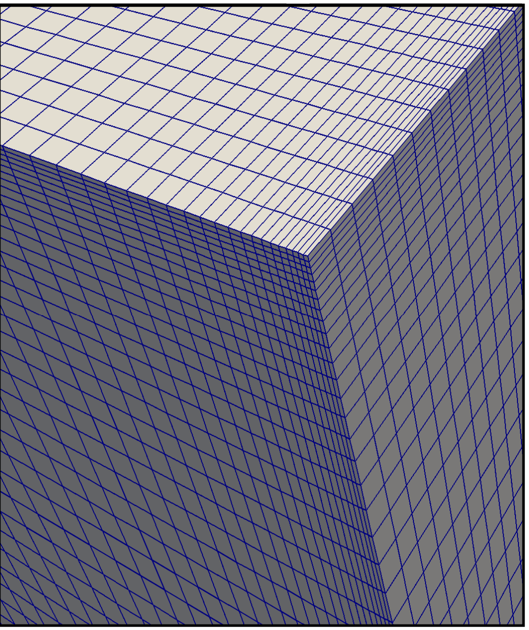
SAND2016-6882C

Mesh Scaling for Affordable Solution Verification

Matt Staten, Brian Carnes

July 29, 2016

WCCM, Seoul, South Korea



Exceptional service in the national interest

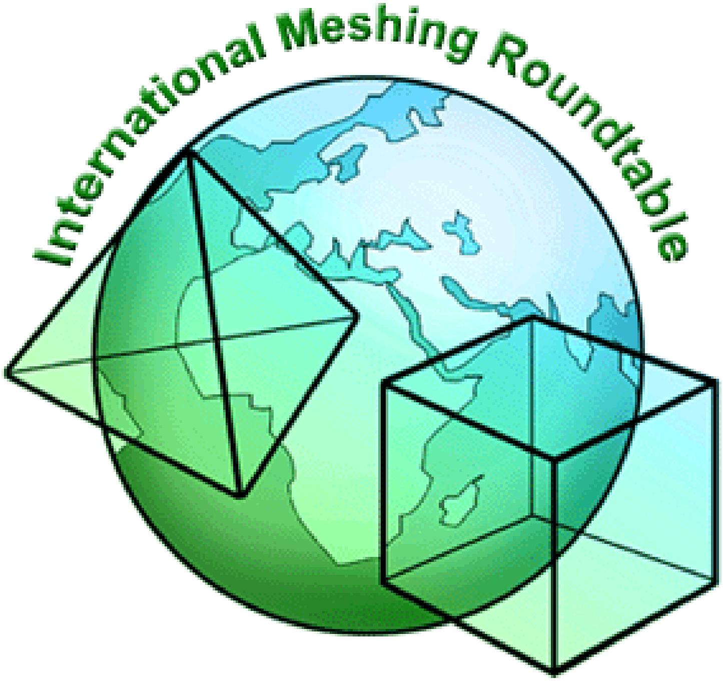
Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. SAND NO. XXXX-XXXXX.

25th International Meshing Roundtable

http://imr.sandia.gov/25imr

September 27-30, 2016

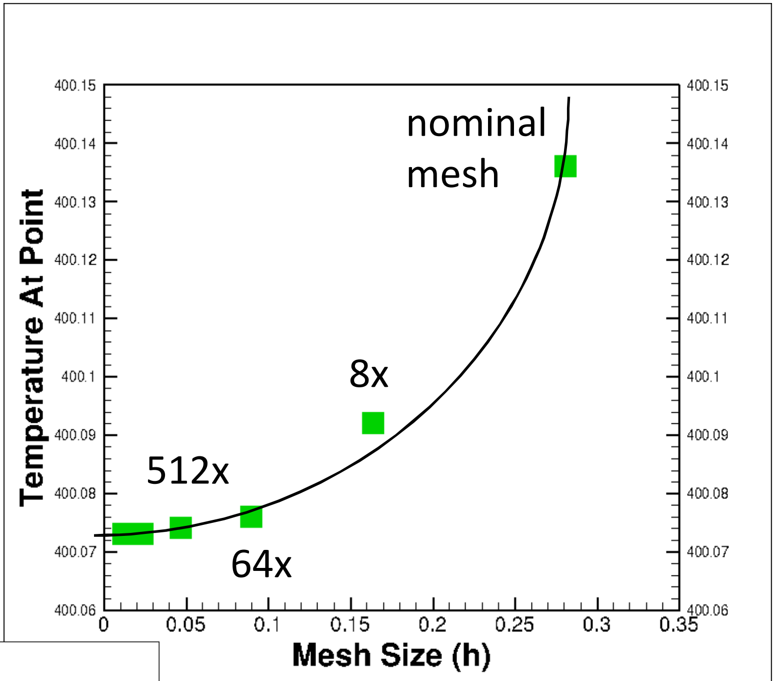
Washington DC, USA

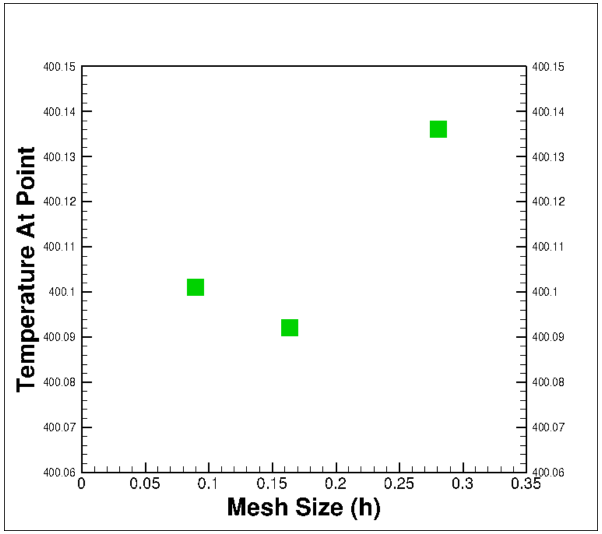


This presentation is based upon a paper to be published in the IMR proceedings.

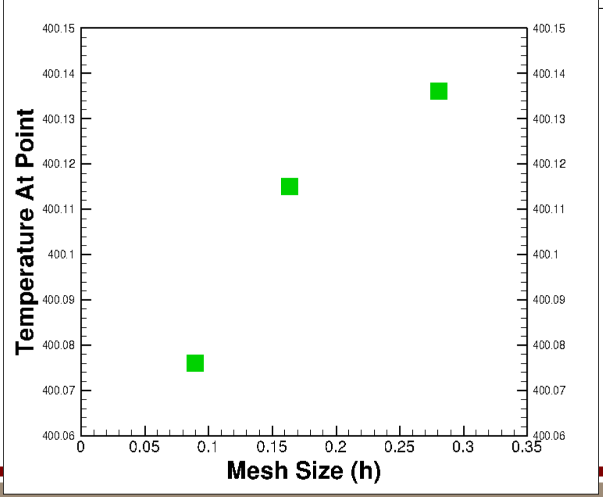
Solution Verification Using UMR is Inefficient and Failure Prone

- Incremental cost of adding another 8x UMR mesh is 16x (CPU time for explicit)
- Extrapolation can fail – you may need to keep refining the mesh





Data oscillation - yields undefined convergence rate

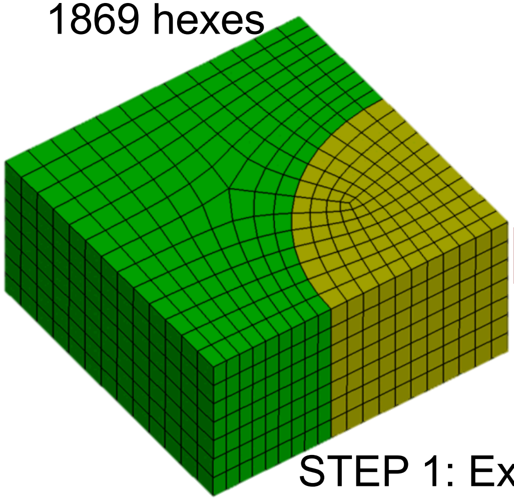


Data is diverging – yields negative convergence rate

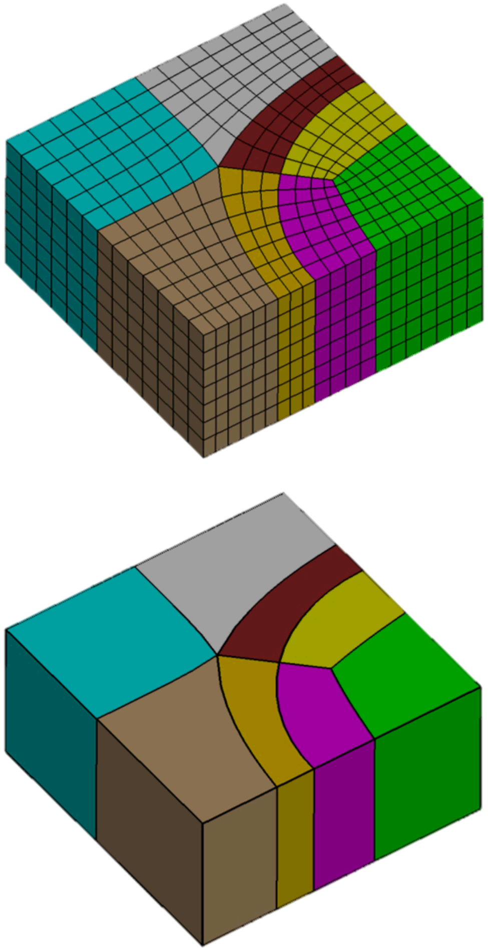
How Mesh Scaling Works

Blocks Decomposition is determined by both mesh irregularities and volume boundaries.

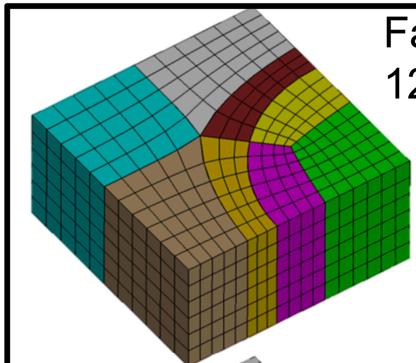
Initial Mesh
1869 hexes



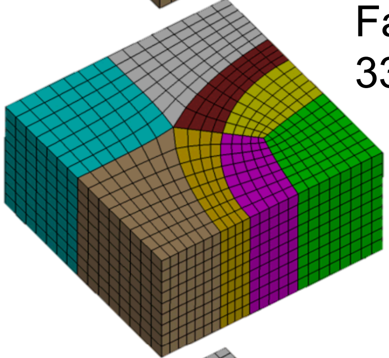
STEP 1: Extract the Block Decomposition



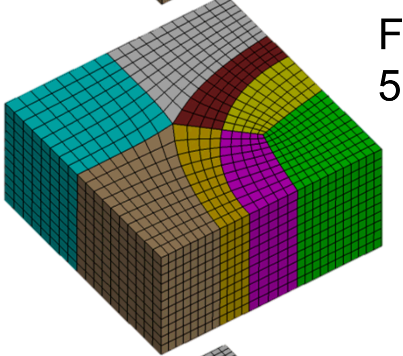
Step 2: Remesh each block at a different size



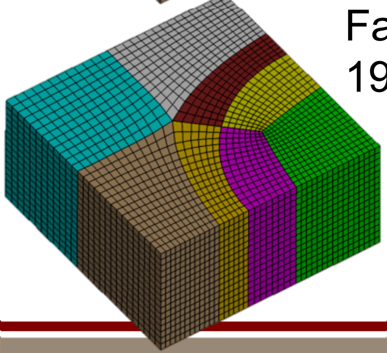
Factor 0.667
1230 hexes



Factor 1.75
3321 hexes



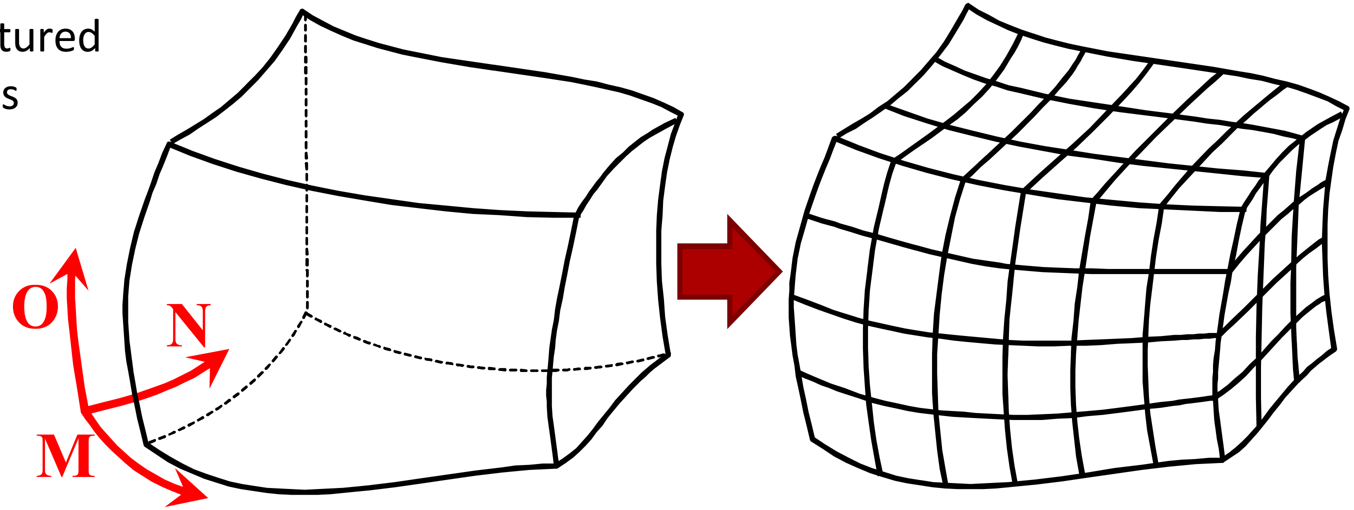
Factor 3.0
5841 hexes



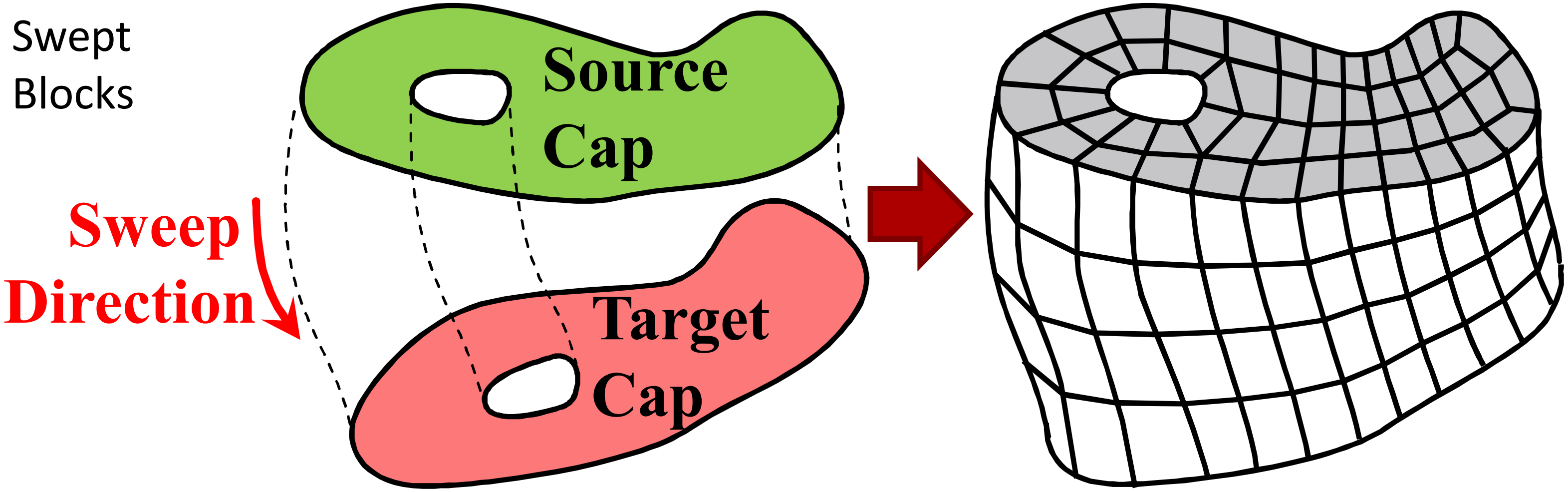
Factor 10.0
19,040 hexes

Block Types Supported

Structured Blocks



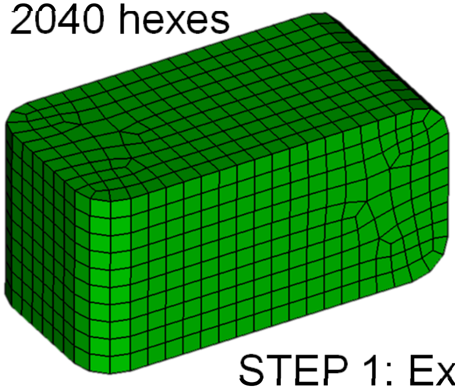
Swept Blocks



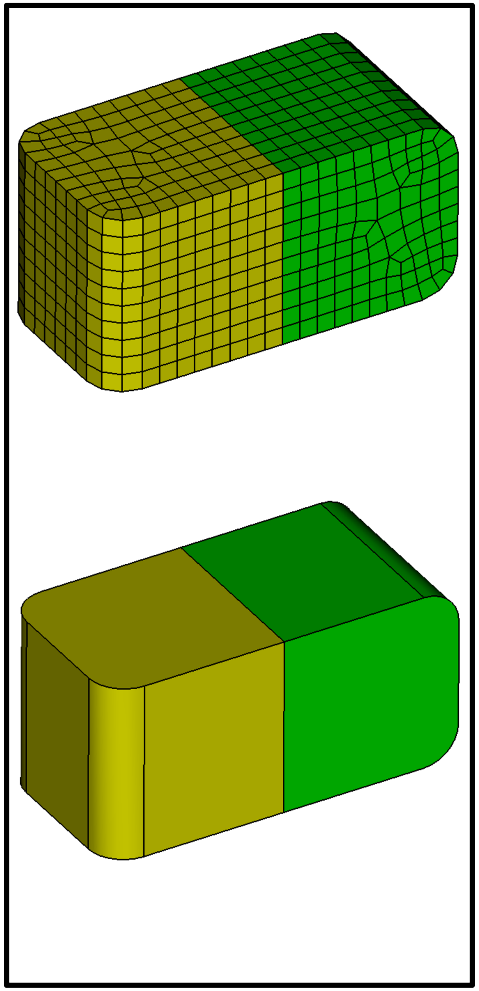
How Mesh Scaling with Swept Blocks Works

Blocks Decomposition is determined by both mesh irregularities, volume boundaries, and swept mesh topology detection.

Initial Mesh
2040 hexes

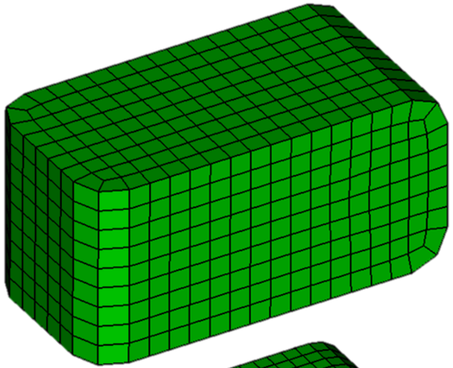


STEP 1: Extract the Block Decomposition

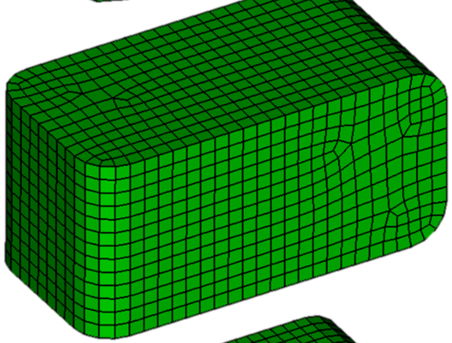


Step 2: Remesh each block at a different size

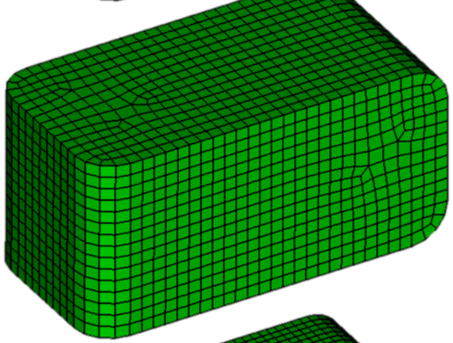
Factor 0.667
1422 hexes



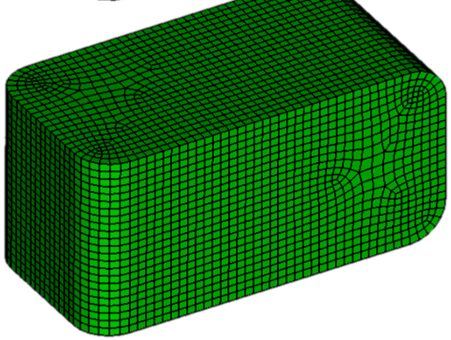
Factor 1.75
4147 hexes



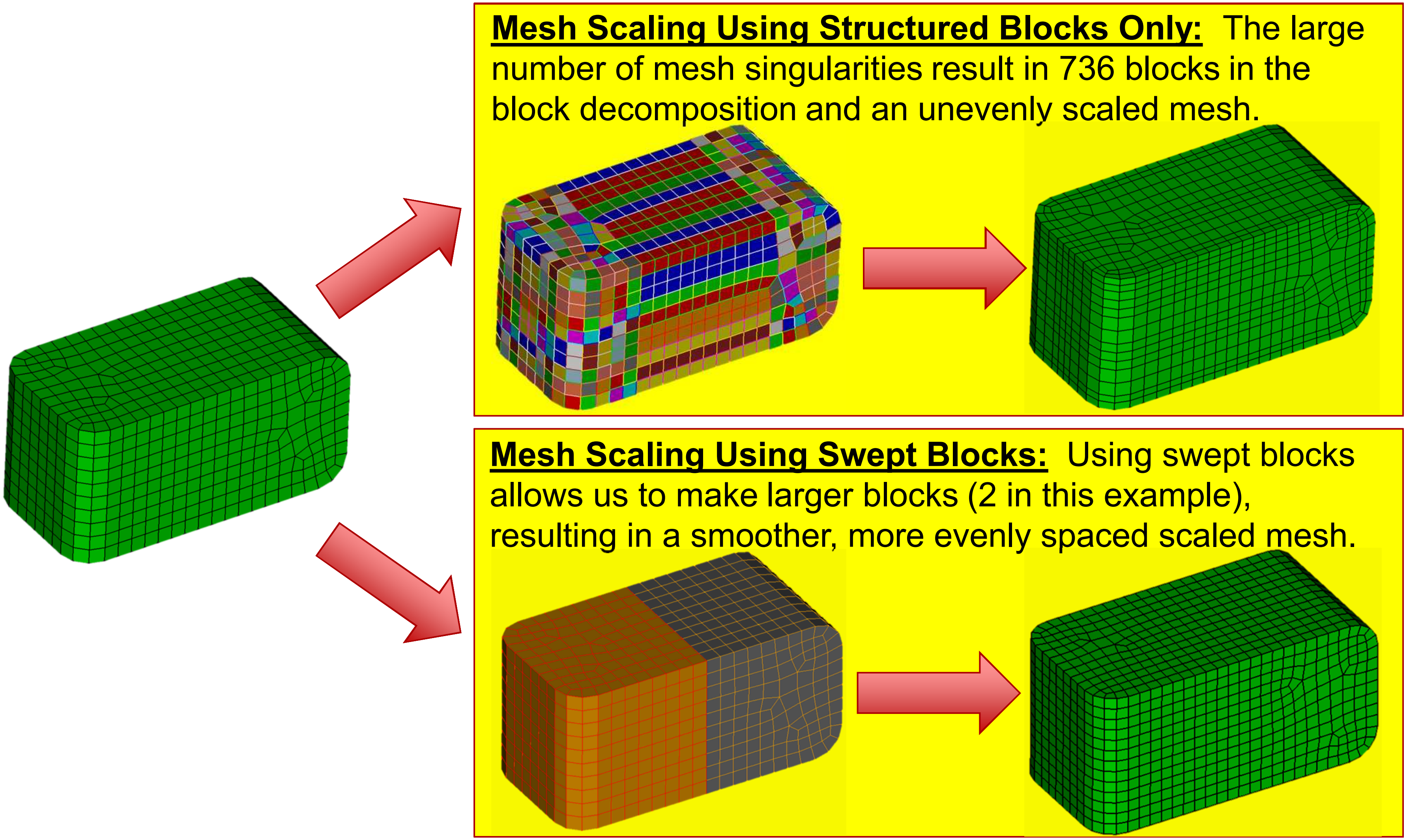
Factor 3.0
6480 hexes



Factor 10.0
22,011 hexes



Comparing Swept vs Structured Blocks

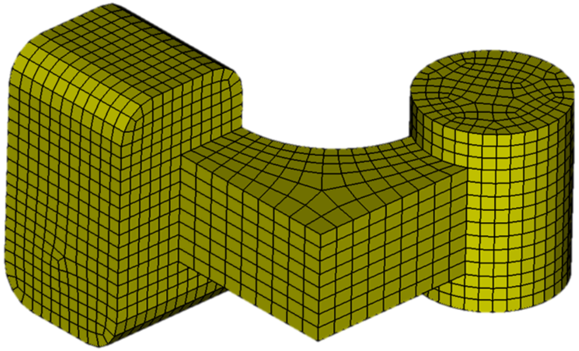


Mesh Scaling Using Structured Blocks Only: The large number of mesh singularities result in 736 blocks in the block decomposition and an unevenly scaled mesh.

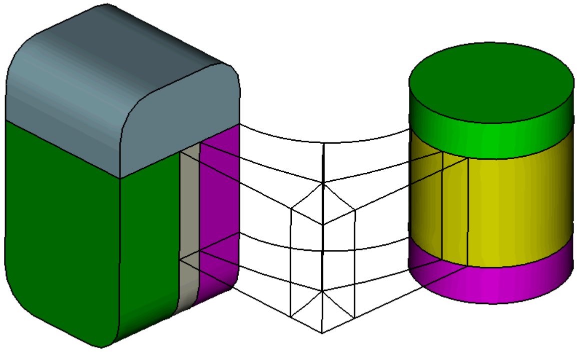
Mesh Scaling Using Swept Blocks: Using swept blocks allows us to make larger blocks (2 in this example), resulting in a smoother, more evenly spaced scaled mesh.

In large complex NW models, the block decomposition has a mixture of swept and structured blocks. Swept blocks are first added where sweeps can be found, and structured blocks fill in the remaining non-swept regions.

Mesh Scaling typically defines a mix of Swept and Structured Blocks

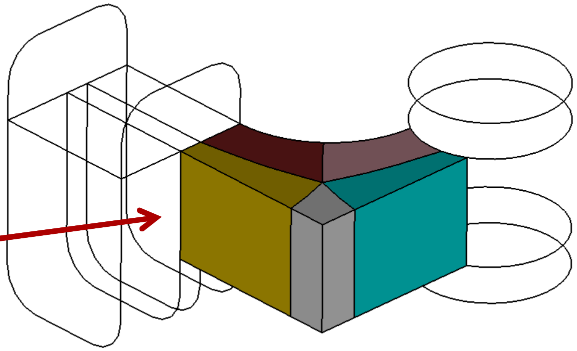


Initial Mesh
5206 hexes



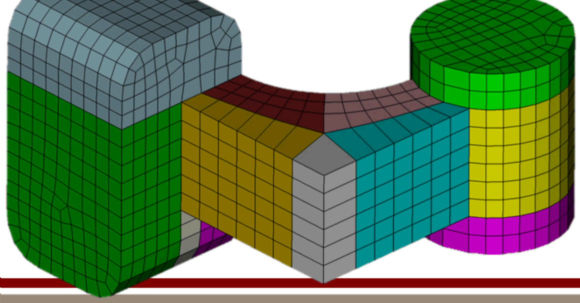
The block decomposition is composed of:

- 7 swept blocks, and
- 5 structured blocks

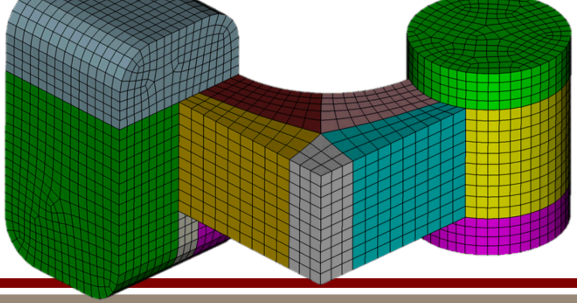


Scaled Meshes:

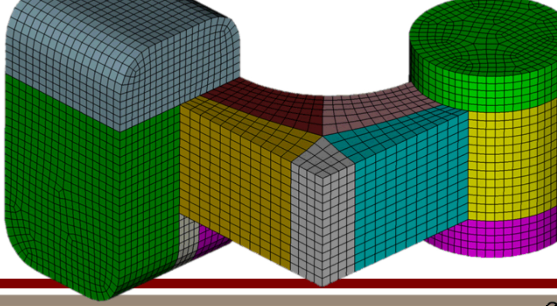
0.5X: 2490 hexes



2X: 10,922 hexes



4X: 21,347 hexes



Why not just remesh original Blocking?

- Original blocking comes in 2 types:
  - Block structured blocking of mapped zones (CFD), typical of meshing air surrounding objects
  - Geometry decomposition into pave-and-sweep sub-volumes, typical of solid mechanics meshes of objects, rather than the air.
- The original blocking may not be available.
  - Model archived without blocking.
- The original blocking may not just "remesh" easily.
  - Block structure may have been assigned intervals that become in consistent when meshing at a different size.
- Mesh may have been modified after generating from original blocking (refinement, pillowing). Remeshing will throw away those modifications.
- Analysis attributes are often applied to the mesh, rather than blocking. Remeshing will require re-assignment of those attributes.

Input Parameters to Mesh Scaling

- Existing all-hex mesh
  - tets, pyramids, and wedges not currently supported.

- Multiplier on number of hex elements in the mesh (***Multiplier***)
  - 10,000 hexes scaled by 3X results in roughly 30,000 hexes.

- Min Interval parameter (***MinInt***)
  - Used to ensure mesh changes a little bit everywhere.

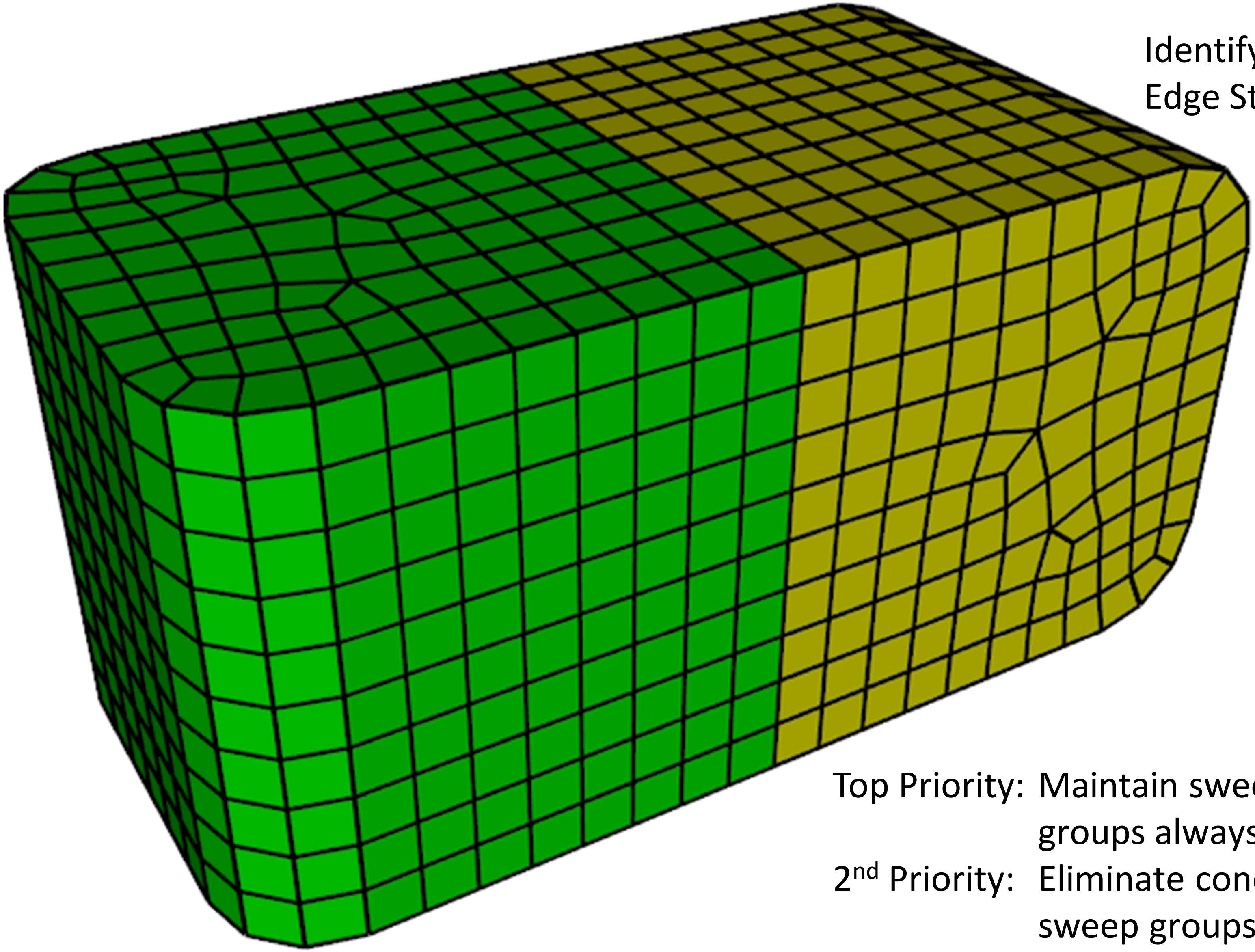
Steps to Mesh Scaling

- STEP 1 : Identify Swept Mesh Topology
- STEP 2 : Identify Constraints
- STEP 3 : Propagate Block Constraints
- STEP 4 : Construct Block Decomposition
- STEP 5 : Extract Sizing Information
- STEP 6 : Delete Original Mesh
- STEP 7 : Compute New Curve Intervals
- STEP 8 : Remesh Block Decomposition

Steps to Mesh Scaling

- STEP 1 : Identify Swept Mesh Topology
- STEP 2 : Identify Constraints
- STEP 3 : Propagate Block Constraints
- STEP 4 : Construct Block Decomposition
- STEP 5 : Extract Sizing Information
- STEP 6 : Delete Original Mesh
- STEP 7 : Compute New Curve Intervals
- STEP 8 : Remesh Block Decomposition

STEP 1 : Identify Swept Mesh Topology



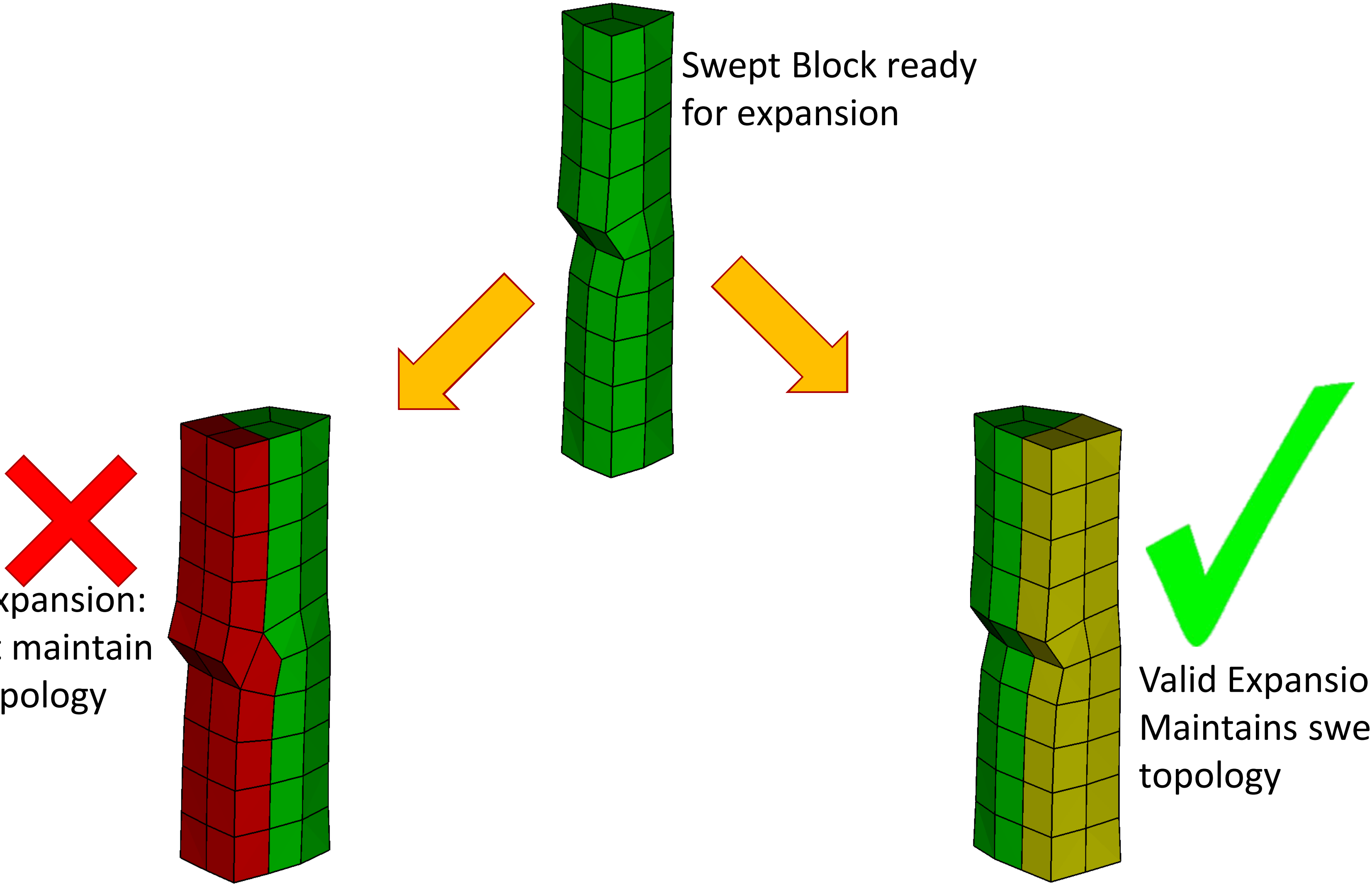
Identify Interior Singular Edge Strings

Collect hexes adjacent to each string, expand, and merge

Top Priority:	Maintain sweepable sweep groups always
2nd Priority:	Eliminate concavities on sweep groups
3rd Priority:	Expand smaller sweep groups

STEP 1 : Identify Swept Mesh Topology

Swept Block Expansions Must Maintain Swept Topology



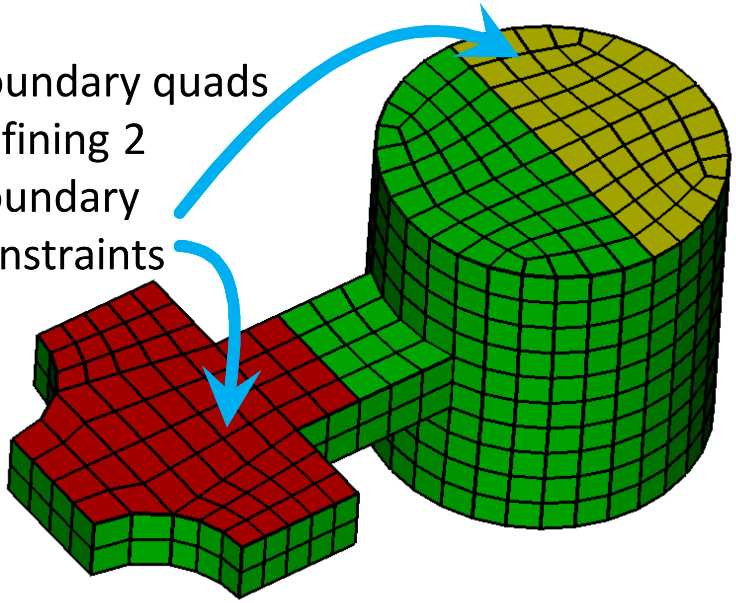
Swept Block ready for expansion

Invalid Expansion: Does not maintain swept topology

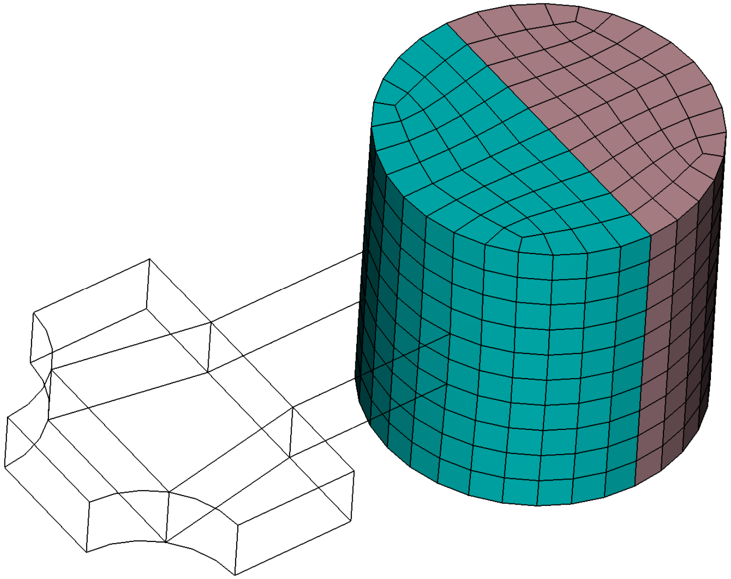
Valid Expansion: Maintains swept topology

STEP 1 : Identify Swept Mesh Topology

Additional example model



Boundary quads defining 2 boundary constraints



On this example model, 2 sweep groups are defined. The yellow boundary constraint stops the sweep groups from joining into one.

The remainder of the model, which in this case also has a swept topology, is left out of the sweep groups because it does not contain any interior singular edge strings.

Steps to Mesh Scaling

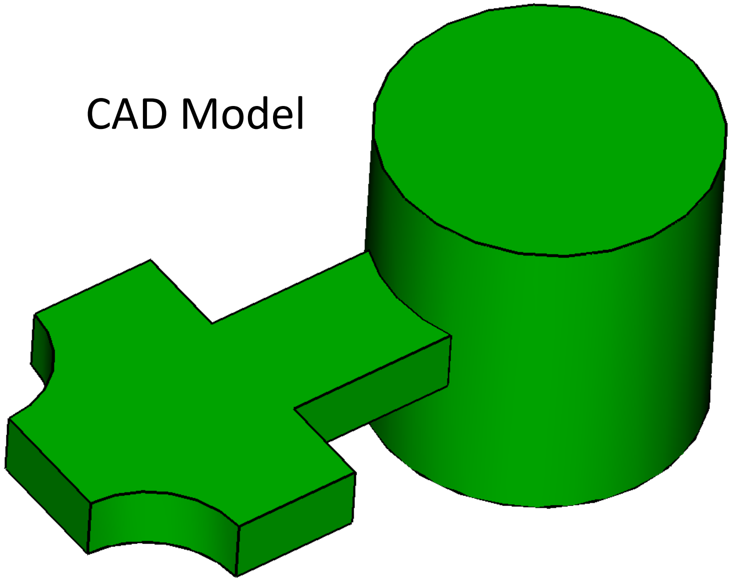
- STEP 1 : Identify Swept Mesh Topology
- STEP 2 : Identify Constraints
- STEP 3 : Propagate Block Constraints
- STEP 4 : Construct Block Decomposition
- STEP 5 : Extract Sizing Information
- STEP 6 : Delete Original Mesh
- STEP 7 : Compute New Curve Intervals
- STEP 8 : Remesh Block Decomposition

STEP 2 : Identify Constraints

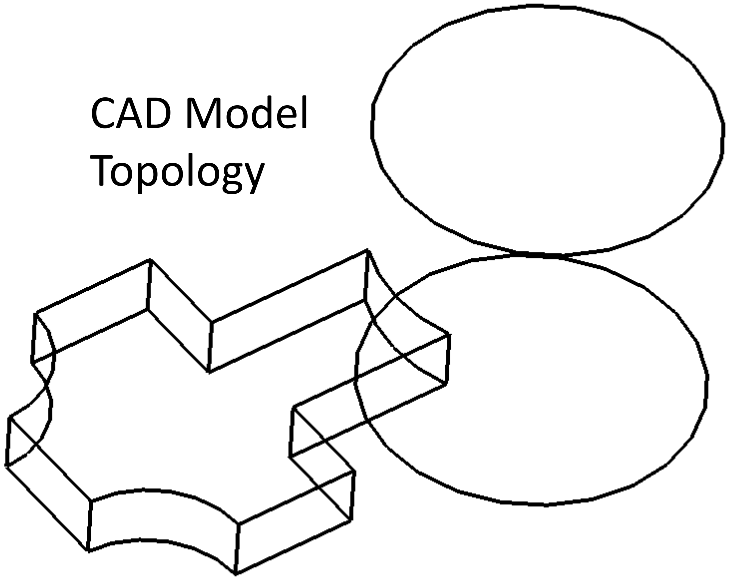
- In order to identify constraints, we define a new model topology, T_{MS}. We imprint into T_{MS} :
  - All Swept blocks
    - A CAD body that has more than one sweep is split to capture the sweep blocks
  - CAD Topology
    - All CAD vertices, curves, and surfaces are captured in T_{MS}.
  - Analysis Constraints
    - Material zones separated into separate volumes in T_{MS},
    - Point Loads imprinted into T_{MS} as vertices,
    - Boundaries of surface loads imprinted into the boundary surfaces,
    - etc.

STEP 2 : Identify Constraints

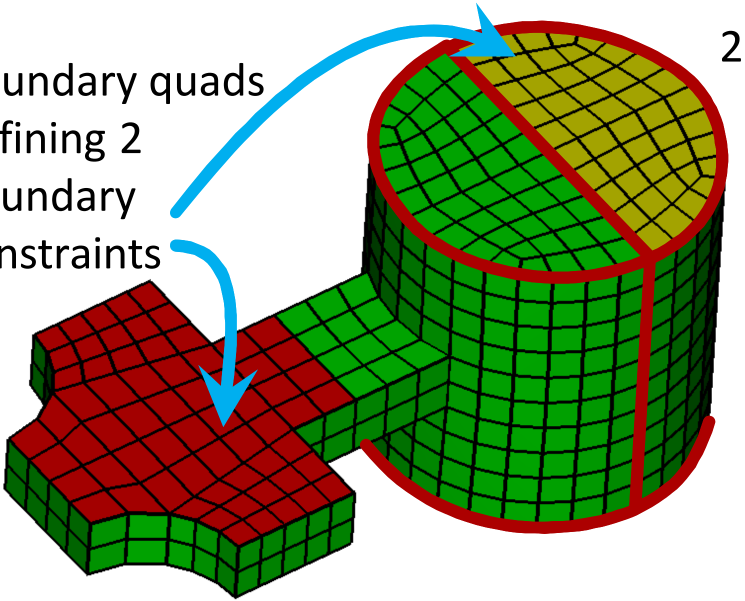
Constructing Mesh Scaling Topology, T_{MS}



CAD Model

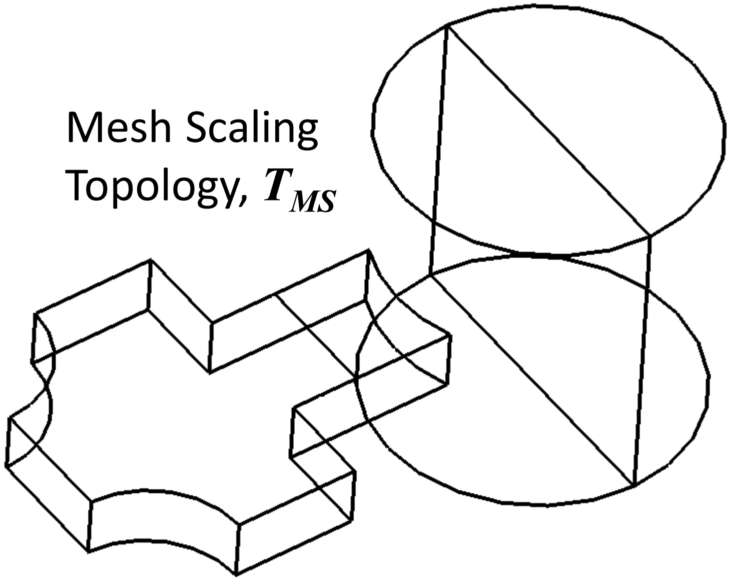


CAD Model Topology



Boundary quads defining 2 boundary constraints

2 sweep groups

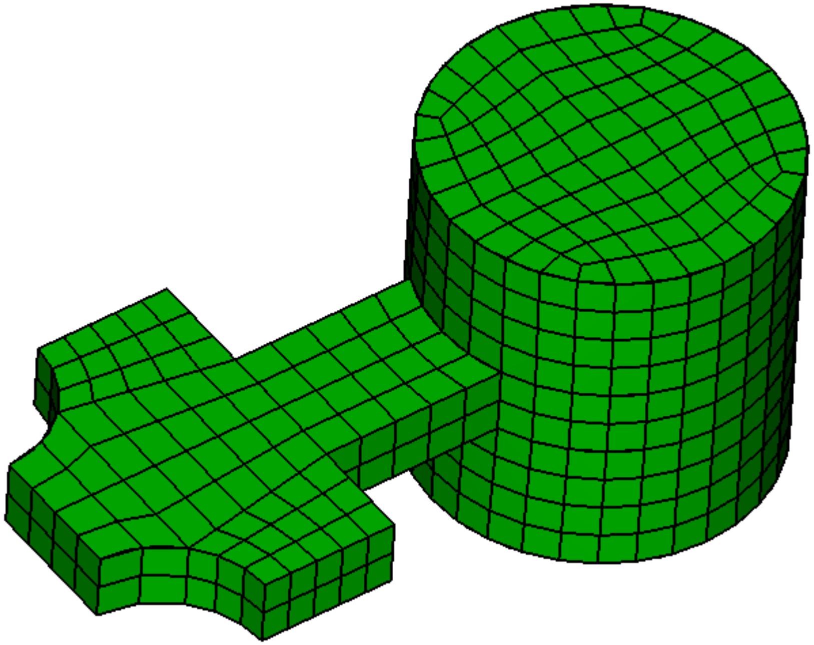


Mesh Scaling Topology, T_{MS}

STEP 2 : Identify Constraints

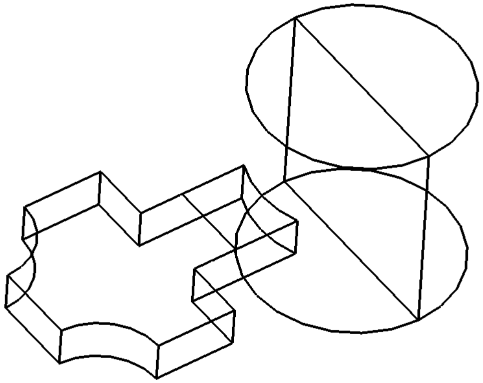
- We define:
  - A set of mesh faces, Q_C, which must be maintained in the scaled mesh
  - A set of mesh edges, E_C, which must be maintained in the scaled mesh
  - A set of mesh nodes, N_C, which must be maintained in the scaled mesh
  - A set of rib edges, E_{ribs}, on each sweep

- Q_C contains any quad face on any surface in T_{MS}.

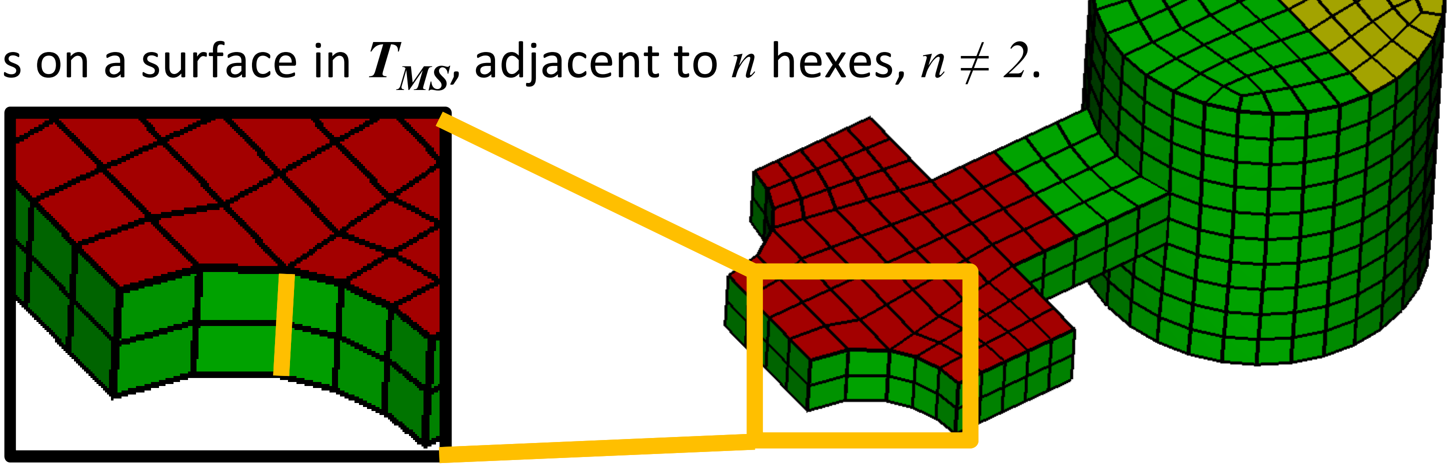


STEP 2 : Identify Constraints, E_C

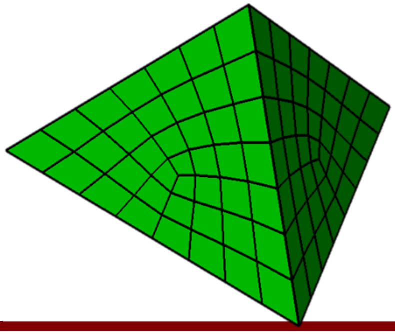
- E_C contains:
  - All edges on any curve in T_{MS}.

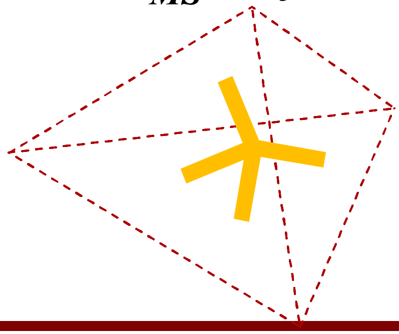


  - All edges on a surface in T_{MS}, adjacent to n hexes, $n \neq 2$.



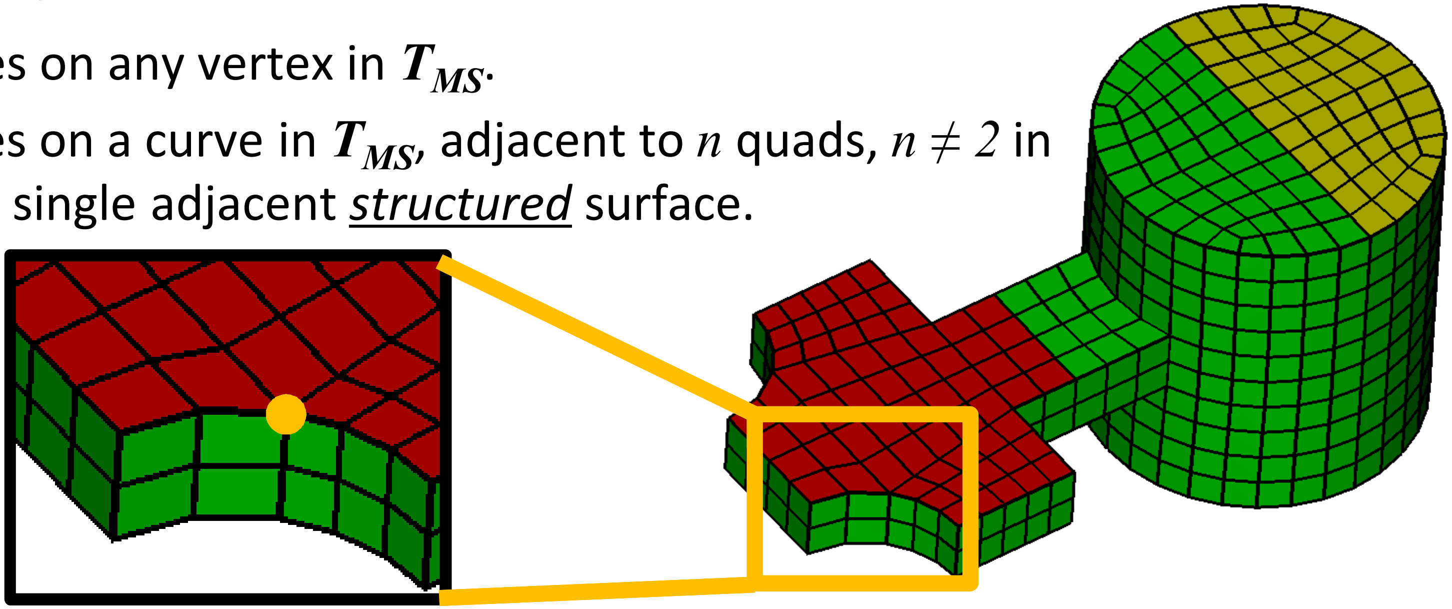
  - All edges inside a *structured* volume in T_{MS}, adjacent to n hexes, $n \neq 4$.



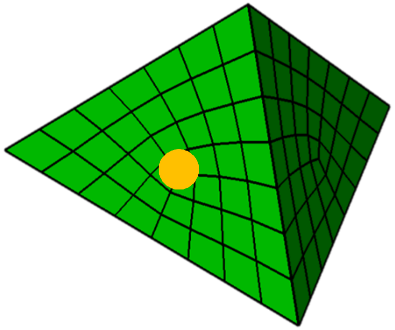


STEP 2 : Identify Constraints, N_C

- N_C contains:
  - All nodes on any vertex in $\boldsymbol{T}_{MS}$.
  - All nodes on a curve in $\boldsymbol{T}_{MS}$, adjacent to n quads, $n \neq 2$ in any one single adjacent *structured* surface.



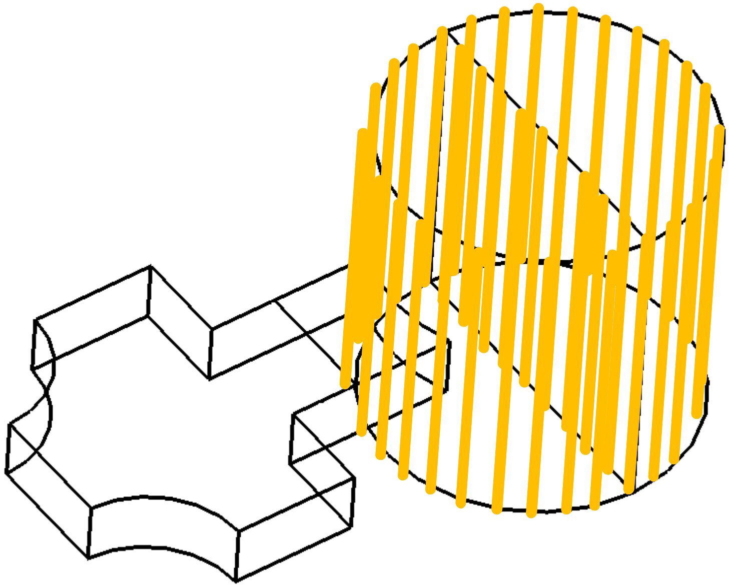
  - All nodes on a *structured* surface in $\boldsymbol{T}_{MS}$, adjacent to n hexes, $n \neq 4$.



  - Additional cases outlined in paper.

STEP 2 : Identify Constraints, E_{ribs}

- E_{ribs} contains all edges on boundaries of swept blocks running parallel to sweep direction:

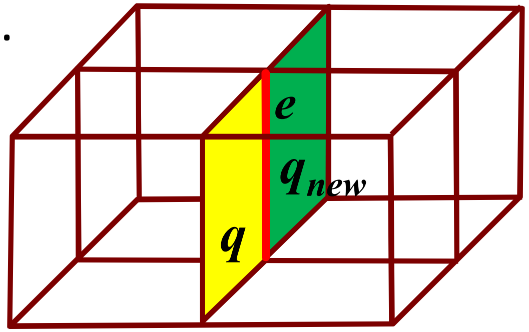


Steps to Mesh Scaling

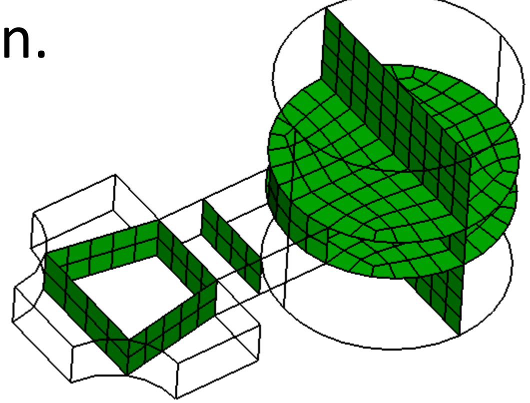
- STEP 1 : Identify Swept Mesh Topology
- STEP 2 : Identify Constraints
- STEP 3 : Propagate Block Constraints
- STEP 4 : Construct Block Decomposition
- STEP 5 : Extract Sizing Information
- STEP 6 : Delete Original Mesh
- STEP 7 : Compute New Curve Intervals
- STEP 8 : Remesh Block Decomposition

STEP 3 : Propagate Constraints

- We modify the Constraint Sets:
  - Place all edges adjacent to any node in $\boldsymbol{N_C}$ into $\boldsymbol{E_C}$.
  - Place all quads adjacent to any edge in $\boldsymbol{E_C}$ into $\boldsymbol{Q_C}$.
  - Iteratively propagate from each edge, e, on the boundary of each quad, q, in $\boldsymbol{Q_C}$ to q_{new}. Insert each q_{new} into $\boldsymbol{Q_C}$. Exclude from propagation any e in $\boldsymbol{E_{ribs}}$.



- QC now contains all quads on the boundary of each block in the decomposition.

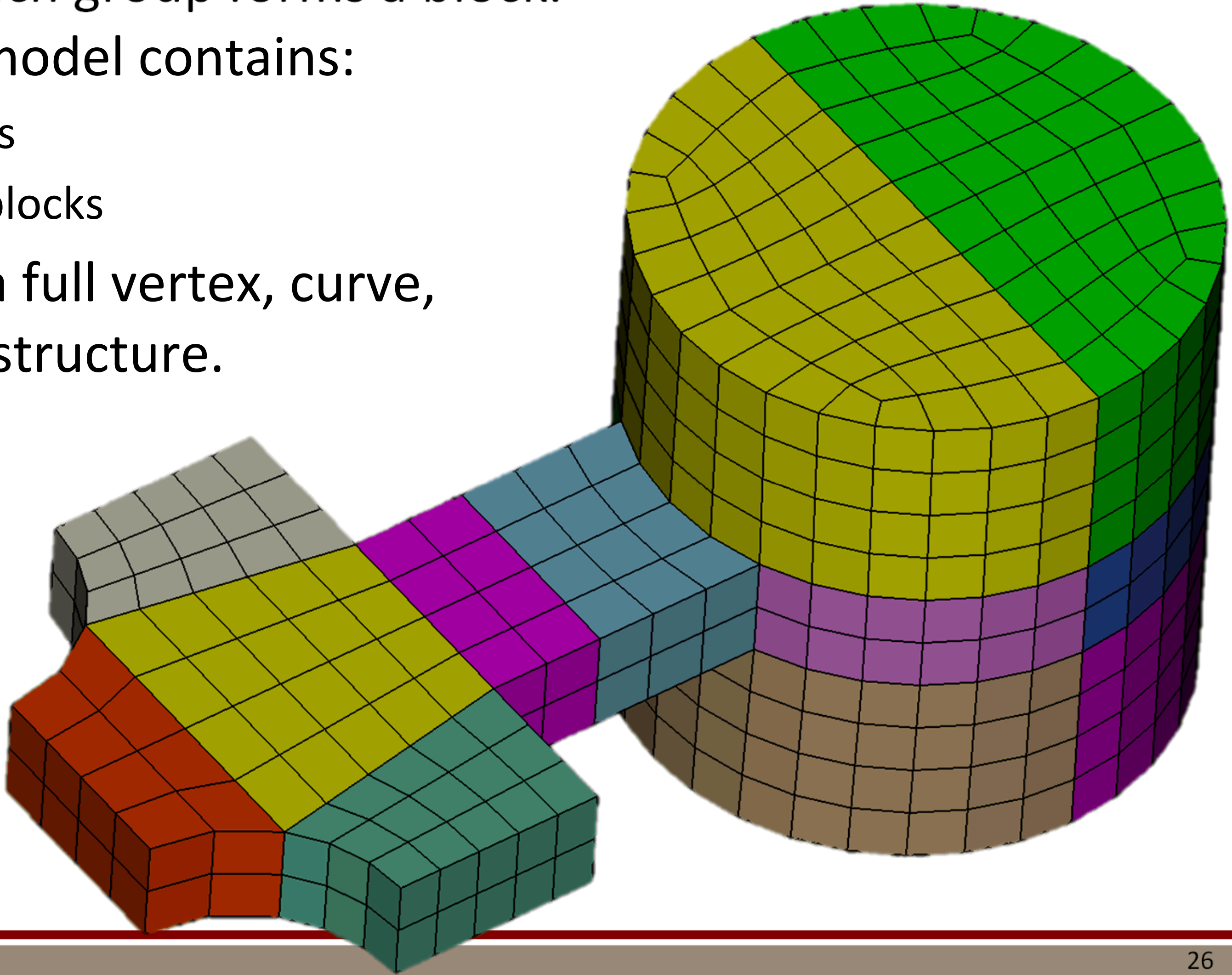


Steps to Mesh Scaling

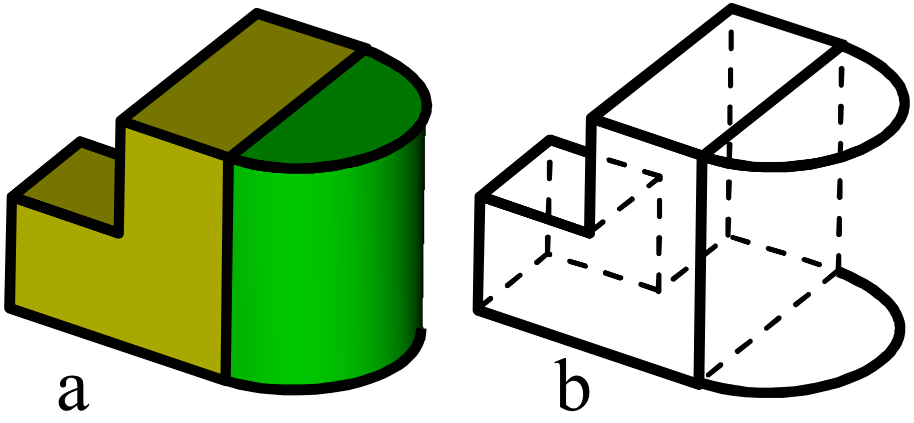
- STEP 1 : Identify Swept Mesh Topology
- STEP 2 : Identify Constraints
- STEP 3 : Propagate Block Constraints
- STEP 4 : Construct Block Decomposition
- STEP 5 : Extract Sizing Information
- STEP 6 : Delete Original Mesh
- STEP 7 : Compute New Curve Intervals
- STEP 8 : Remesh Block Decomposition

STEP 4 : Construct Block Decomposition

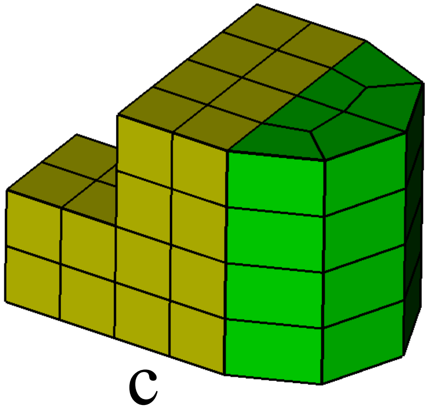
- Partition hexes into groups constrained to not cross over any quad in Q_C. Each group forms a block.
- This example model contains:
  - 6 swept blocks
  - 6 structured blocks
- We construct a full vertex, curve, surface, block structure.



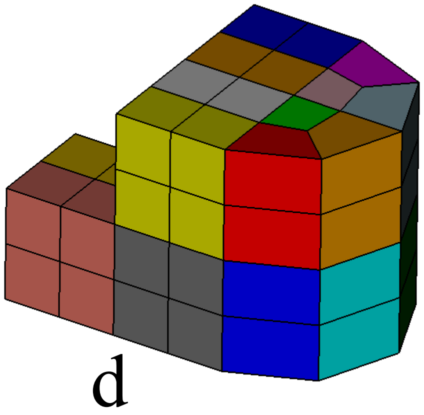
Additional Example of Block Decomposition



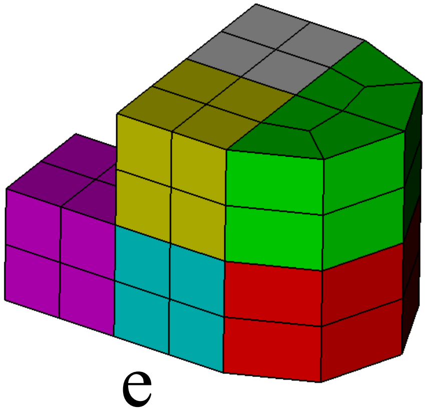
Initial CAD Model



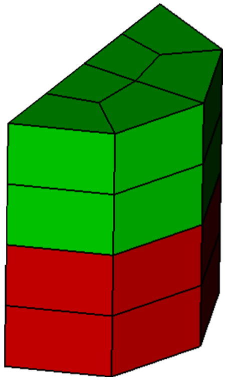
Initial Mesh



Block Decomposition if we only allow structured blocks



Block Decomposition if we allow swept blocks



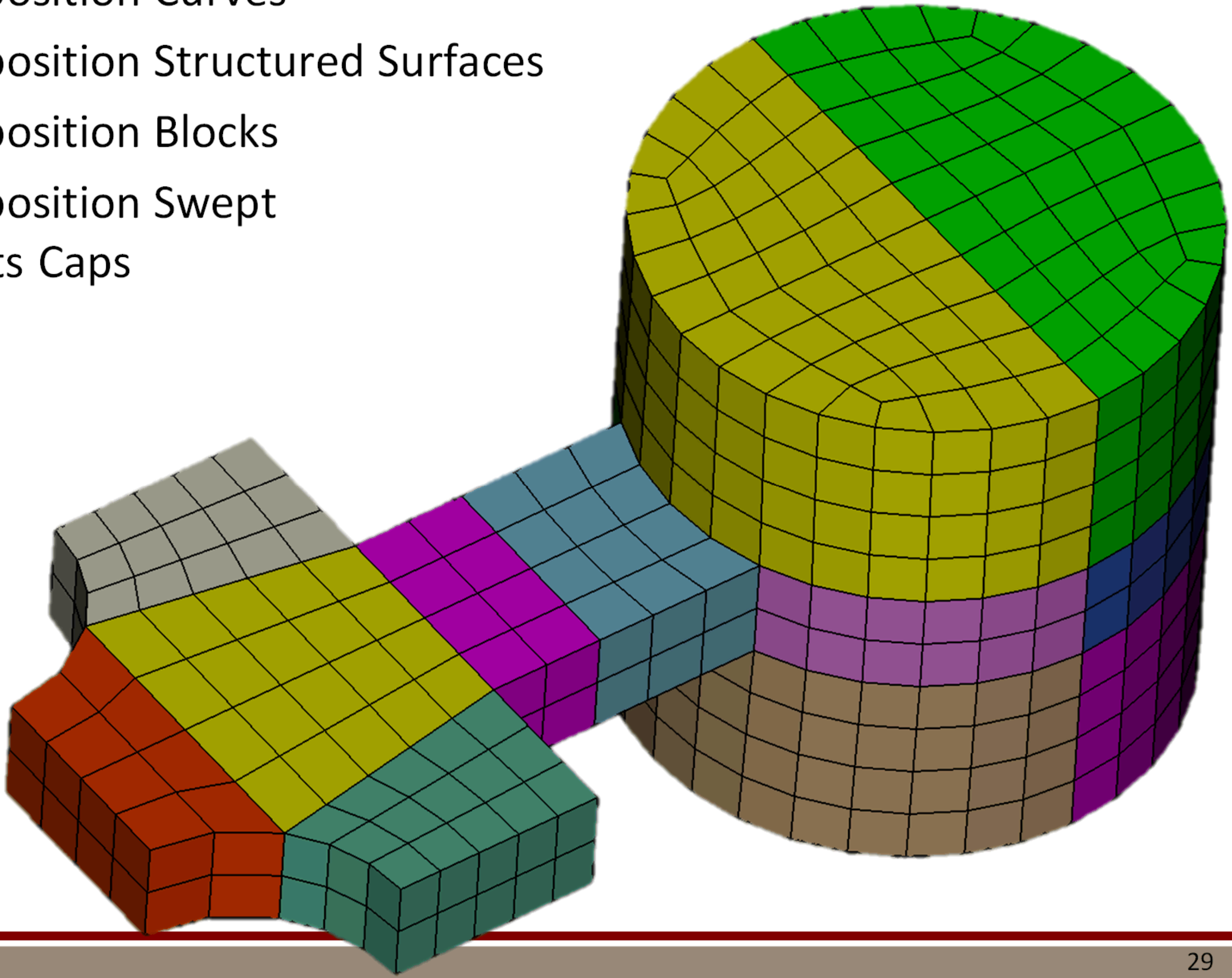
2 inter-dependent swept blocks

Steps to Mesh Scaling

- STEP 1 : Identify Swept Mesh Topology
- STEP 2 : Identify Constraints
- STEP 3 : Propagate Block Constraints
- STEP 4 : Construct Block Decomposition
- STEP 5 : Extract Sizing Information
- STEP 6 : Delete Original Mesh
- STEP 7 : Compute New Curve Intervals
- STEP 8 : Remesh Block Decomposition

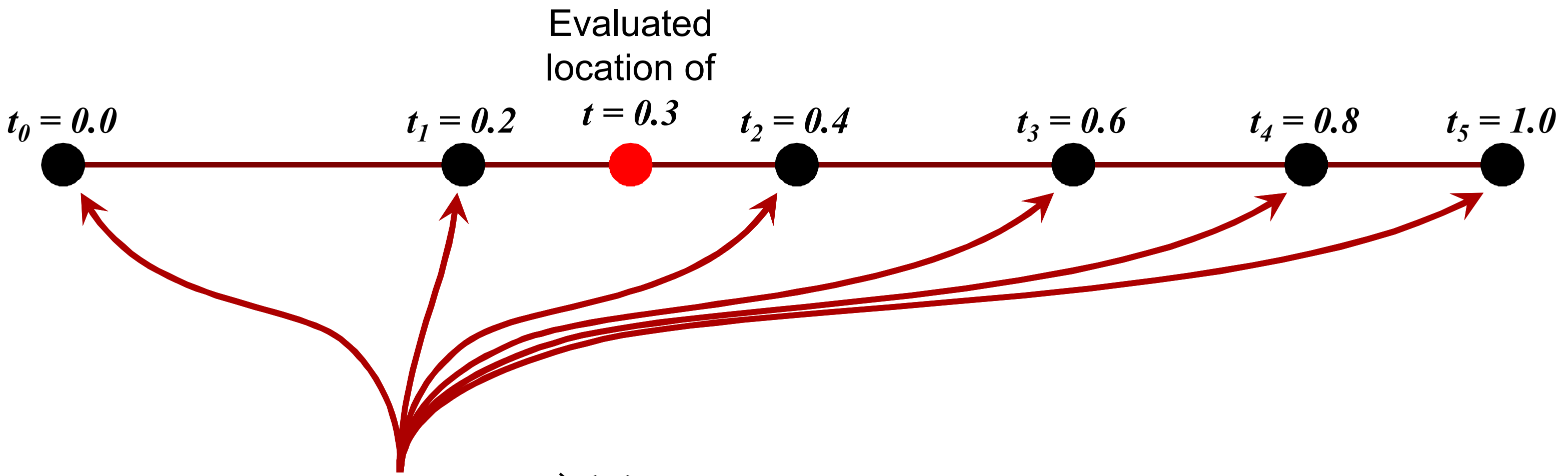
STEP 5 : Extract Sizing Information

- Sizing is stored for:
  - Block Decomposition Curves
  - Block Decomposition Structured Surfaces
  - Block Decomposition Blocks
  - Block Decomposition Swept Source/Targets Caps



STEP 5 : Extract Sizing Information

Sizing for Decomposition Curves and Structured Surfaces and Volumes



- Location of nodes, $\vec{x}(t)$, in the original mesh on curves are stored and assigned a t parameter based on its topological location along the curve.
- Evaluations are piecewise linear.

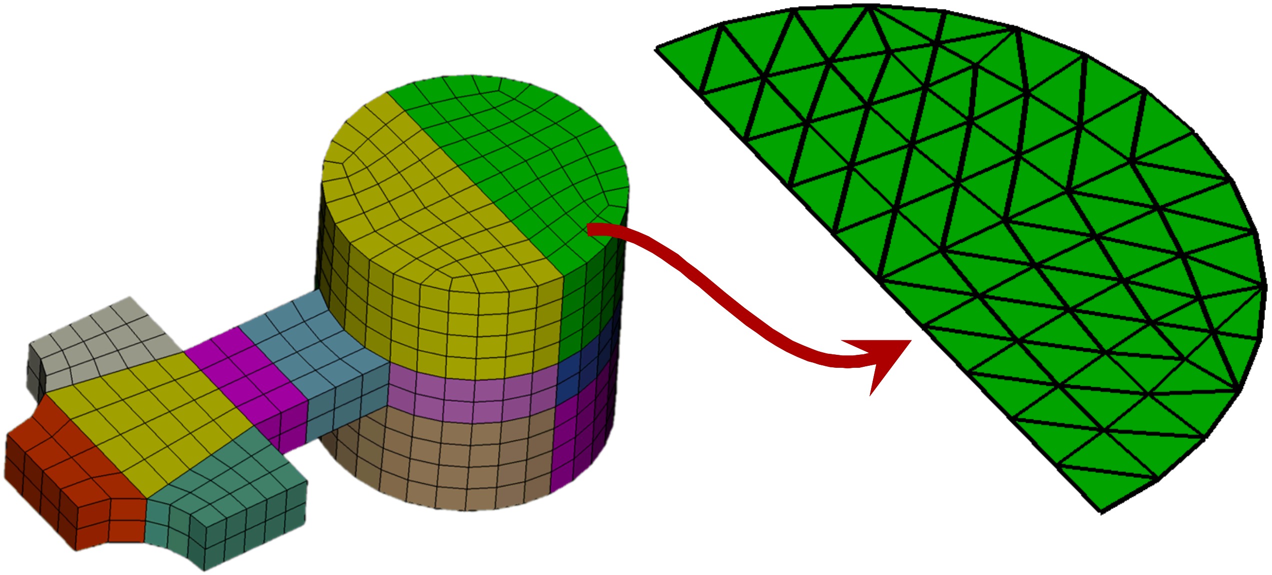
$$\vec{x}(t_{new}) = \left(\frac{t_b - t_{new}}{t_b - t_a}\right)\vec{x}(t_a) + \left(\frac{t_{new} - t_a}{t_b - t_a}\right)\vec{x}(t_b)$$

- 2D and 3D versions of this parameterization are used for structured surfaces and blocks.

STEP 5 : Extract Sizing Information

Sizing for Decomposition Source/Target Cap Surfaces

- A sizing function is constructed for each sweep block cap surface.
  - We construct a tri mesh by splitting the quads from the original mesh
  - At each node in the tri mesh, we store the average size of adjacent edges from the original mesh, multiplied by the cubed-root of the multiplier.
  - During remeshing, we query the sizing function to ensure scaled mesh roughly approximates size gradations from original mesh.



f(x,y,z) = Scaled Size

Steps to Mesh Scaling

- STEP 1 : Identify Swept Mesh Topology
- STEP 2 : Identify Constraints
- STEP 3 : Propagate Block Constraints
- STEP 4 : Construct Block Decomposition
- STEP 5 : Extract Sizing Information
- STEP 6 : Delete Original Mesh
- STEP 7 : Compute New Curve Intervals
- STEP 8 : Remesh Block Decomposition

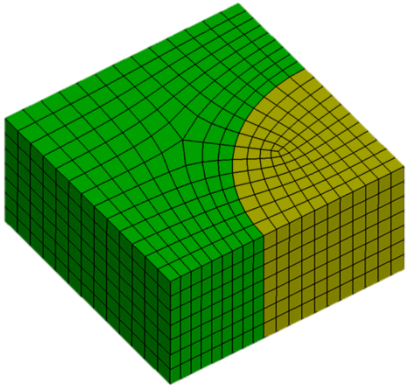
Steps to Mesh Scaling

- STEP 1 : Identify Swept Mesh Topology
- STEP 2 : Identify Constraints
- STEP 3 : Propagate Block Constraints
- STEP 4 : Construct Block Decomposition
- STEP 5 : Extract Sizing Information
- STEP 6 : Delete Original Mesh
- STEP 7 : Compute New Curve Intervals
- STEP 8 : Remesh Block Decomposition

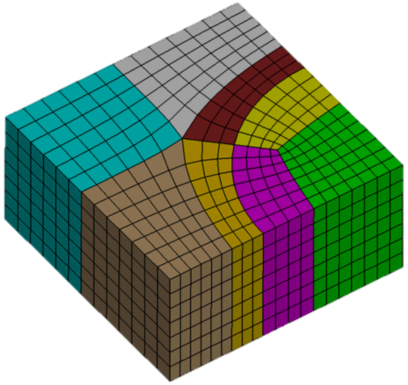
STEP 7 : Compute New Curve Intervals

Define Curve Groups

- Intervals = number of mesh edges on a curve in the block decomposition. New intervals must be assigned based on ***Multiplier***.
- We first define curve groups. Each curve group contains a set of edges which are constrained to have the same number of intervals.

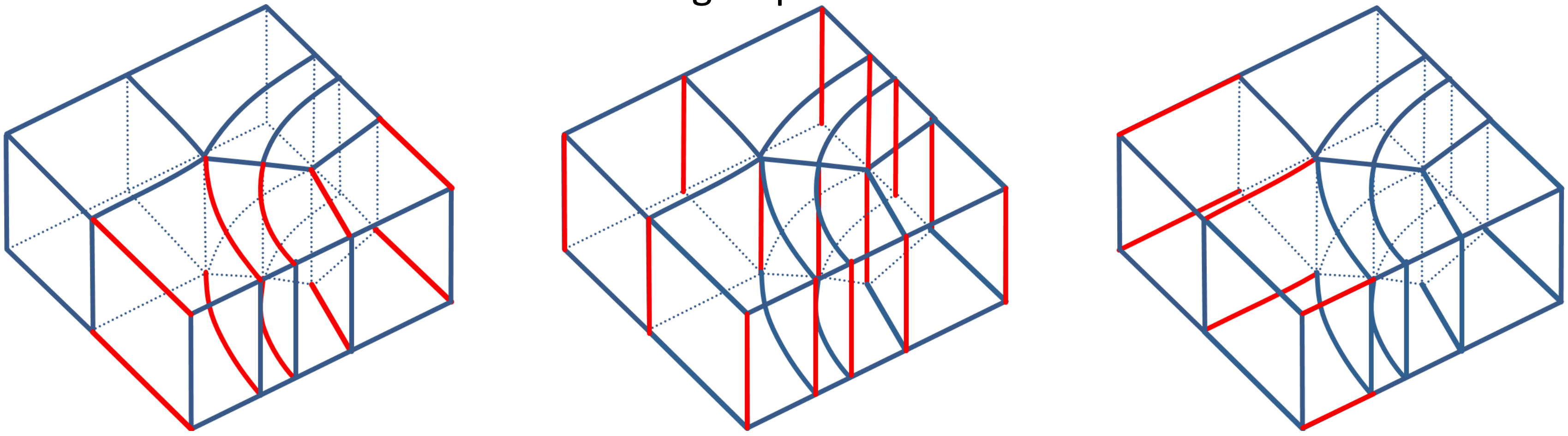


Initial Mesh



Block Decomposition

3 of the 7 curve groups in this model.



STEP 7 : Compute New Curve Intervals

Curve Multiplier

- All curves in curve group ***g*** must have the same number of intervals
- The input ***Multiplier***, multiplies the number of hex elements (3D).

$$NumHex_{scaled}^{model} = \boldsymbol{Multiplier} \times NumHex_{orig}^{model}$$

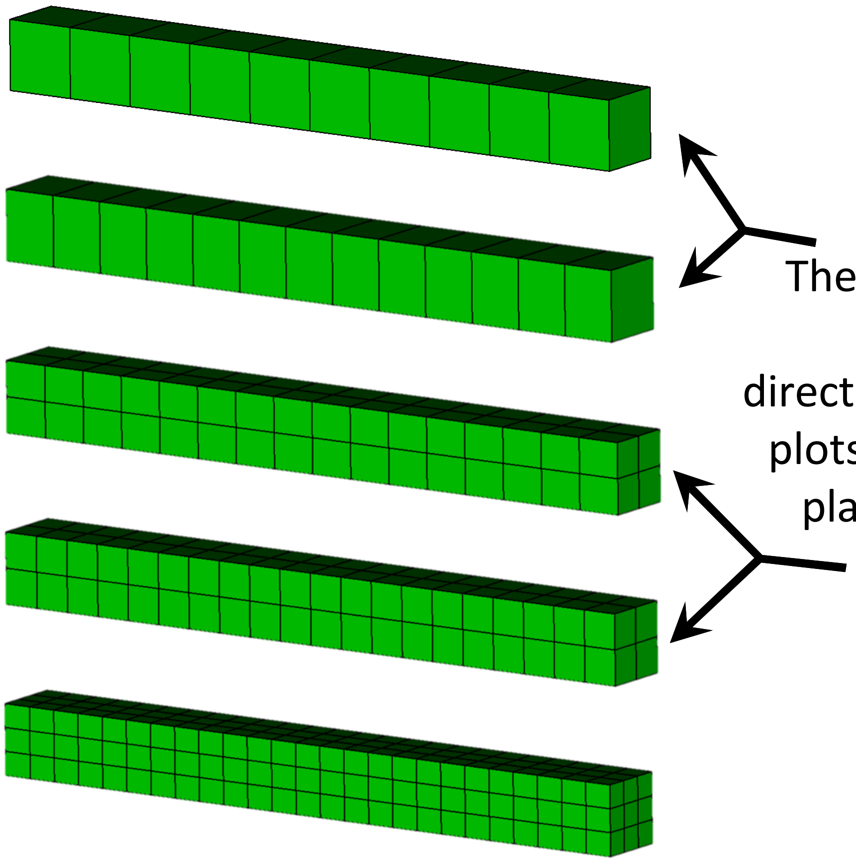
- It can be shown that the number of intervals on a curve must increase by the cubed-root of the ***Multiplier***. The simplistic equation to compute the new intervals on curve group ***g*** is then:

$$Intervals_{scaled}^{g} = \sqrt[3]{\boldsymbol{Multiplier}} \times Intervals_{orig}^{g}$$

- But it is not quite that easy, because $Intervals_{scaled}^{g}$ must be an integer.

Can We Simply Round-off to Nearest Integer?

$1 * \sqrt[3]{2} = 1.26$, rounds to 1!



Initial Mesh: 1x1x10:

2X scaled mesh: 1x1x13

4X scaled mesh: 2x2x16

8X scaled mesh: 2x2x20

16X scaled mesh: 3x3x25

These 2 meshes only change in one direction. Convergence plots will likely have a plateau! Same with these 2 meshes.

We add *Minimum Interval* parameter to ensure a minimum of 1 increase from mesh to mesh.

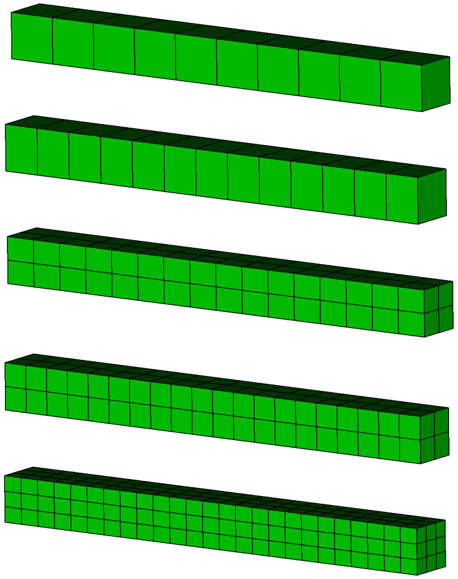
STEP 7 : Compute New Curve Intervals

Minimum Interval Parameter

- ***MinInt*** = positive integer parameter {0, 1, 2, …}

$$Intervals^{g}_{min} = MAXIMUM\left(\left\lfloor \sqrt[3]{\boldsymbol{Multiplier}} \times Intervals^{i}_{orig} \right\rceil, \boldsymbol{MinInt}\right)$$

Simple Round-off



Initial: 10 hex

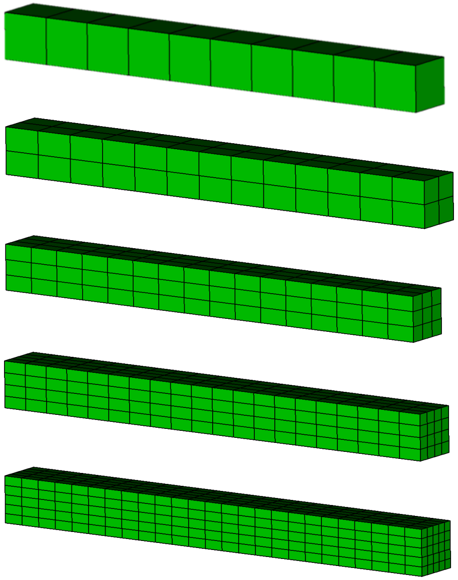
2X: 13 hex

4X: 64 hex

8X: 80 hex

16X: 225 hex

Using ***MinInt***



Initial: 10 hex

(2X, 1): 52 hex

(4X, 2): 144 hex

(8X, 3): 320 hex

(16X, 4): 625 hex

Using ***MinInt*** ensures that each mesh in the series is slightly finer in all 3 directions at every location, at the expense of overshooting the desired ***Multiplier***.

STEP 7 : Compute New Curve Intervals

Interval Assignment

- If there are no swept blocks (i.e. no paved sweep caps), then the intervals on each curve group are independent of all other curve groups, and can be computed as explained already.

- If there are swept blocks, intervals on curve groups become inter-dependent in order to keep sum-even constraints on each sweep cap.

- We use Interval Assignment from Scott Mitchell (6th & 22nd IMR)
  - One variable for each curve group, $\boldsymbol{g}$.
    - $Intervals_{min}^{g} = MAXIMUM\left(\left\lfloor \sqrt[3]{\boldsymbol{Multiplier}} \times Intervals_{orig}^{i} \right\rfloor, \boldsymbol{MinInt}\right)$
    - $Intervals_{max}^{g} = Intervals_{min}^{g} + S$
  - One sum-even constraint for each interdependent group of sweep blocks.

$$\sum_{i=1}^{num\_curves\_on\_surface} Intervals_i^g = 2k \mid k \in integers \geq 2$$

Steps to Mesh Scaling

- STEP 1 : Identify Swept Mesh Topology
- STEP 2 : Identify Constraints
- STEP 3 : Propagate Block Constraints
- STEP 4 : Construct Block Decomposition
- STEP 5 : Extract Sizing Information
- STEP 6 : Delete Original Mesh
- STEP 7 : Compute New Curve Intervals
- STEP 8 : Remesh Block Decomposition

STEP 8 : Remesh Block Decomposition

Remeshing Structured Entities

Remeshing of the block decomposition is done in the following steps:

1. Place new node at location of each block decomposition vertex.

2. Place $\boldsymbol{n}$ mesh edges on each curve in the block decomposition, where $\boldsymbol{n}$ is the number of intervals assigned to the corresponding curve group during STEP 7.
   - Place nodes at $\frac{1}{n}, \frac{2}{n}, \ldots \frac{(n-1)}{n}$

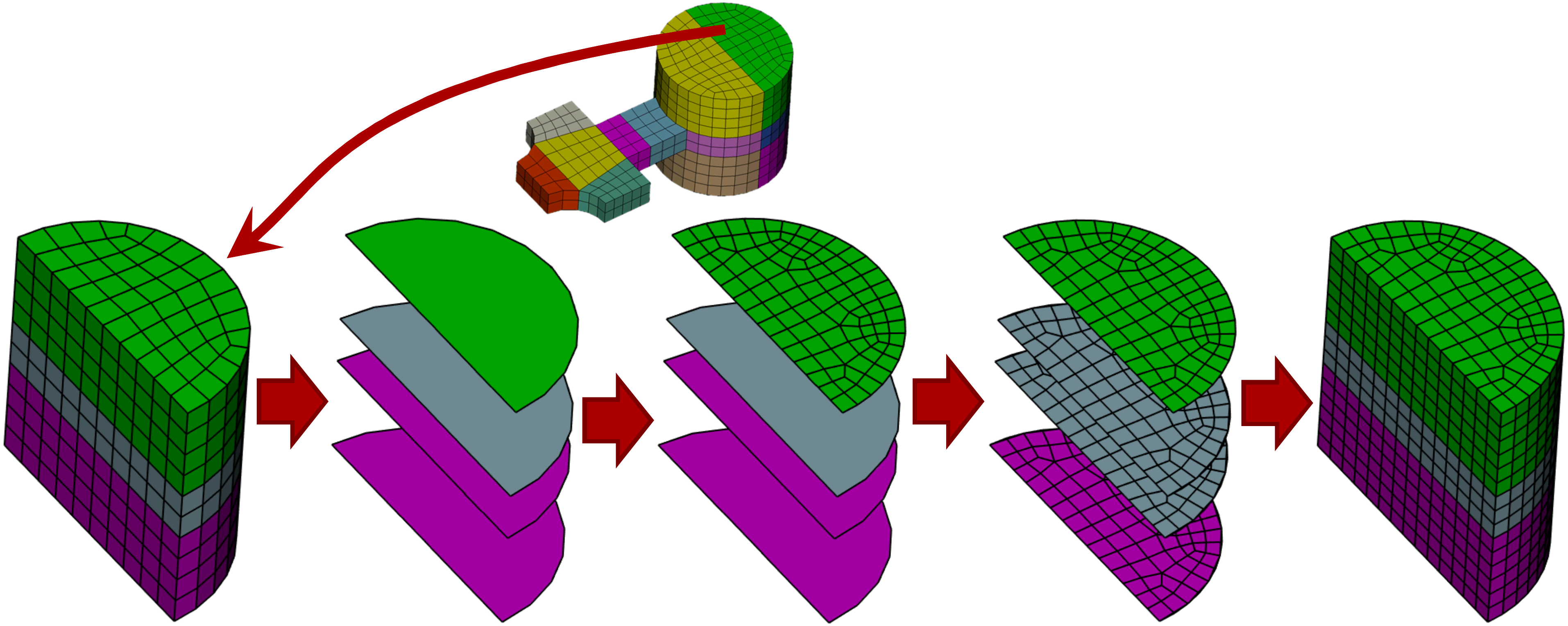
3. Place $\boldsymbol{m}$ X $\boldsymbol{n}$ quads on each structured surface (i.e. adjacent to a structured block or linking surface of a swept block), where $\boldsymbol{m}$ and $\boldsymbol{n}$ are the number of intervals assigned to the 2 corresponding curve groups.
   - Place nodes at $\left(\frac{1}{m}, \frac{1}{n}\right), \left(\frac{2}{m}, \frac{1}{n}\right), \left(\frac{2}{m}, \frac{2}{n}\right), \ldots \left(\frac{(m-1)}{m}, \frac{(n-1)}{n}\right)$

4. Place $\boldsymbol{m}$ X $\boldsymbol{n}$ X $\boldsymbol{o}$ hexes in each structured block in a similar manner.

STEP 8 : Remesh Block Decomposition

Remeshing Swept Blocks

Remesh one cap surface for each inter-dependent group of swept blocks. Copy mesh to other caps. Then sweep each one-to-one.



Initial mesh on inter-dependent sweep blocks

Four inter-dependent cap surfaces

Mesh one of the cap surfaces

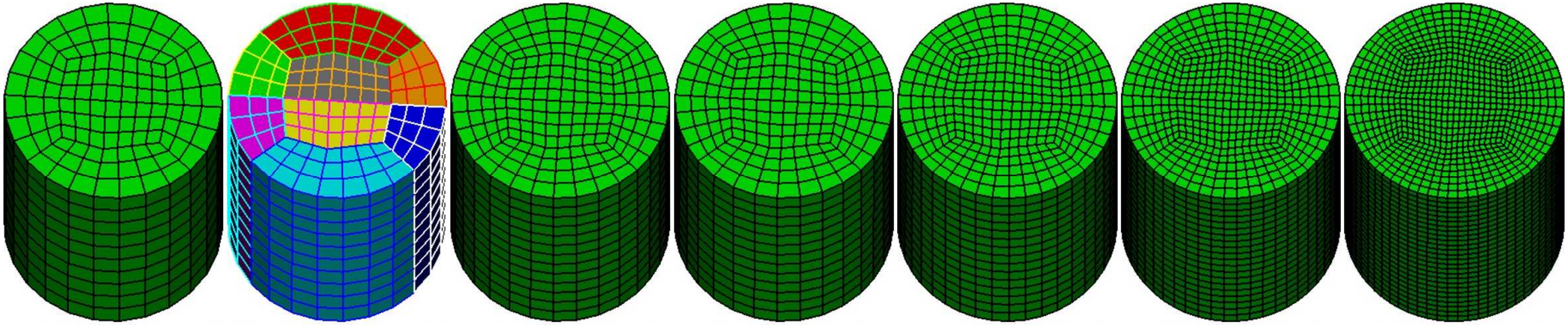
Copy mesh to other cap surfaces

3 independent one-to-one sweeps

Structured vs. Swept Blocks

Initial Mesh has a lot of structure

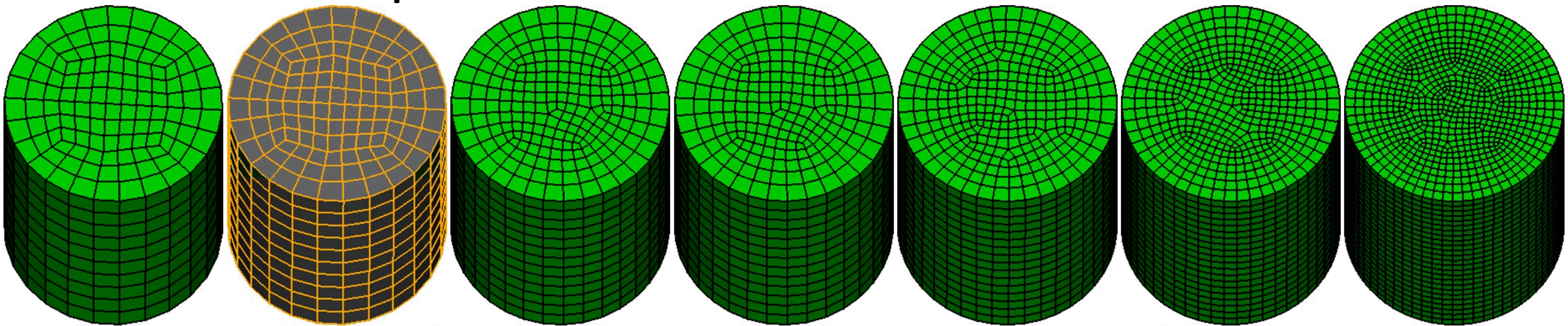
Scale with structured block only:



a) Original b) Block Decomposition c) 2X, MinInt 0 d) 2X MinInt 1 e) 4X MinInt 2 f) 8X MinInt 3 g) 16X MinInt 4

Structured Blocks maintains that structure

Scale with swept blocks:



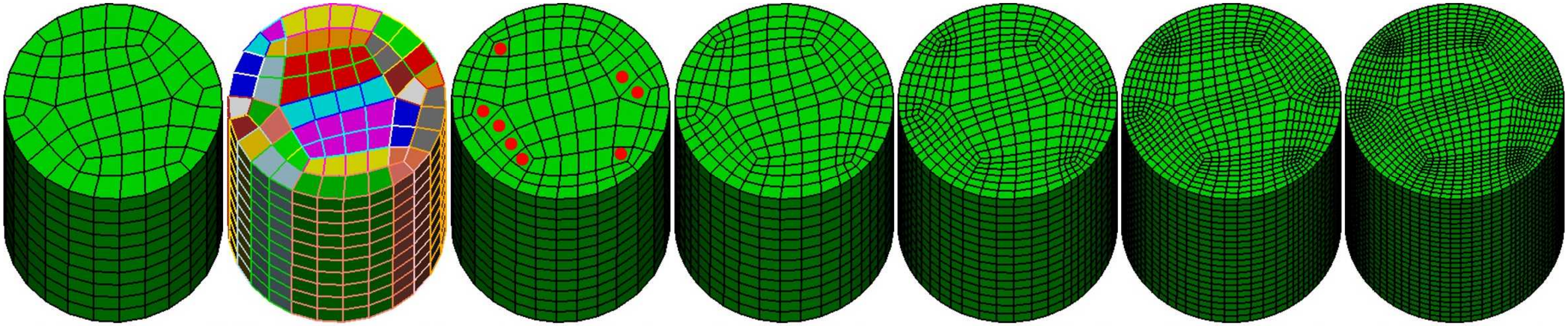
a) Original b) Block Decomposition c) 2X, MinInt 0 d) 2X MinInt 1 e) 4X MinInt 2 f) 8X MinInt 3 g) 16X MinInt 4

Swept Blocks destroys structure

Structured vs. Swept Blocks

Initial Mesh is unstructured (paved)

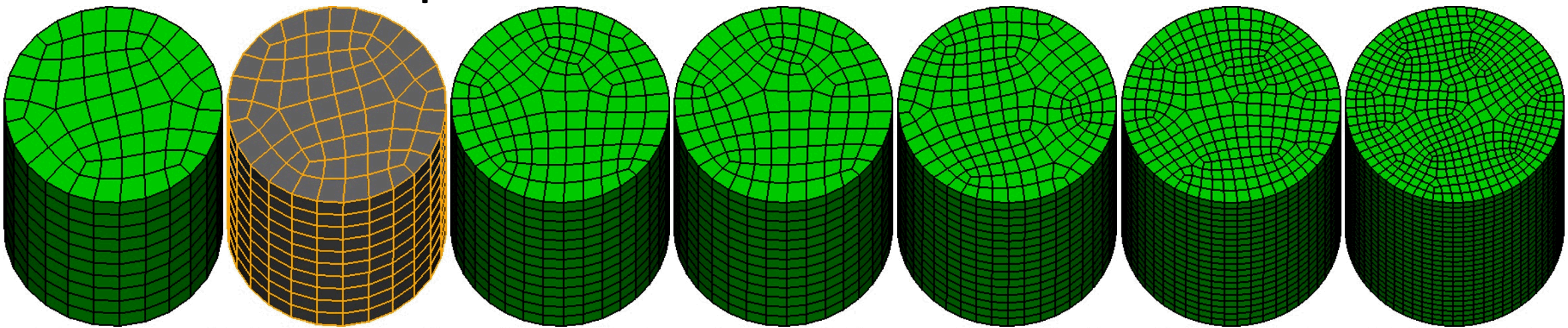
Scaled with structured blocks only:



a) Original b) Block Decomposition c) 2X, MinInt 0 d) 2X MinInt 1 e) 4X MinInt 2 f) 8X MinInt 3 g) 16X MinInt 4

Structured Blocks produces streaky, uneven mesh

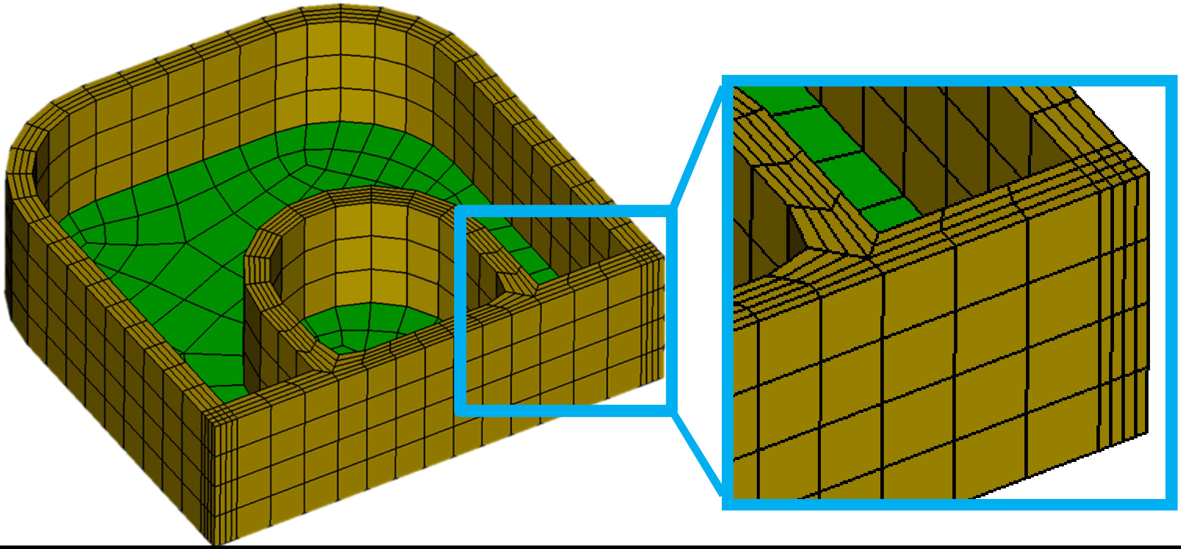
Scaled with swept blocks:



a) Original b) Block Decomposition c) 2X, MinInt 0 d) 2X MinInt 1 e) 4X MinInt 2 f) 8X MinInt 3 g) 16X MinInt 4

Swept Blocks produces nicely scaled mesh similar to original

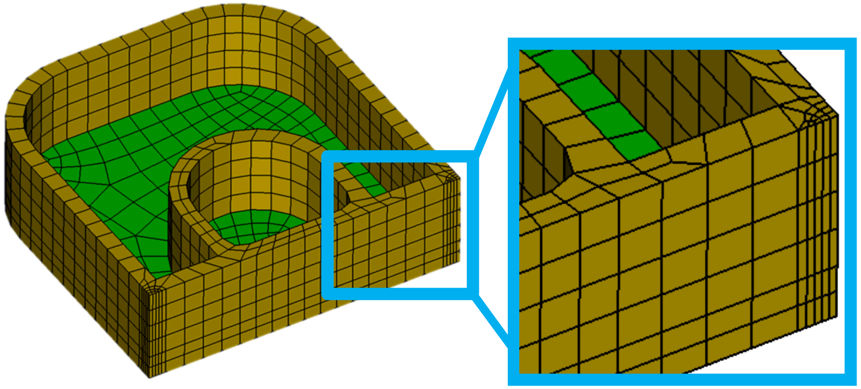
Mesh Scaling Can Destroy Mesh Structure



Initial Mesh: user has taken great care to build a structured mesh on rim.

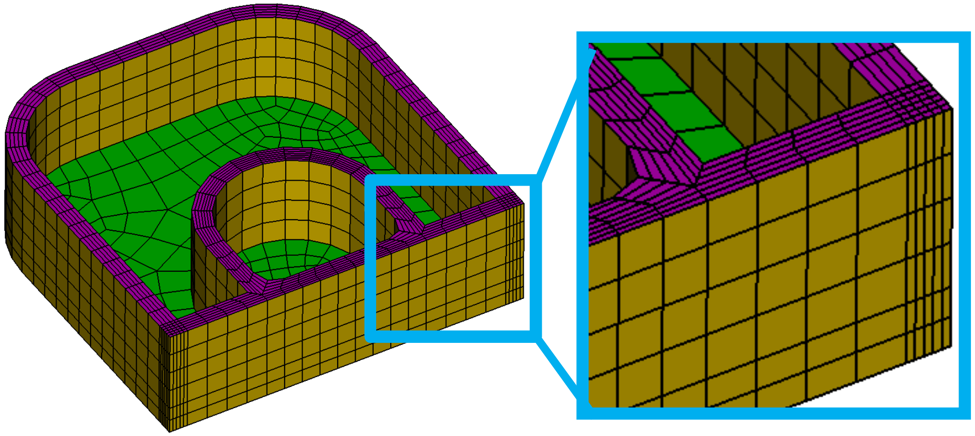
The yellow elements are identified as a single swept block. There are also 2 green swept blocks

Scaling with swept blocks



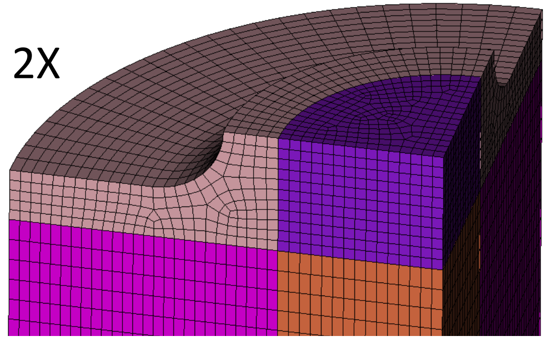
The source of each swept block is remeshed with the paver. The paver does not duplicate high aspect ratio structured meshes.

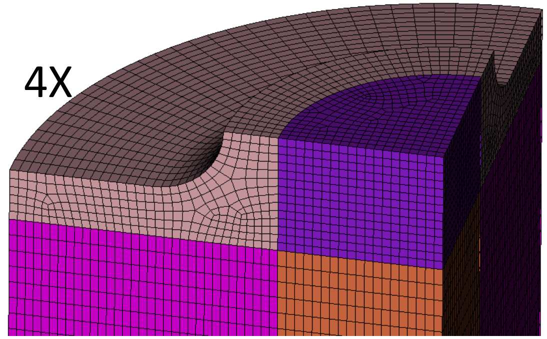
User can override and force structured blocks in specified zones

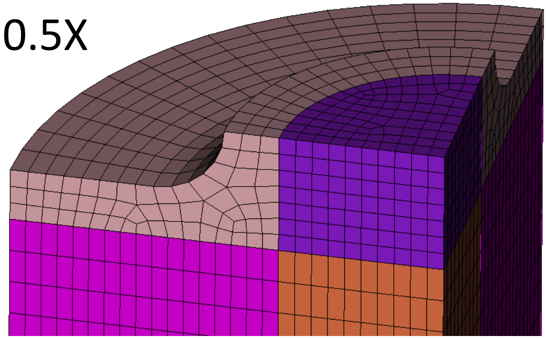


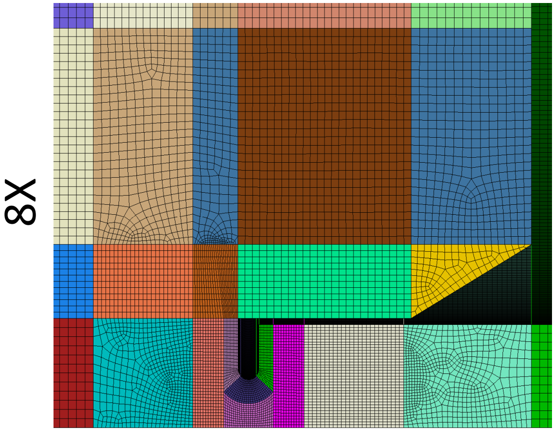
Mesh Scaling Enables Efficient, Robust Solution Verification

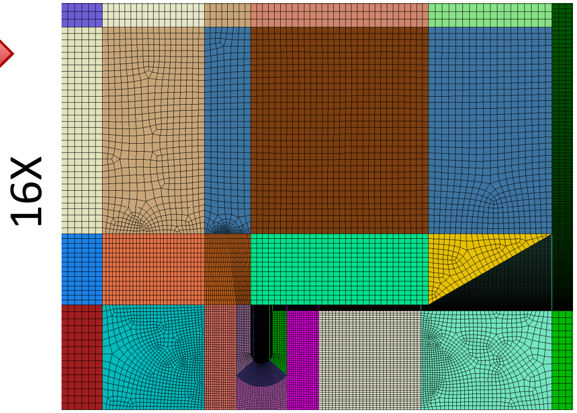
- Generate intermediate meshes other than 8x/64x scalings (efficiency)
- Coarsen the base mesh (efficiency)
- Generate additional meshes with only incrementally more elements (efficiency, robustness)







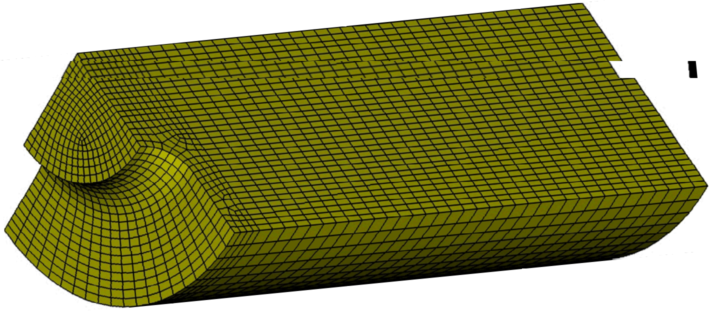




Can’t afford another 8x UMR, can just do a 2x mesh scaling

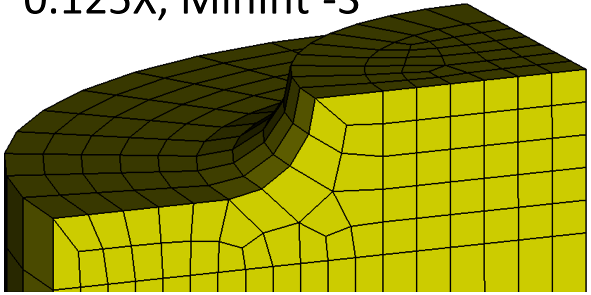
Notch Problem

- Tension test with 1/8 symmetry
- Material model was changed from Johnson-Cook to elastic-plastic, results are for implicit
- Two versions: explicit and implicit
- QoIs: Reaction force, max equivalent plastic strain.

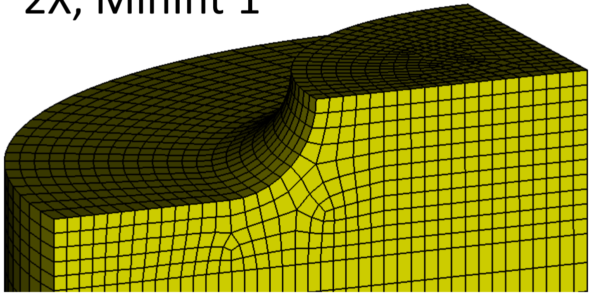


Original Mesh: 22,860 Hexes
1 element blocks
4 nodesets

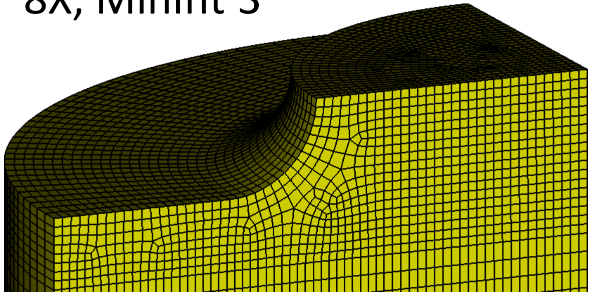
0.125X, MinInt -3



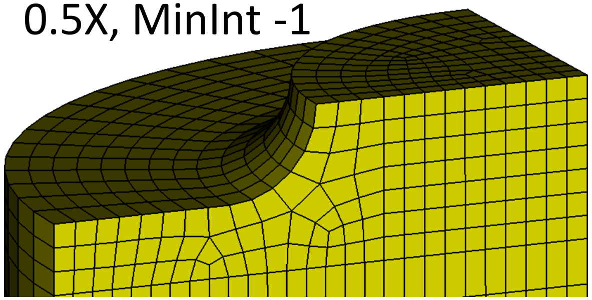
2X, MinInt 1



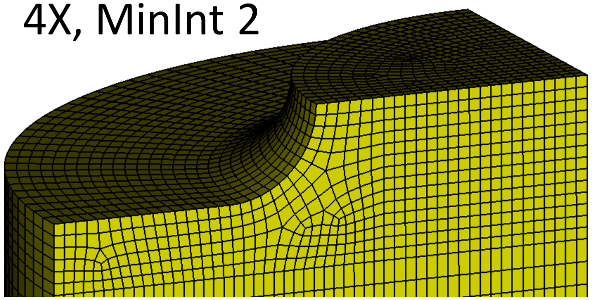
8X, MinInt 3



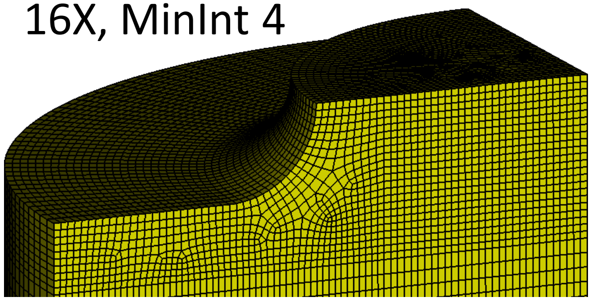
0.5X, MinInt -1



4X, MinInt 2

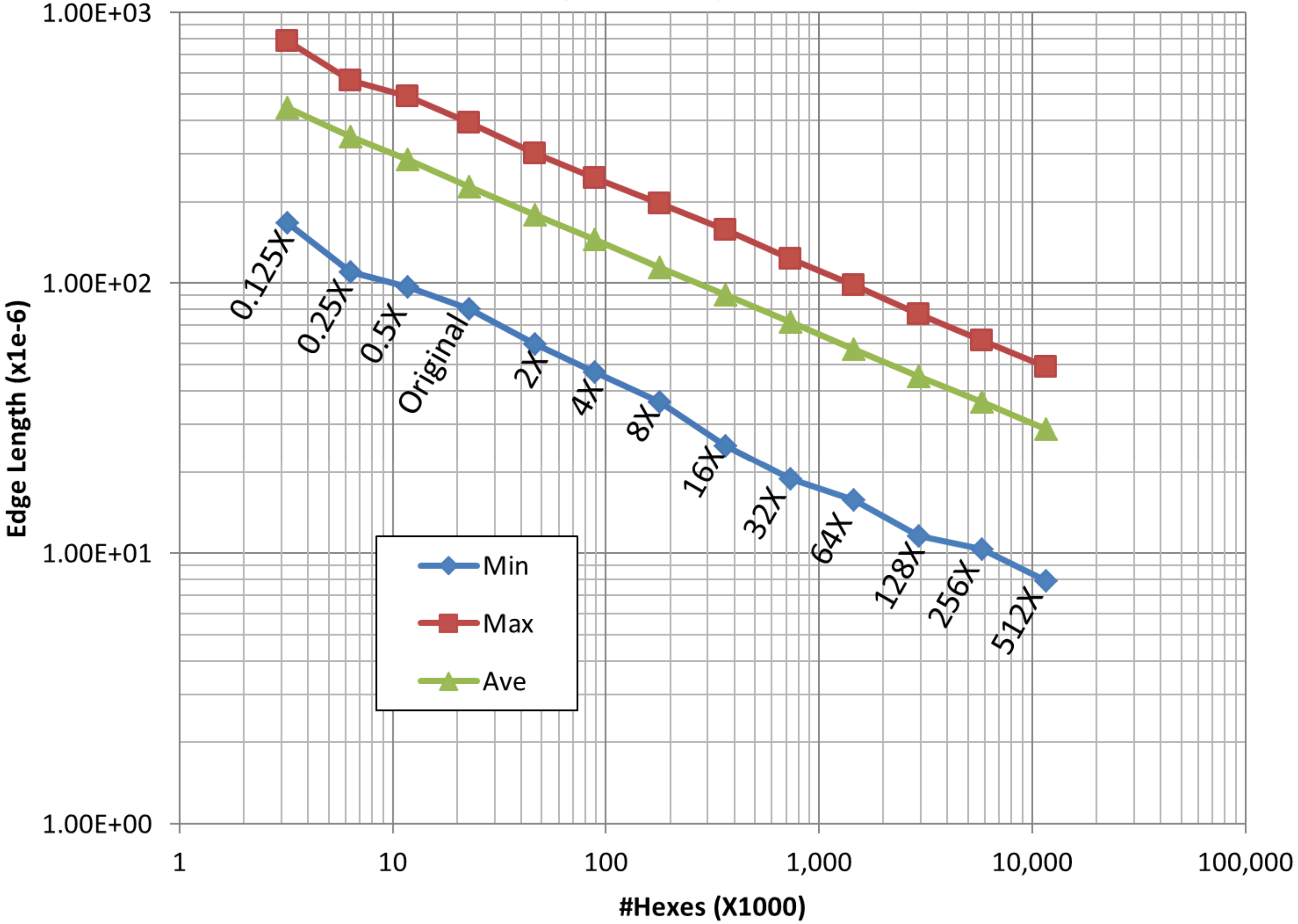


16X, MinInt 4



Notch Model

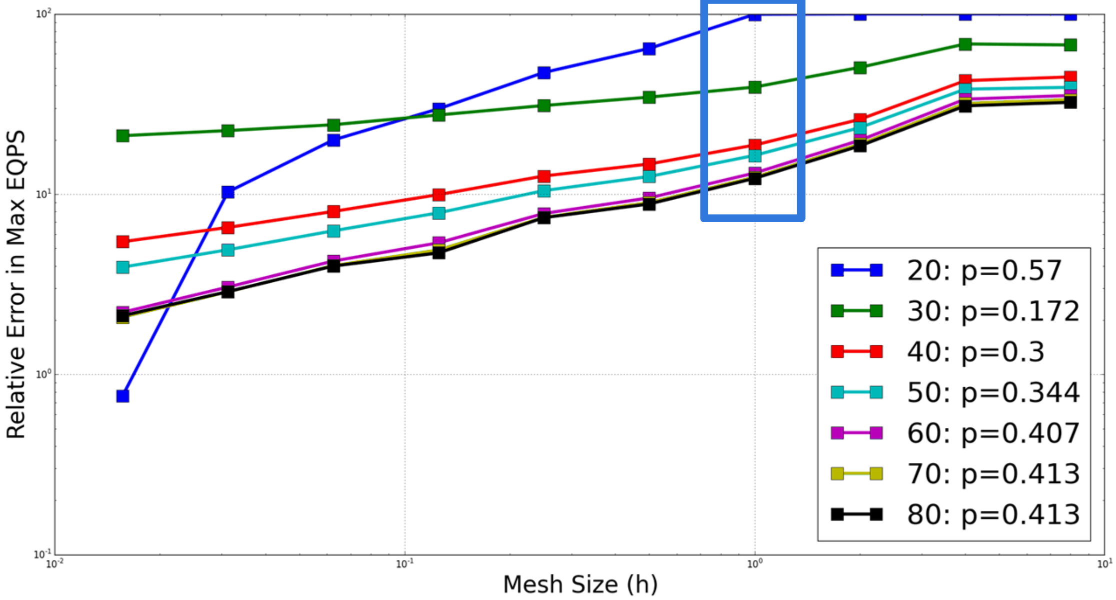
Edge Lengths

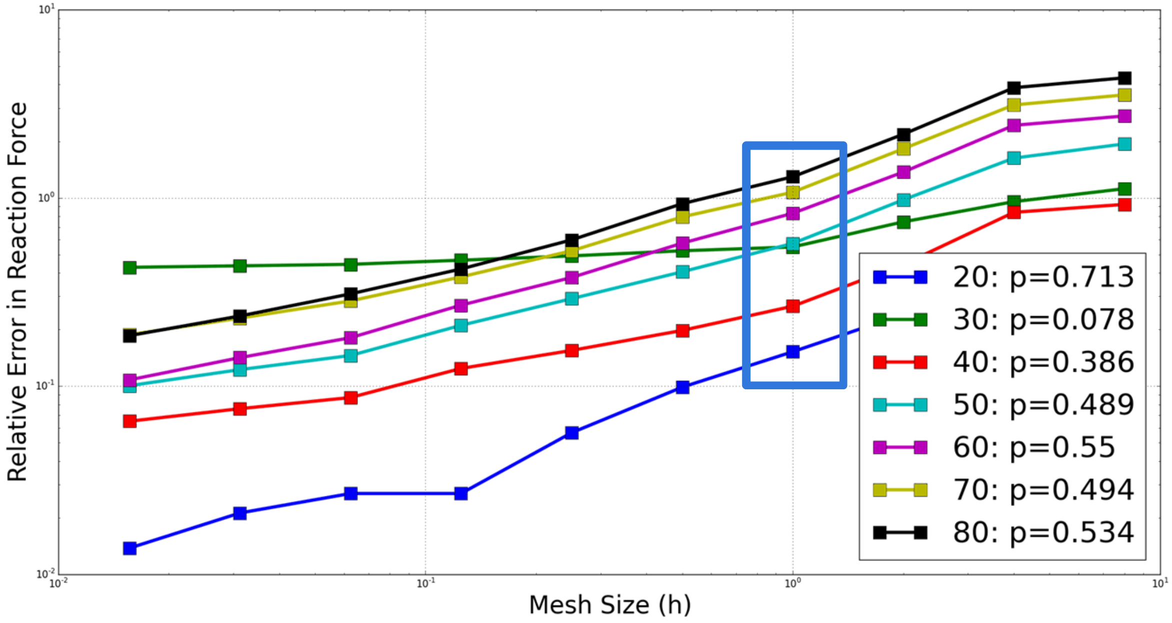


Geometric constraints & surface paver are reasons for variance from exact linear in Min/Max values, especially in coarsening.

Notch: Errors Using Multi-Mesh Extrapolation

- Extrapolation using nonlinear least squares approach (using base mesh plus 4 finer meshes)
- Estimated rates for max EQPS approach 0.4 at end of load steps
- Extrapolation provides useful error estimates

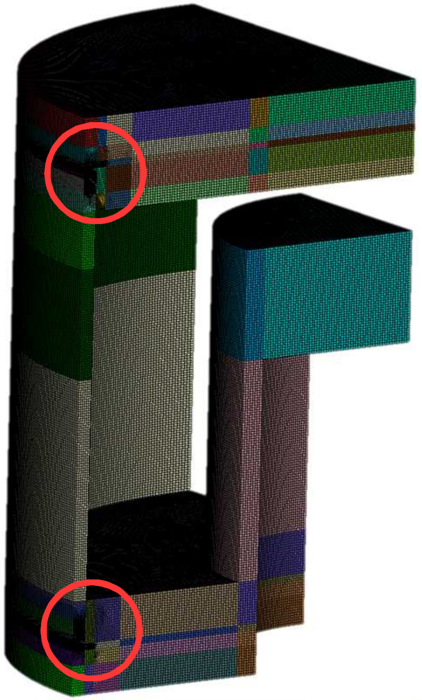


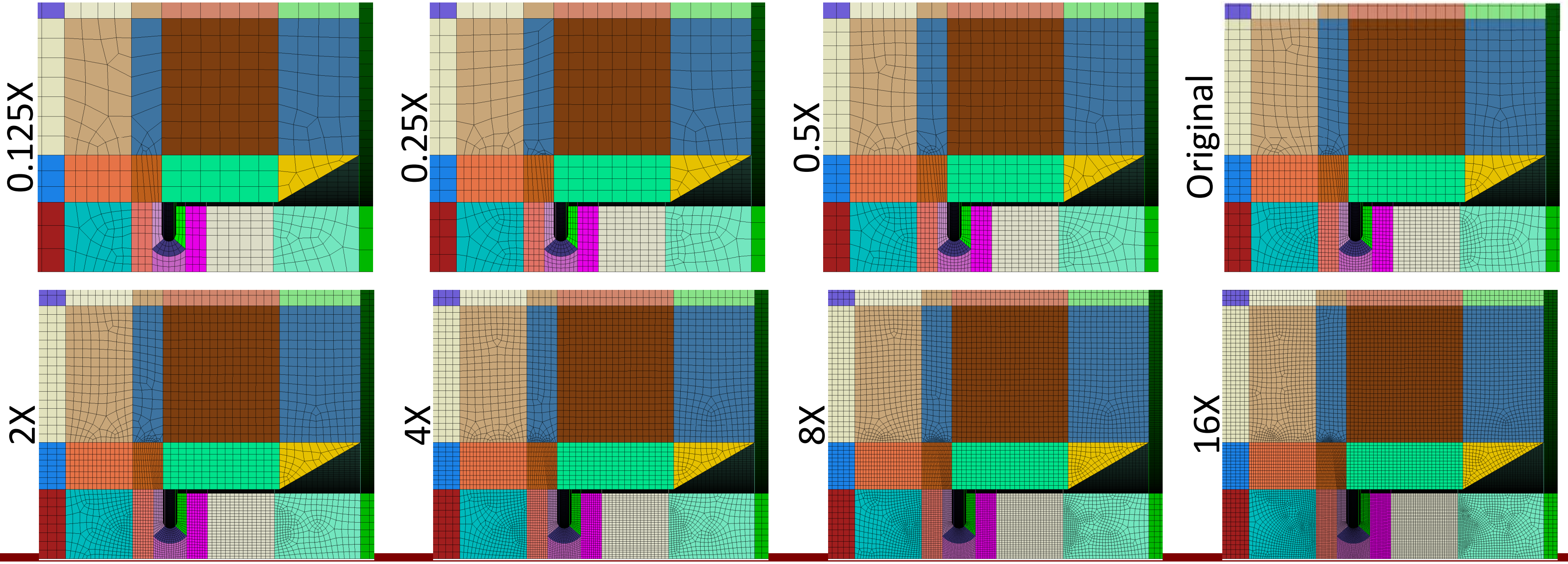


- Estimated rates for reaction force approach 0.5 at end of load steps
- Extrapolation provides useful error estimates
- Nominal mesh error: 0.1-2%

PCAP Thermal/Mechanical

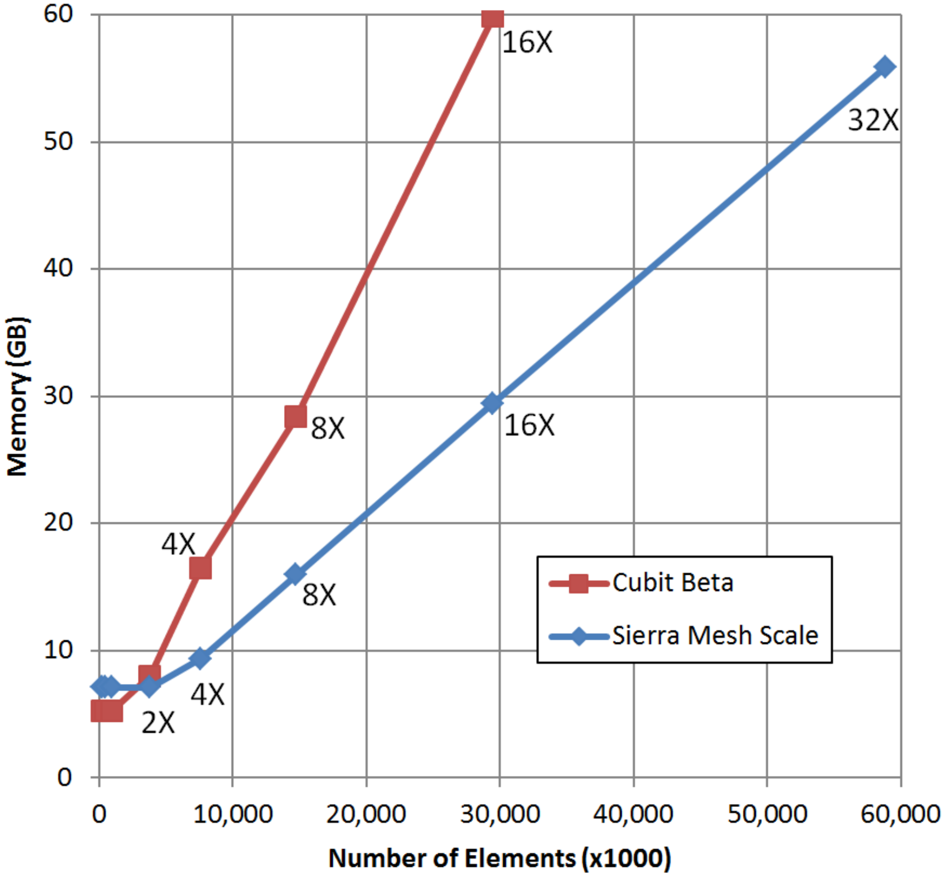
- Internal pressurization loading
- Material model is MLEP (Multi-Linear Elastic Plastic)
- Nominal mesh has 1.8M elements, 5 blocks, 1/4 symmetry
- QoIs: Max EQPS, max tearing parameter (in weld regions)





PCAP: Memory High Watermark

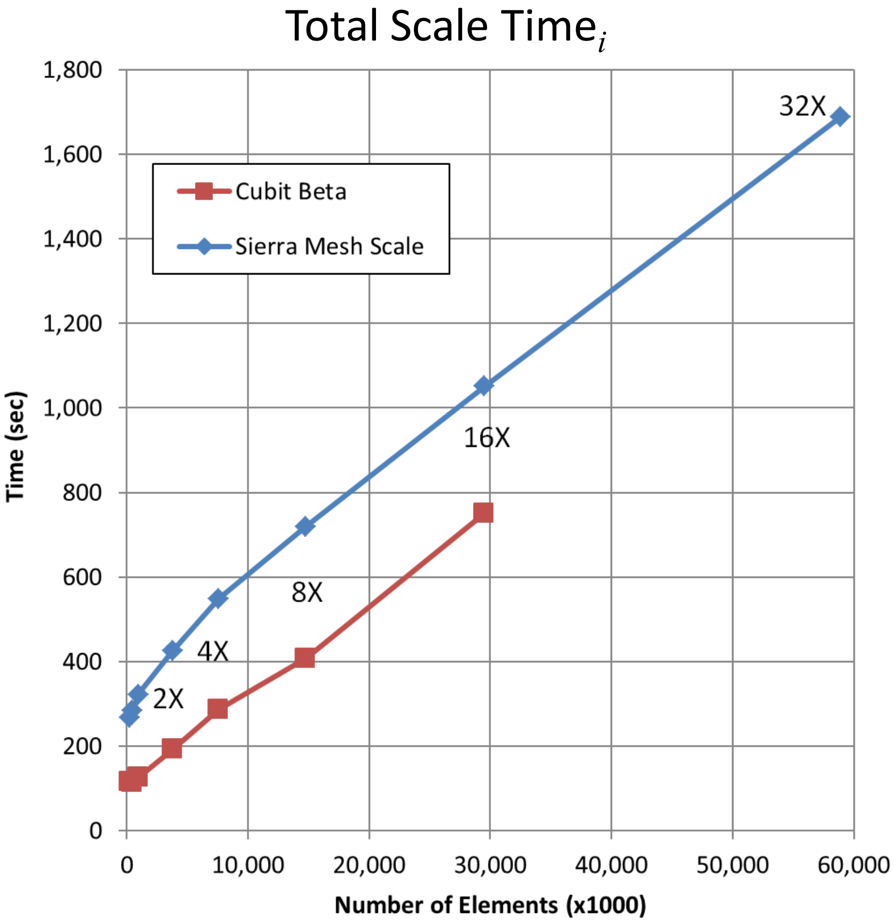
- Doubling the mesh with a 2X multiplier, roughly doubles the required memory
- Coarsening the mesh has constant memory requirement, governed by the input mesh.
- Cubit takes roughly twice as much memory as Sierra Mesh Scale



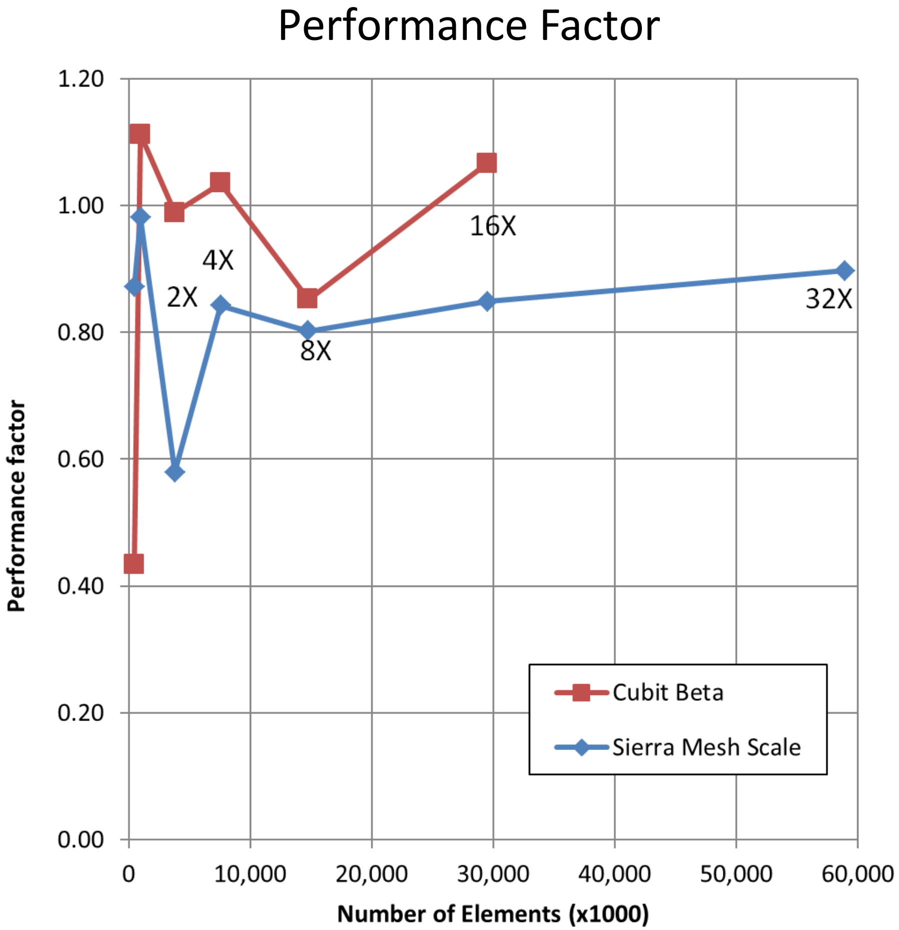
PCAP : Performance

$$Total\ Scale\ Time_i = Time_C + Time_i$$

$$Performance\ Factor_i = \frac{{Time_i}/{Time_{i-1}}}{{\#Elems_i}/{\#Elems_{i-1}}} = \frac{time\ to\ remesh\ block\ decomposition\ doubles}{\#\ elems\ in\ scaled\ mesh\ doubles} \approx 1.0$$

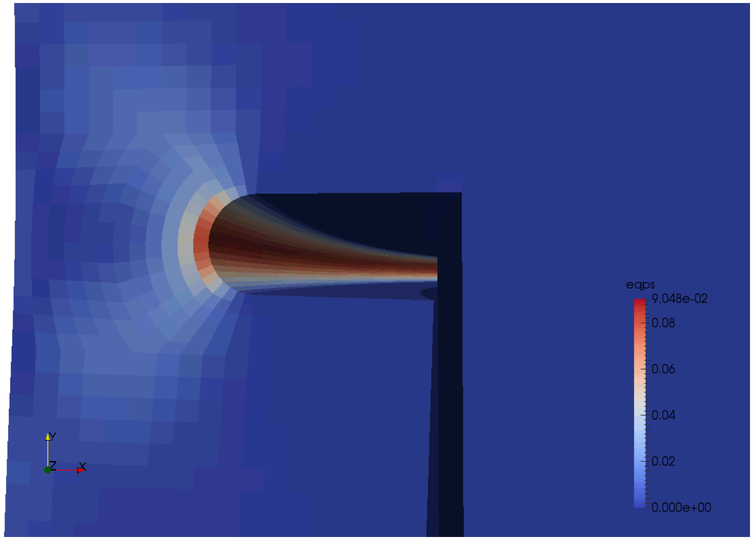


Total Scale Time$_i$

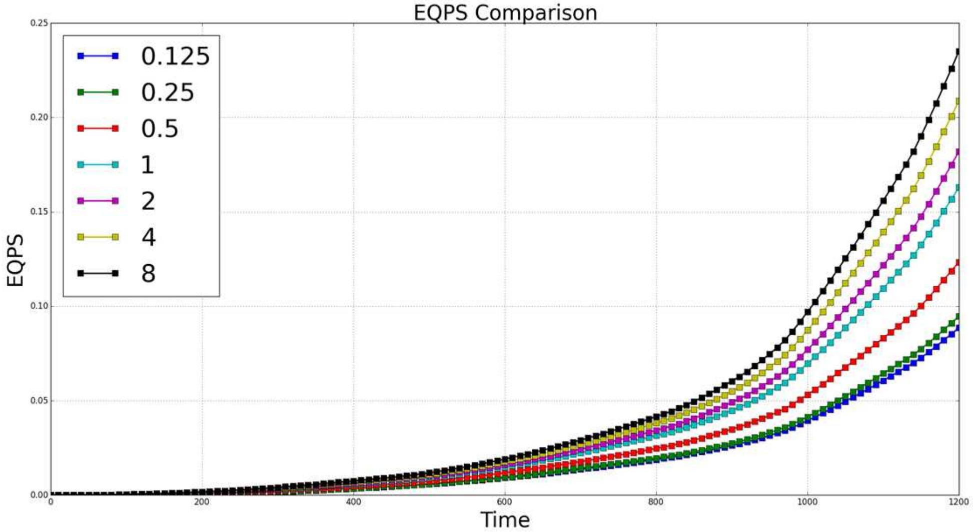


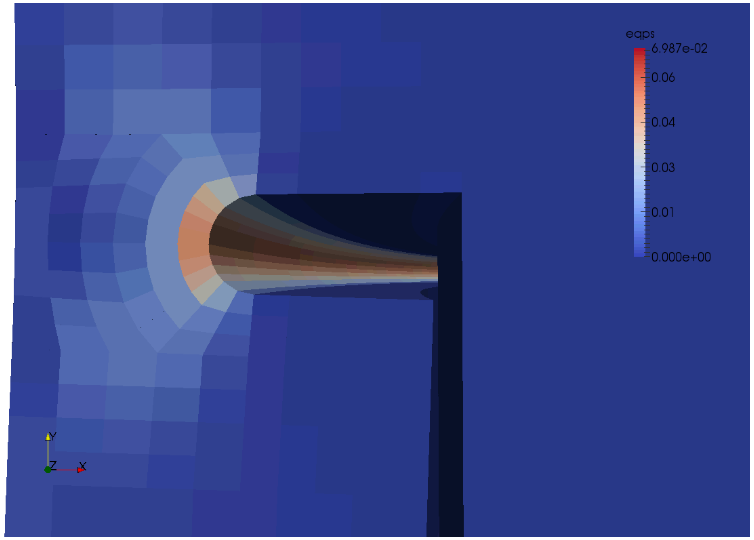
Performance Factor

PCAP Mechanical: Max EQPS

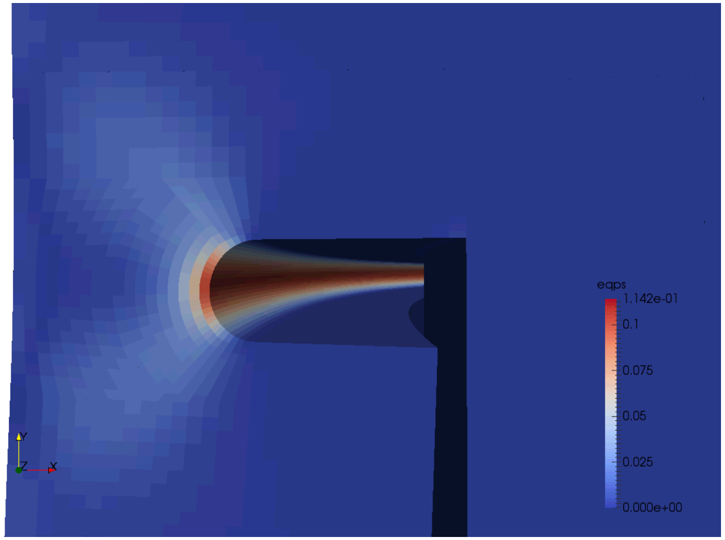


Base Mesh





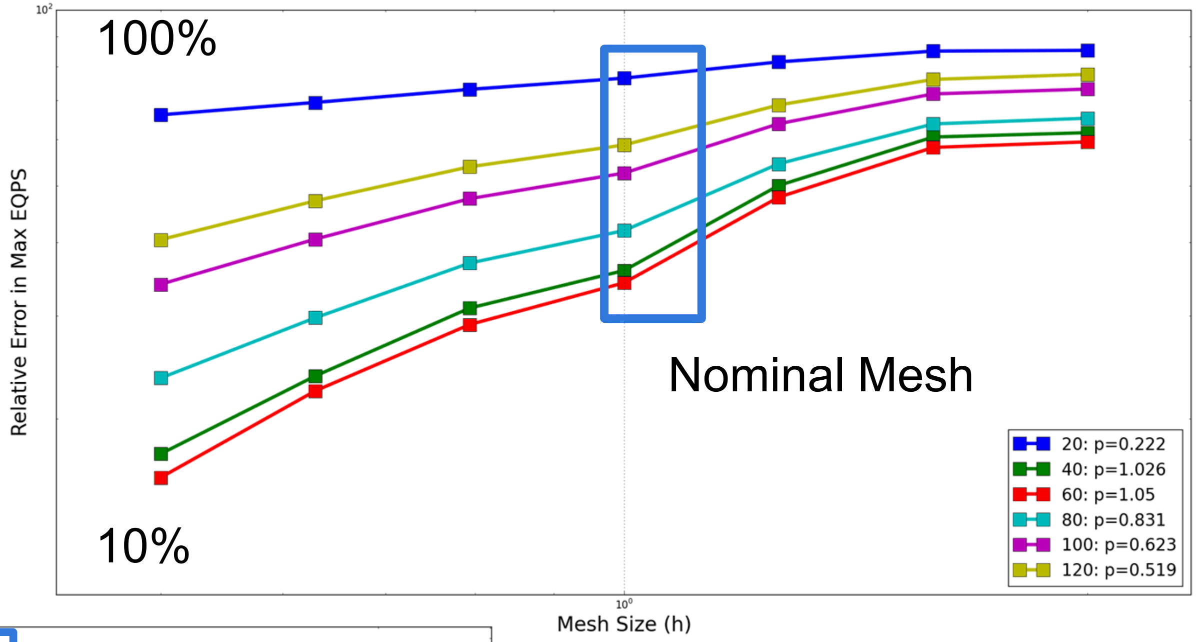
1/4 Scale

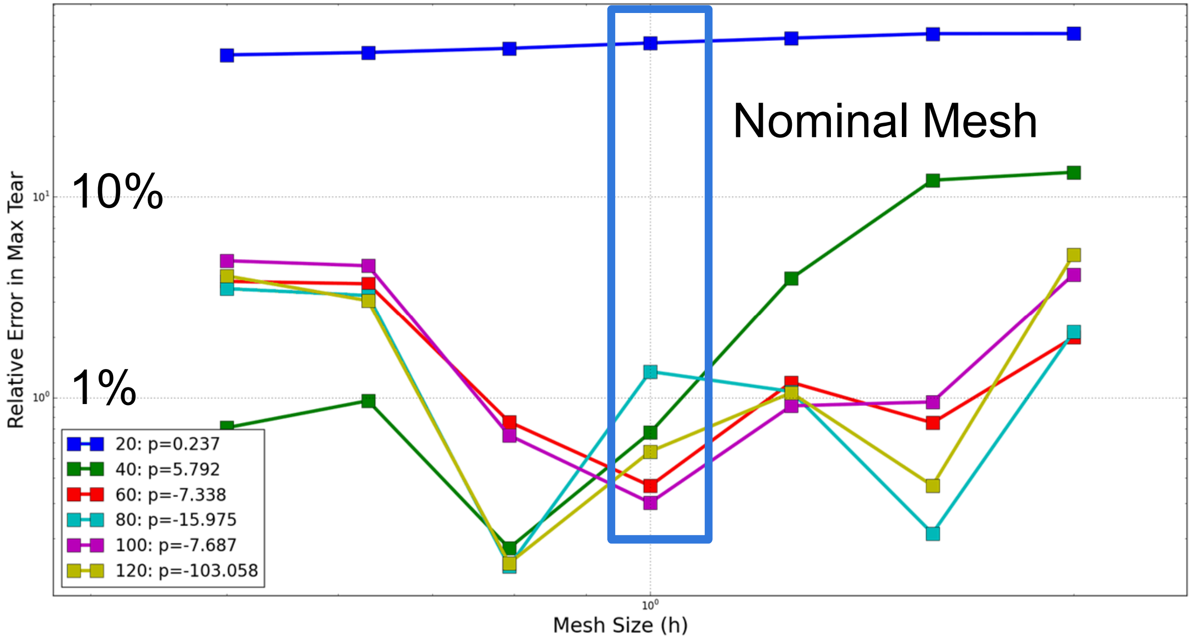


4x Scale

PCAP: Errors Using Multi-Mesh Extrapolation

- Extrapolation using the nonlinear least squares approach (using 1/2x scaled mesh plus 4 finer meshes)
- Estimated rates for max EQPS approach 0.5 at end
- Extrapolation provides useful error estimates
- Nominal mesh error: 30-80%





- Estimated rates for max tearing parameter are **non-convergent**
- Extrapolation does not provide useful error estimates
- Nominal mesh error: unknown

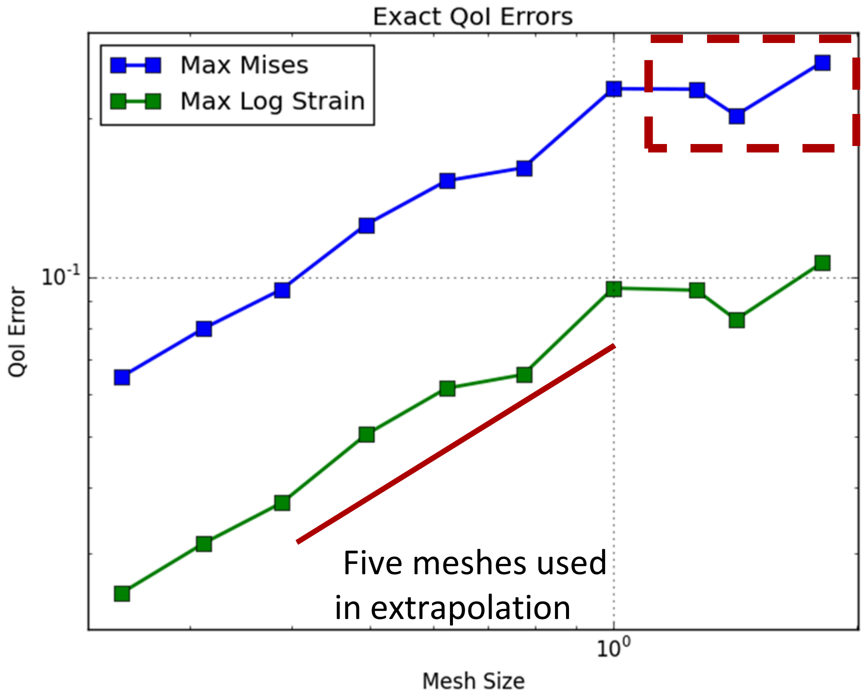
Solution Verification
Conclusions and Recommendations

- Mesh scaling can produce meshes that are competitive w/remeshing
- Further evidence will be needed to determine the limits of mesh scaling on complex geometries
- Cost savings can be realized (Notch: 64% savings in CPU hours)
- Extrapolation still can fail, but this cannot be fixed w/just meshing

Recommendations

- Use a minimum scale factor: 2.0
- Verify mesh quality before running the analysis
- No more than 1-2 coarsenings; avoid if possible
- Use 3-5 meshes for extrapolation
- Avoid coarser meshes that are far off trend
- Extrapolation can work for many QoIs but should be used with caution (make sure your QoI has been verified!)

Off trend date from coarsening



Five meshes used in extrapolation

MMS Cylinder QoI Errors

Future Work: Weakest Link in Mesh Scaling

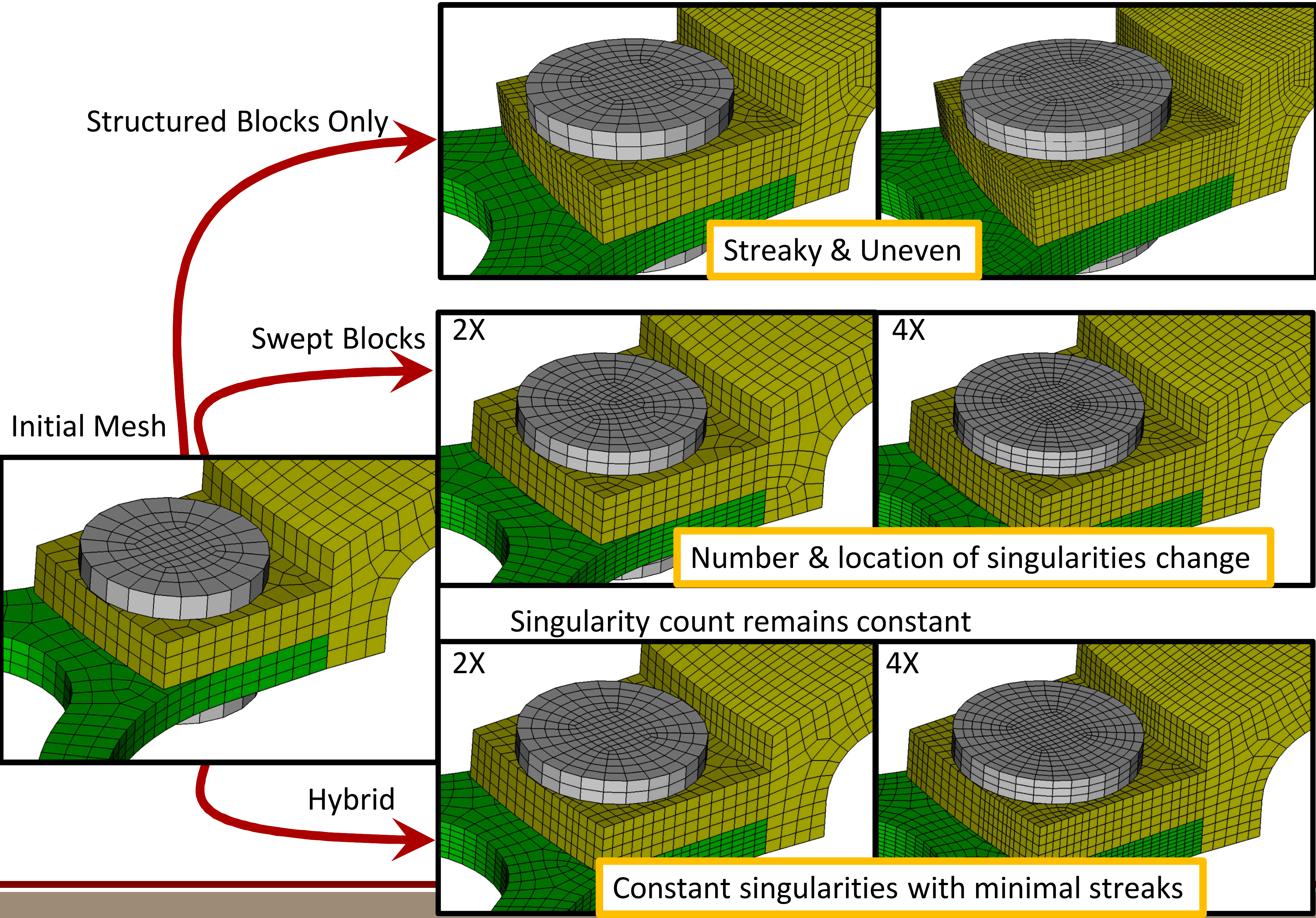
The Paver is the cause of most problems with mesh scaling:

- Changes number and locations of singularities, destroys structure.
- Non-deterministic, overshooting/undershooting target ***Multiplier***.
- Non-deterministic, different results on different platforms.
- Inherently serial algorithm, can't scale to HPC.
- Heuristic, unhandled cases appear sometimes.

We are experimenting with a new "Hybrid" approach to mesh scaling:

- Constructs 2 block decompositions, one purely structured, one with swept blocks.
- Mappings between block decompositions.
- Allows for modification of original mesh with strategic dicing and smoothing (eliminating Paver).
- Modifies the mesh, rather than throw away and rebuild mesh.

Structured vs Swept Blocks vs Hybrid Approach



Initial Mesh

Structured Blocks Only

Streaky & Uneven

Swept Blocks

Number & location of singularities change

Singularity count remains constant

Hybrid

Constant singularities with minimal streaks