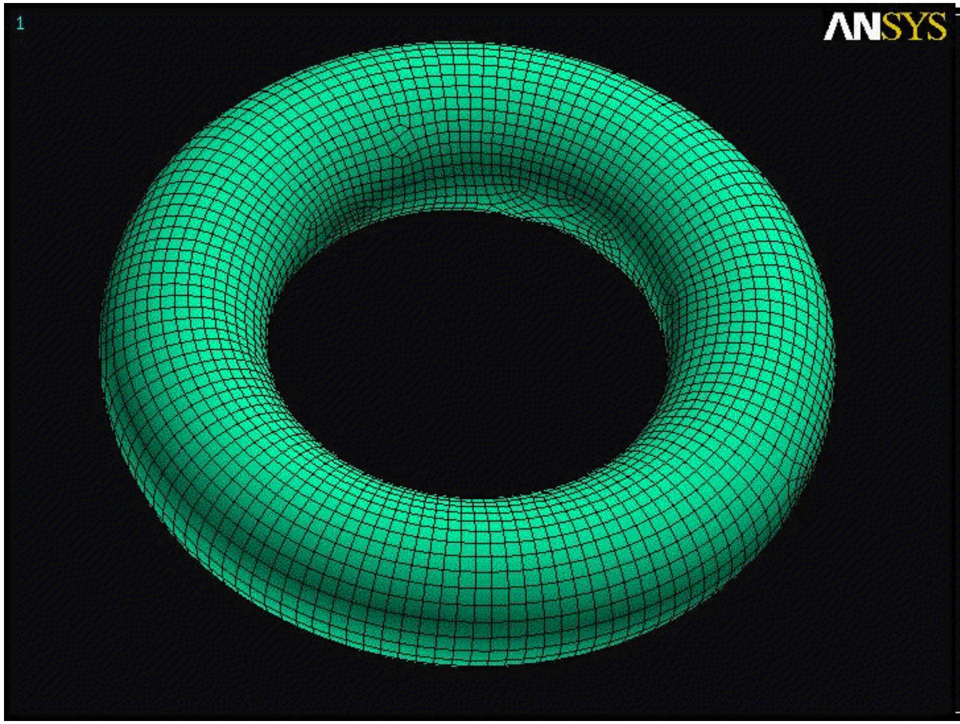
Q-Morph

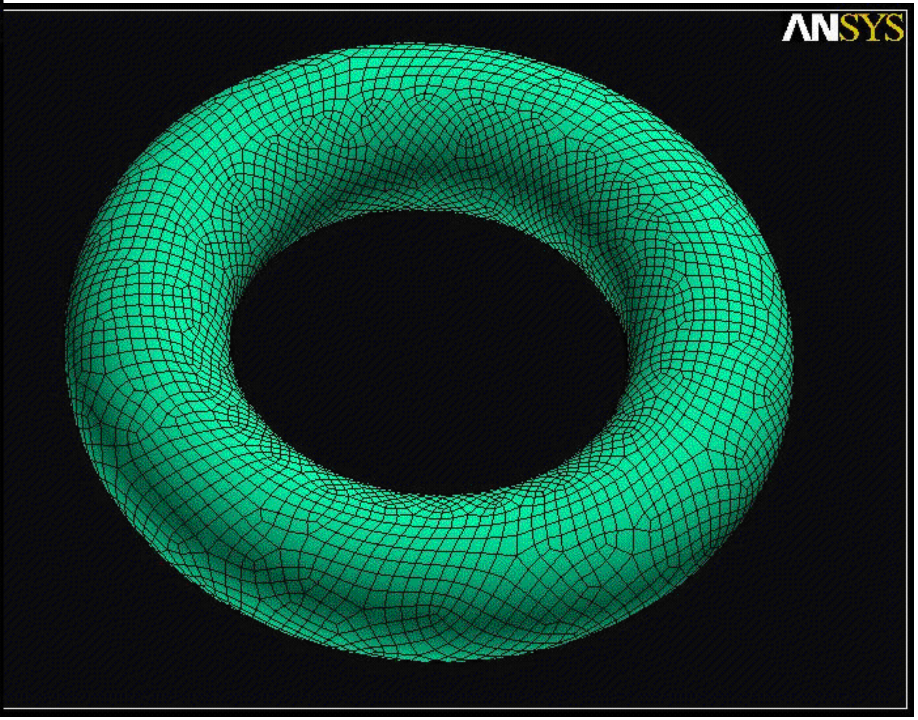


Q-Morph



Q-Morph

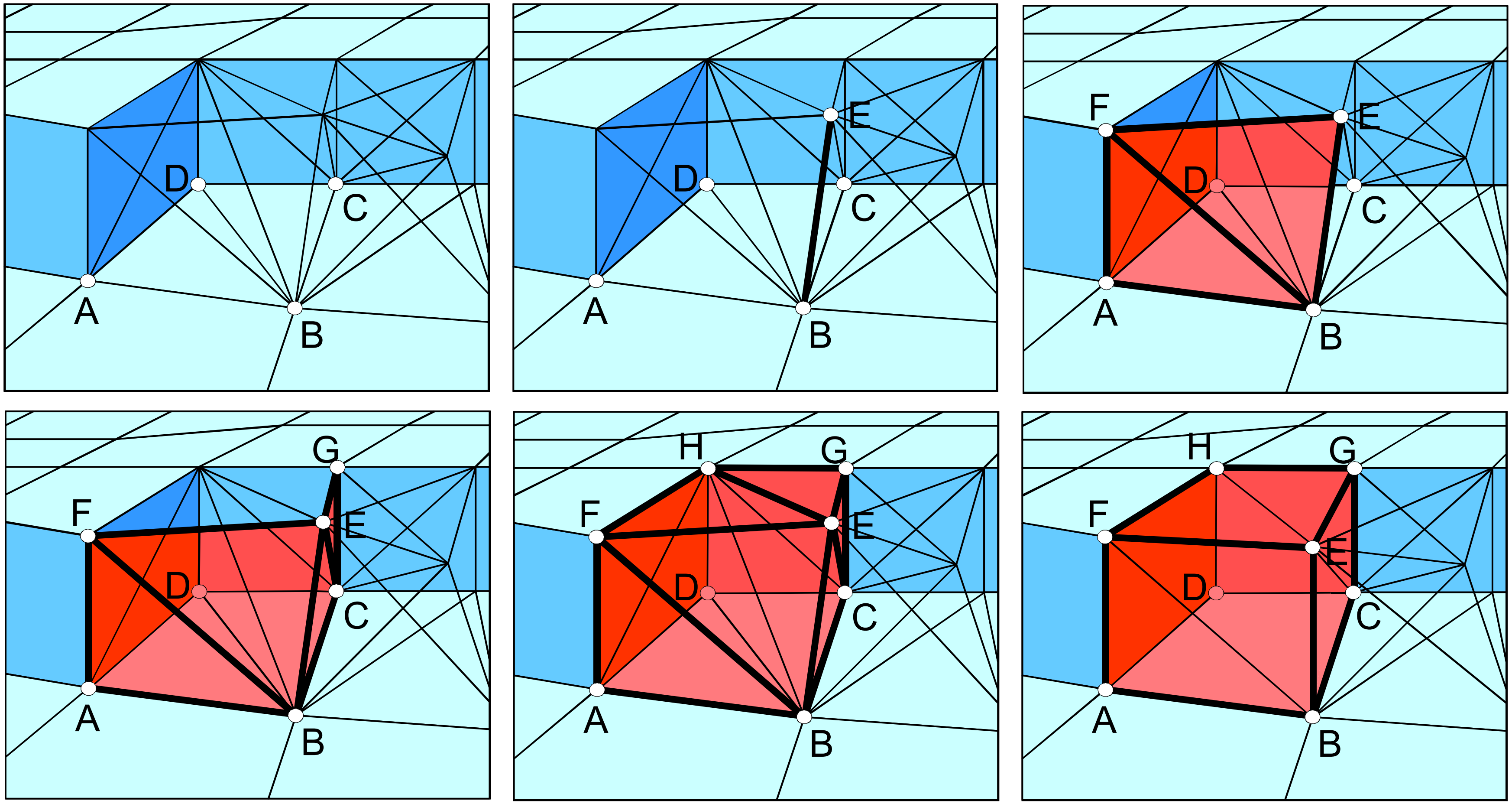




Q-Morph

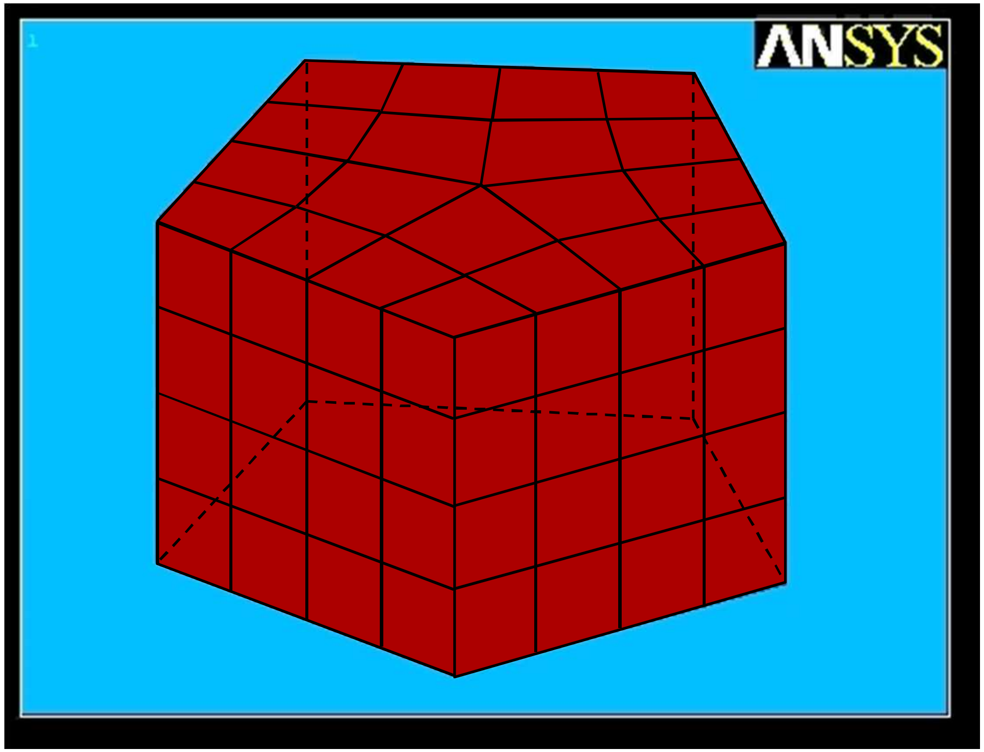
Lee,Lo Method

H-Morph



Edges and faces are recovered from a tet mesh to form the boundary of an element. Tets are then merged to form the hex.

H-Morph

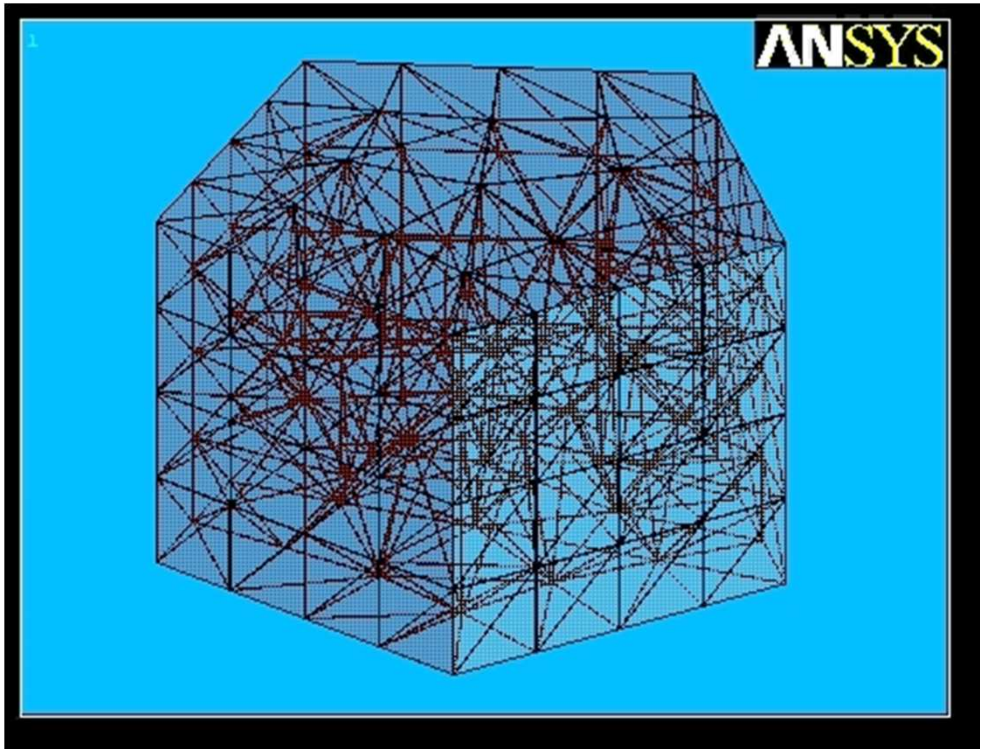


Example H-Morph meshing procedure

“Hex-Dominant Meshing”

(Owen, 00)

H-Morph

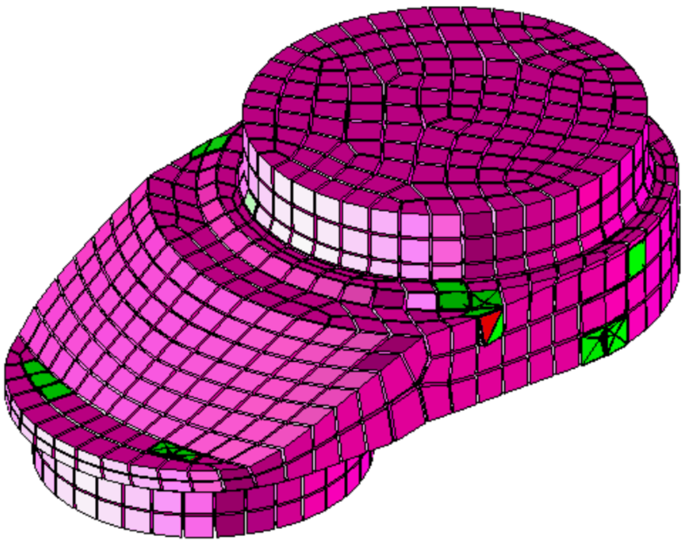


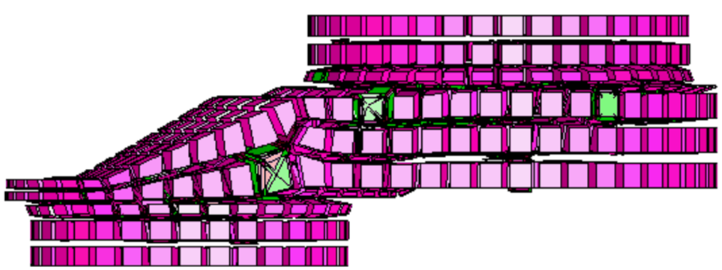
Example H-Morph meshing procedure

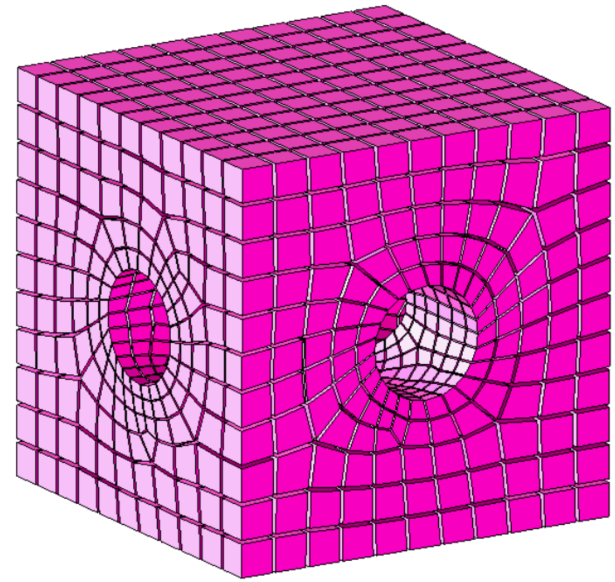
“Hex-Dominant Meshing”

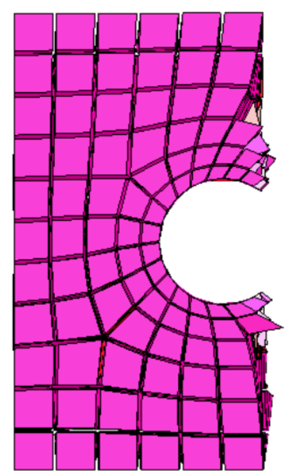
(Owen, 00)

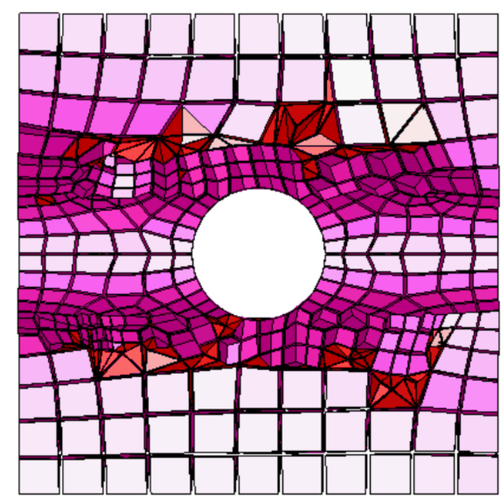
H-Morph



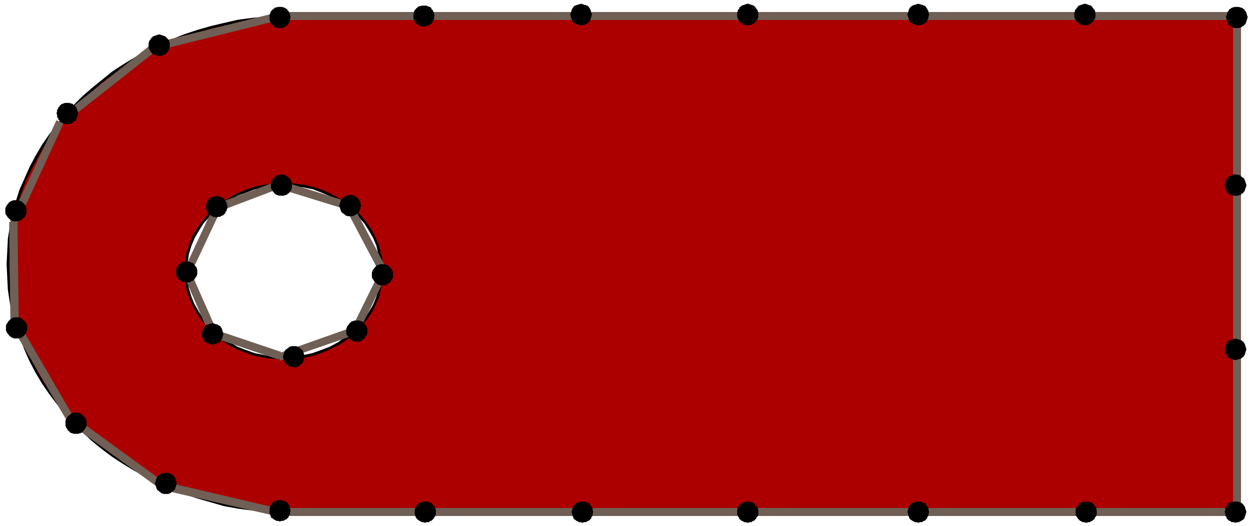








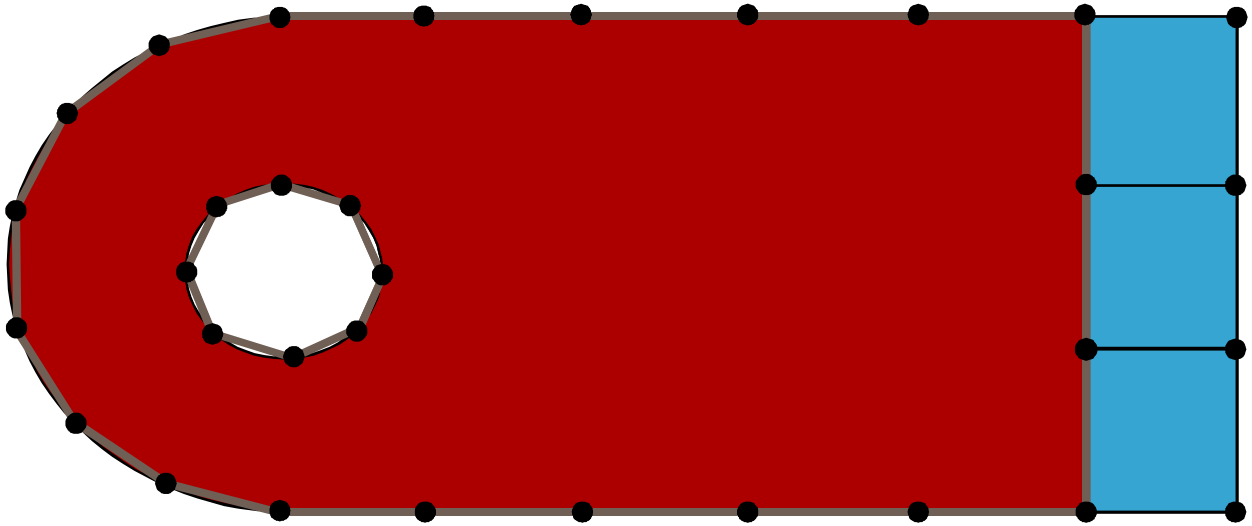
Paving



- Advancing Front: Begins with front at boundary
- Forms rows of elements based on front angles
- Must have even number of intervals for all-quad mesh

(Blacker,92)(Cass,96)

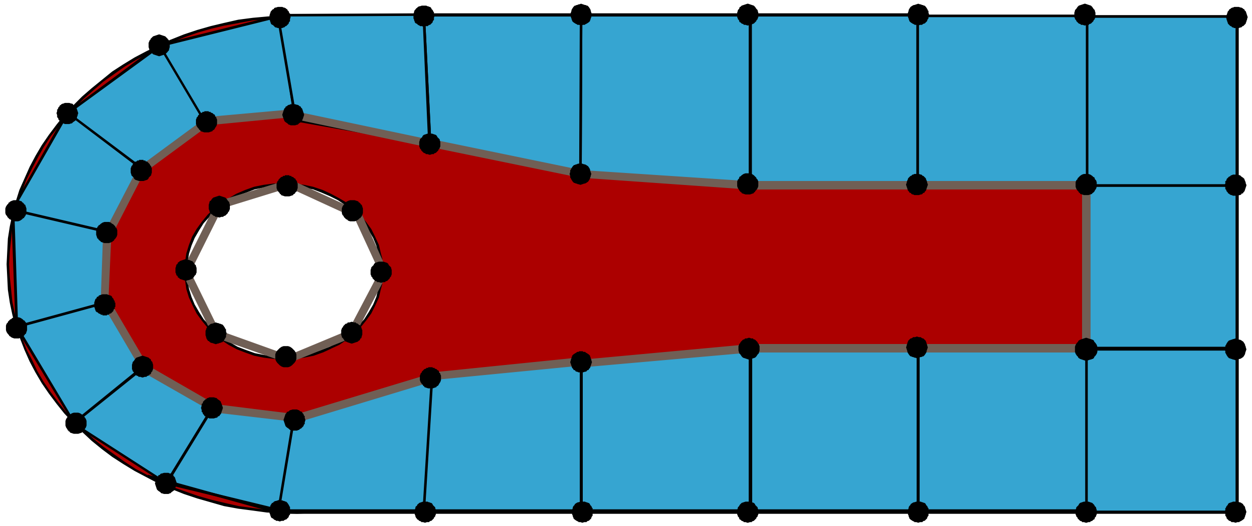
Paving



- Advancing Front: Begins with front at boundary
- Forms rows of elements based on front angles
- Must have even number of intervals for all-quad mesh

(Blacker,92)(Cass,96)

Paving

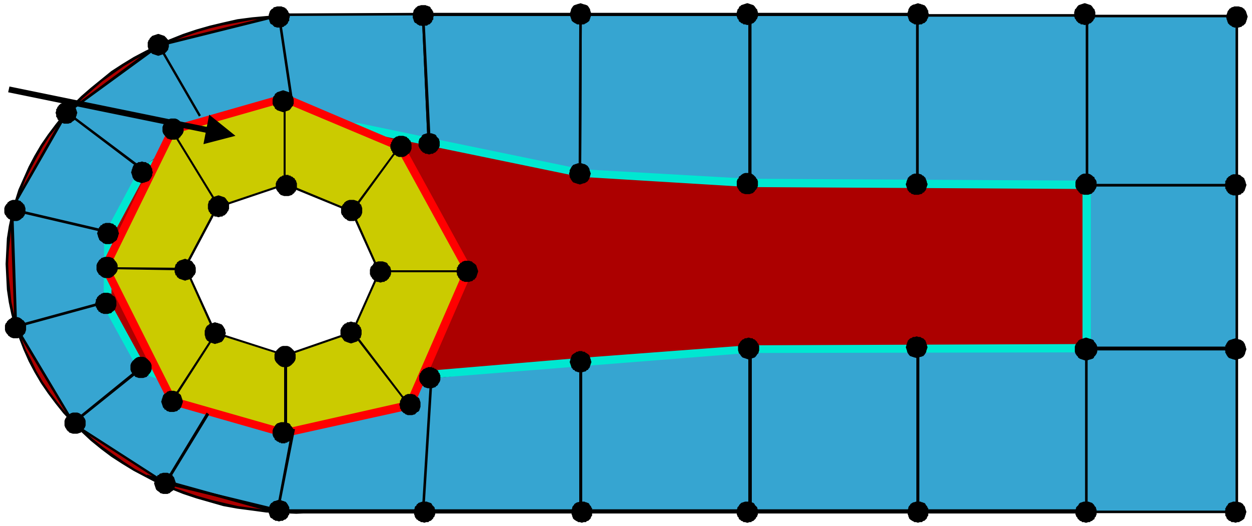


•Advancing Front: Begins with front at boundary
•Forms rows of elements based on front angles
•Must have even number of intervals for all-quad mesh

(Blacker,92)(Cass,96)

Paving

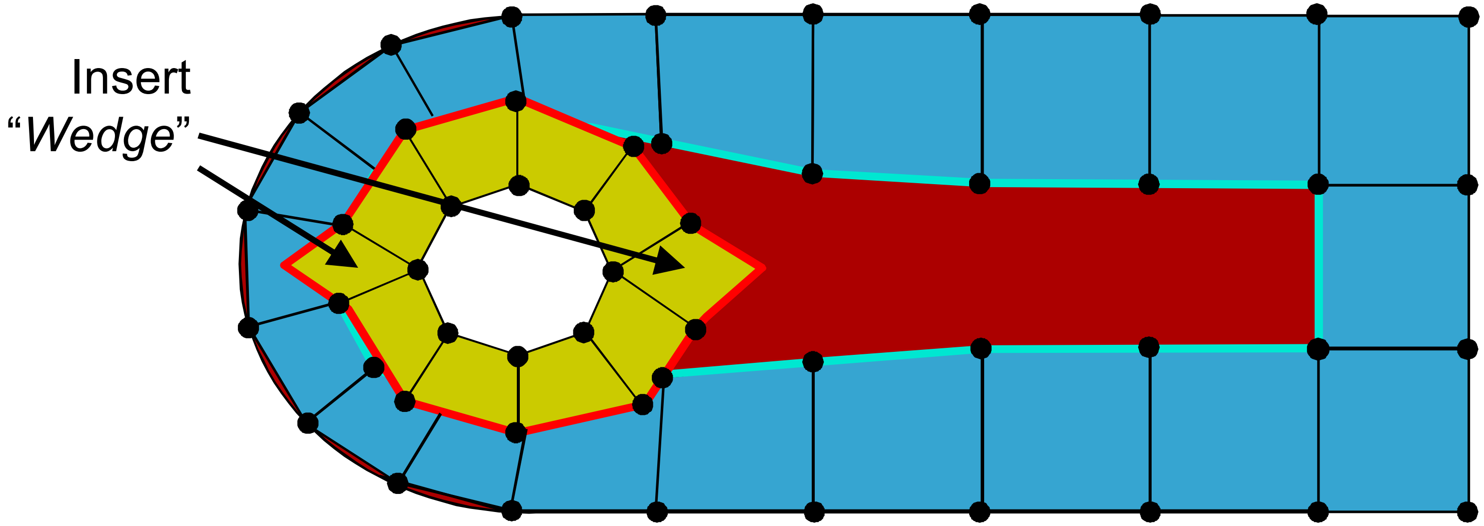
Form new row and check for overlap



•Advancing Front: Begins with front at boundary
•Forms rows of elements based on front angles
•Must have even number of intervals for all-quad mesh

(Blacker,92)(Cass,96)

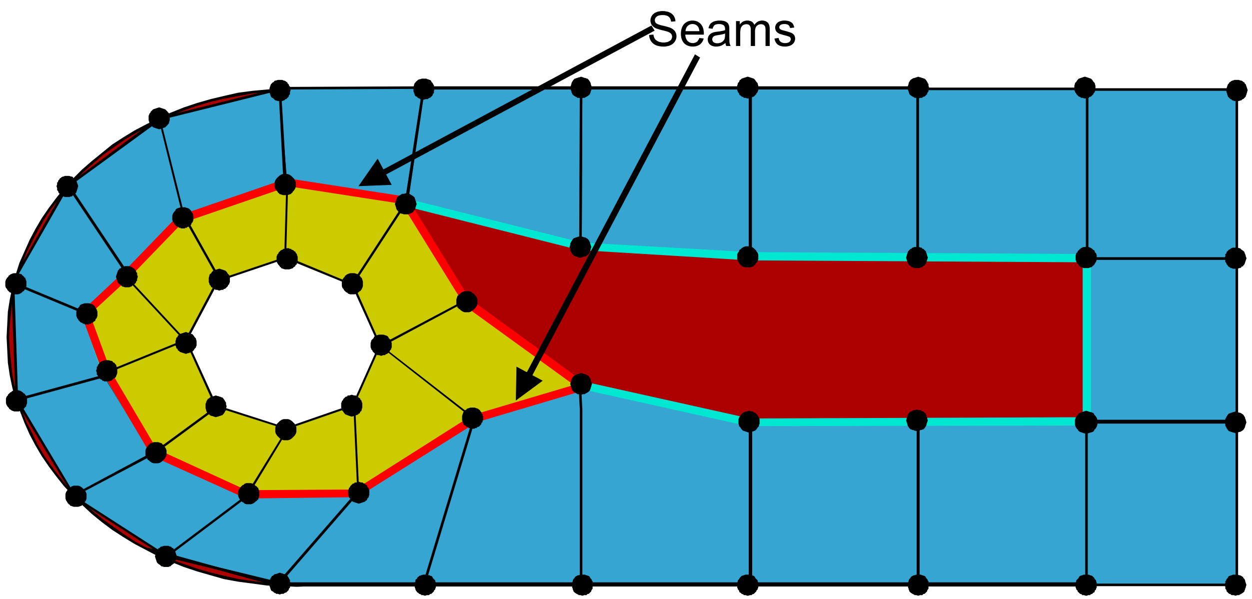
Paving



- Advancing Front: Begins with front at boundary
- Forms rows of elements based on front angles
- Must have even number of intervals for all-quad mesh

(Blacker,92)(Cass,96)

Paving

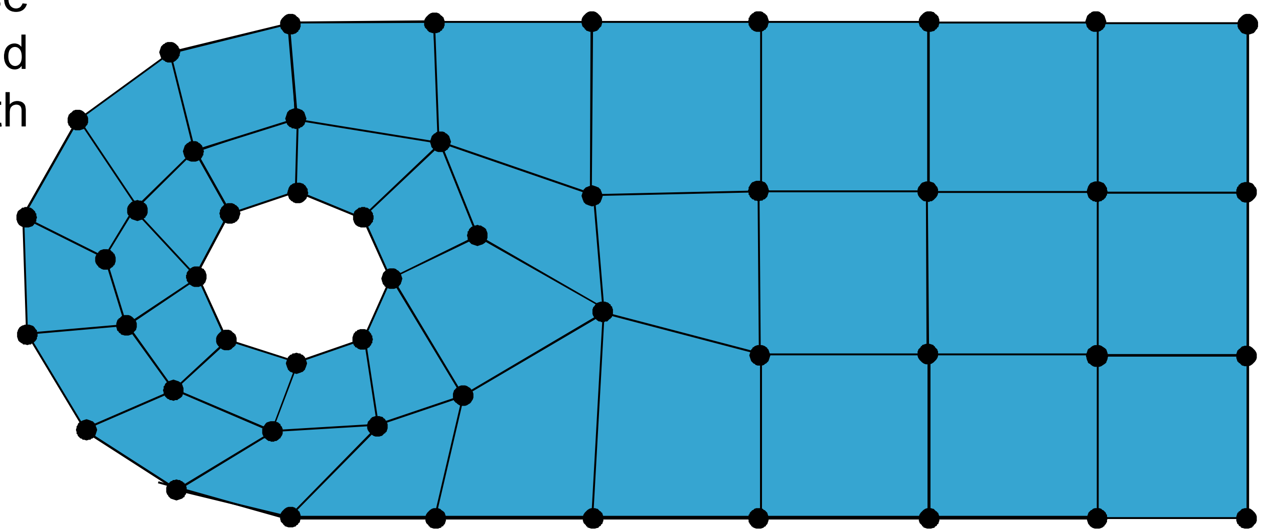


- Advancing Front: Begins with front at boundary
- Forms rows of elements based on front angles
- Must have even number of intervals for all-quad mesh

(Blacker,92)(Cass,96)

Paving

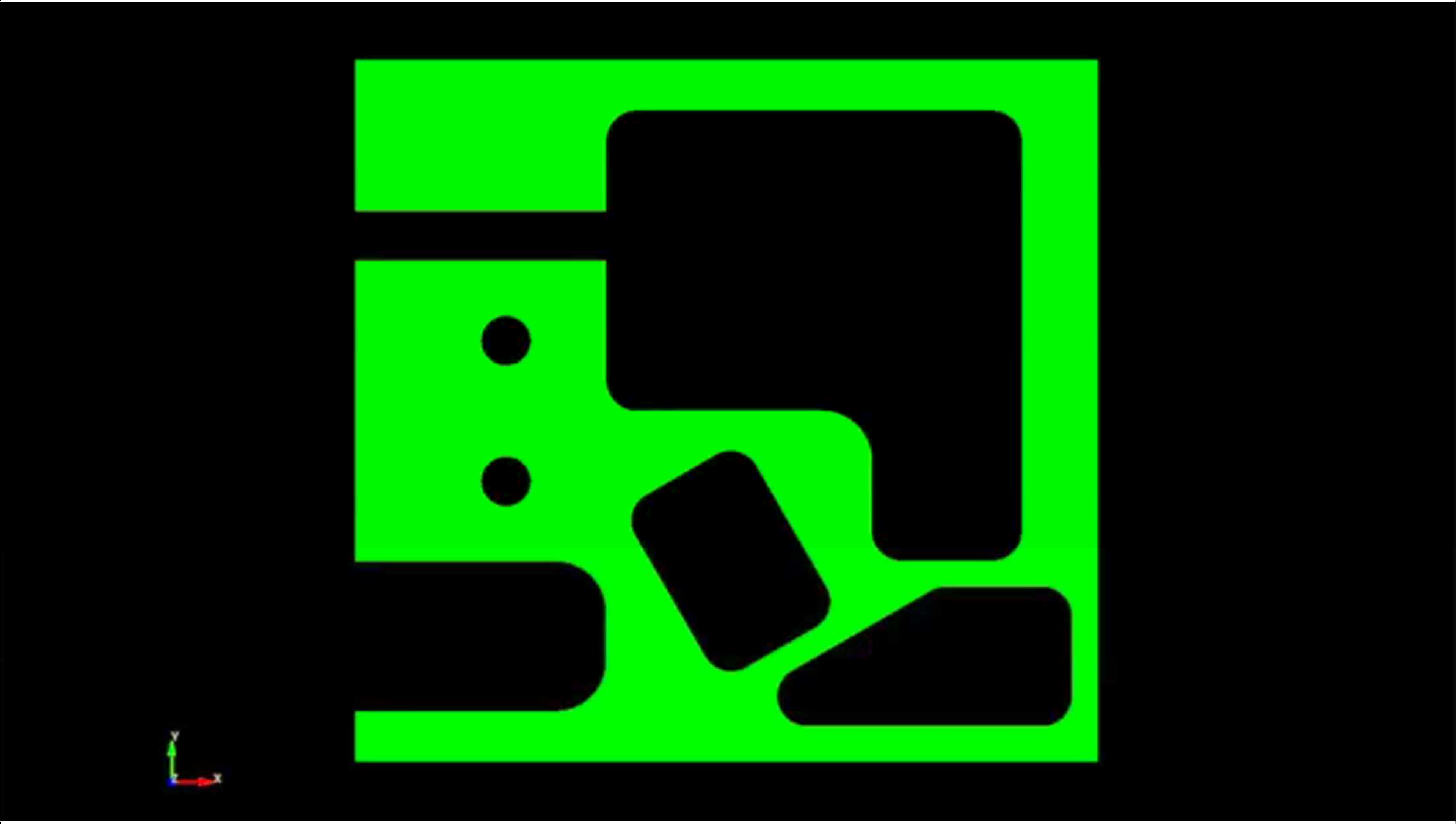
Close
Loops and
smooth



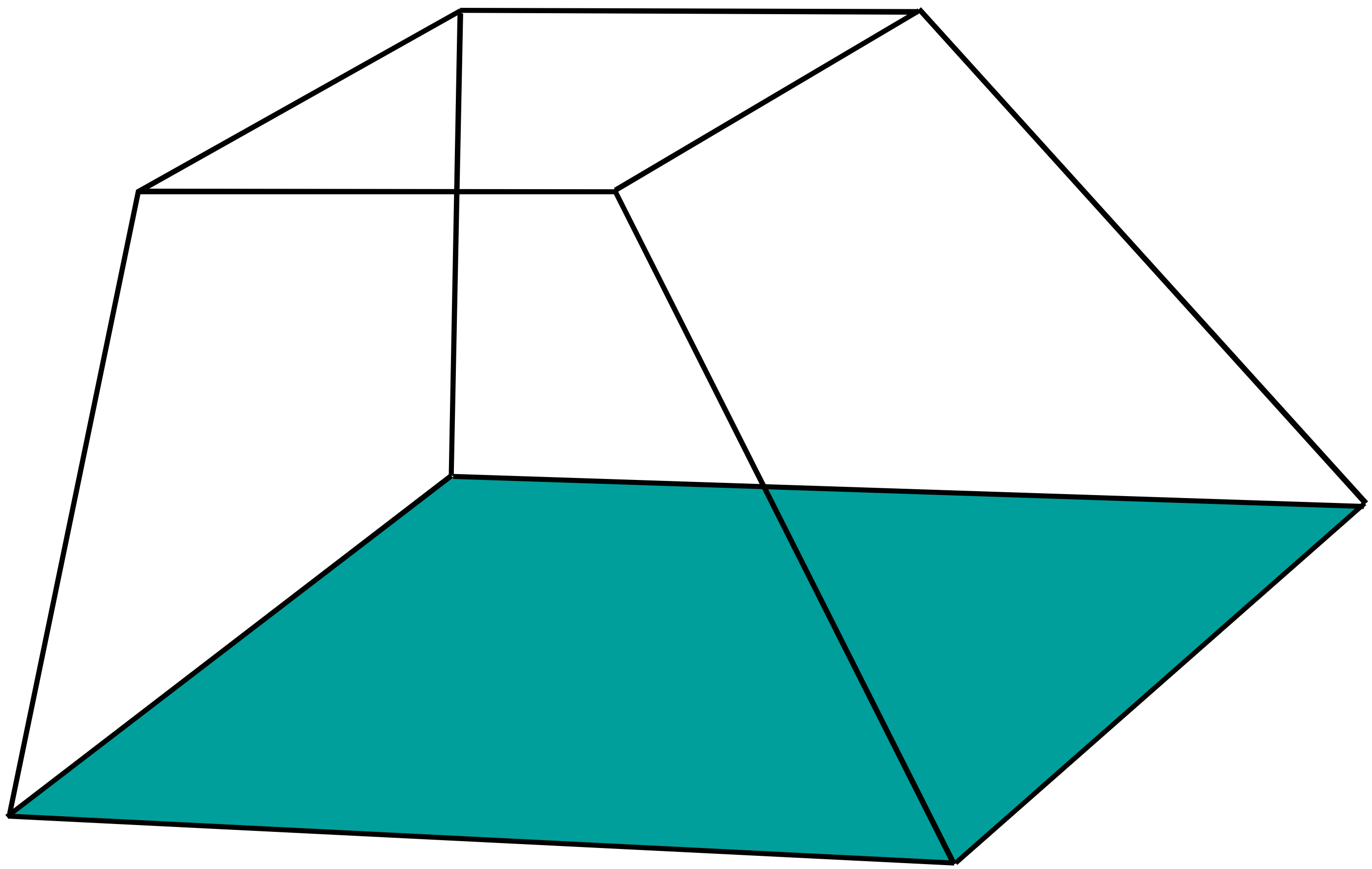
•Advancing Front: Begins with front at boundary
•Forms rows of elements based on front angles
•Must have even number of intervals for all-quad mesh

(Blacker,92)(Cass,96)

Paving



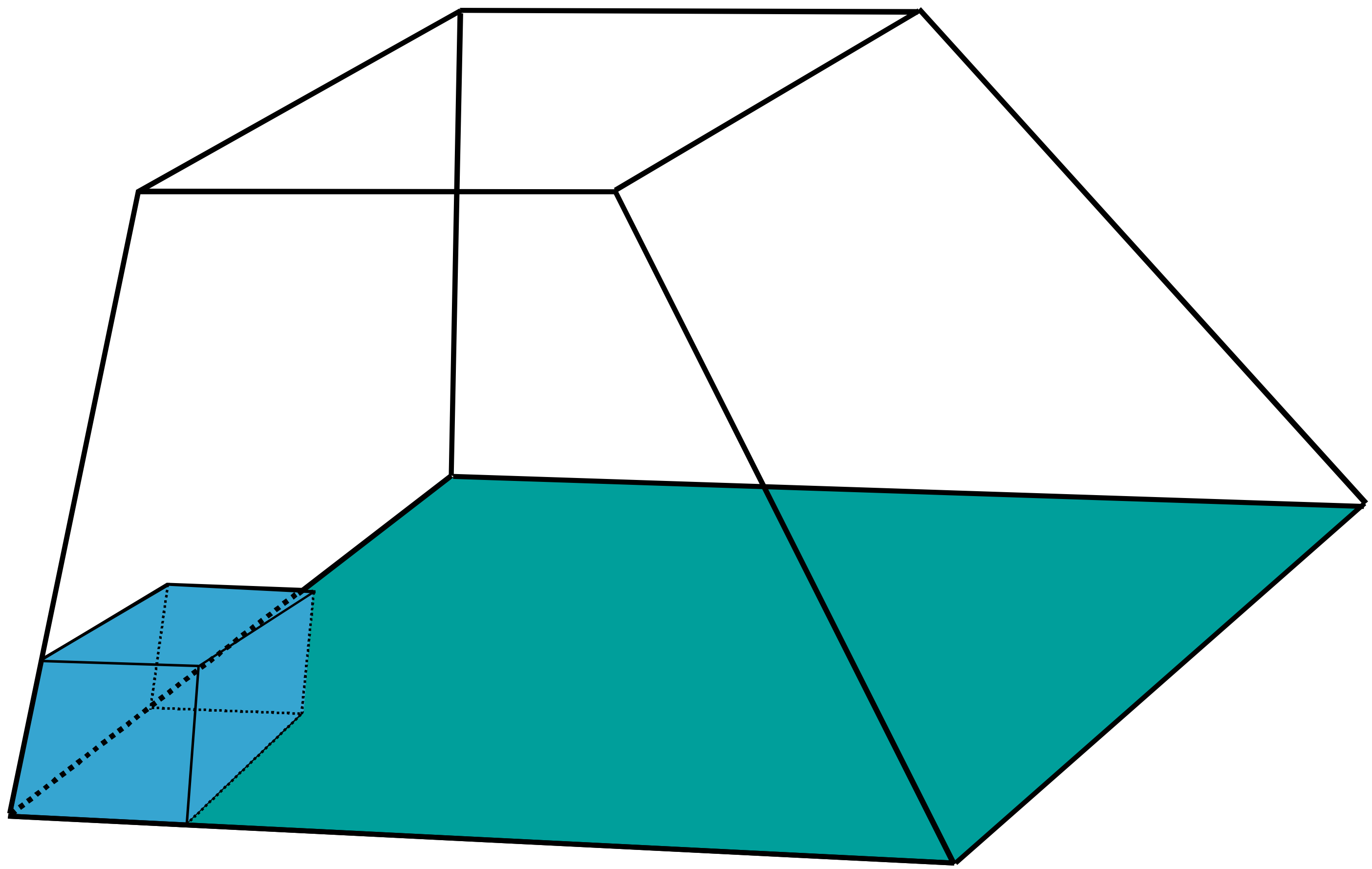
Plastering



•3D extension of “paving”
•Row-by row or element-by-element

(Blacker, 93)

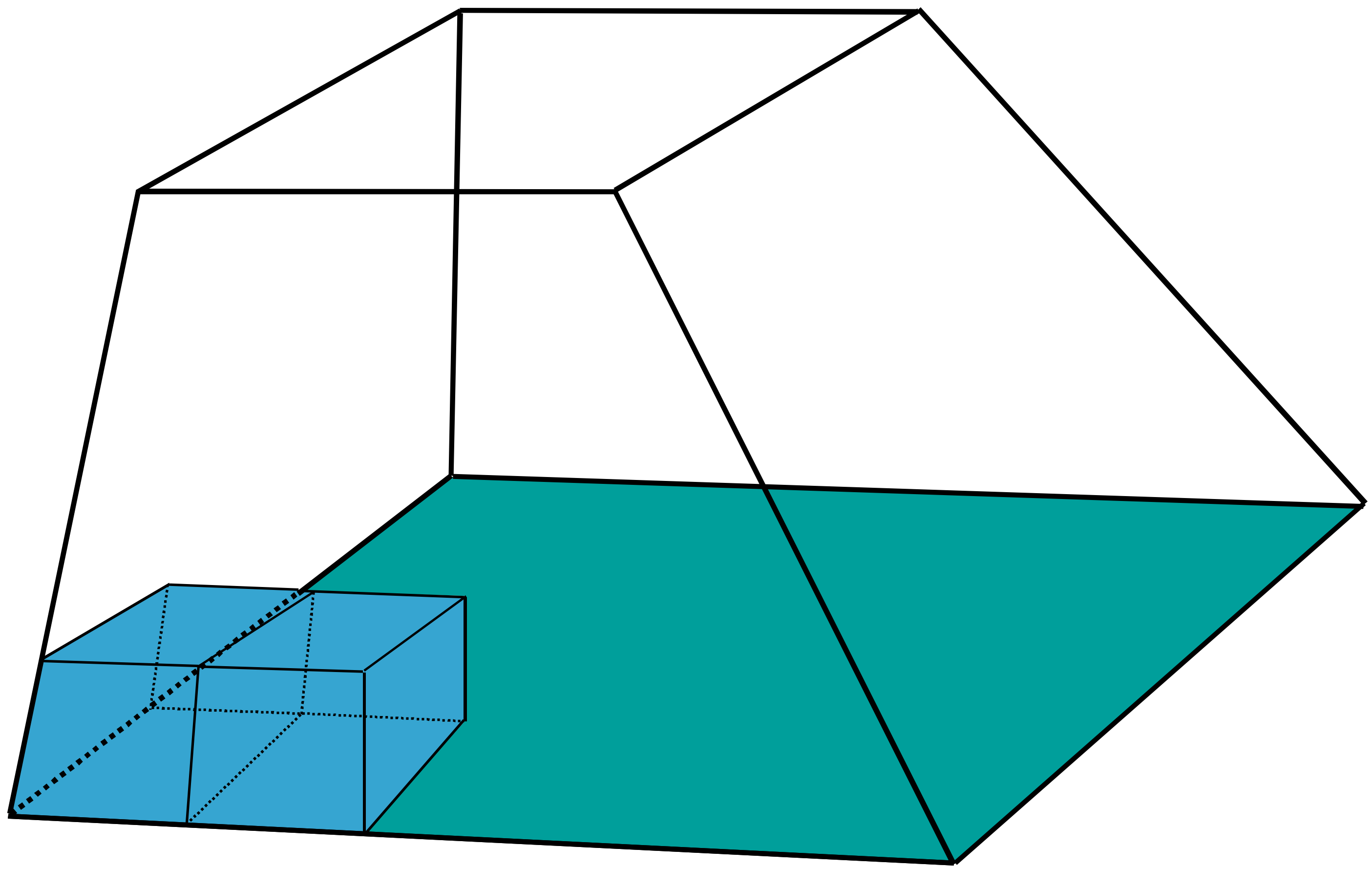
Plastering



•3D extension of “paving”
•Row-by row or element-by-element

(Blacker, 93)

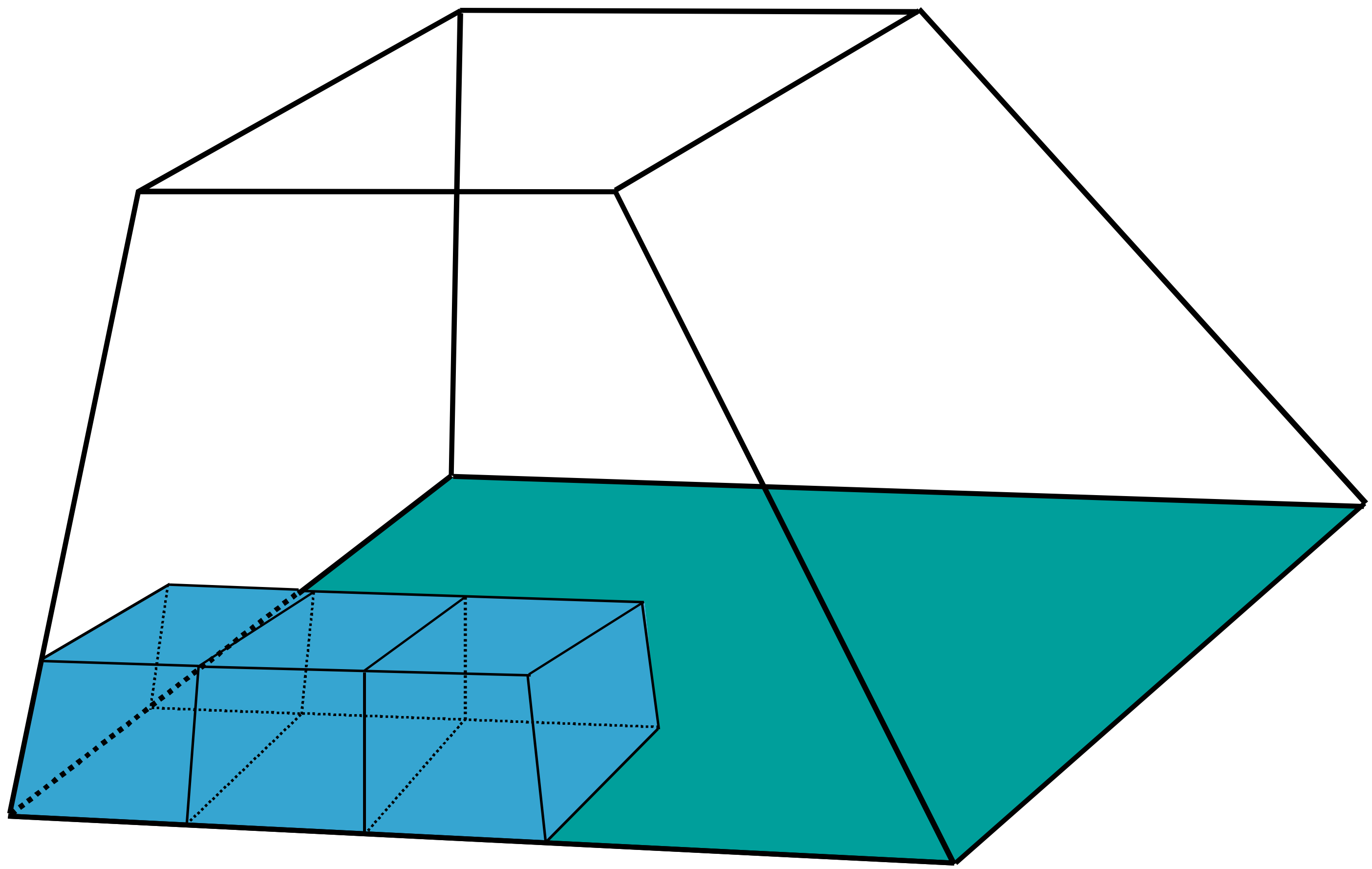
Plastering



•3D extension of “paving”
•Row-by row or element-by-element

(Blacker, 93)

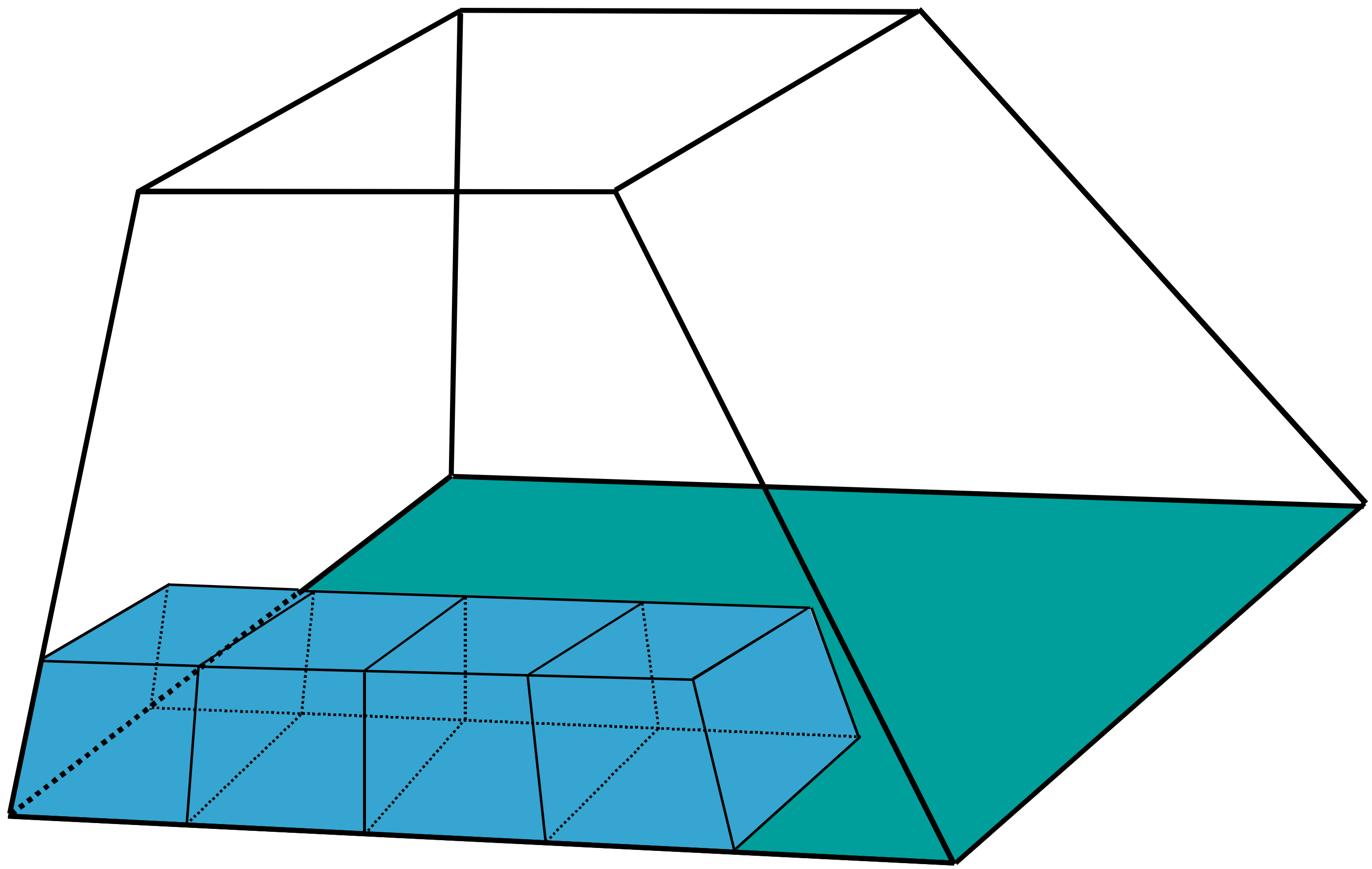
Plastering



(Blacker, 93)

- 3D extension of “paving”
- Row-by row or element-by-element

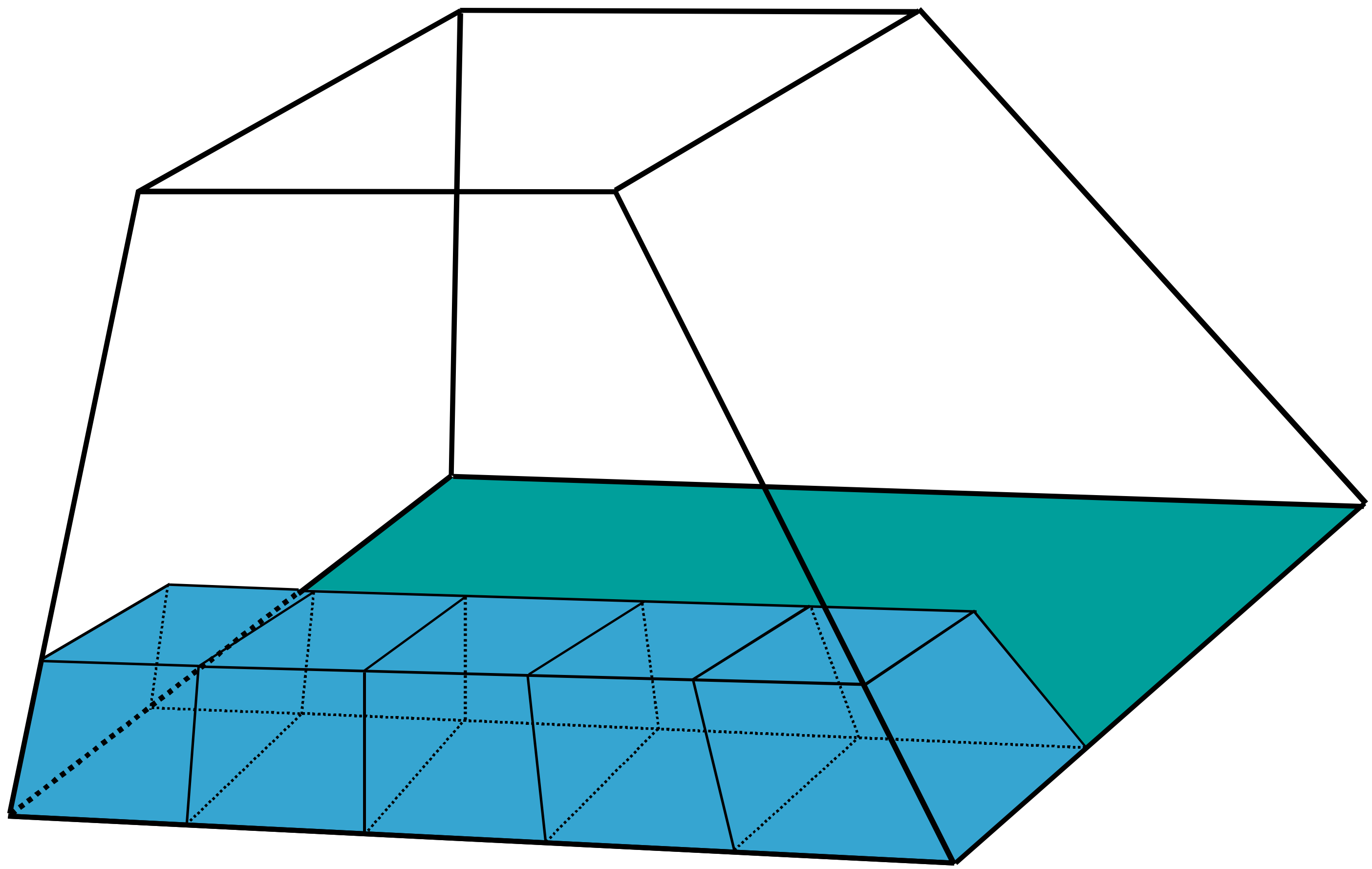
Plastering



•3D extension of “paving”
•Row-by row or element-by-element

(Blacker, 93)

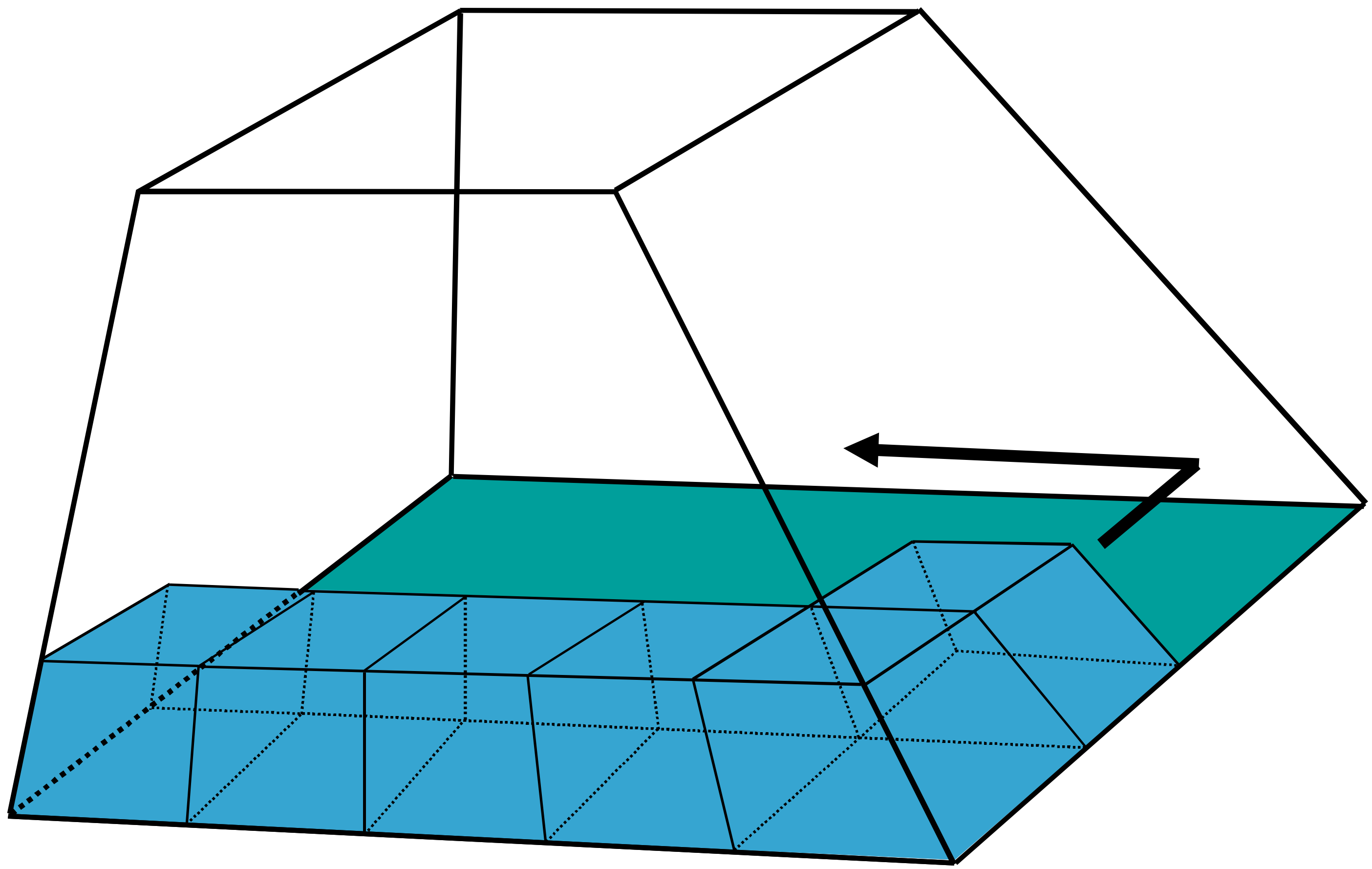
Plastering



•3D extension of “paving”
•Row-by row or element-by-element

(Blacker, 93)

Plastering

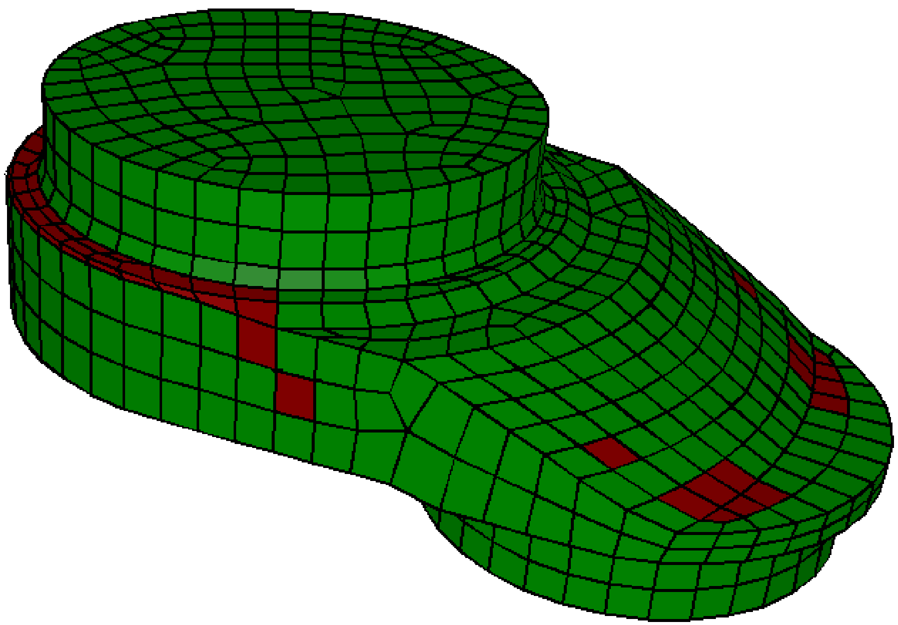


•3D extension of “paving”
•Row-by row or element-by-element

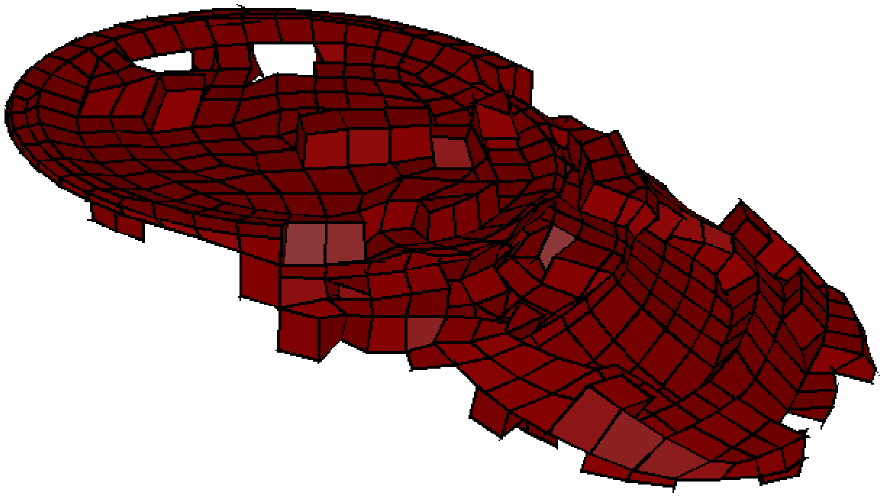
(Blacker, 93)

Hex-tet plastering

Exterior Hex mesh



RemainingVoid



Ford Crankshaft

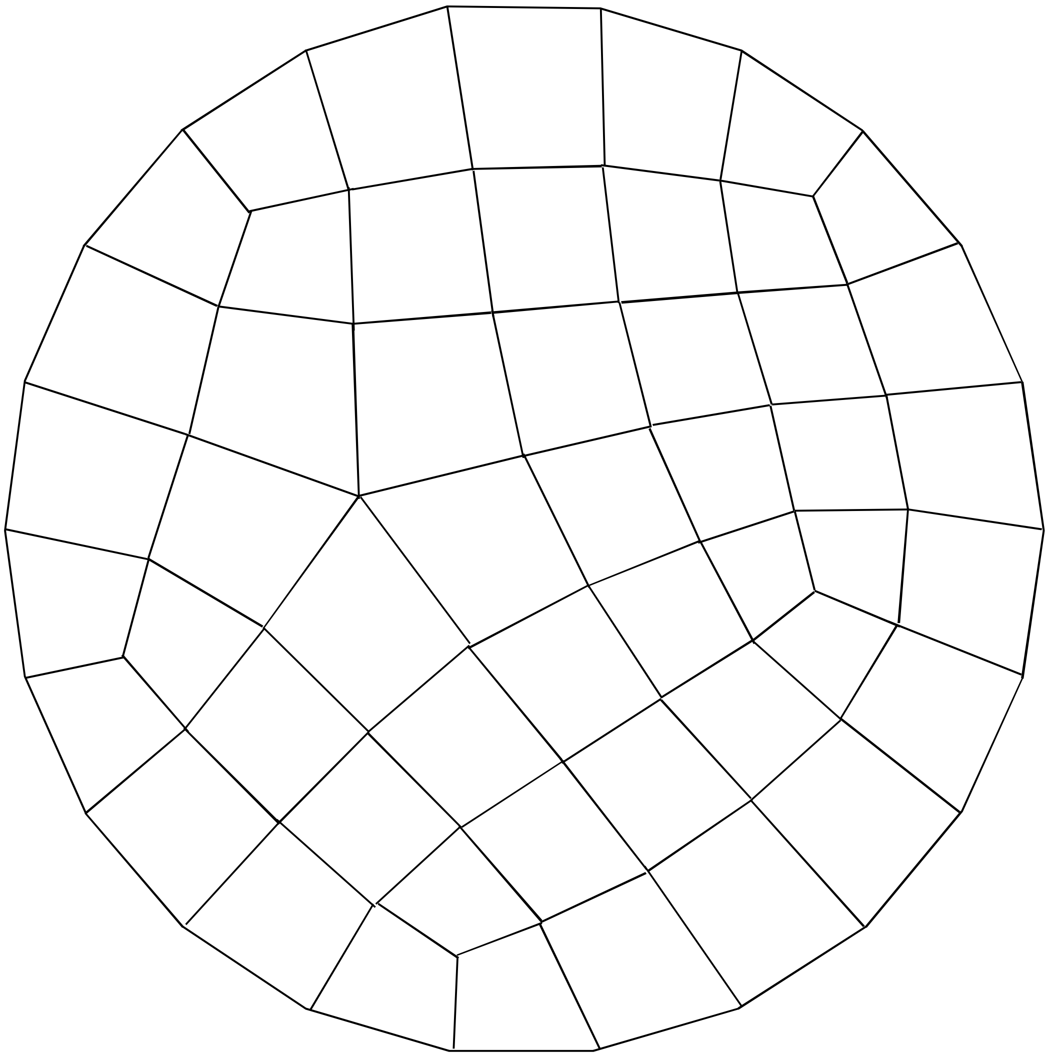
Plastering+Tet Meshing

“Hex-Dominant Meshing”

Quadrilateral Dual Representation

The elemental representation of a mesh, composed of elements, edges, and nodes, is known as the *primal*.

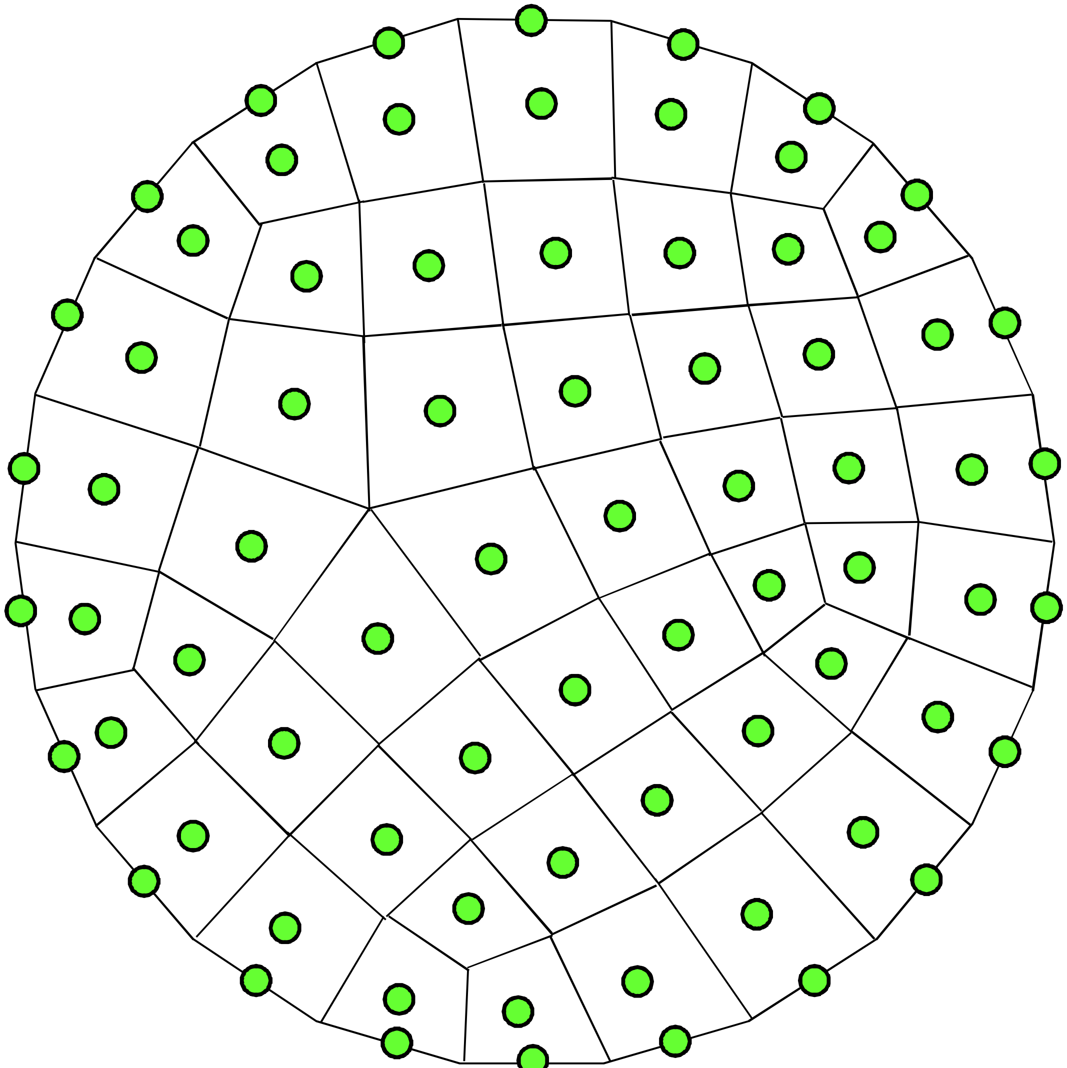
Quadrilateral meshes have a *dual* representation, similar to the voroni skeleton of a triangular delaunay mesh.



Quadrilateral Dual Representation

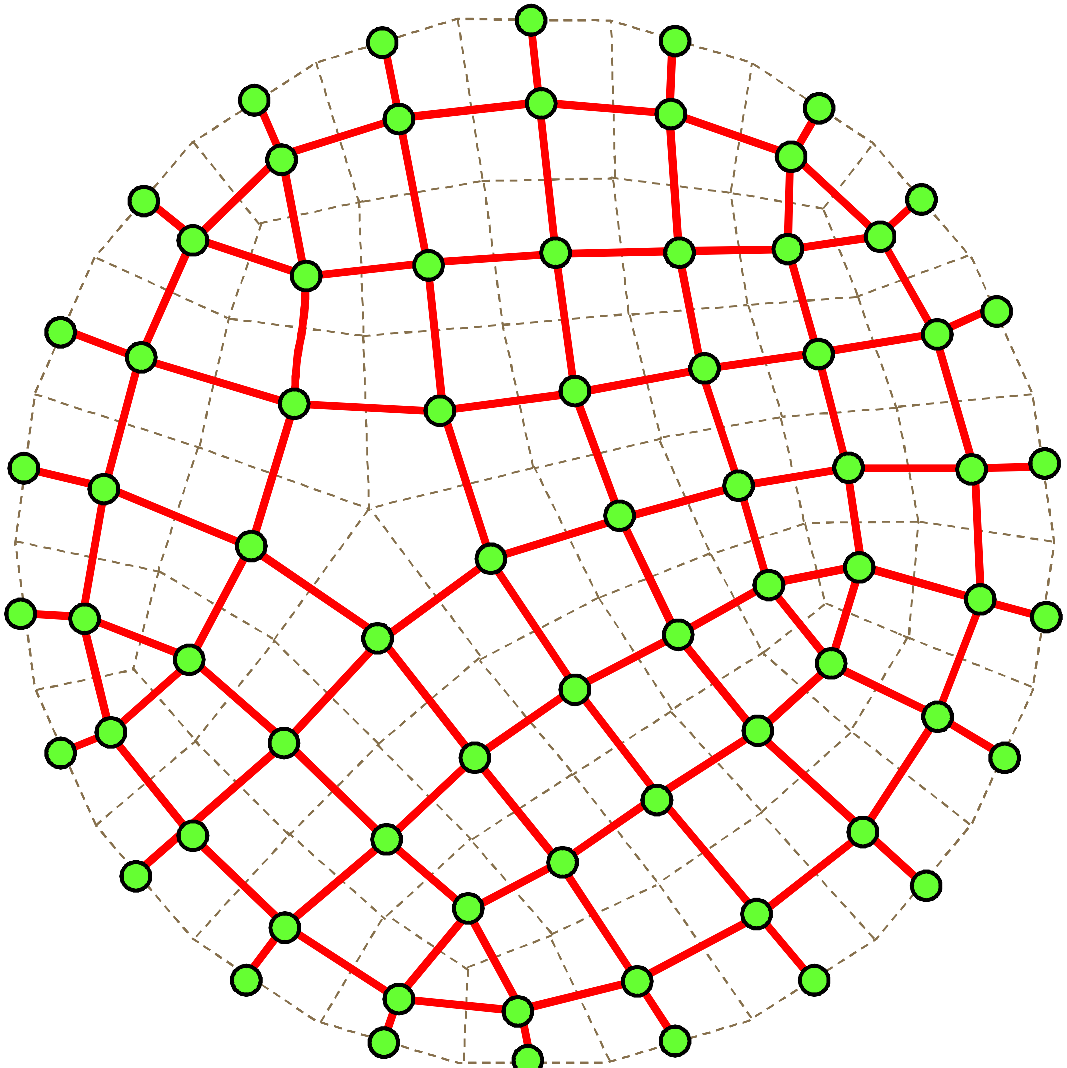
A dual vertex, v_i, is defined at the centroid of each quadrilateral element.

A dual vertex is also placed at the centroid of every boundary edge.



Quadrilateral Dual Representation

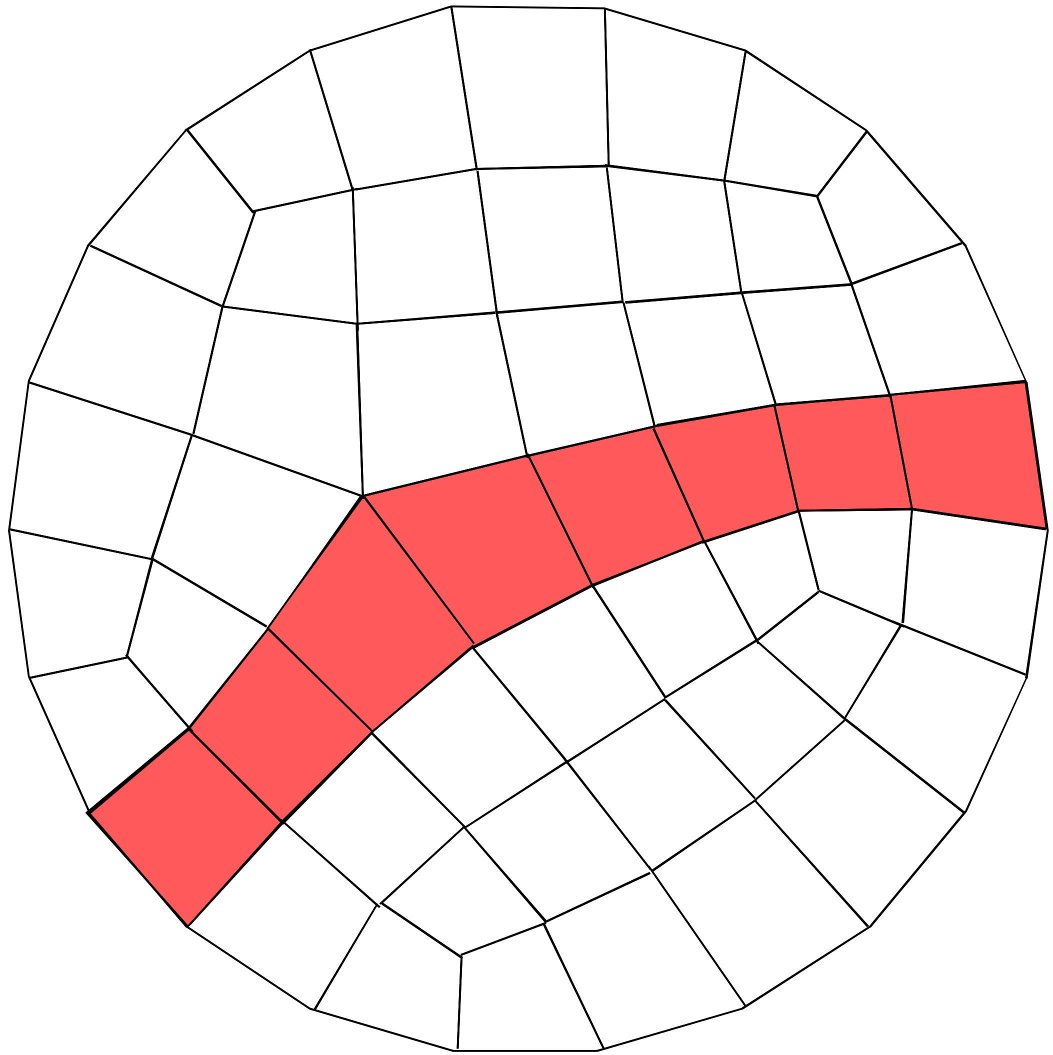
Connecting the dual vertices through adjacent elements creates the edges of the dual.



Quadrilateral Dual Representation

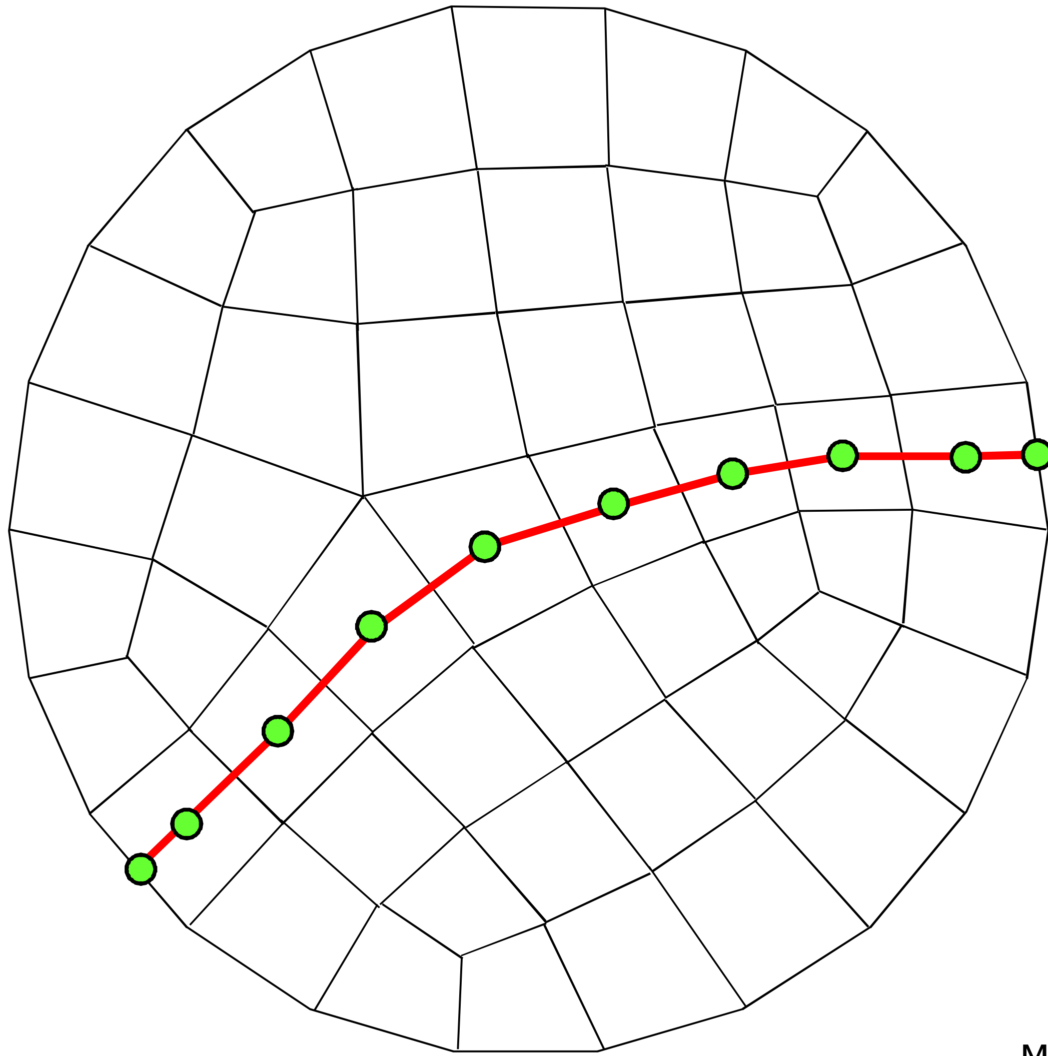
Quadrilateral meshes have an inherent row structure. The red quads illustrate one row.

Each row corresponds to one dual chord.



Quadrilateral Dual Representation

The set of all dual edges which connect quads in each row forms a dual chord, c_i.

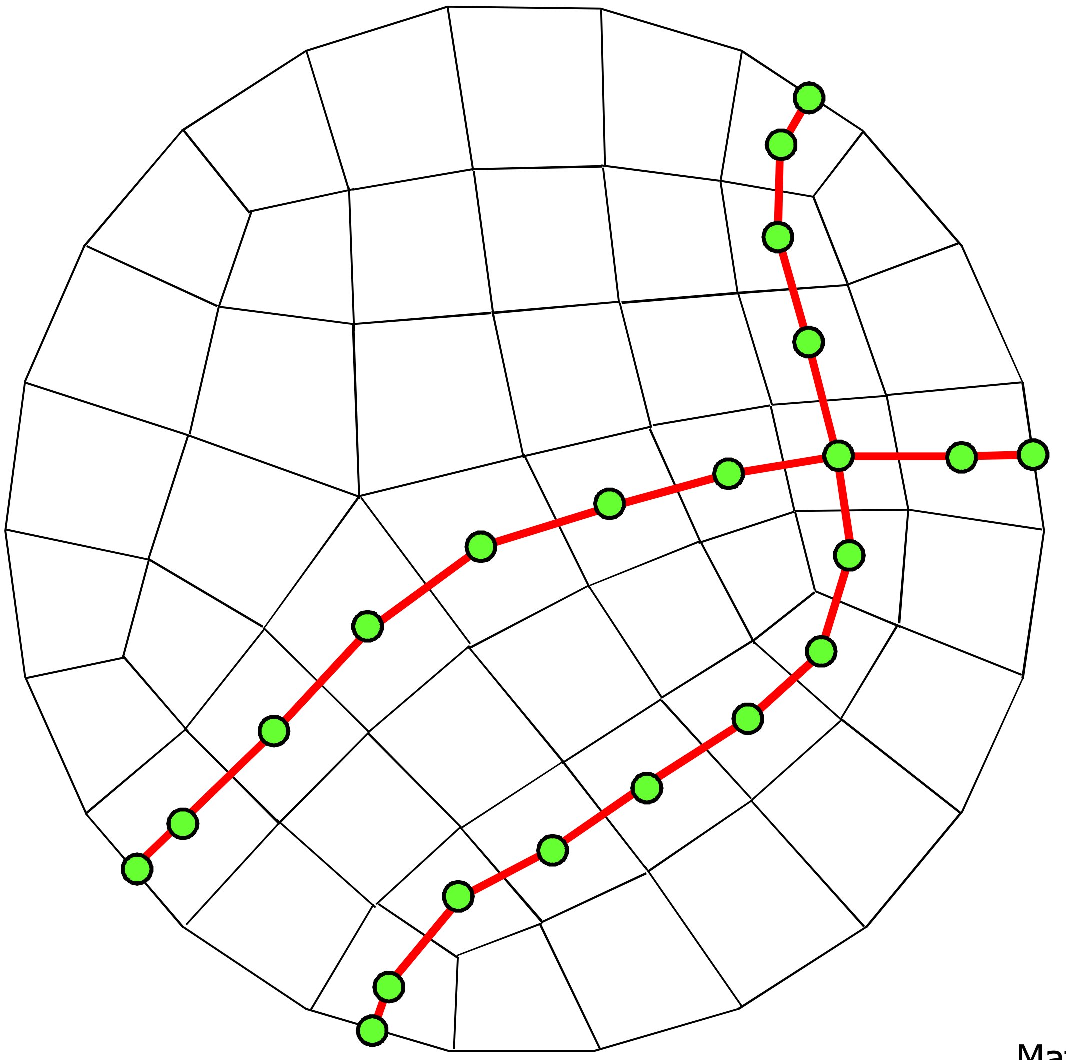


Matt Staten

Quadrilateral Dual Representation

Each dual edge is part of exactly one dual chord.

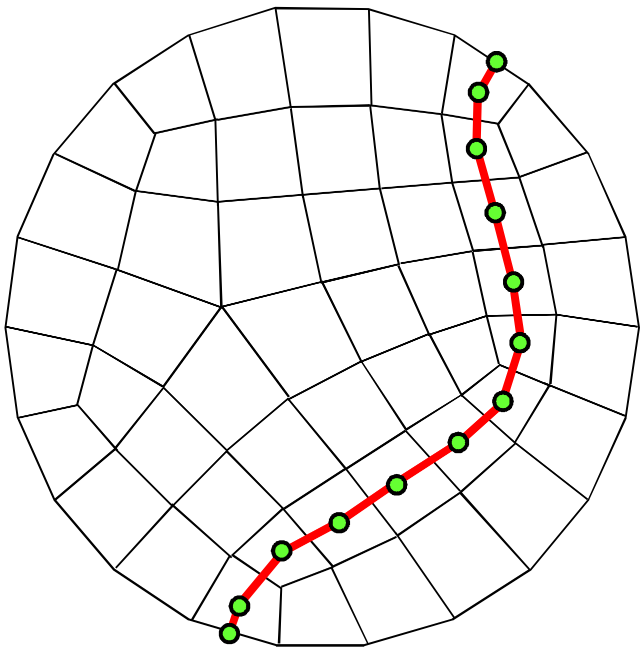
The vertex at the centroid of a quad is the intersection of 2 dual chords.



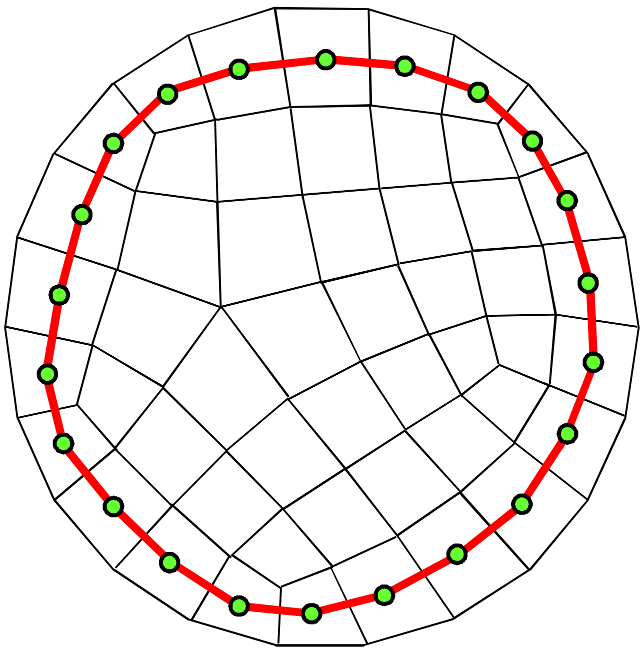
Matt Staten

Quadrilateral Dual Representation

Dual chords must be either circular, or connect two boundaries.



Connects two boundaries

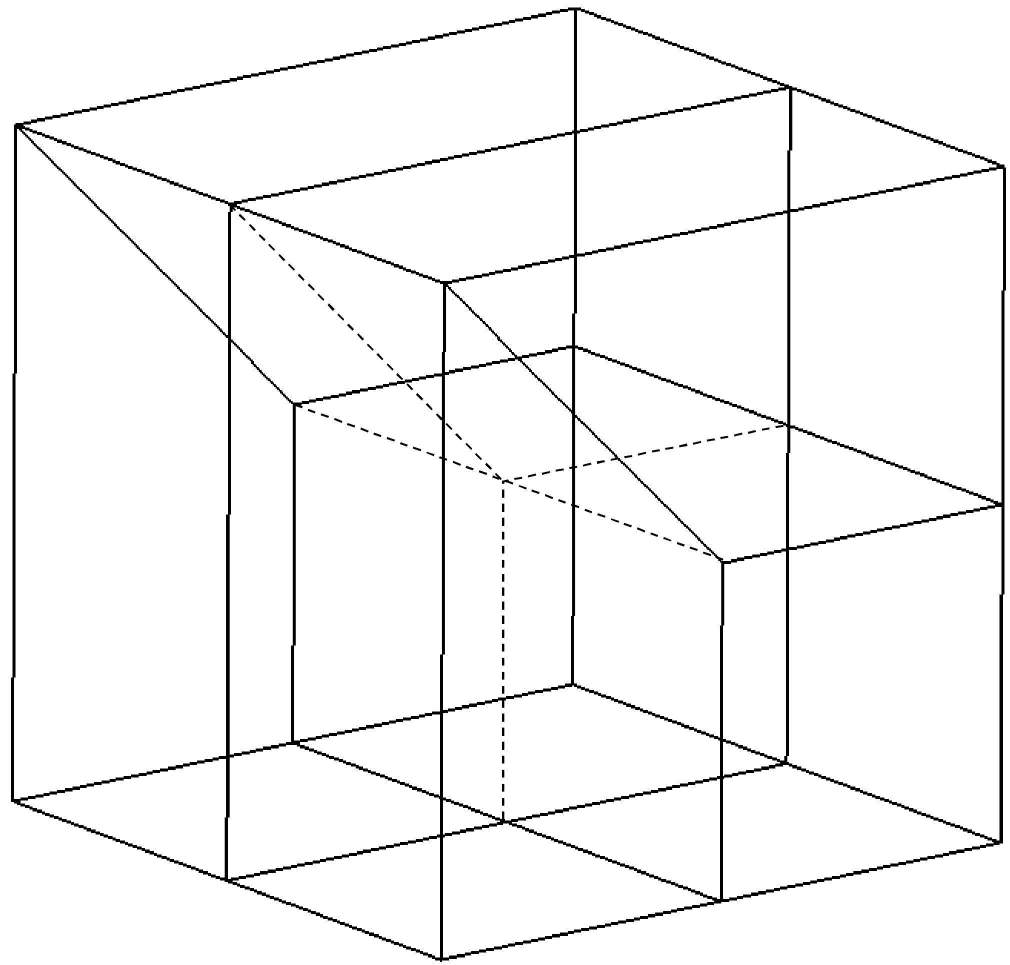


Circular

Hexahedral Dual Representation

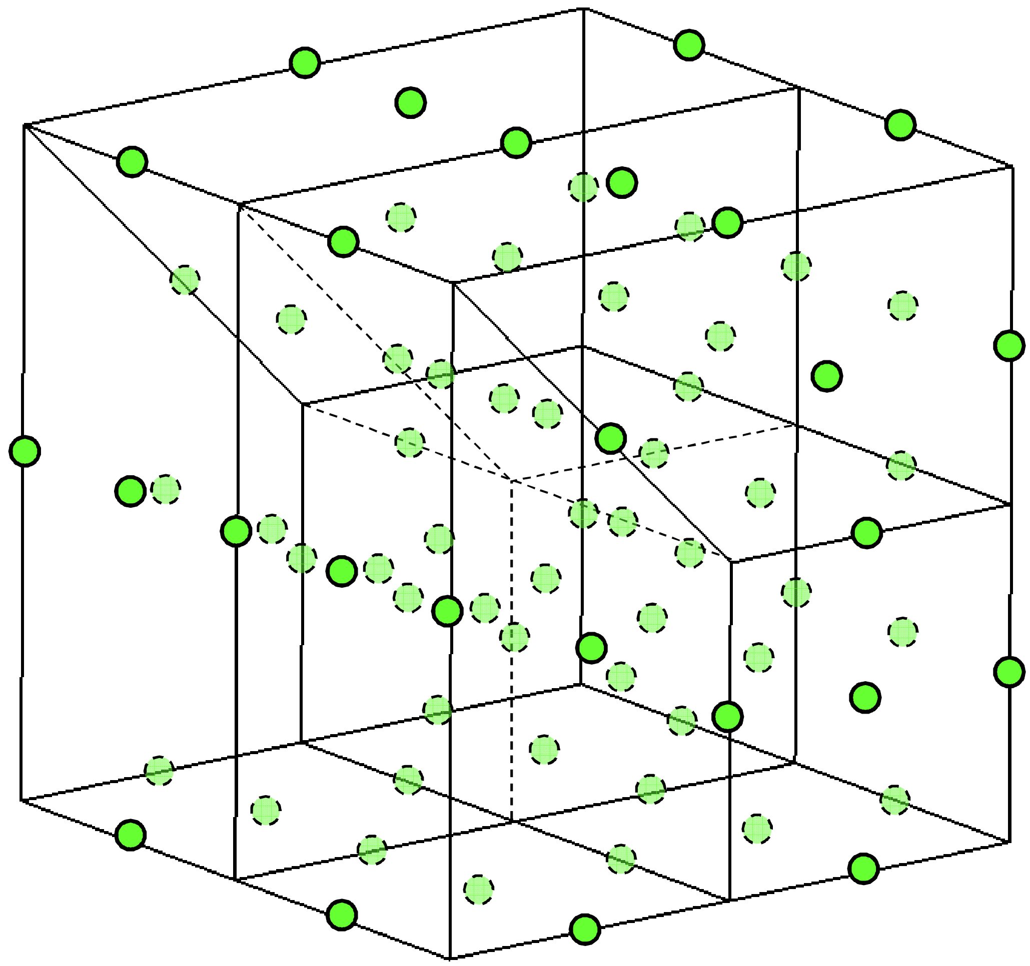
The elemental representation of a hexahedral mesh, composed of hexahedra, faces, edges, and nodes, is known as the *primal*.

Hexahedral meshes also have a *dual* representation, similar to the voroni skeleton of a triangular delaunay mesh.



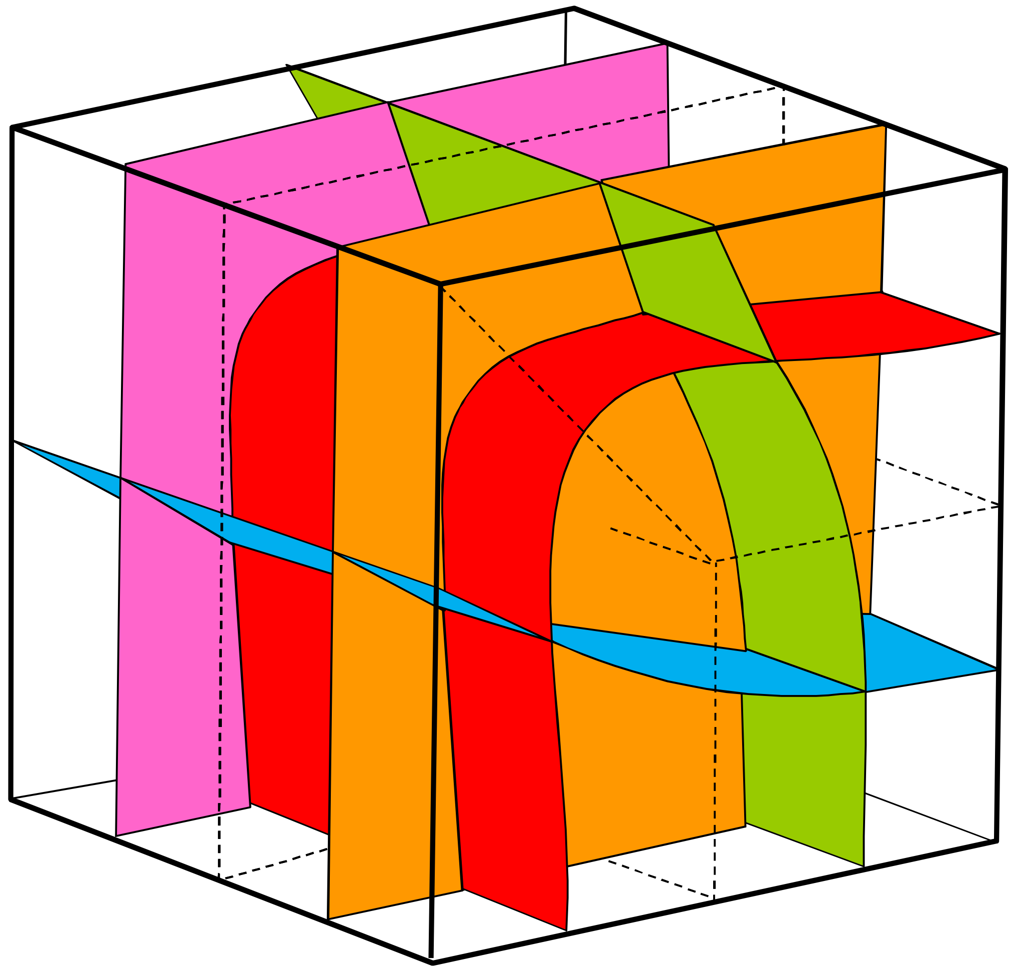
Hexahedral Dual Representation

A dual vertex, v_i, is defined at the centroid of each hexahedral element, boundary quad face, and boundary edge.



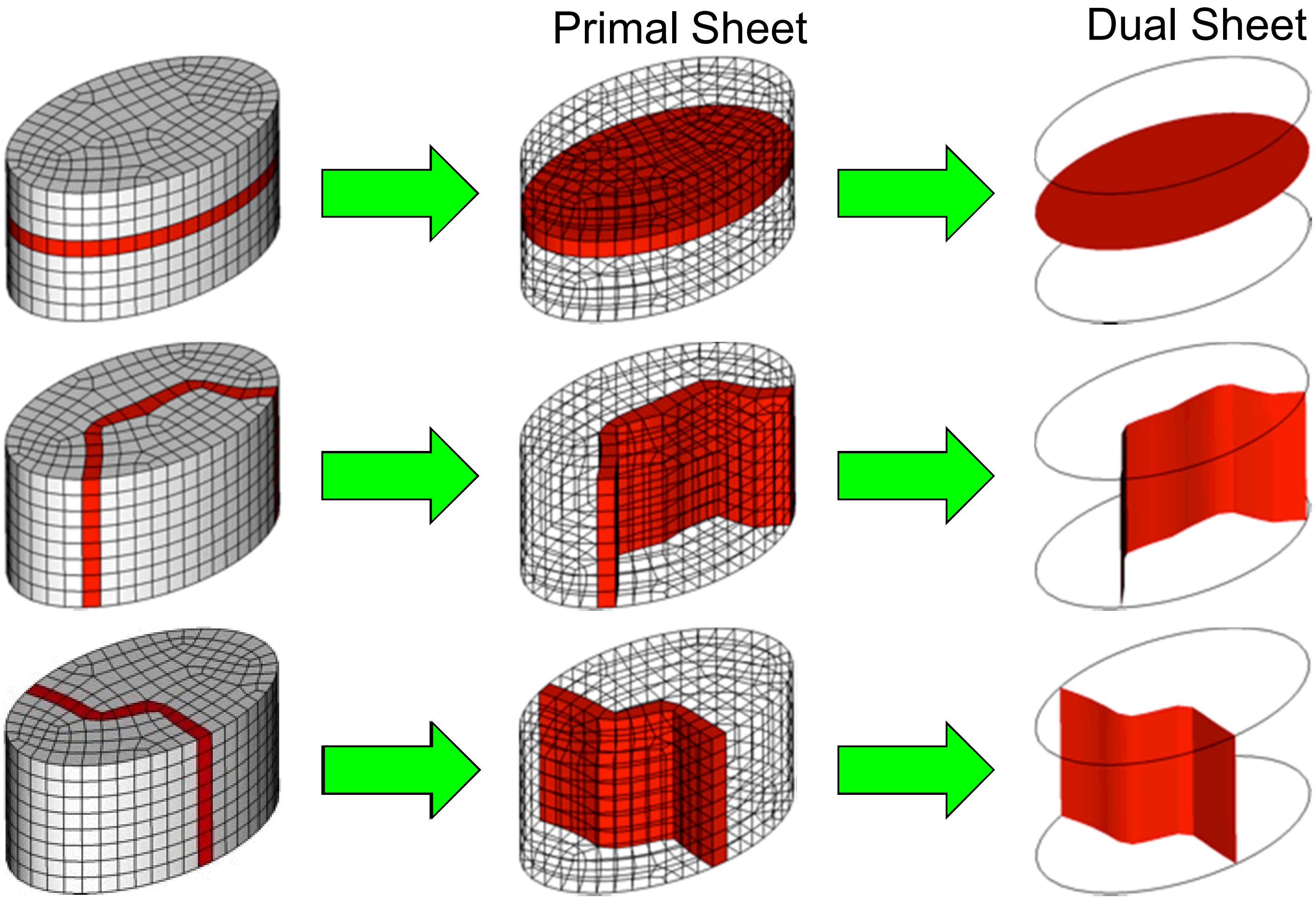
Hexahedral Dual Representation

Connecting the dual vertices through adjacent elements creates the edges and faces of the dual.

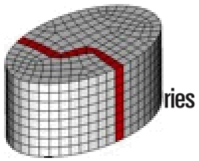


Quad/Hex Mesh Dual

Hexahedral meshes have an inherent layer structure. Each layer forms a sheet, having both a primal and a dual representation.



Hexahedral Dual Representation

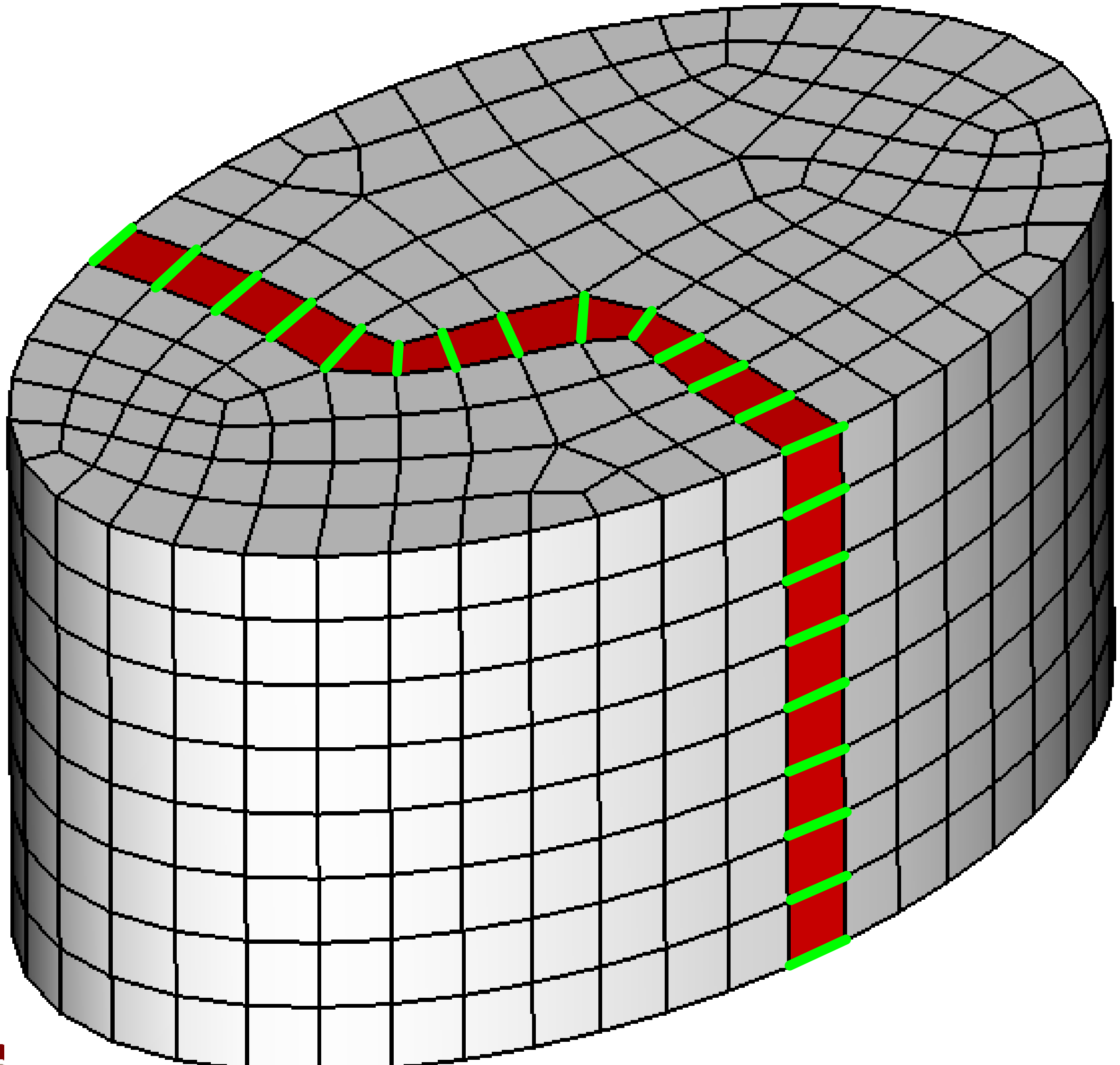


Another way to define a sheet is through edge traversal.

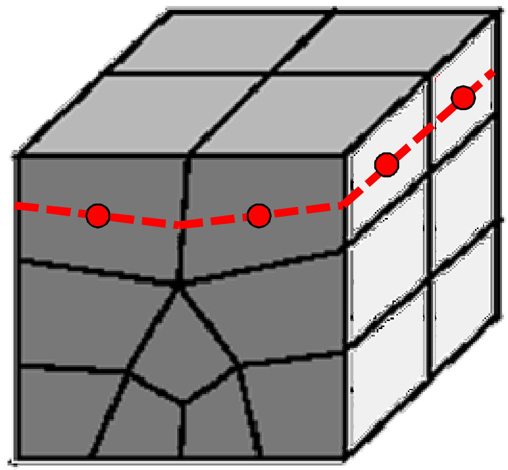
Create a group of edges by propagating from a single edge through adjacent hexahedra and through their topologically opposite edges.

Connecting midpoints of edges forms the dual sheet.

The hexahedra traversed forms the primal sheet.

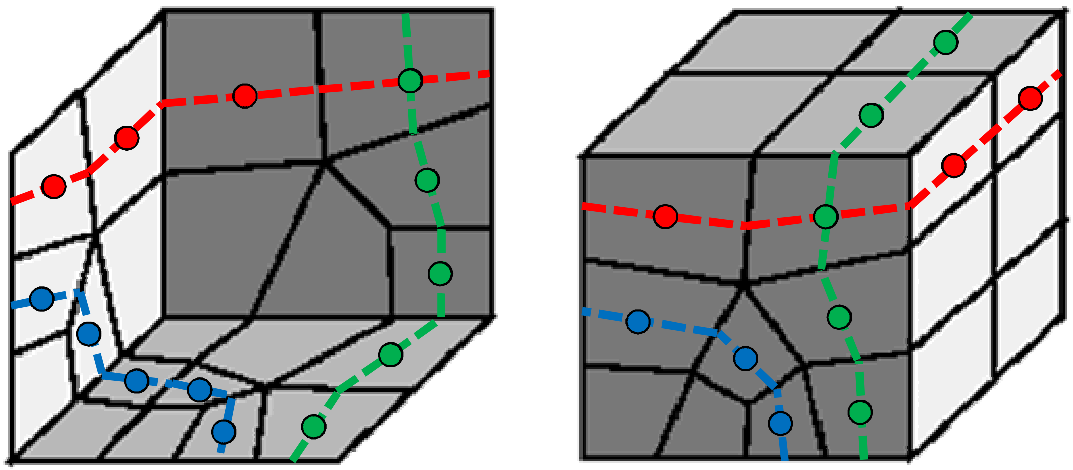


Whisker Weaving



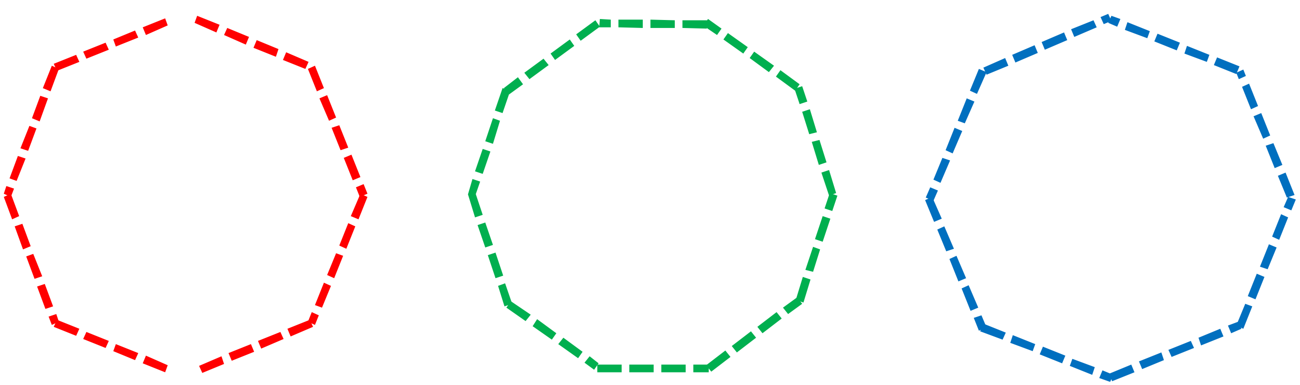
- Start with conformal quad mesh on surface.
- Must have even number of quads on boundary
- Complete chord loop can be defined starting from an arbitrary quad on the surface

Whisker Weaving



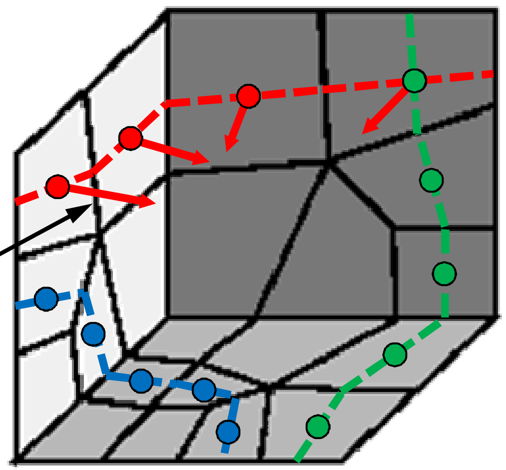
Each chord loop must bound a complete sheet (layer of hexes)

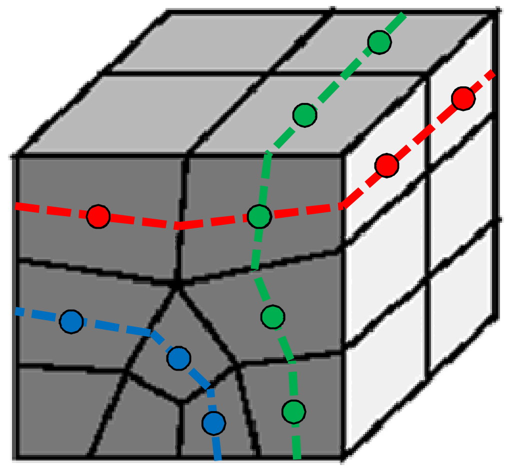
Sheet diagrams



chord loop 1 chord loop 2 chord loop 3

Whisker Weaving

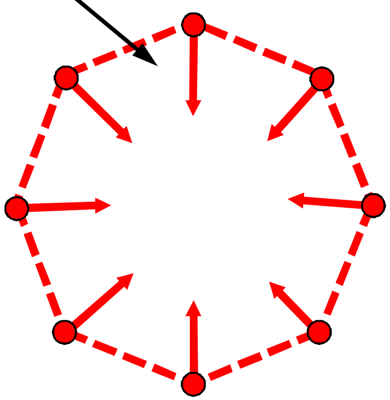




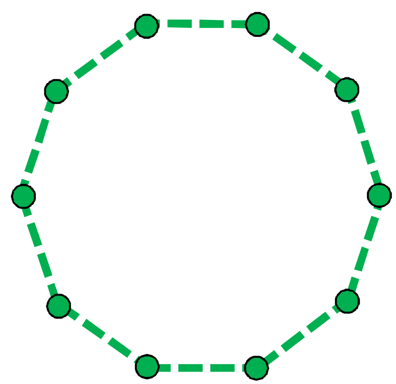
Chords (whiskers) advanced towards interior

Each chord loop must bound a complete sheet (layer of hexes)

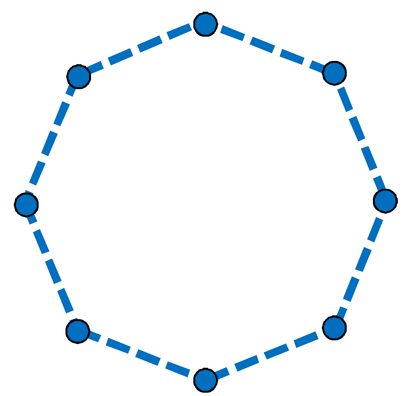
Sheet diagrams



sheet diagram 1



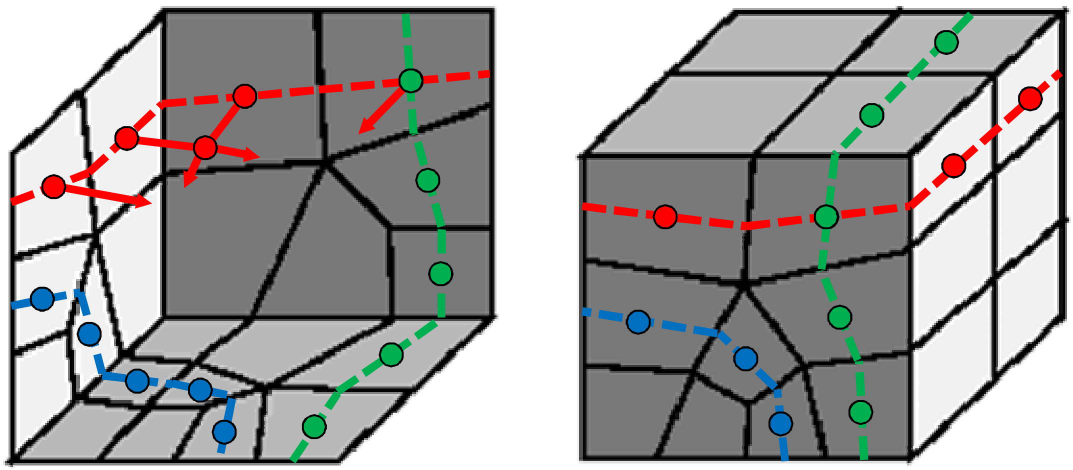
sheet diagram 2



sheet diagram 3

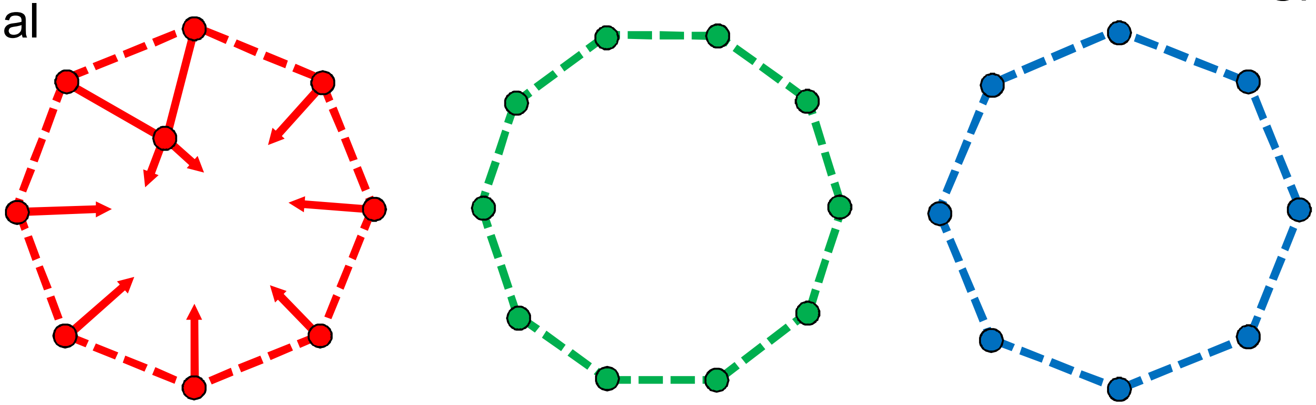
Whisker Weaving

Logic to intersect chords is used. Intersection of chords represents one hex in the primal



Each chord loop must bound a complete sheet (layer of hexes)

Sheet diagrams

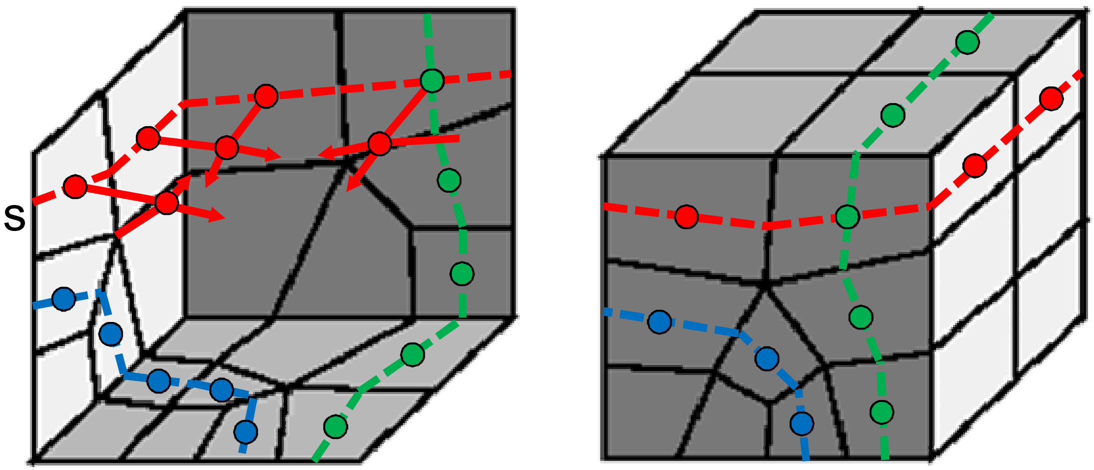


chord loop 1 chord loop 2 chord loop 3

Whisker Weaving

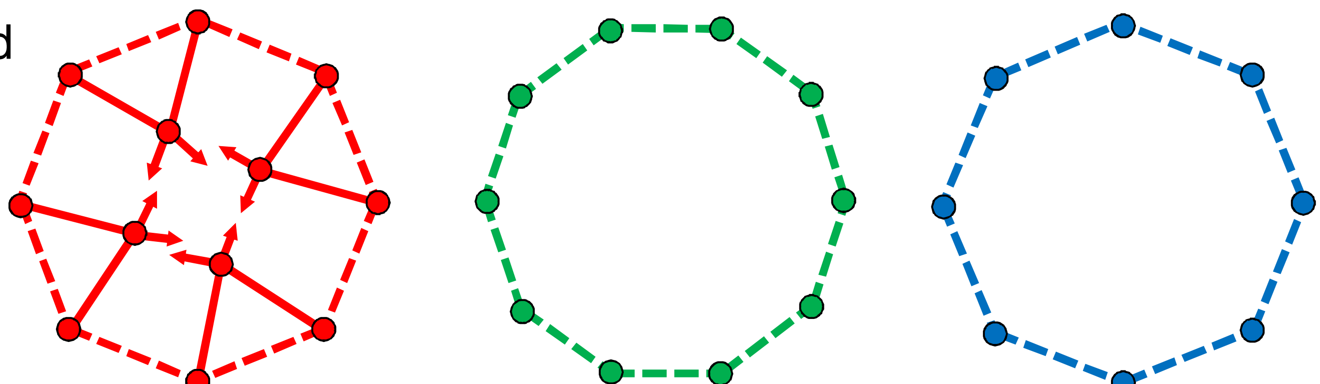
Exactly four chords per intersection is required

Dangling chords (whiskers) must be resolved



Each chord loop must bound a complete sheet (layer of hexes)

Sheet diagrams

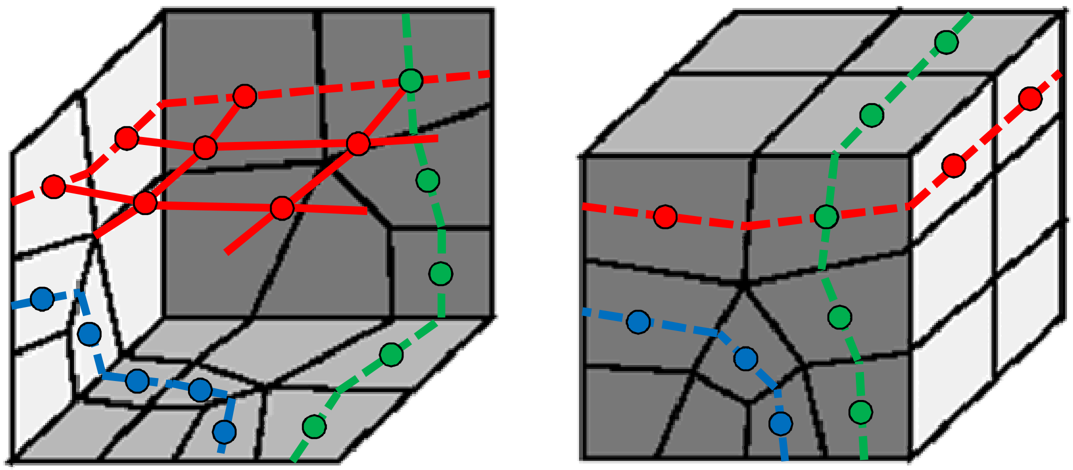


sheet diagram 1 | sheet diagram 2 | sheet diagram 3

Whisker Weaving

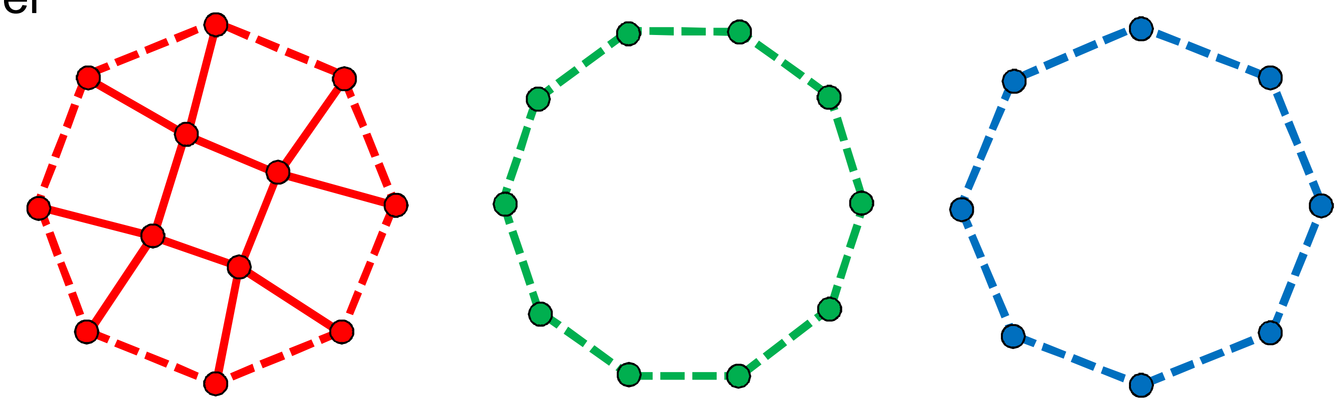
Sandia National Laboratories

Completed sheet diagram represents one completed hex layer



Each chord loop must bound a complete sheet (layer of hexes)

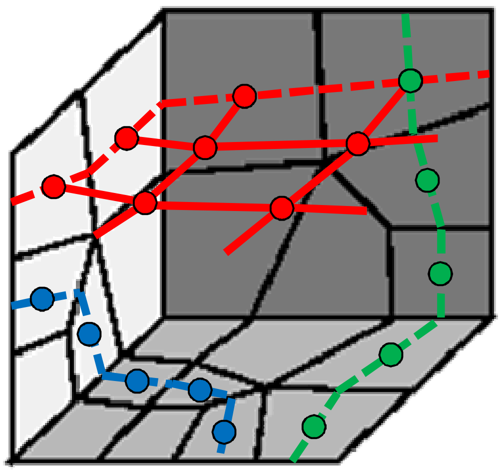
Sheet diagrams

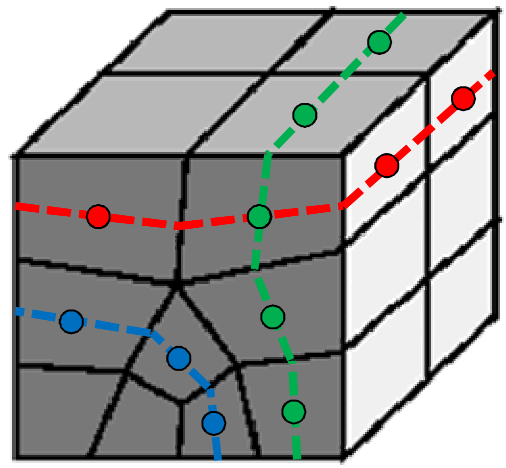


sheet diagram 1 sheet diagram 2 sheet diagram 3

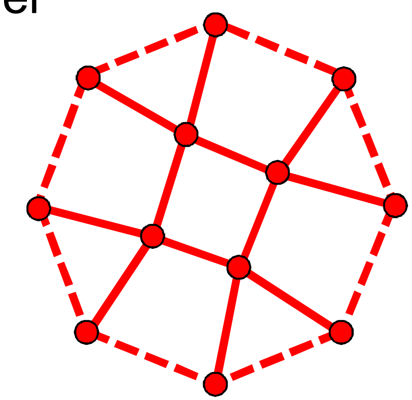
Whisker Weaving

Completed sheet diagram represents one completed hex layer

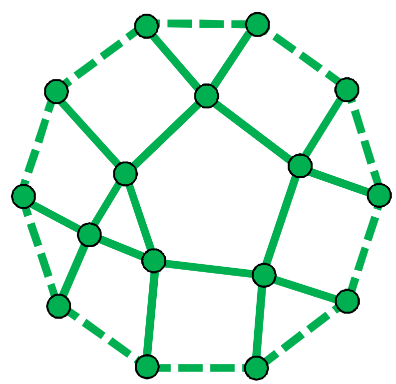




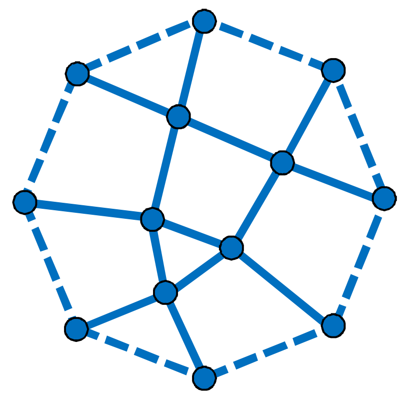
Sheet diagrams are completed for every chord loop defined in the surface mesh



sheet diagram 1



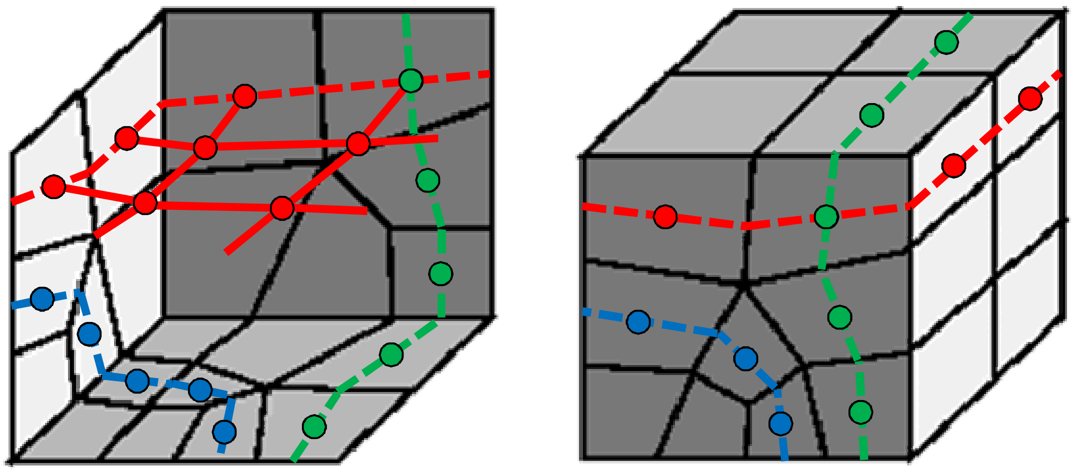
sheet diagram 2



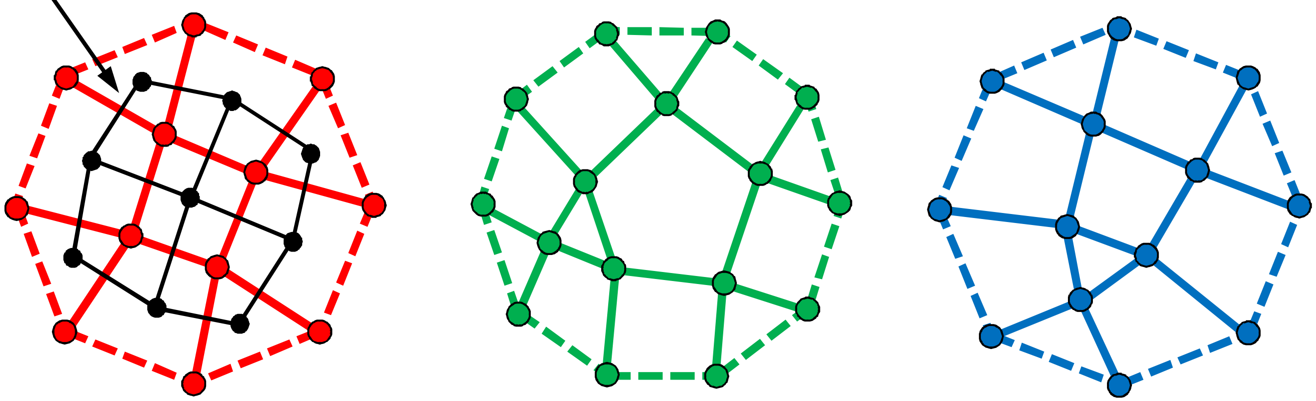
sheet diagram 3

Whisker Weaving

Primal hex topology of the layer can be extracted from the sheet diagram



Sheet diagrams are completed for every chord loop defined in the surface mesh



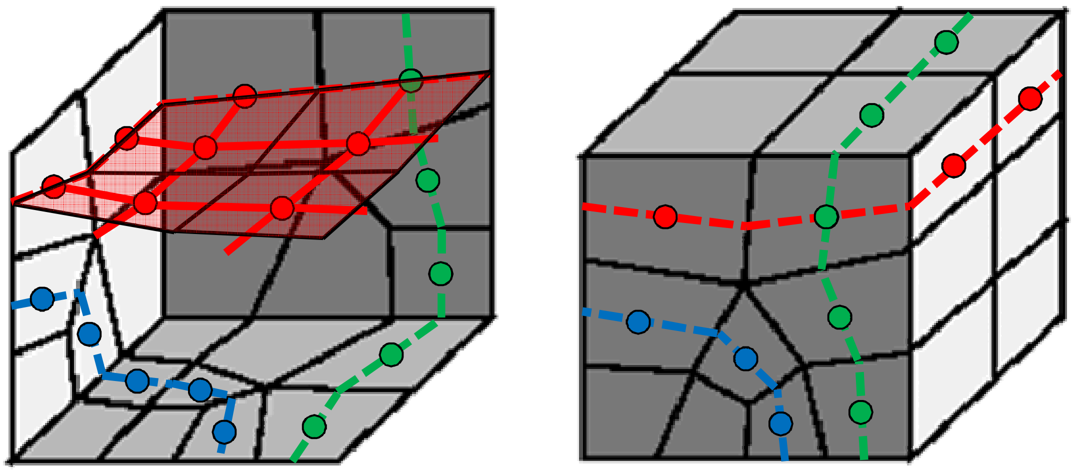
sheet diagram 1

sheet diagram 2

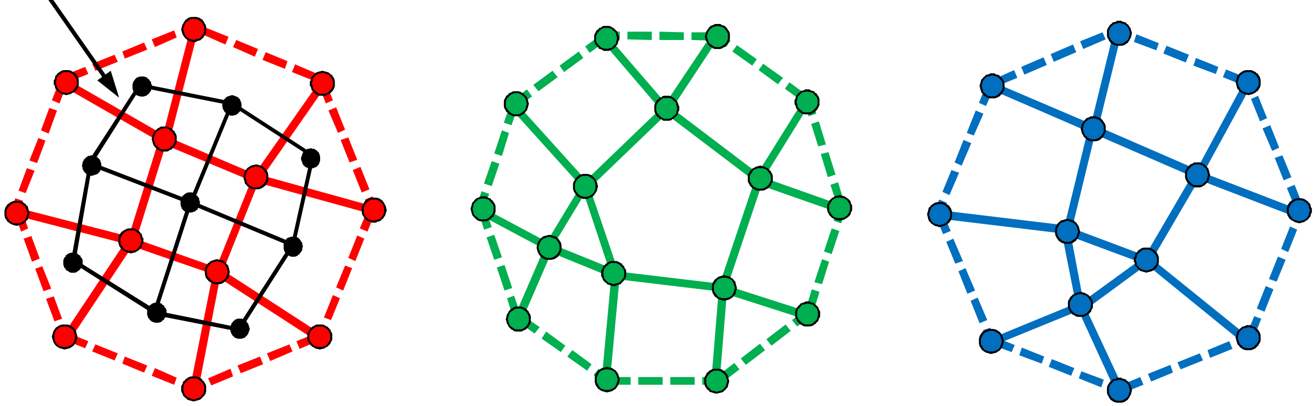
sheet diagram 3

Whisker Weaving

Primal hex topology of the layer can be extracted from the sheet diagram

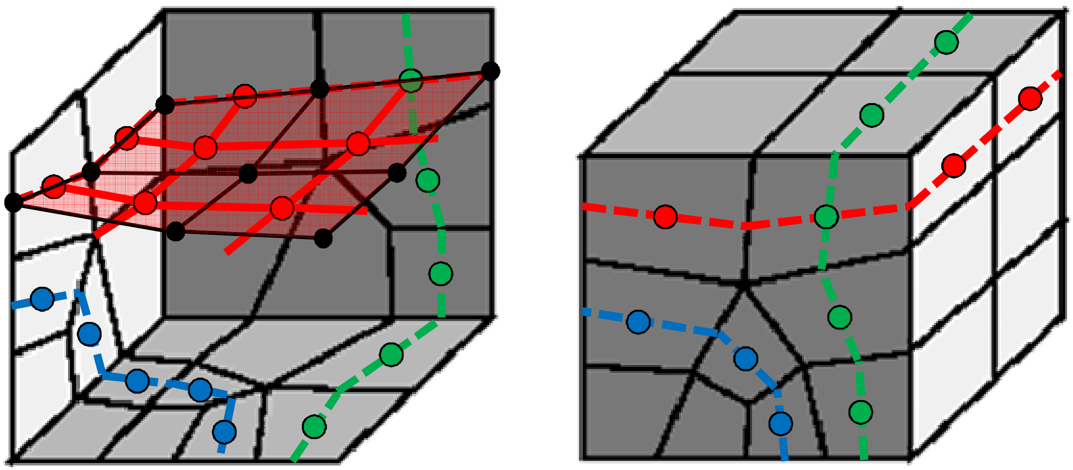


Sheet diagrams are completed for every chord loop defined in the surface mesh

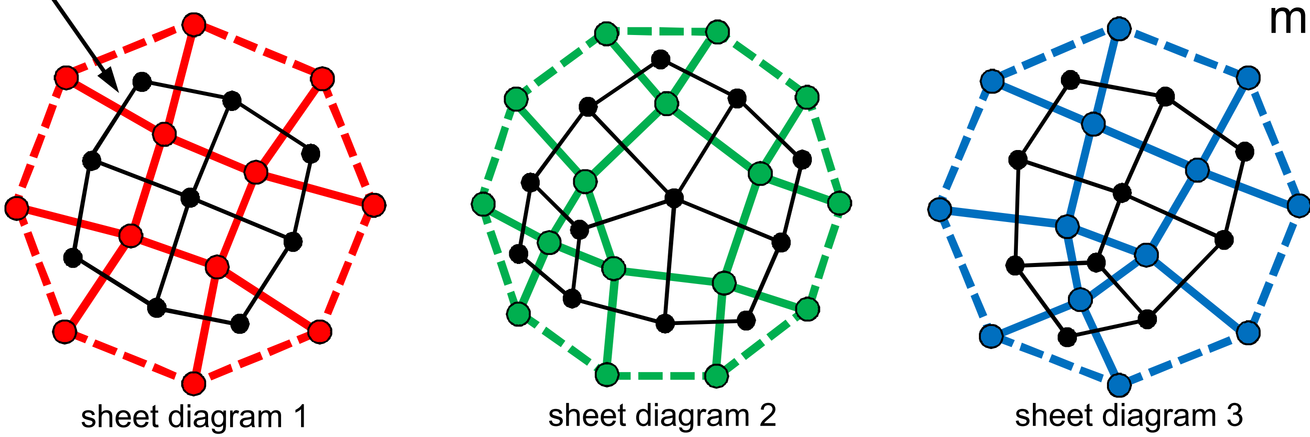


Whisker Weaving

Primal hex topology of the layer can be extracted from the sheet diagram

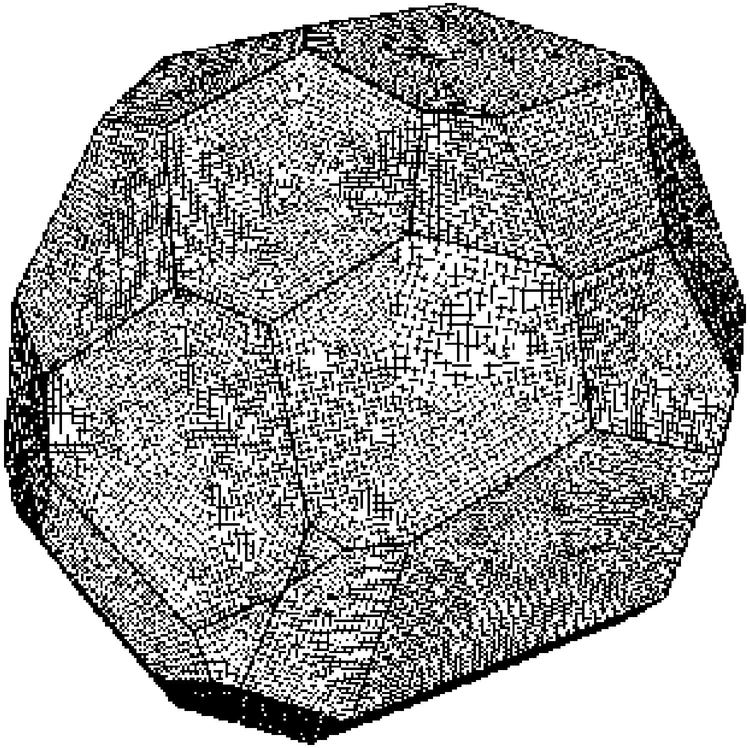


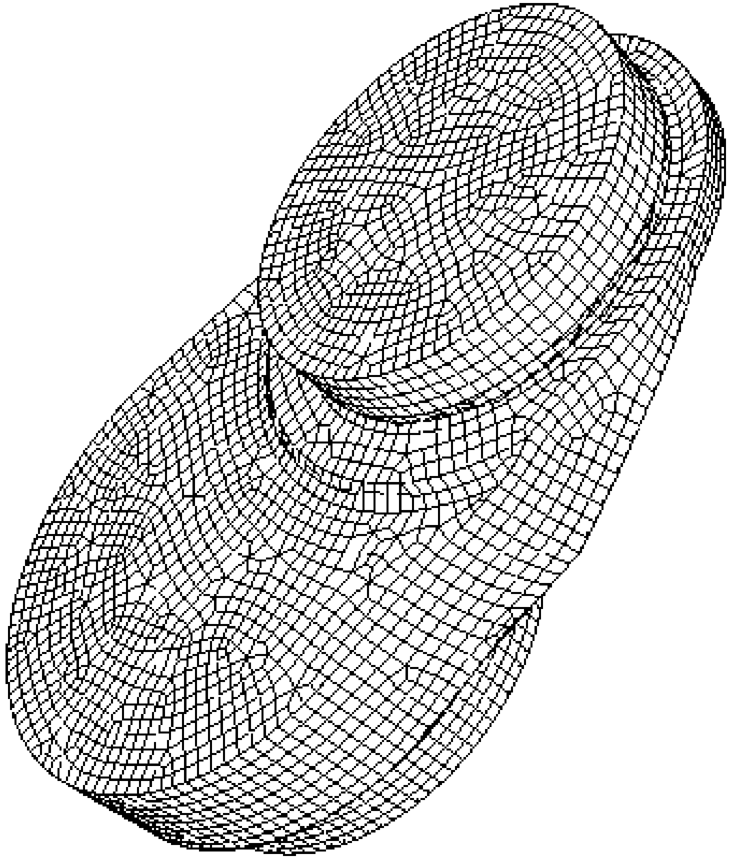
Extracting primal hex topology from each completed sheet diagram will construct the complete topology of the mesh



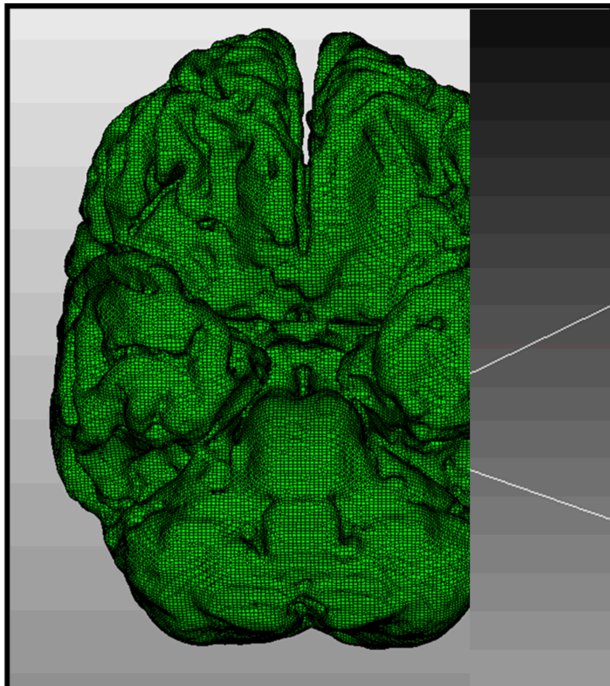
Whisker Weaving

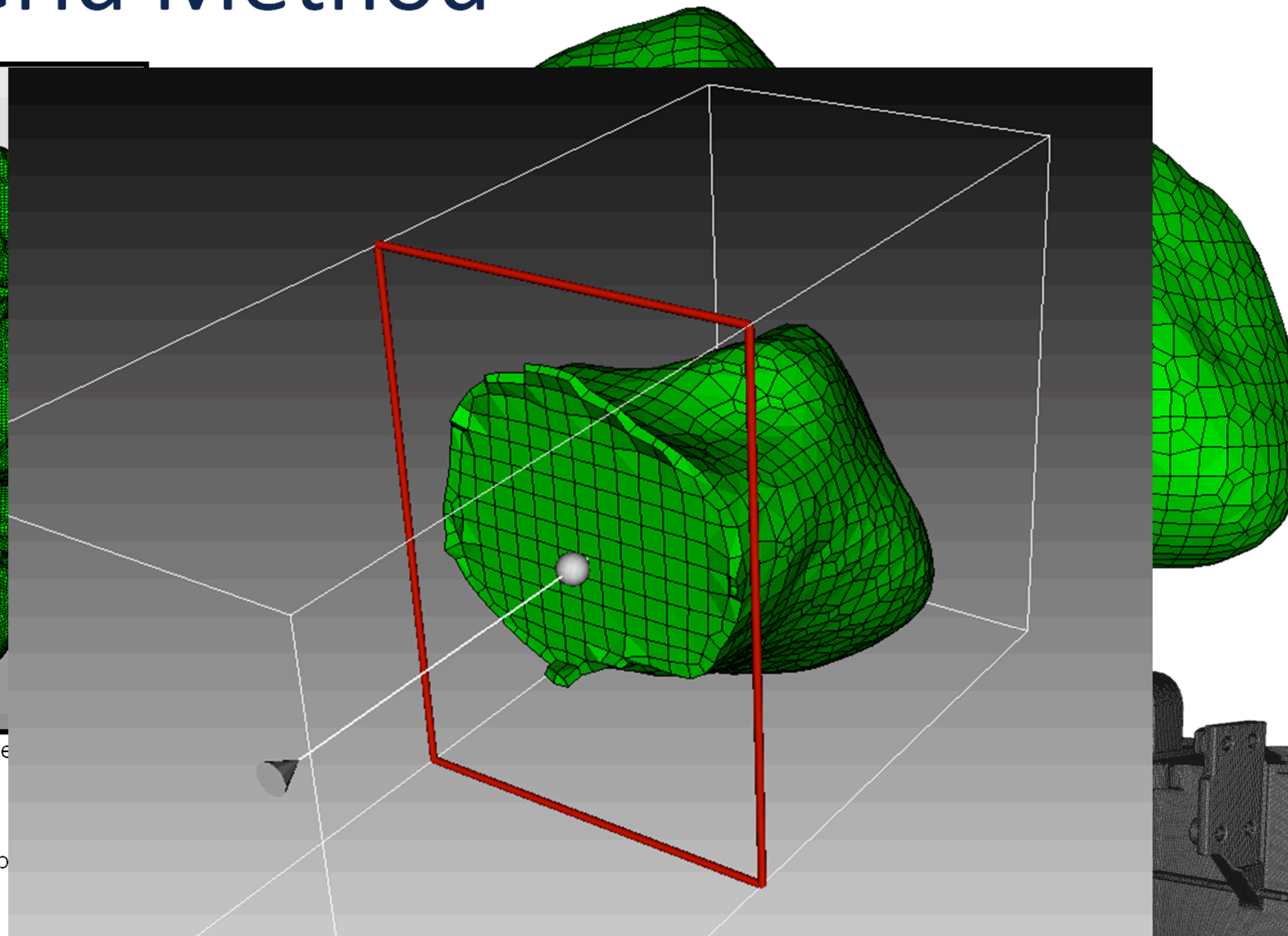
Examples of successful meshes using whisker weaving





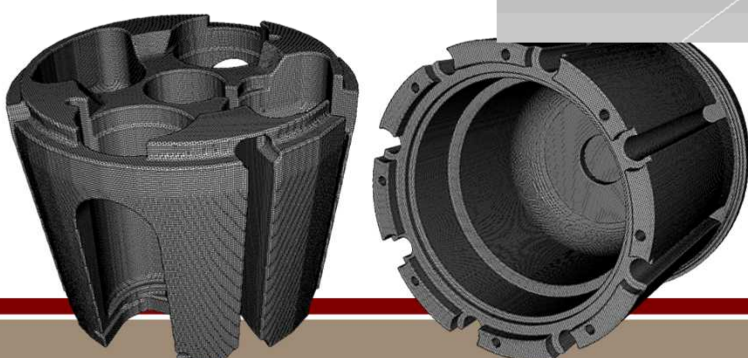
Overlay Grid Method

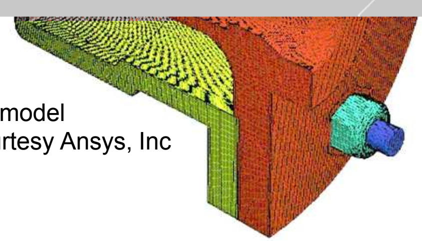




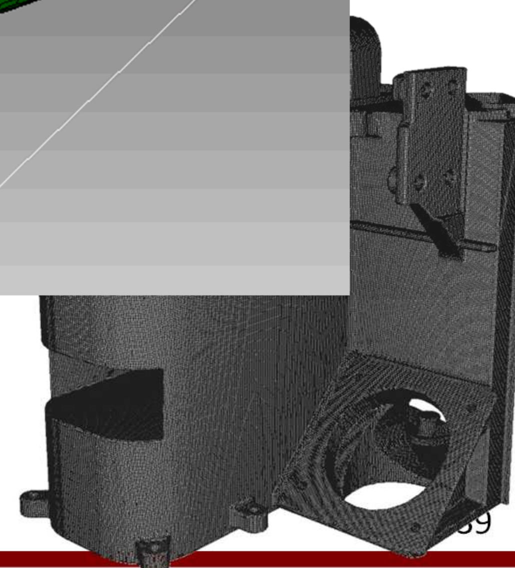
STL MRI Brain Model, Courtesy Bryce Owe
Brigham Young University, Provo, UT

Weapon Component Models Courtesy Step
US Army, Picitinni. Used with Permission

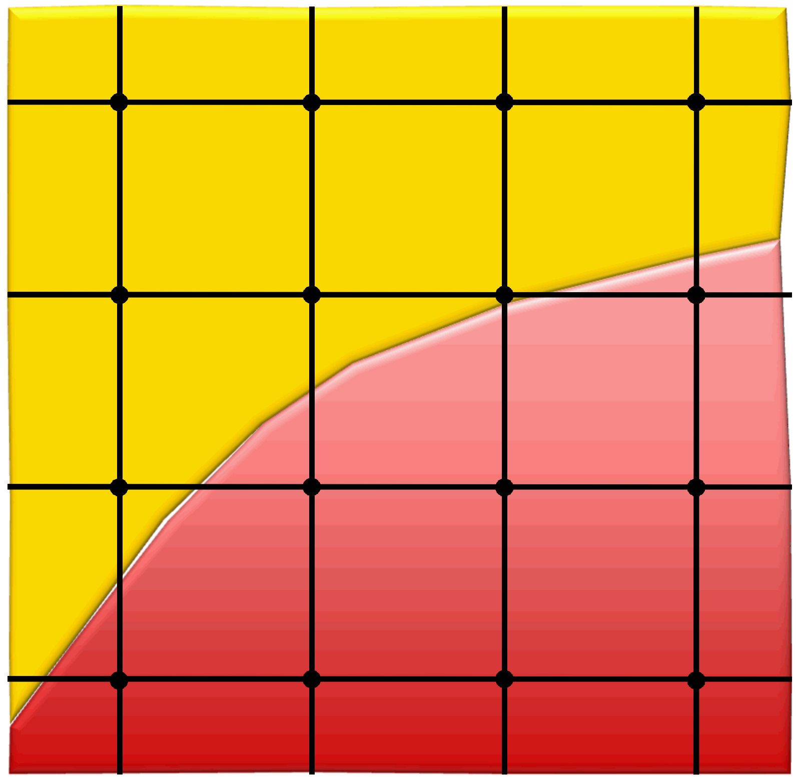




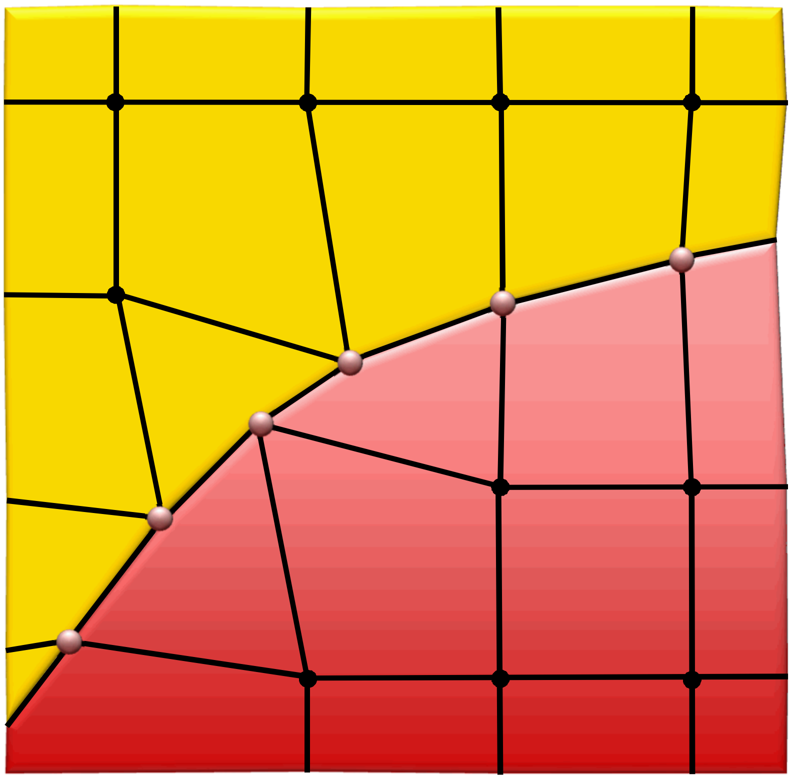
V2 model
courtesy Ansys, Inc



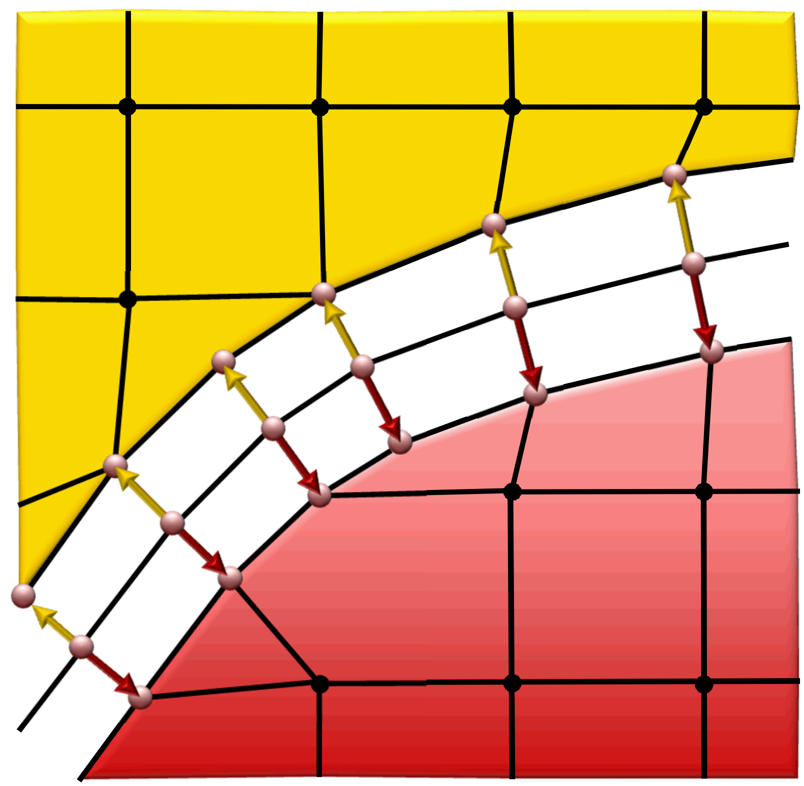
Overlay Grid Method



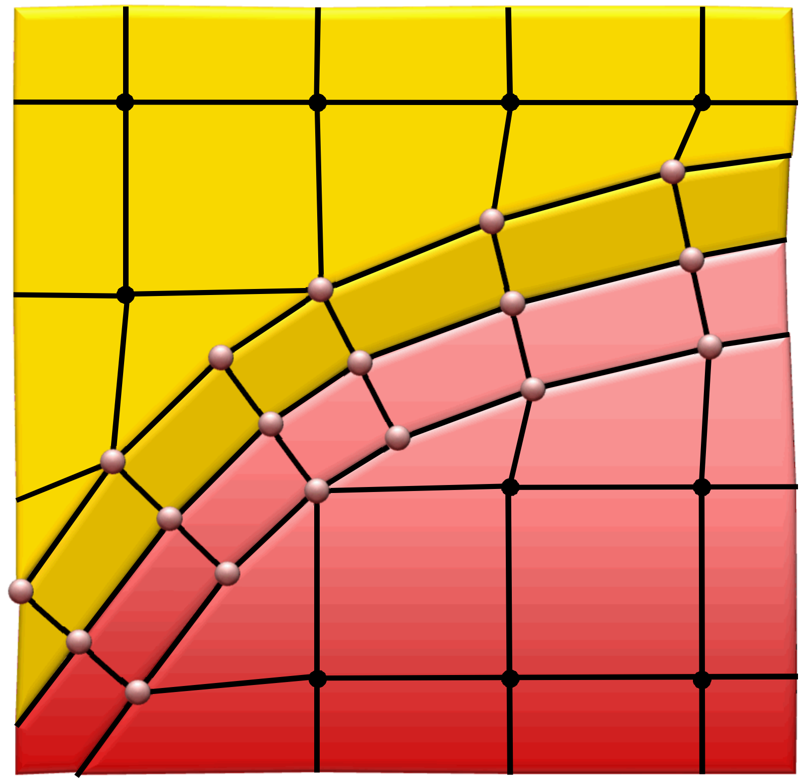
Overlay Grid Method



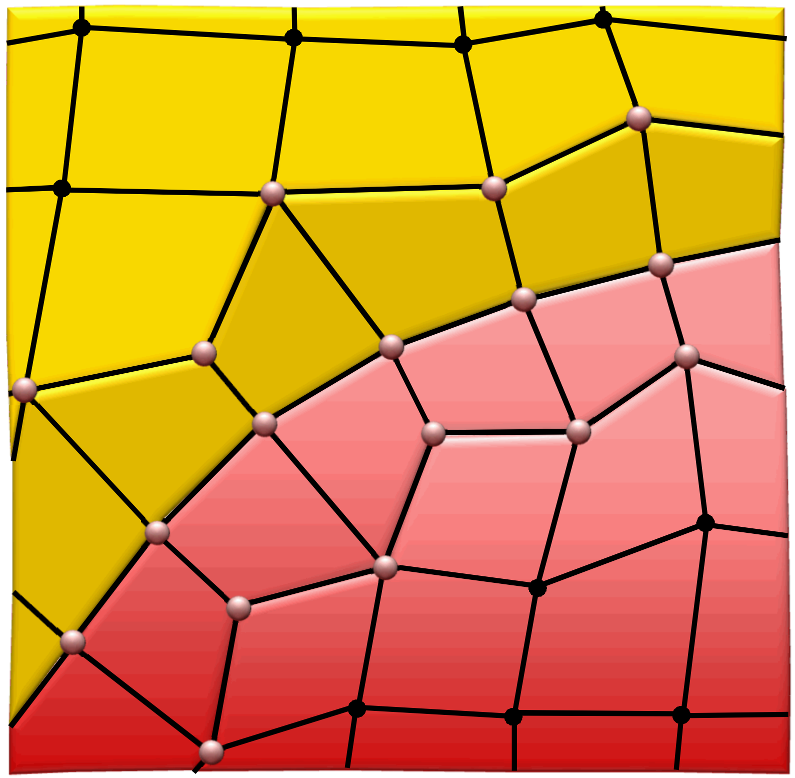
Overlay Grid Method



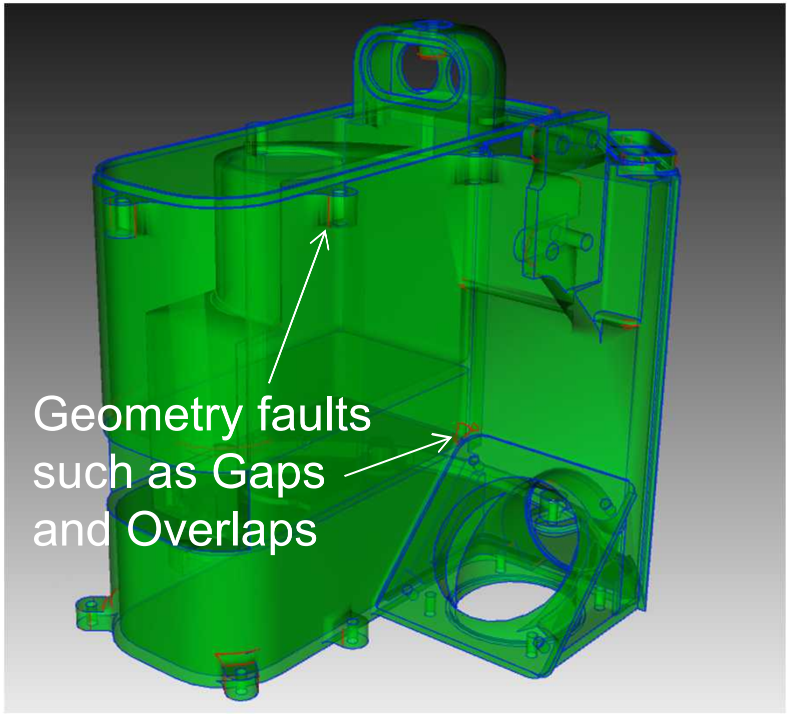
Overlay Grid Method



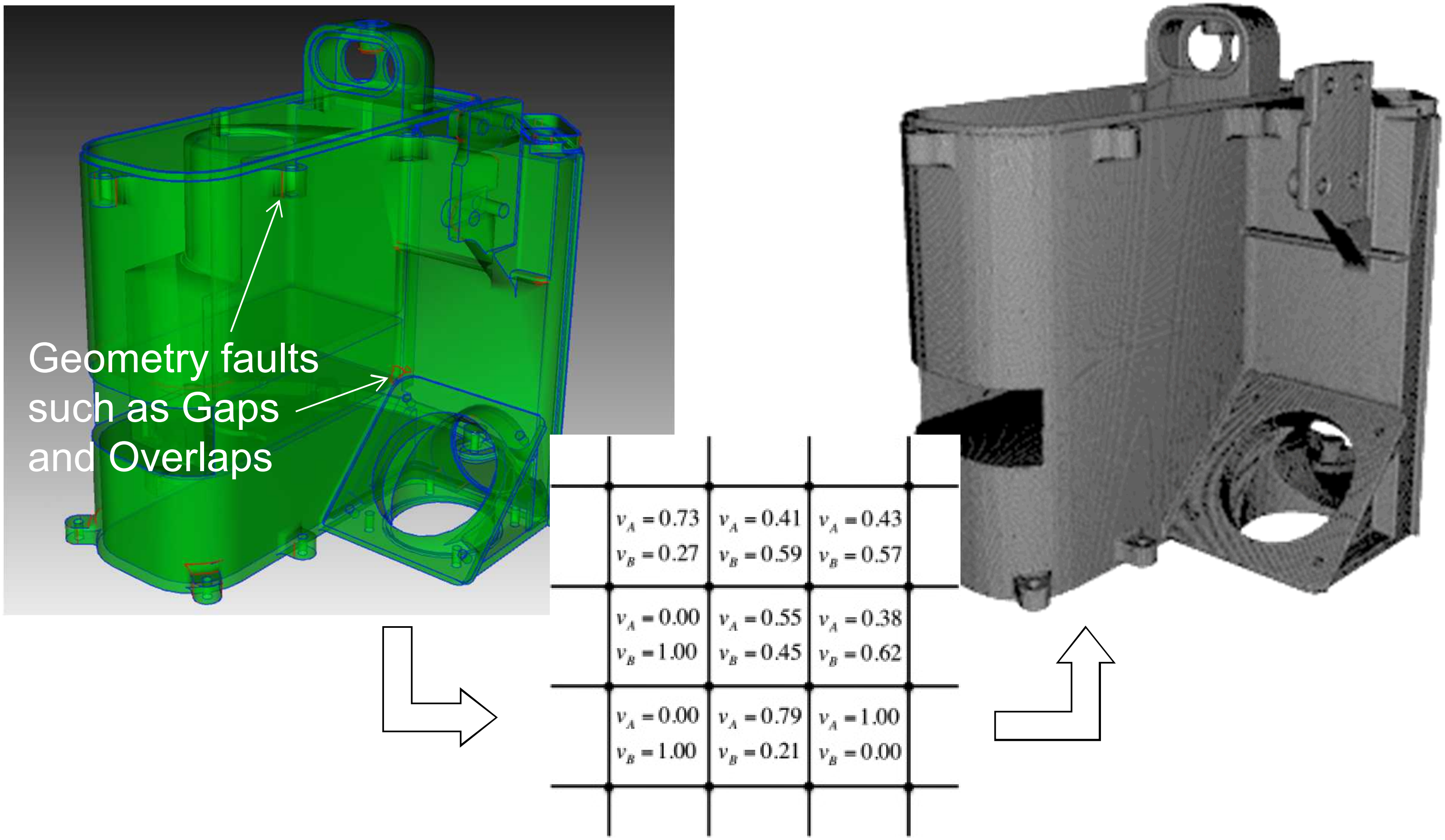
Overlay Grid Method



Overlay Grid Method

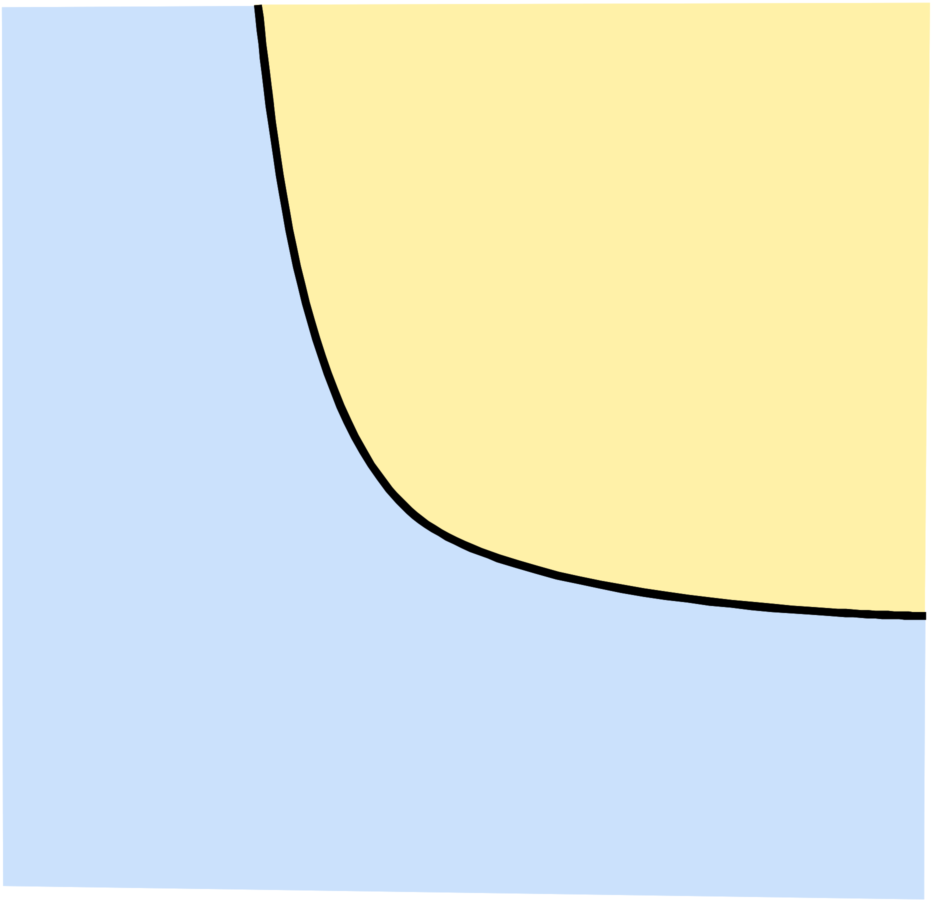


Overlay Grid Method

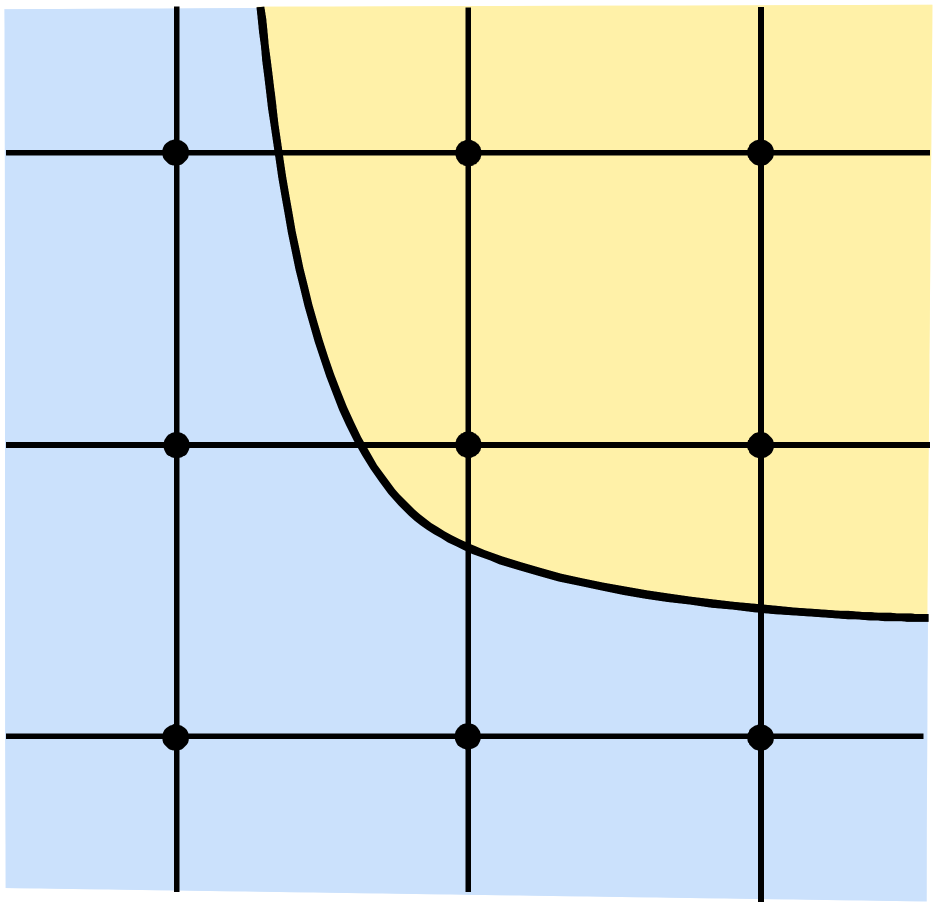


Geometry is first converted to “Volume Fraction” Data before meshing.

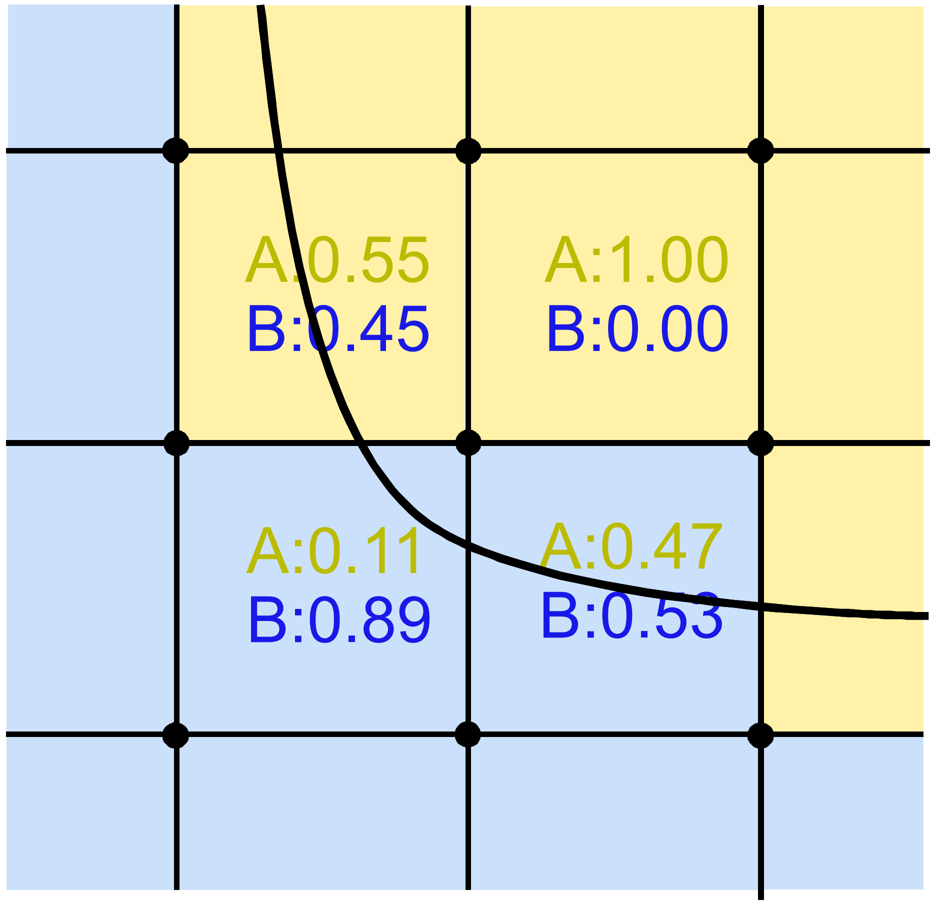
Interface Approximation



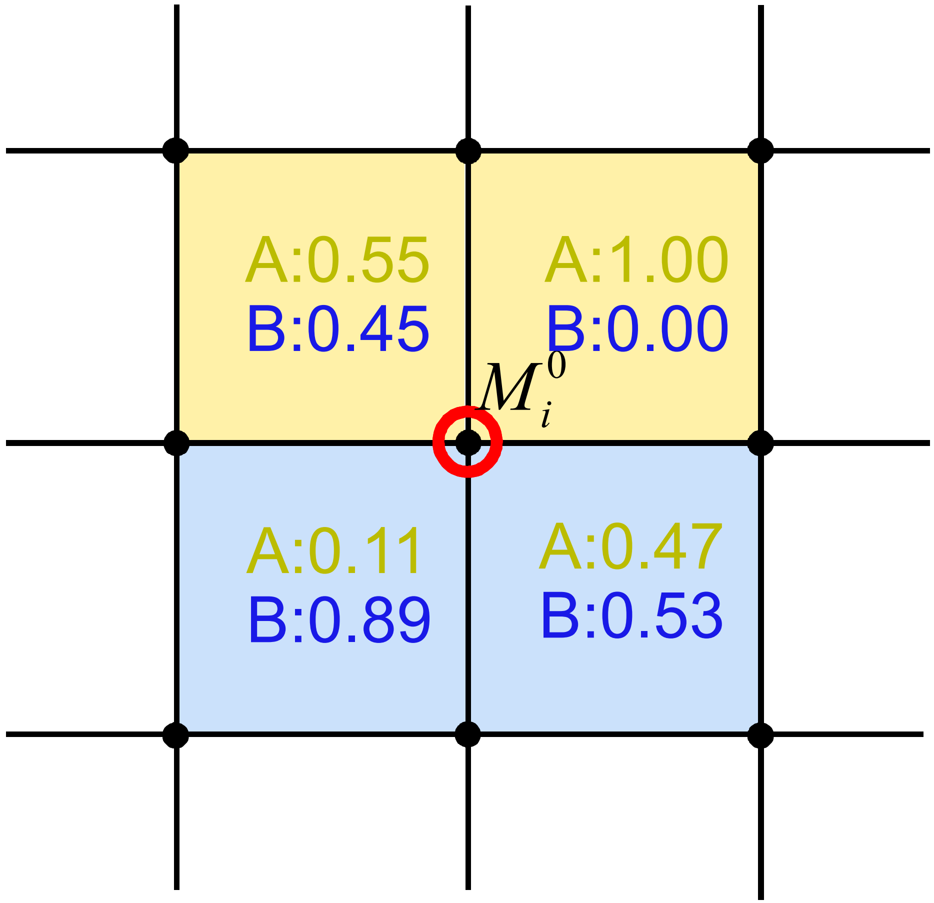
Interface Approximation



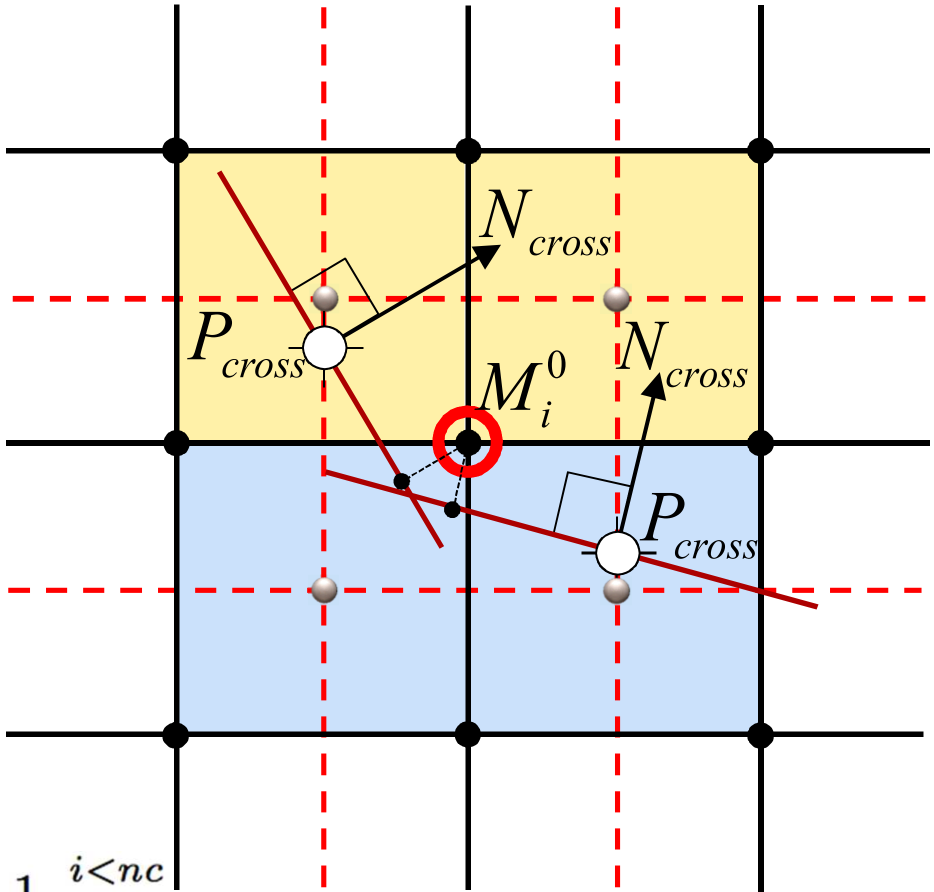
Interface Approximation



Interface Approximation



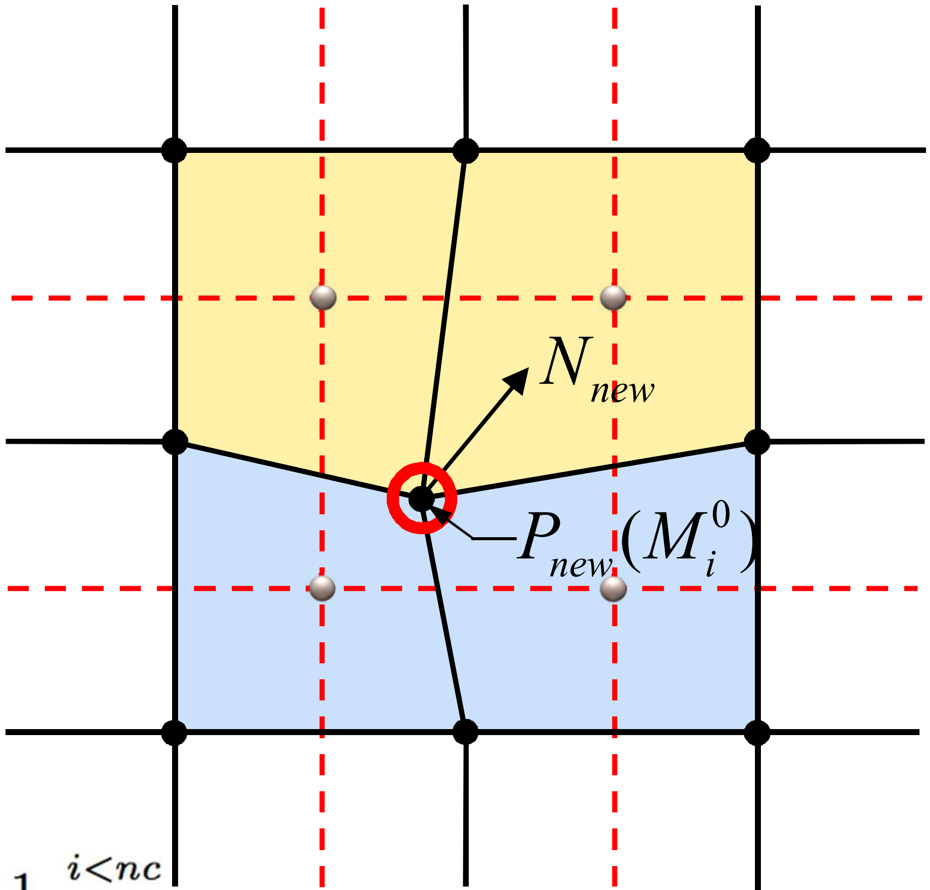
Interface Approximation



$$P_{new} = \frac{1}{nc} \sum_{i=0}^{i<nc} P_0 - (N_{cross})_i \cdot (P_0 - (P_{cross})_i) \times (N_{cross})_i$$

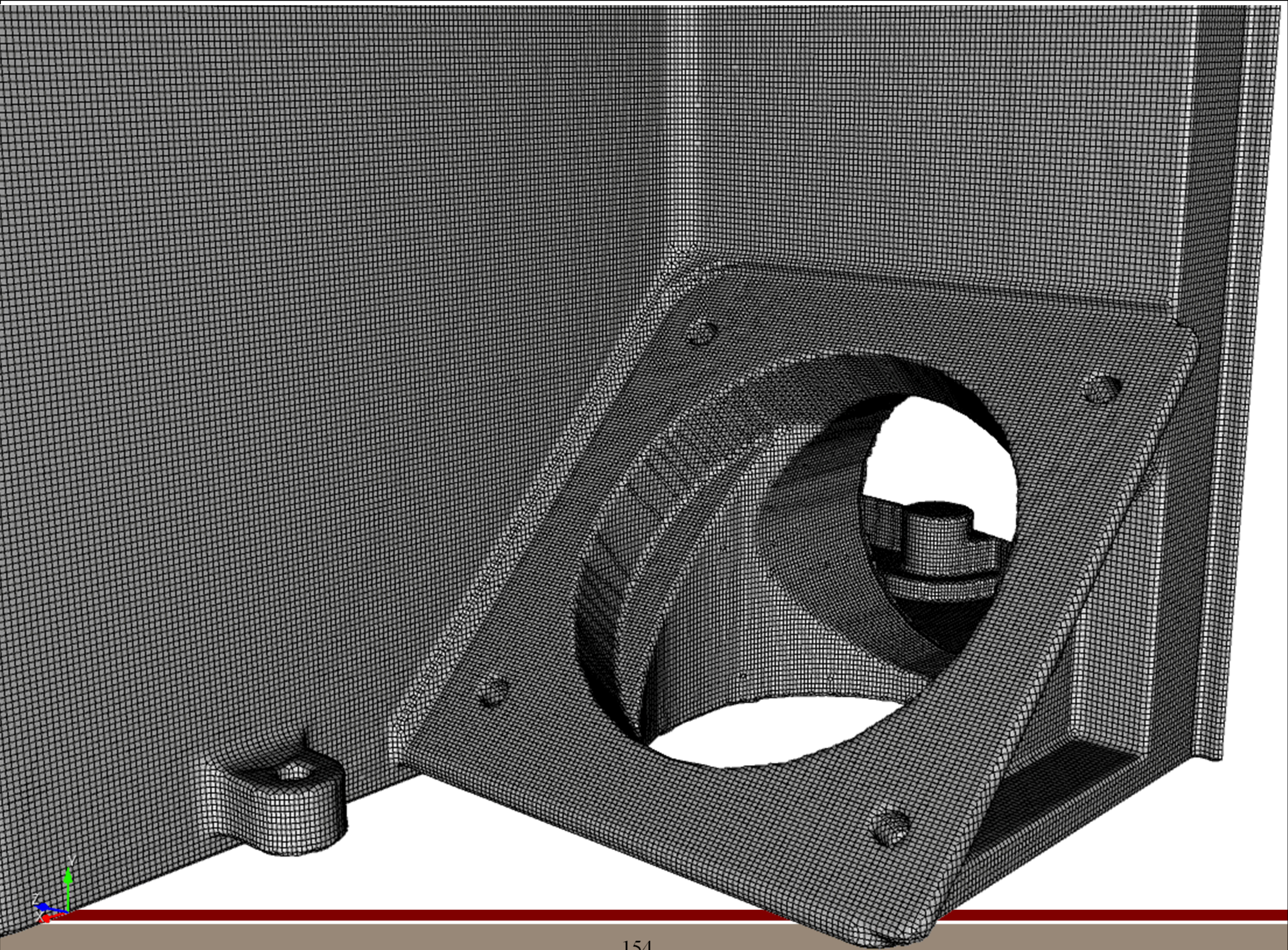
$$(N_{new})_n = \left| \sum_{i=0}^{i<nc_n} (N_{cross})_i \right|$$

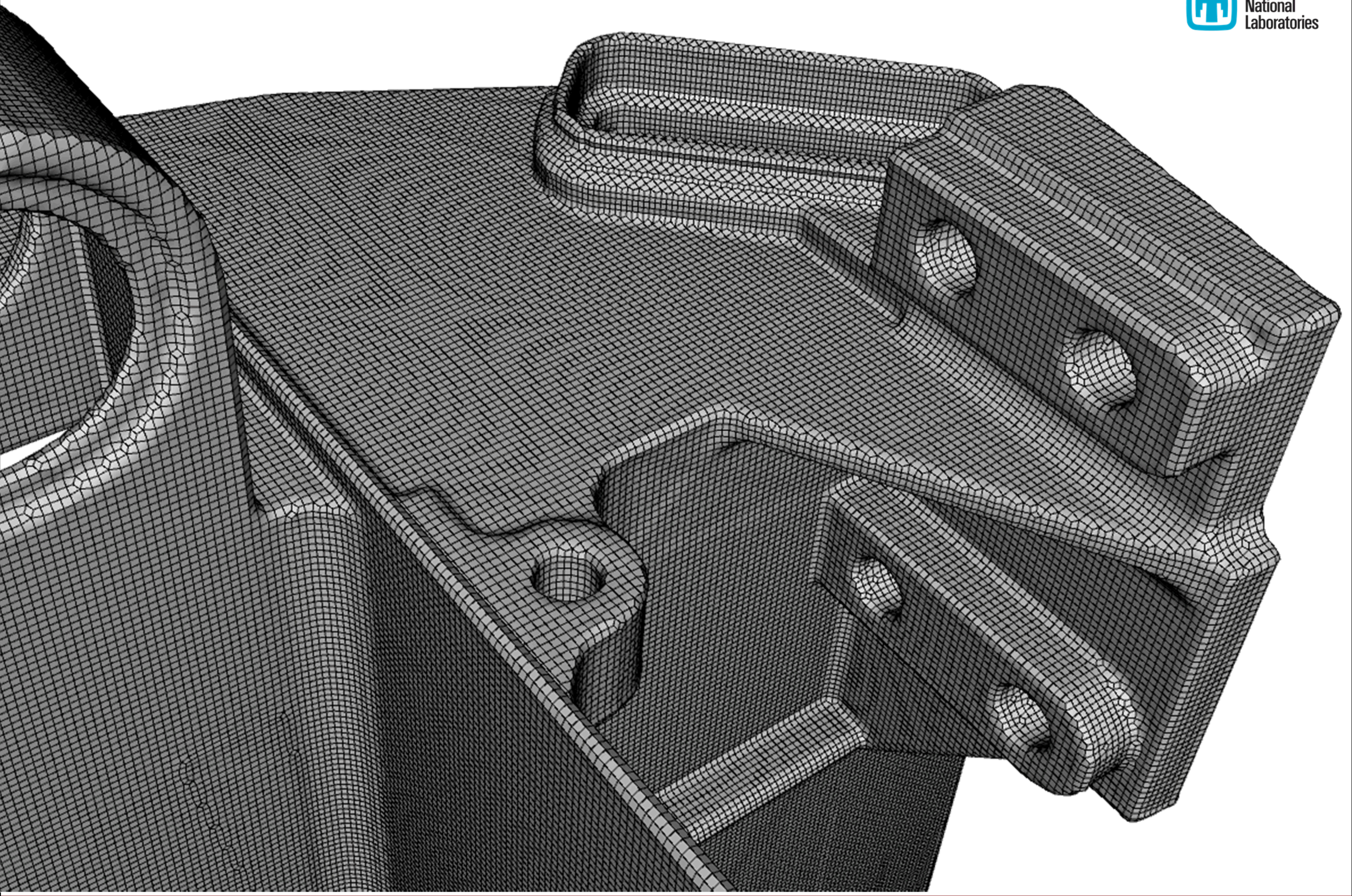
Interface Approximation



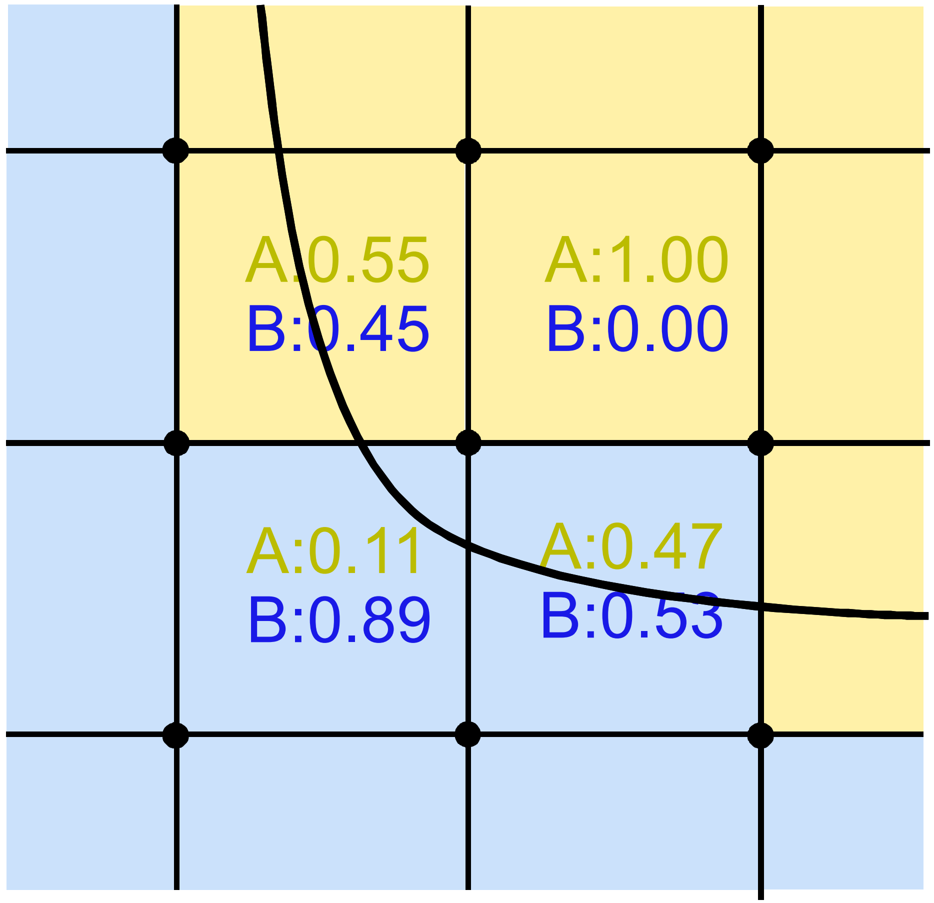
$$P_{new} = \frac{1}{nc} \sum_{i=0}^{i<nc} P_0 - (N_{cross})_i \cdot (P_0 - (P_{cross})_i) \times (N_{cross})_i$$

$$(N_{new})_n = \left| \sum_{i=0}^{i<nc_n} (N_{cross})_i \right|$$

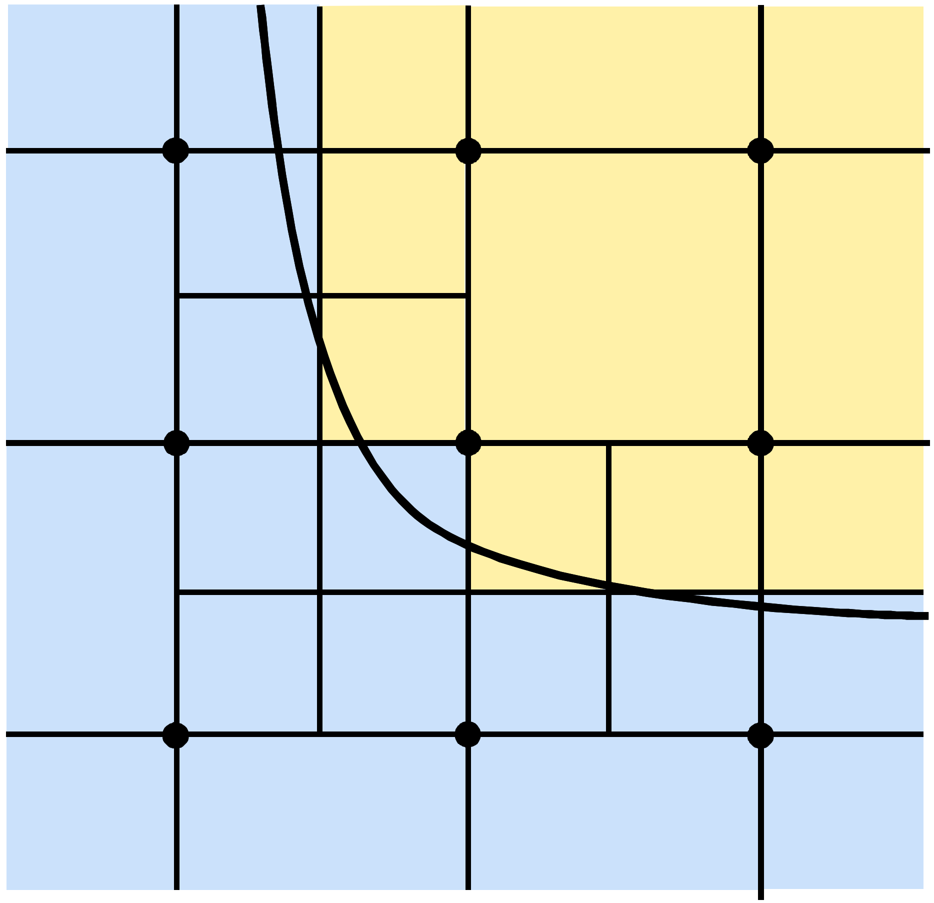




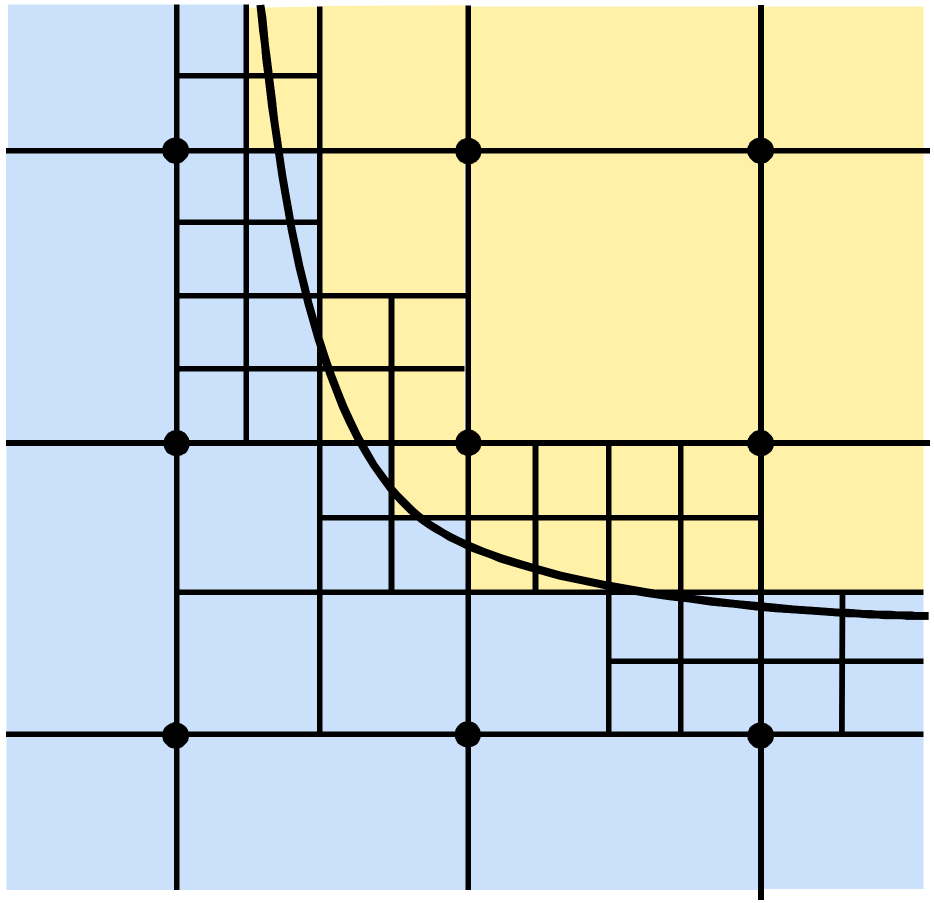
Interface Approximation



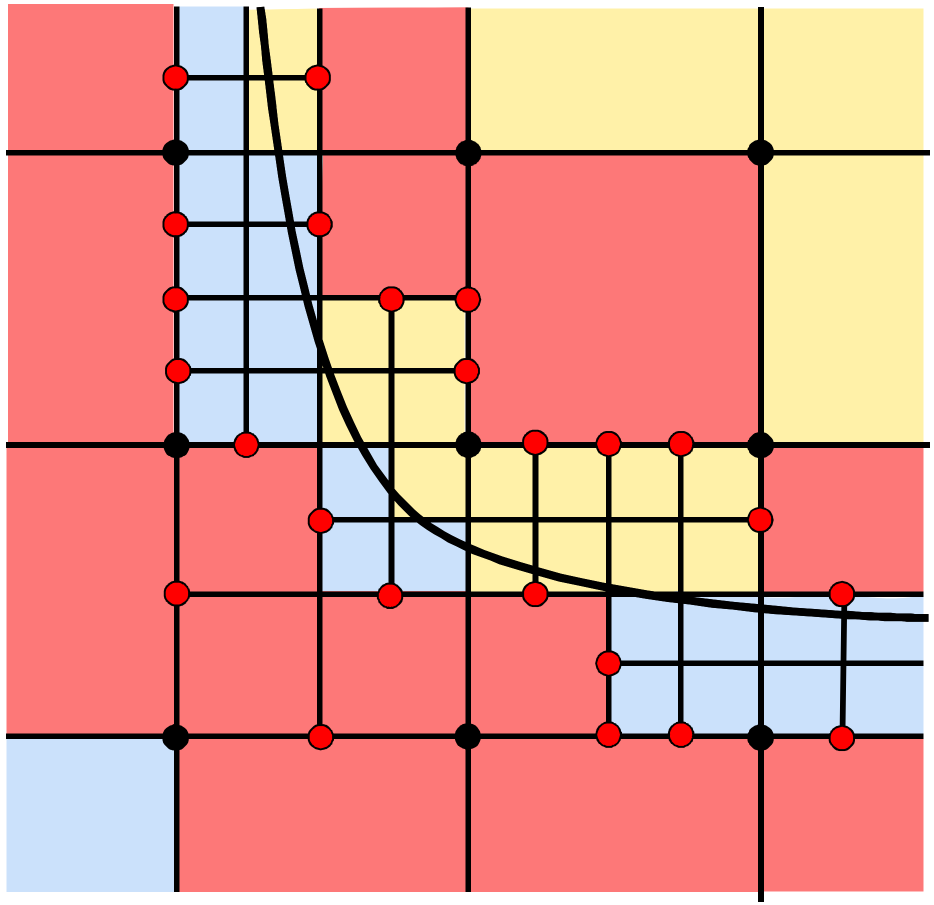
Interface Approximation



Interface Approximation

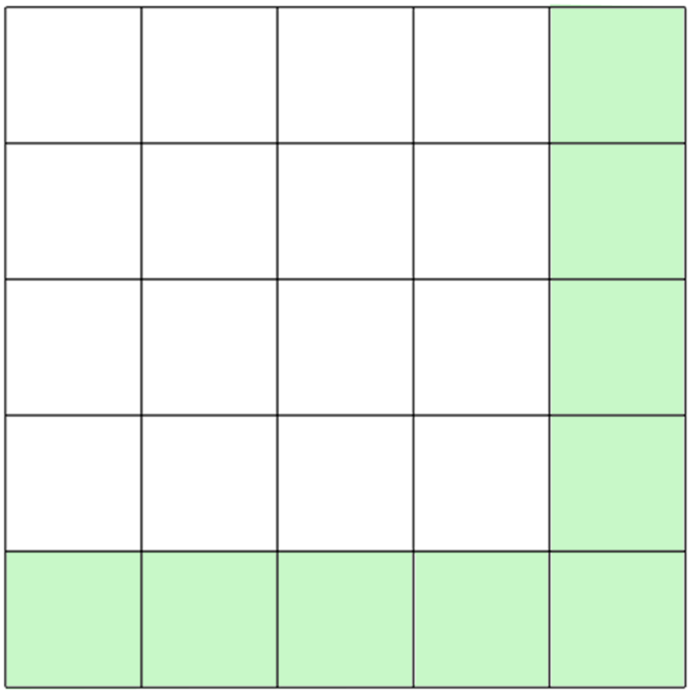


Interface Approximation

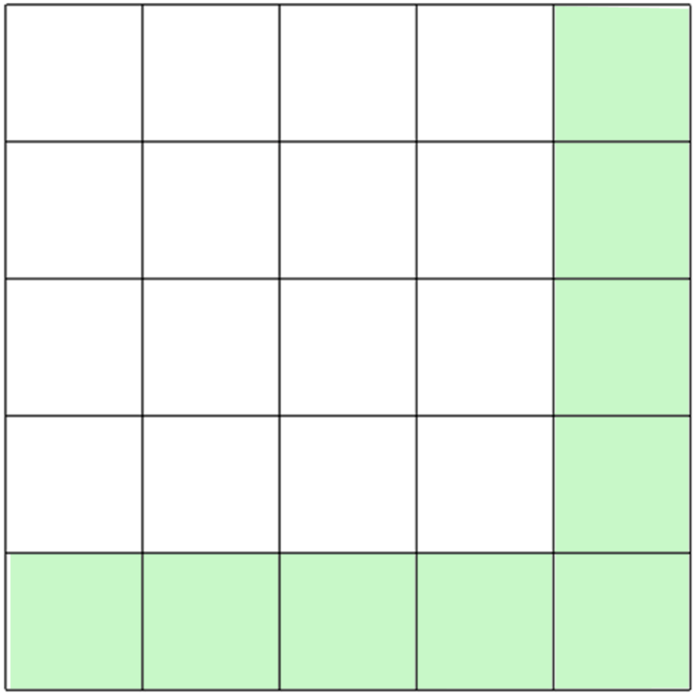


Hex Refinement

3-Refinement



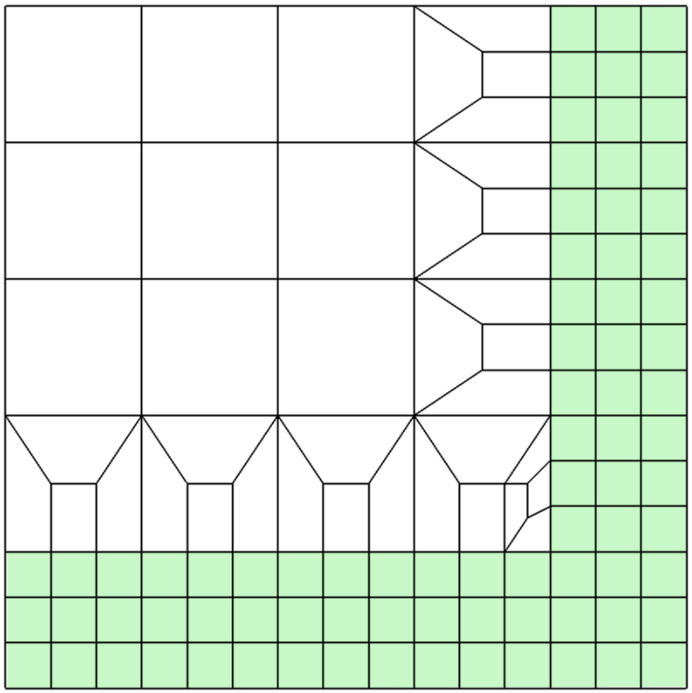
2-Refinement



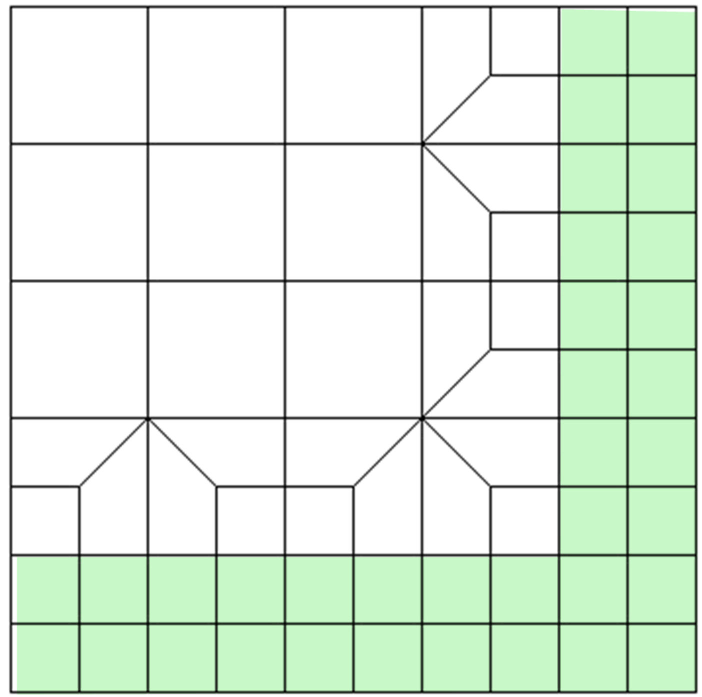
160

Hex Refinement

3-Refinement

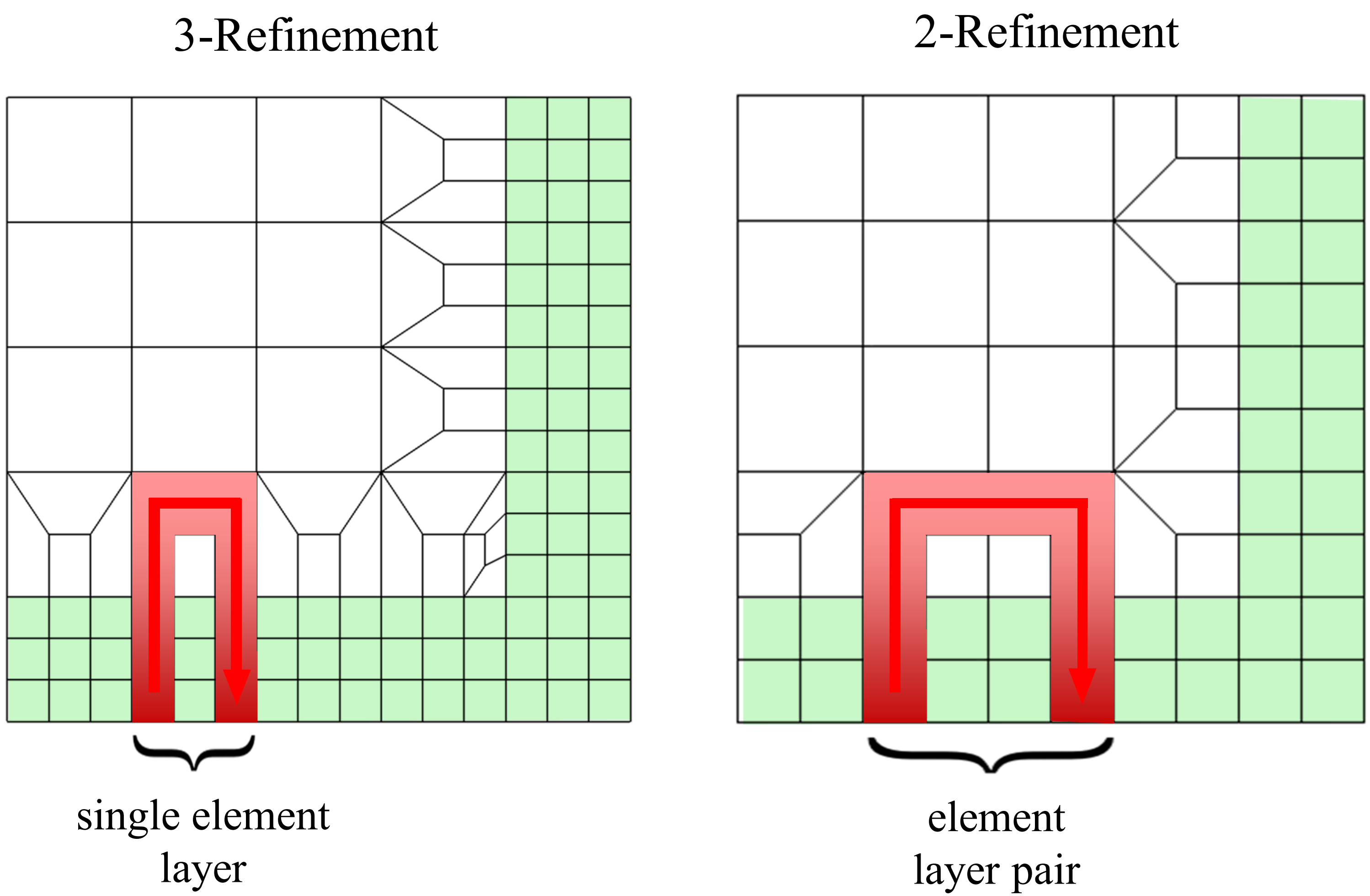


2-Refinement



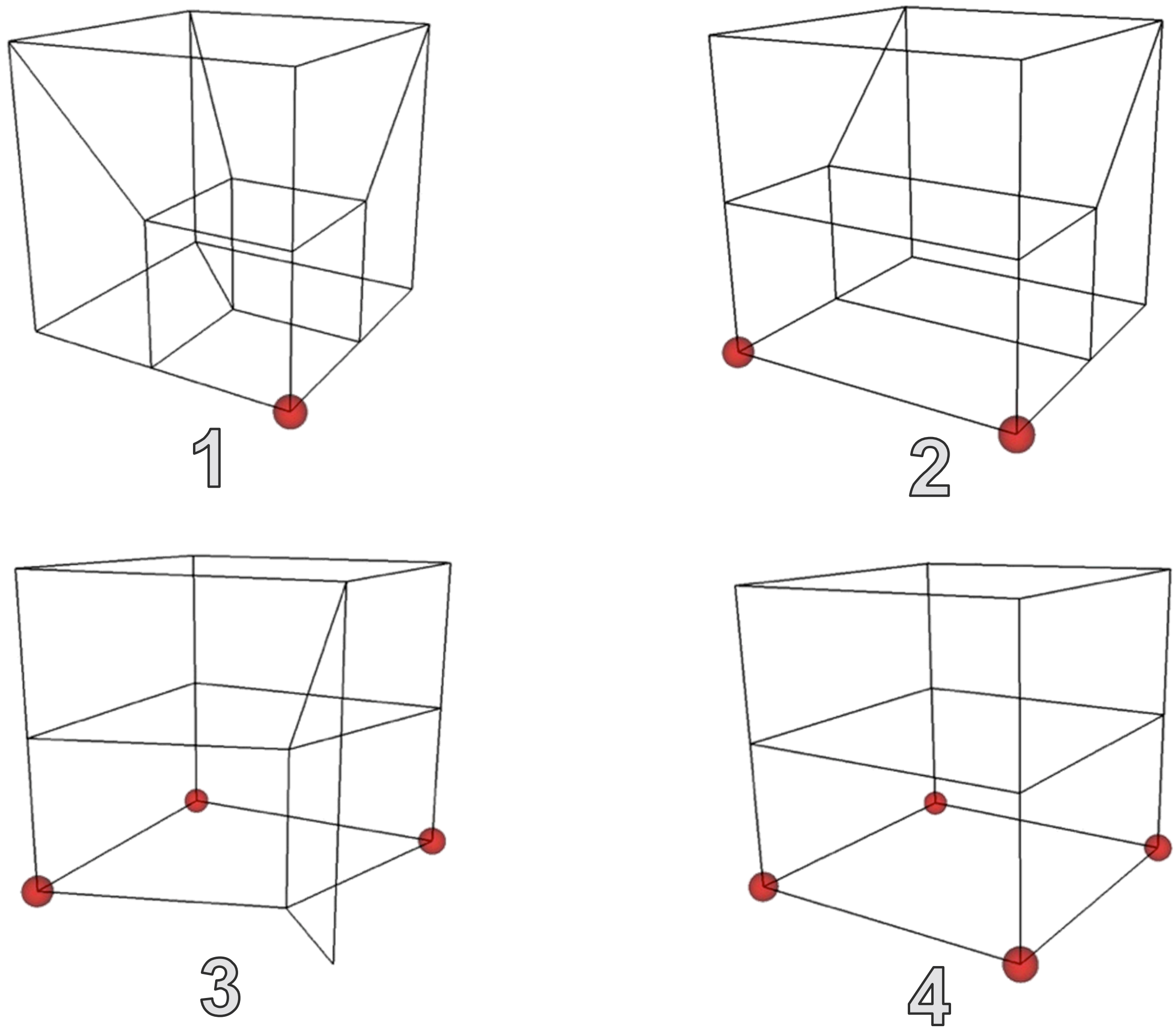
161

Hex Refinement

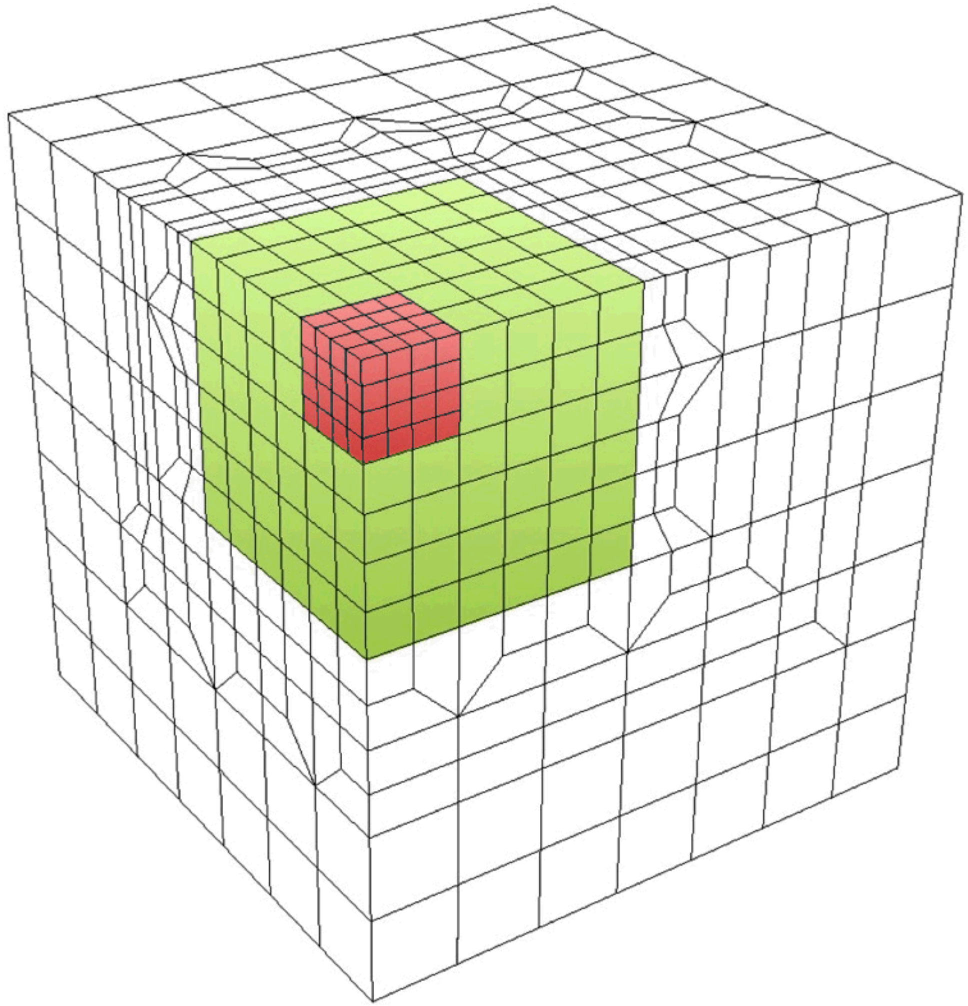


162

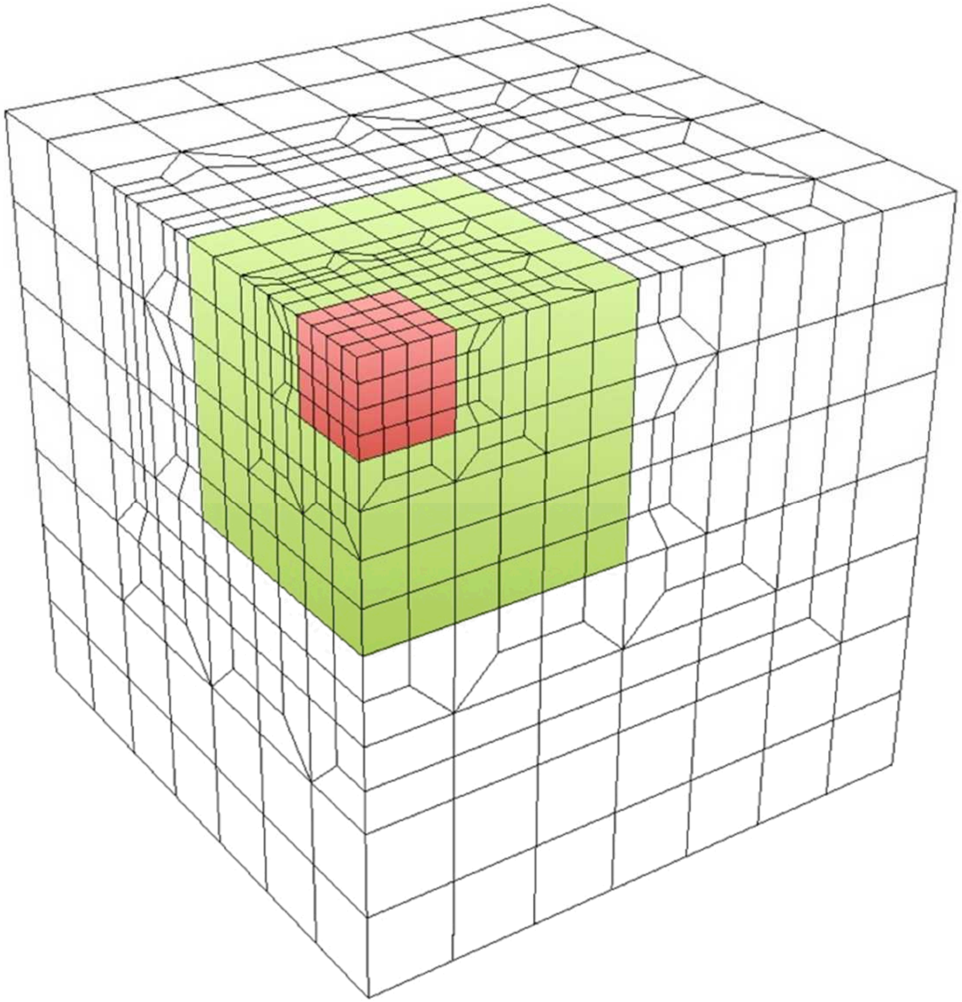
2-Refinement Templates



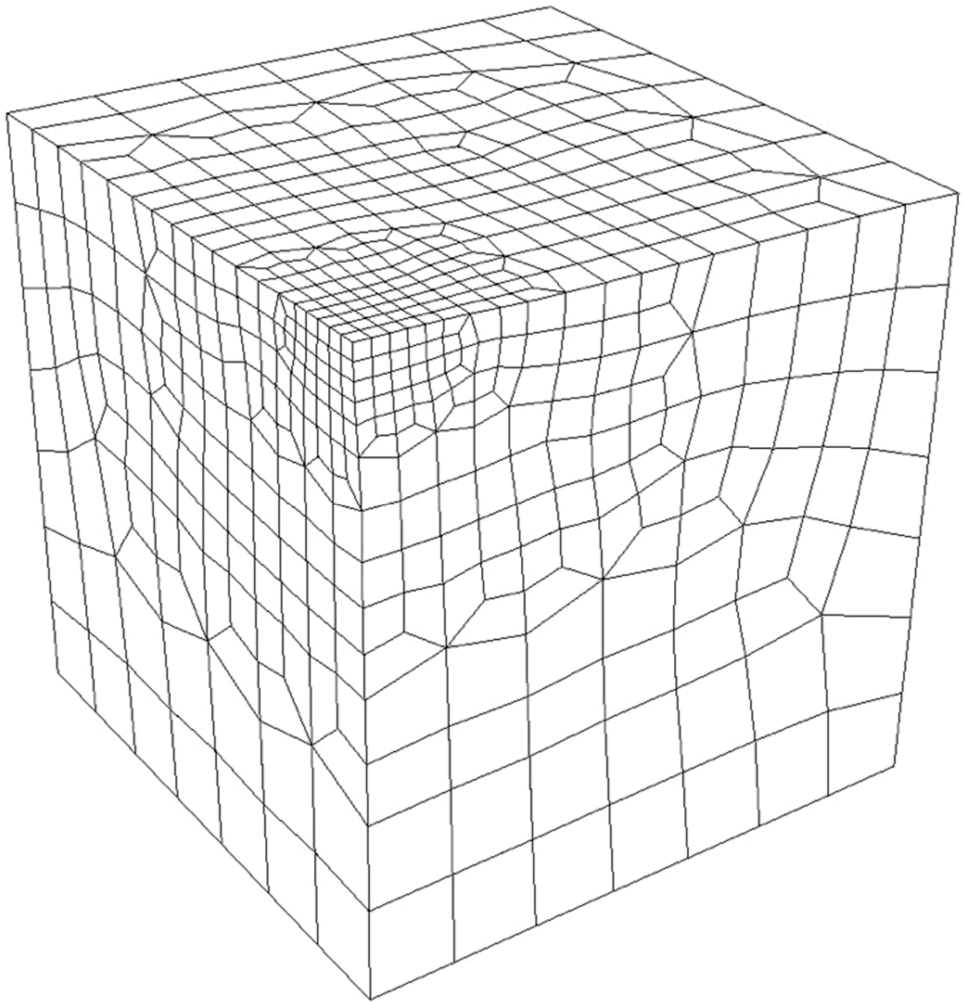
Multi-level Refinement

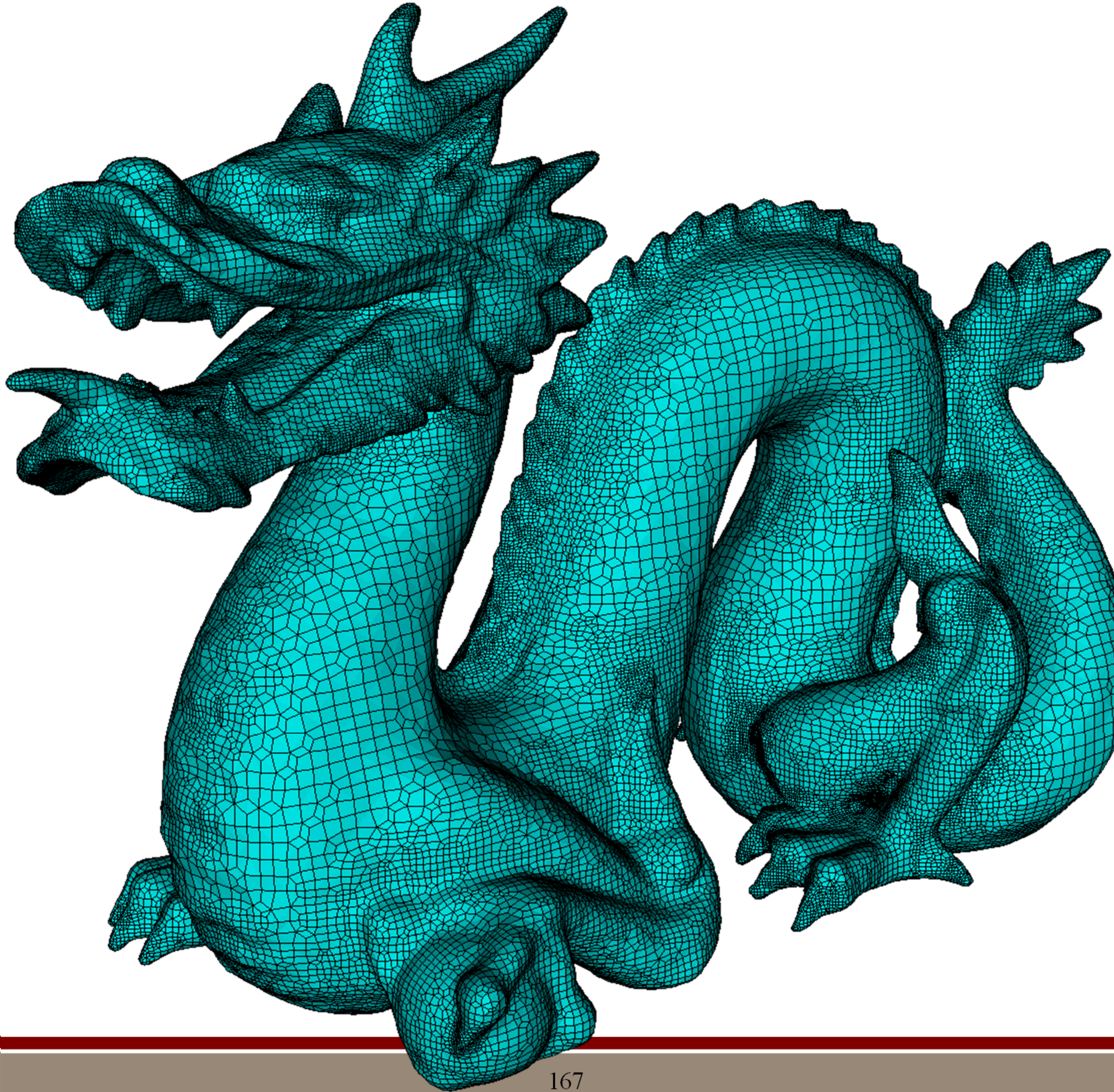


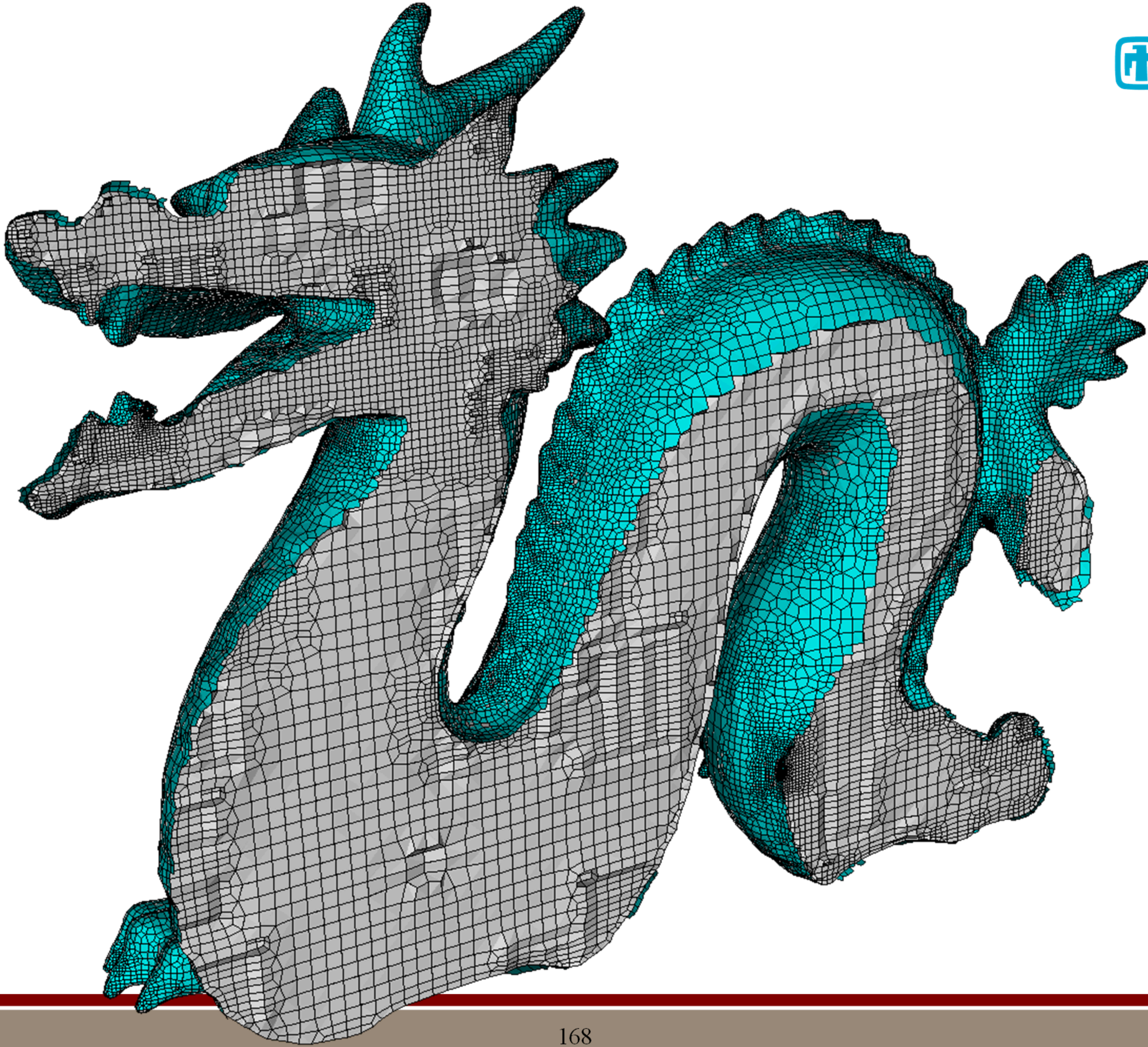
Multi-level Refinement



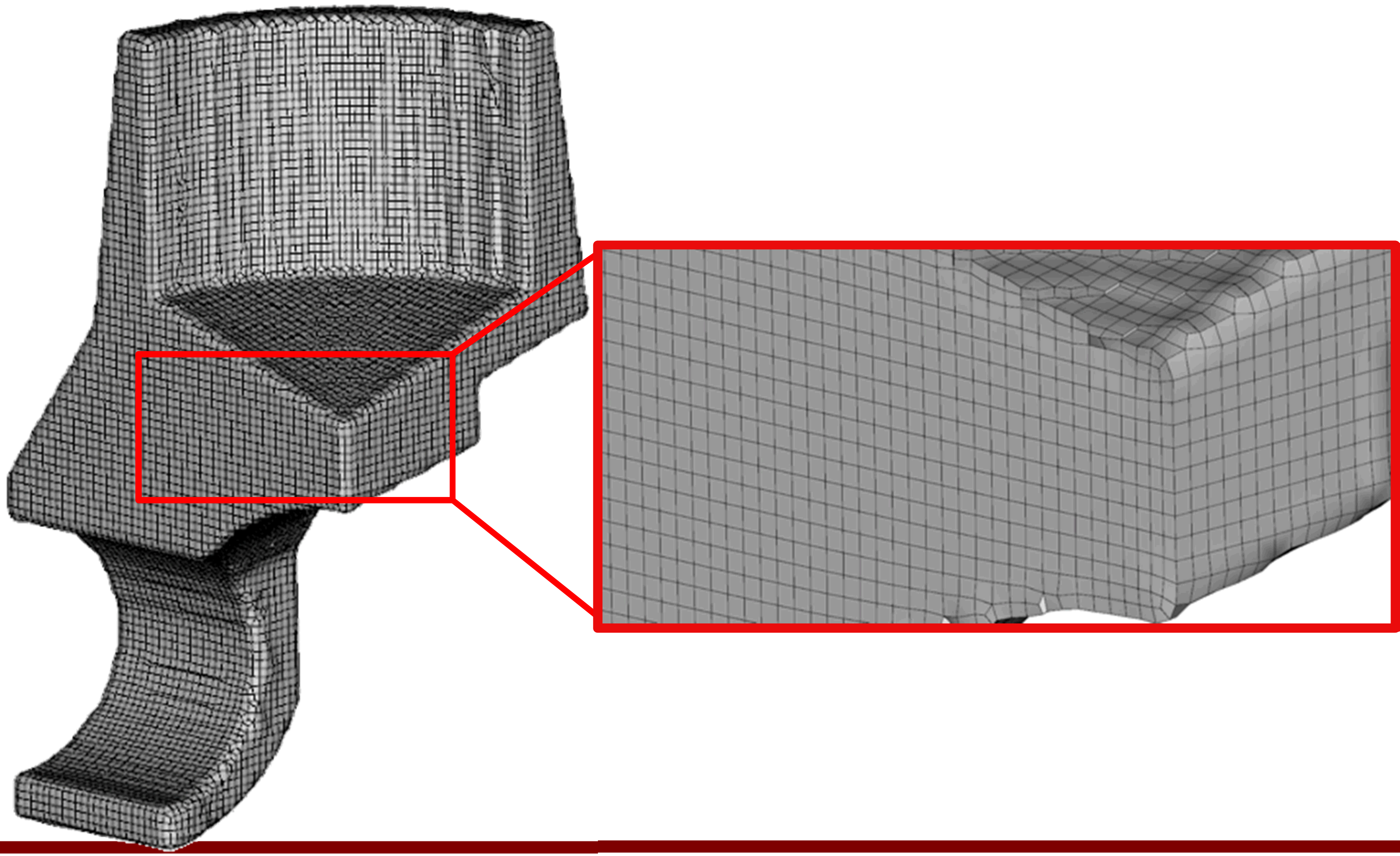
Multi-level Refinement



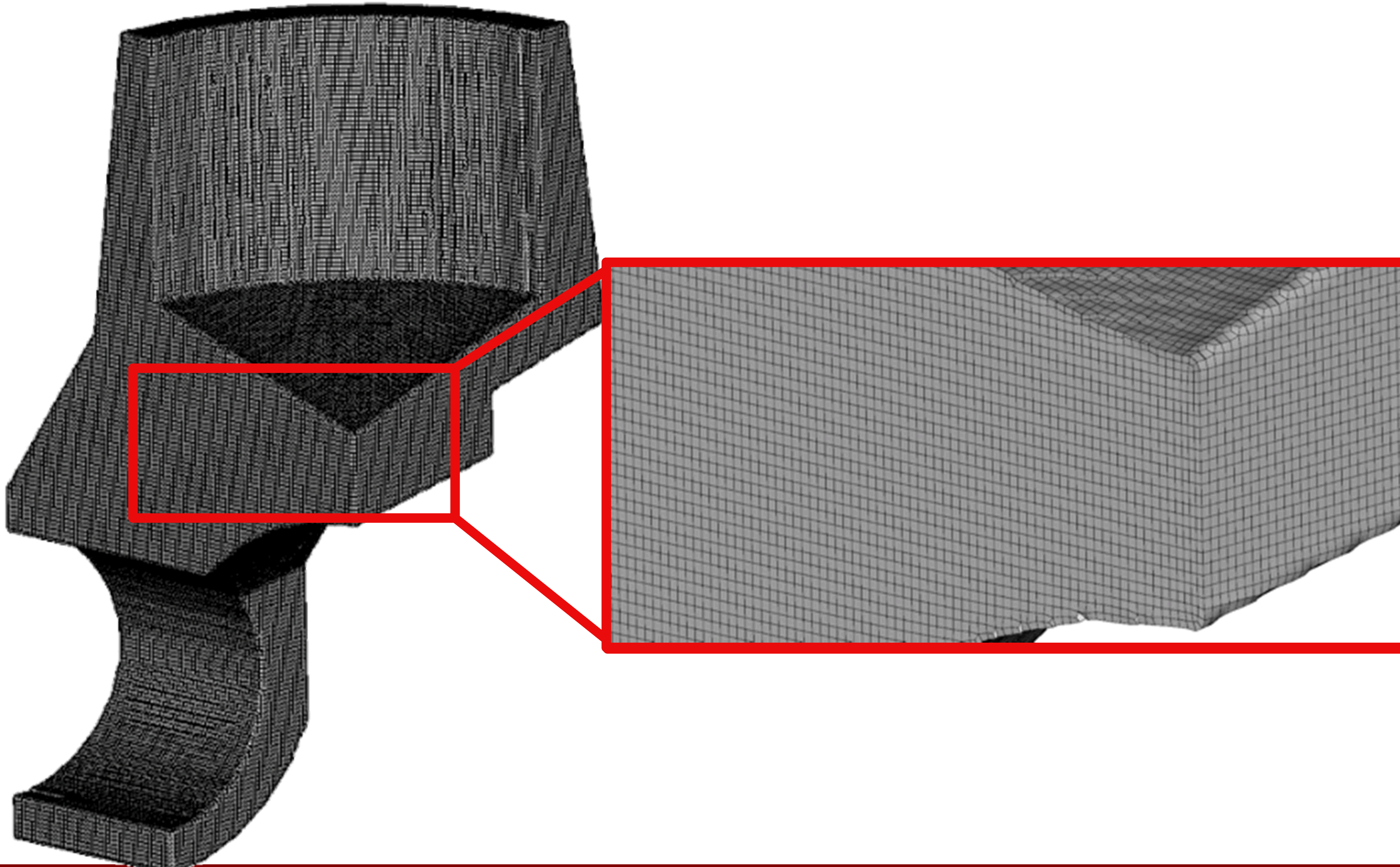




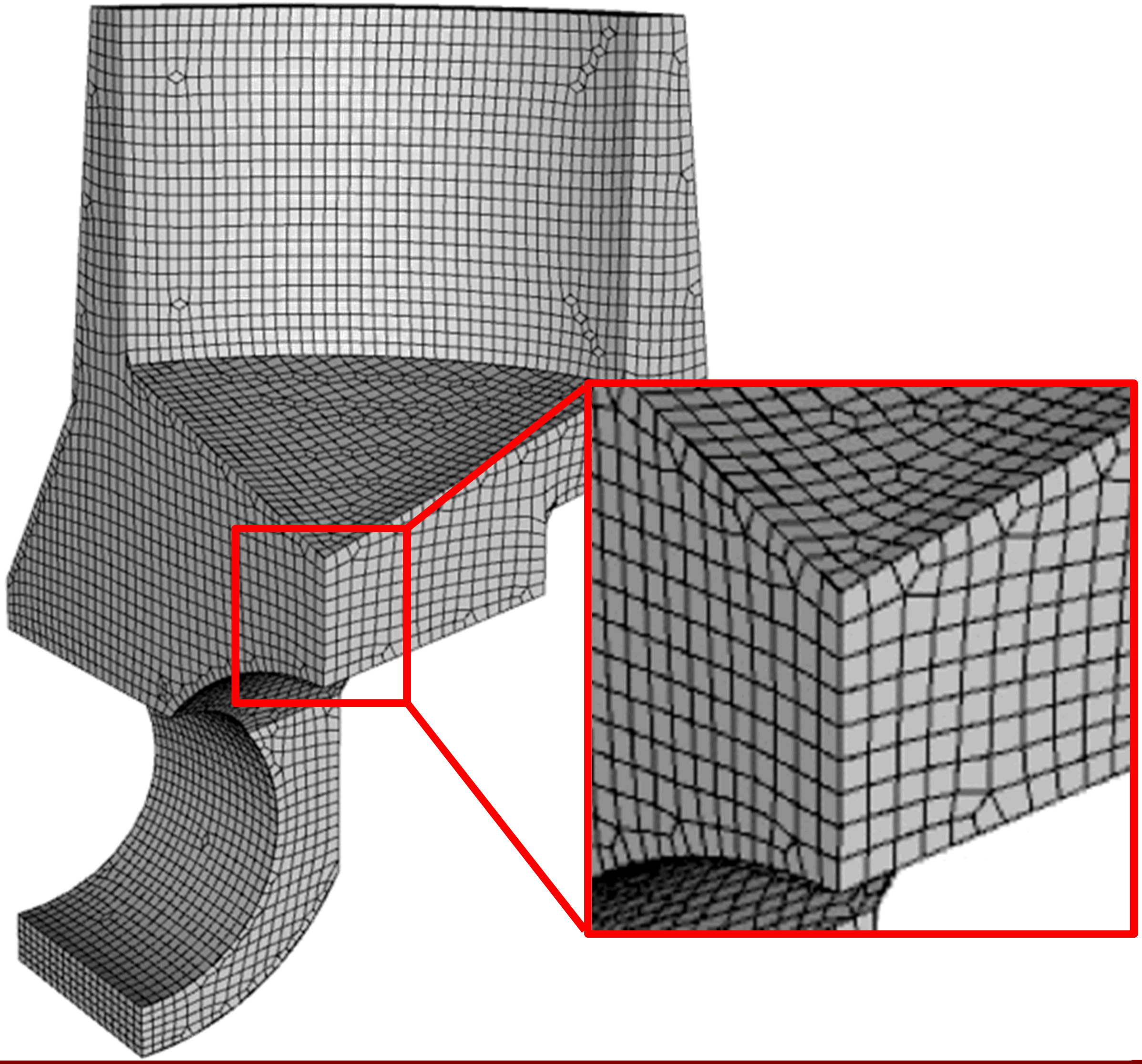
Overlay-Grid Methods



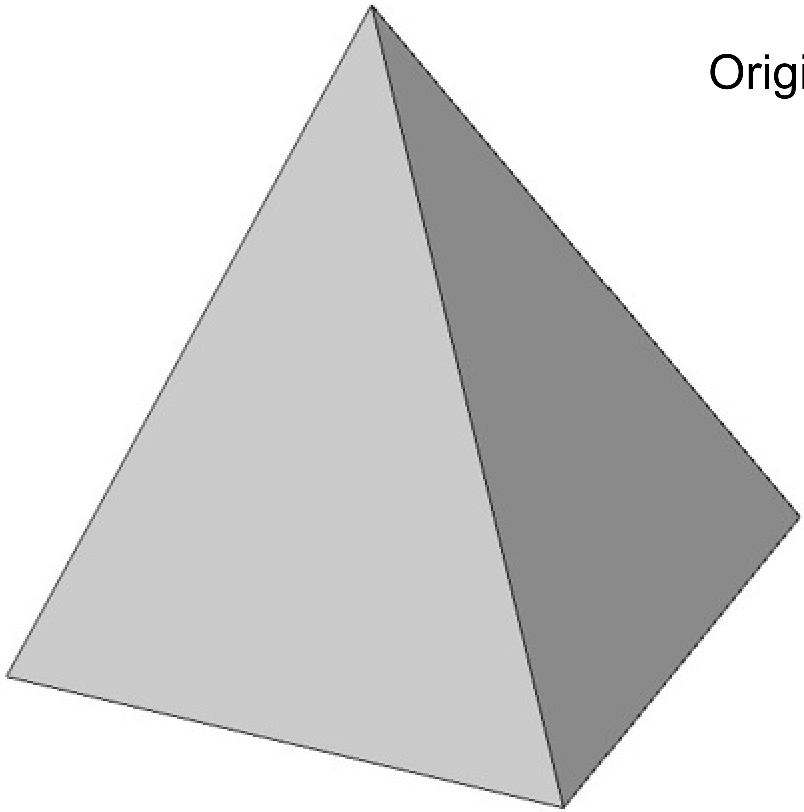
Overlay-Grid Methods



Capturing Features in a CAD Model

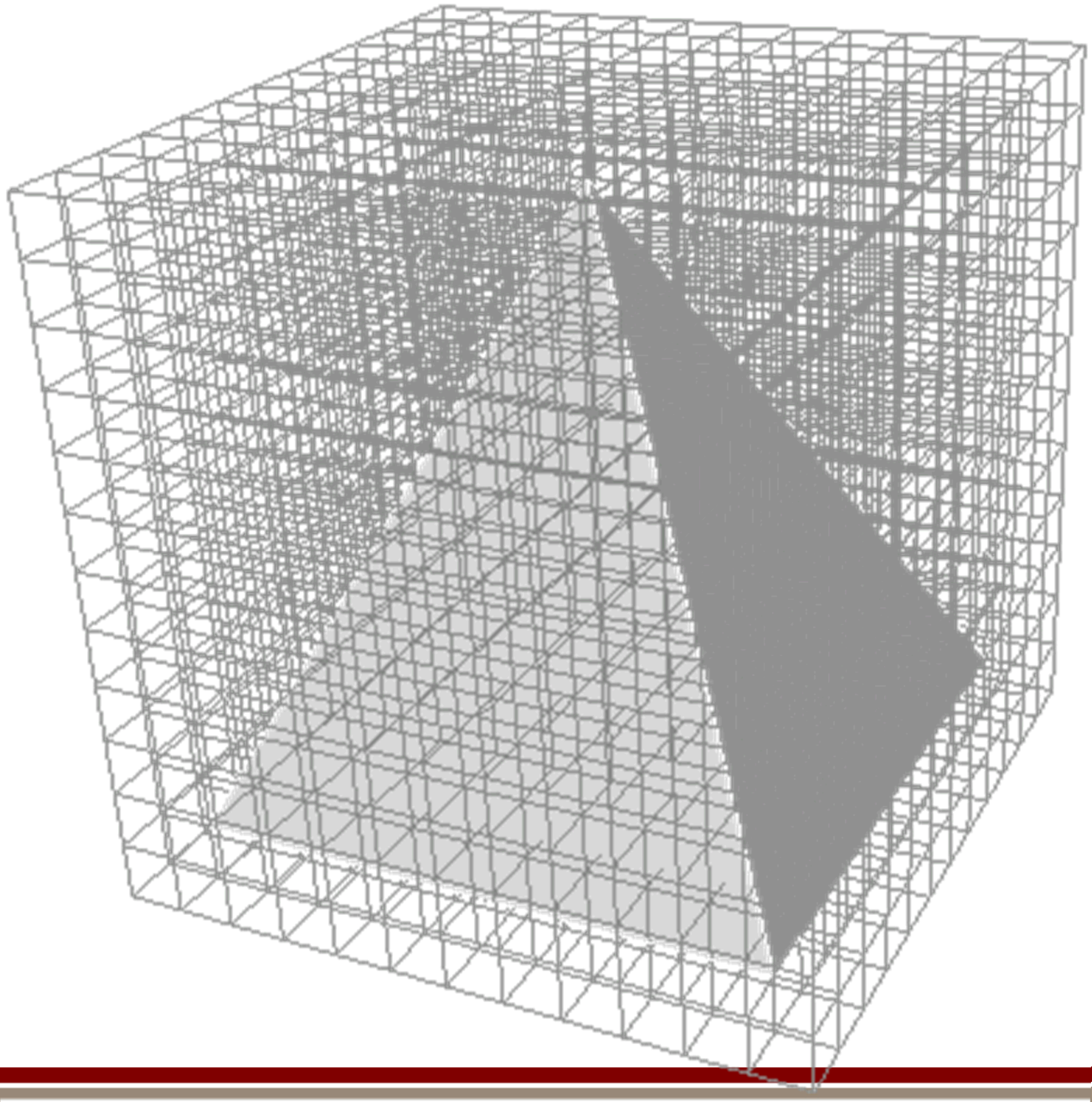


Grid-Based Methods



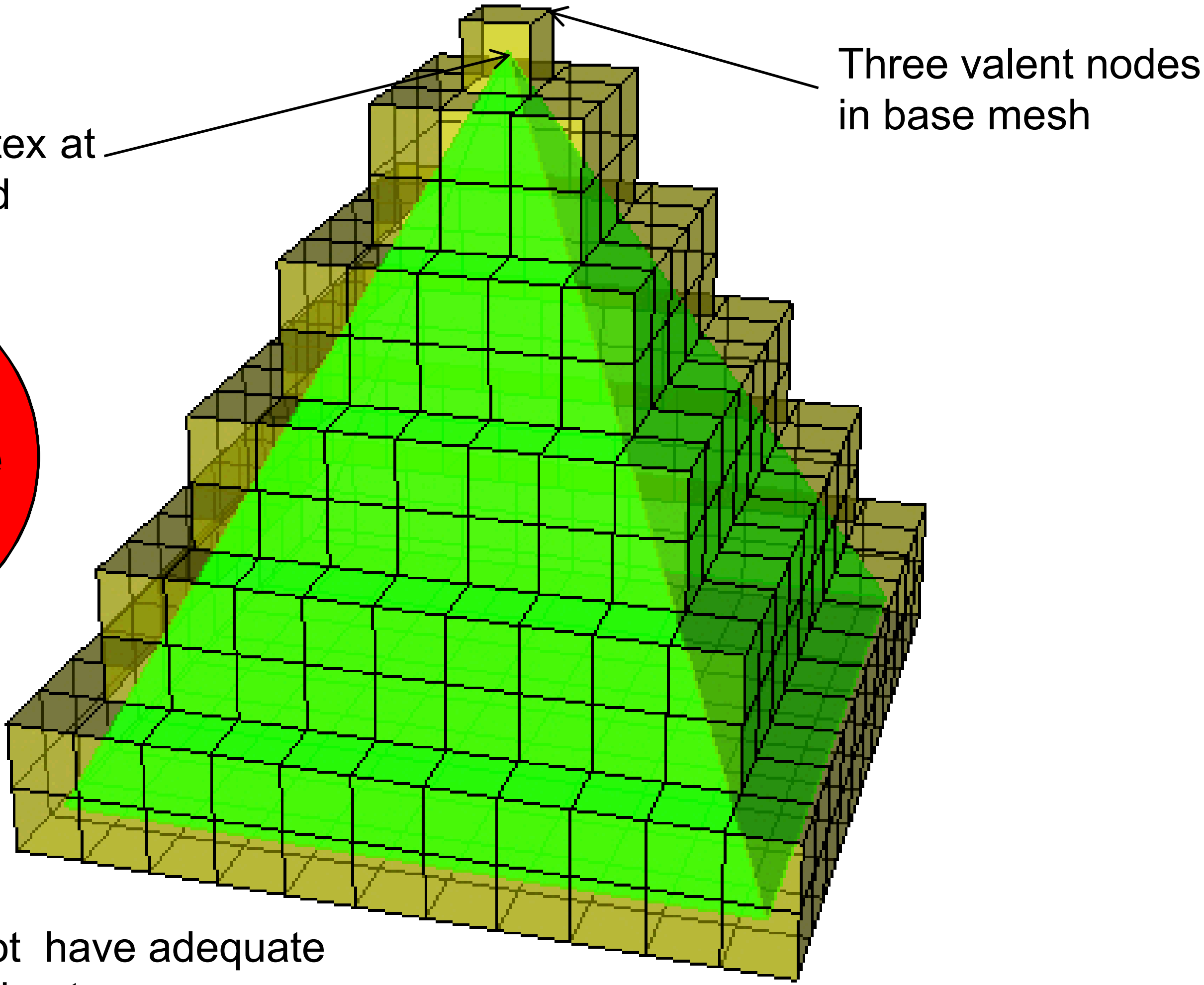
Original geometry

Grid-Based Methods



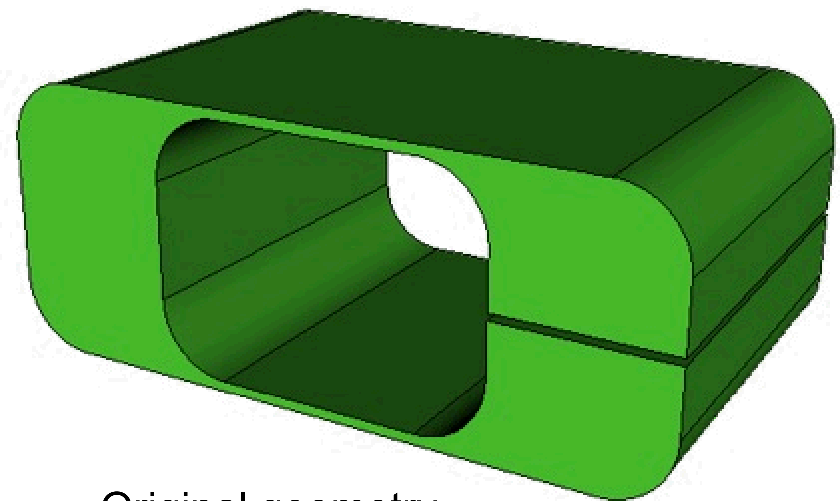
Overlay Grid

Capturing Features

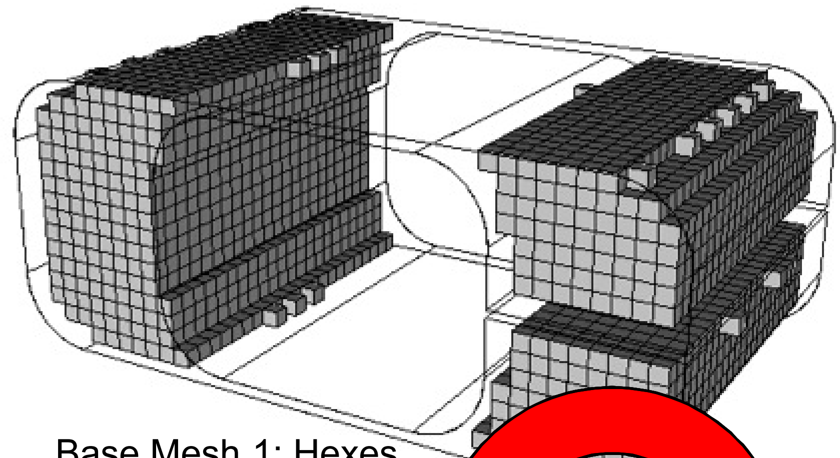


A base mesh may not have adequate topology for embedding to occur

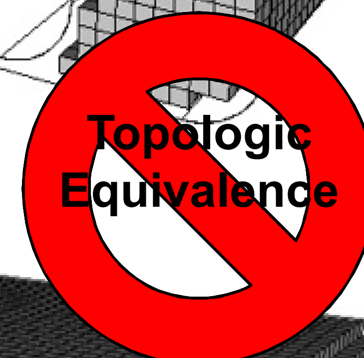
Capturing Features

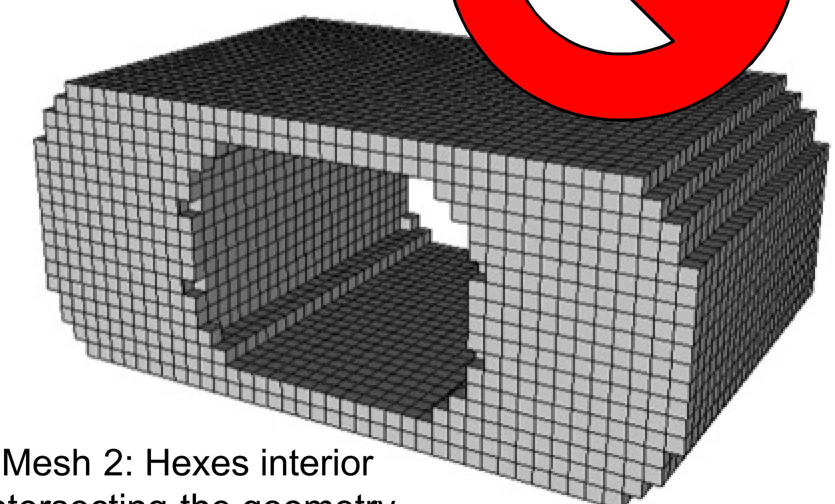


Original geometry



Base Mesh 1: Hexes completely contained within geometry

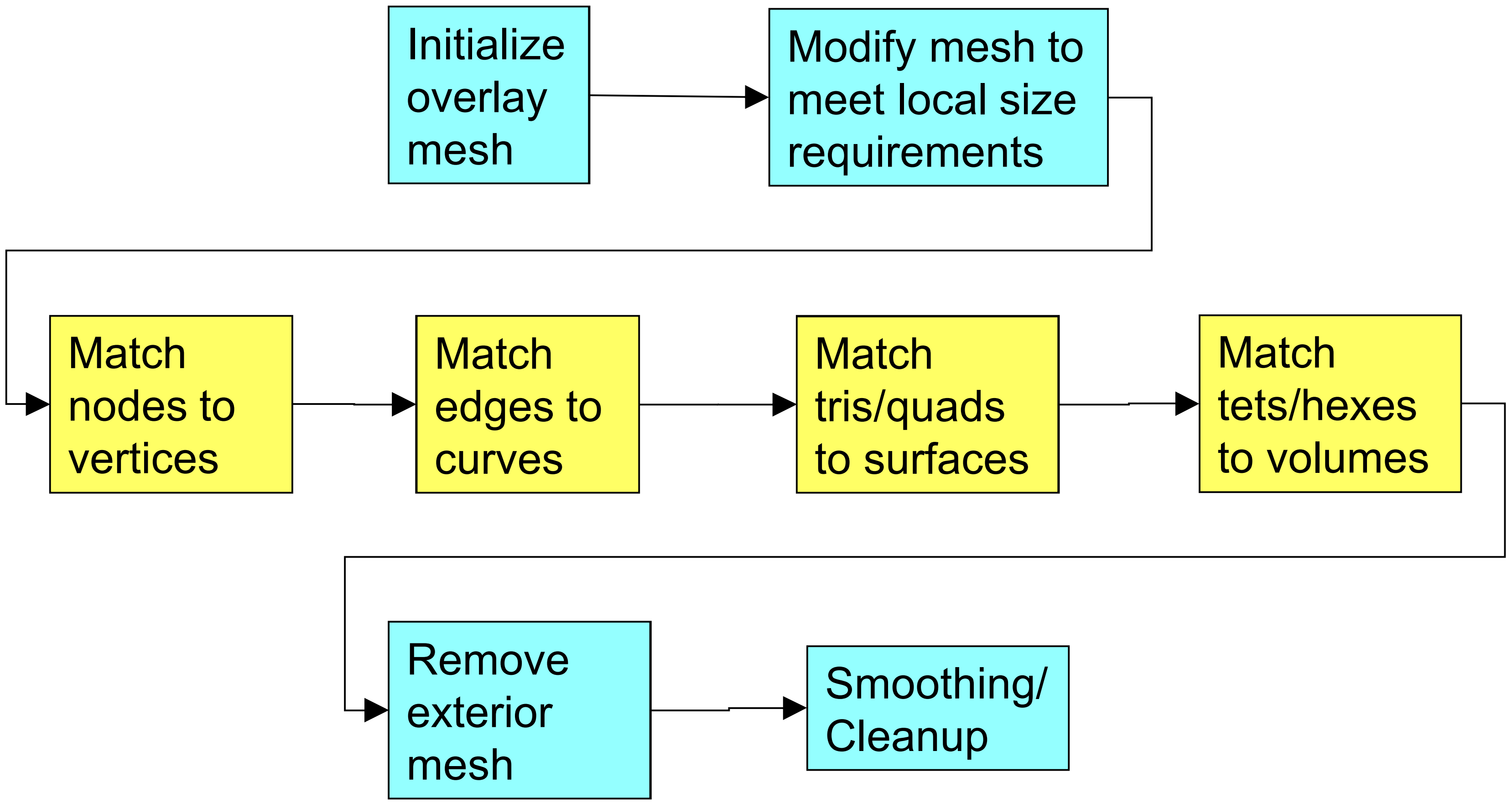




Base Mesh 2: Hexes interior and intersecting the geometry

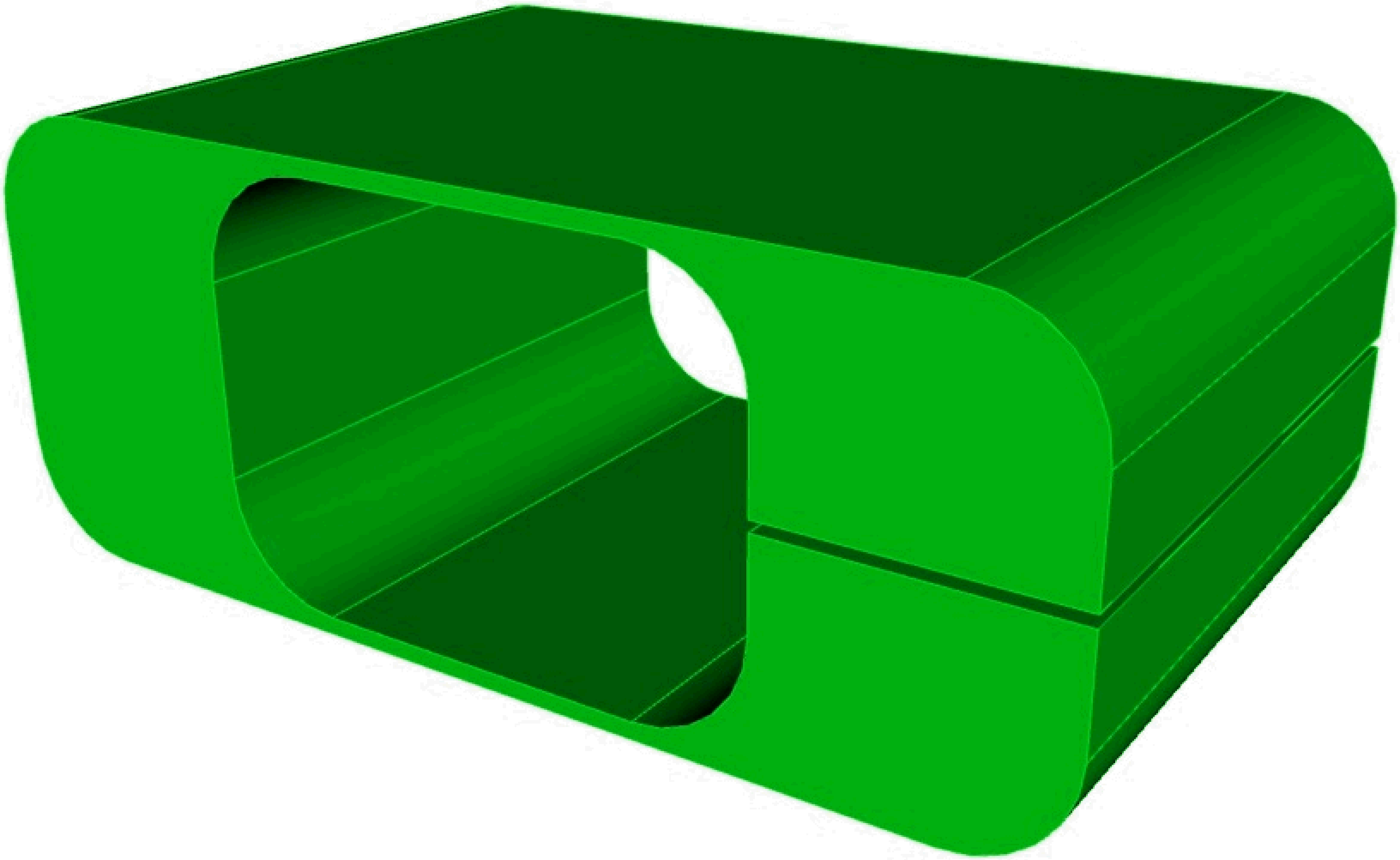
A base mesh may not have adequate topology for embedding to occur

Mesh Generation Process

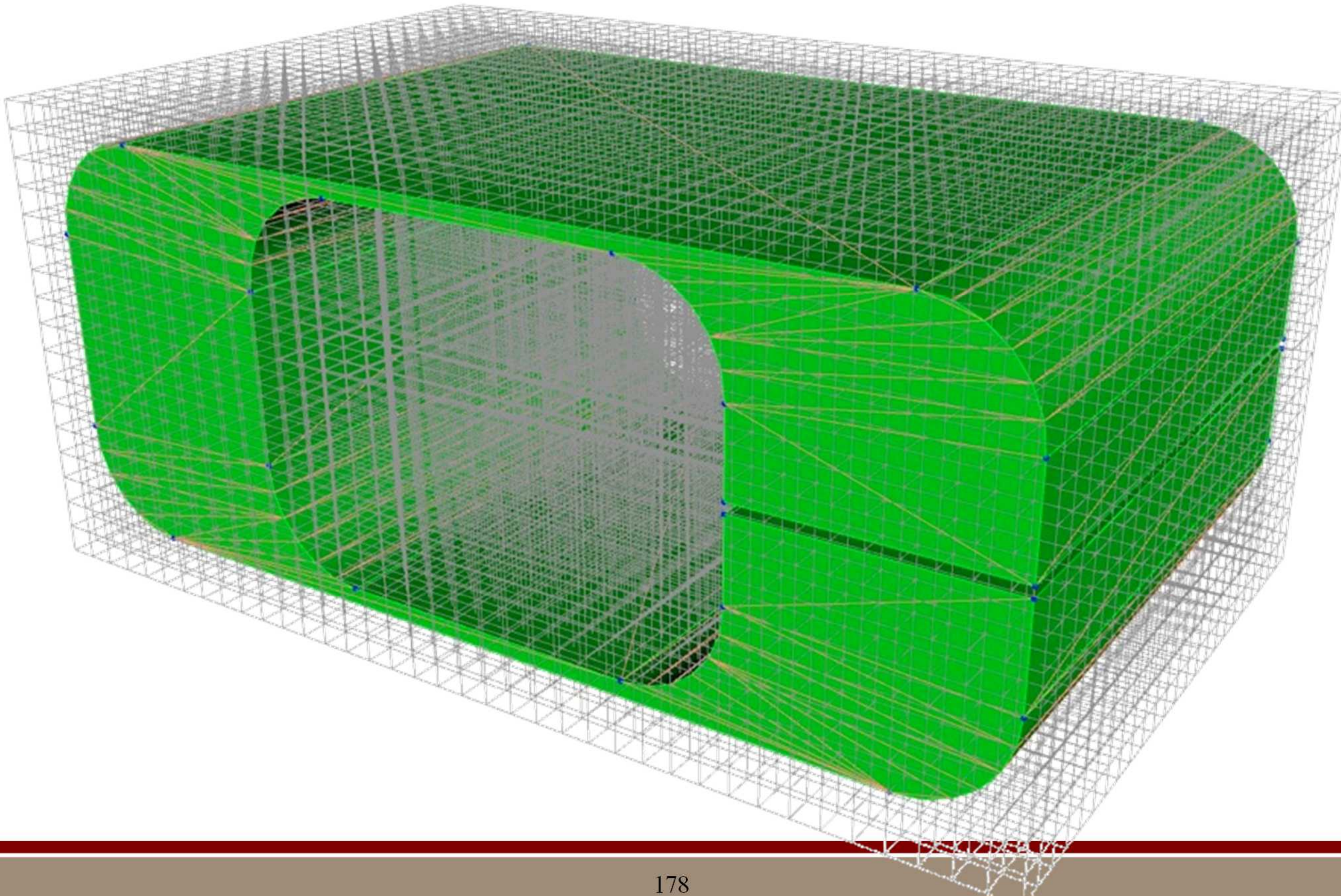


Mesh-First Mesh Generation

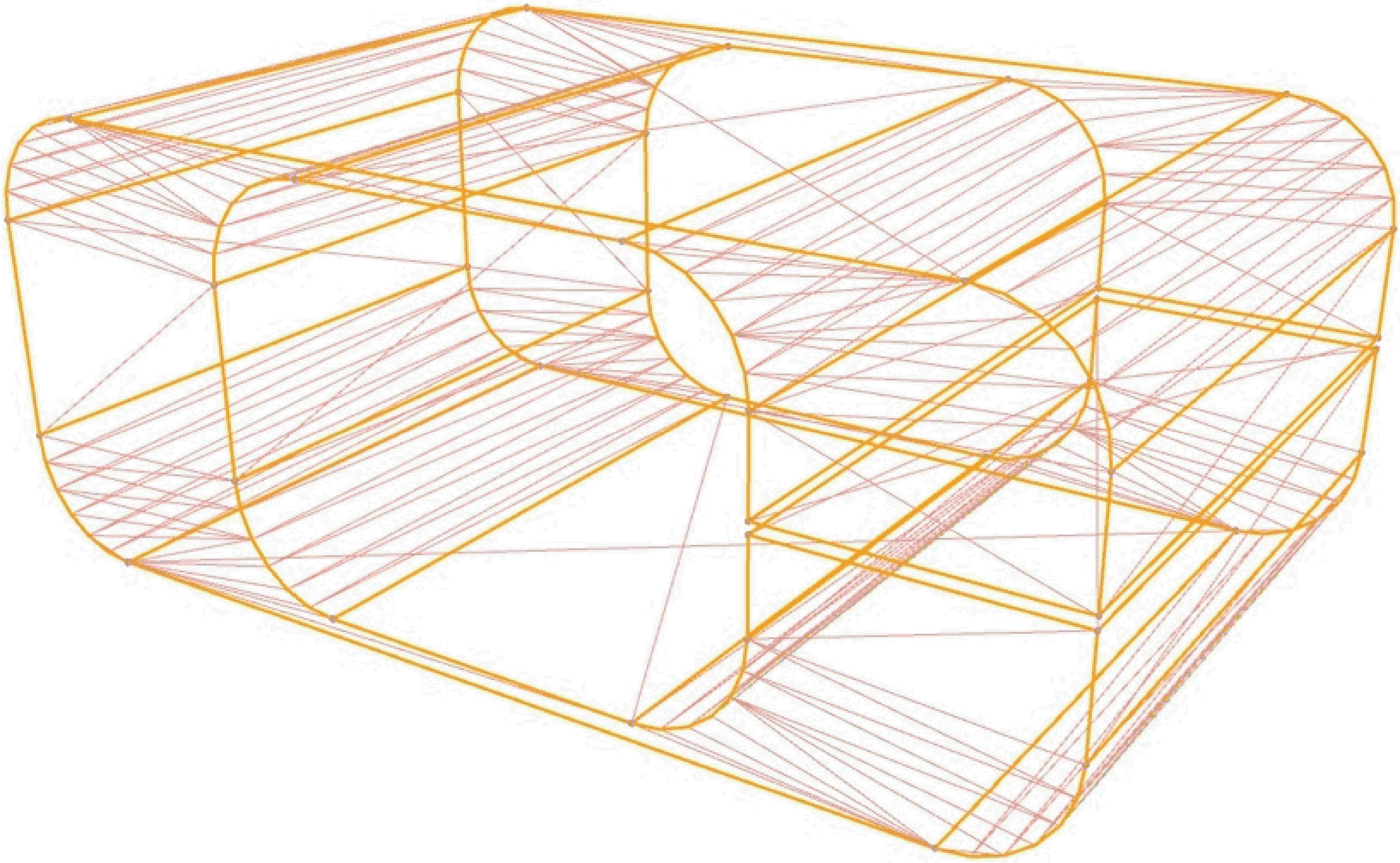
Capturing Features



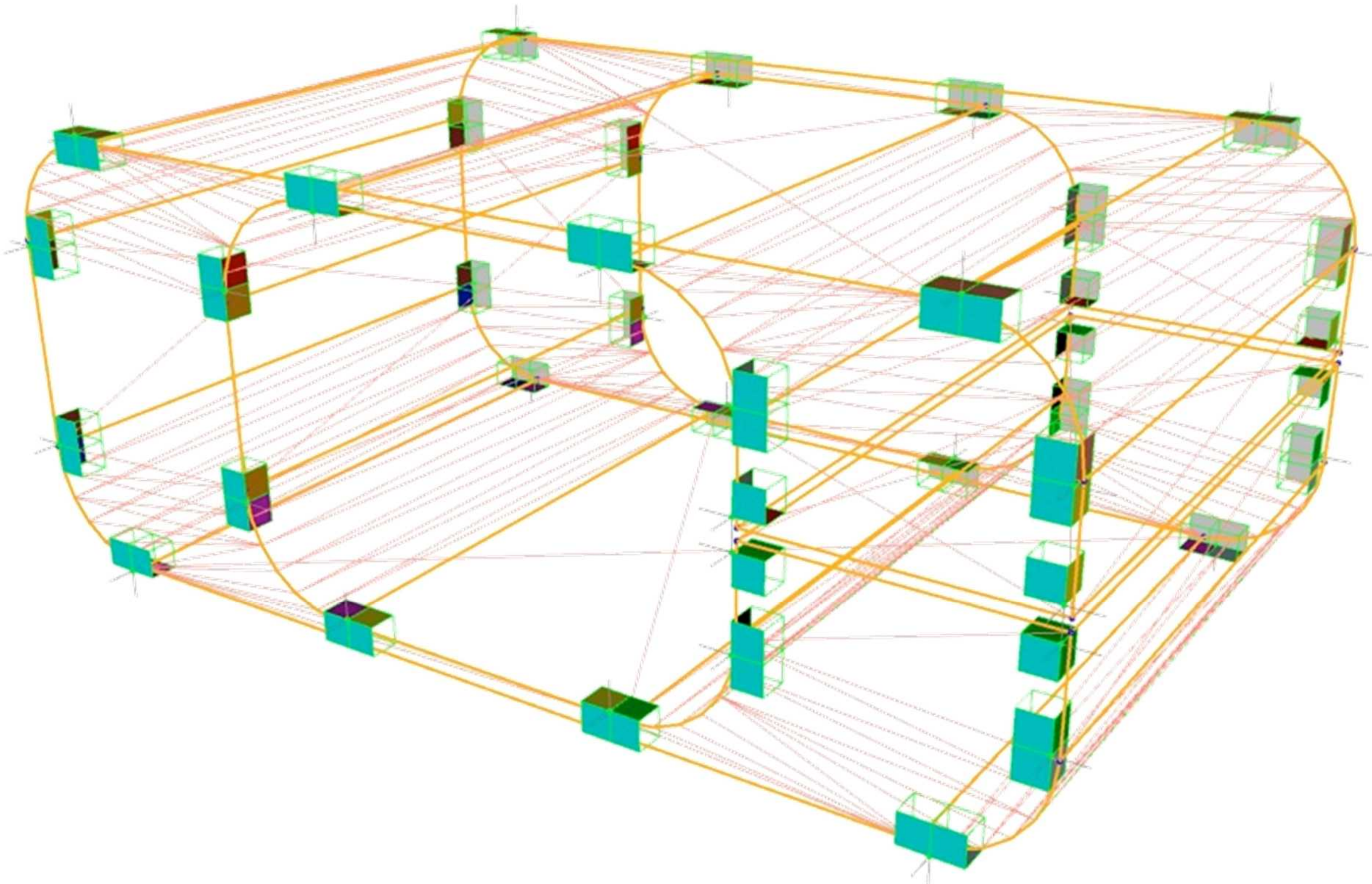
Capturing Features



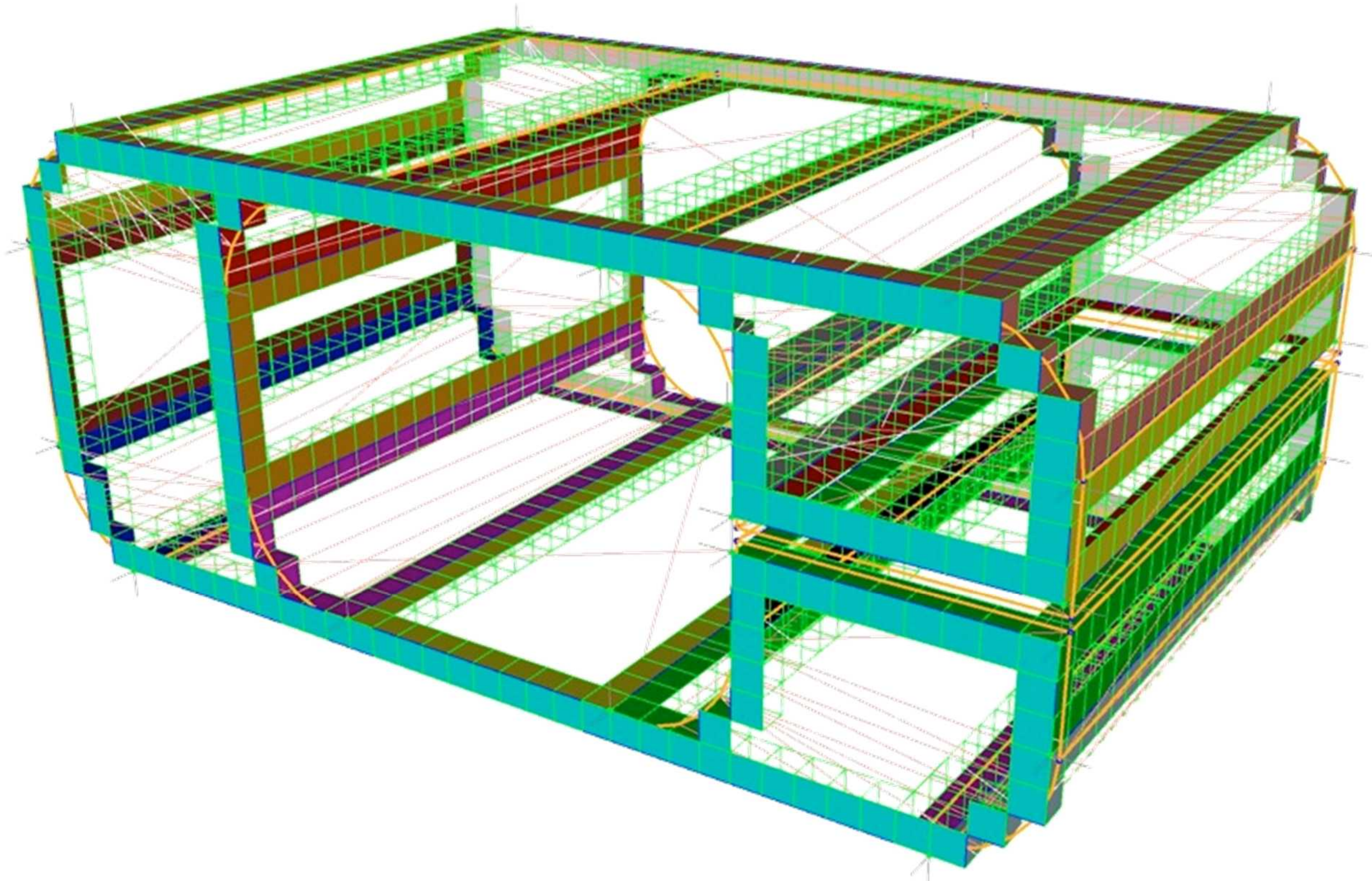
Capturing Features



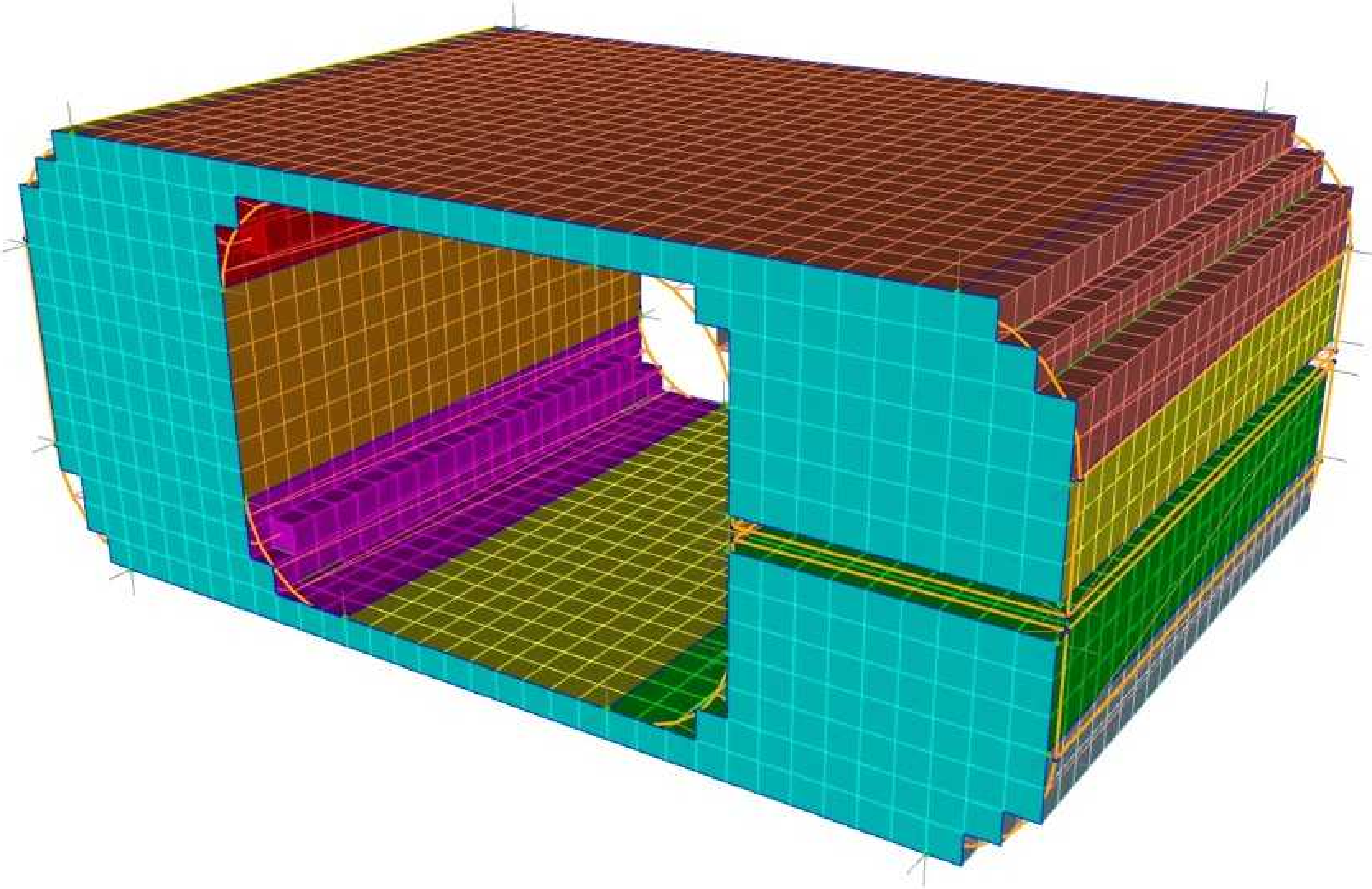
Capturing Features



Capturing Features

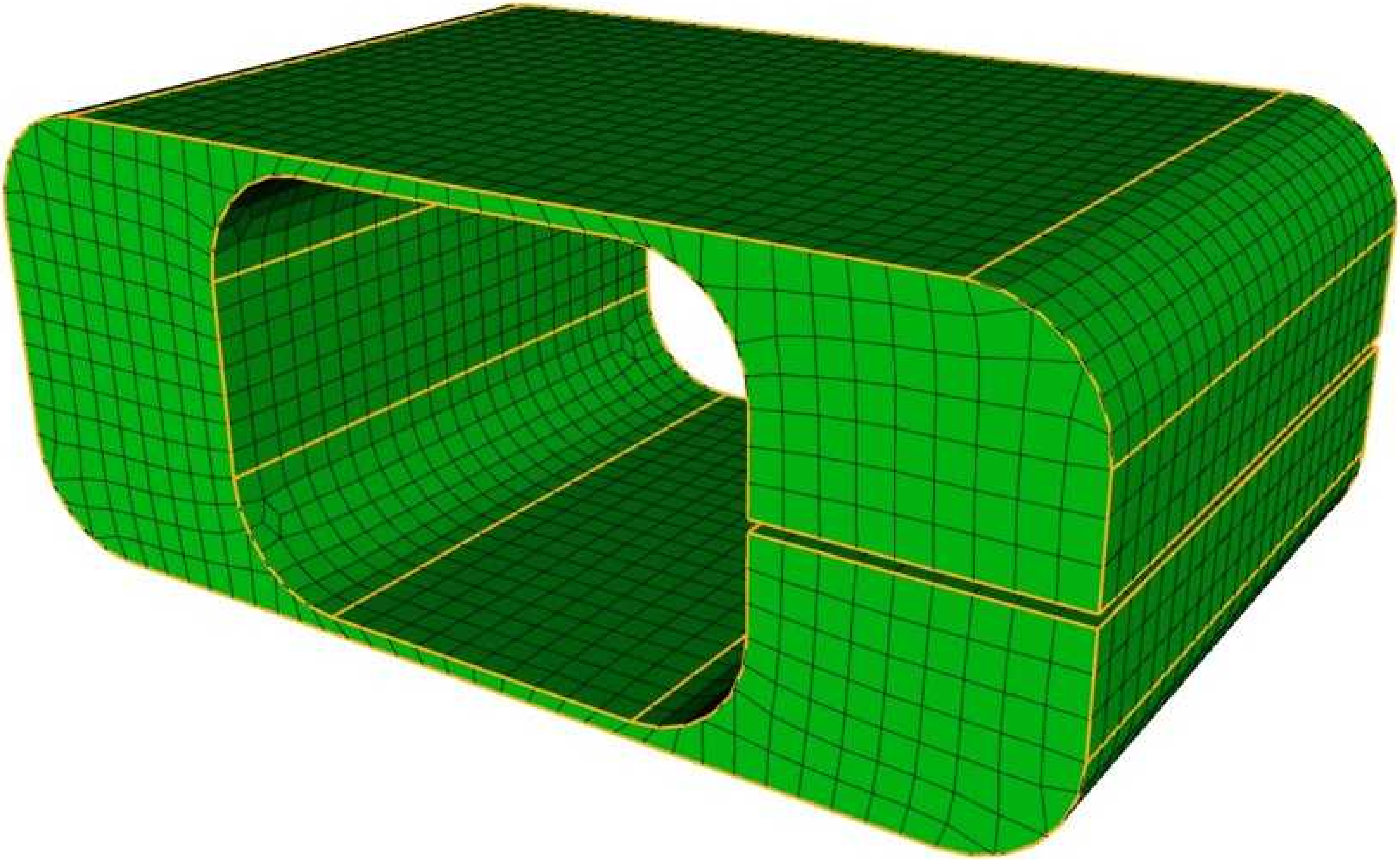


Capturing Features

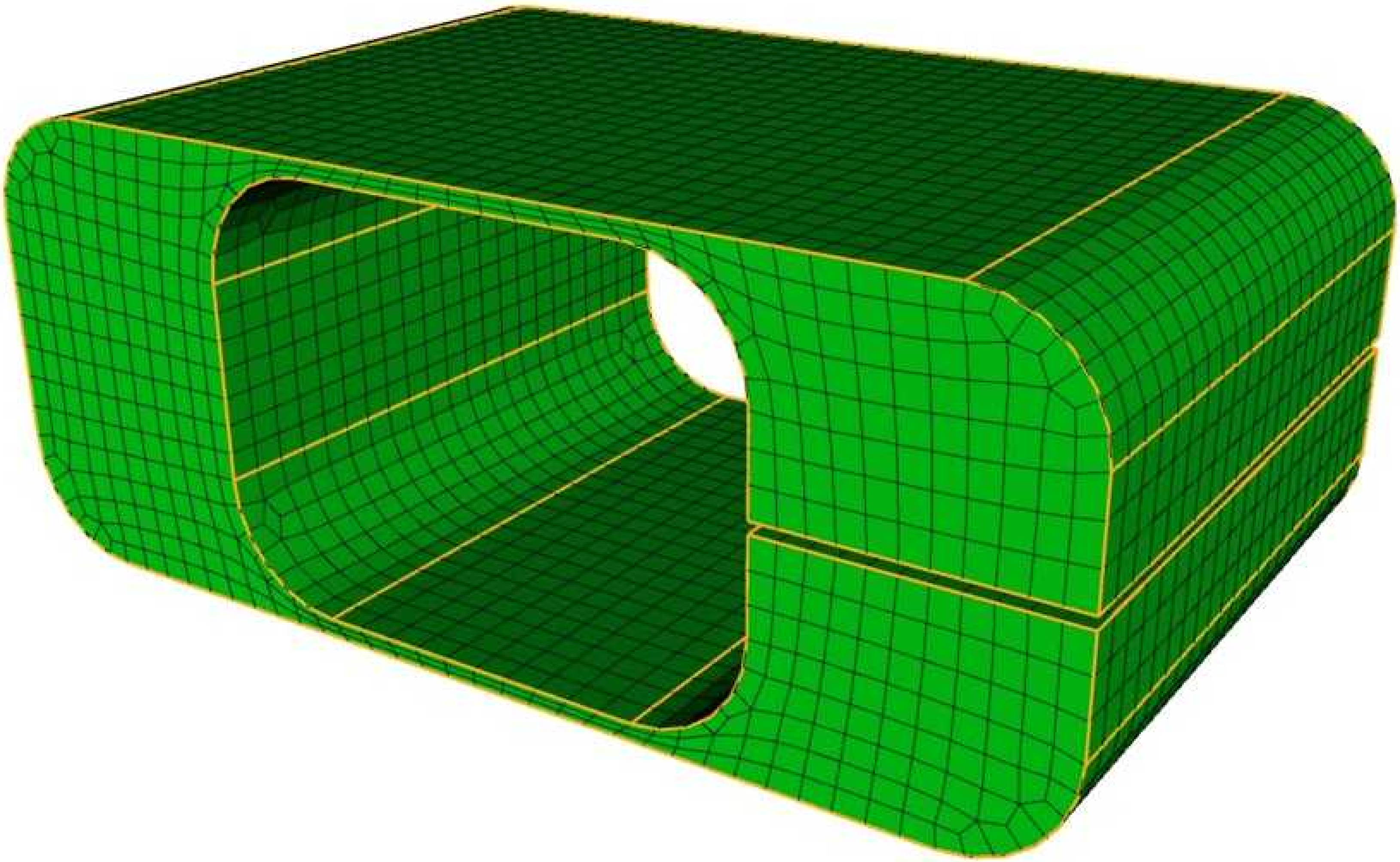


Capturing Features

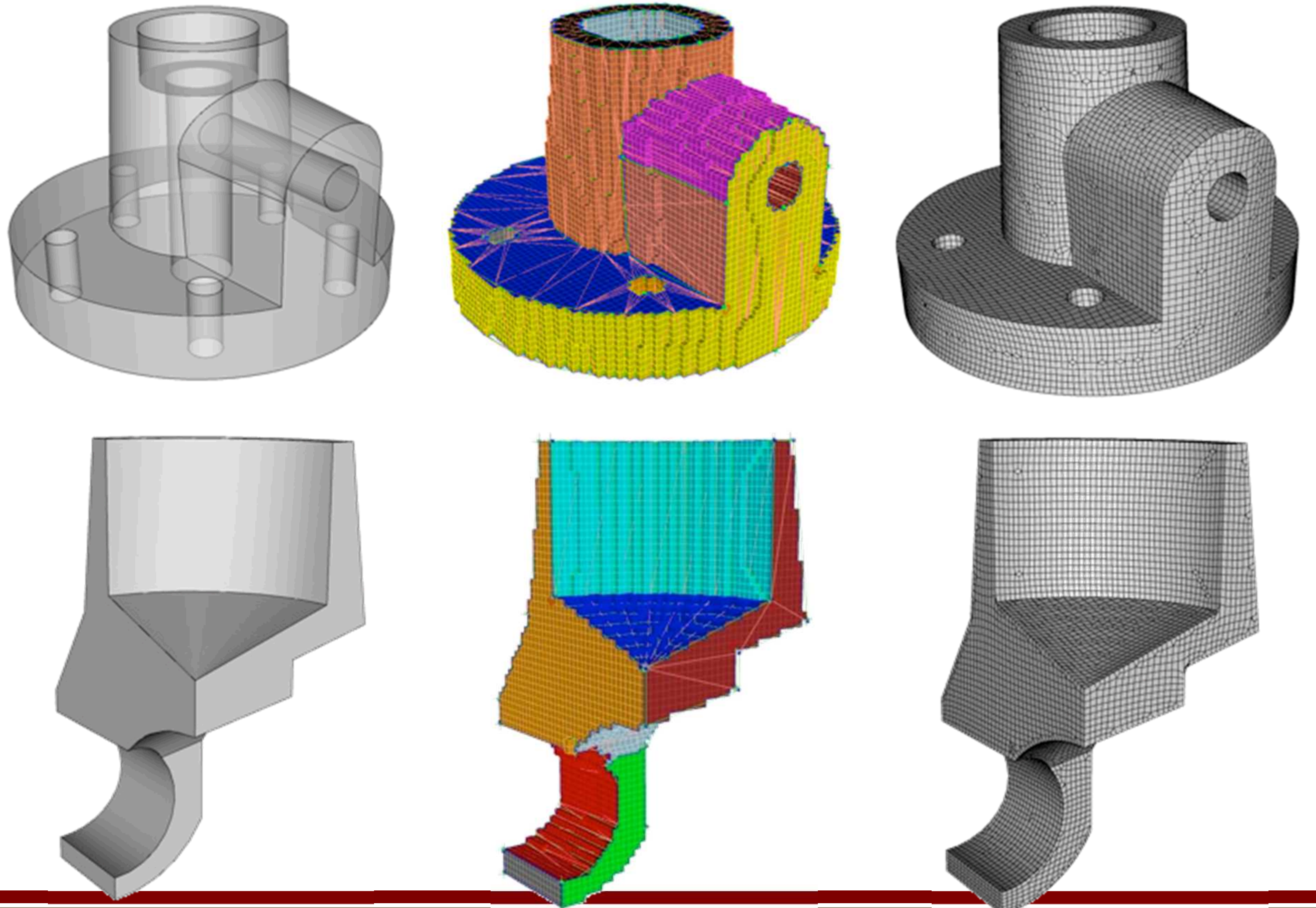
Sandia National Laboratories



Capturing Features

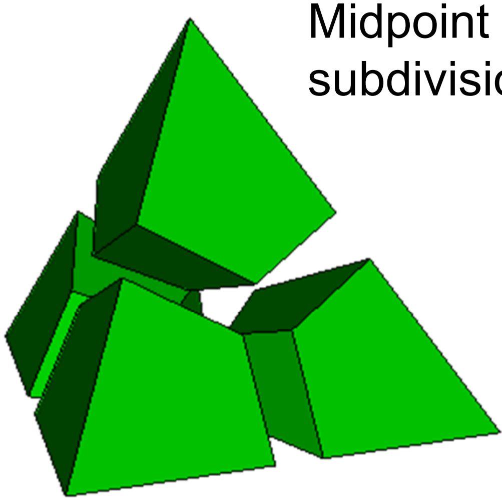


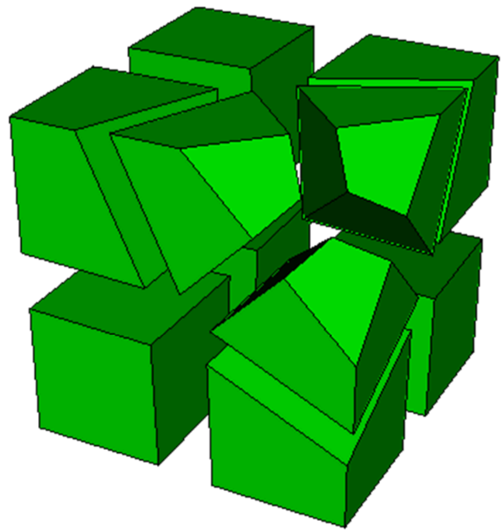
Capturing Features

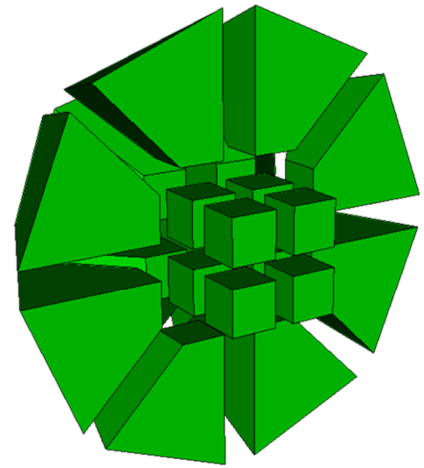


Automatic Block Decomposition

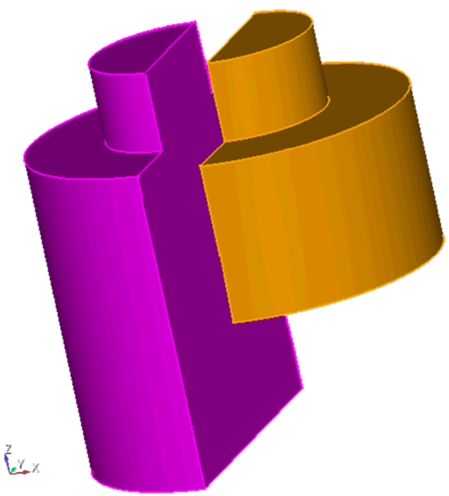
Midpoint subdivision

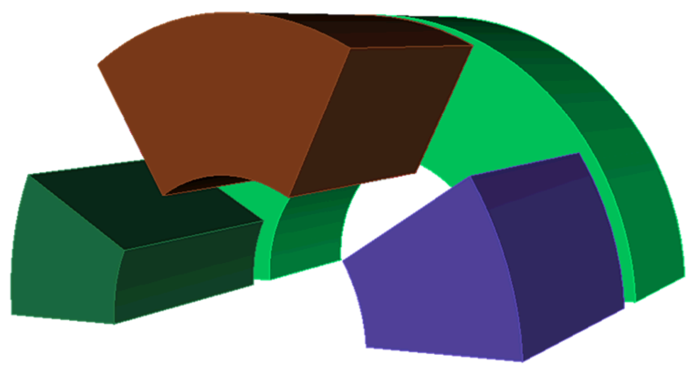






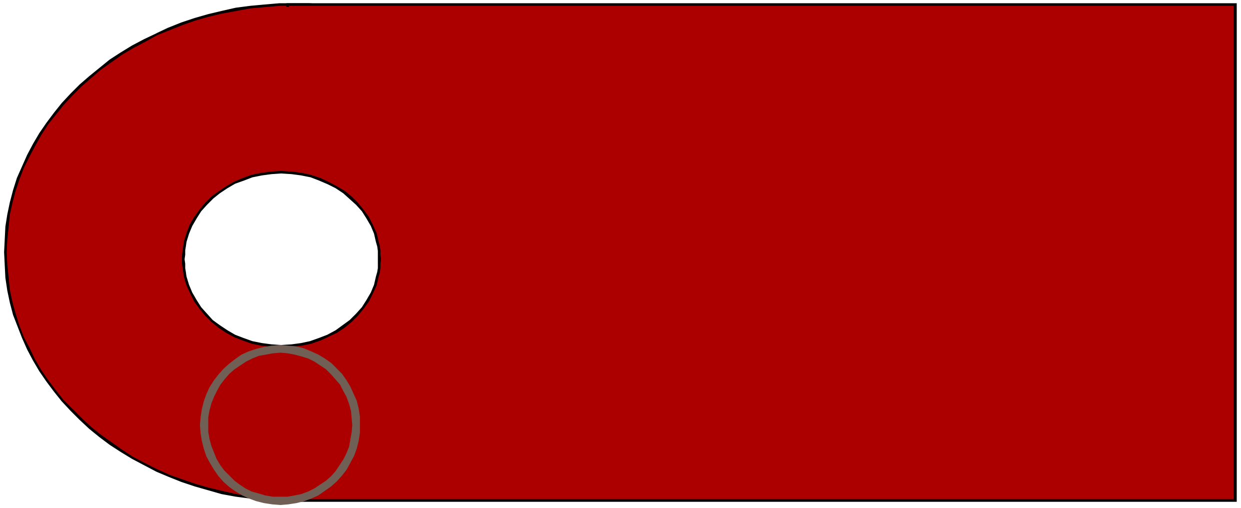
Many-to-many sweeping





Sub-mapping

Medial Axis



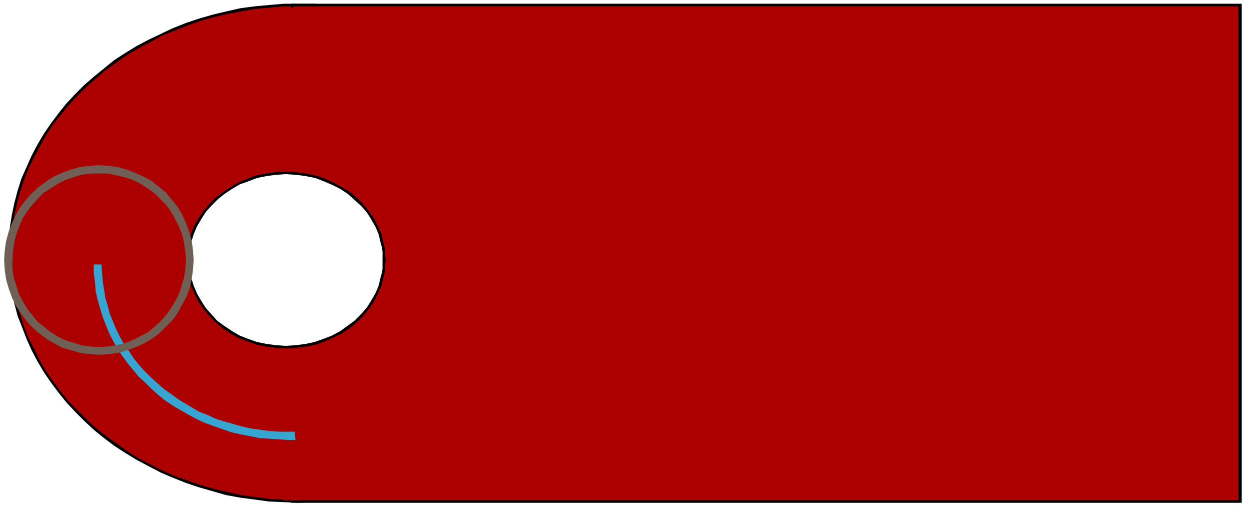
Medial Axis

•Medial Object - Roll a Maximal circle or sphere through the model. The center traces the medial object

•Medial Object used as a tool to automatically decompose model into simpler mapable or sweepable parts

(Price, 95;97)(Tam,91)

Medial Axis



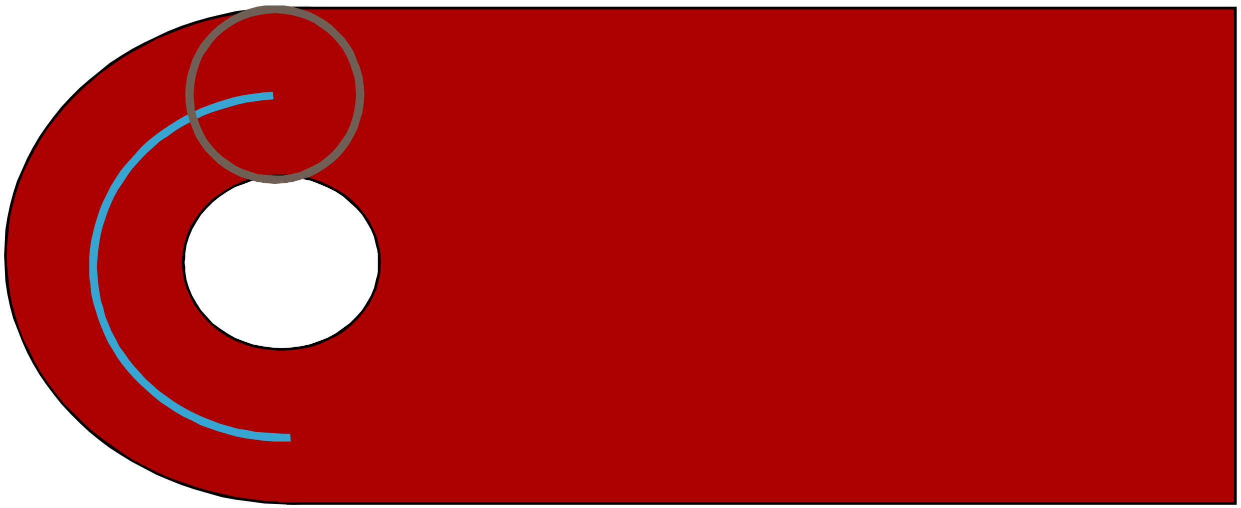
Medial Axis

•Medial Object - Roll a Maximal circle or sphere through the model. The center traces the medial object

•Medial Object used as a tool to automatically decompose model into simpler mapable or sweepable parts

(Price, 95;97)(Tam,91)

Medial Axis



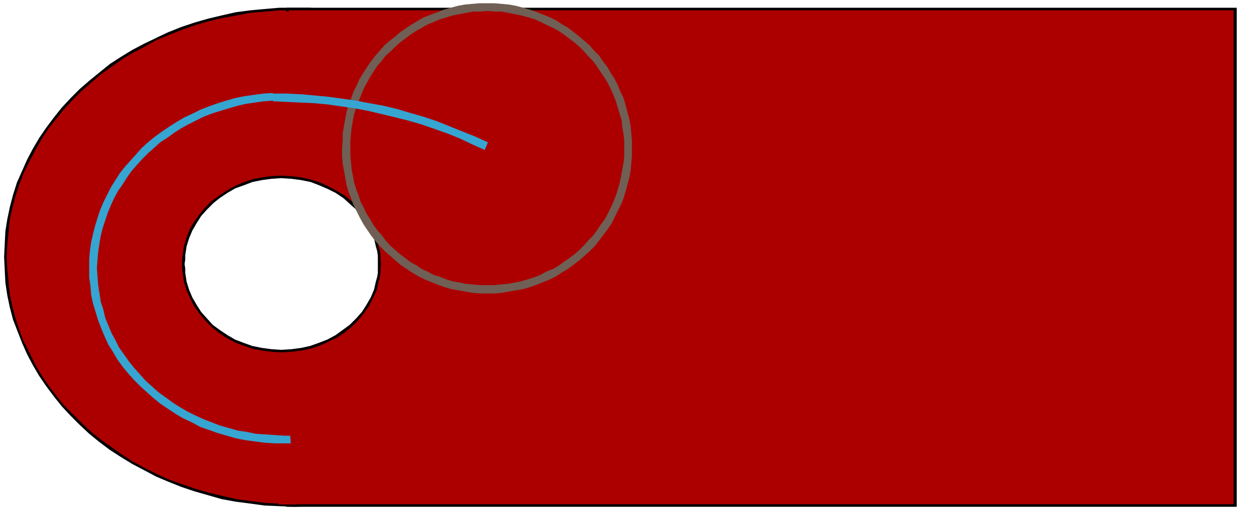
Medial Axis

•Medial Object - Roll a Maximal circle or sphere through the model. The center traces the medial object

•Medial Object used as a tool to automatically decompose model into simpler mapable or sweepable parts

(Price, 95;97)(Tam,91)

Medial Axis



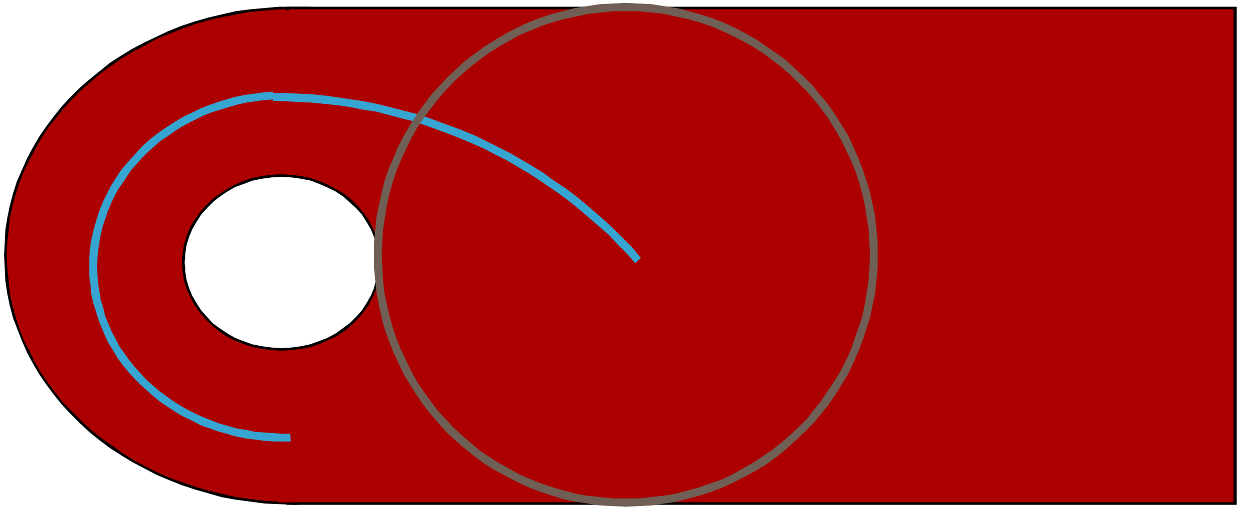
Medial Axis

•Medial Object - Roll a Maximal circle or sphere through the model. The center traces the medial object

•Medial Object used as a tool to automatically decompose model into simpler mapable or sweepable parts

(Price, 95;97)(Tam,91)

Medial Axis



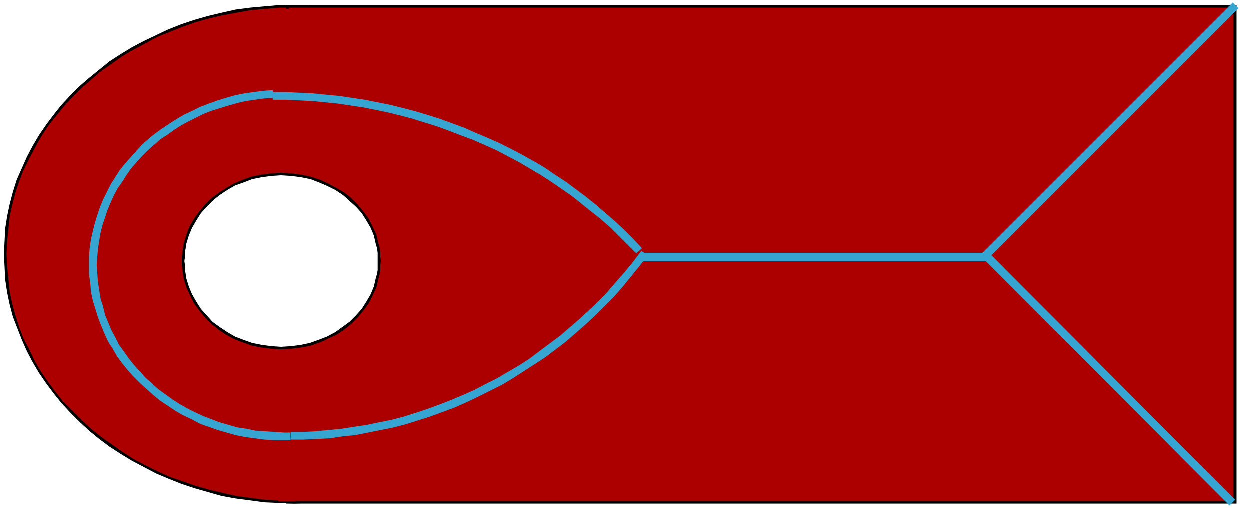
Medial Axis

•Medial Object - Roll a Maximal circle or sphere through the model. The center traces the medial object

•Medial Object used as a tool to automatically decompose model into simpler mapable or sweepable parts

(Price, 95;97)(Tam,91)

Medial Axis



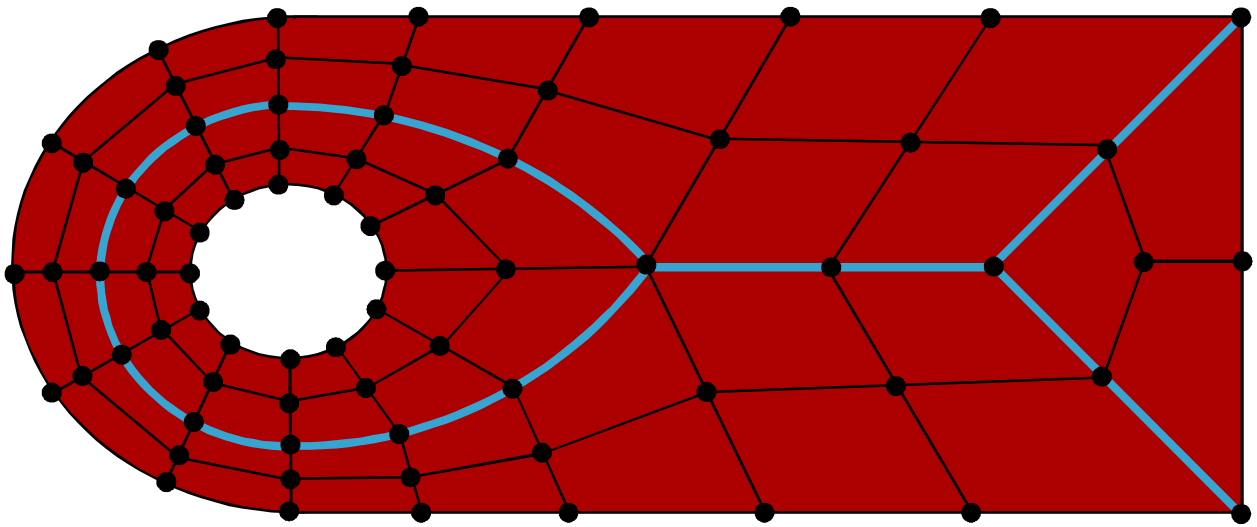
Medial Axis

•Medial Object - Roll a Maximal circle or sphere through the model. The center traces the medial object

•Medial Object used as a tool to automatically decompose model into simpler mapable or sweepable parts

(Price, 95;97)(Tam,91)

Medial Axis



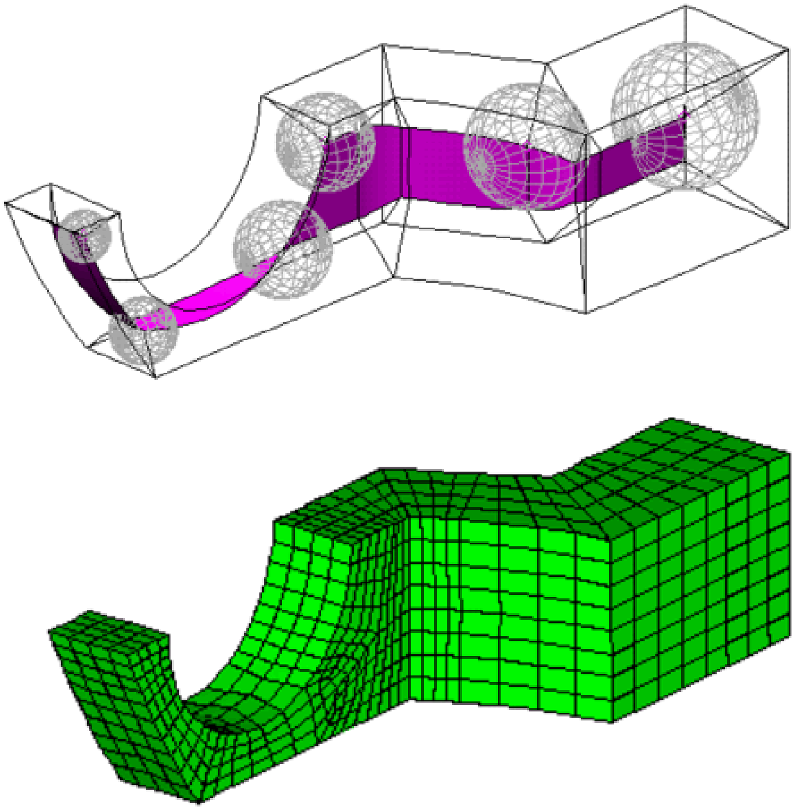
Medial Axis

•Medial Object - Roll a Maximal circle or sphere through the model. The center traces the medial object

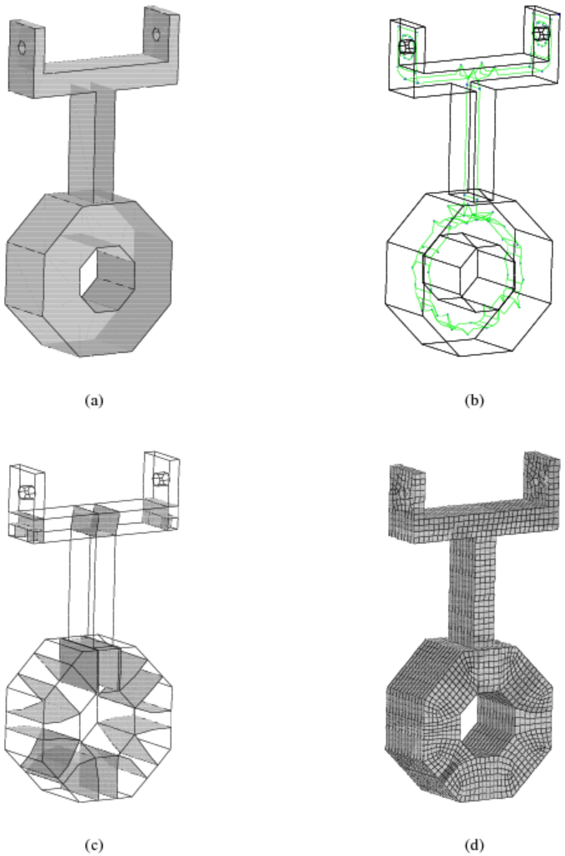
•Medial Object used as a tool to automatically decompose model into simpler mapable or sweepable parts

(Price, 95;97)(Tam,91)

Medial Axis



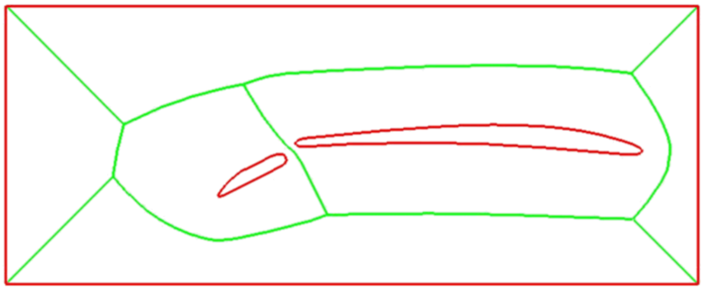
Medial Axis + Midpoint Subdivision
(Price, 95)



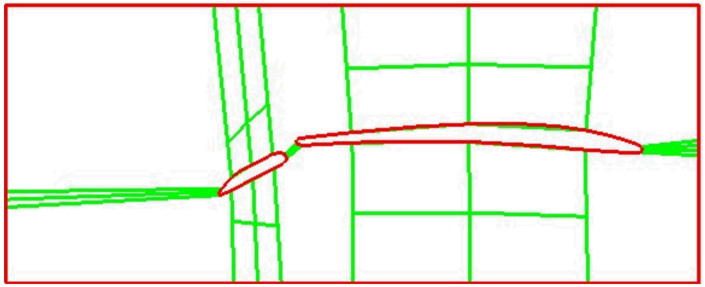
Embedded Voronoi Graph
(Sheffer, 98)

Block decomposition using Medial Axis

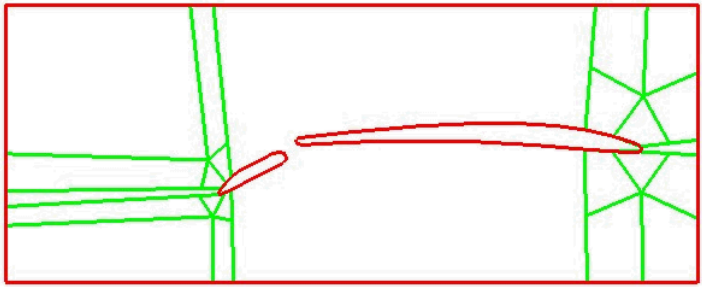
Cecil Armstrong – Queens University, Belfast



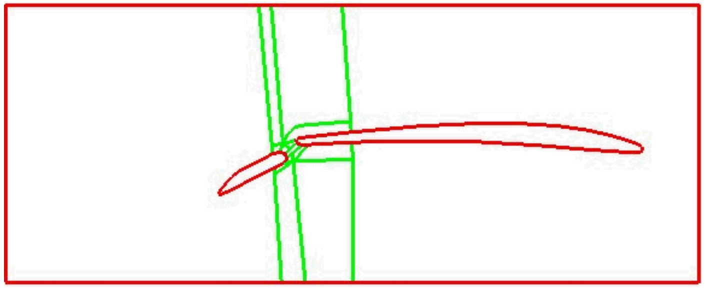
Boundary with medial axis



Four sides sub-regions
(between boundary entities in proximity)



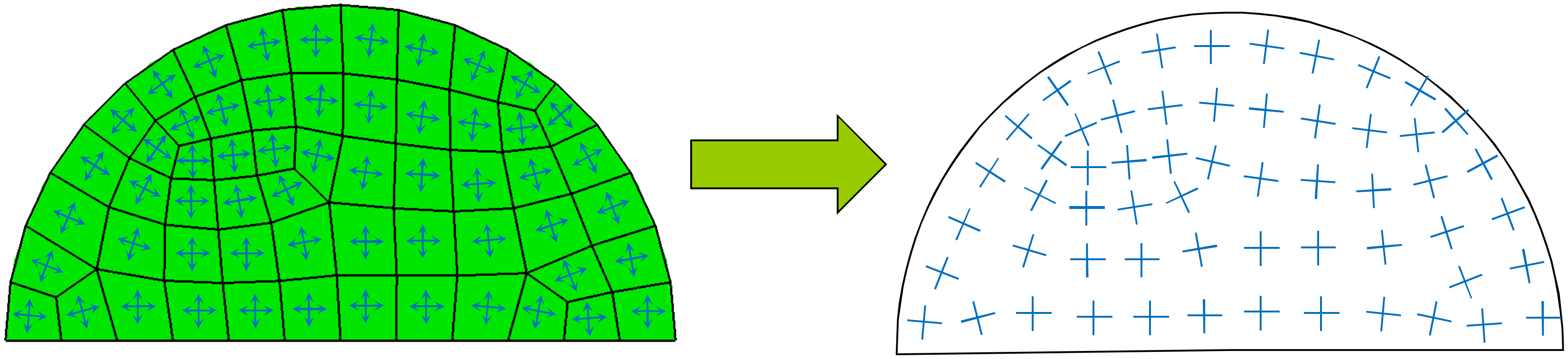
Five sided sub-regions



Six sided sub-regions

Frame Fields

Each quad in a mesh has 2 inherent directions. We can extract a field of local direction frames from any quad mesh.



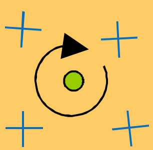
A mesh generated with Paving

Reference: "Kowalski, Ledoux, Frey, "A PDE based approach to multi domain partitioning and quadrilateral meshing," IMR21, 2012.

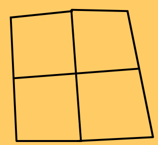
Properties of Frame Fields

Property 1:
There are singularities defined at the zeros of the piecewise linear interpolation of the frame fields.

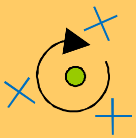
Case 1: Full 360° turn



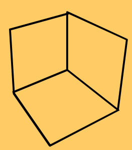
No singularity



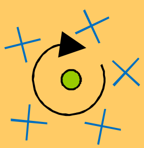
Case 2: 270° turn



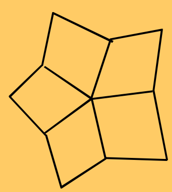
Singularity 3



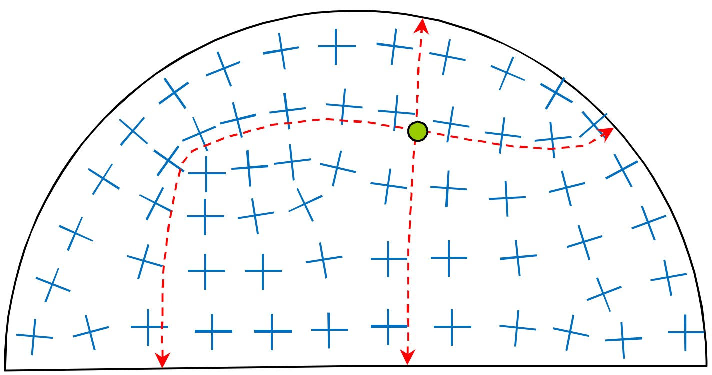
Case 3: 450° turn



Singularity 5



Property 2:
You can trace streamlines by interpolating the frames. In general Streamlines proceed in 2 directions from any point.



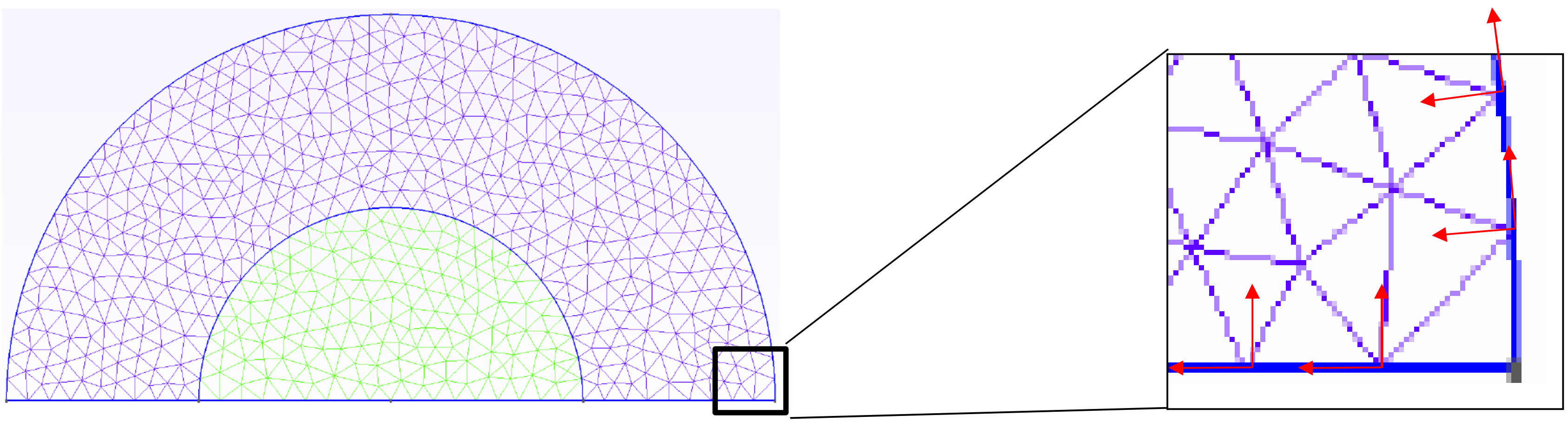
Steamlines that end at singularities are called separatrices.

Reference: "Kowalski, Ledoux, Frey, "A PDE based approach to multidomain partitioning and quadrilateral meshing," IMR21, 2012.

Can we first generate a frame field and then generate a quad mesh from it?

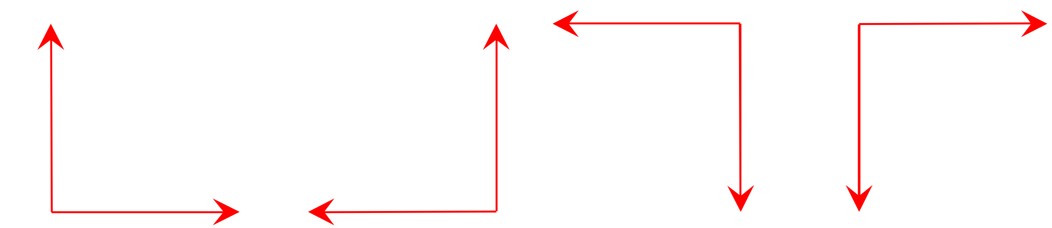
Yes, Kowalski (IMR21) builds a frame field by solving a PDE (Laplace) with Dirichlet BCs over a triangle mesh.

For BCs, normals and tangents are computed for each boundary node.



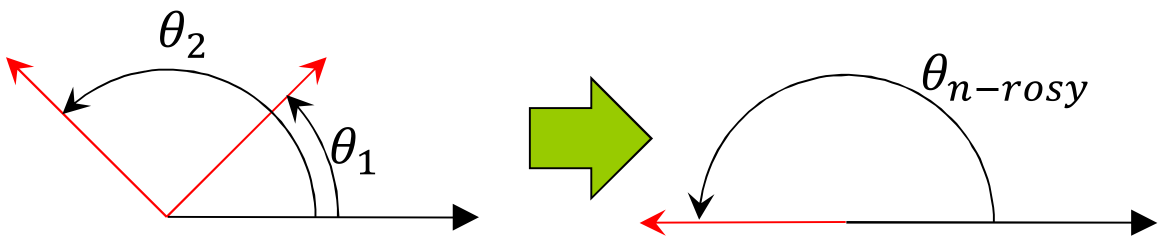
N-Rosy Definition

90° rotations of a frame are equivalent



Convert any planar coordinate frame into a single planar vector.

$$Given\colon\ (\theta_1, \theta_2)$$

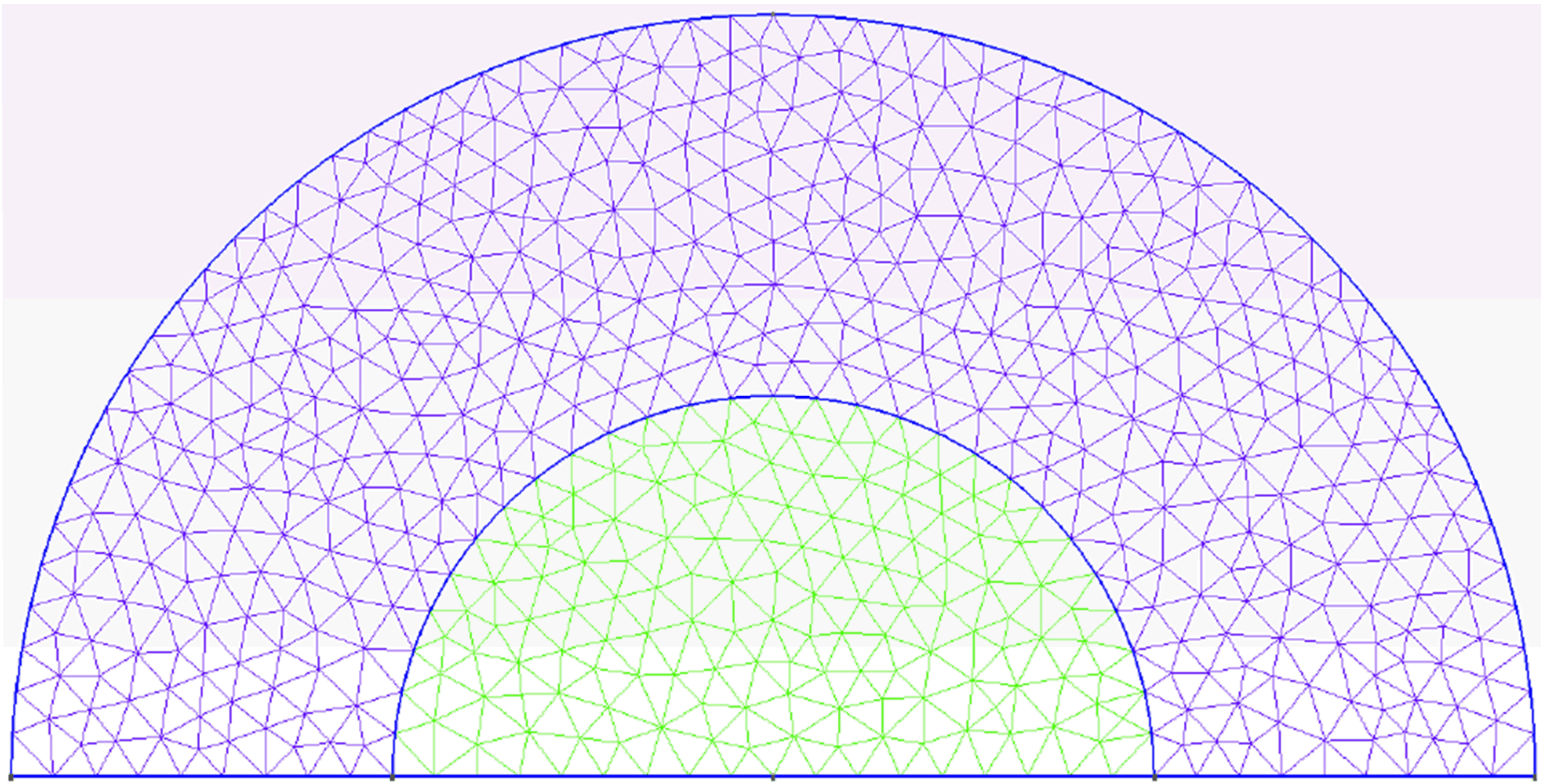


$$\theta_{n-rosy} = 4 * \theta_1 \,\%\, 360 = 4 * \theta_2 \,\%\, 360$$

$$\theta_1 = \frac{\theta_{n-rosy}}{4}$$

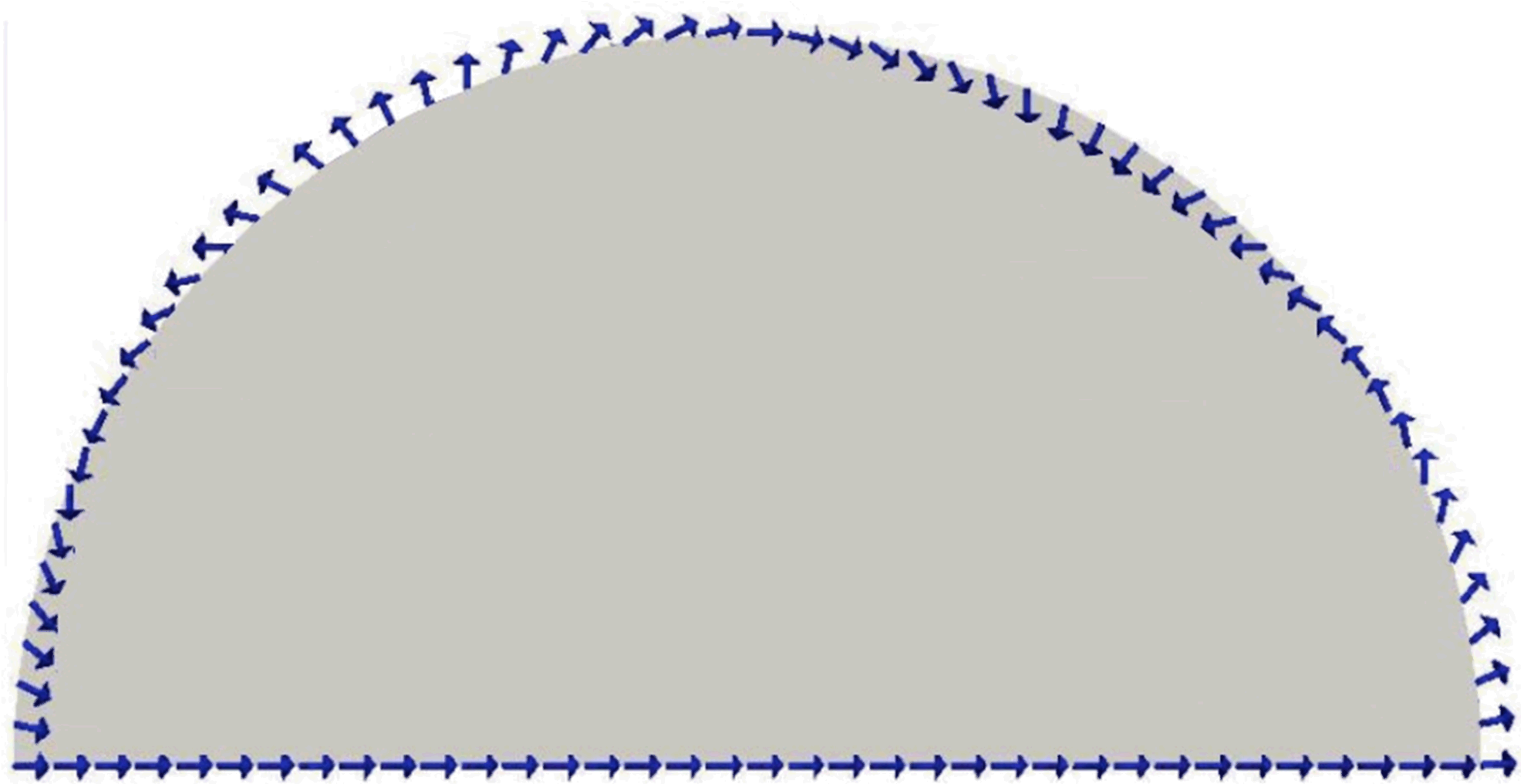
$$\theta_2 = \theta_1 + 90°$$

2D Frame Fields



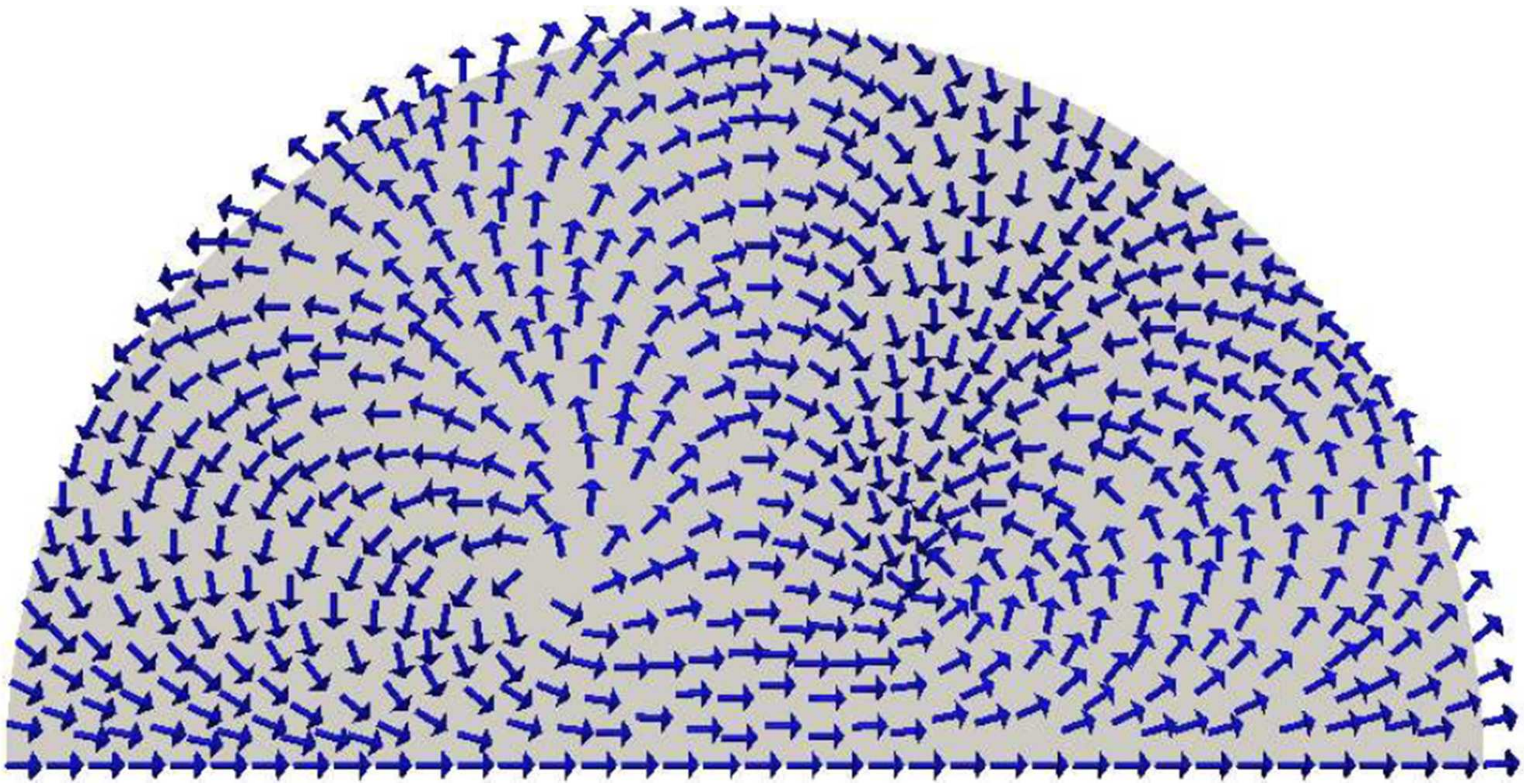
STEP 1: Build a planar tri mesh on the surface. Compute BCS: n-rosy from tangent/normal at each boundary node.

2D Frame Fields



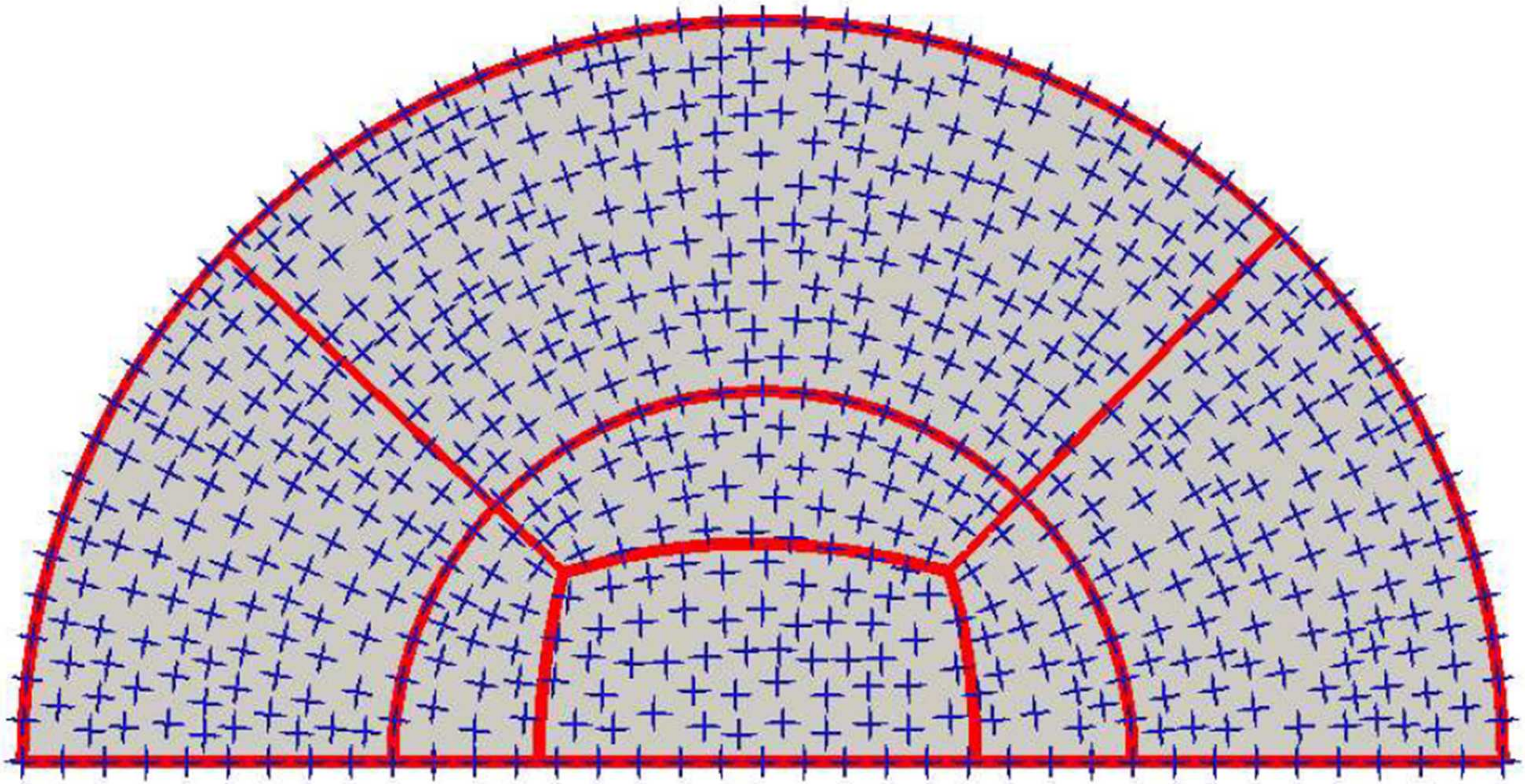
STEP 2: Compute BCS: n-rosy from tangent/normal at each boundary node.

2D Frame Fields



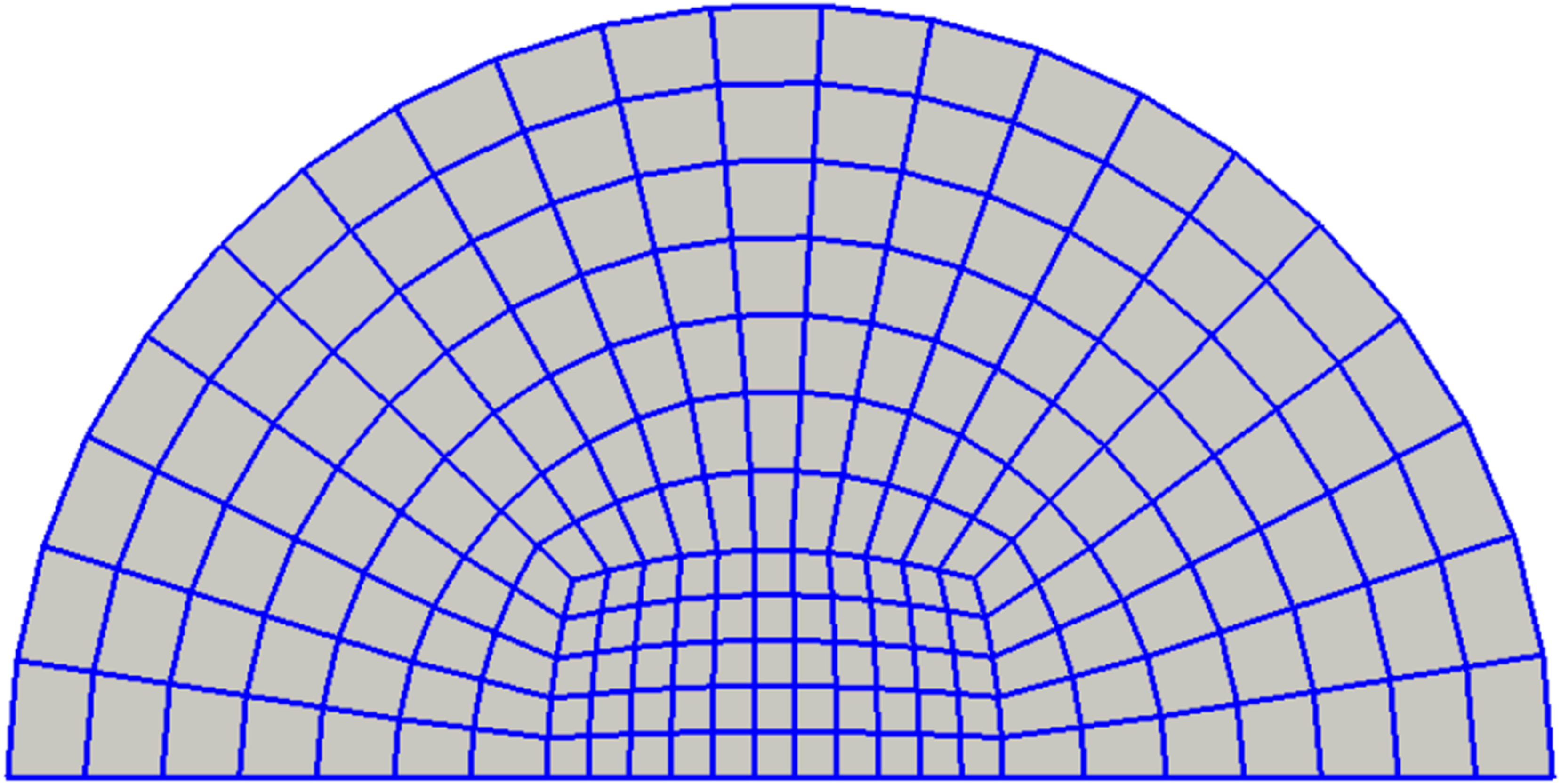
STEP 3: Solve Laplace equation.

2D Frame Fields



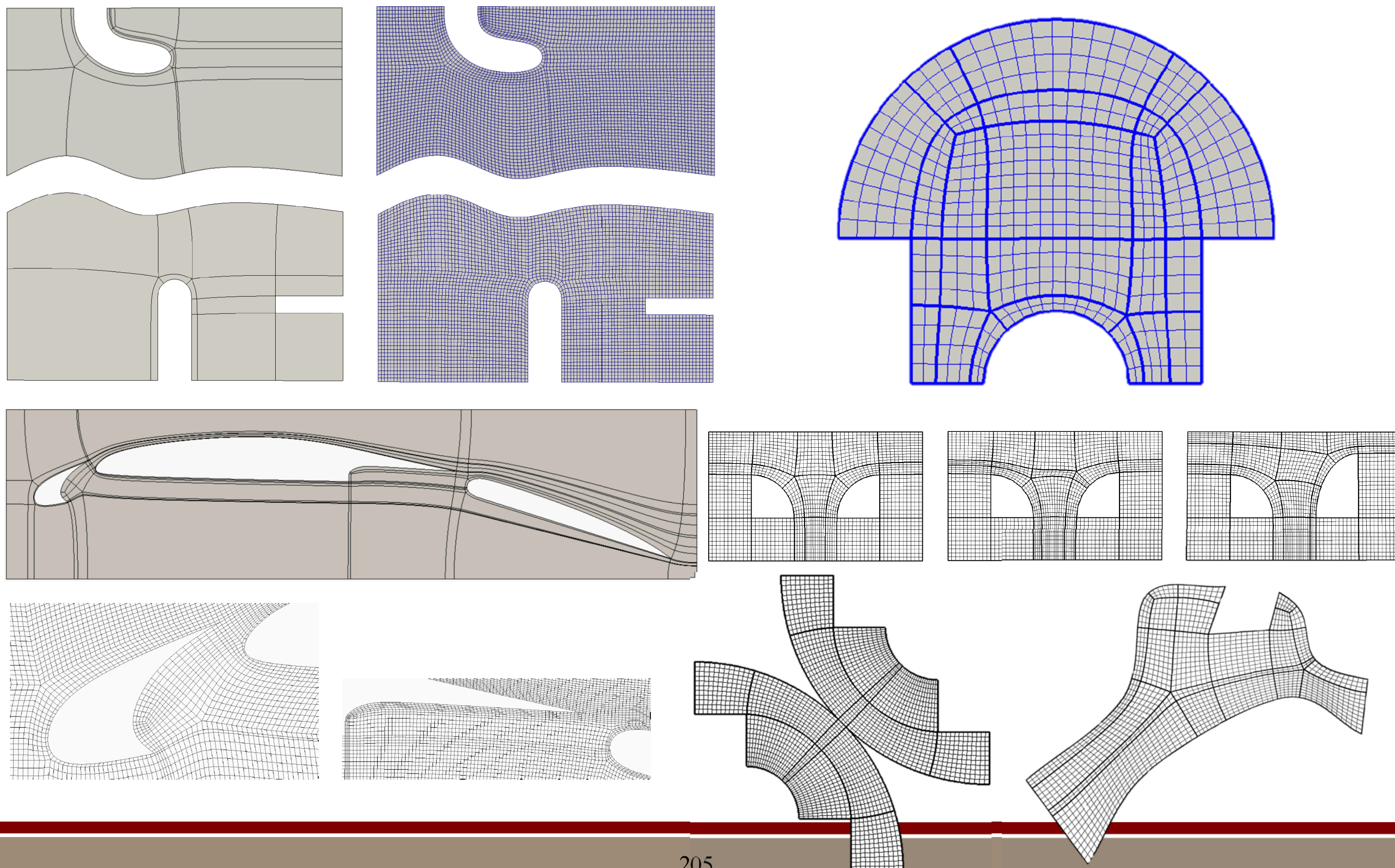
STEP 4: Convert back to frames, compute singular points, trace separatrices, yielding a block decomposition.

2D Frame Fields

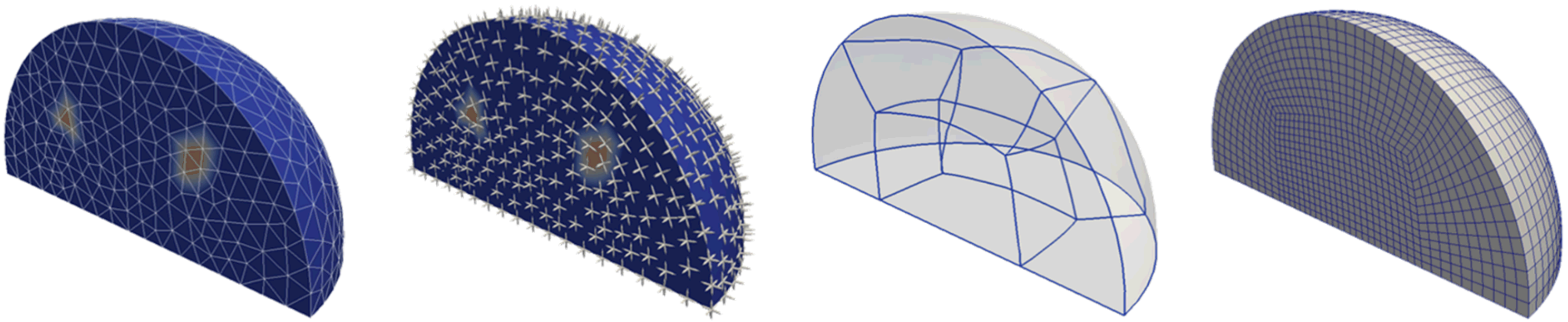


STEP 5: Map each block decomposition.

Frame Field Examples



Volumetric Frame Fields



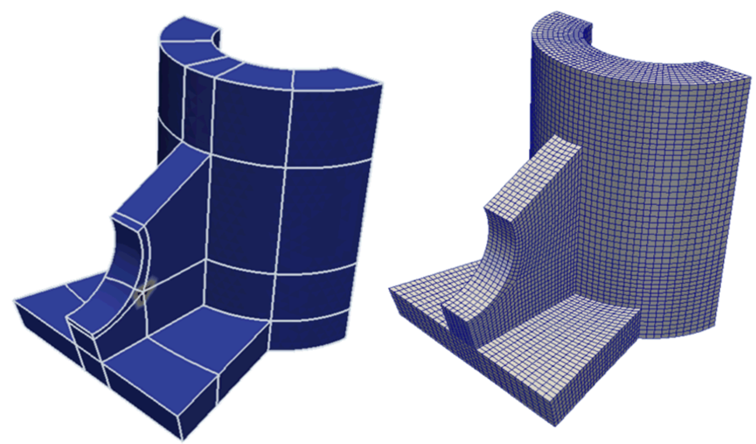
(a) Initial Tet Mesh

(b) Frame field generated over Tet Mesh, via optimization approach

(c) Singularity graph extracted defining block structure

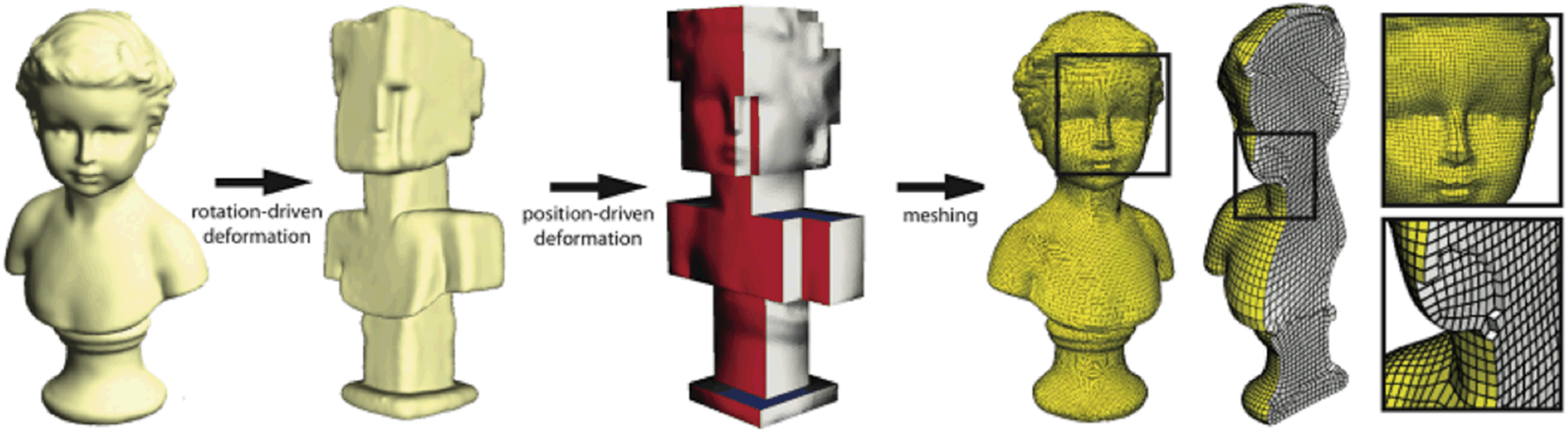
(d) Mesh generated on blocks

N. Kowalski, F. Ledoux, P. Frey, “Block Structured Hexahedral Meshes for CAD Models using 3D Frame Fields”, International Meshing Roundtable 2014



Volumetric Parameterizations

PolyCube

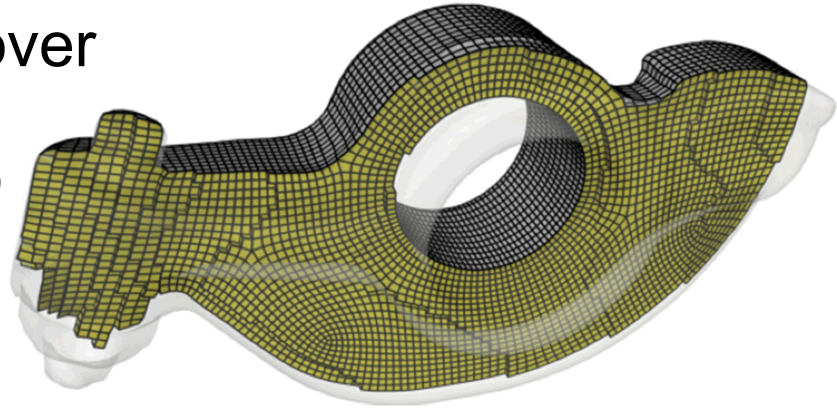


- J. Gregson, A. Sheffer, E. Zhang, “All-hex Mesh Generation via Volumetric PolyCube Deformation,” Eurographics Symposium on Geometry Processing, July 2011

Volumetric Parameterizations

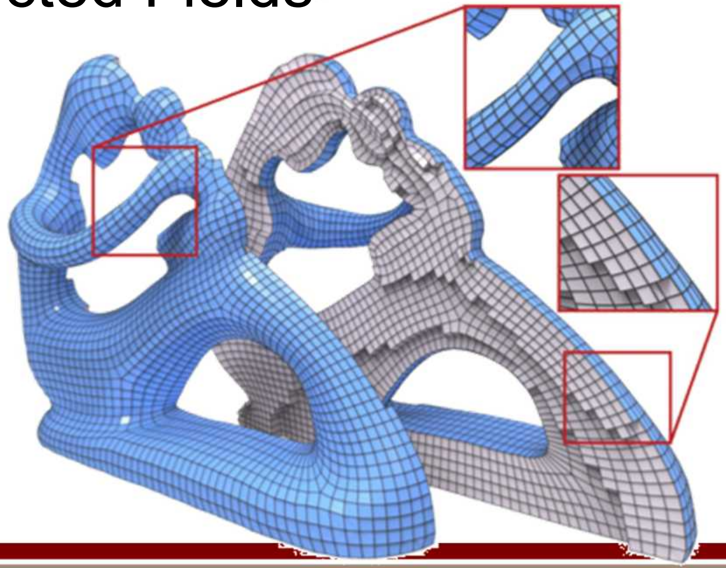
CubitCover

- M. Nieser, U. Reitebuch, K. Polthier, "CubeCover – Parameterizations of 3D Volumes", Eurographics Symposium on Geometry Processing 2011



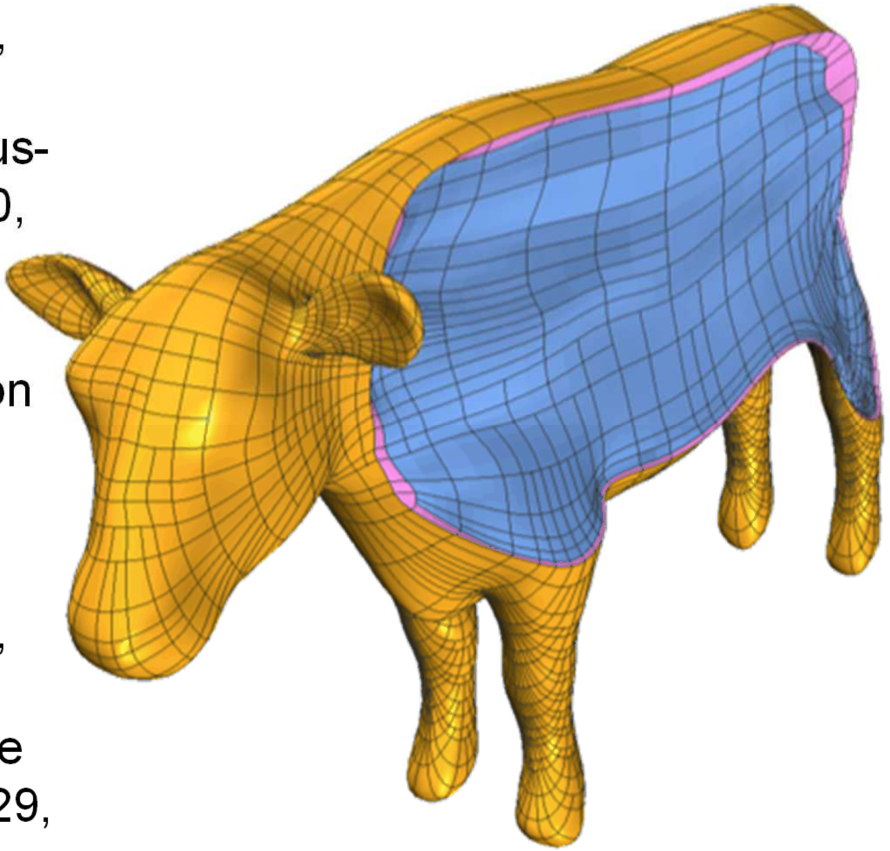
Singularity Restricted Fields

- Y. Li, Y. Liu, W. Xu, W. Wang, B. Guo, "All-Hex Meshing using Singularity-Restricted Field," Microsoft Research Asia, The University of Hong Kong



Hex Meshing and IsoGeometrics

- Direct Generation of T-Splines
  - Y. Zhang, W. Wang, T.J.R. Hughes, “Solid T-Spline Construction from Boundary Representations for Genus-Zero Geometry,” ICES Report 11-40, November 2011
  - W. Wang, Y. Zhang, L. Liu, T.J.R. Hughes, “Solid T-Spline Construction from Boundary Triangulations with Arbitrary Genus Topology”, ICES Report 12-13, April 2012
  - Y. Zhang, W. Wang, T.J.R. Hughes, “Conformal Solid T-Spline Construction from Boundary T-spline Representations,” ICES Report 12-29, July 2012
  - And others…



Hybrid Methods

CFD Meshing

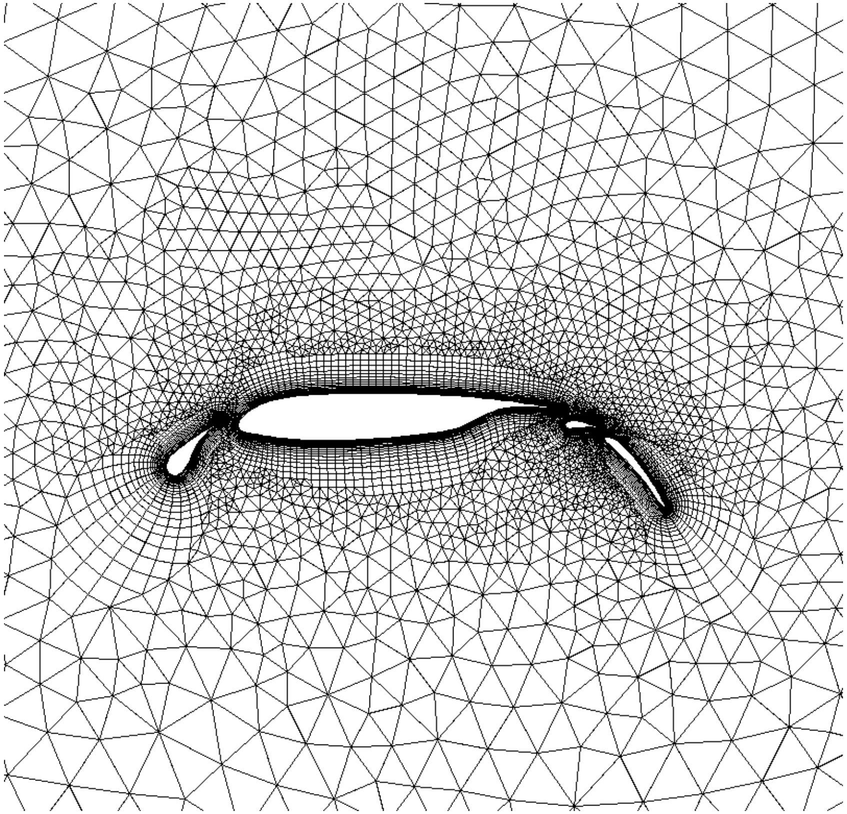


Image courtesy of Roy P. Koomullil, Engineering Research Center, Mississippi State University, http://www.erc.msstate.edu/~roy/

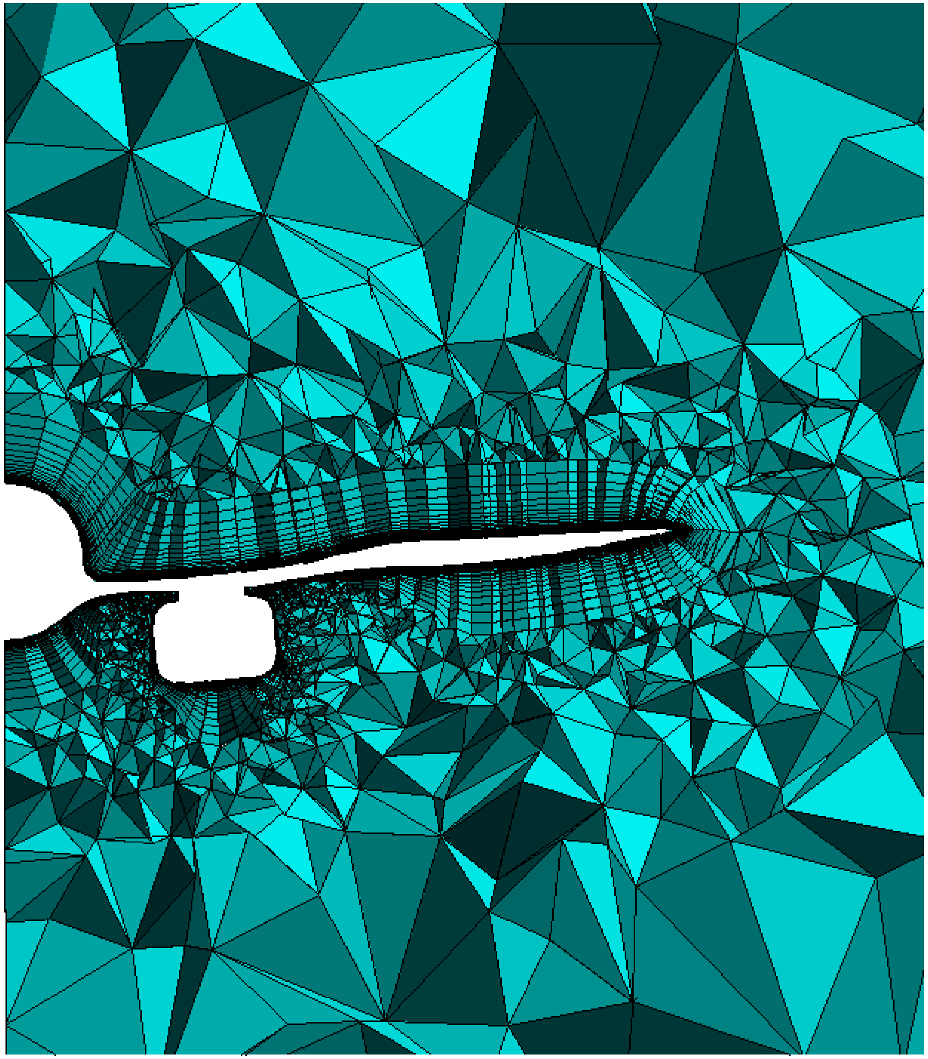
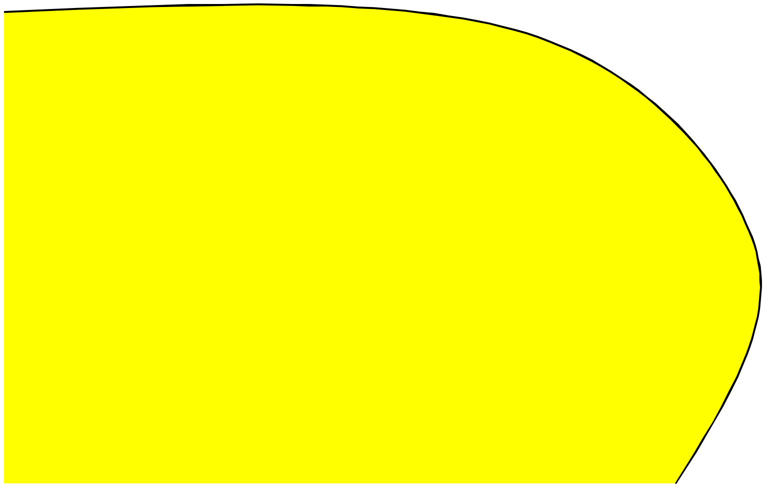


Image courtesy of acelab, University of Texas, Austin, http://acelab.ae.utexas.edu

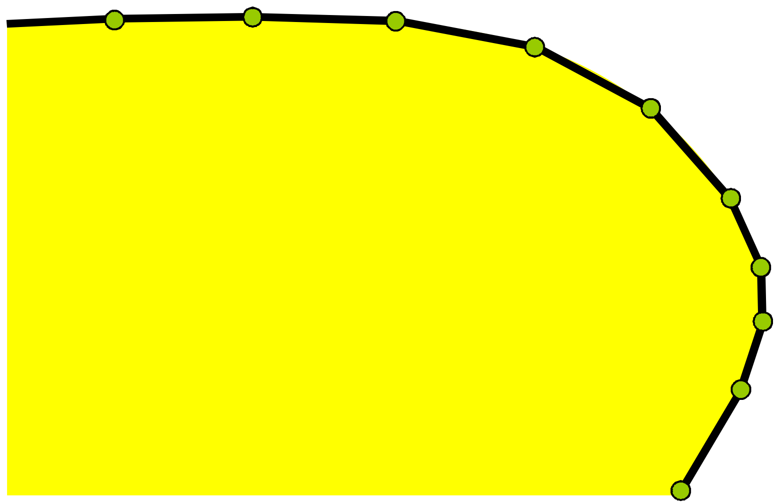
Hybrid Methods

Advancing Layers Method



Hybrid Methods

Advancing Layers Method

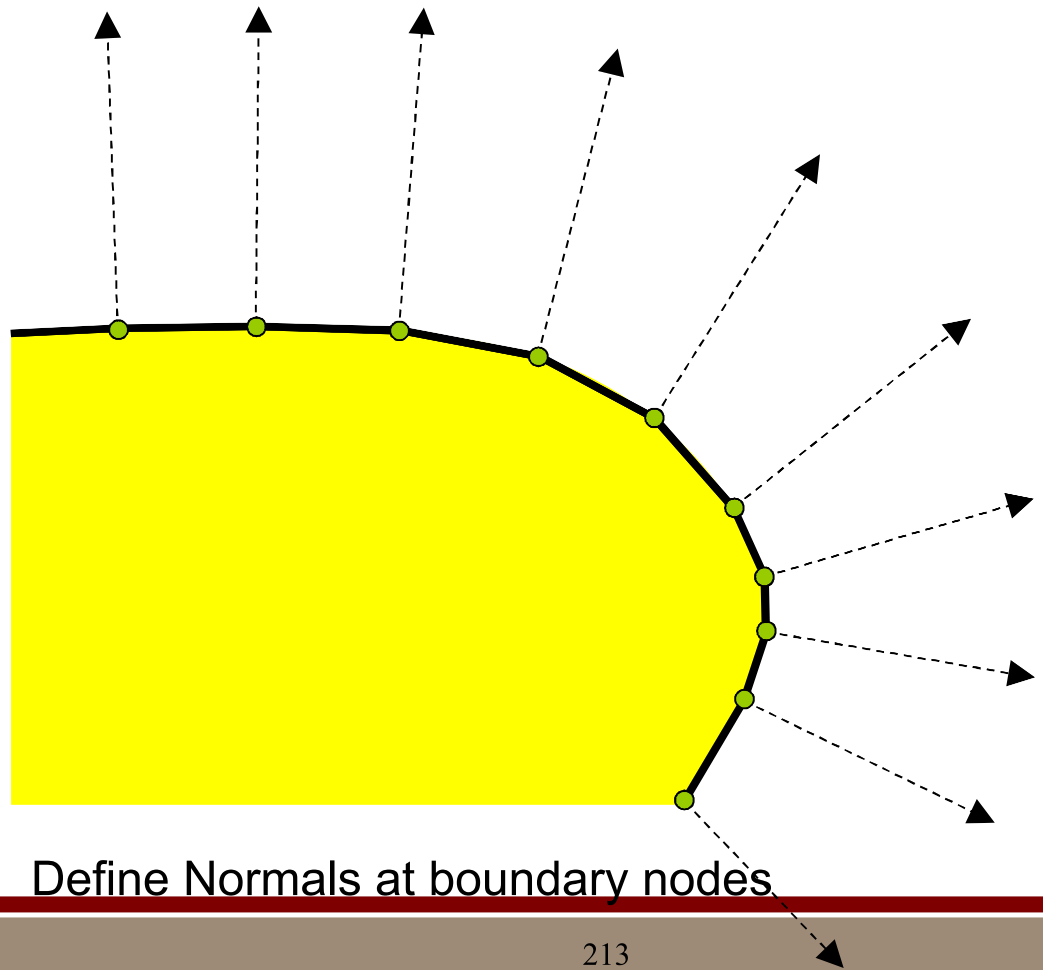


Discretize Boundary

Hybrid Methods

Advancing Layers Method

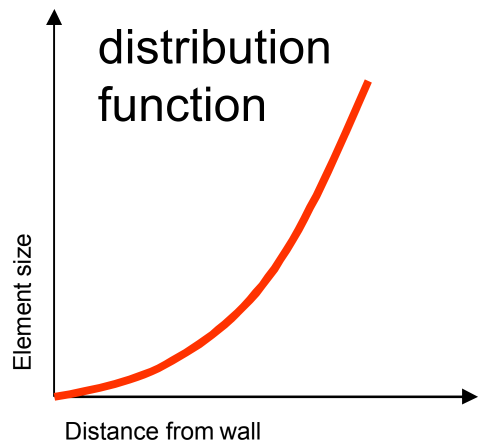
(Pirzadeh, 1994)

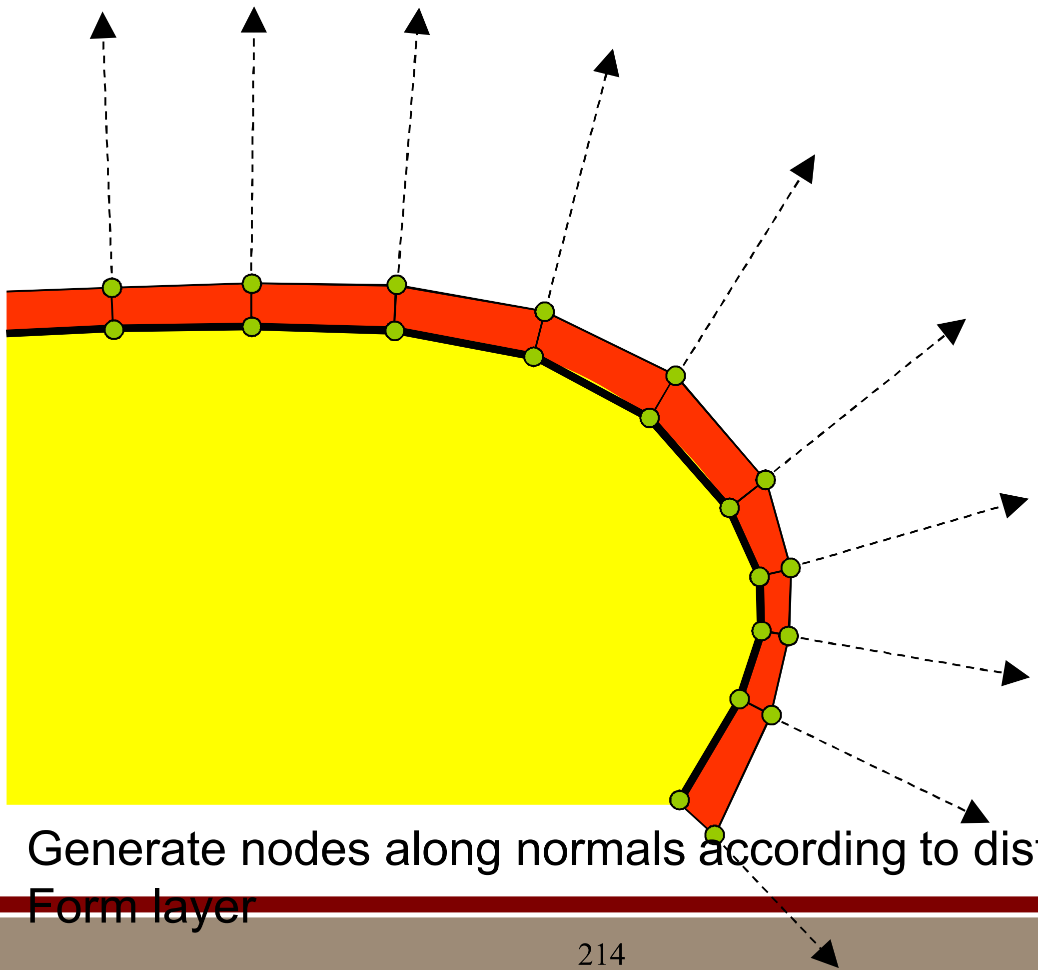


Define Normals at boundary nodes

Hybrid Methods

Advancing Layers Method



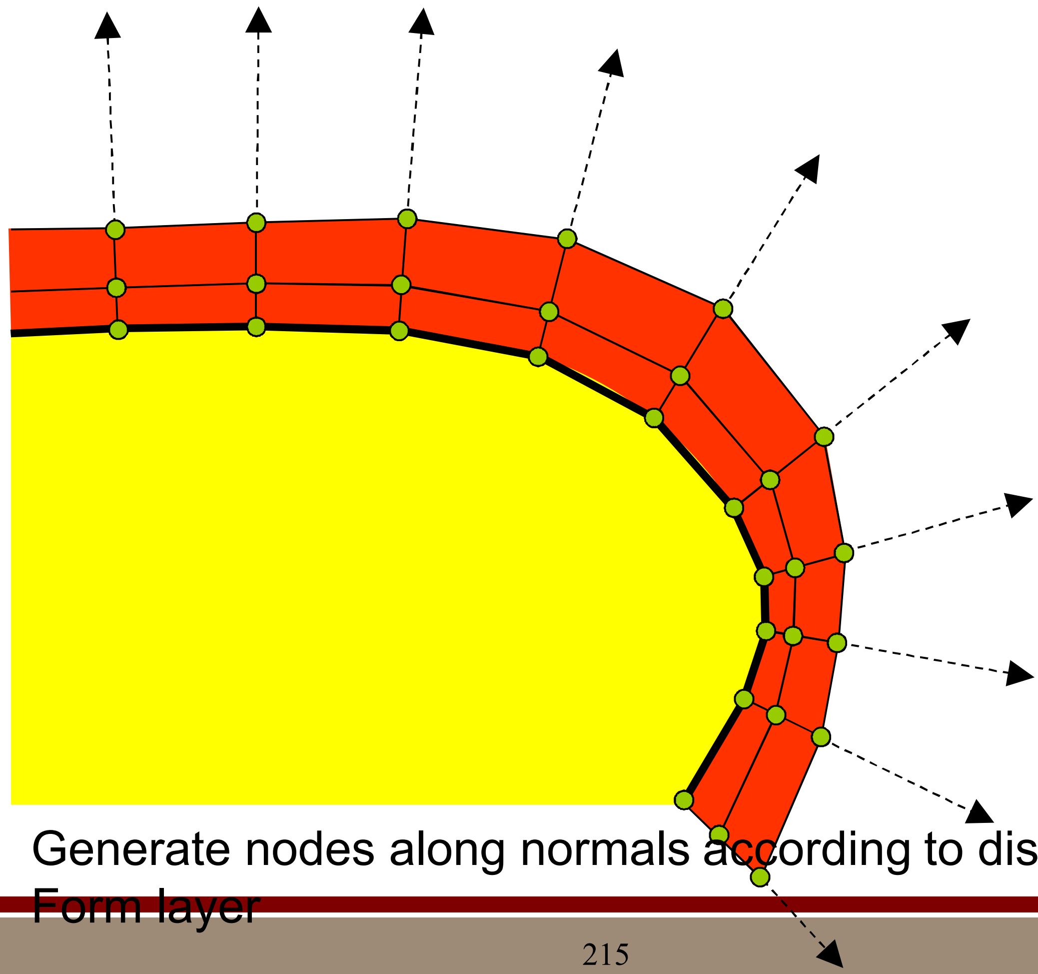


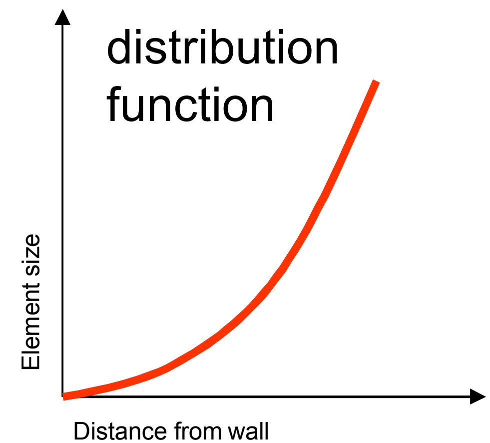
Generate nodes along normals according to distribution function

Form layer

Hybrid Methods

Advancing Layers Method

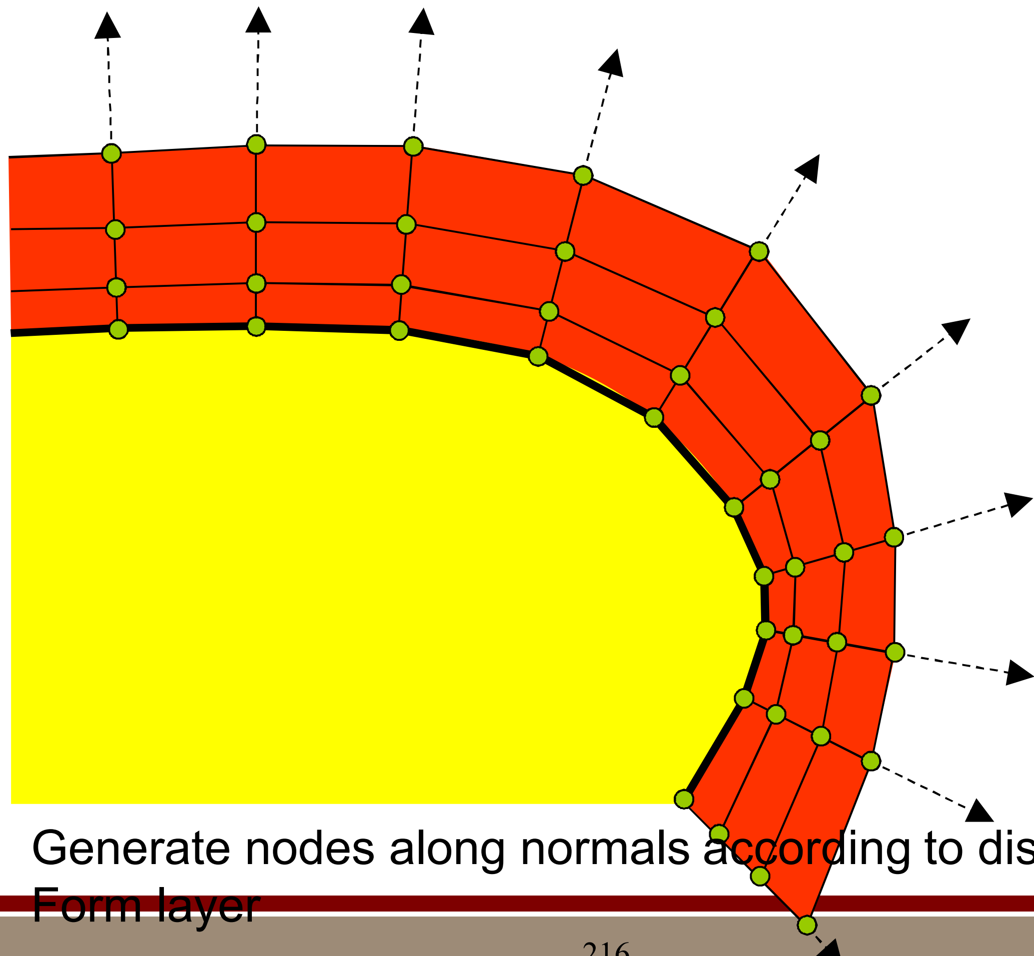


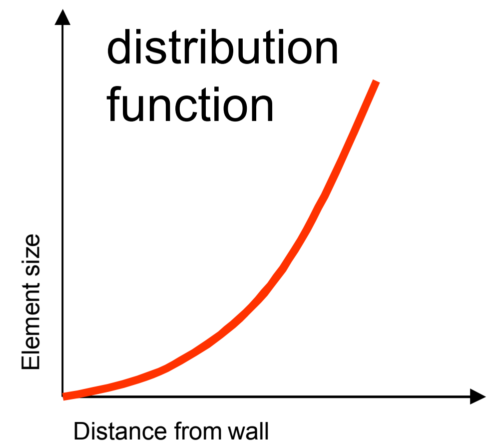


Generate nodes along normals according to distribution function
Form layer

Hybrid Methods

Advancing Layers Method

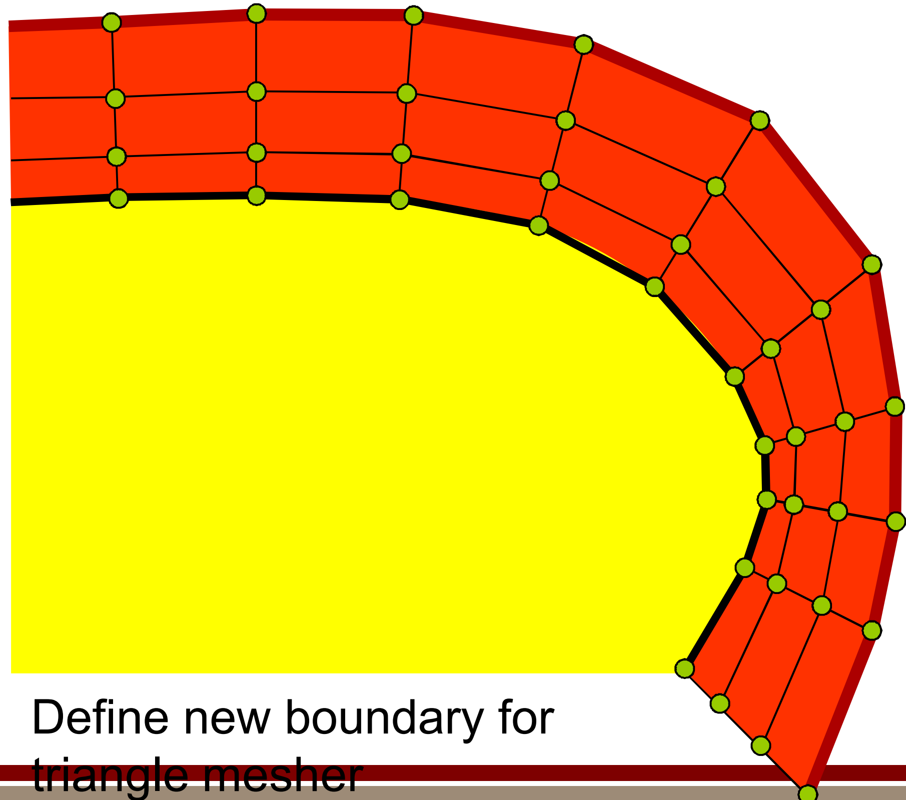




Generate nodes along normals according to distribution function
Form layer

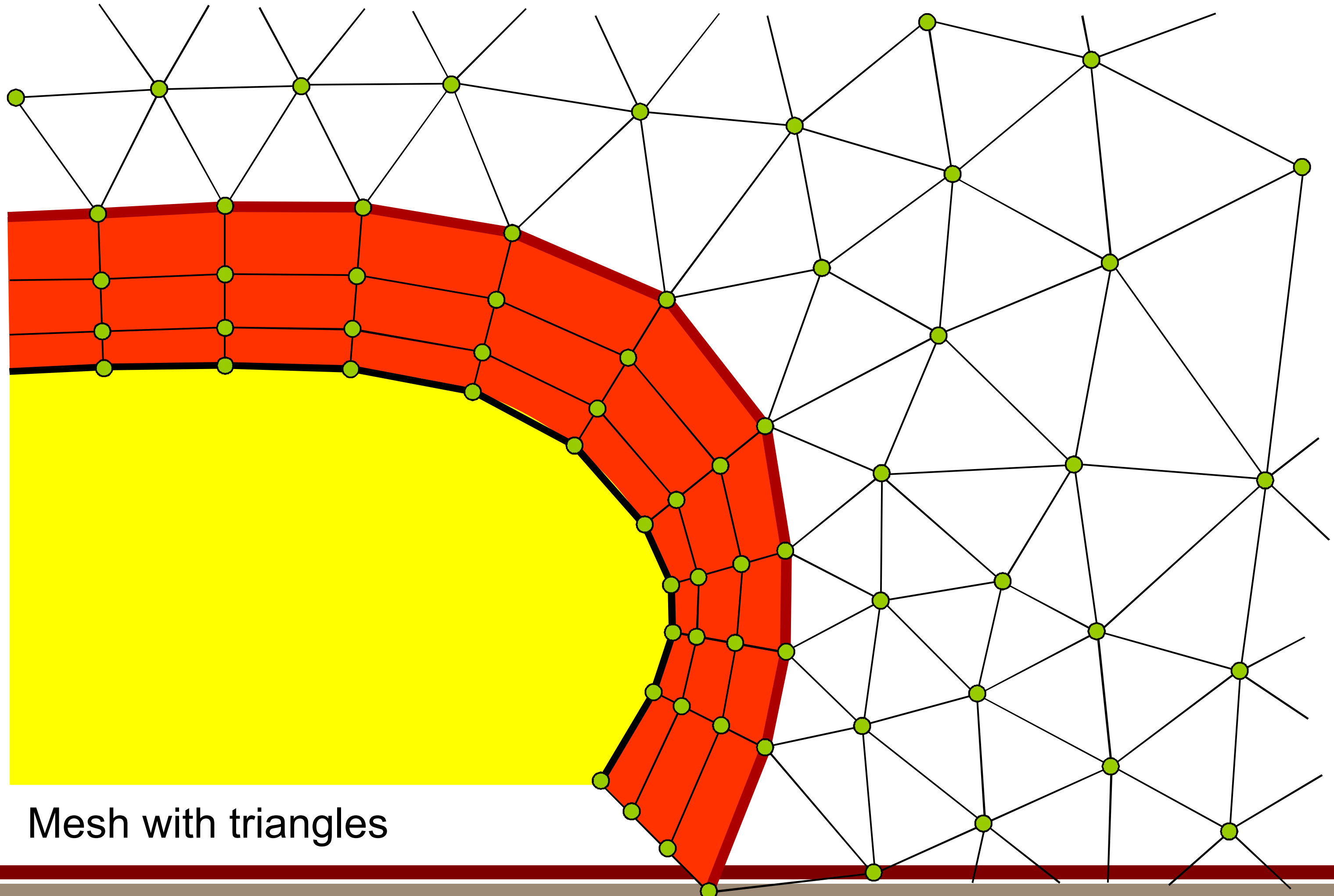
Hybrid Methods

Advancing Layers Method



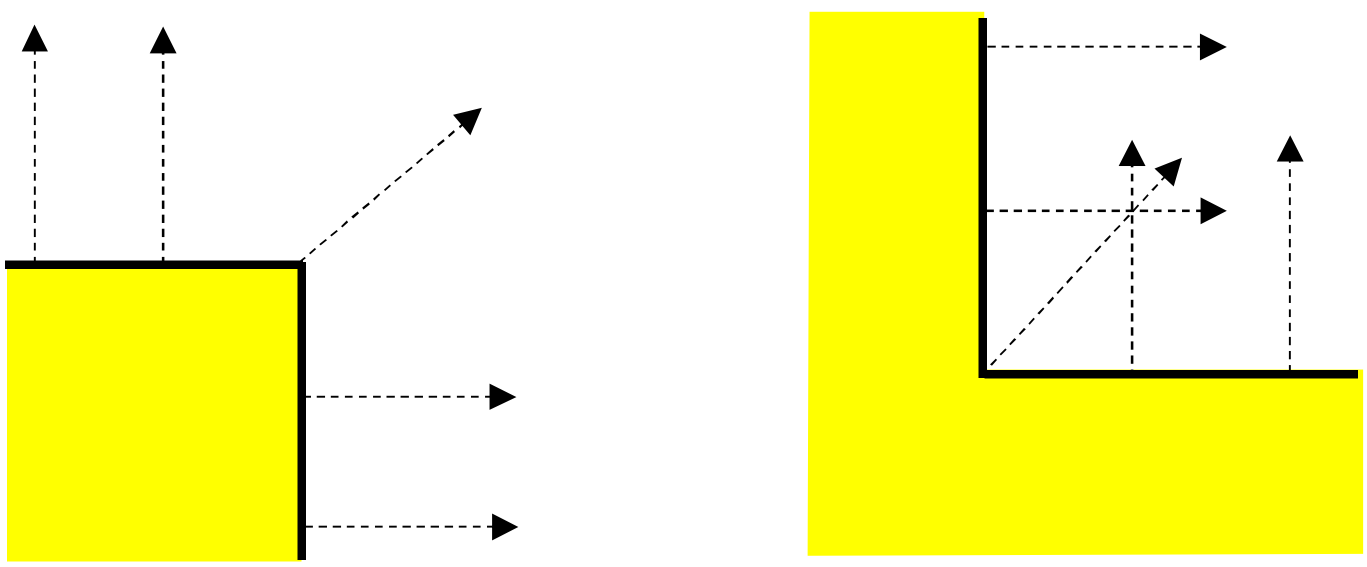
Define new boundary for triangle mesher

Hybrid Methods



Mesh with triangles

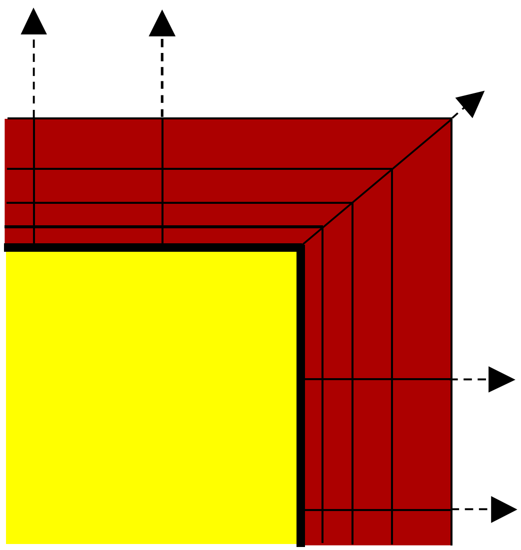
Hybrid Methods



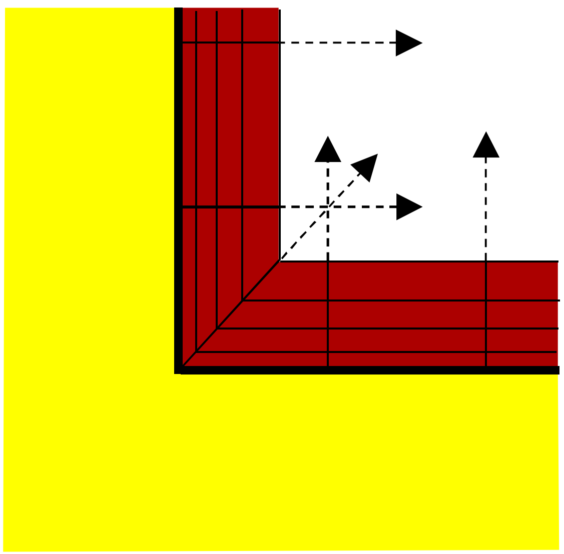
Convex Corner

Concave Corner

Hybrid Methods

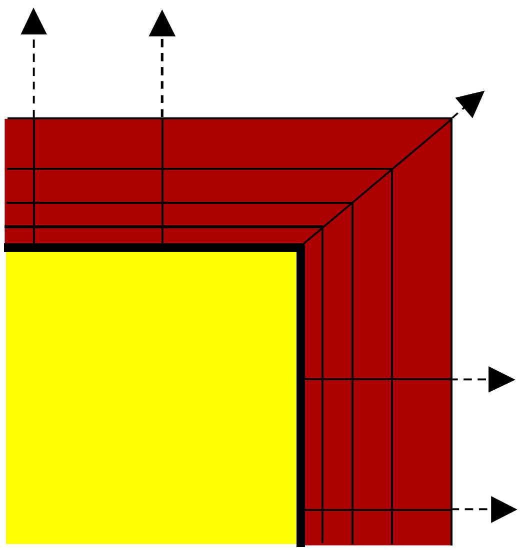


Convex Corner

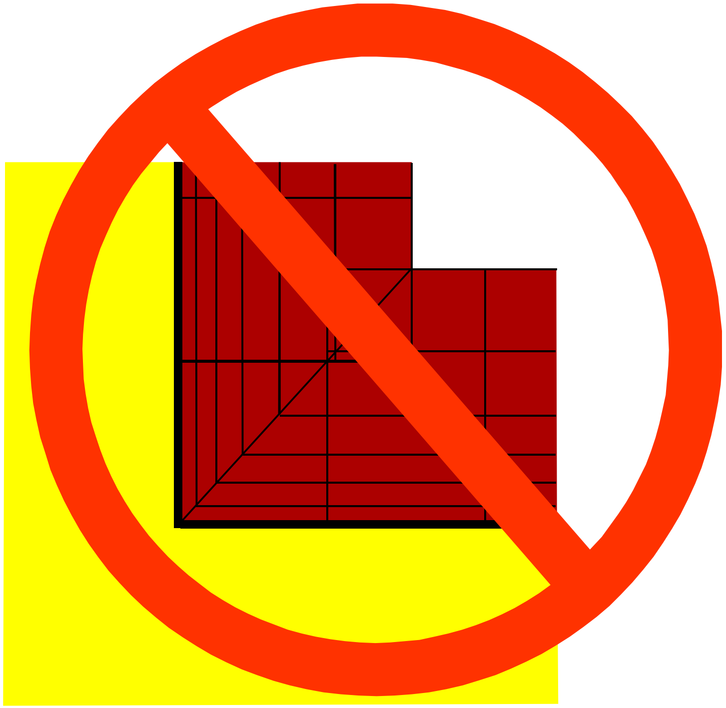


Concave Corner

Hybrid Methods

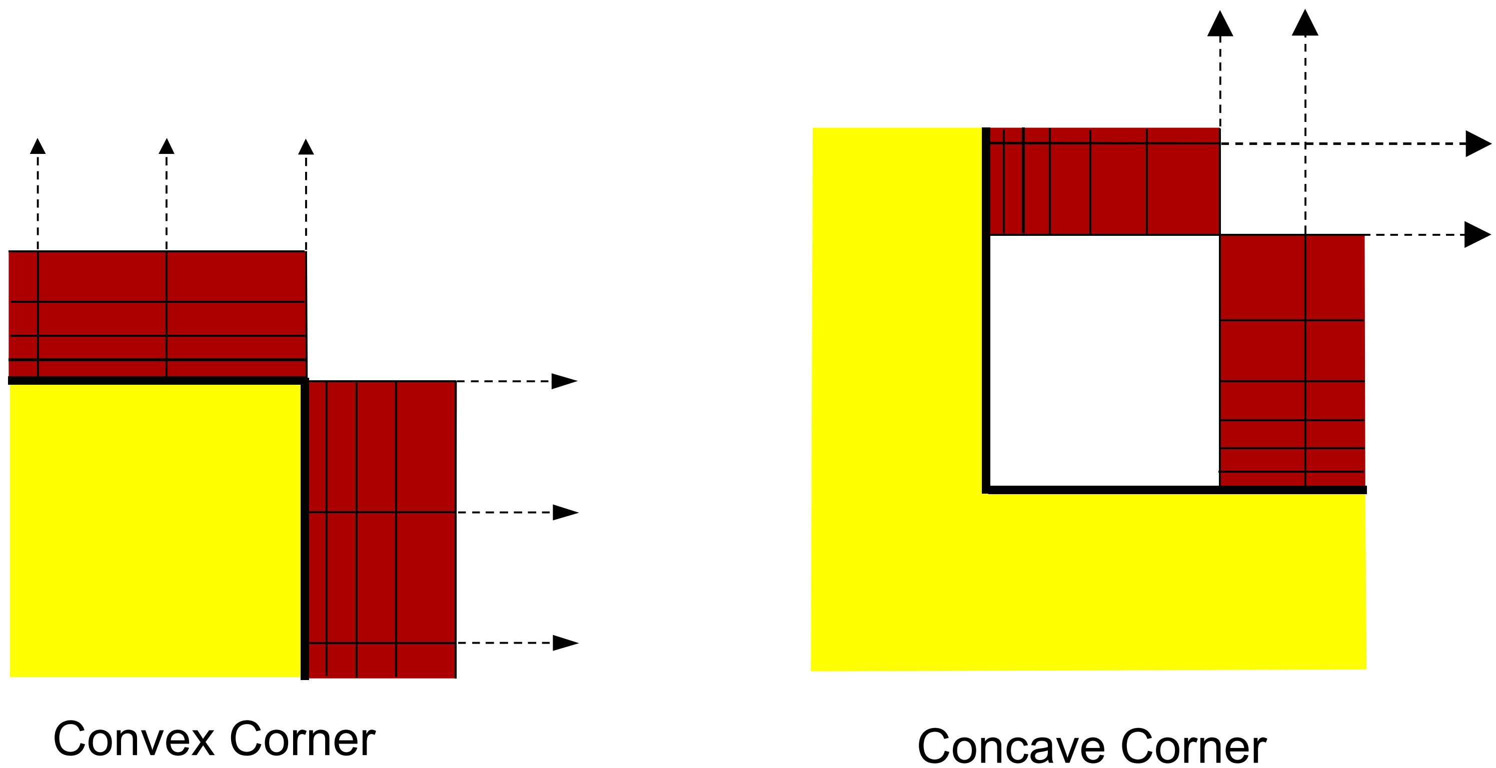


Convex Corner



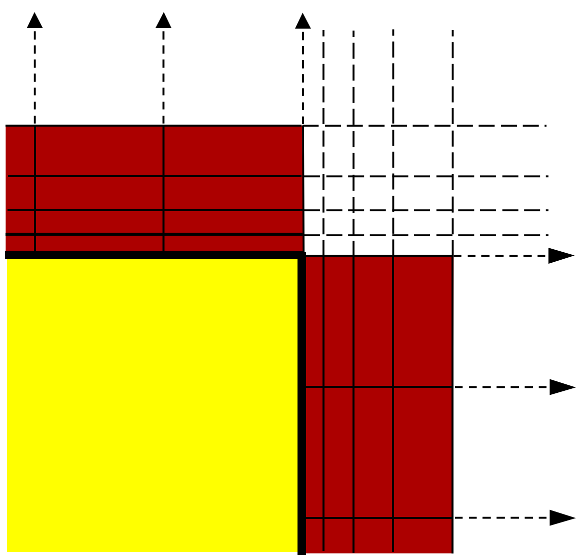
Concave Corner

Hybrid Methods

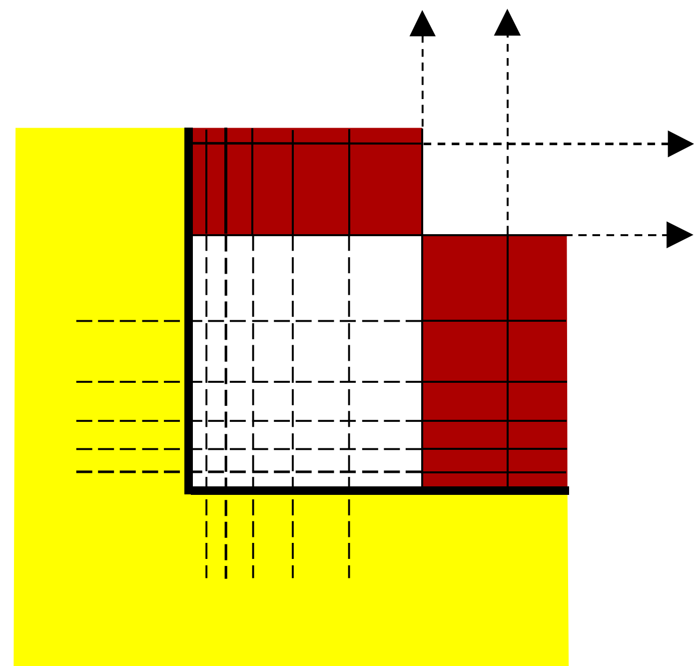


Blend
Regions

Hybrid Meshes



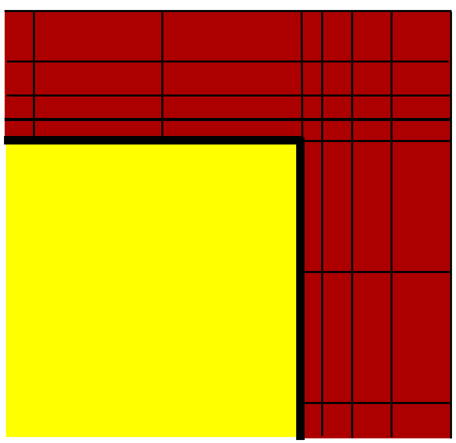
Convex Corner



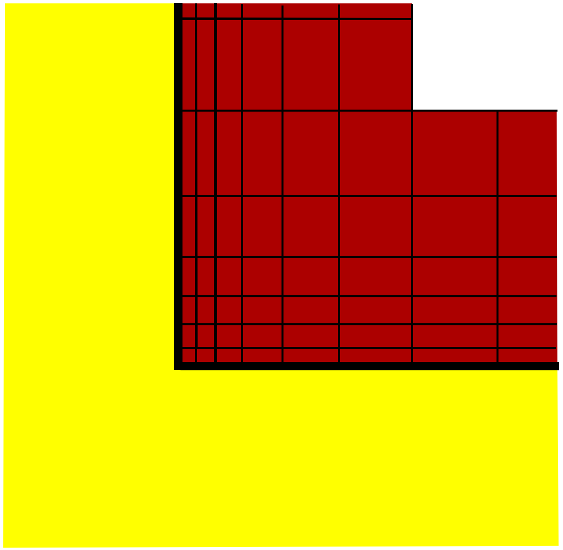
Concave Corner

Blend
Regions

Hybrid Methods



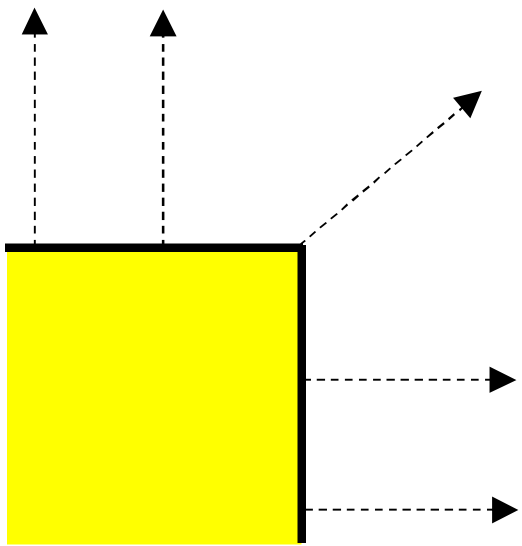
Convex Corner



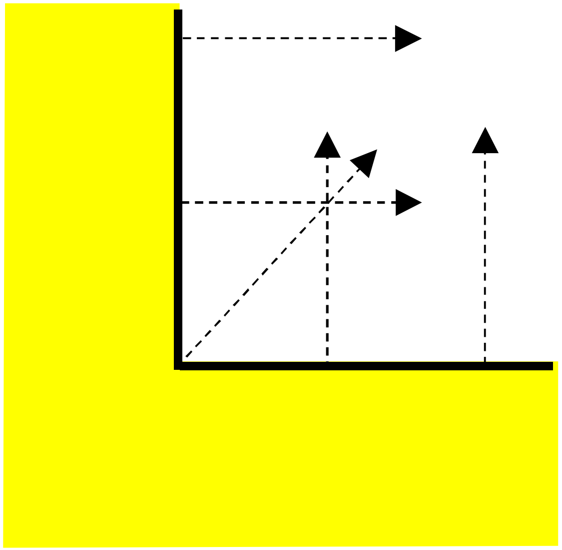
Concave Corner

Blend Regions

Hybrid Methods



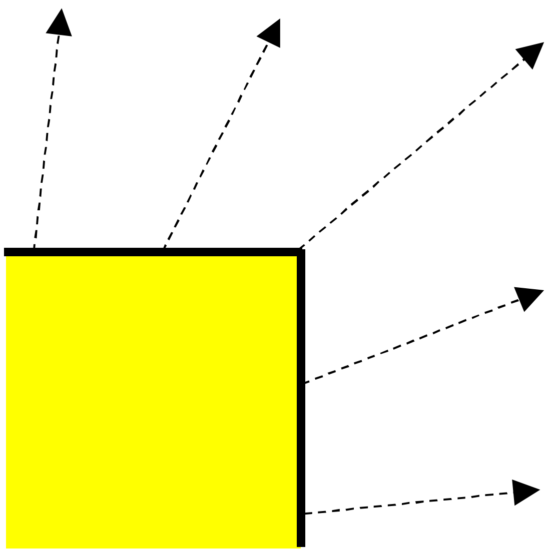
Convex Corner



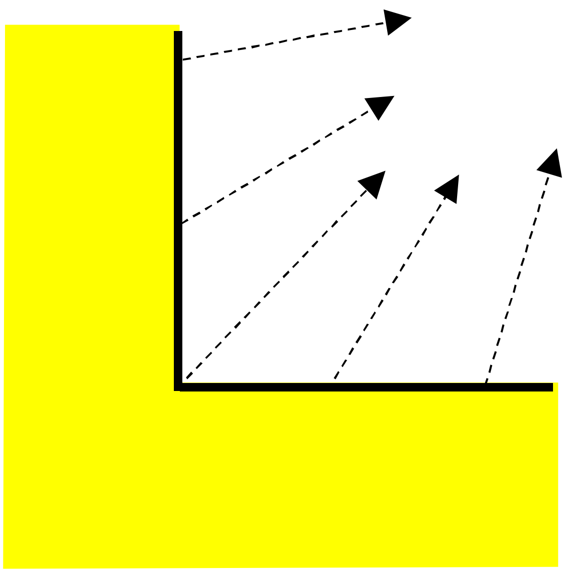
Concave Corner

Smoothed Normals

Hybrid Methods



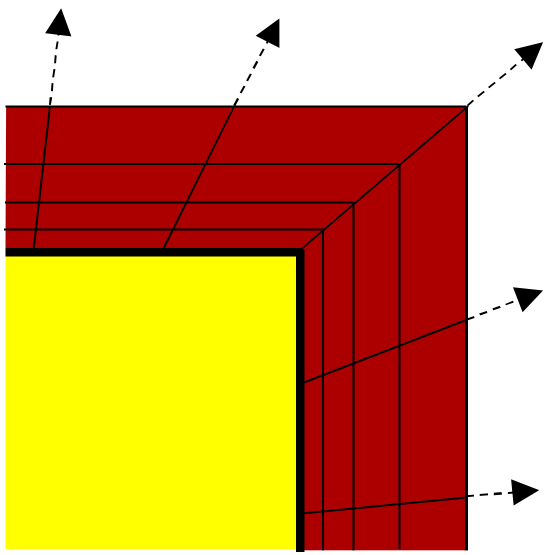
Convex Corner



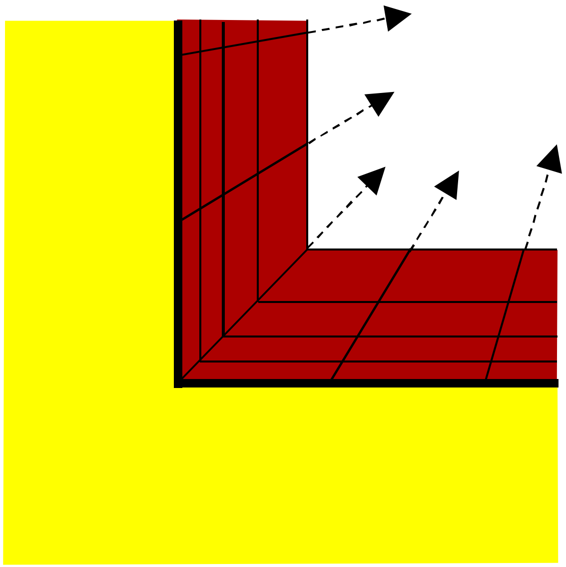
Concave Corner

Smoothed Normals

Hybrid Methods



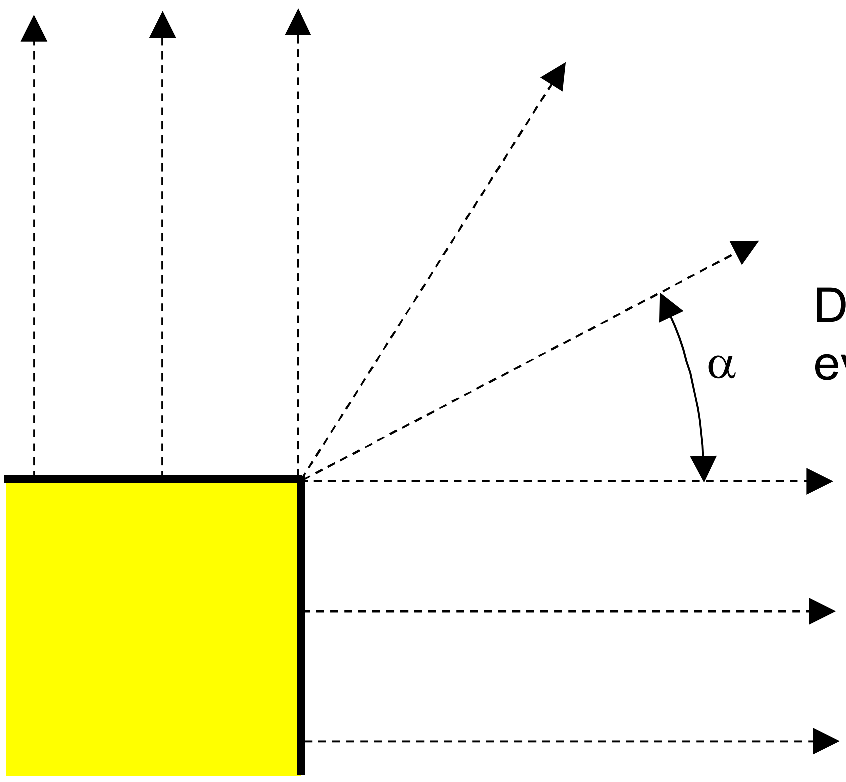
Convex Corner



Concave Corner

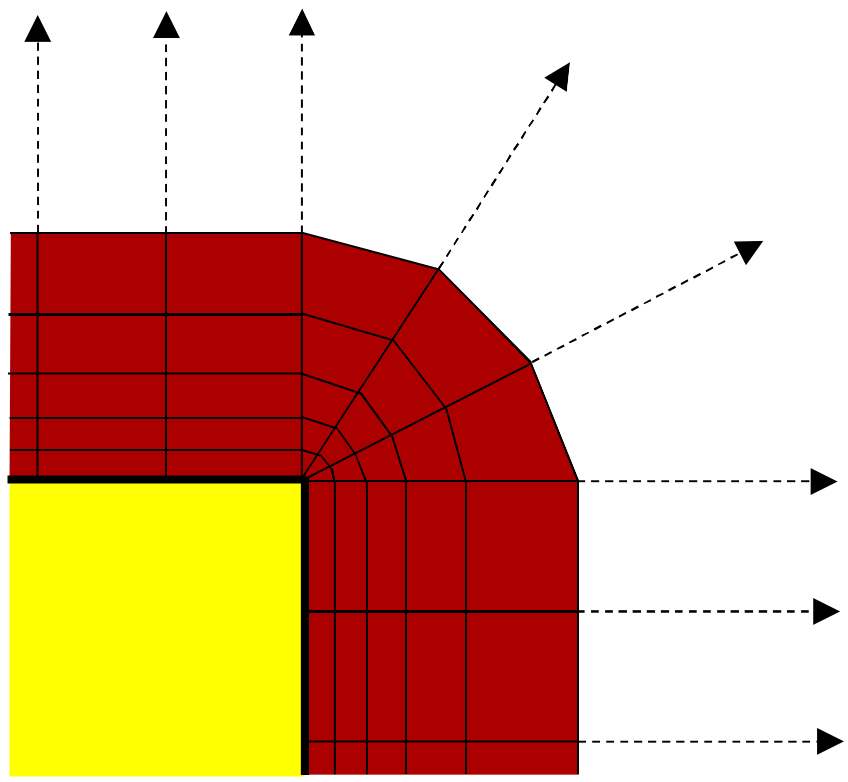
Smoothed Normals

Hybrid Methods



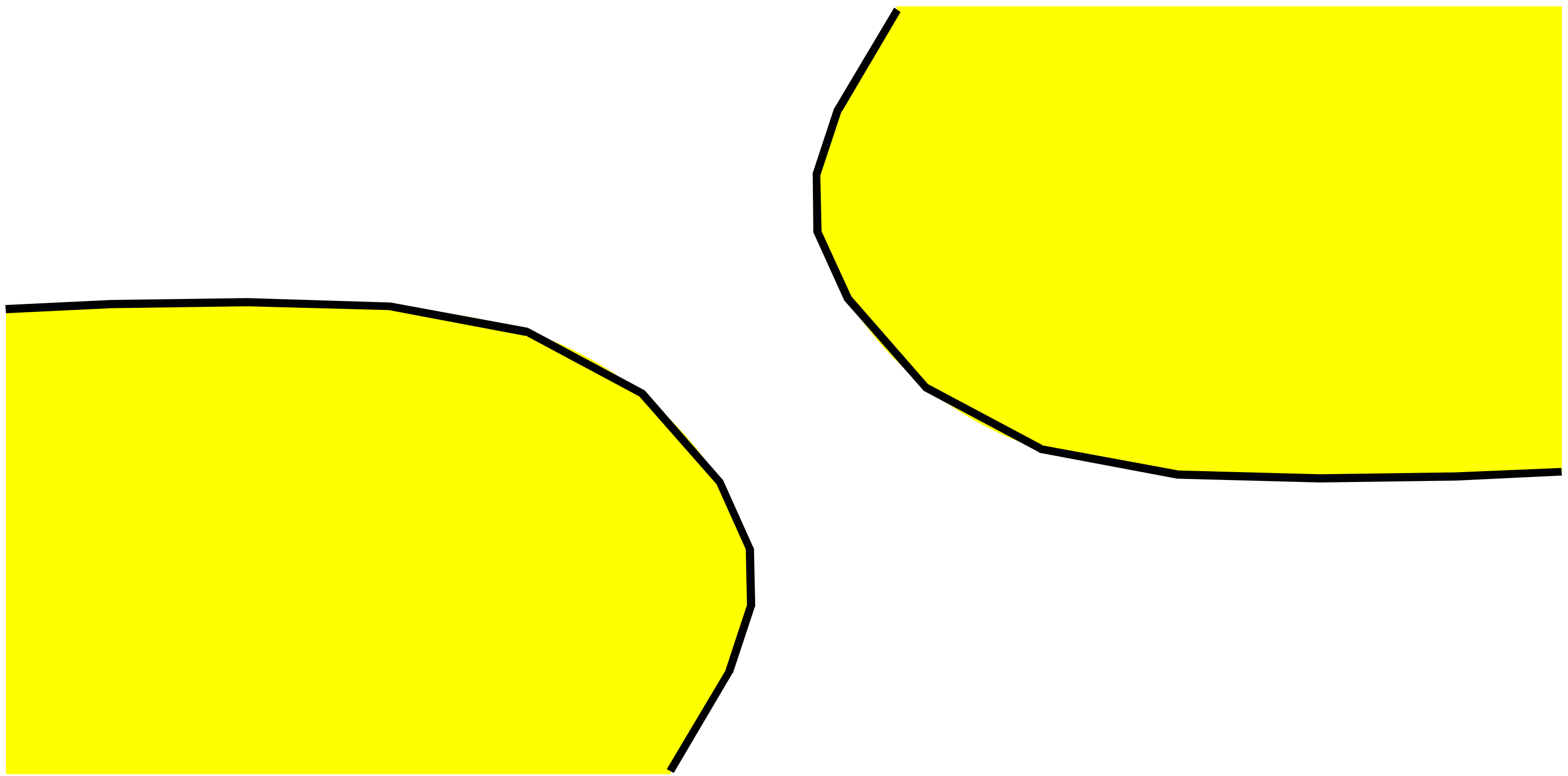
Multiple Normals

Hybrid Methods



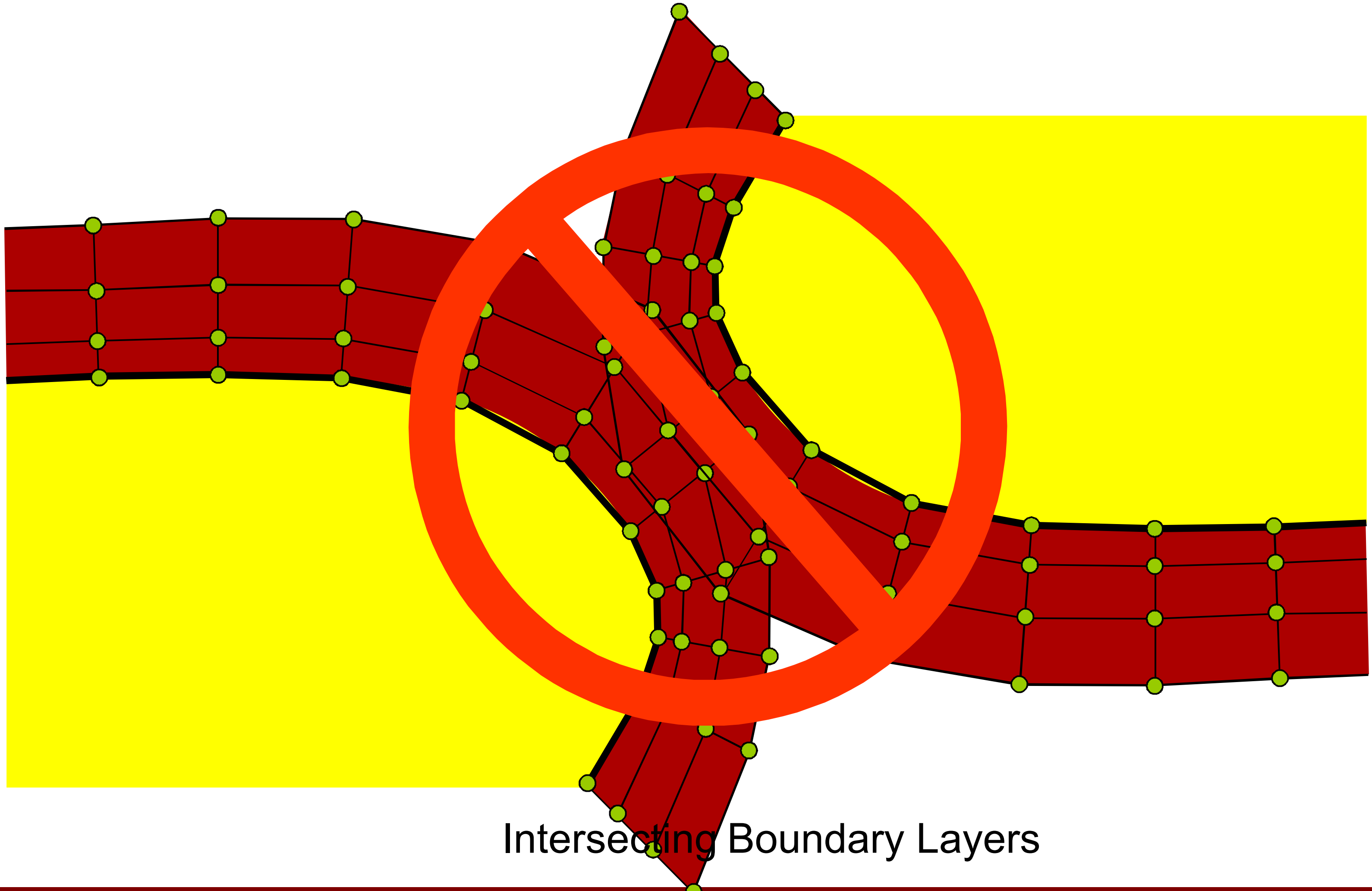
Multiple Normals

Hybrid Methods



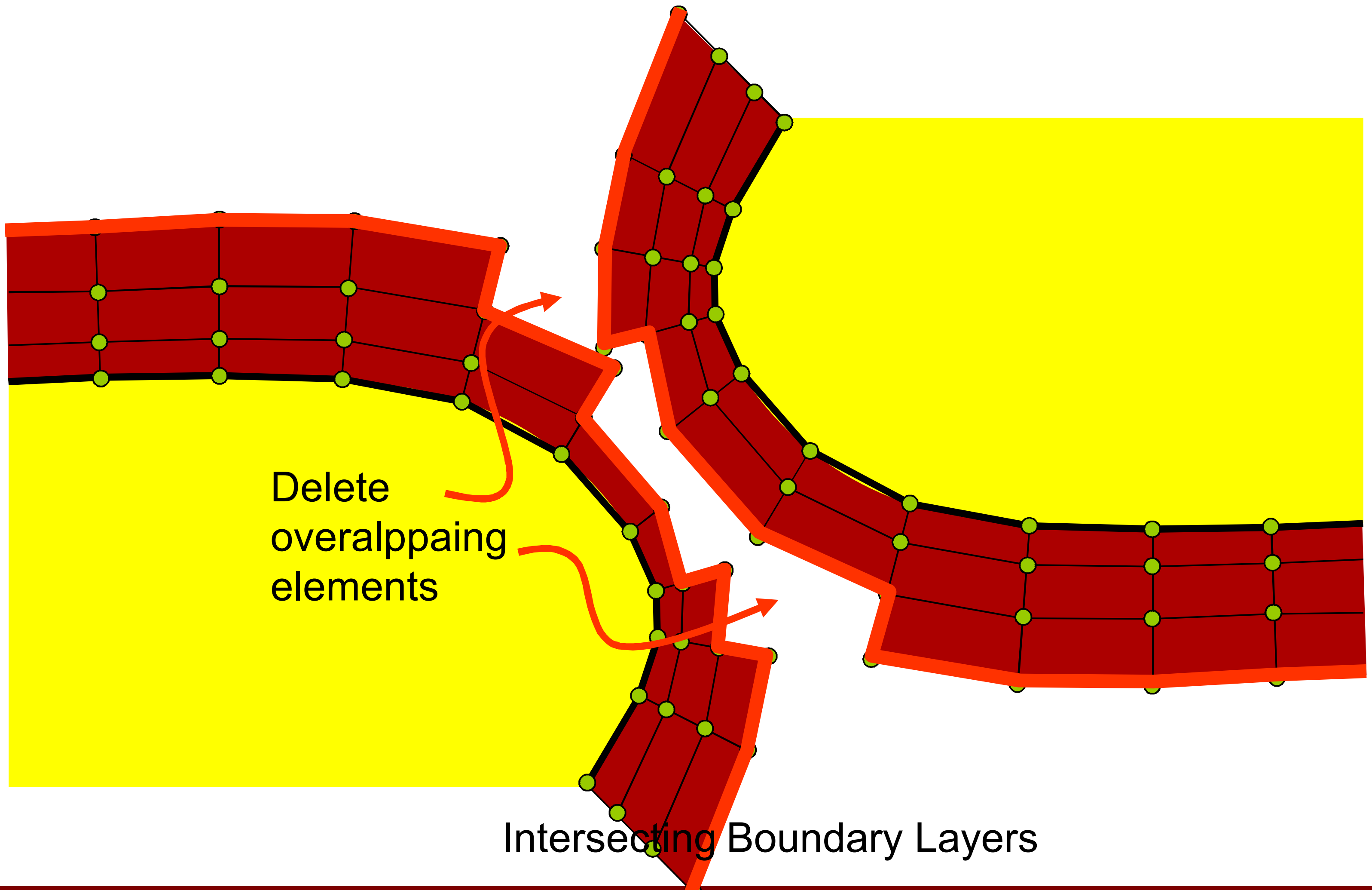
Intersecting Boundary Layers

Hybrid Methods



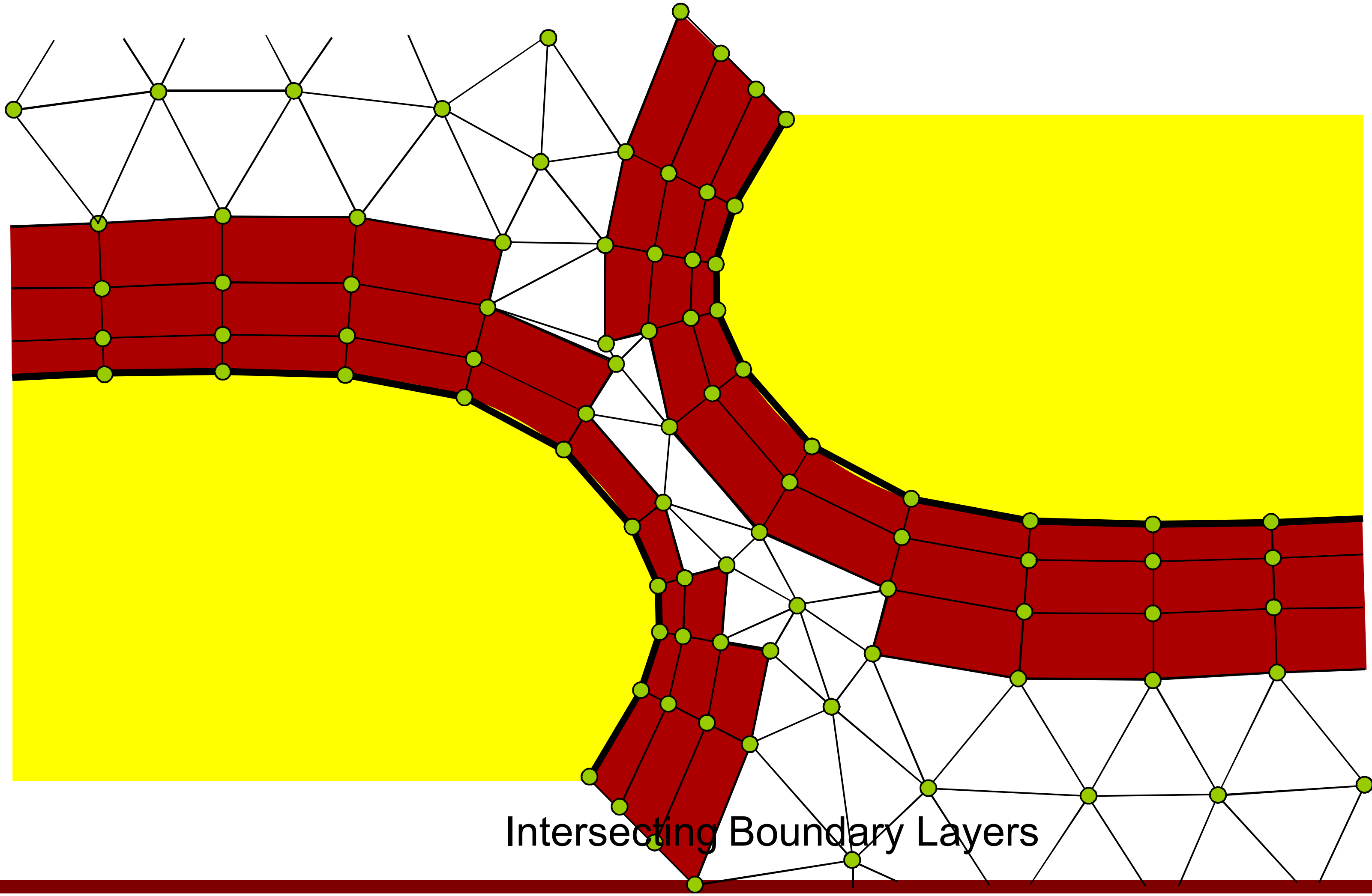
Intersecting Boundary Layers

Hybrid Methods



Intersecting Boundary Layers

Hybrid Methods



Intersecting Boundary Layers

Hybrid Methods

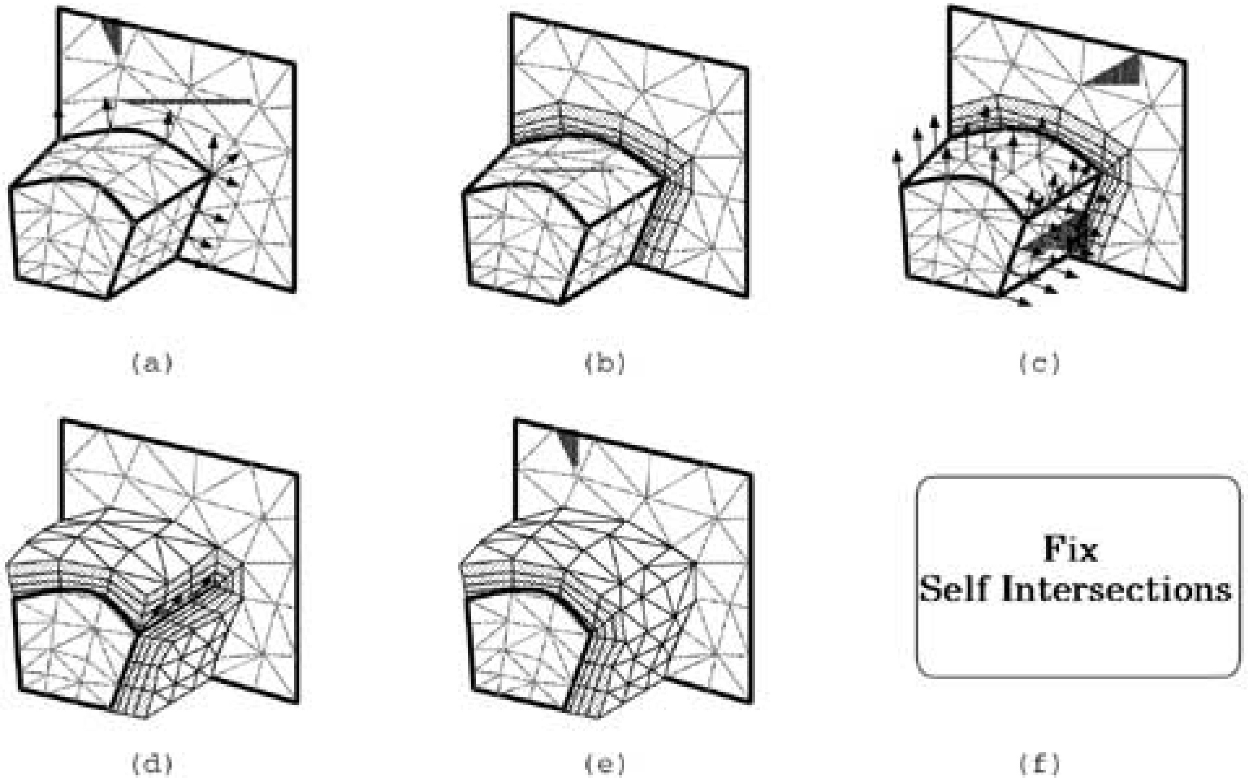


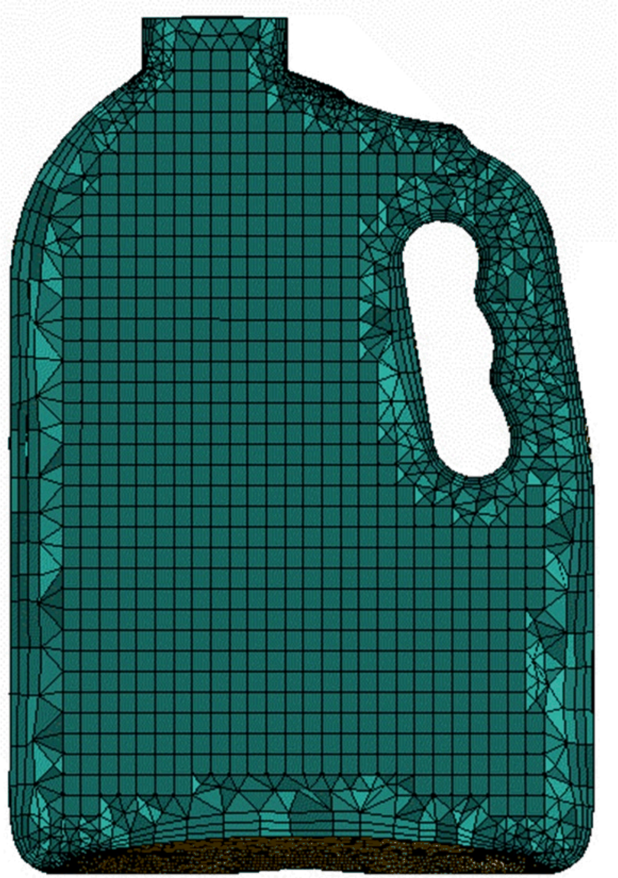
Image courtesy of SCOREC, Rensselaer Polytechnic Institute, http://www.scorec.rpi.edu/

(Garimella, Shephard, 2000)

Hybrid Methods

ANSYS ICEM CFD http://www.ansys.com/products/icemcfd-mesh/tetra/hybrid.html

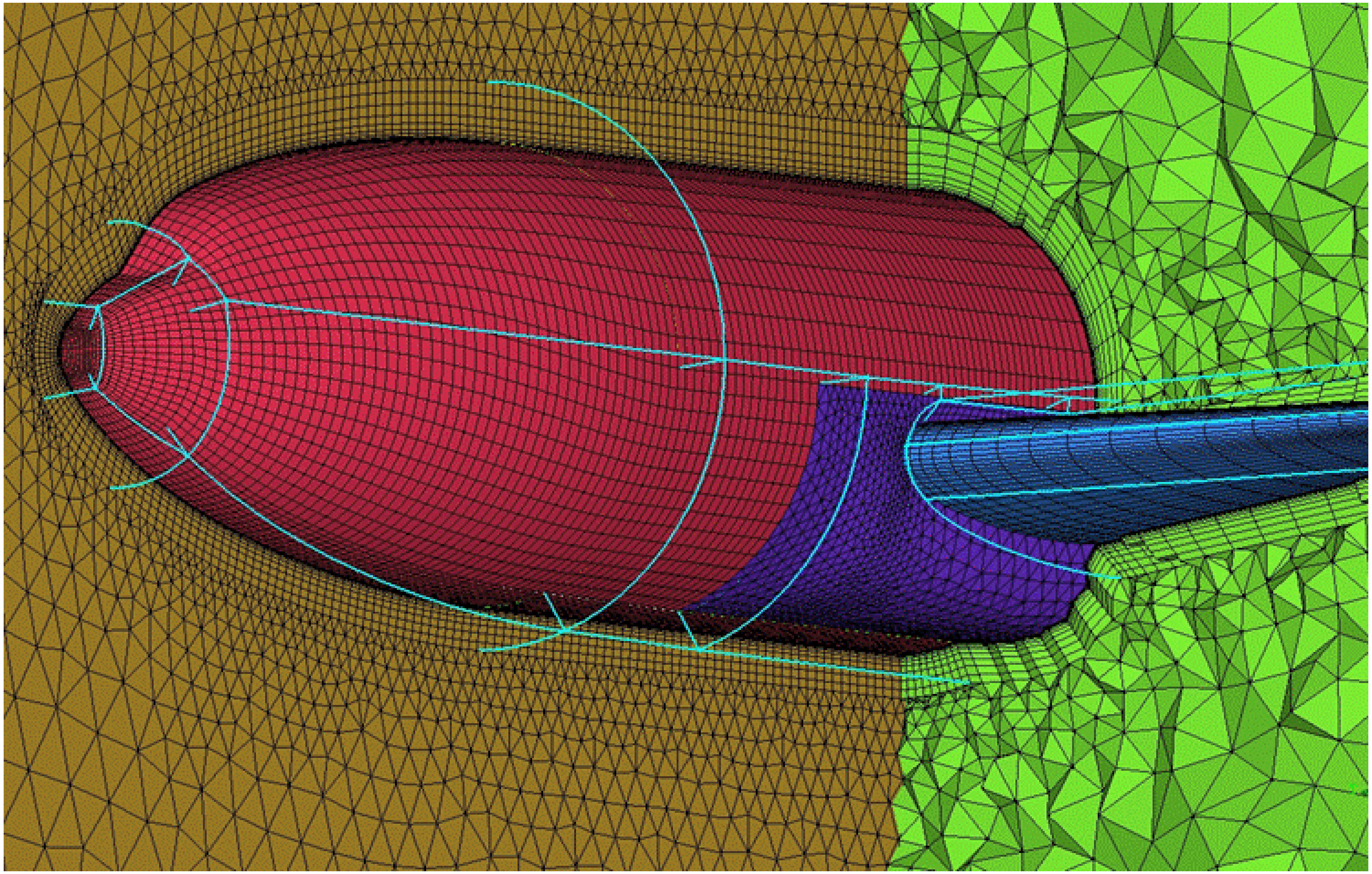
Hexahedron boundary layer with interior tetrahedra



Prism (wedge) boundary layer with tet transition to interior regular hexahedron grid

Hybrid Methods

ANSYS ICEM CFD http://www.ansys.com/products/icemcfd-mesh/hexa/index.htm



Multi-block structured grid combined with tetrahedron far-field mesh