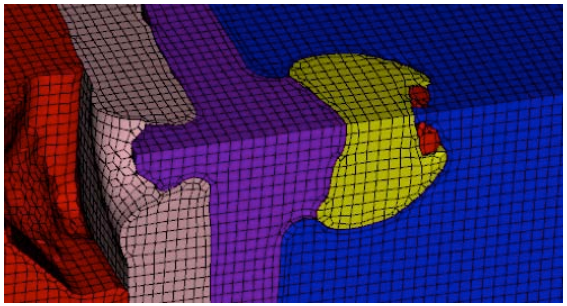
Parallel Hex Meshing from Volume Fractions

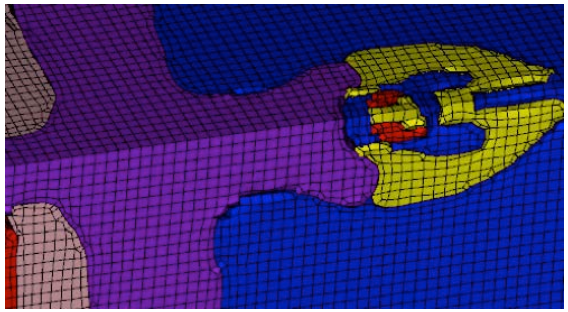
Steven J. Owen, Matthew L. Staten

November 7, 2012

NDIA

Denver, CO





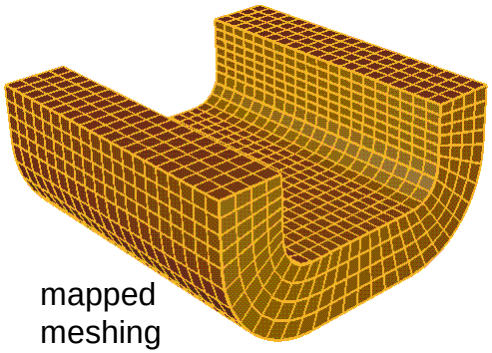
LOCKHEED MARTIN

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

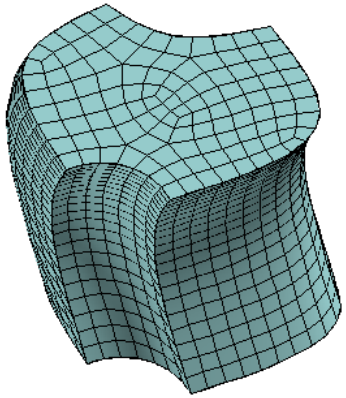
Sandia National Laboratories

Hexahedral Meshing

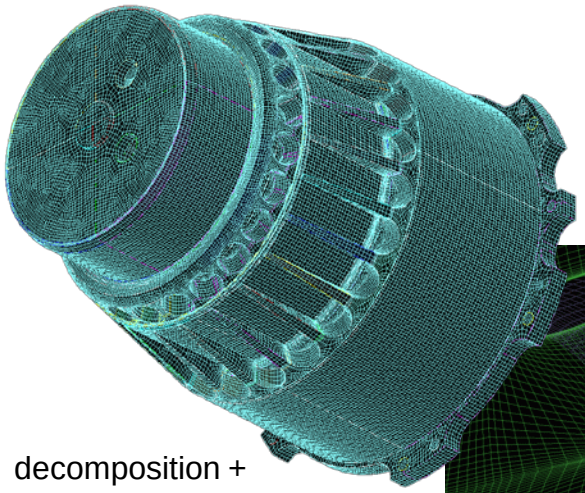
Structured



mapped meshing

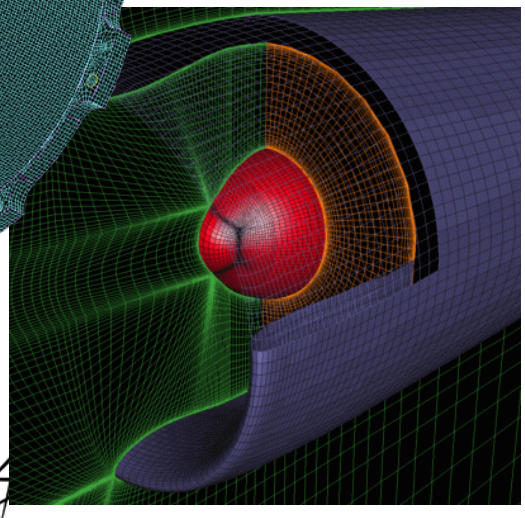


sweeping

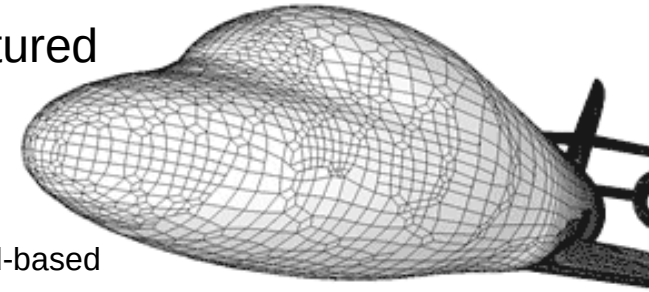


decomposition + sweeping

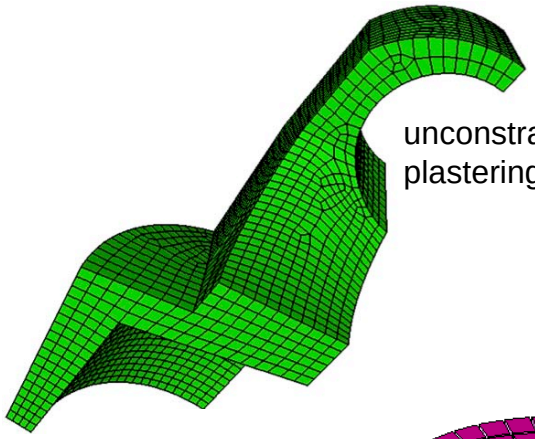
block structured



Unstructured

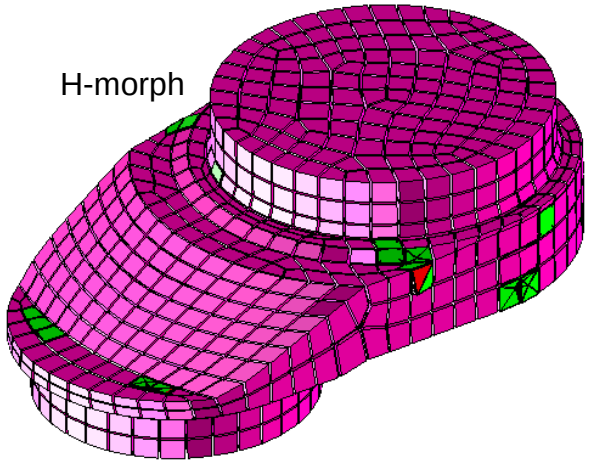


grid-based

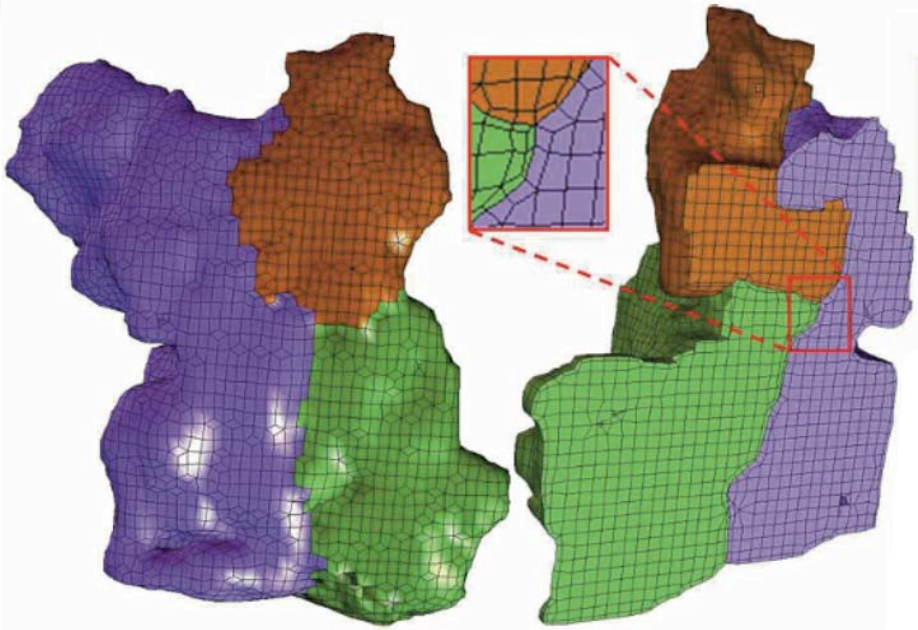


unconstrained plastering

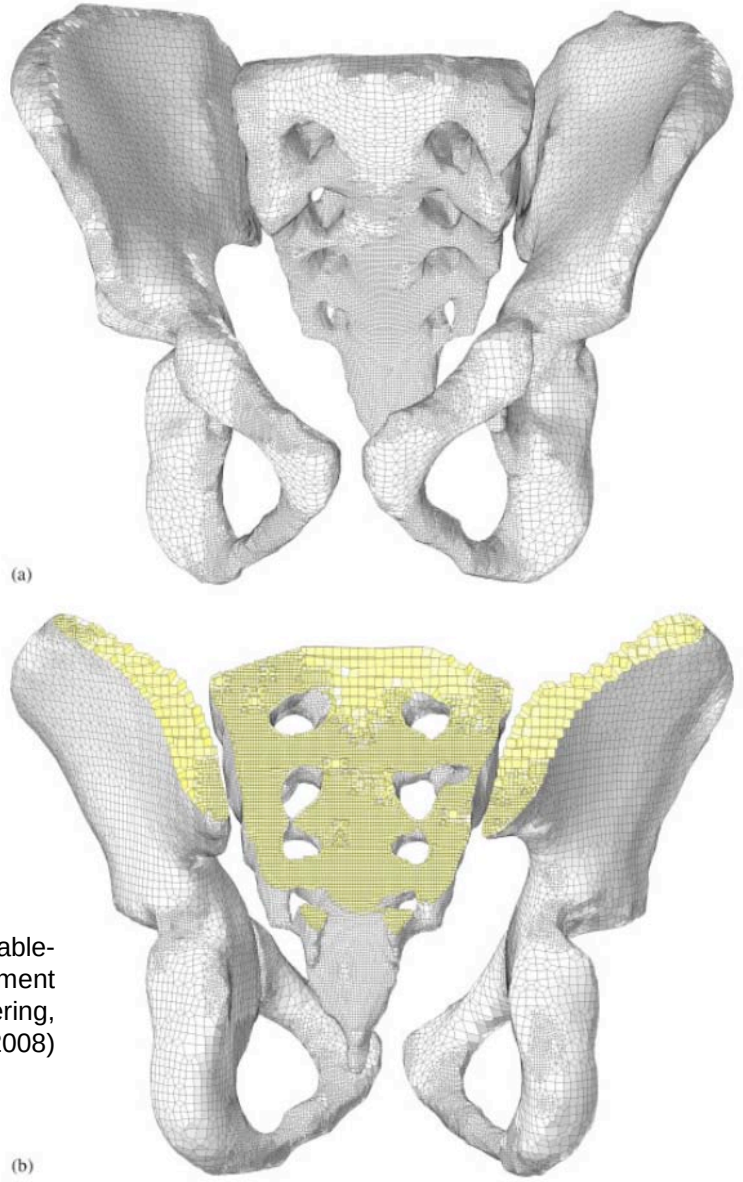
H-morph



Grid-Based Methods



Yongjie Zhang, Thomas Hughes, and Chandrijit Bajaj, “Automatic 3D Mesh Generation for a Domain with Multiple Materials”, Proceedings, 16th International Meshing Roundtable, pp. 367-386, (2007)

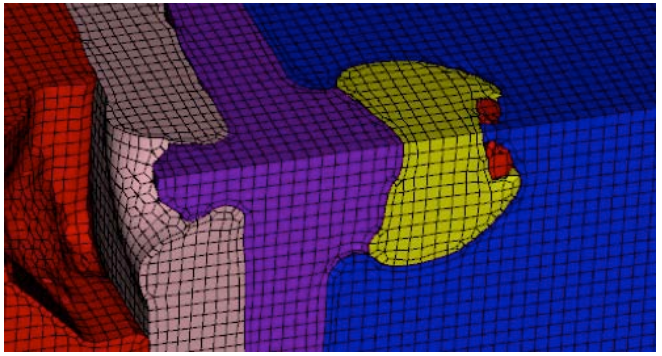


Yasushi Ito, Alan M. Shih, and Bharat K. Soni, “Octree-based reasonable-quality hexahedral mesh generation using a new set of refinement templates,” International Journal for Numerical Methods in Engineering, published online DOI: 10.1002/nme.2470 (2008)

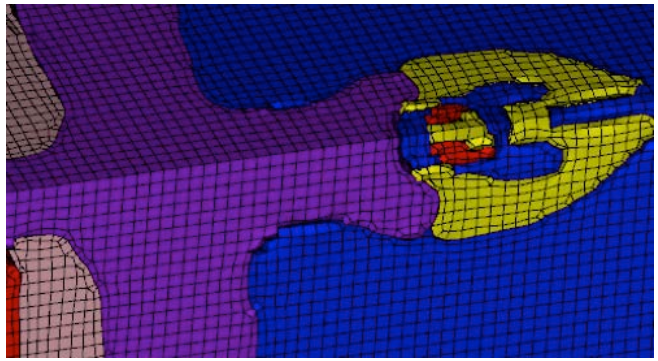
Sculptor

$v_A = 0.73$ $v_B = 0.27$	$v_A = 0.41$ $v_B = 0.59$	$v_A = 0.43$ $v_B = 0.57$
$v_A = 0.00$ $v_B = 1.00$	$v_A = 0.55$ $v_B = 0.45$	$v_A = 0.38$ $v_B = 0.62$
$v_A = 0.00$ $v_B = 1.00$	$v_A = 0.79$ $v_B = 0.21$	$v_A = 1.00$ $v_B = 0.00$

Volume Fraction Data



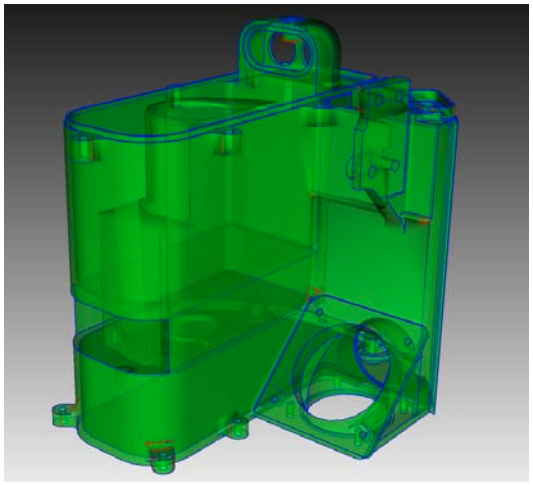
All-hex mesh



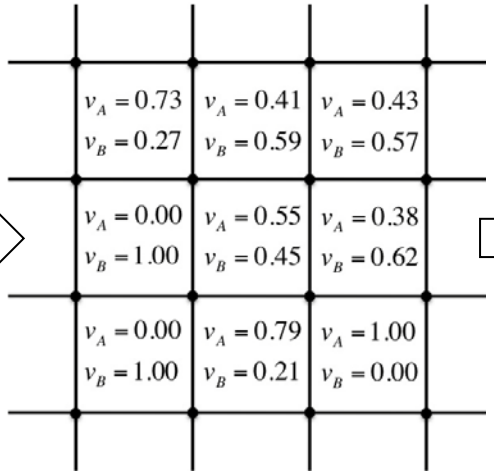
LOCKHEED MARTIN

Sandia National Laboratories

Sculptor



STL File

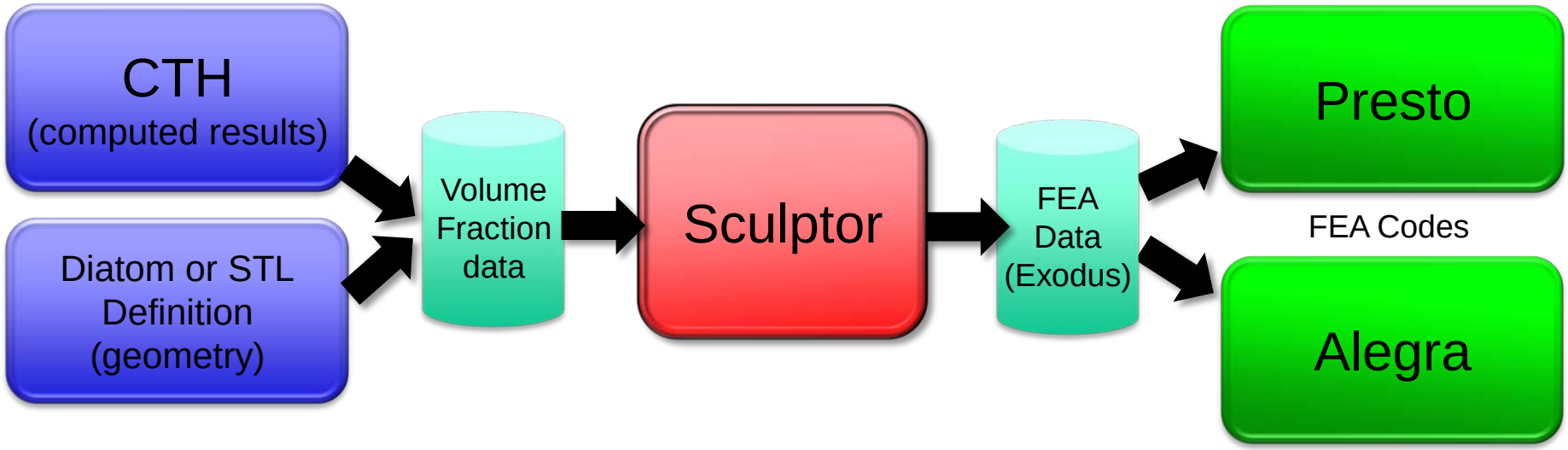


Volume Fraction Data

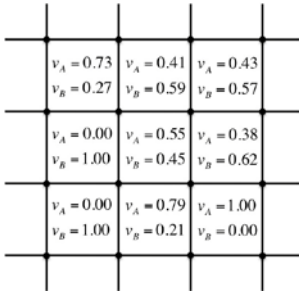
All-hex mesh

Implementation

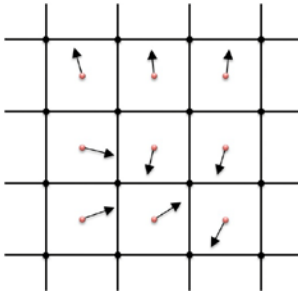
Sculptor is a new tool for generating all-hex meshes from volume fraction data generated for use in Sandia's FEA codes



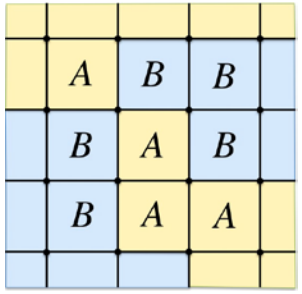
The Procedure



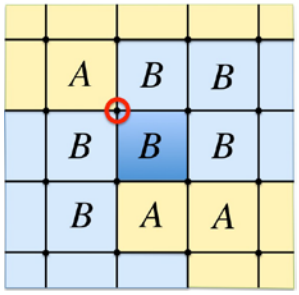
1 Establish Parallel Cartesian Grid



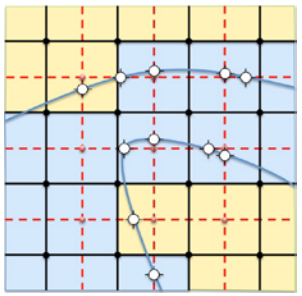
2 Estimate Gradients at Cell Centers



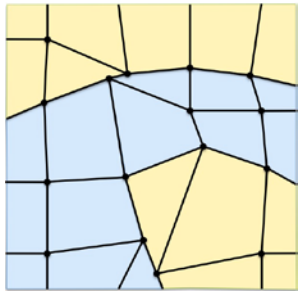
3 Assign Materials to Cells



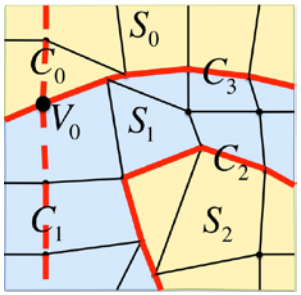
4 Resolve Non-Manifold Cases



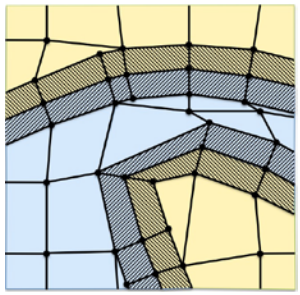
5 Compute Virtual Edge Crossings



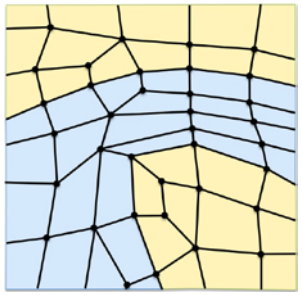
6 Move Grid Points to Iso-surface



7 Create Geometry Definition



8 Insert Hex Buffer Layer

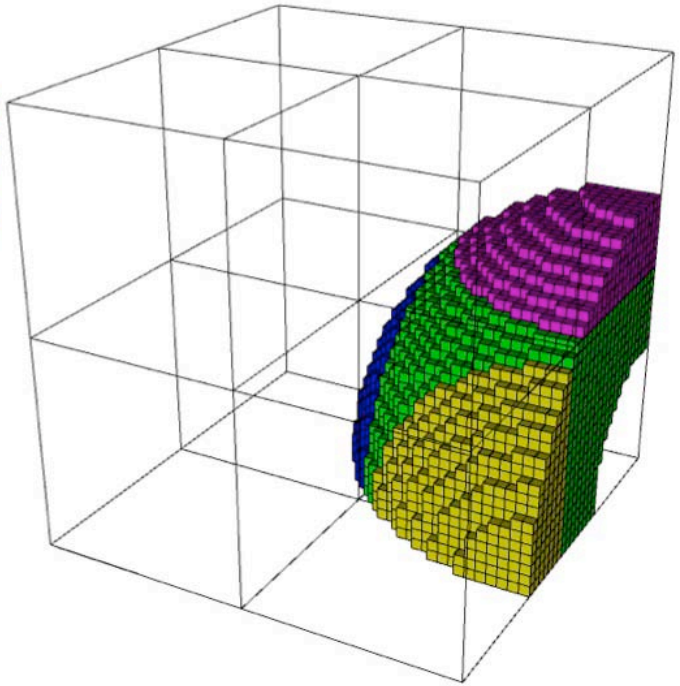


9 Smooth

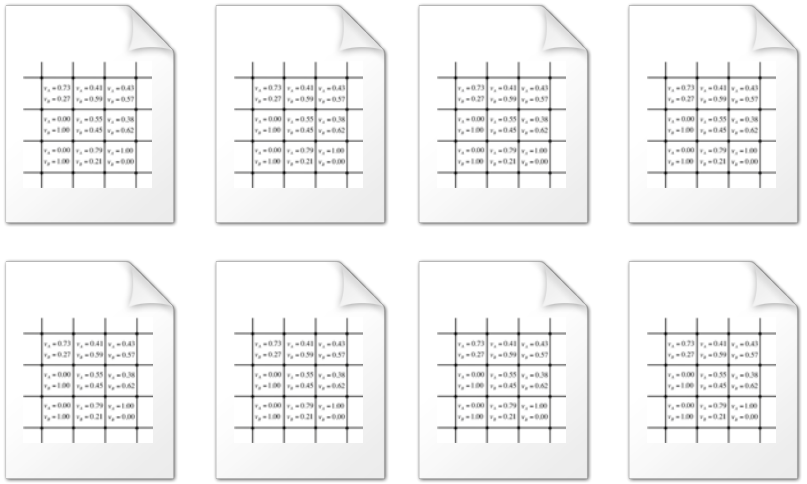
LOCKHEED MARTIN

Sandia National Laboratories

Establish Parallel Cartesian Grid



Eulerian Code Result
on N Processors



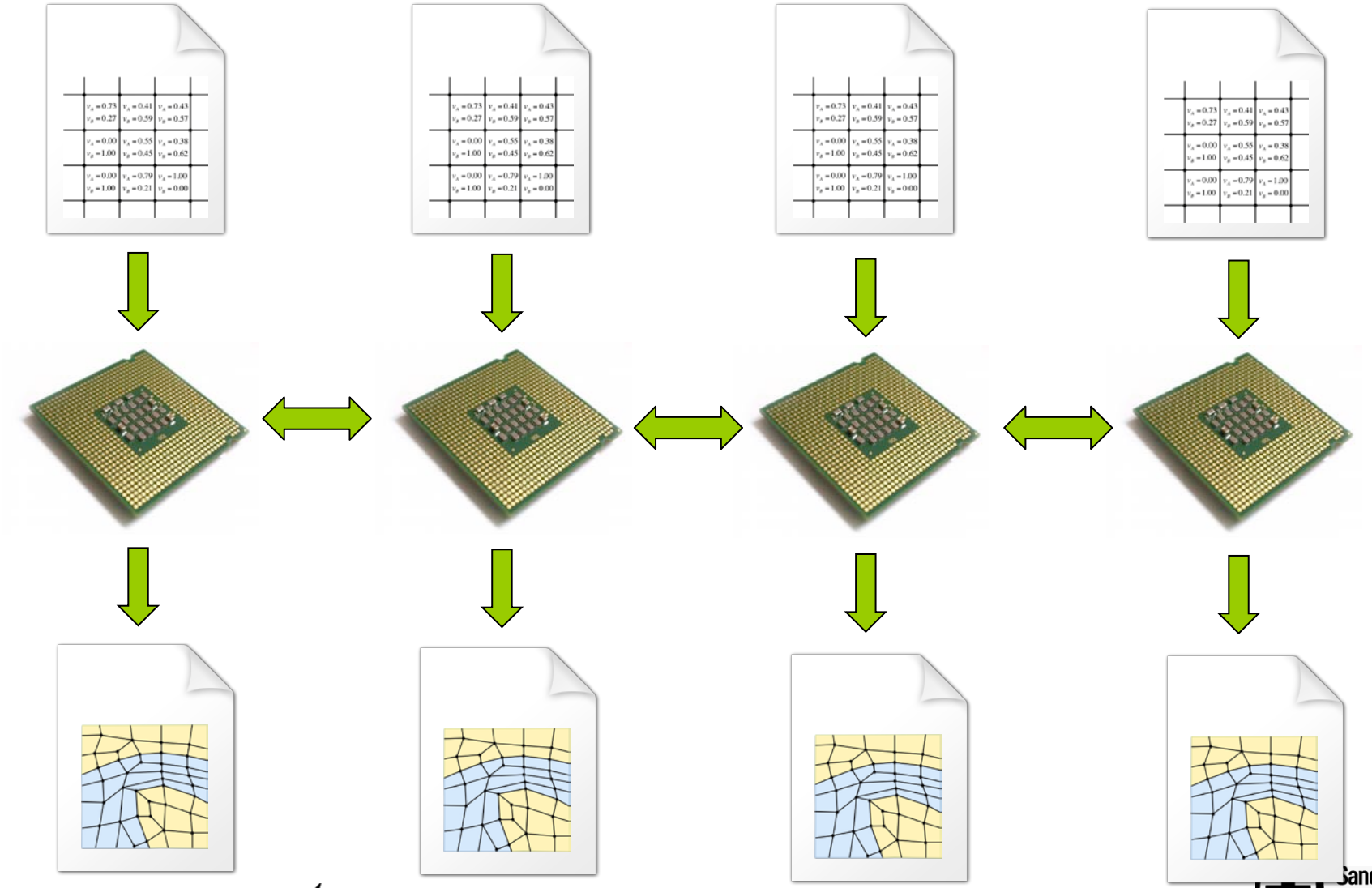
Result written to N files
Or through API

- Cartesian Grid Dimensions
- Neighbor Processor IDs
- 1 volume fraction per material per cell

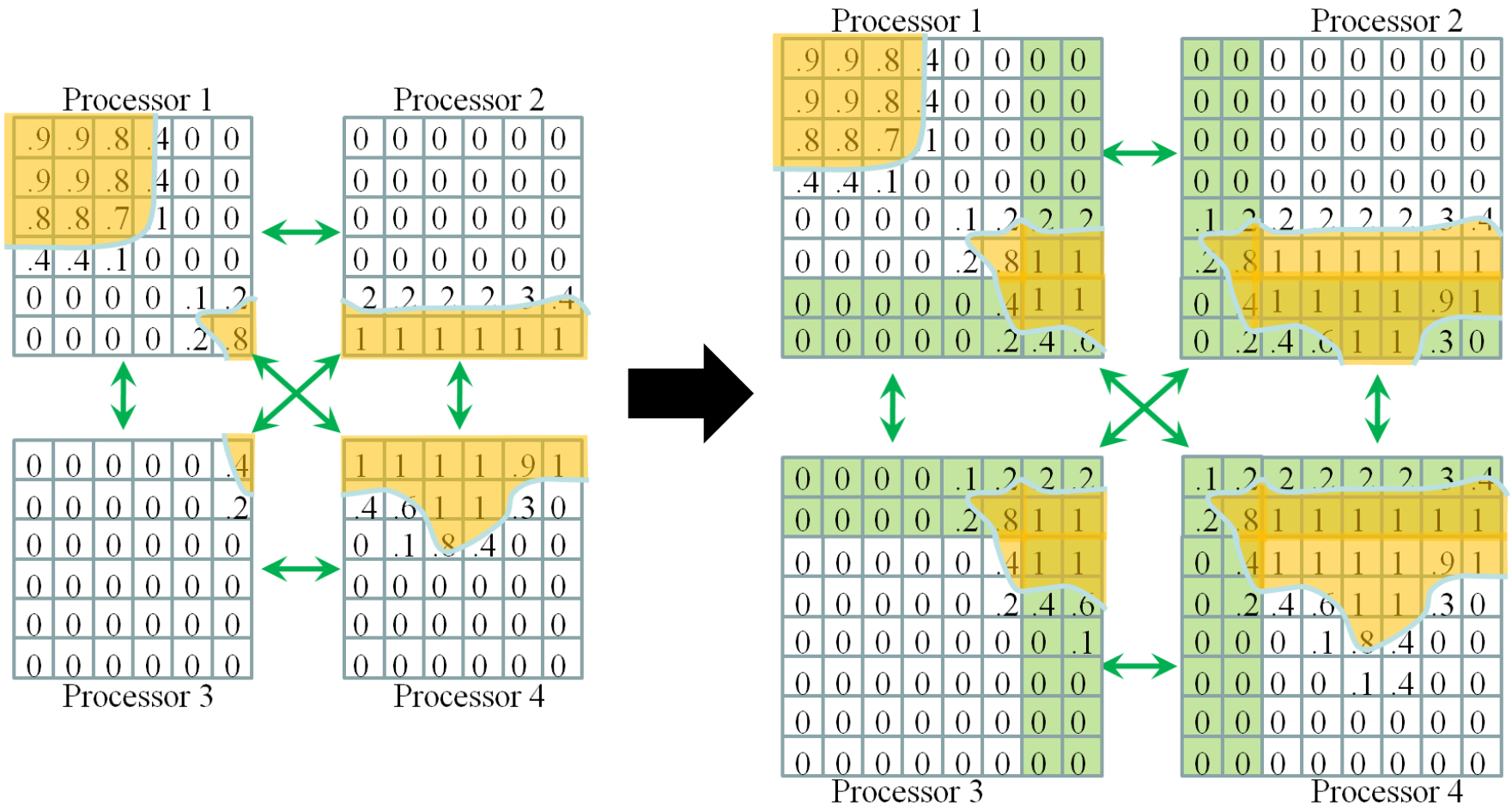
LOCKHEED MARTIN

Sandia National Laboratories

Establish Parallel Cartesian Grid

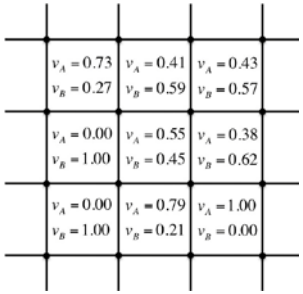


Establish Parallel Cartesian Grid

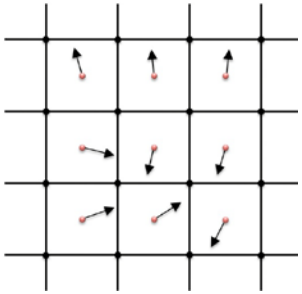


Two layer ghosting ensures smooth level sets

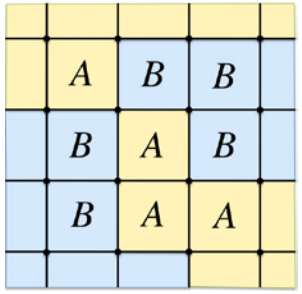
The Procedure



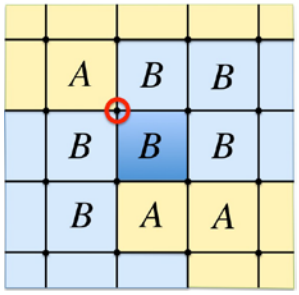
1 Establish Parallel Cartesian Grid



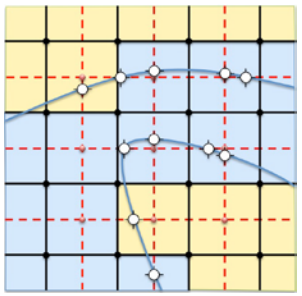
2 Estimate Gradients at Cell Centers



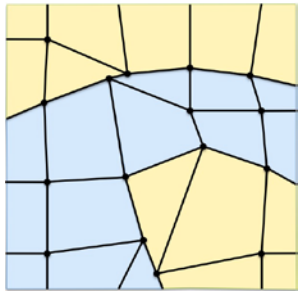
3 Assign Materials to Cells



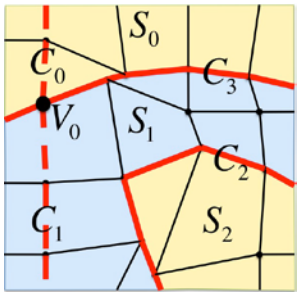
4 Resolve Non-Manifold Cases



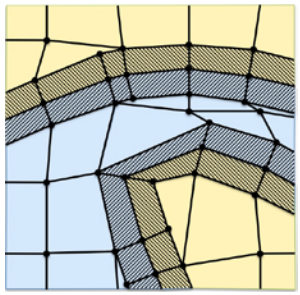
5 Compute Virtual Edge Crossings



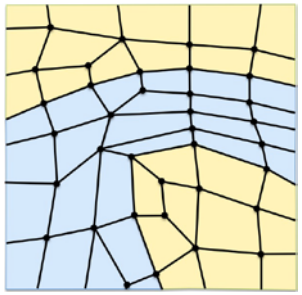
6 Move Grid Points to Iso-surface



7 Create Geometry Definition



8 Insert Hex Buffer Layer

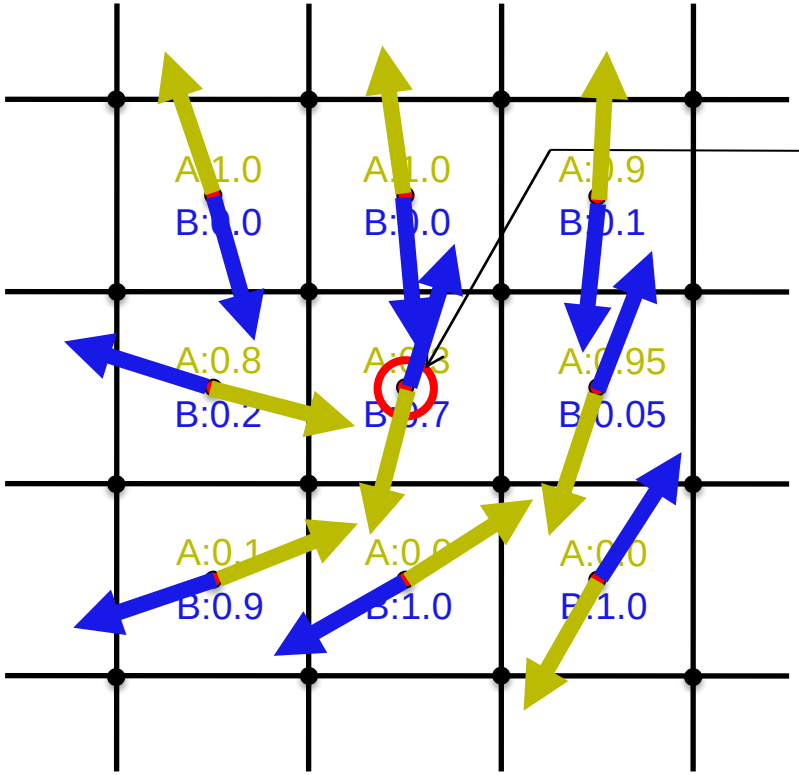


9 Smooth

LOCKHEED MARTIN

Sandia National Laboratories

Estimate Gradients at Cell Centers

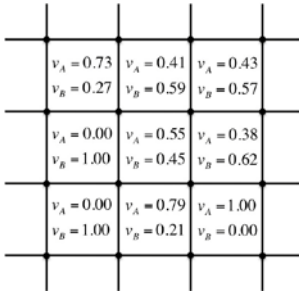


Compute Gradient at Cell Center

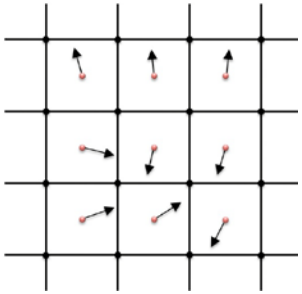
$$\begin{bmatrix} \sum \Delta x_i^2 & \sum \Delta x_i \Delta y_i & \sum \Delta x_i \Delta z_i \\ \sum \Delta x_i \Delta y_i & \sum \Delta y_i^2 & \sum \Delta y_i \Delta z_i \\ \sum \Delta x_i \Delta z_i & \sum \Delta y_i \Delta z_i & \sum \Delta z_i^2 \end{bmatrix} \begin{Bmatrix} \frac{\partial v}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial v}{\partial z} \end{Bmatrix} = \begin{bmatrix} \sum \Delta x_i \Delta (v_n)_i \\ \sum \Delta y_i \Delta (v_n)_i \\ \sum \Delta z_i \Delta (v_n)_i \end{bmatrix}$$

Least Squares Approximation

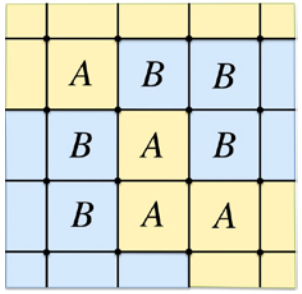
The Procedure



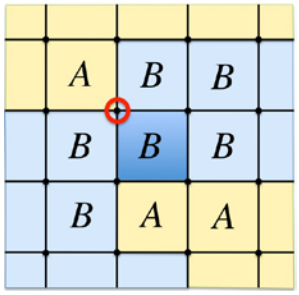
1 Establish Parallel Cartesian Grid



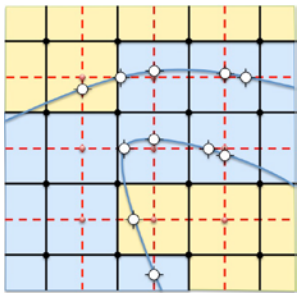
2 Estimate Gradients at Cell Centers



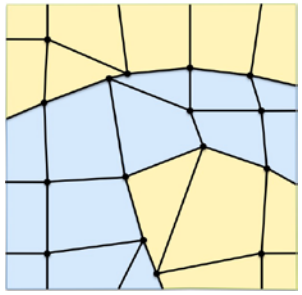
3 Assign Materials to Cells



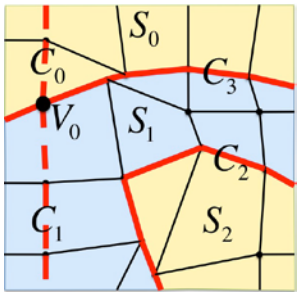
4 Resolve Non-Manifold Cases



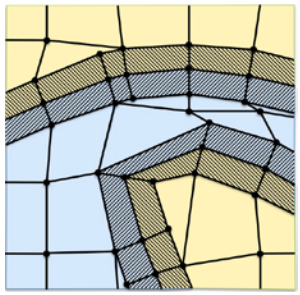
5 Compute Virtual Edge Crossings



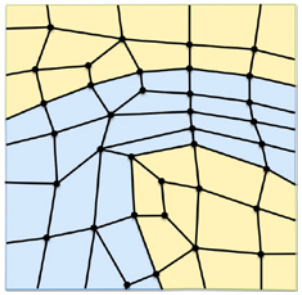
6 Move Grid Points to Iso-surface



7 Create Geometry Definition



8 Insert Hex Buffer Layer

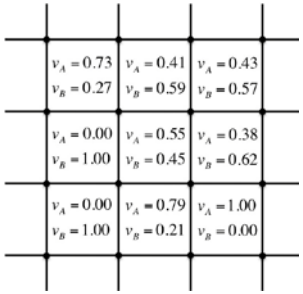


9 Smooth

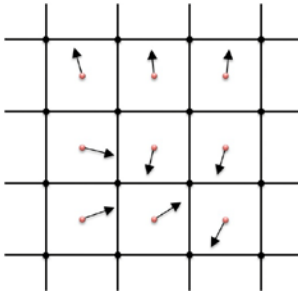
LOCKHEED MARTIN

Sandia National Laboratories

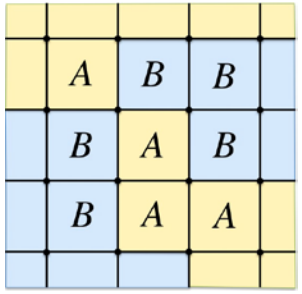
The Procedure



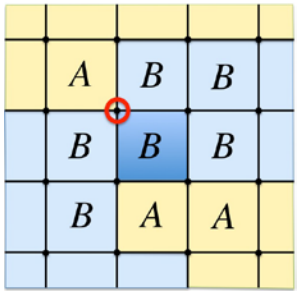
1 Establish Parallel Cartesian Grid



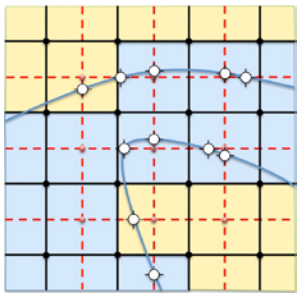
2 Estimate Gradients at Cell Centers



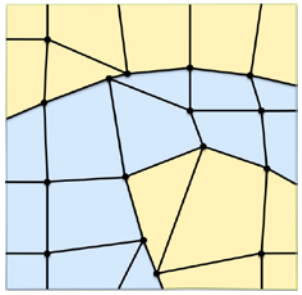
3 Assign Materials to Cells



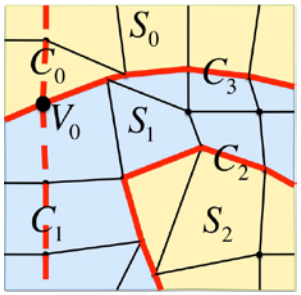
4 Resolve Non-Manifold Cases



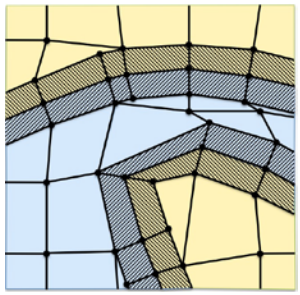
5 Compute Virtual Edge Crossings



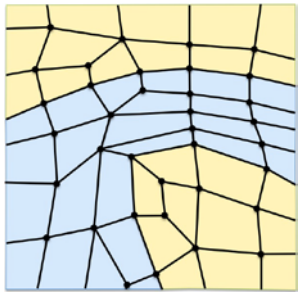
6 Move Grid Points to Iso-surface



7 Create Geometry Definition



8 Insert Hex Buffer Layer

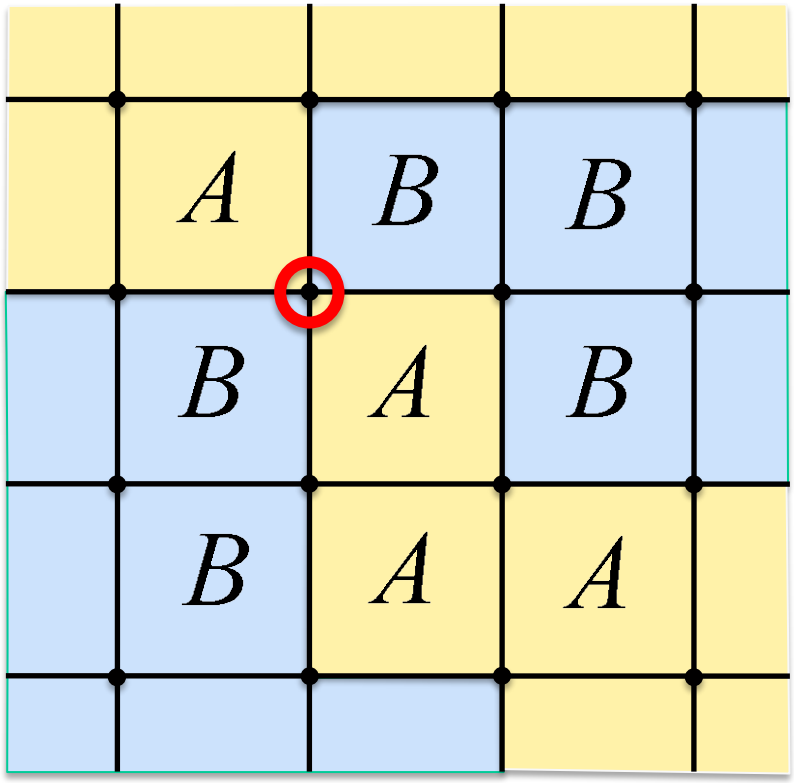


9 Smooth

LOCKHEED MARTIN

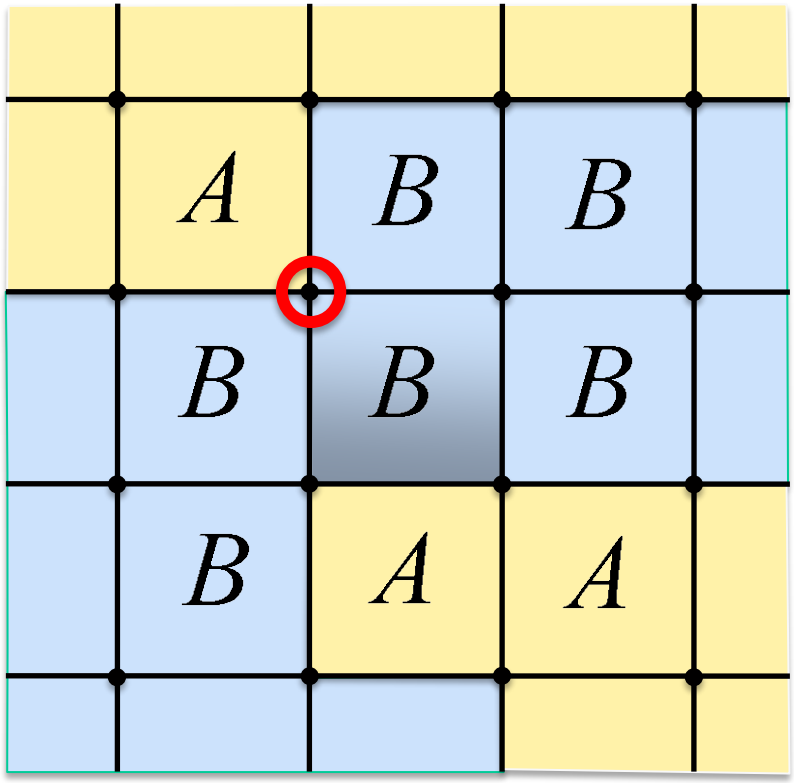
Sandia National Laboratories

Resolve Non-Manifold Cases



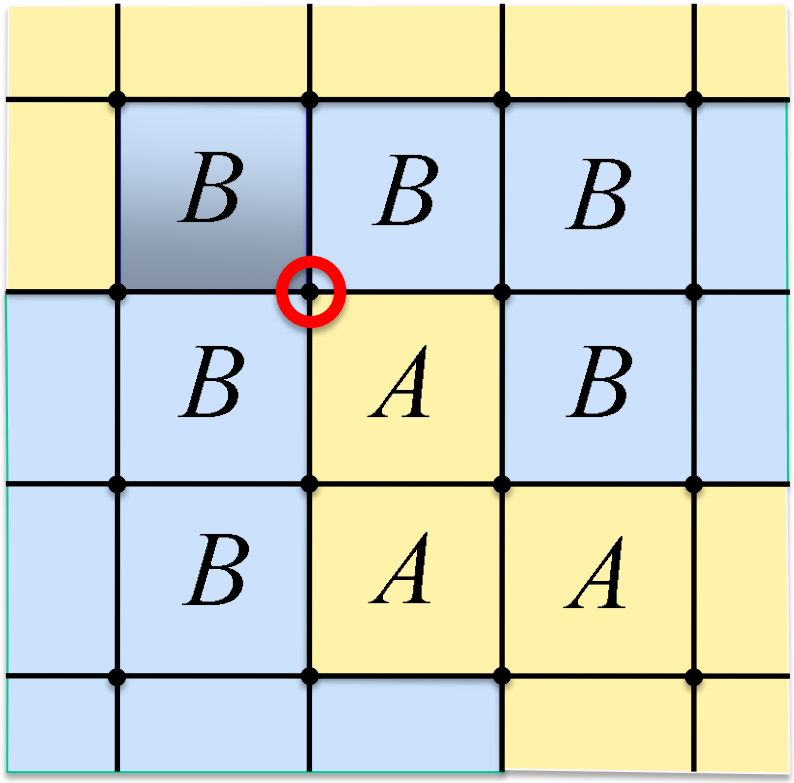
Non-manifold: Cells of the same material meet at a node, but no faces

Resolve Non-Manifold Cases



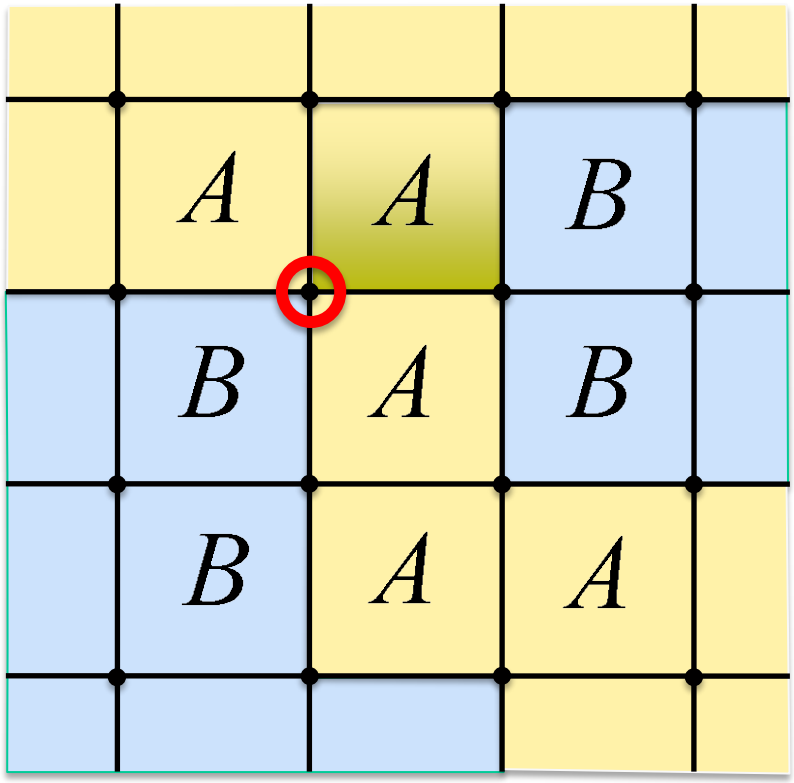
Non-manifold: Cells of the same material meet at a node, but no faces

Resolve Non-Manifold Cases



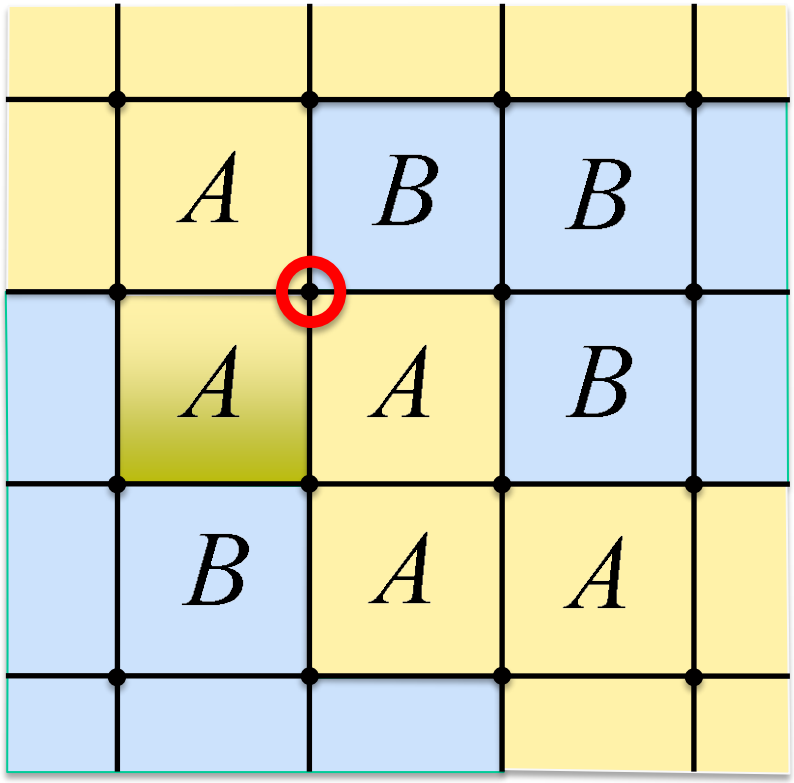
Non-manifold: Cells of the same material meet at a node, but no faces

Resolve Non-Manifold Cases



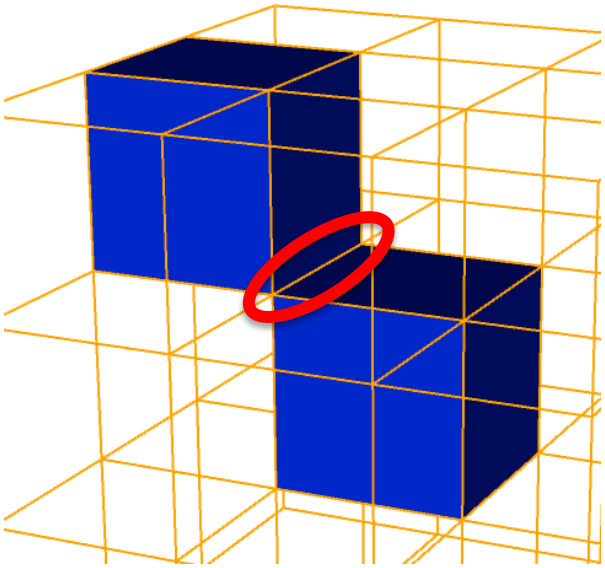
Non-manifold: Cells of the same material meet at a node, but no faces

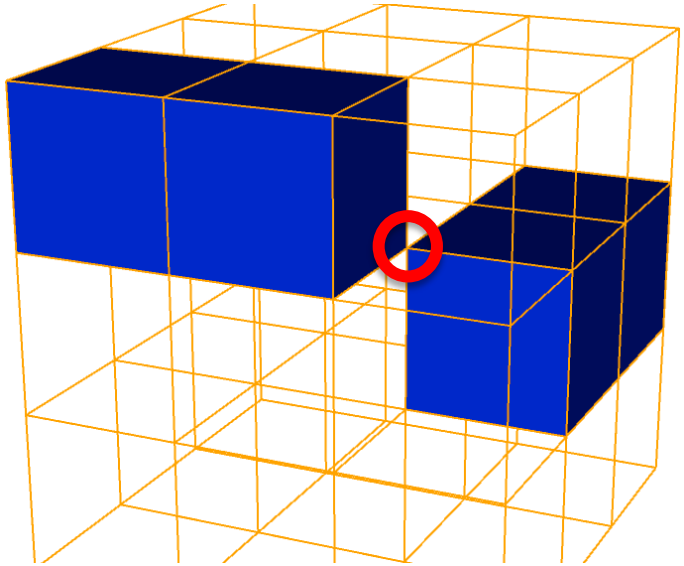
Resolve Non-Manifold Cases



Non-manifold: Cells of the same material meet at a node, but no faces

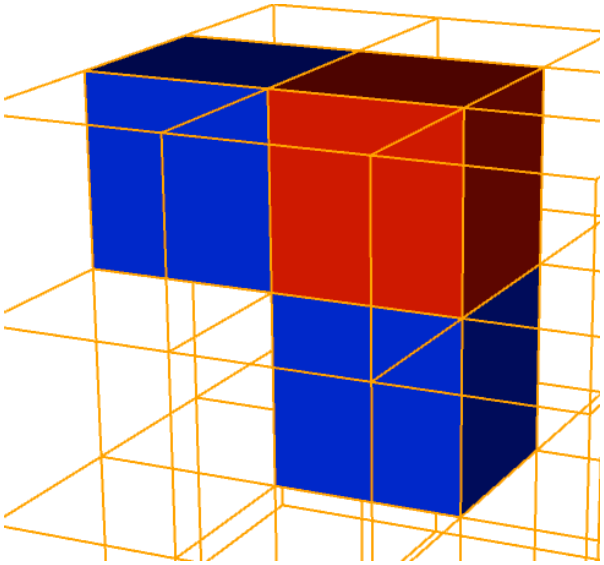
Resolve Non-Manifold Cases

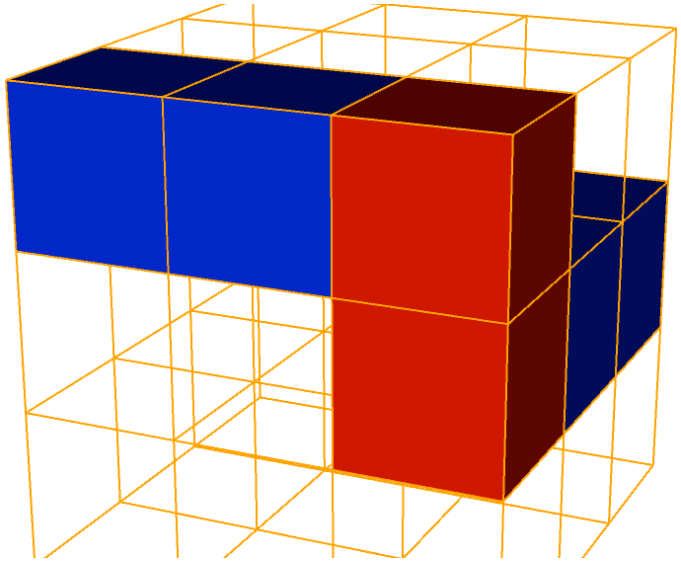




Non-manifold: Cells of the same material meet at a node or edge, but no faces

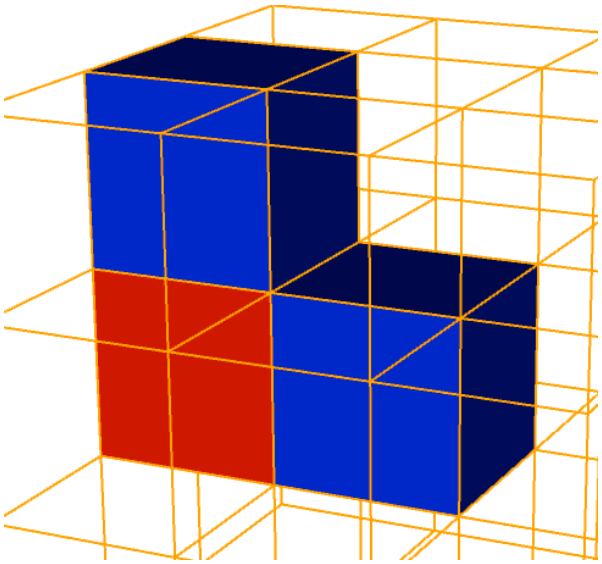
Resolve Non-Manifold Cases

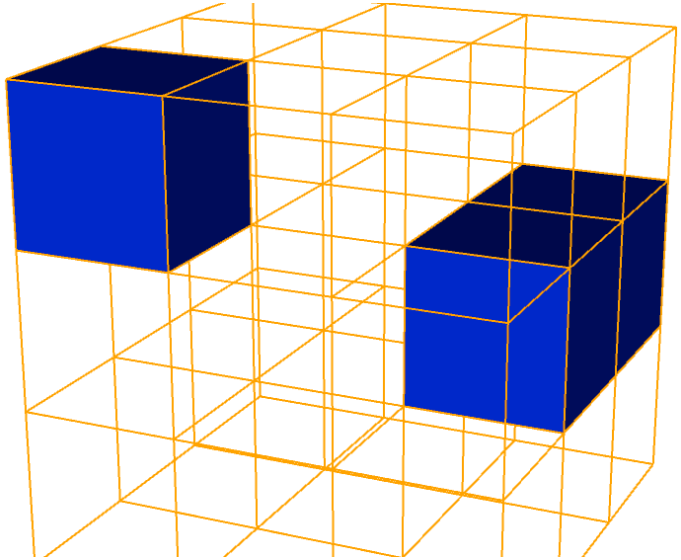




Non-manifold: Cells of the same material meet at a node or edge, but no faces

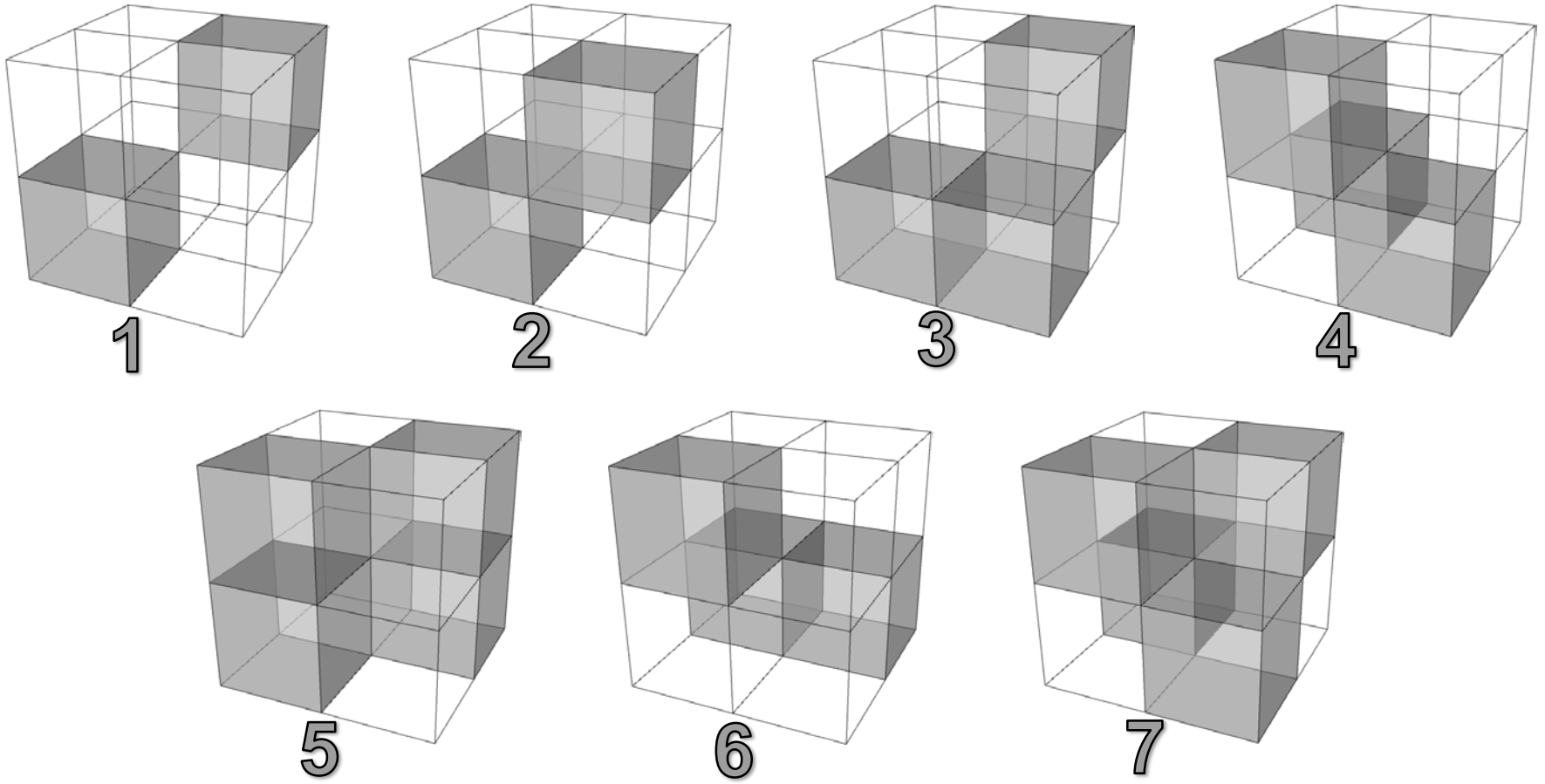
Resolve Non-Manifold Cases





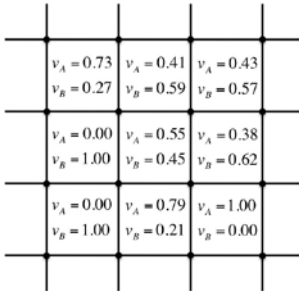
Non-manifold: Cells of the same material meet at a node or edge, but no faces

Resolve Non-Manifold Cases

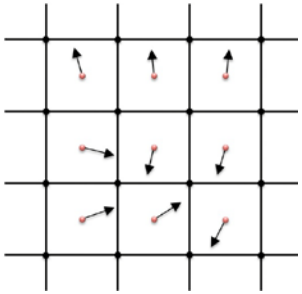


7 unique cases of *non-manifoldness* at a node in 3D

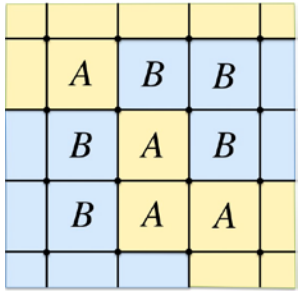
The Procedure



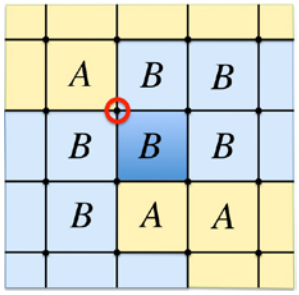
1 Establish Parallel Cartesian Grid



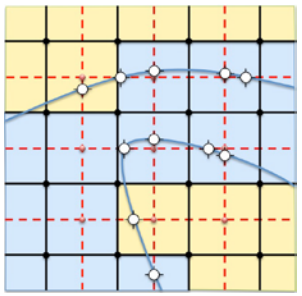
2 Estimate Gradients at Cell Centers



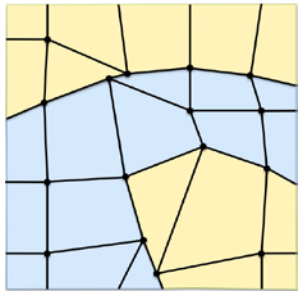
3 Assign Materials to Cells



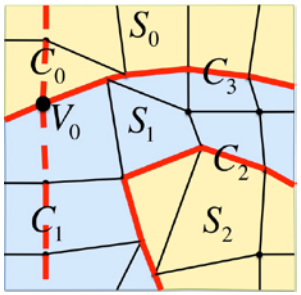
4 Resolve Non-Manifold Cases



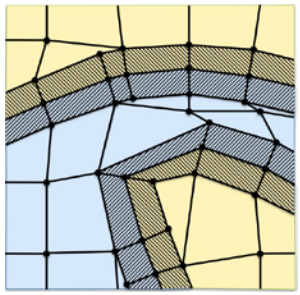
5 Compute Virtual Edge Crossings



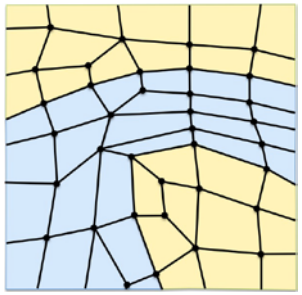
6 Move Grid Points to Iso-surface



7 Create Geometry Definition



8 Insert Hex Buffer Layer

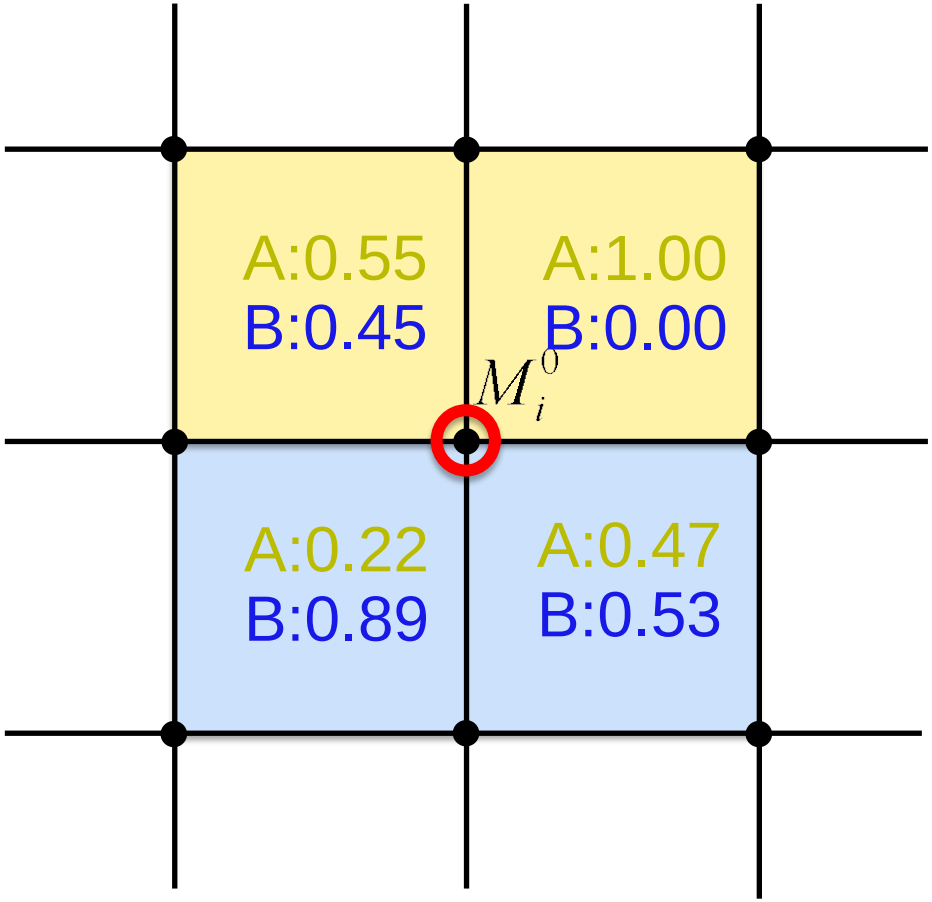


9 Smooth

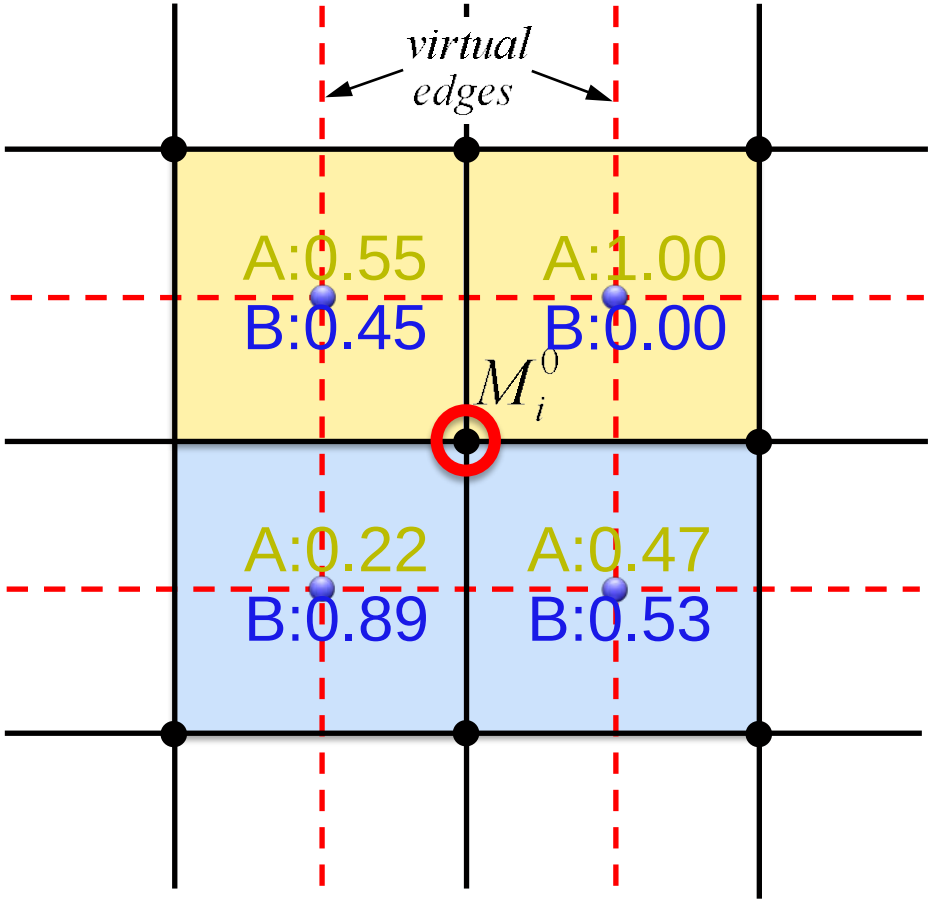
LOCKHEED MARTIN

Sandia National Laboratories

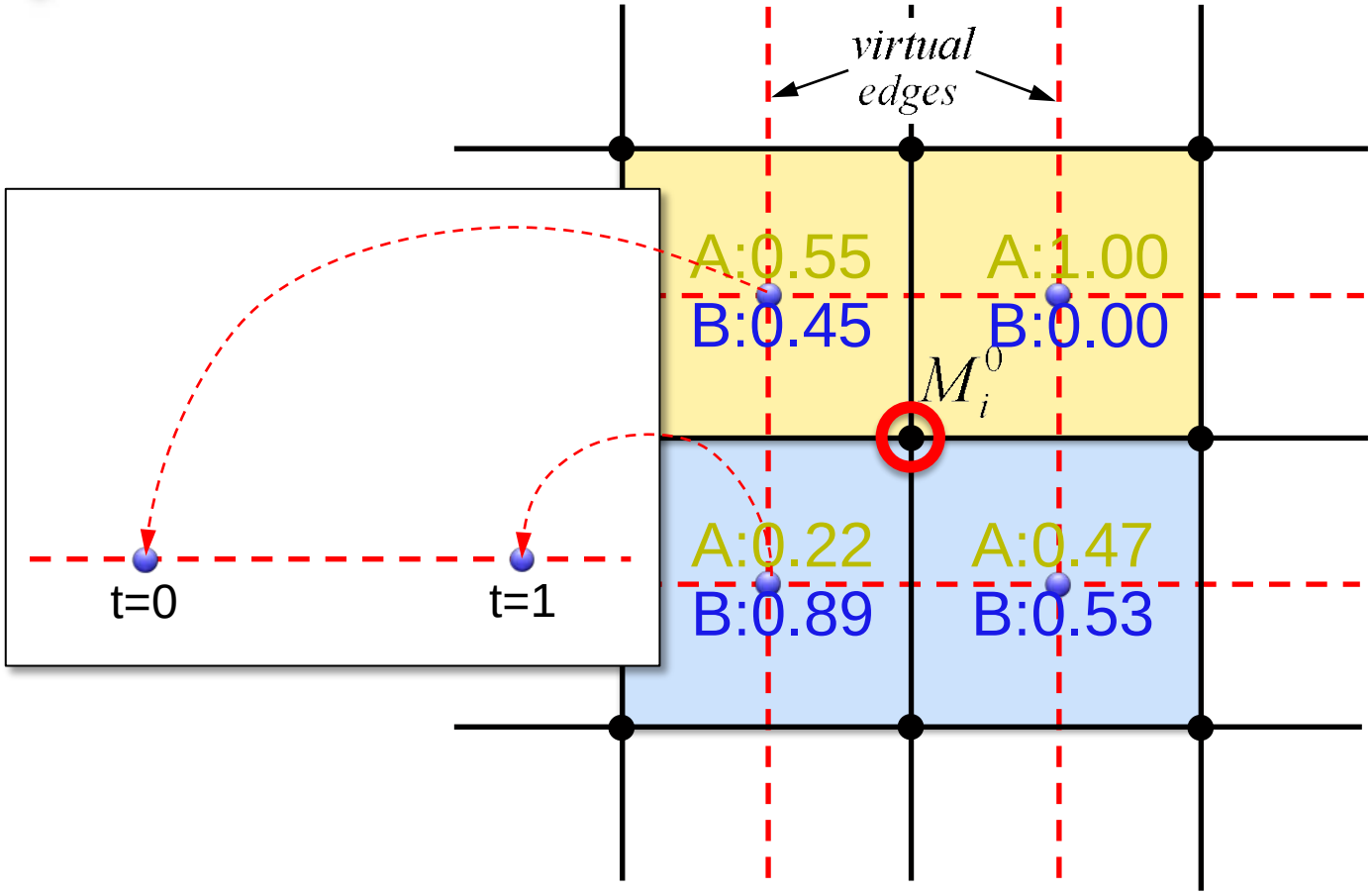
Compute Virtual Edge Crossings



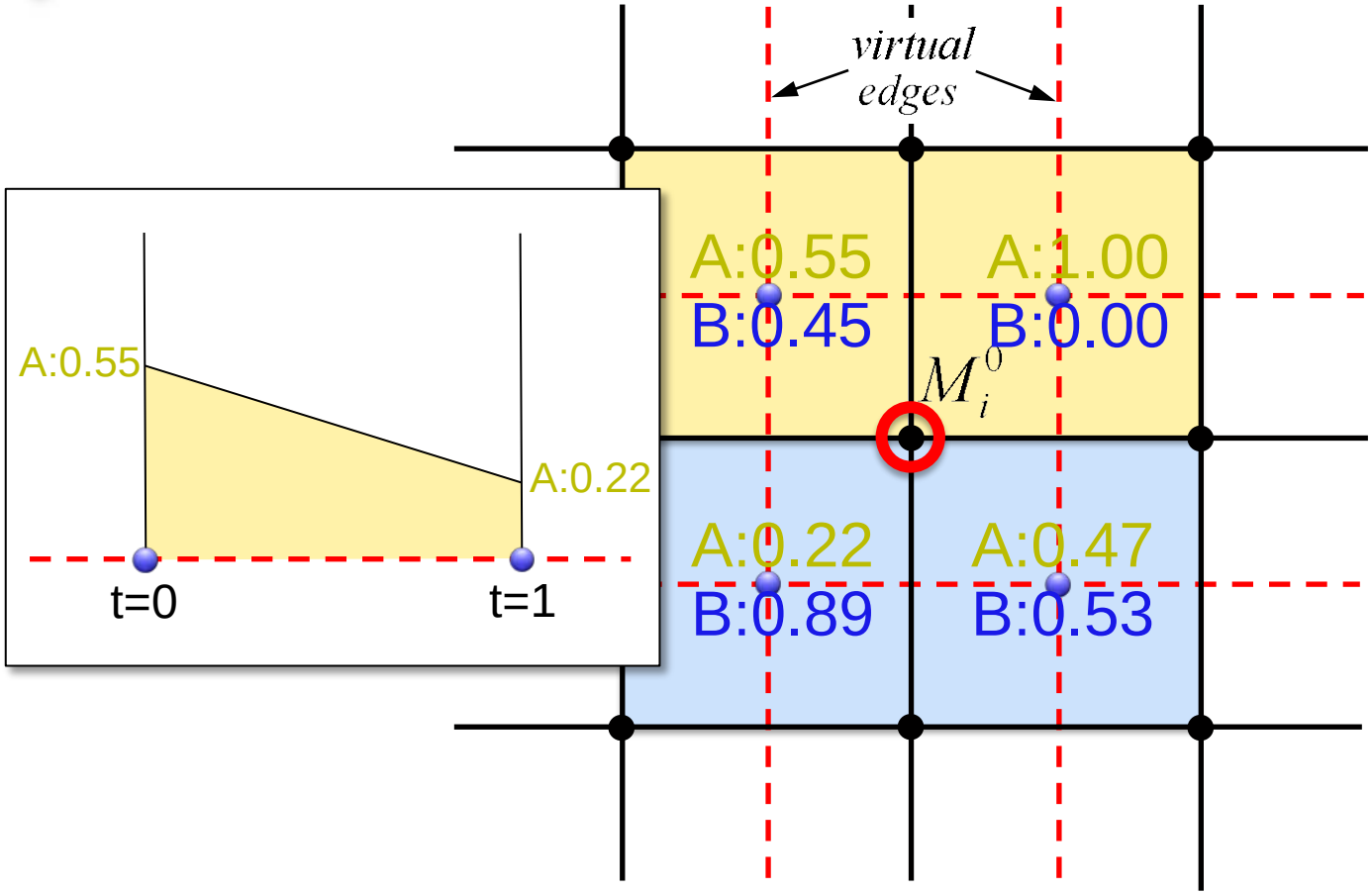
Compute Virtual Edge Crossings



Compute Virtual Edge Crossings

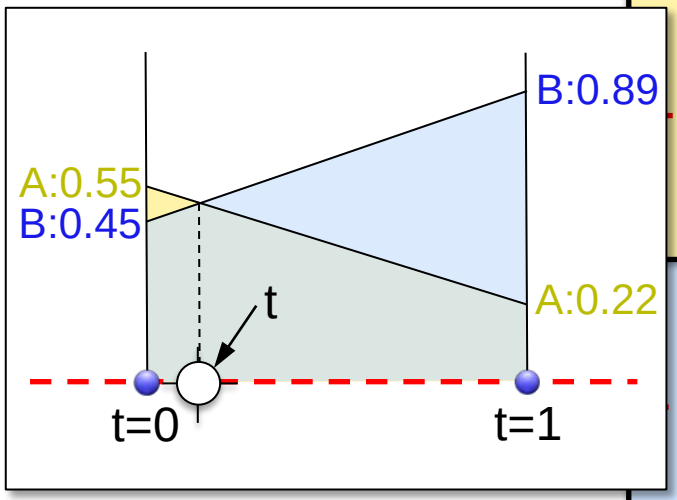


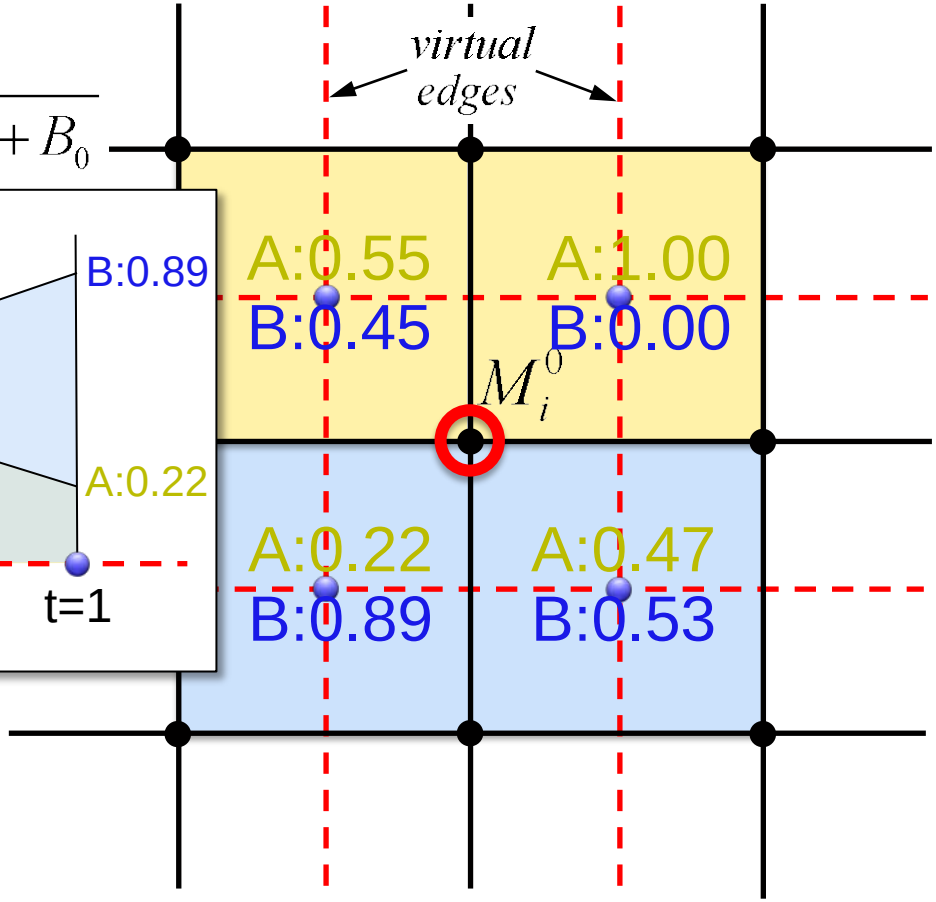
Compute Virtual Edge Crossings



Compute Virtual Edge Crossings

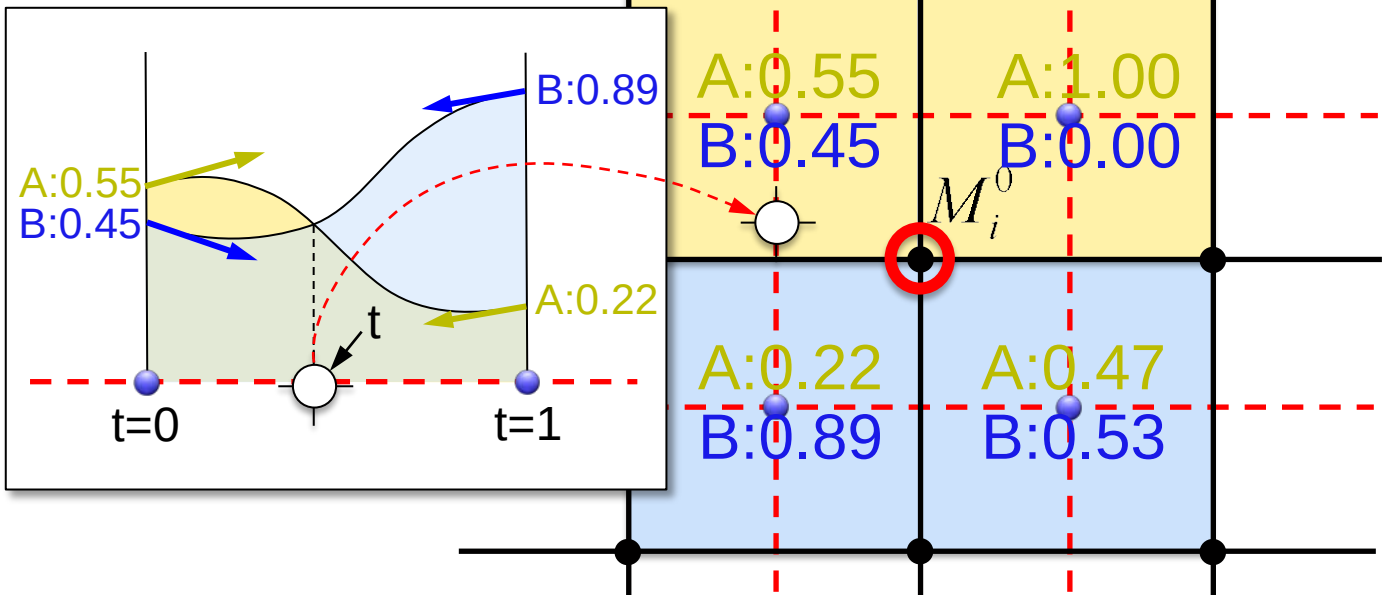
$$t = \frac{B_0 - A_0}{A_1 - A_0 - B_1 + B_0}$$





Compute Virtual Edge Crossings

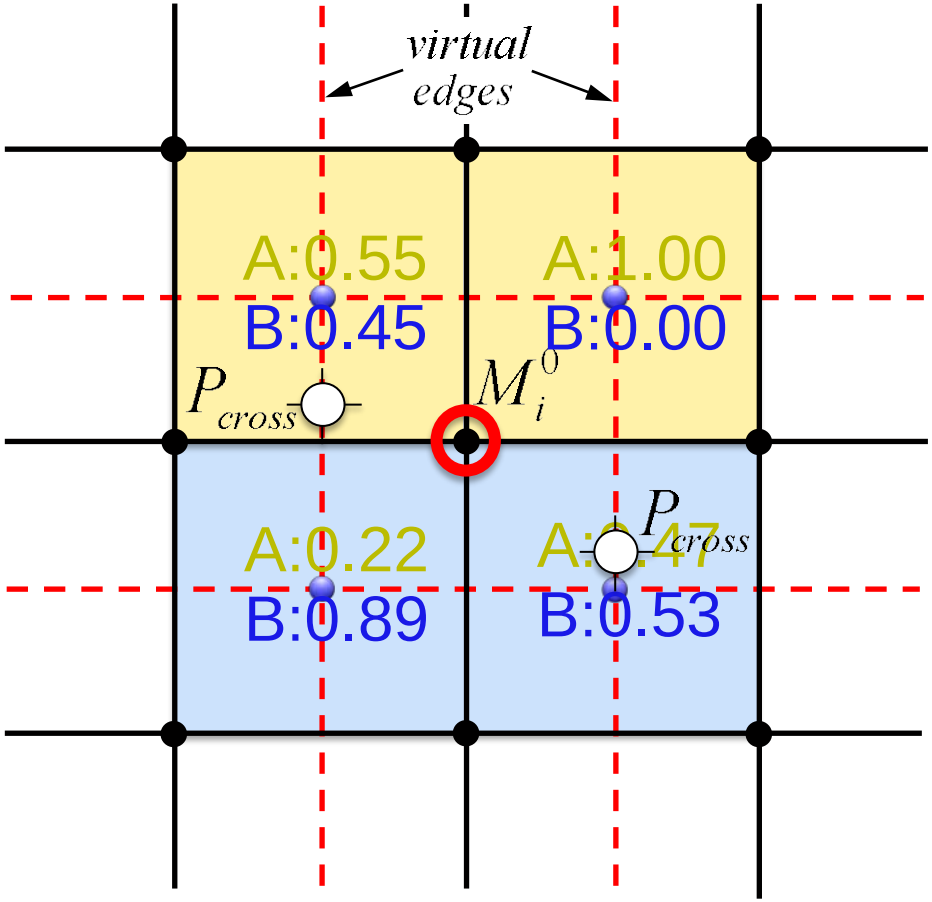
$$t = \frac{B_0 - A_0}{A_1 - A_0 - B_1 + B_0}$$



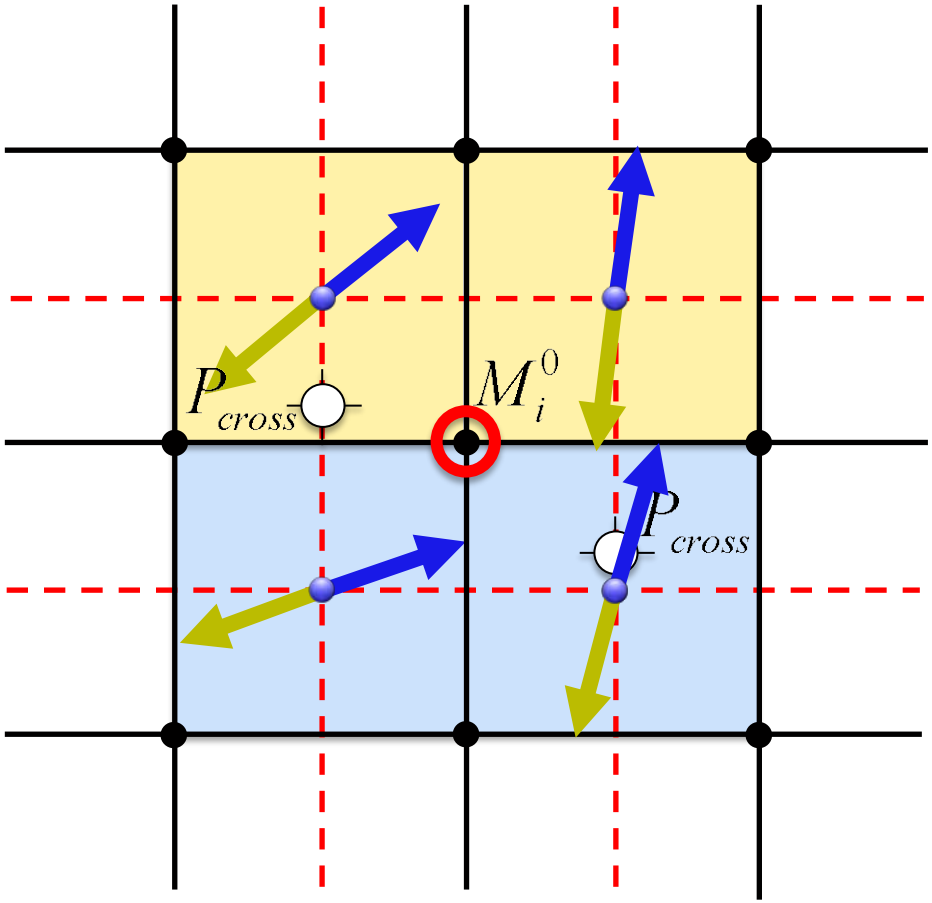
$$A(t) = A_0\left(2t^3 - 3t^2 + 1\right) + A_1\left(-2t^3 + 3t^2\right) + A_0'\left(t^3 - 2t^2 + t\right) + A_1'\left(t^3 - t^2\right)$$

$$B(t) = B_0\left(2t^3 - 3t^2 + 1\right) + B_1\left(-2t^3 + 3t^2\right) + B_0'\left(t^3 - 2t^2 + t\right) + B_1'\left(t^3 - t^2\right)$$

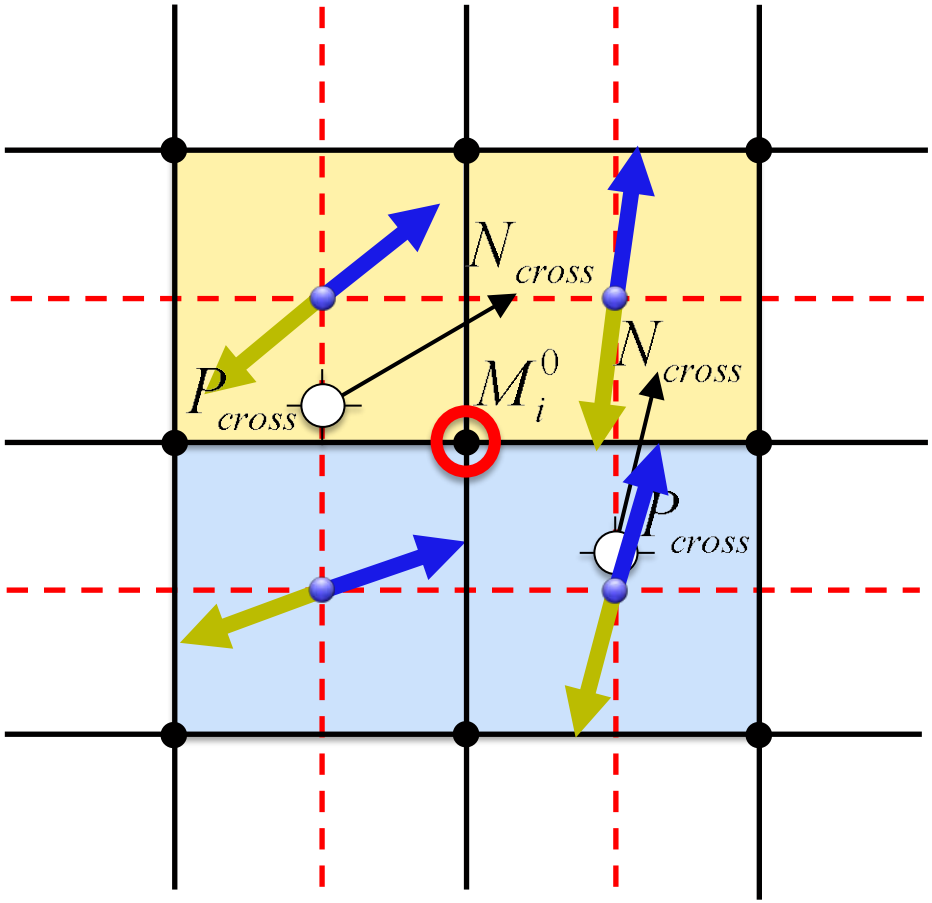
Compute Virtual Edge Crossings



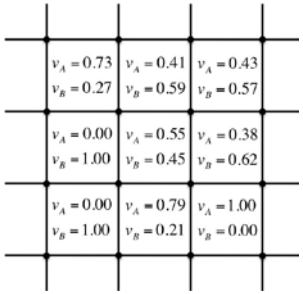
Compute Virtual Edge Crossings



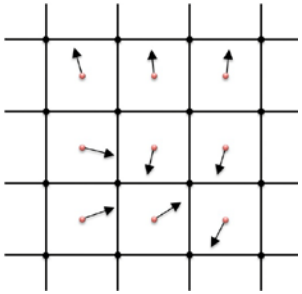
Compute Virtual Edge Crossings



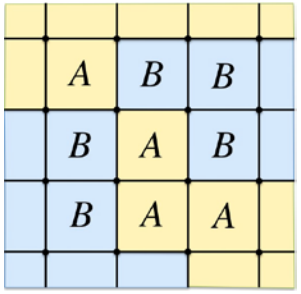
The Procedure



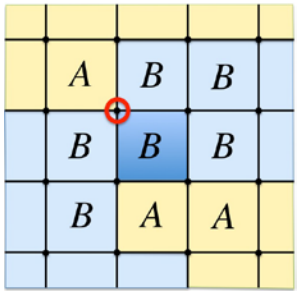
1 Establish Parallel Cartesian Grid



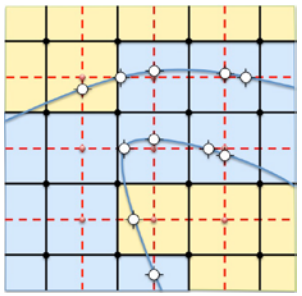
2 Estimate Gradients at Cell Centers



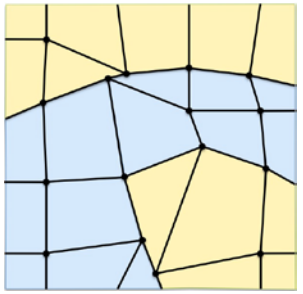
3 Assign Materials to Cells



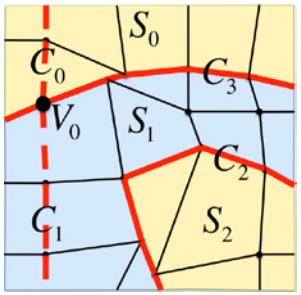
4 Resolve Non-Manifold Cases



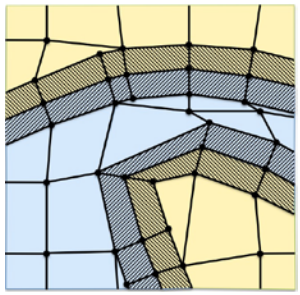
5 Compute Virtual Edge Crossings



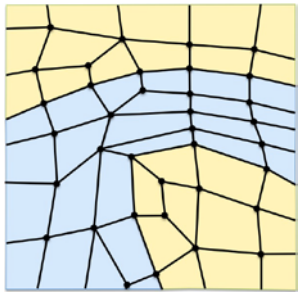
6 Move Grid Points to Iso-surface



7 Create Geometry Definition



8 Insert Hex Buffer Layer

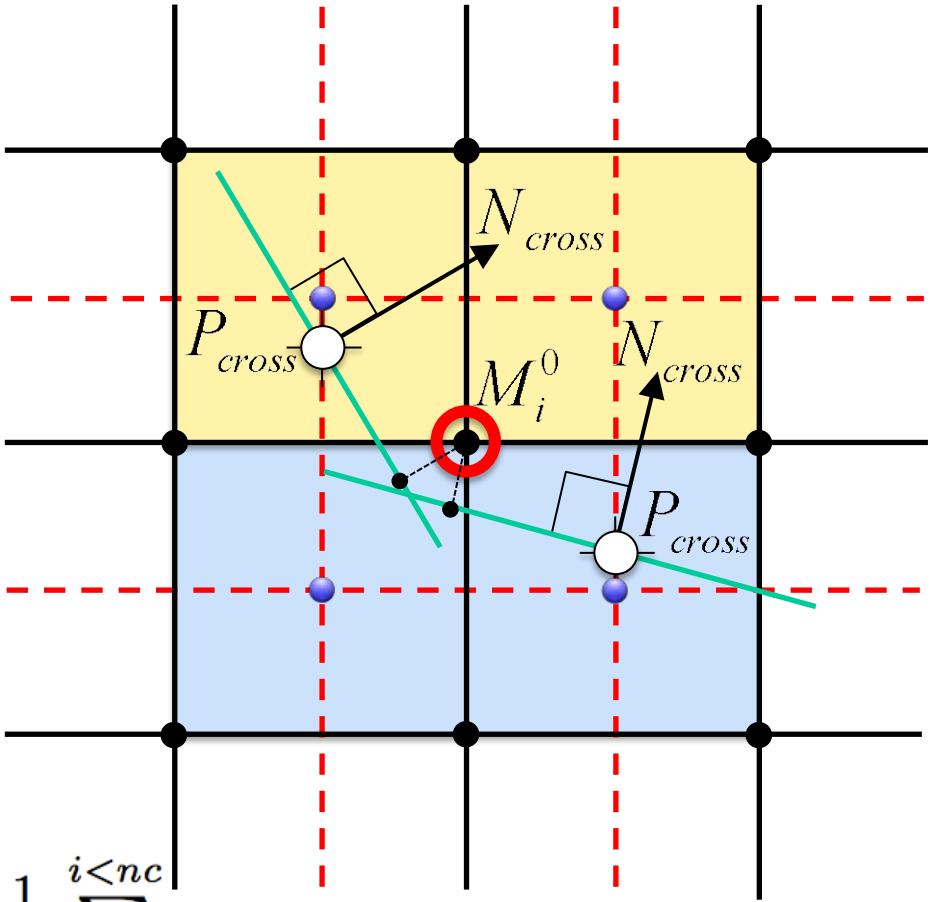


9 Smooth

LOCKHEED MARTIN

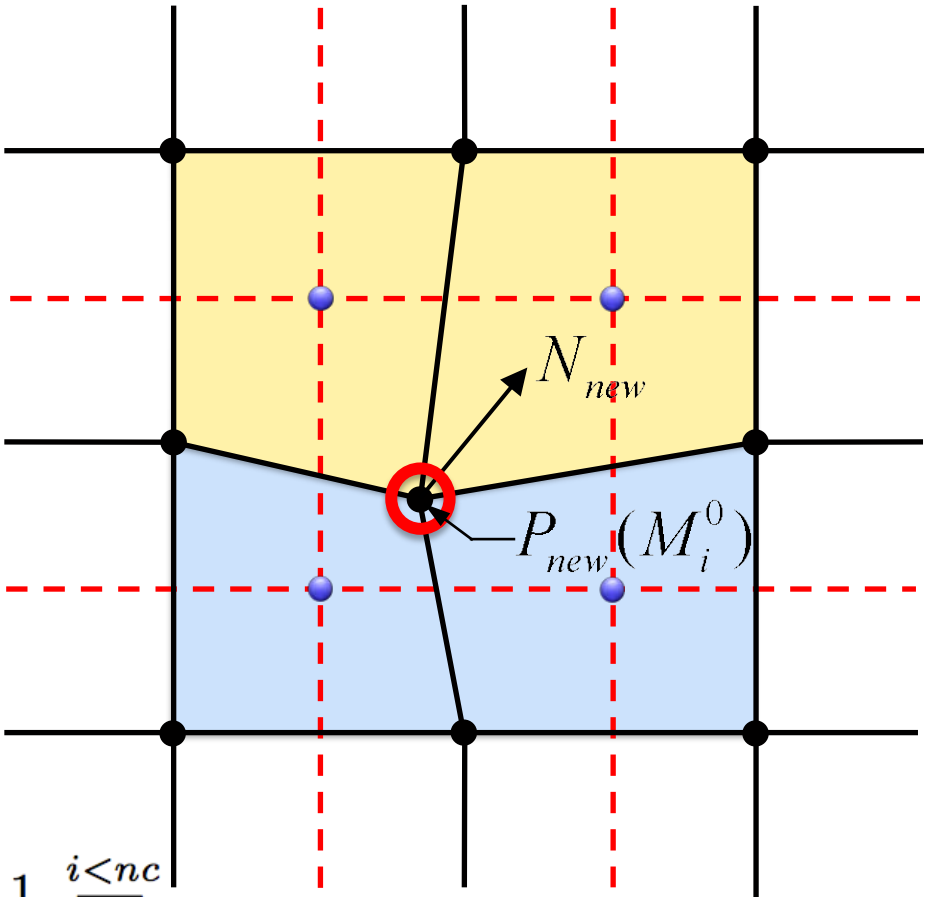
Sandia National Laboratories

Move Grid Points to Iso-Surface



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

Move Grid Points to Iso-Surface

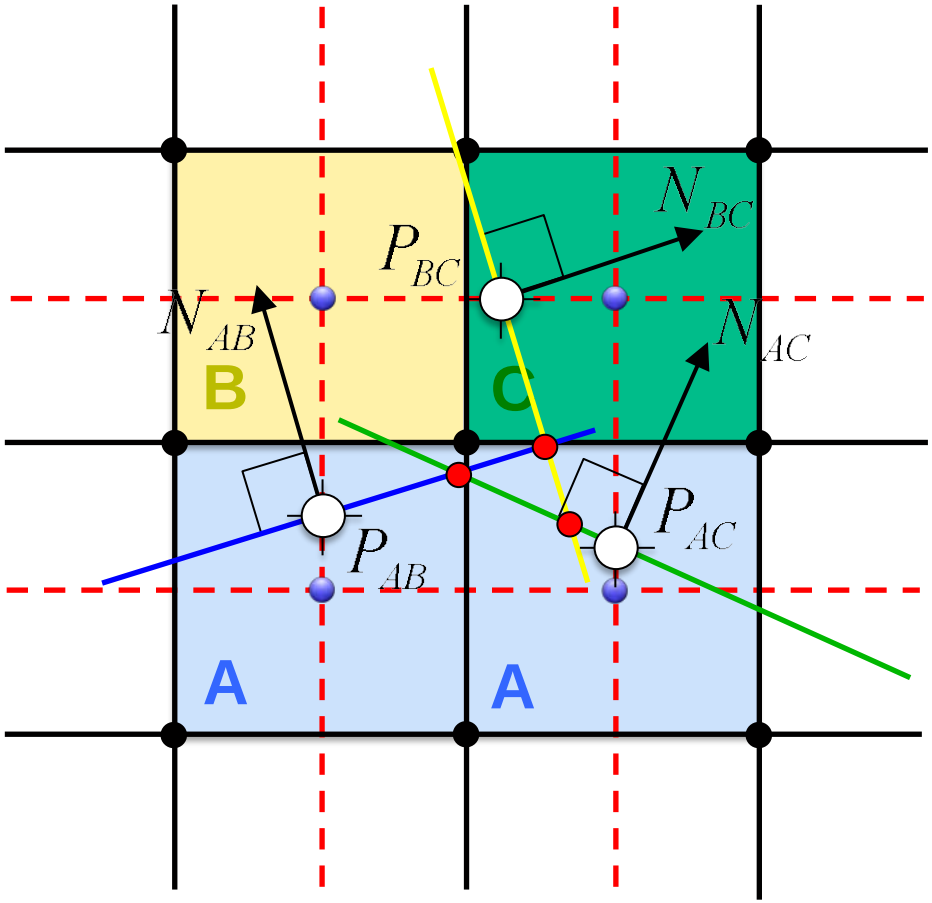


$$P_{new} = \frac{1}{nc} \sum_{i=0}^{i<nc} P_0 - (N_{cross})_i \cdot (P_0 - (P_{cross})_i) \times (N_{cross})_i$$

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

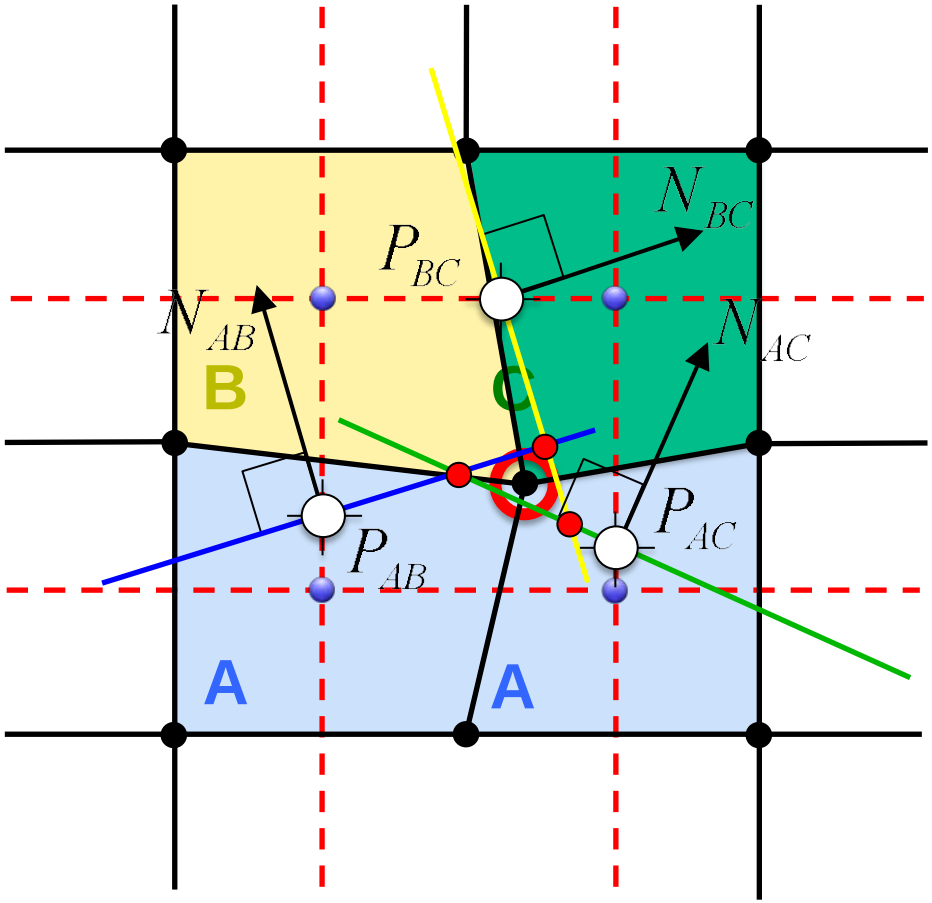
Move Grid Points to Iso-Surface

Multiple Materials

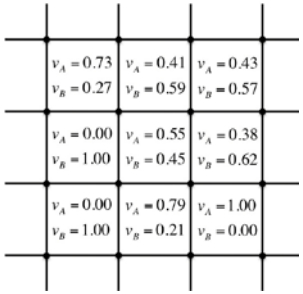


Move Grid Points to Iso-Surface

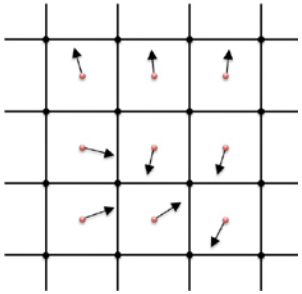
Multiple Materials



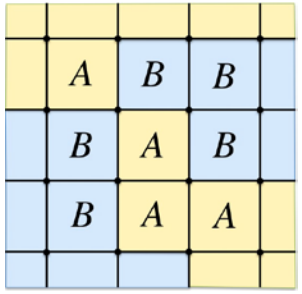
The Procedure



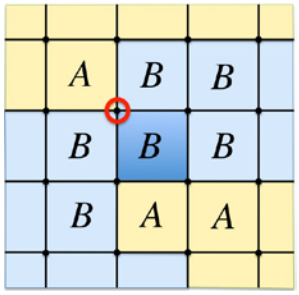
1 Establish Parallel Cartesian Grid



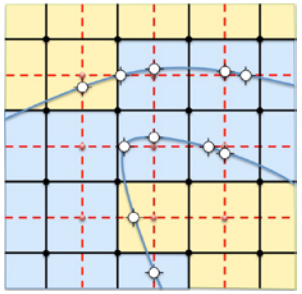
2 Estimate Gradients at Cell Centers



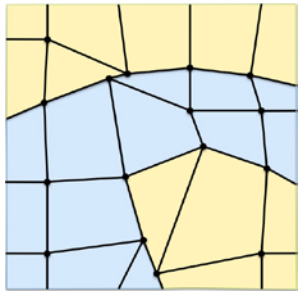
3 Assign Materials to Cells



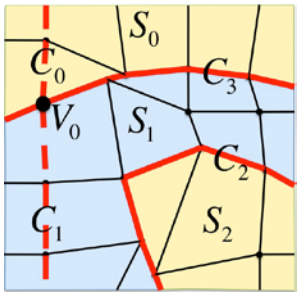
4 Resolve Non-Manifold Cases



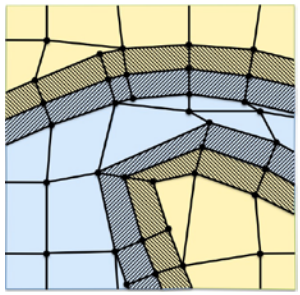
5 Compute Virtual Edge Crossings



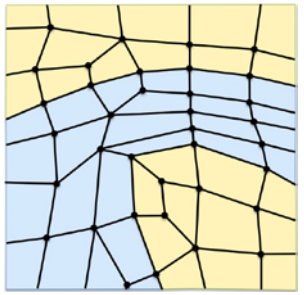
6 Move Grid Points to Iso-surface



7 Create Geometry Definition



8 Insert Hex Buffer Layer

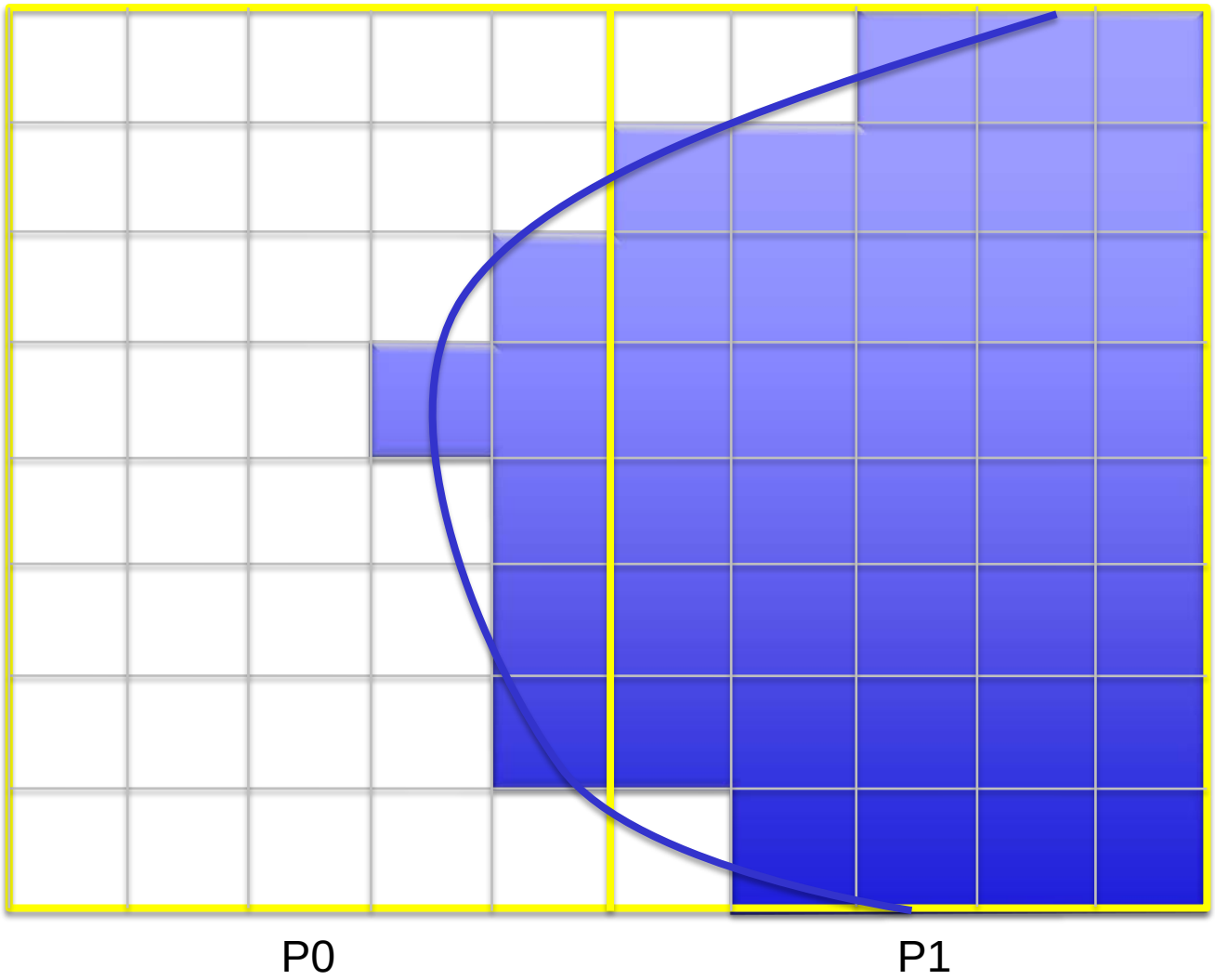


9 Smooth

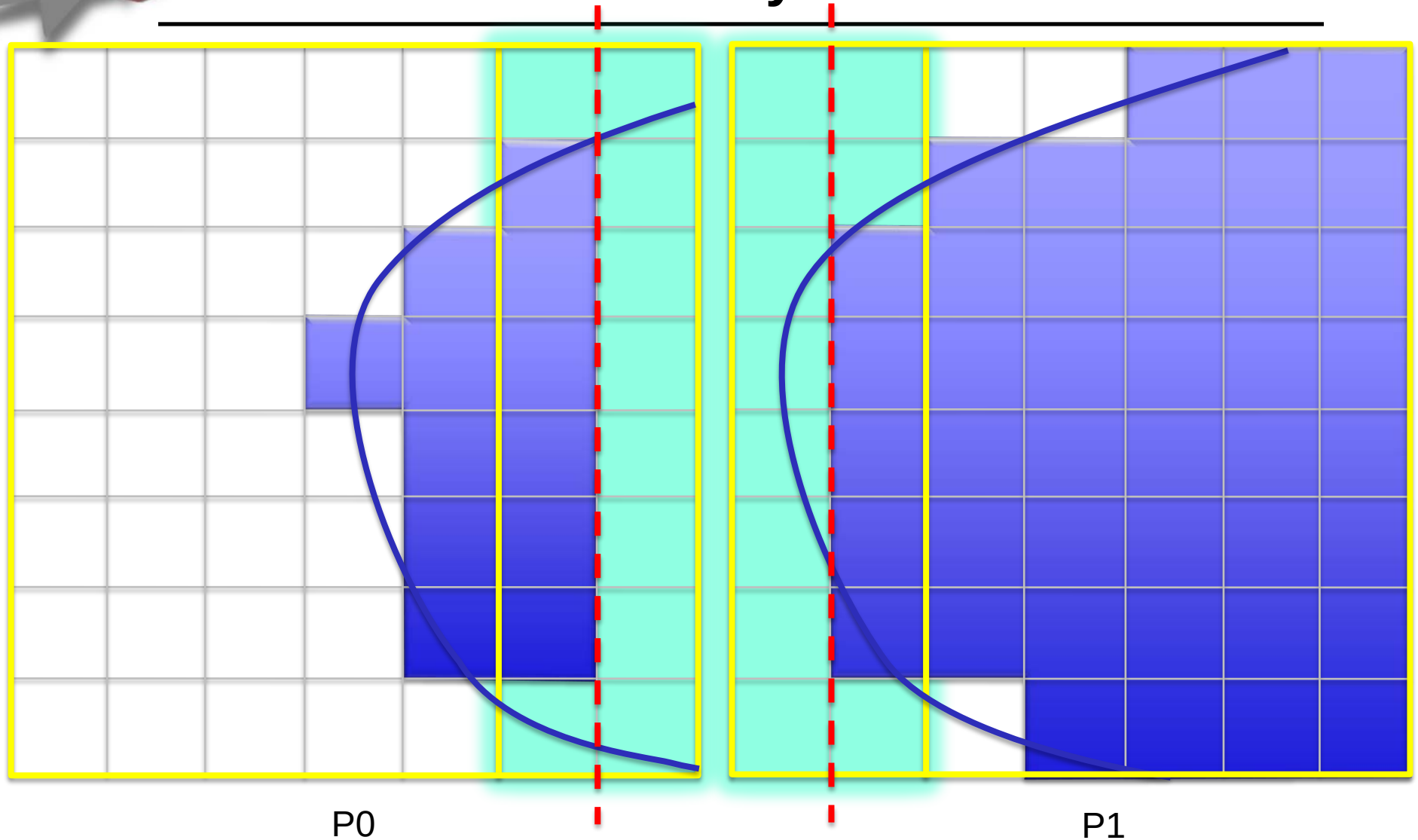
LOCKHEED MARTIN

Sandia National Laboratories

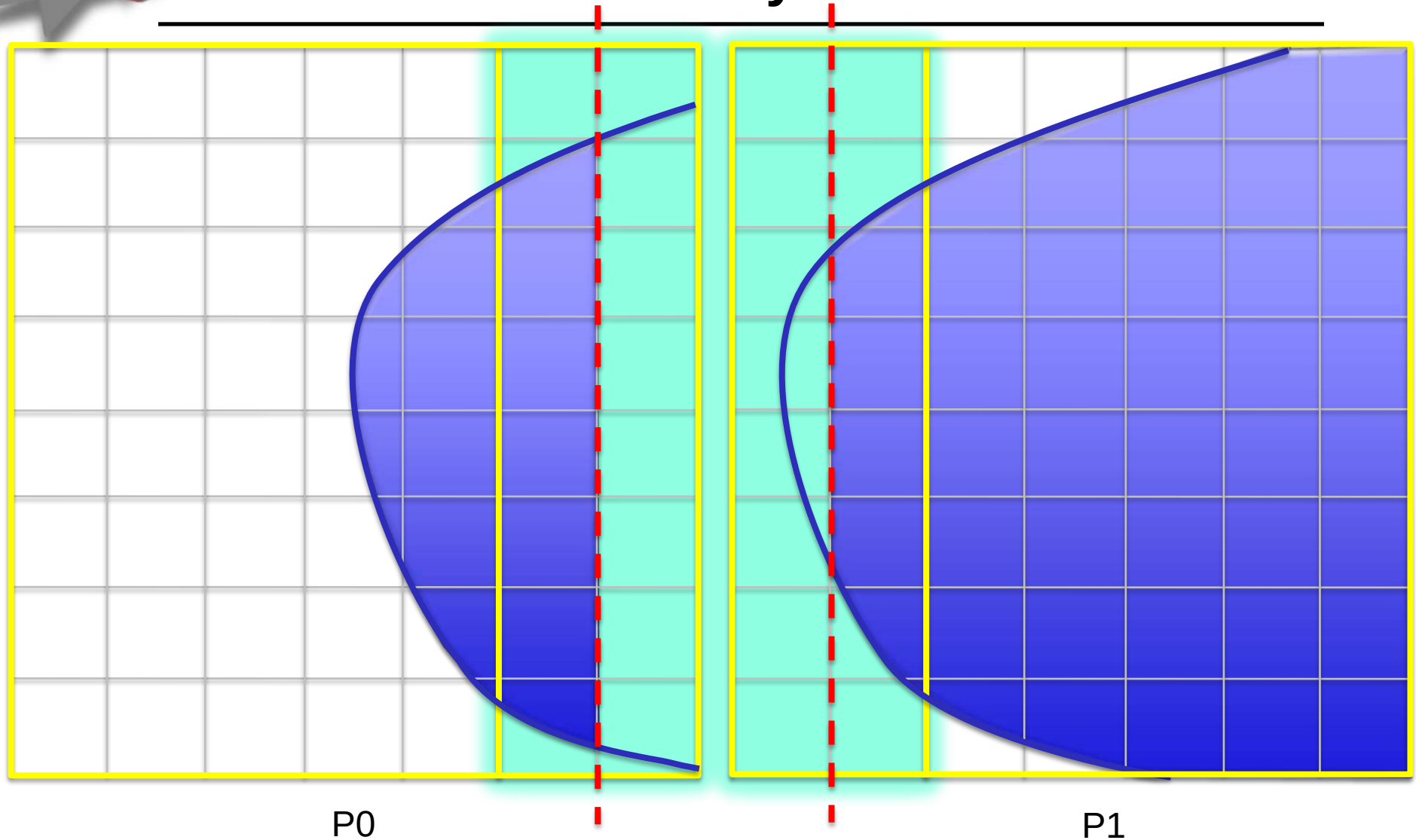
Create Geometry Definition



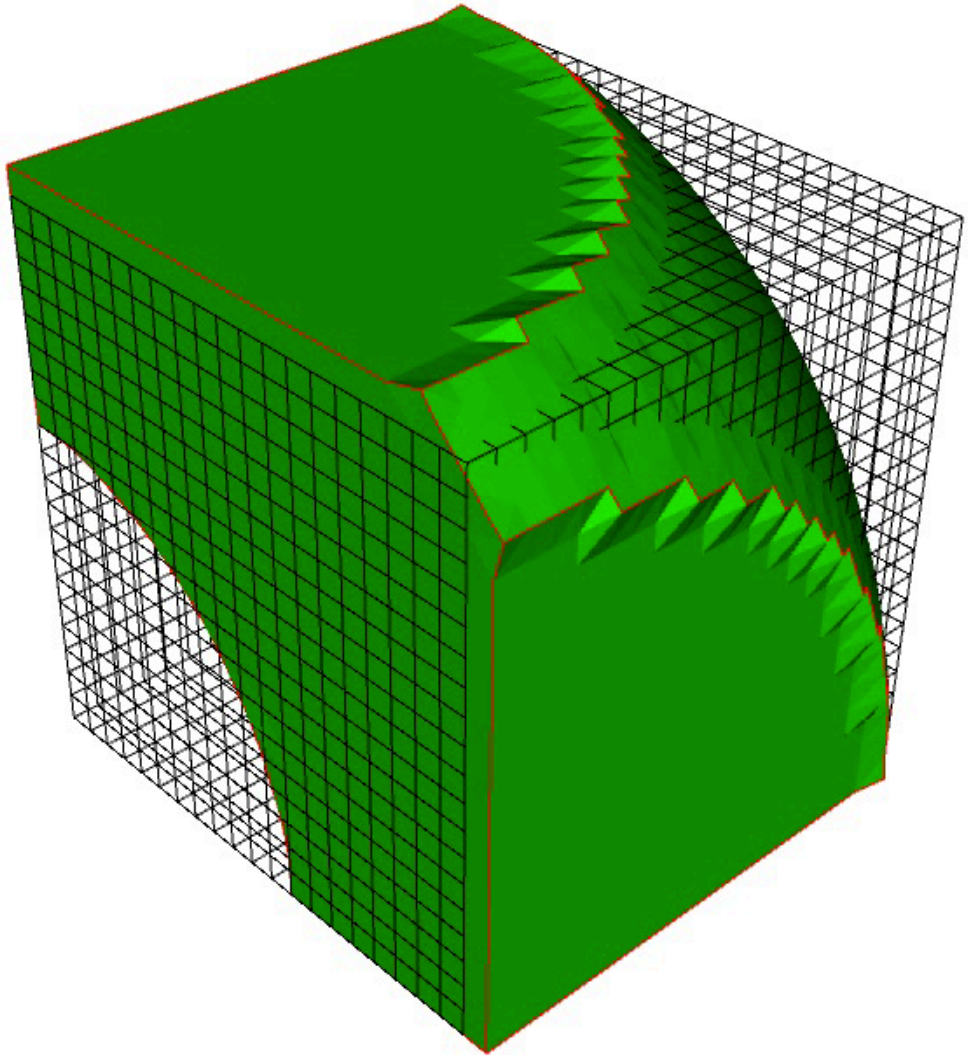
Create Geometry Definition



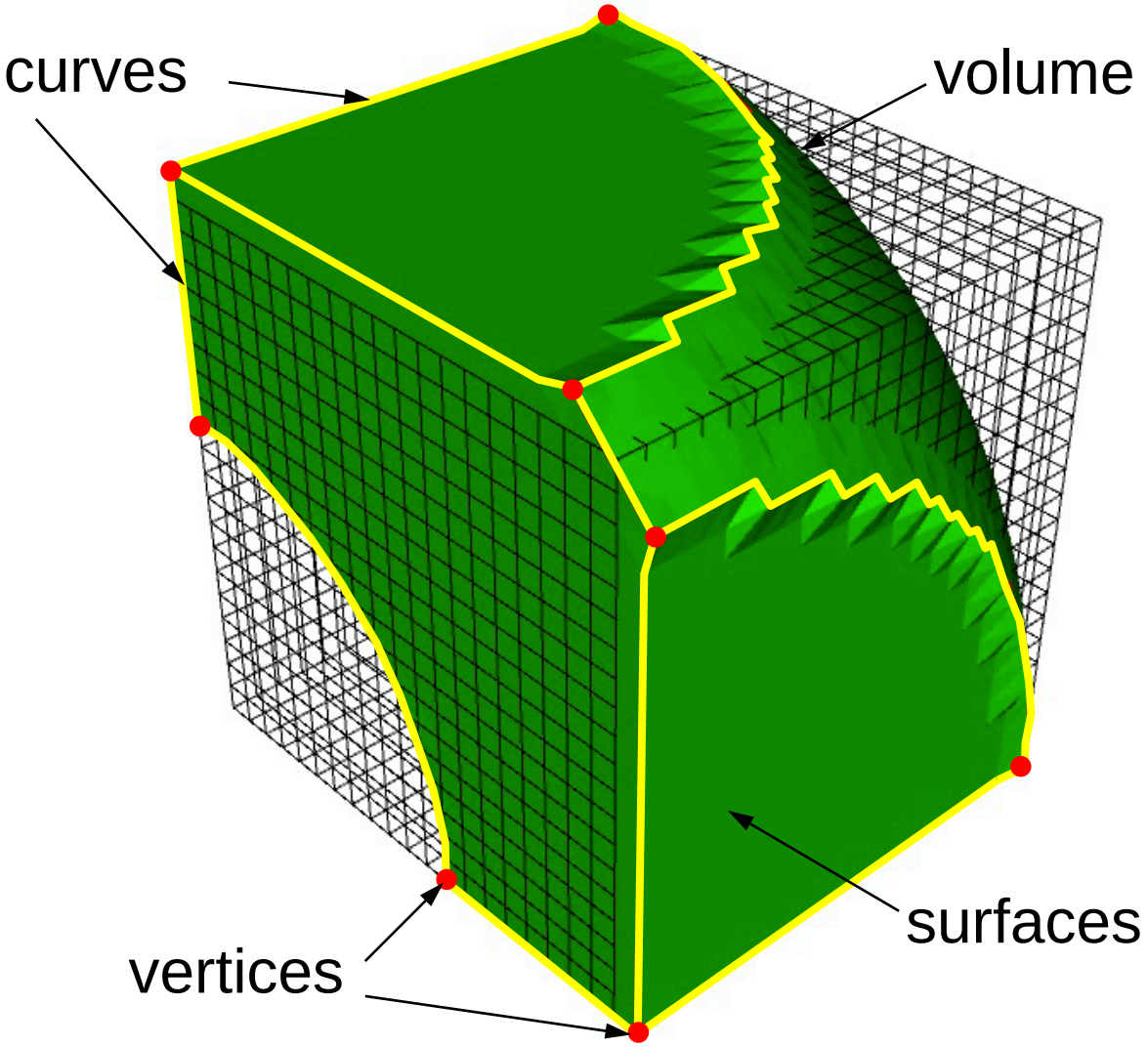
Create Geometry Definition



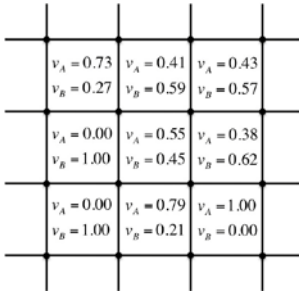
Create Geometry Definition



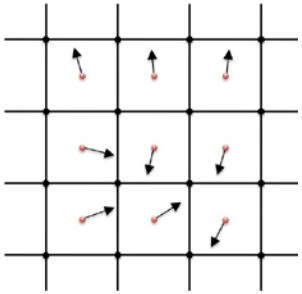
Create Geometry Definition



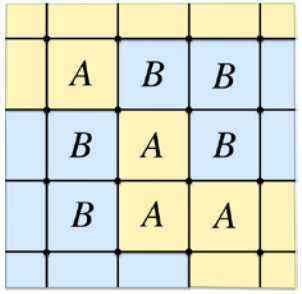
The Procedure



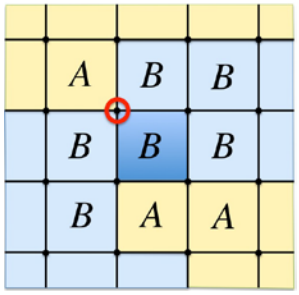
1 Establish Parallel Cartesian Grid



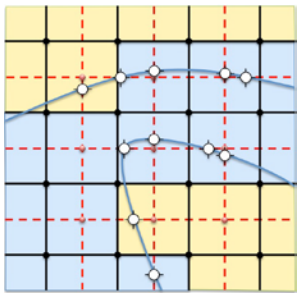
2 Estimate Gradients at Cell Centers



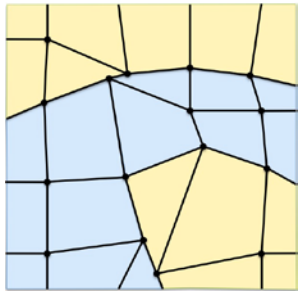
3 Assign Materials to Cells



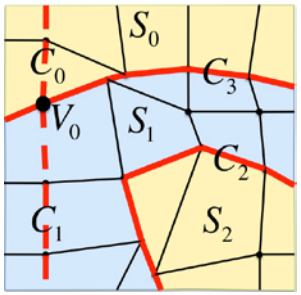
4 Resolve Non-Manifold Cases



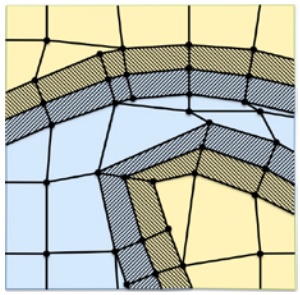
5 Compute Virtual Edge Crossings



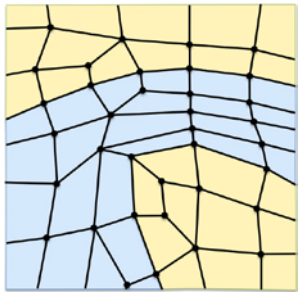
6 Move Grid Points to Iso-surface



7 Create Geometry Definition



8 Insert Hex Buffer Layer

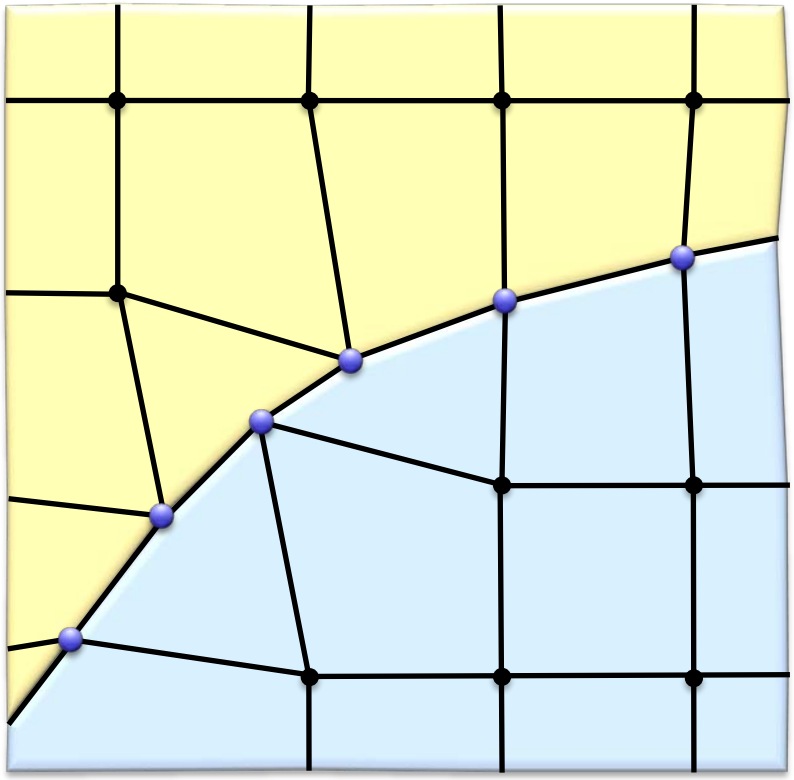


9 Smooth

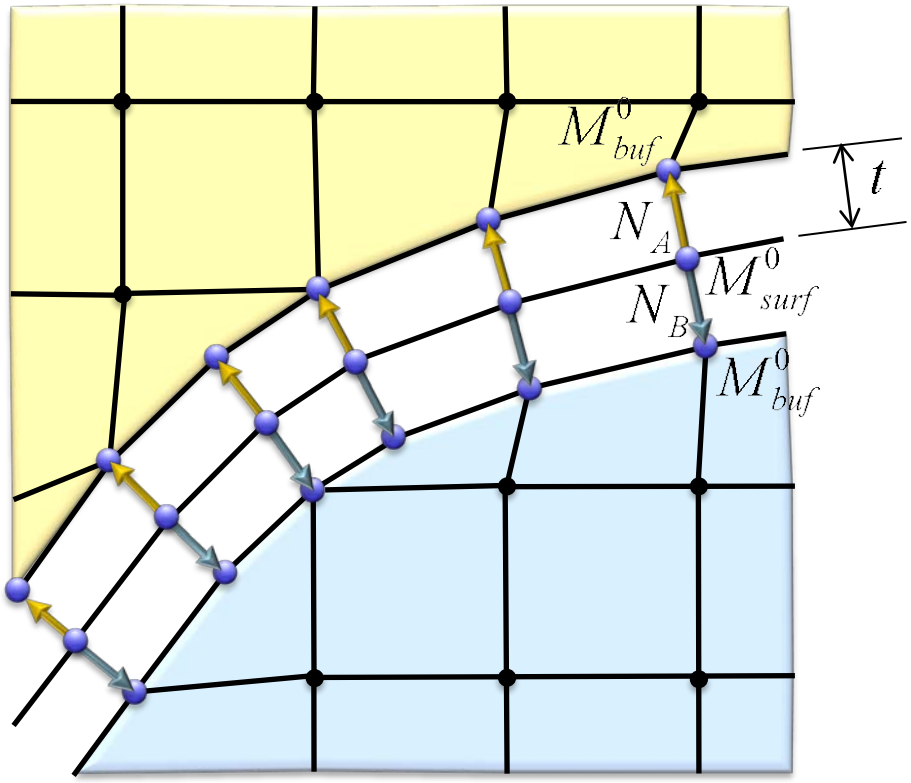
LOCKHEED MARTIN

Sandia National Laboratories

Insert Hex Buffer Layer

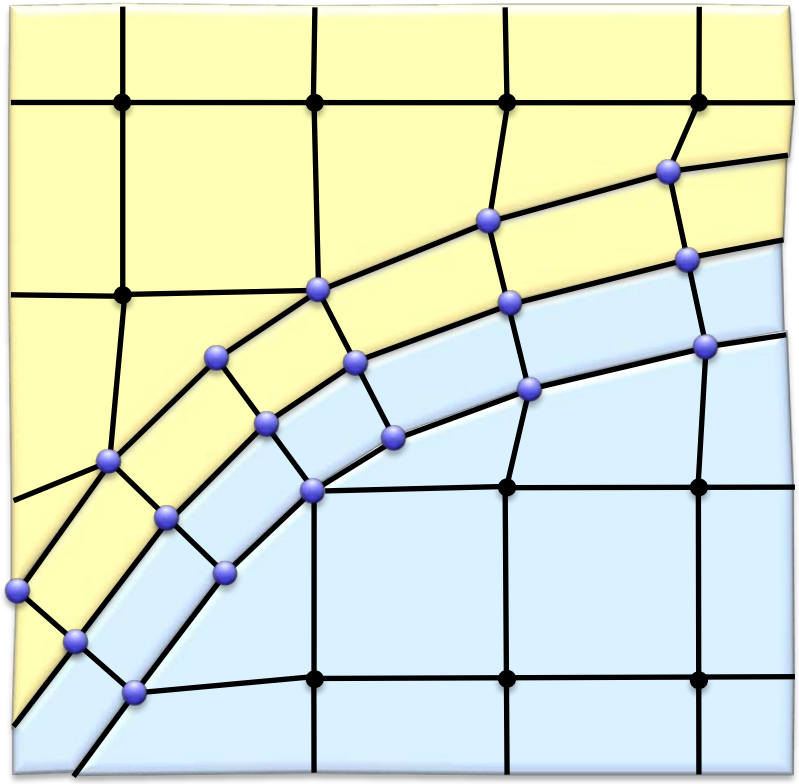


Insert Hex Buffer Layer

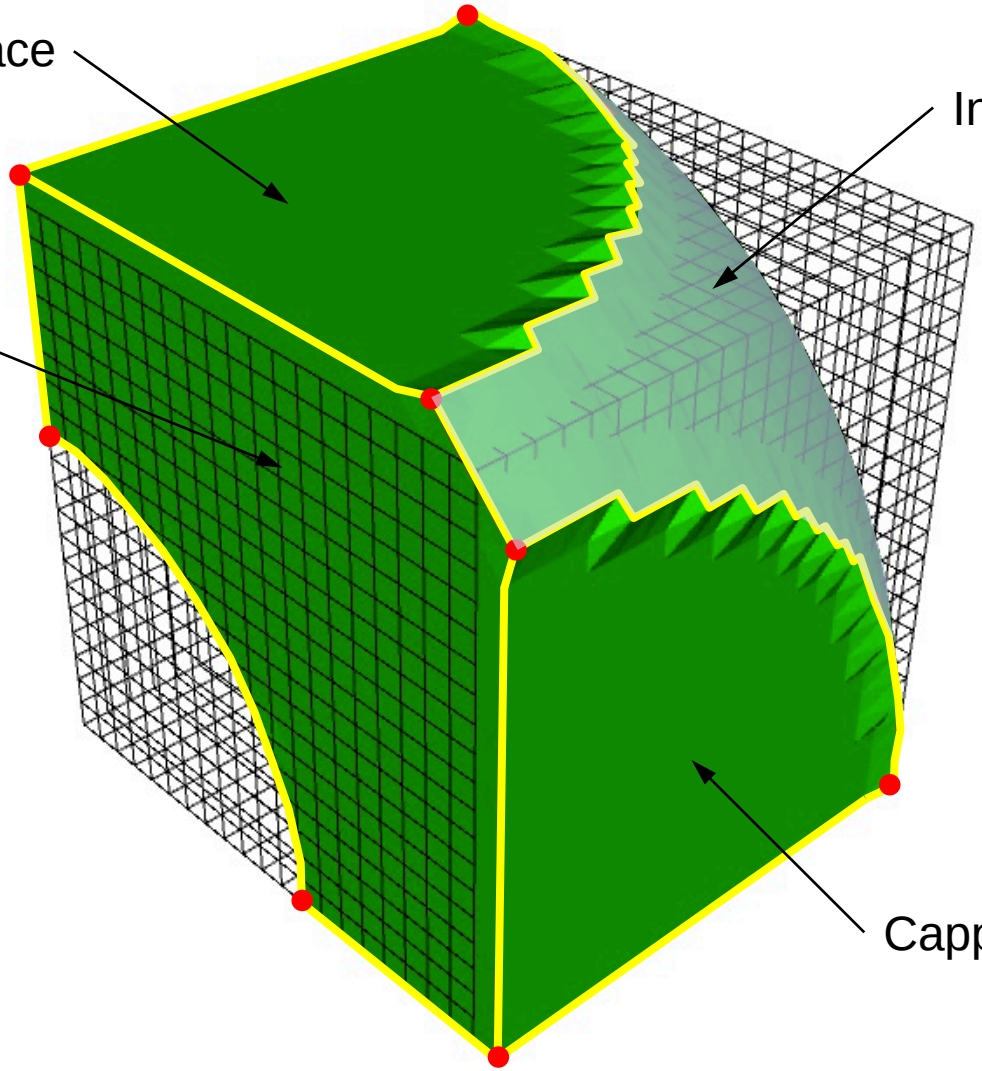


$$P(M^0_{buf})_n = P(M^0_{surf}) + tN(M^0_{surf})_n$$

Insert Hex Buffer Layer



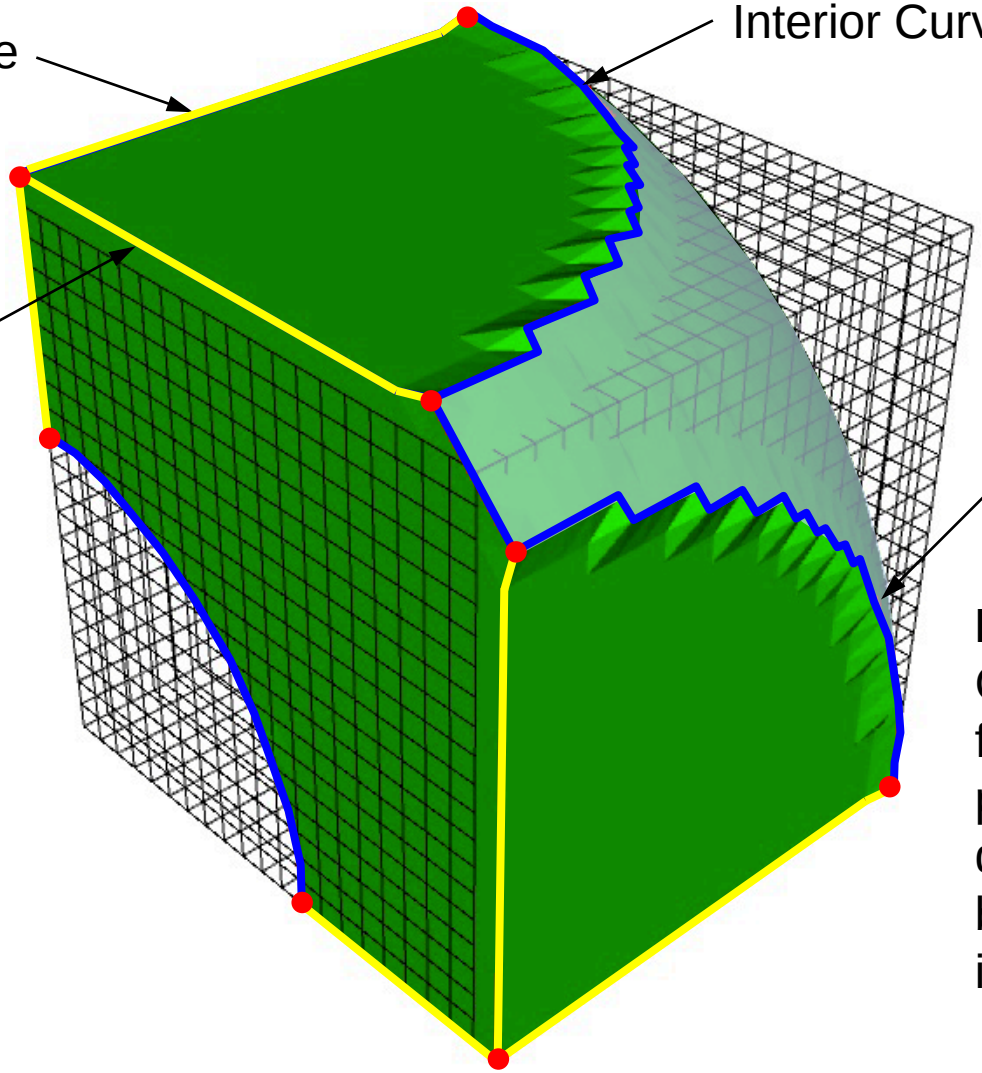
Insert Hex Buffer Layer



Interior Surface: Surface described from iso-surface

Capping Surface: Surface at the interface between two processors or at the domain boundary

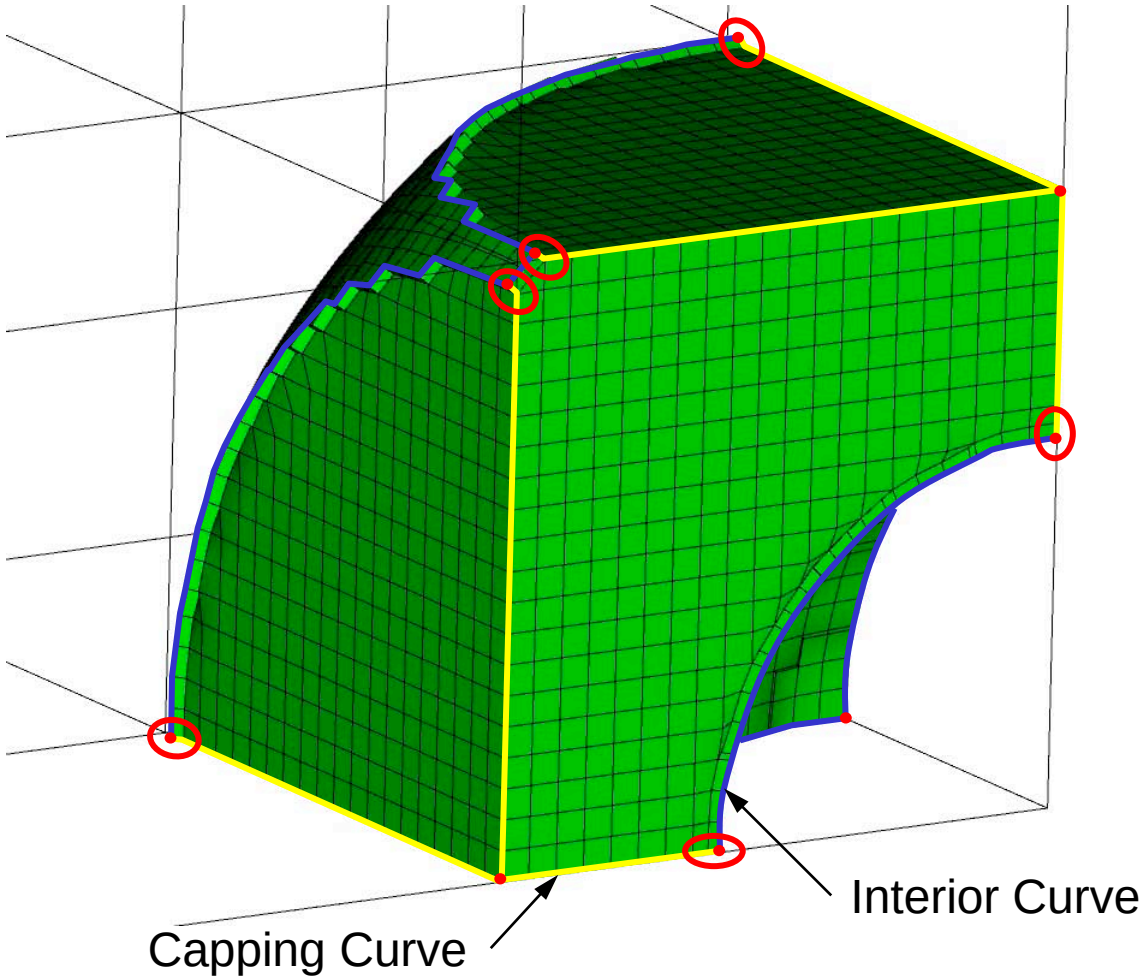
Insert Hex Buffer Layer



Capping Curve: Curve on the edge of a processor domain

Interior Curve: Curve on the face of a processor domain (curves bounding an interior surface)

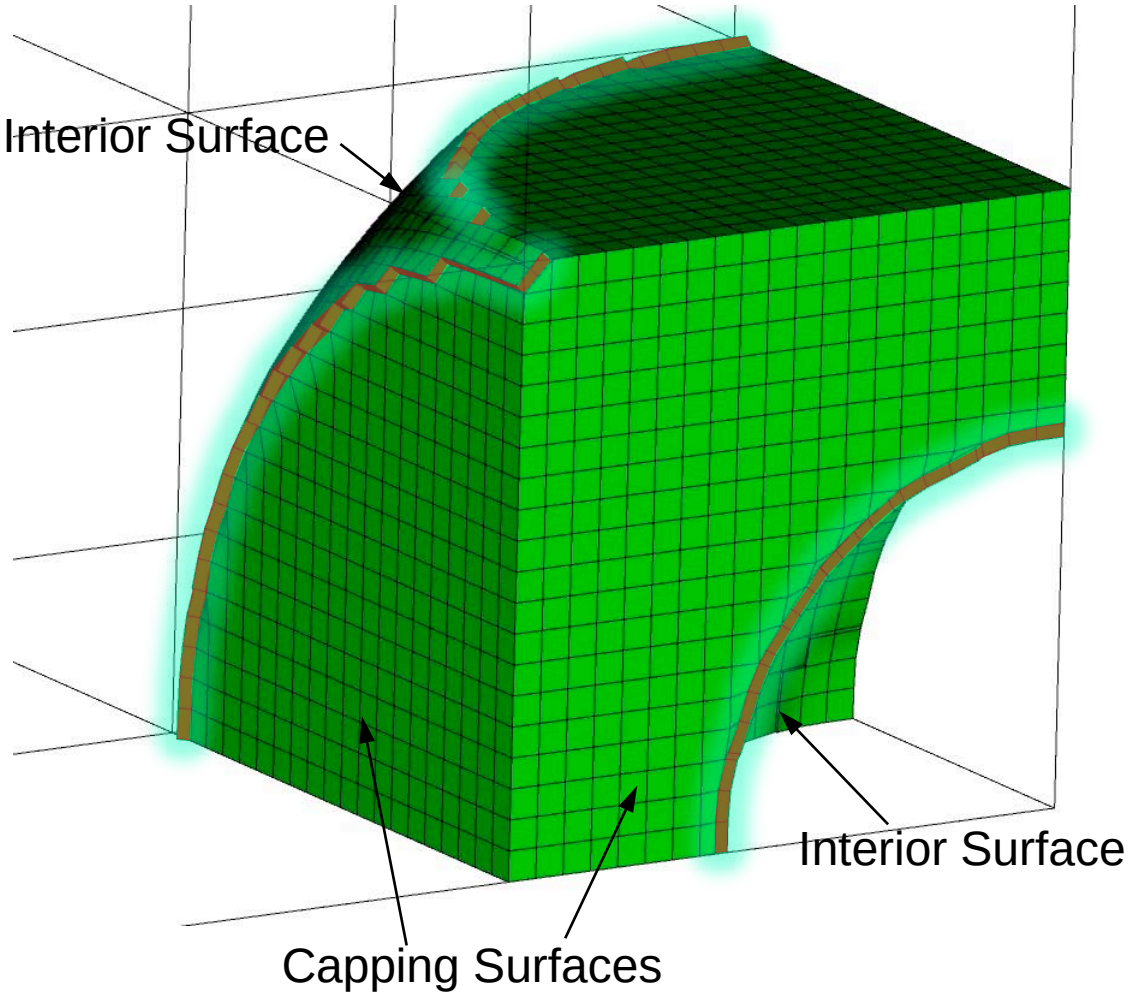
Insert Hex Buffer Layer



Bottom-up creation of hexes

Insert buffer edge on capping curves at vertices

Insert Hex Buffer Layer



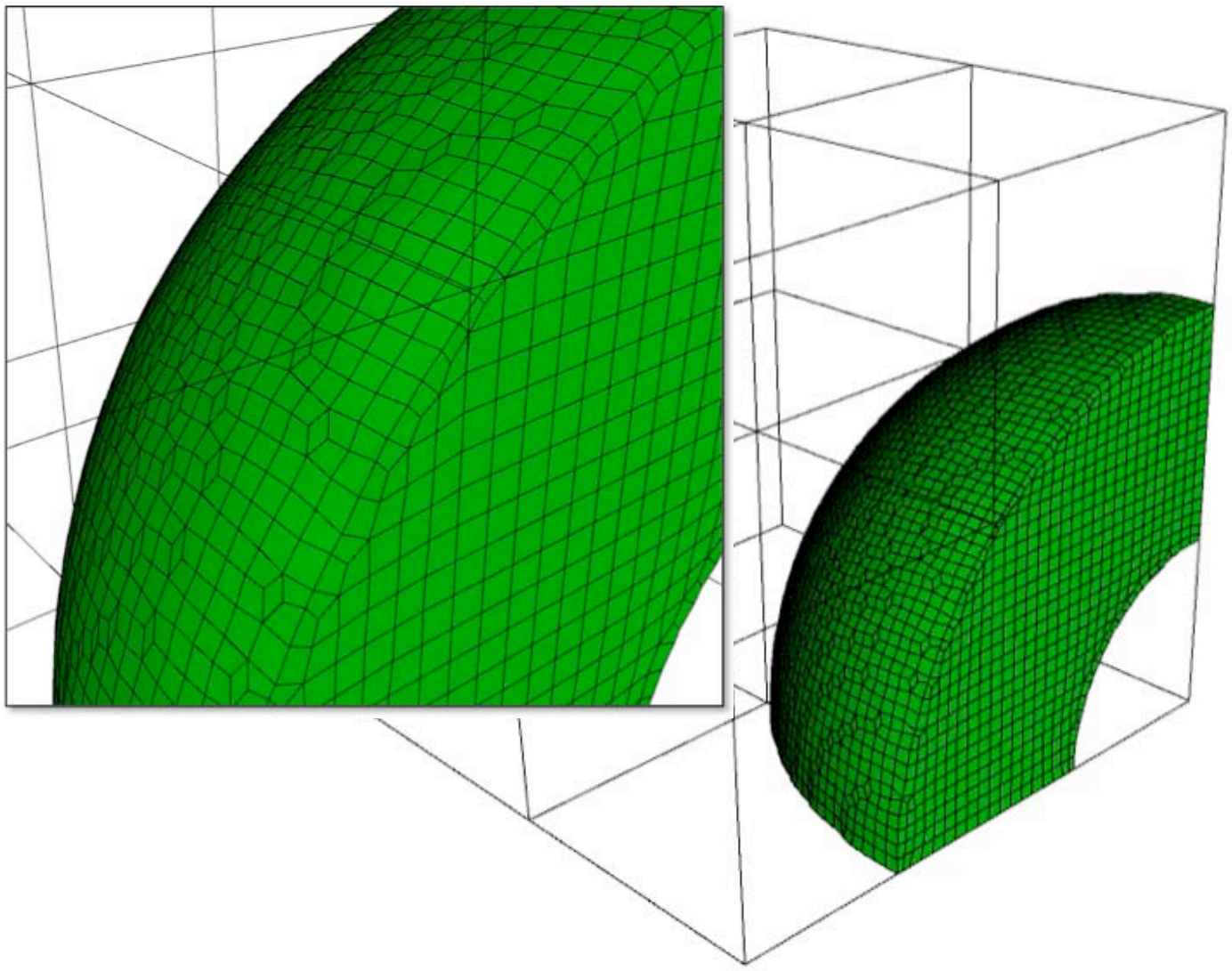
Bottom-up creation of hexes

Insert buffer edge on capping curves at vertices

Insert buffer quad on capping surfaces at interior curves

Insert buffer hex in volume at interior surfaces

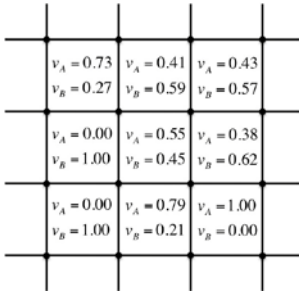
Insert Hex Buffer Layer



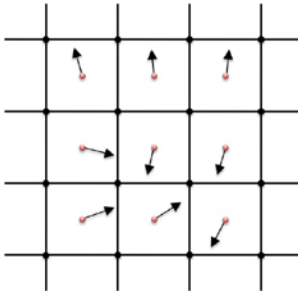
LOCKHEED MARTIN

Sandia National Laboratories

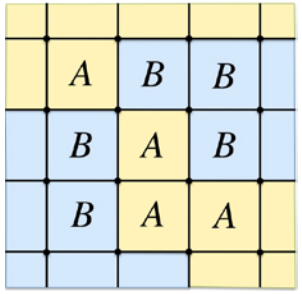
The Procedure



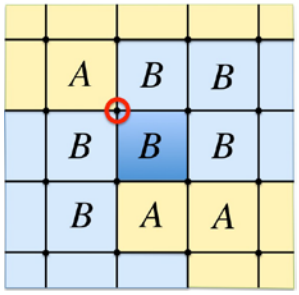
1 Establish Parallel Cartesian Grid



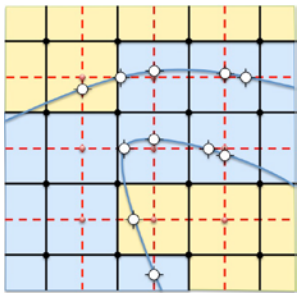
2 Estimate Gradients at Cell Centers



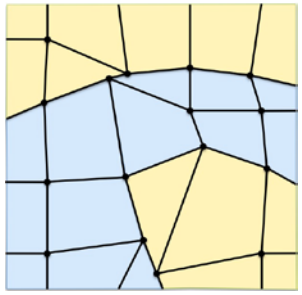
3 Assign Materials to Cells



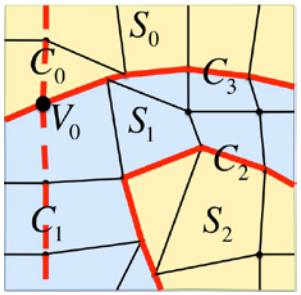
4 Resolve Non-Manifold Cases



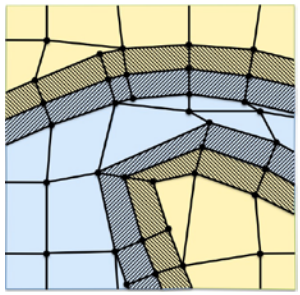
5 Compute Virtual Edge Crossings



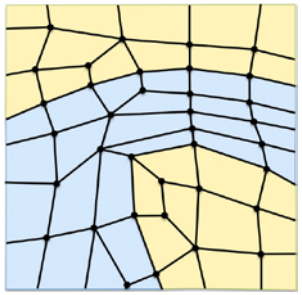
6 Move Grid Points to Iso-surface



7 Create Geometry Definition



8 Insert Hex Buffer Layer

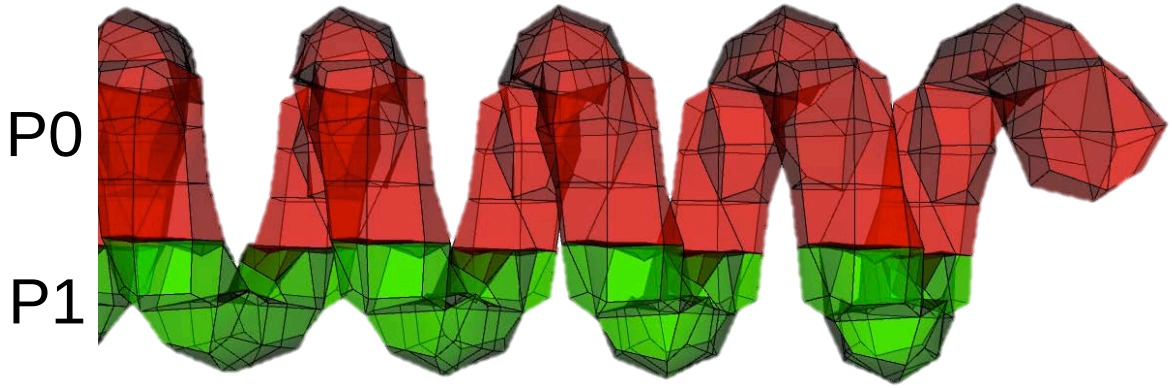


9 Smooth

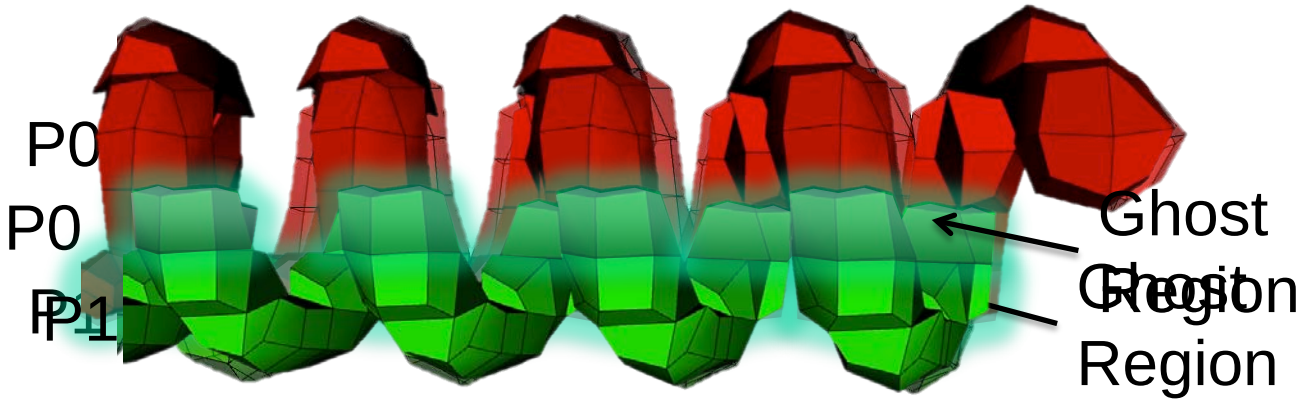
LOCKHEED MARTIN

Sandia National Laboratories

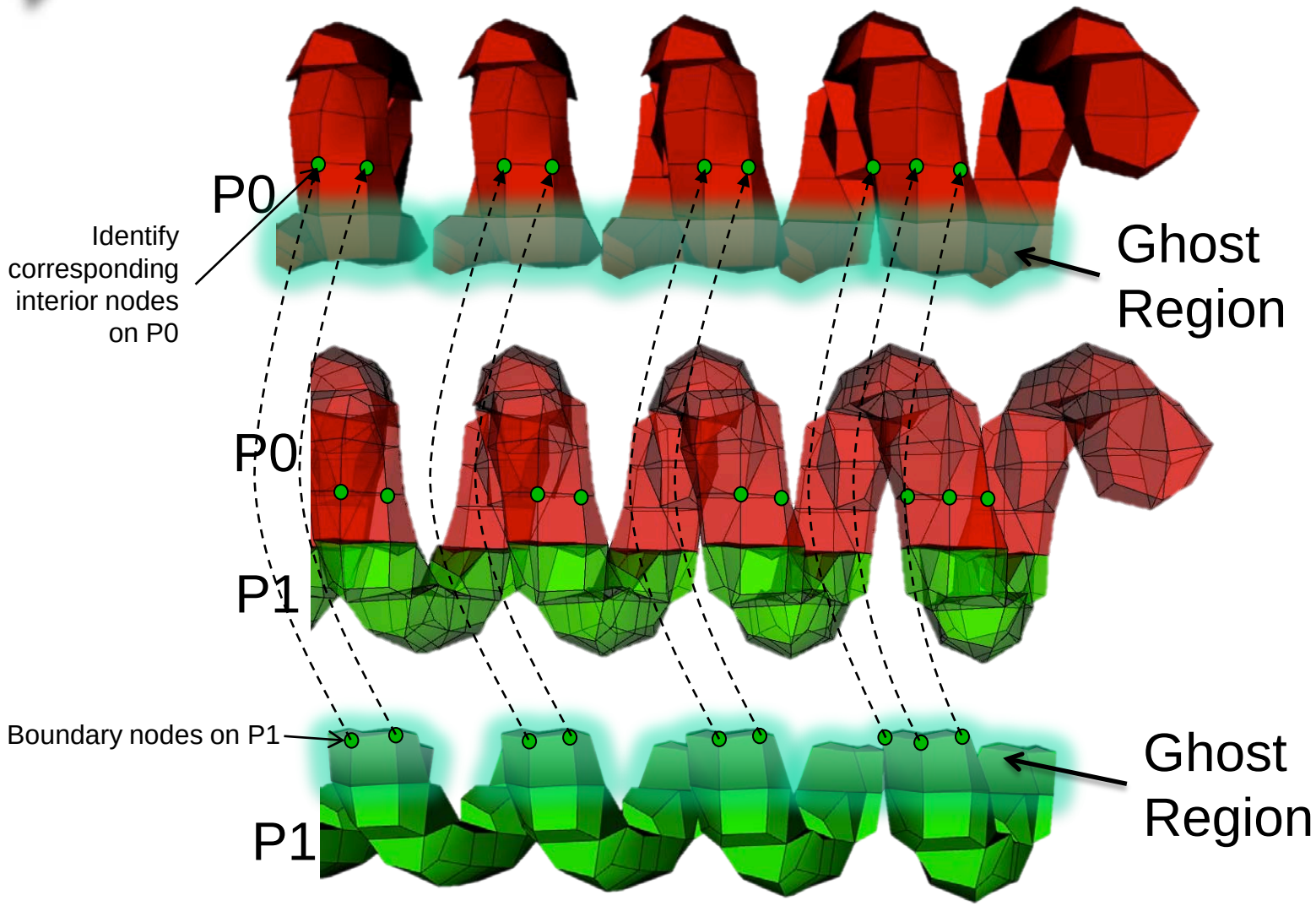
MPI Communication for Smoothing



MPI Communication for Smoothing



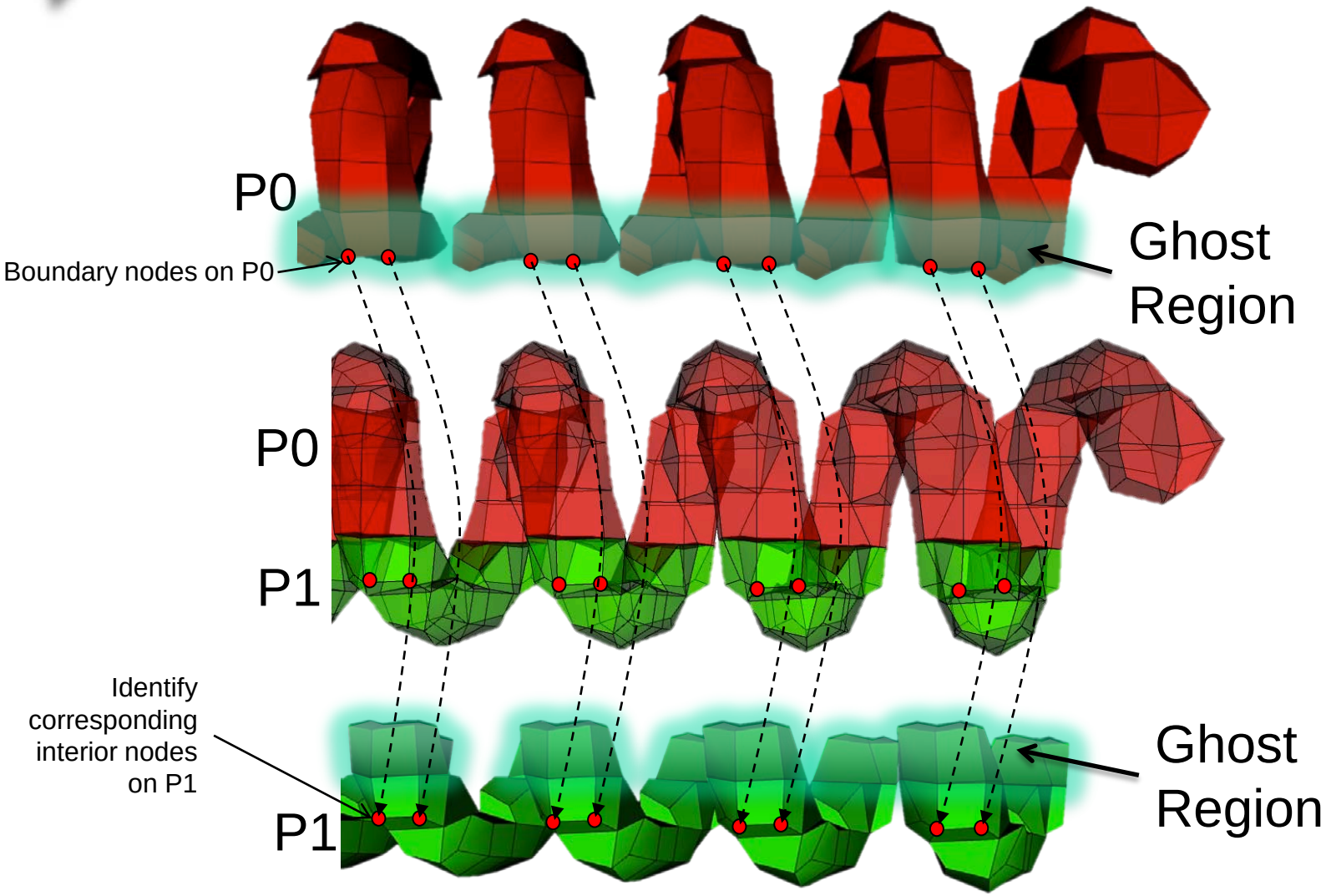
MPI Communication for Smoothing



$L^{0\to1}$

Node locations on P0 used to update boundary nodes on P1

MPI Communication for Smoothing



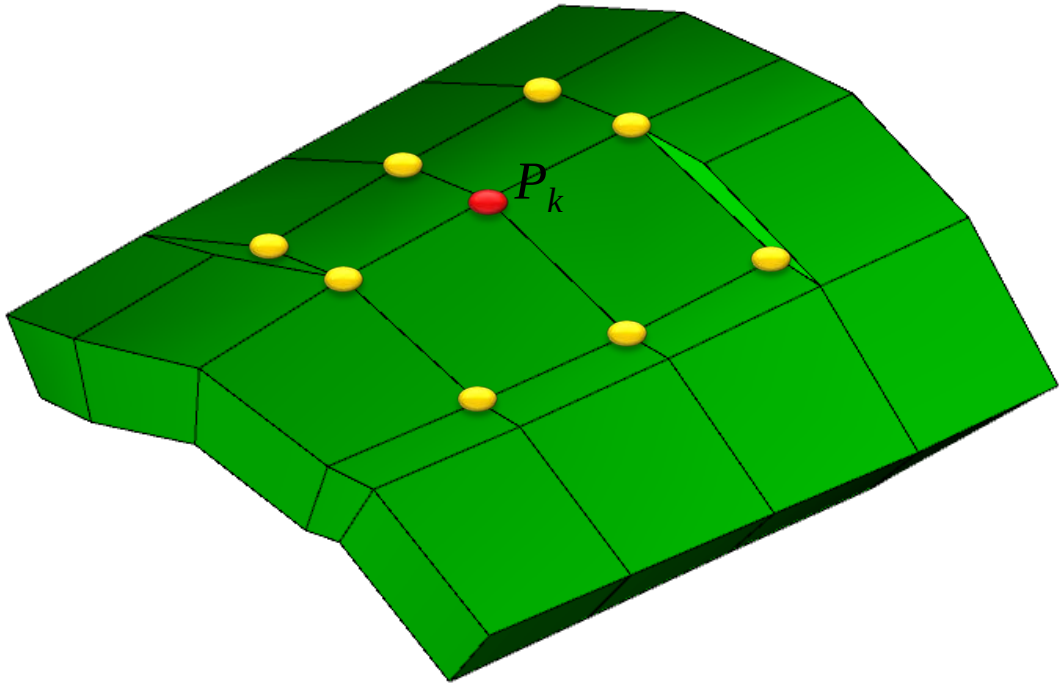
$L^{0\to1}$

Node locations on P0 used to update boundary nodes on P1

$L^{1\to0}$

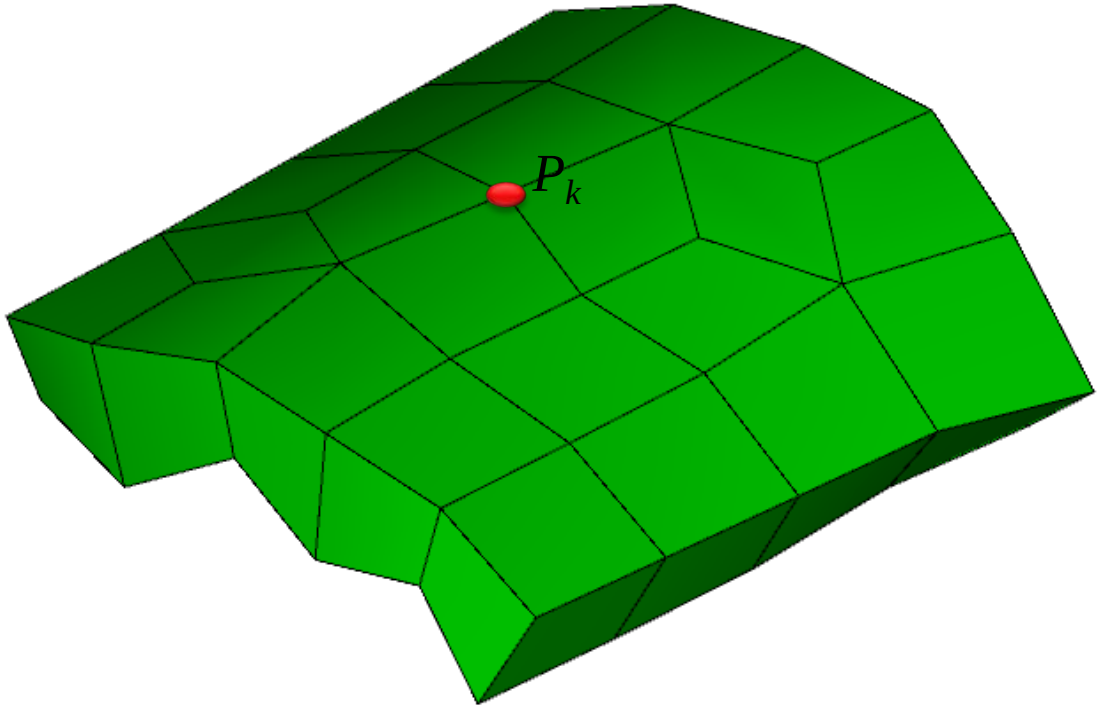
Node locations on P1 used to update boundary nodes on P0

Surface Smoothing



Compute Quadric Patch approximating smooth surface at node

Surface Smoothing

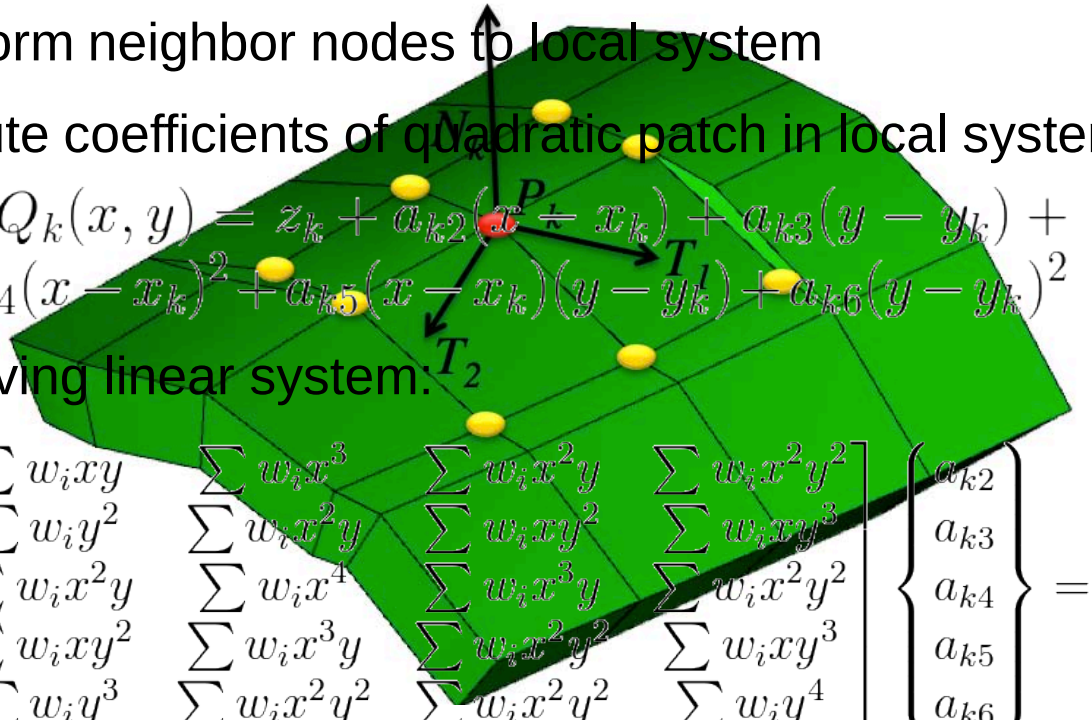


Compute Quadric Patch approximating smooth surface at node

Surface Smoothing

1. A local coordinate system $\{T_1, T_2, N_k\}$ at P_k is defined
2. Transform neighbor nodes to local system
3. Compute coefficients of quadratic patch in local system:

$$Q_k(x,y) = z_k + a_{k2}(x - x_k) + a_{k3}(y - y_k) + a_{k4}(x - x_k)^2 + a_{k5}(x - x_k)(y - y_k) + a_{k6}(y - y_k)^2$$



by solving linear system:

$$\begin{bmatrix} \sum w_i x^2 & \sum w_i xy & \sum w_i x^3 & \sum w_i x^2 y & \sum w_i x^2 y^2 \\ \sum w_i xy & \sum w_i y^2 & \sum w_i x^2 y & \sum w_i xy^2 & \sum w_i xy^3 \\ \sum w_i x^3 & \sum w_i x^2 y & \sum w_i x^4 & \sum w_i x^3 y & \sum w_i x^2 y^2 \\ \sum w_i x^2 y & \sum w_i xy^2 & \sum w_i x^3 y & \sum w_i x^2 y^2 & \sum w_i xy^3 \\ \sum w_i xy^2 & \sum w_i y^3 & \sum w_i x^2 y^2 & \sum w_i x^2 y^2 & \sum w_i y^4 \end{bmatrix} \begin{Bmatrix} a_{k2} \\ a_{k3} \\ a_{k4} \\ a_{k5} \\ a_{k6} \end{Bmatrix} = \begin{Bmatrix} \sum w_i xz \\ \sum w_i yz \\ \sum w_i x^2 z \\ \sum w_i xyz \\ \sum w_i y^2 z \end{Bmatrix}$$

where $x = x_i - x_k, y = y_i - y_k, z = z_i - z_k$

and w_i is an inverse distance weighting.

Surface Smoothing

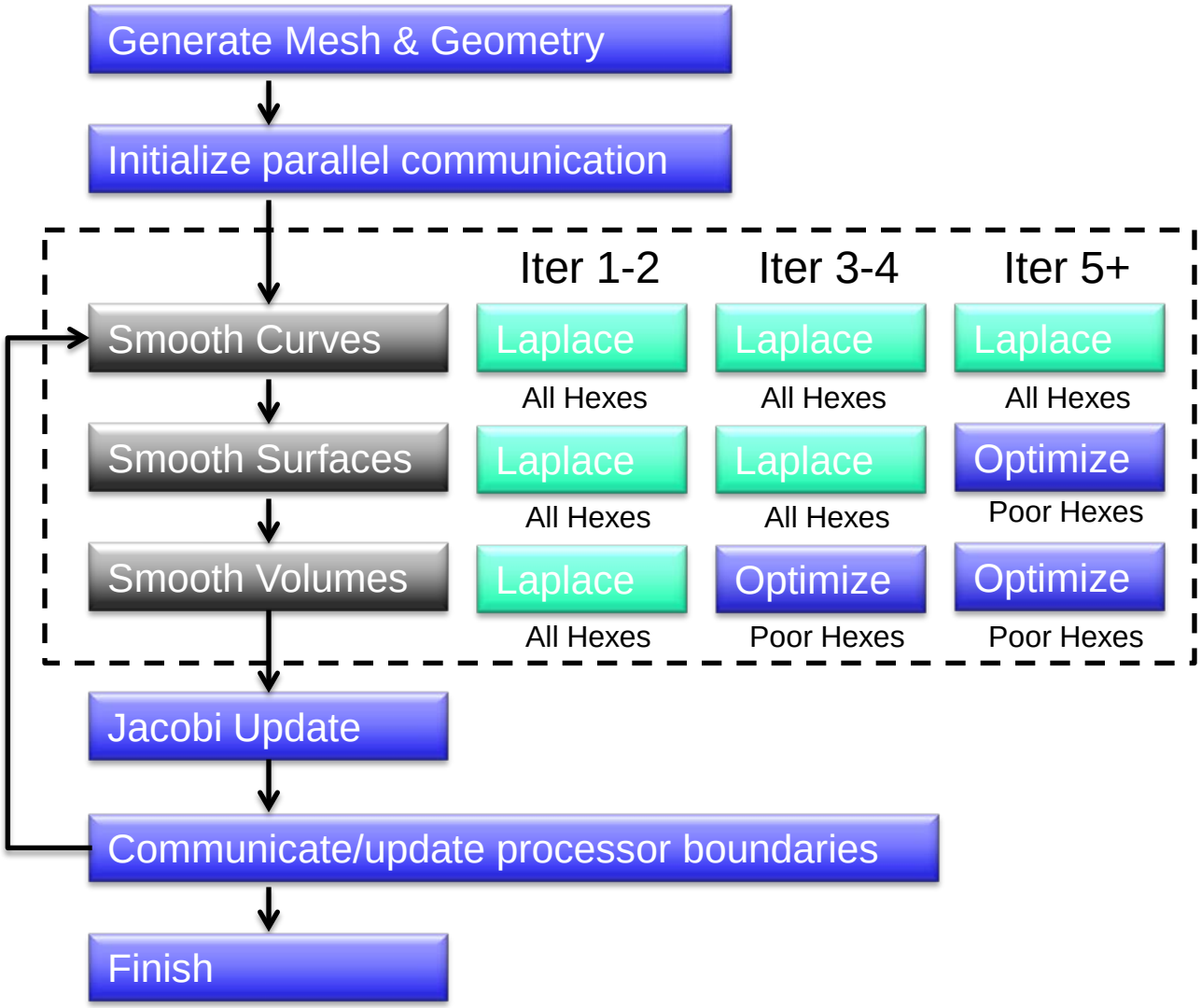
4. Compute 3D Length-weighted Laplacian smoothing on node P_k to get P_k' in global coordinate system.
5. Transform P_k' to local coordinate system:

$$(P_k)_{local} = \begin{Bmatrix} x_k \\ y_k \\ z_k \end{Bmatrix} = \begin{Bmatrix} \left(P_k' - Pk\right) \cdot T_1 \\ \left(P_k' - Pk\right) \cdot T_2 \\ a_{k2}x_k + a_{k3}y_k + a_{k4}x_k^2 + a_{k5}x_k y_k + a_{k6}y_k^2 \end{Bmatrix}$$

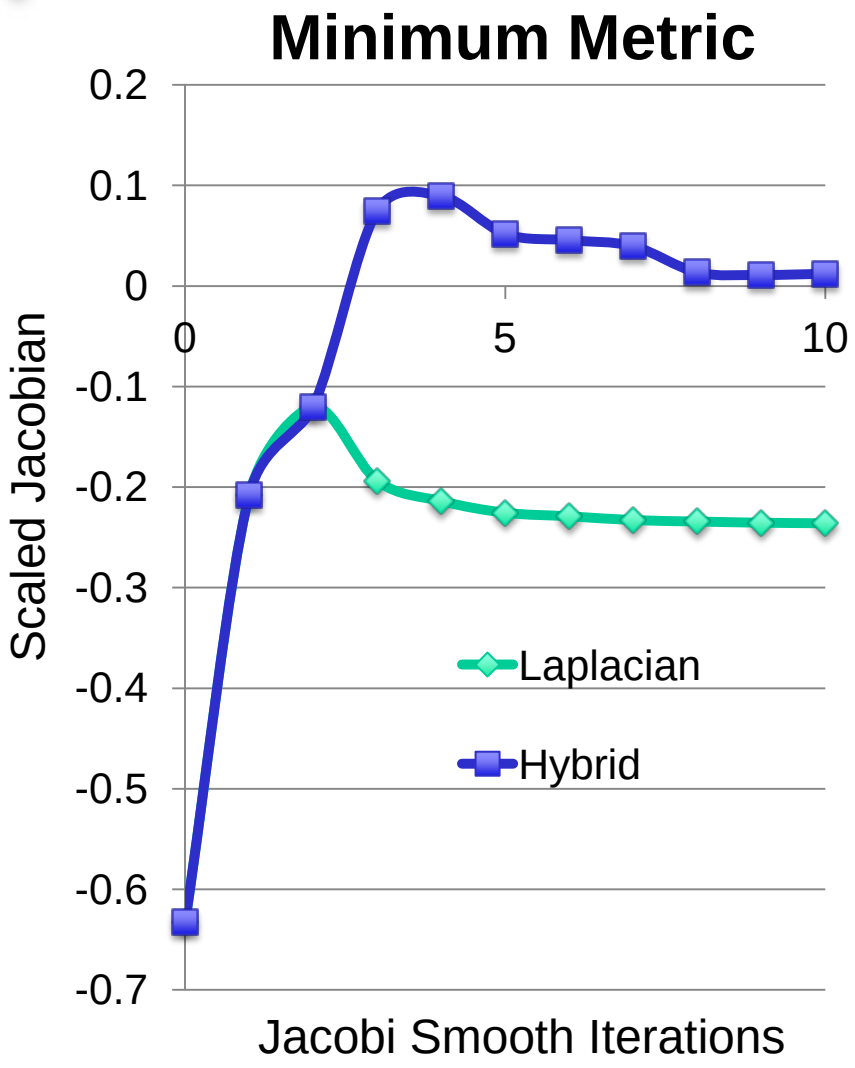
6. Compute new location for P_k on quadratic patch as:

$$(P_k)_{new} = P_k + (P_k)_{local}^T \begin{Bmatrix} T_1 \\ T_2 \\ N_k \end{Bmatrix}$$

Parallel Smoothing Procedure



Jacobi Smoothing



Laplacian Smoothing:

- Very fast, quickly improves poor quality elements
- Minimum quality can degrade with too many iterations

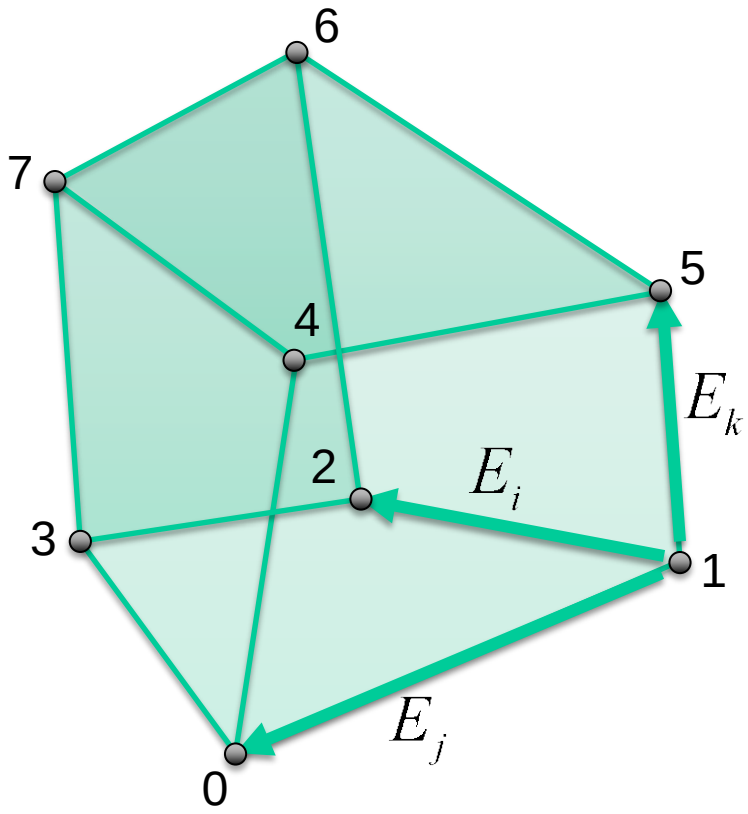
Opitimzation-based Smoothing:

- Slower, but more reliable
- Uninverts and optimizes mesh quality

Hybrid Smoothing:

- Laplacian – 2 iterations
- Optimization– 2+ iterations only on local patches of poor quality

Scaled Jacobian



$$(J_s)_I = det\left\{\hat{E}_i \hat{E}_j \hat{E}_k\right\}^{\top}$$

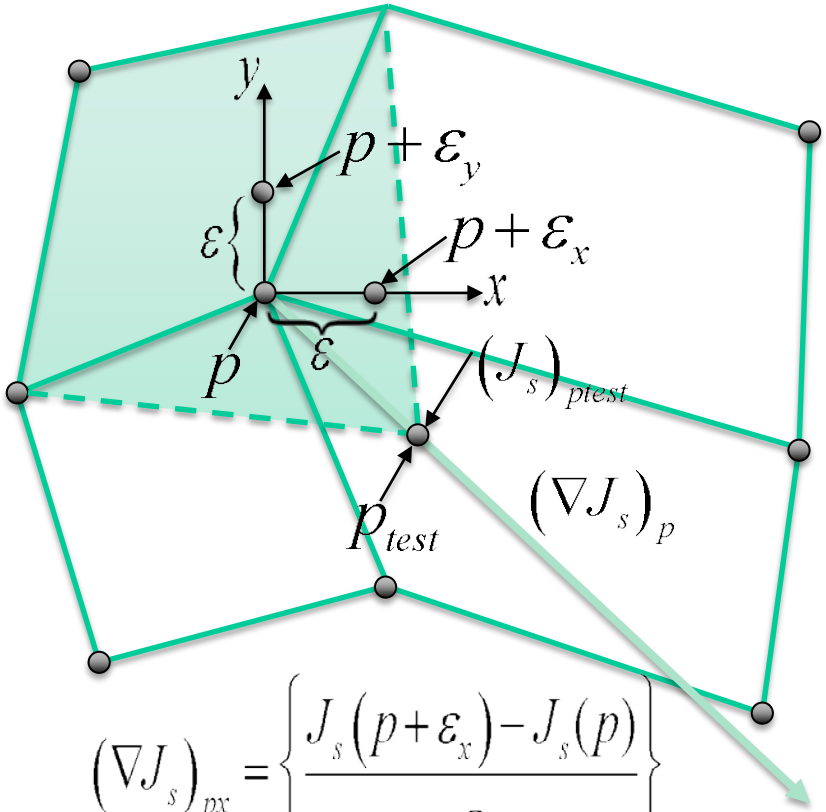
$$J_s = min\left((J_s)_I, I = 0, 1, ...7\right)$$

"Acceptable"

$$J_s \geq 0.2$$

Optimization-based Smoothing

- Compute minimum scaled Jacobian, $(J_s)_p$, of node p in all attached hexes
- Compute numerical gradient $(\nabla J_s)_p$
- Find improved $(J_s)_p$ by searching along vector $(\nabla J_s)_p$



$$(\nabla J_s)_{px} = \left\{ \frac{J_s(p+\varepsilon_x) - J_s(p)}{\varepsilon} \right\}$$

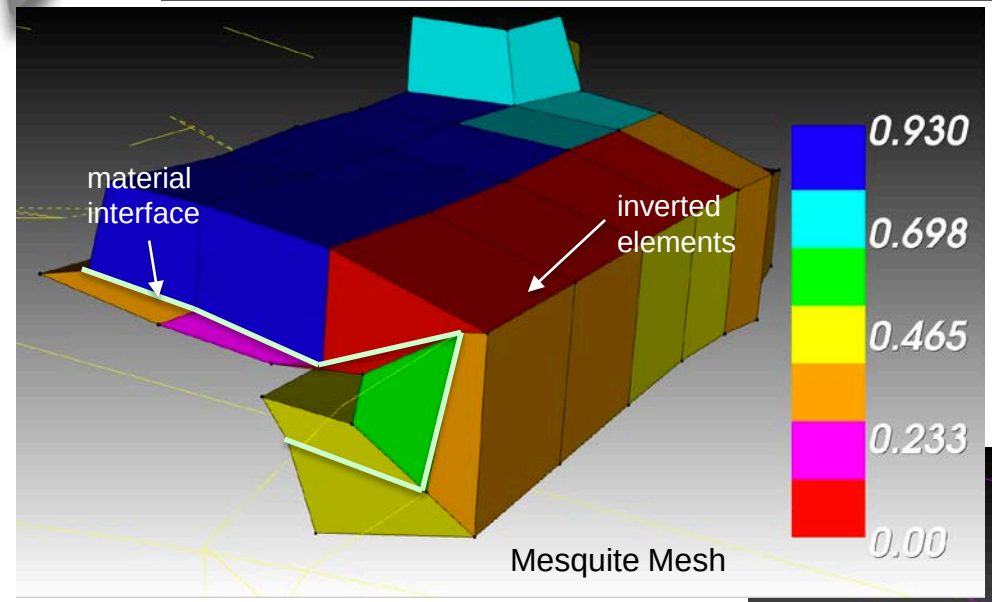
$$(\nabla J_s)_{py} = \left\{ \frac{J_s(p+\varepsilon_y) - J_s(p)}{\varepsilon} \right\}$$

Stopping Criteria

$(J_s)_p > 0.2$

Maximum 3 iterations

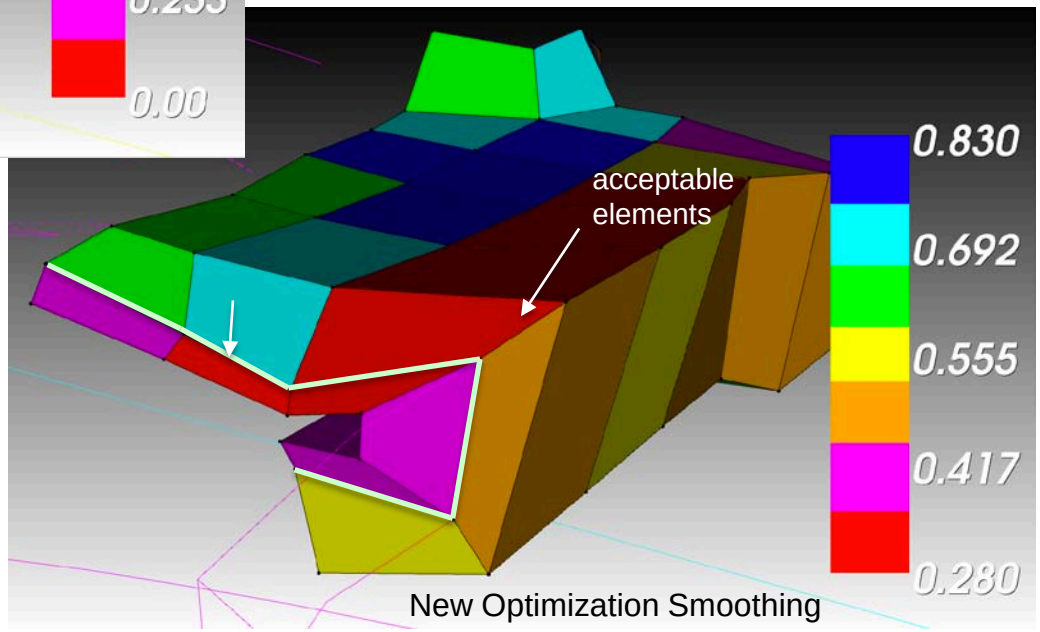
Optimization-based Smoothing



Mesquite Mesh

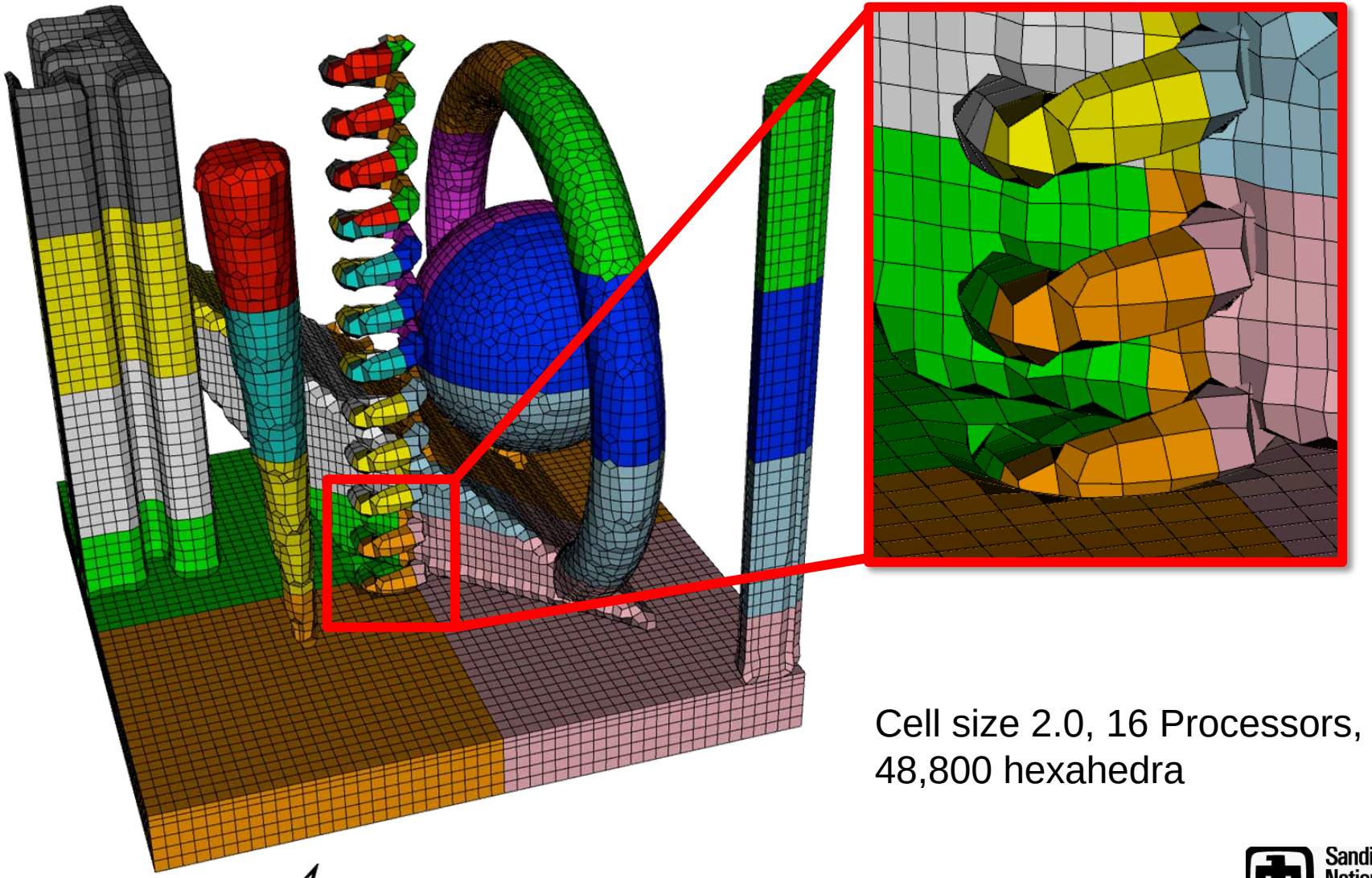
Colors represent min scaled Jacobian at hex corners

Comparison of hex elements near a concave/convex material interface smoothed with Mesquite and New Optimization Method



New Optimization Smoothing

Examples

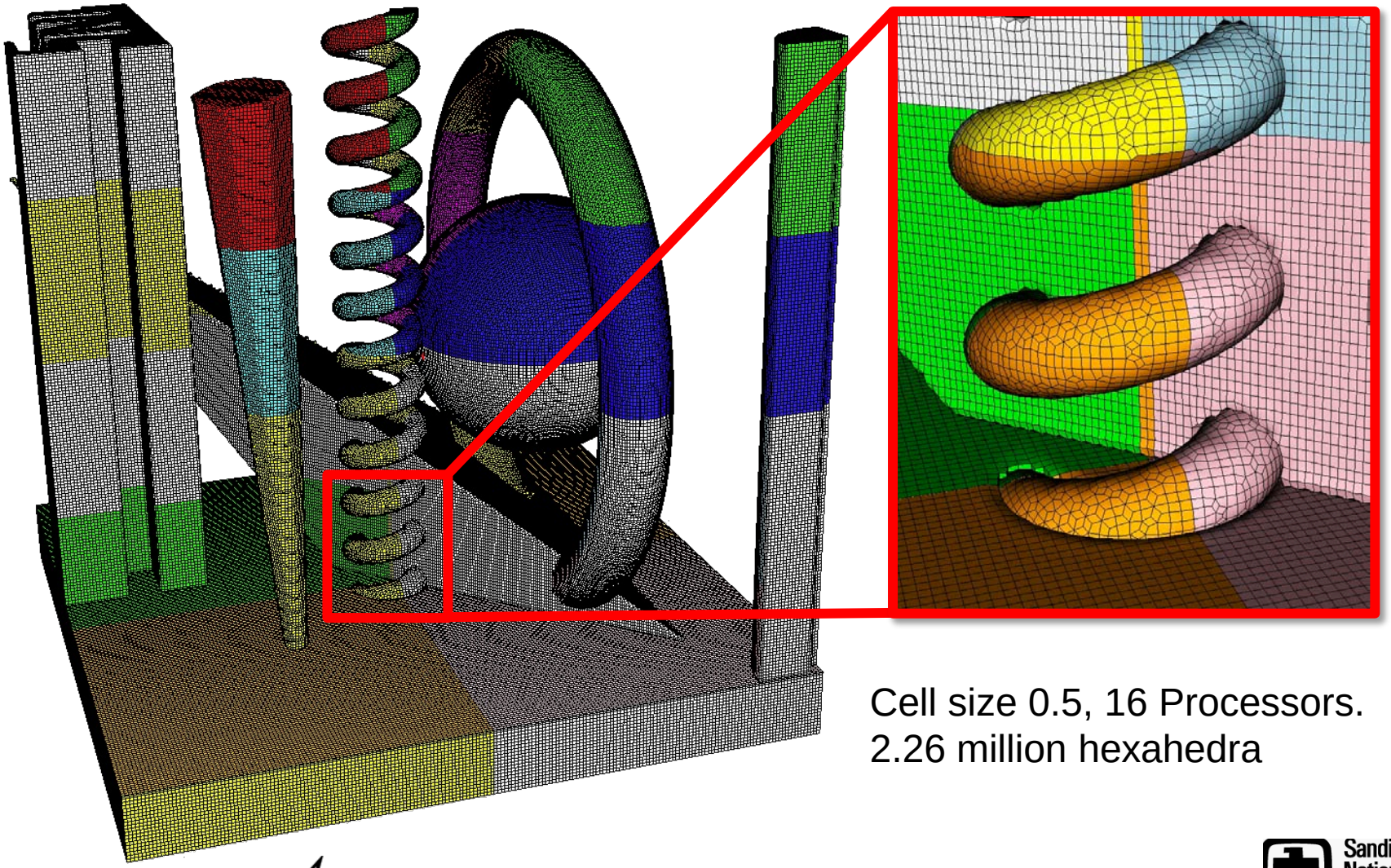


Cell size 2.0, 16 Processors,
48,800 hexahedra

LOCKHEED MARTIN

Sandia National Laboratories

Examples

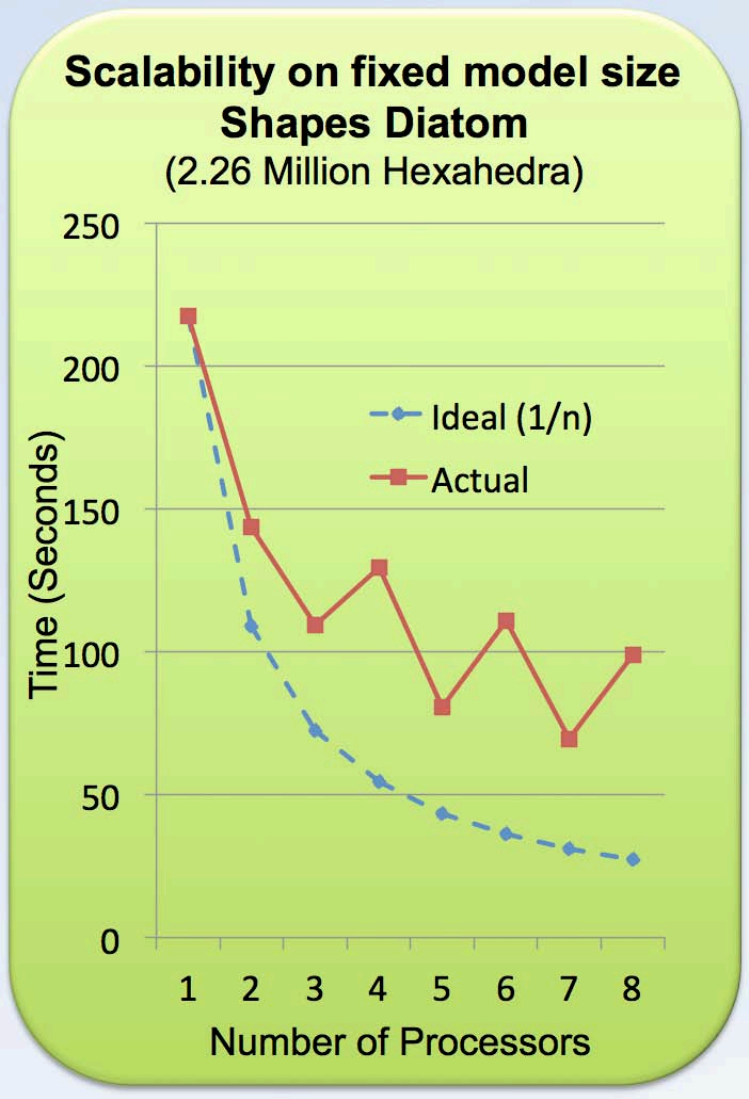


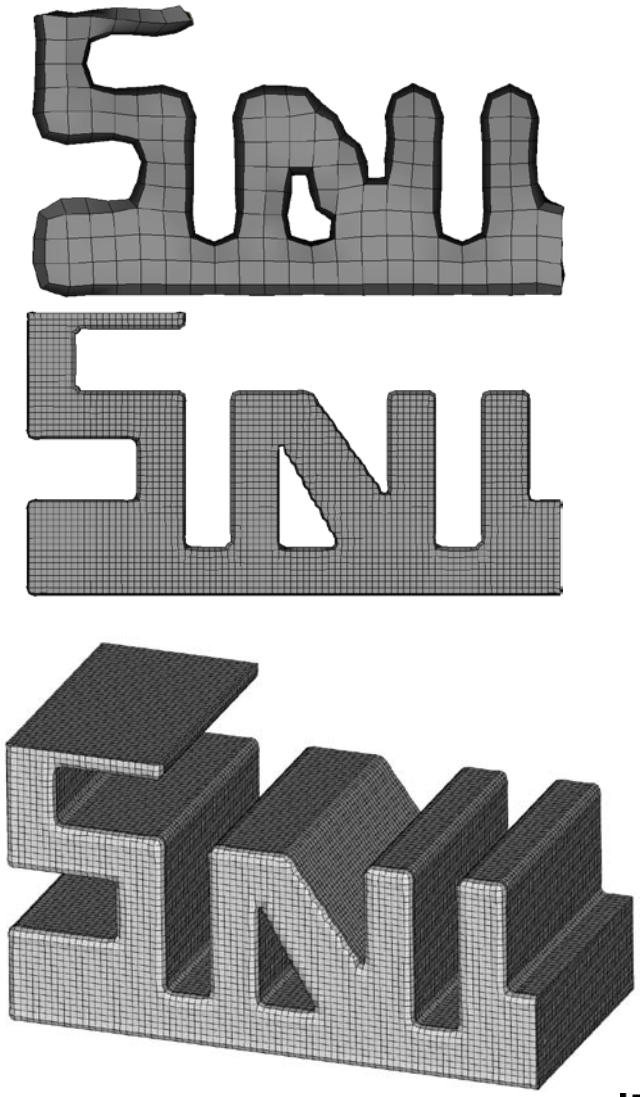
Cell size 0.5, 16 Processors.
2.26 million hexahedra

LOCKHEED MARTIN

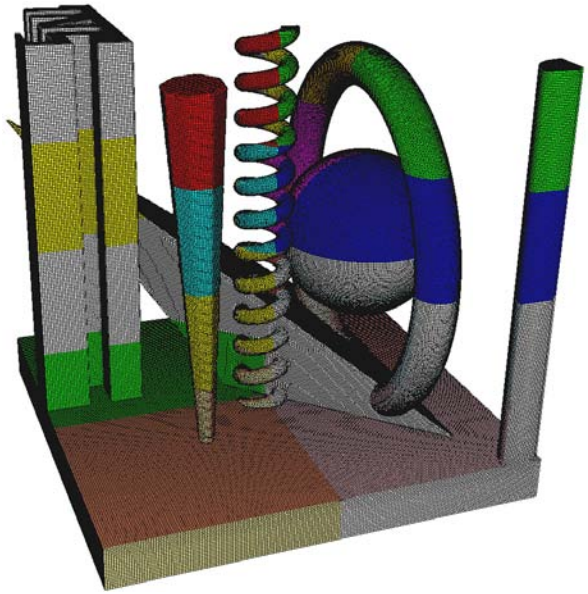
Sandia National Laboratories

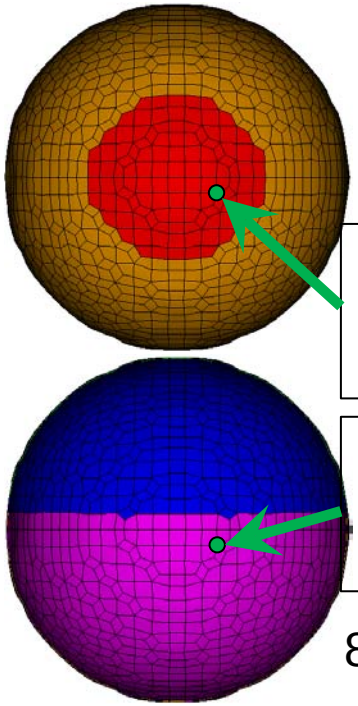
Examples


Scalability on fixed model size
Shapes Diatom
(2.26 Million Hexahedra)
250
200
150
100
50
0
Time (Seconds)
Ideal (1/n)
Actual
1
2
3
4
5
6
7
8
Number of Processors



Serial-Parallel Consistent Meshes





Serial-Parallel Consistency

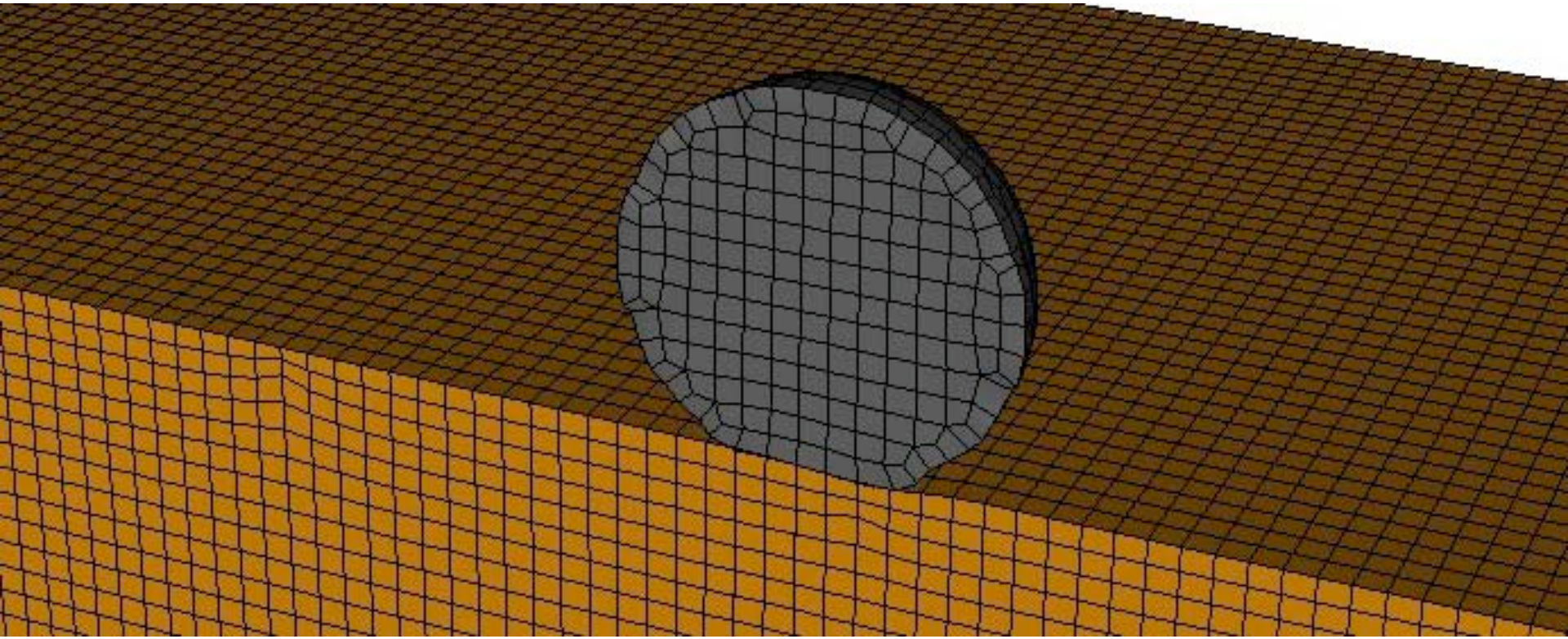
3 Processors

Node 14869 –
X-coord: 29.7442645017447
Y-coord: -1.86475197560341
Z-coord: 69.4254564637732

Node 12751 –
X-coord: 29.7442645017447
Y-coord: -1.86475197560341
Z-coord: 69.4254564637731

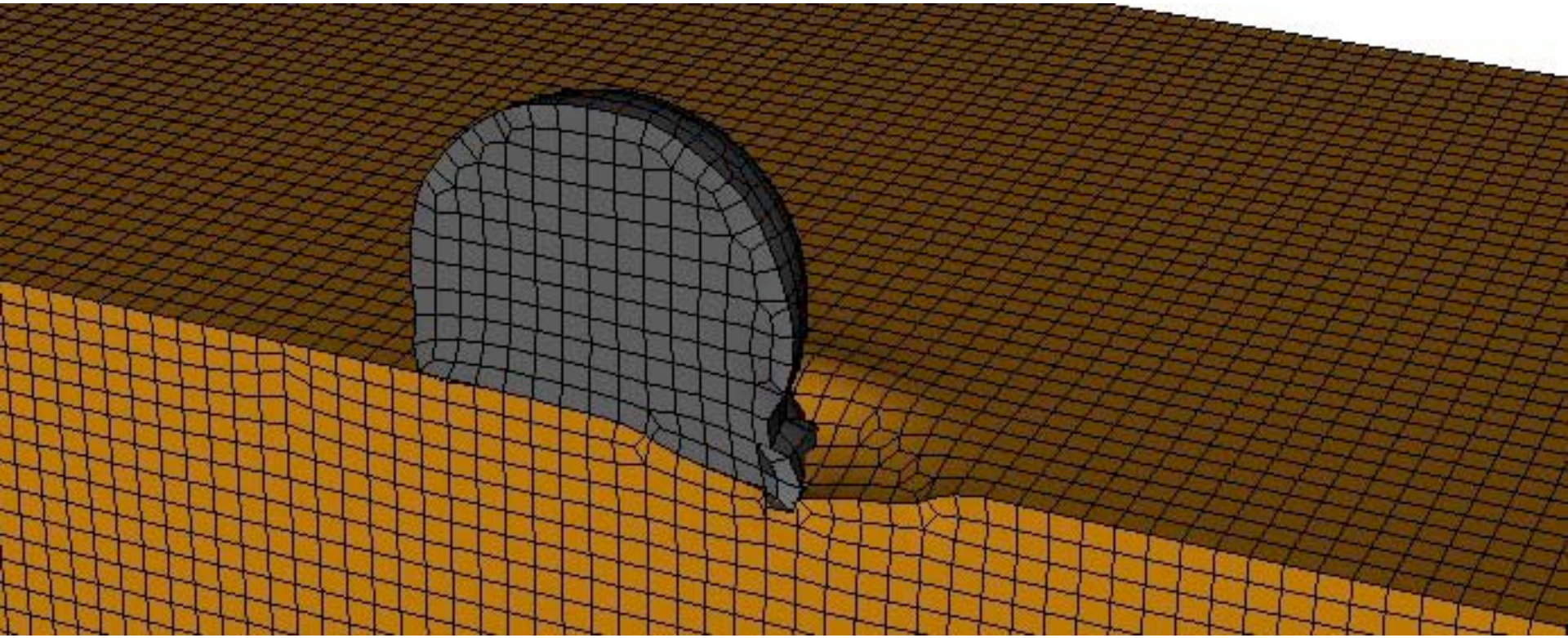
8 Processors

Exact same mesh regardless of domain decomposition



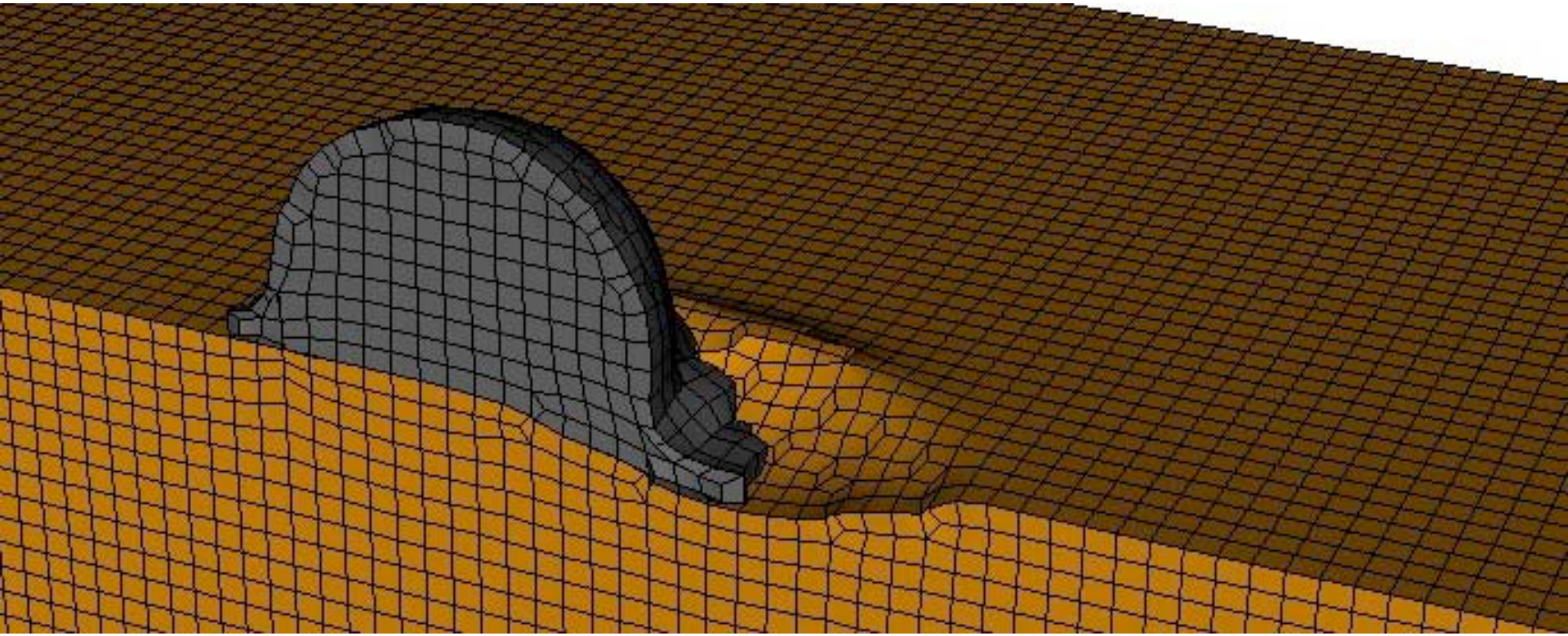
LOCKHEED MARTIN

Sandia National Laboratories



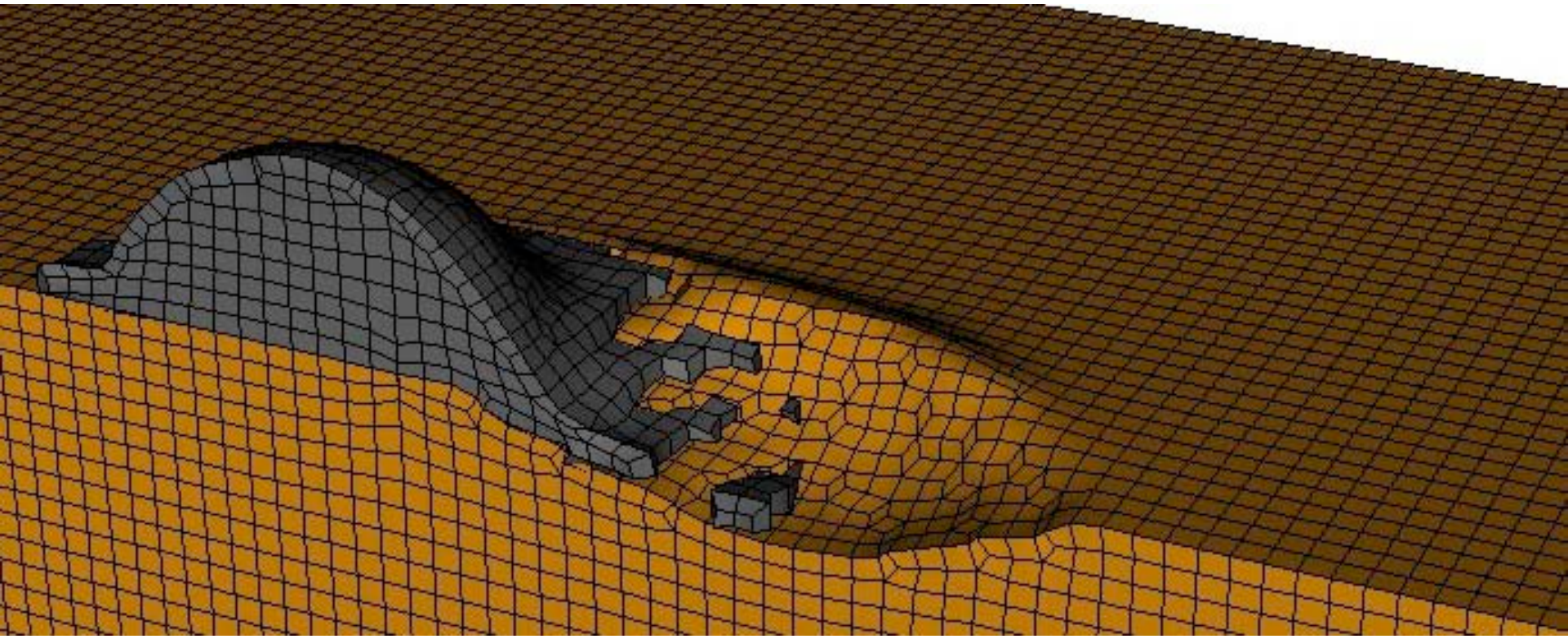
LOCKHEED MARTIN

Sandia National Laboratories



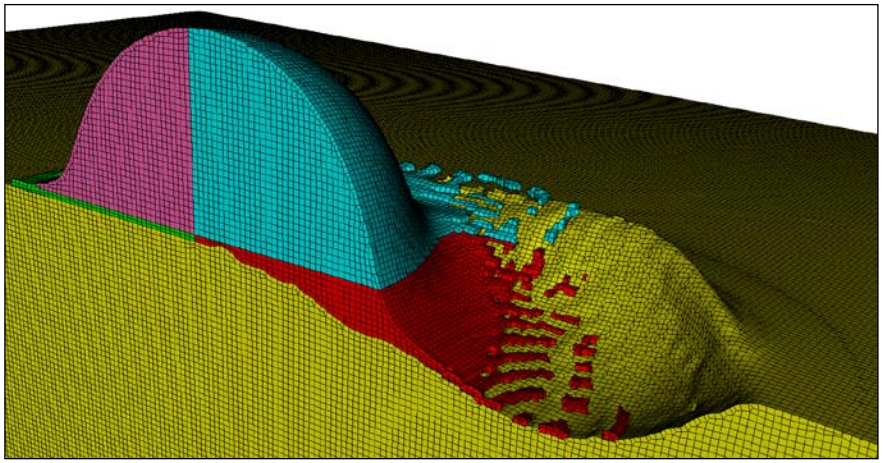
LOCKHEED MARTIN

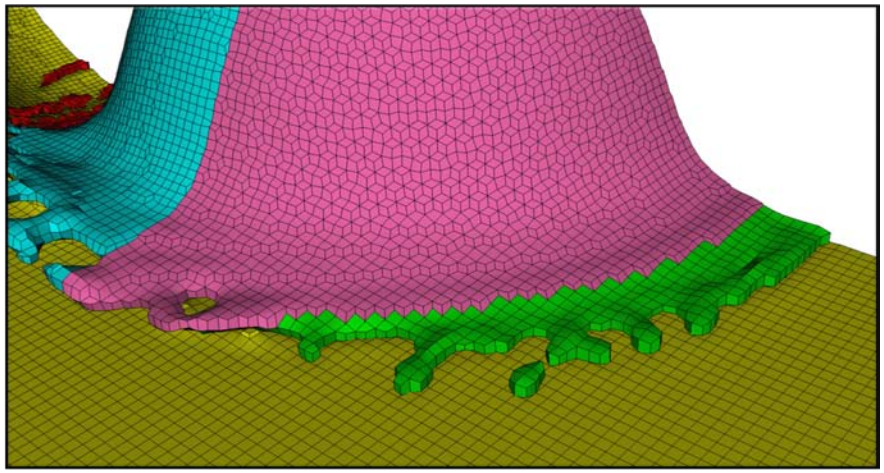
Sandia National Laboratories

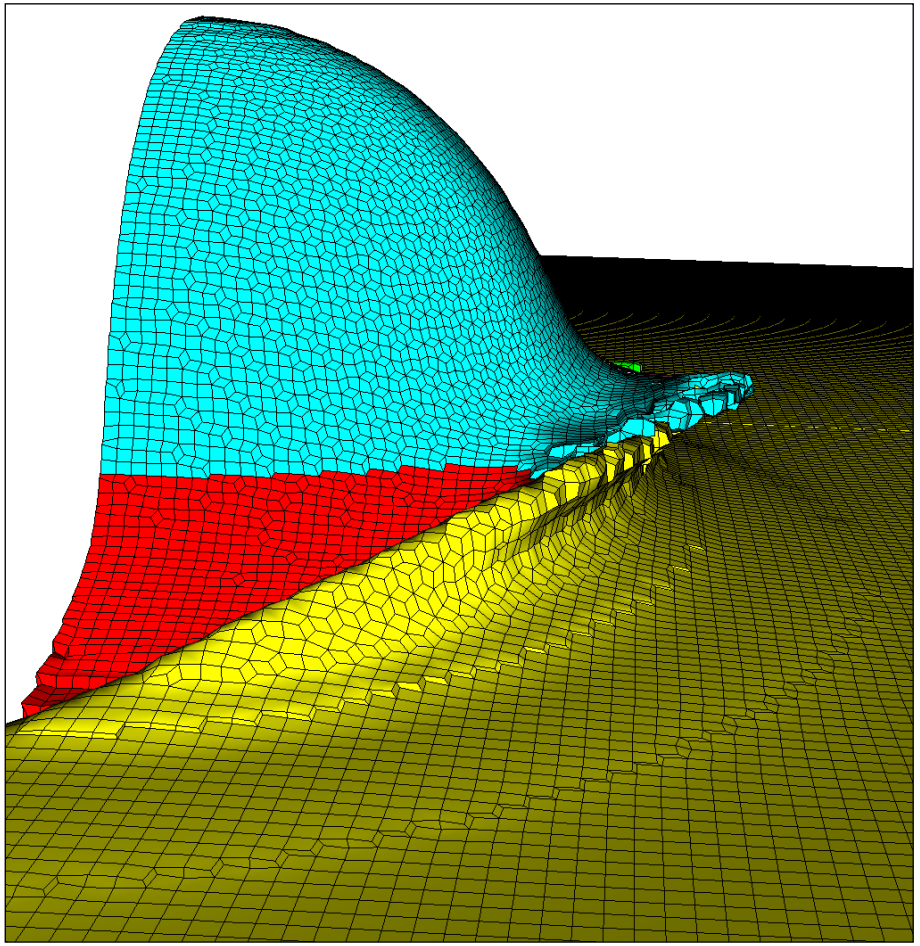


LOCKHEED MARTIN

Sandia National Laboratories



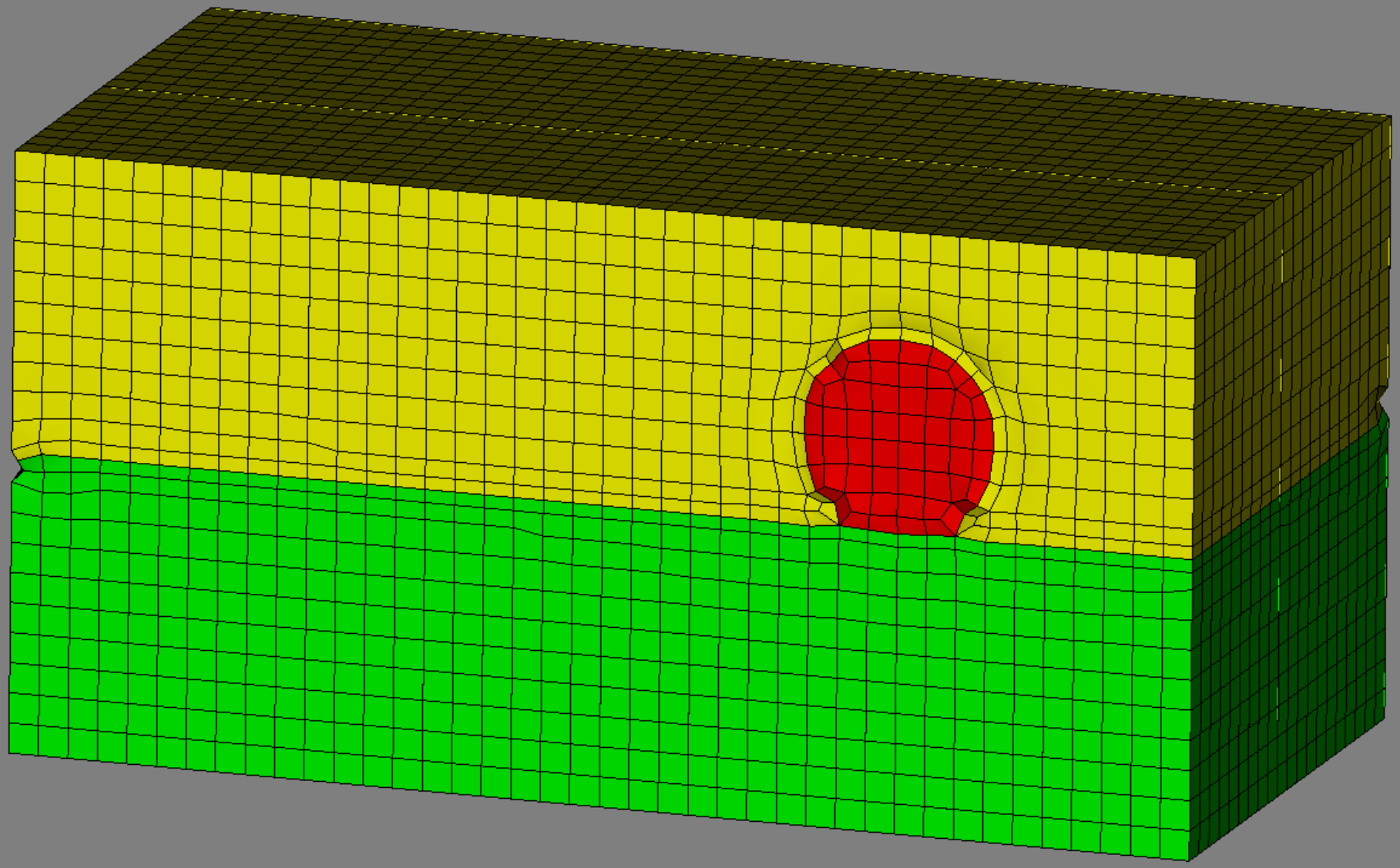




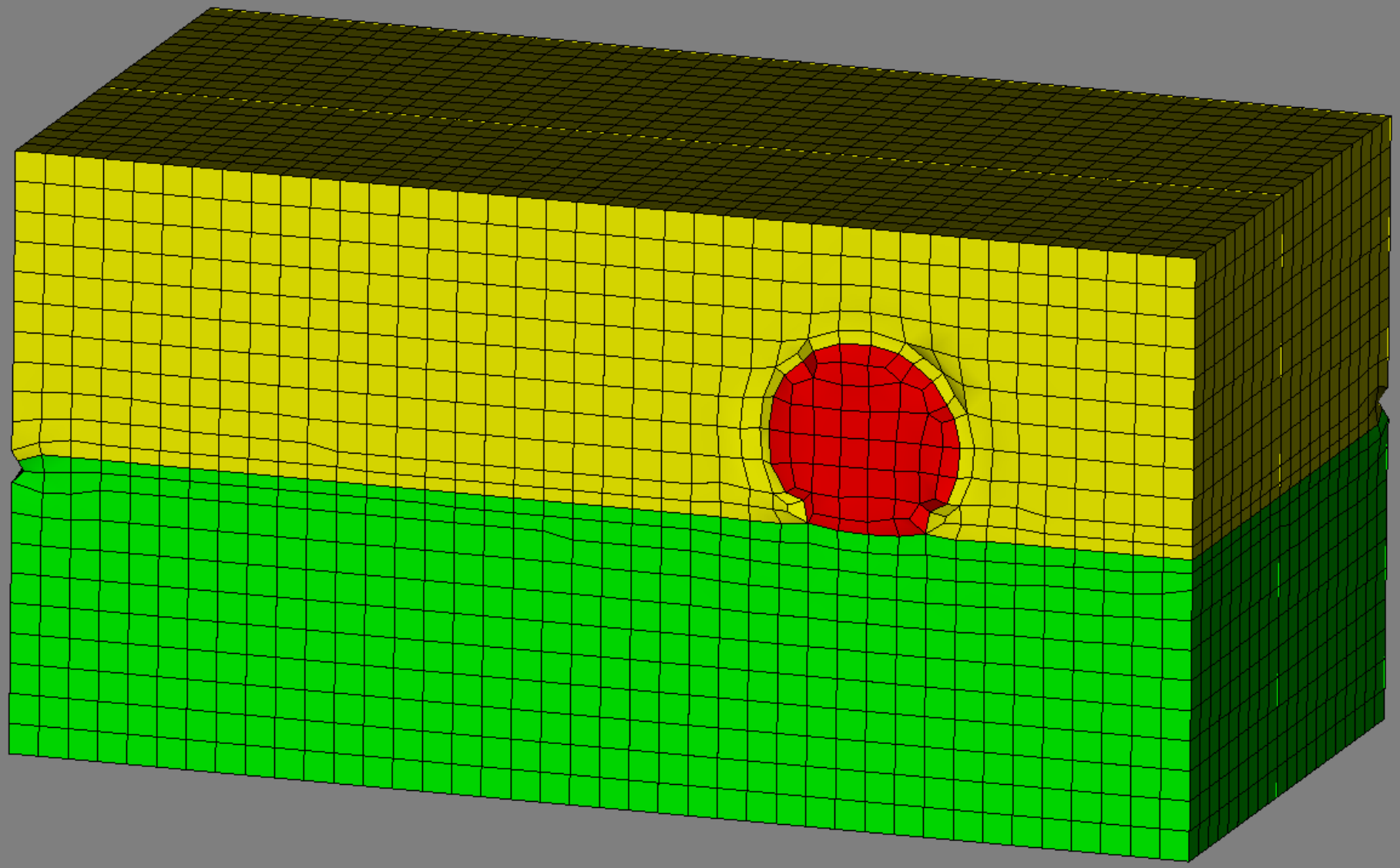
LOCKHEED MARTIN

Sandia National Laboratories

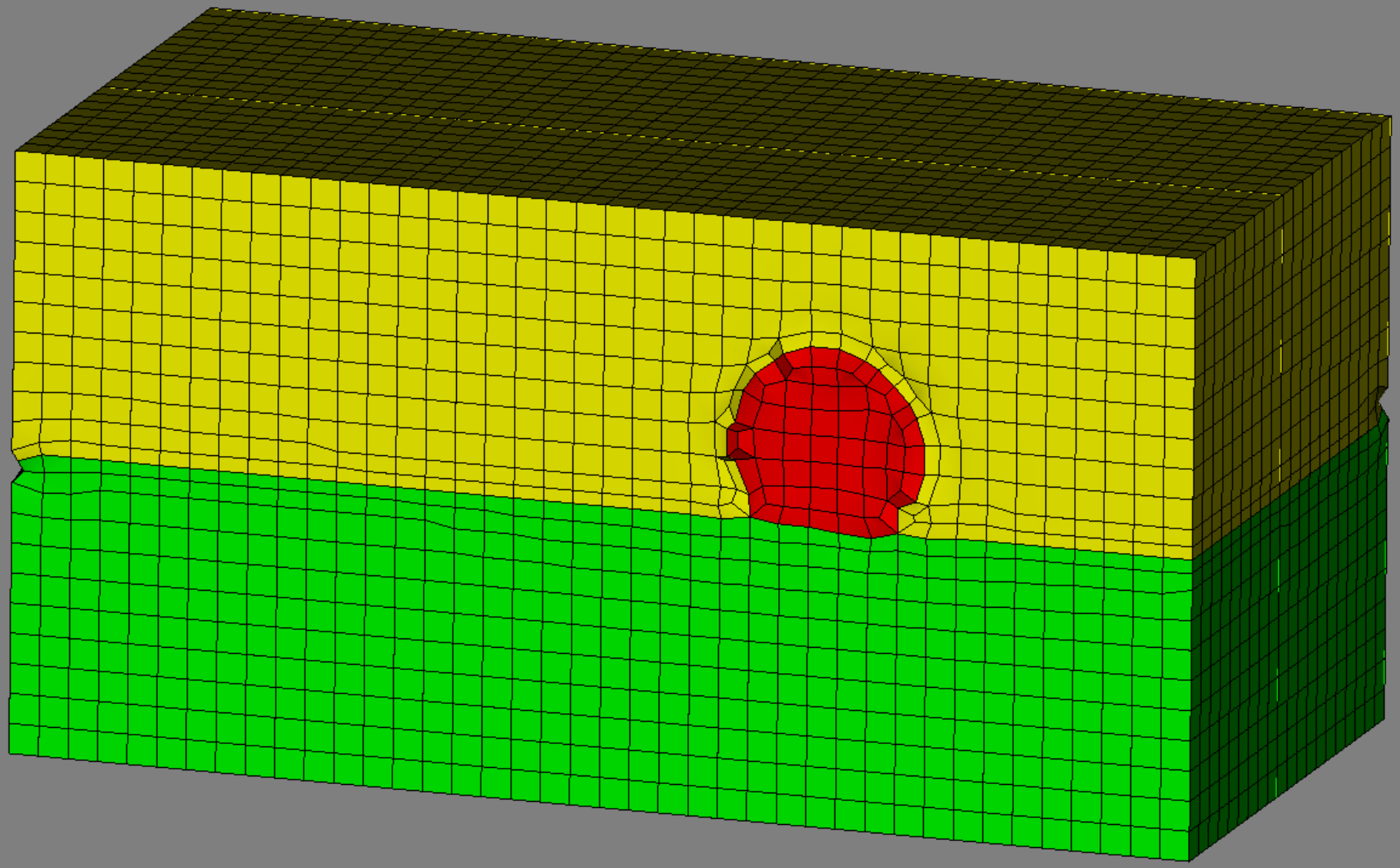
Conforming Mesh Between Multiple Materials and Void on 8 procesors



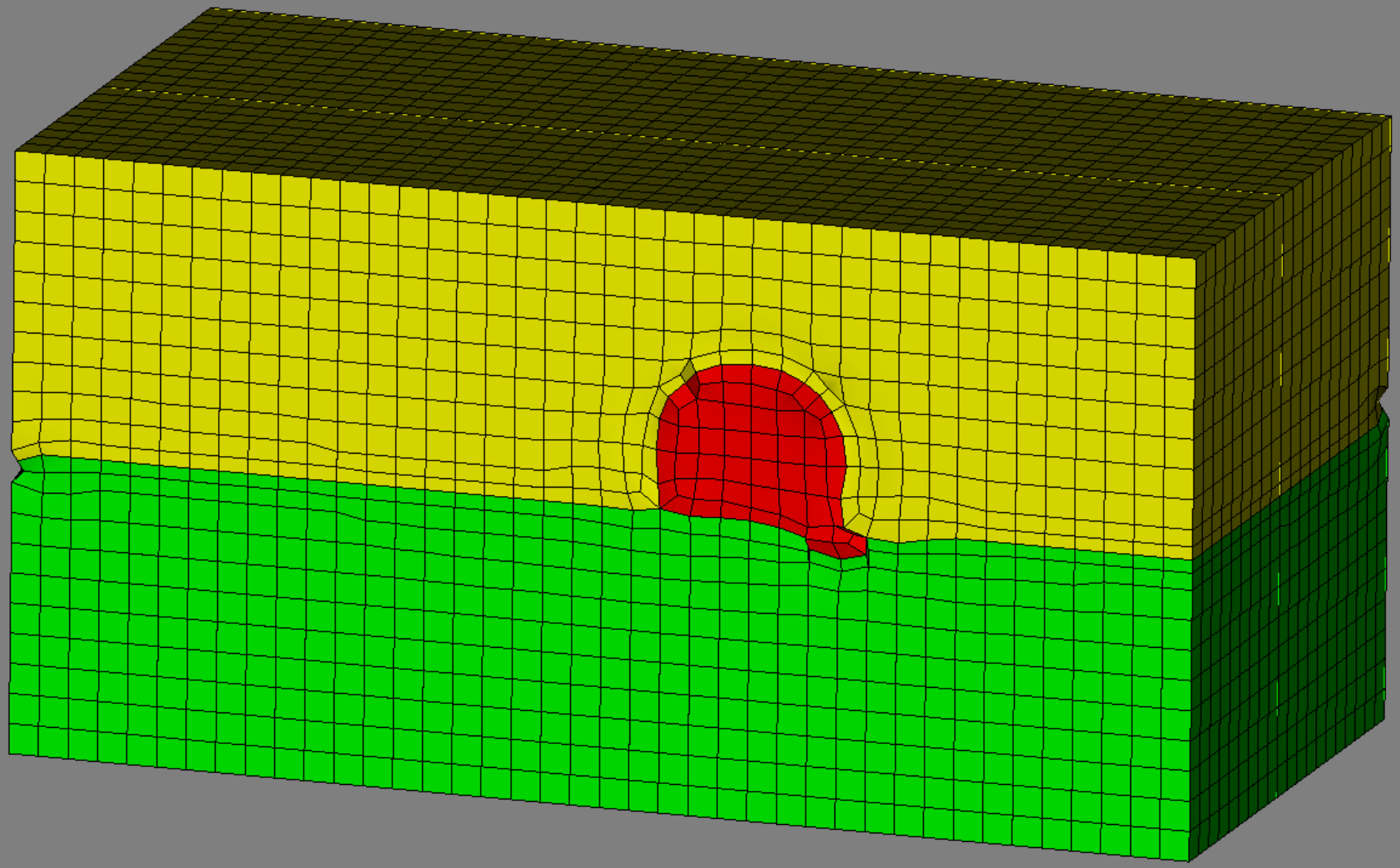
Conforming Mesh Between Multiple Materials and Void on 8 procesors



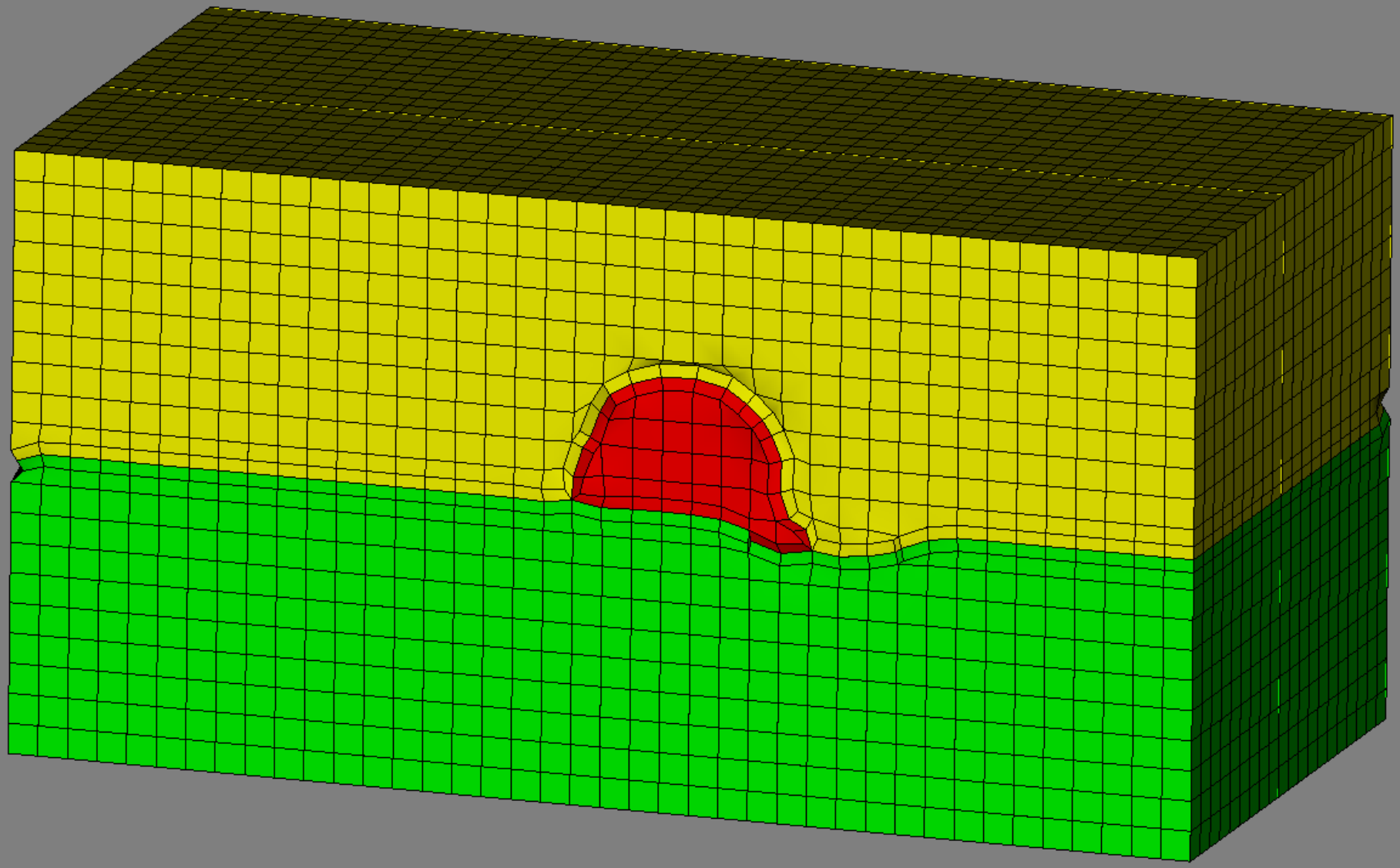
Conforming Mesh Between Multiple Materials and Void on 8 procesors



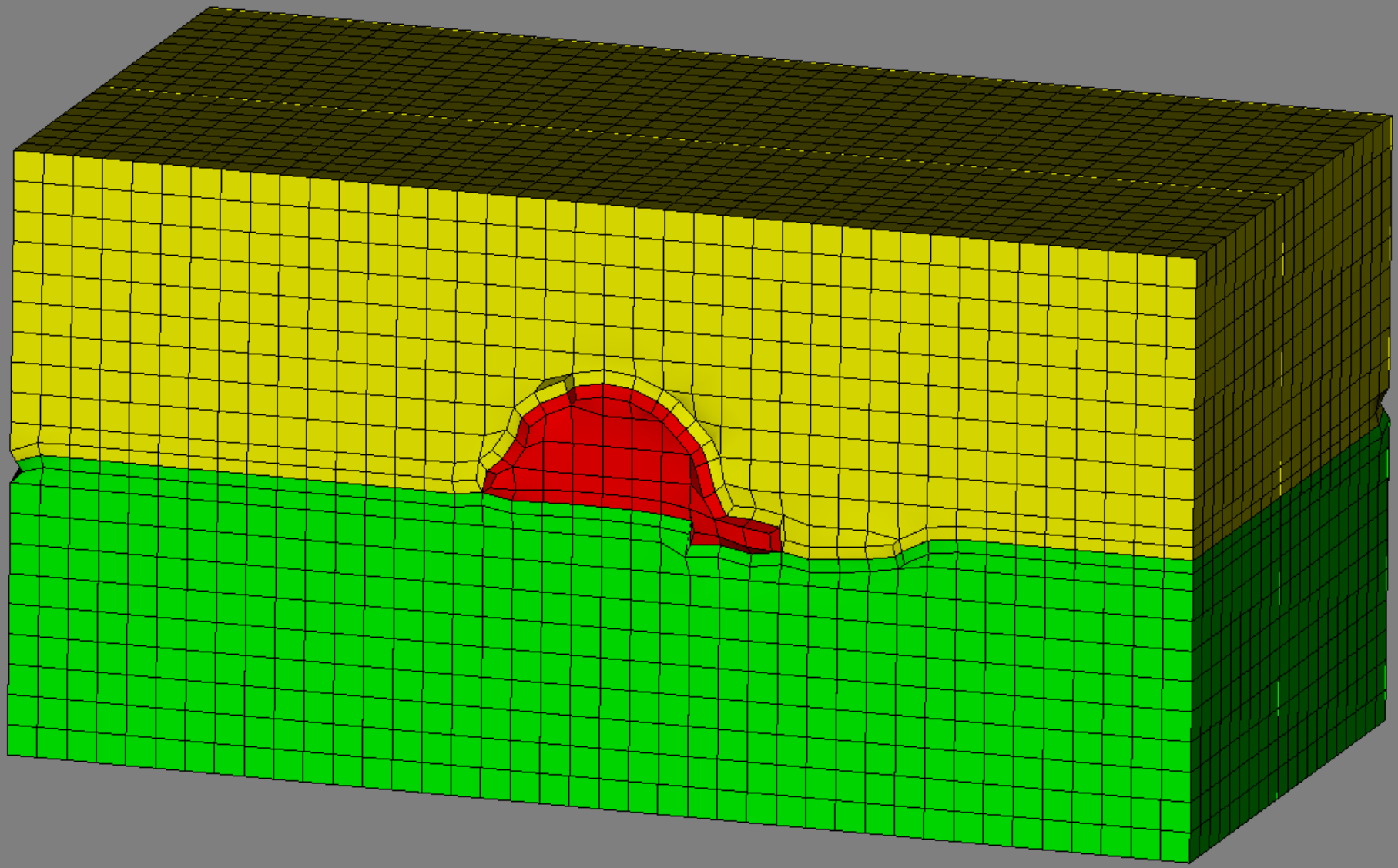
Conforming Mesh Between Multiple Materials and Void on 8 procesors



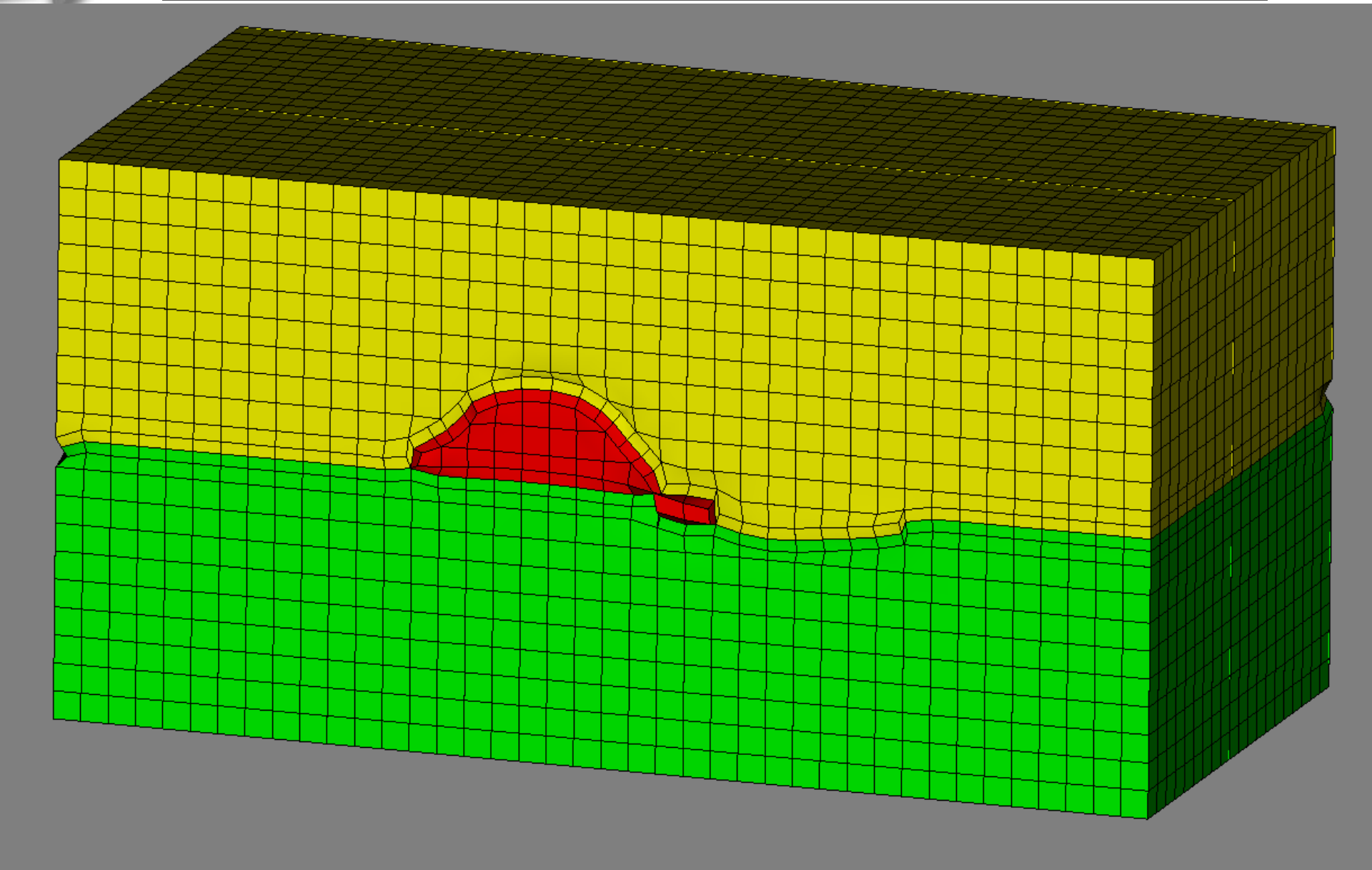
Conforming Mesh Between Multiple Materials and Void on 8 procesors



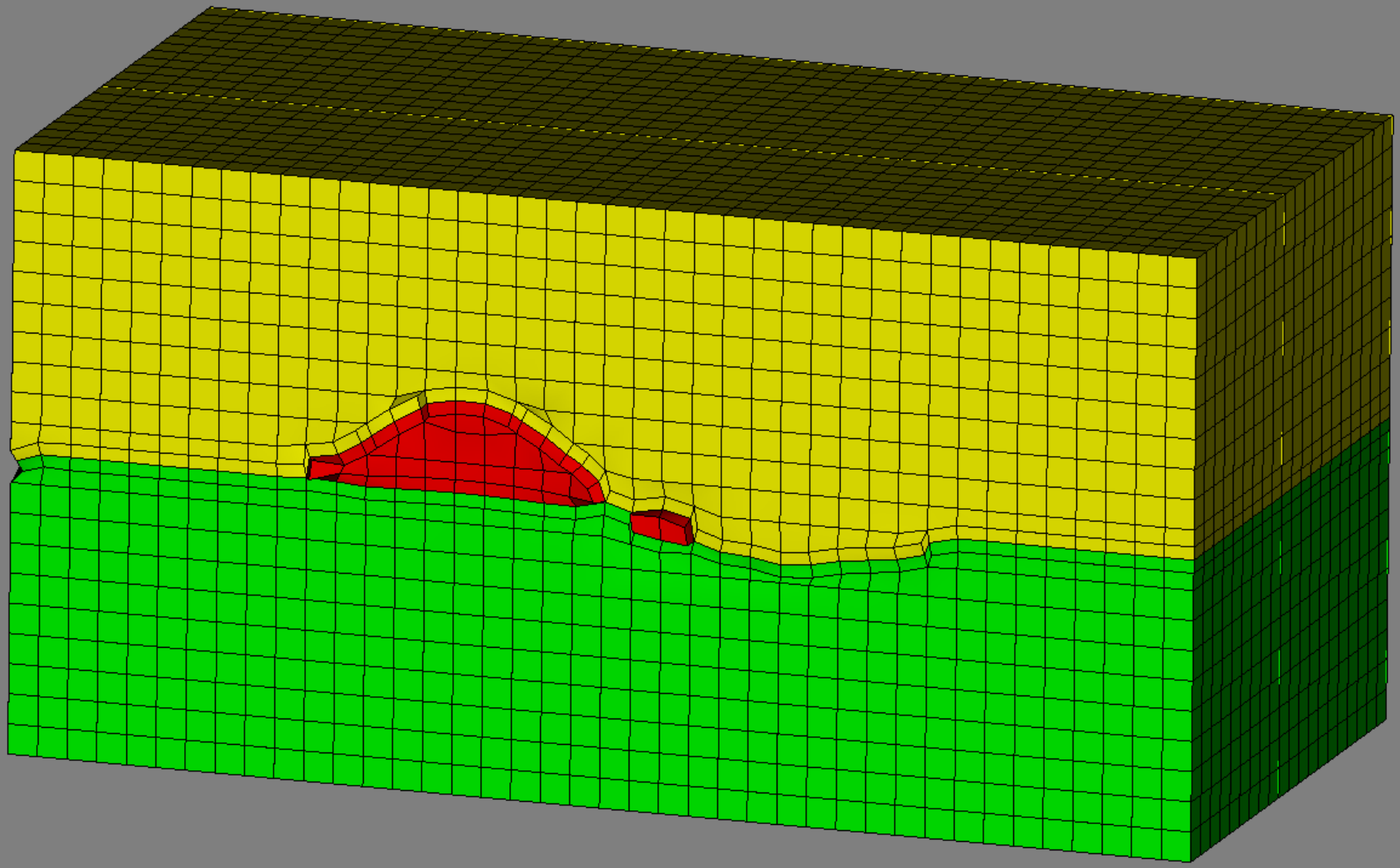
Conforming Mesh Between Multiple Materials and Void on 8 procesors



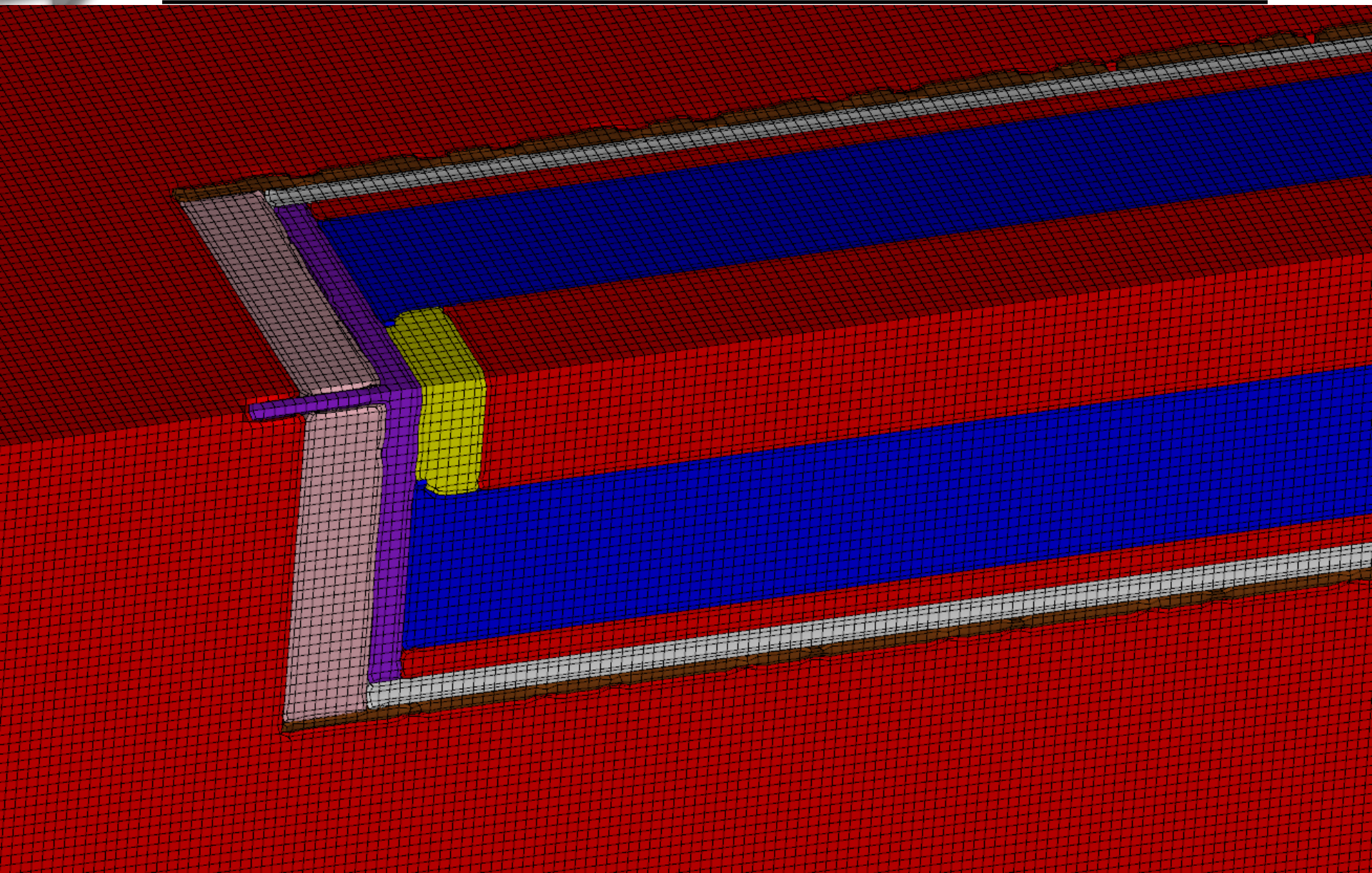
Conforming Mesh Between Multiple Materials and Void on 8 procesors



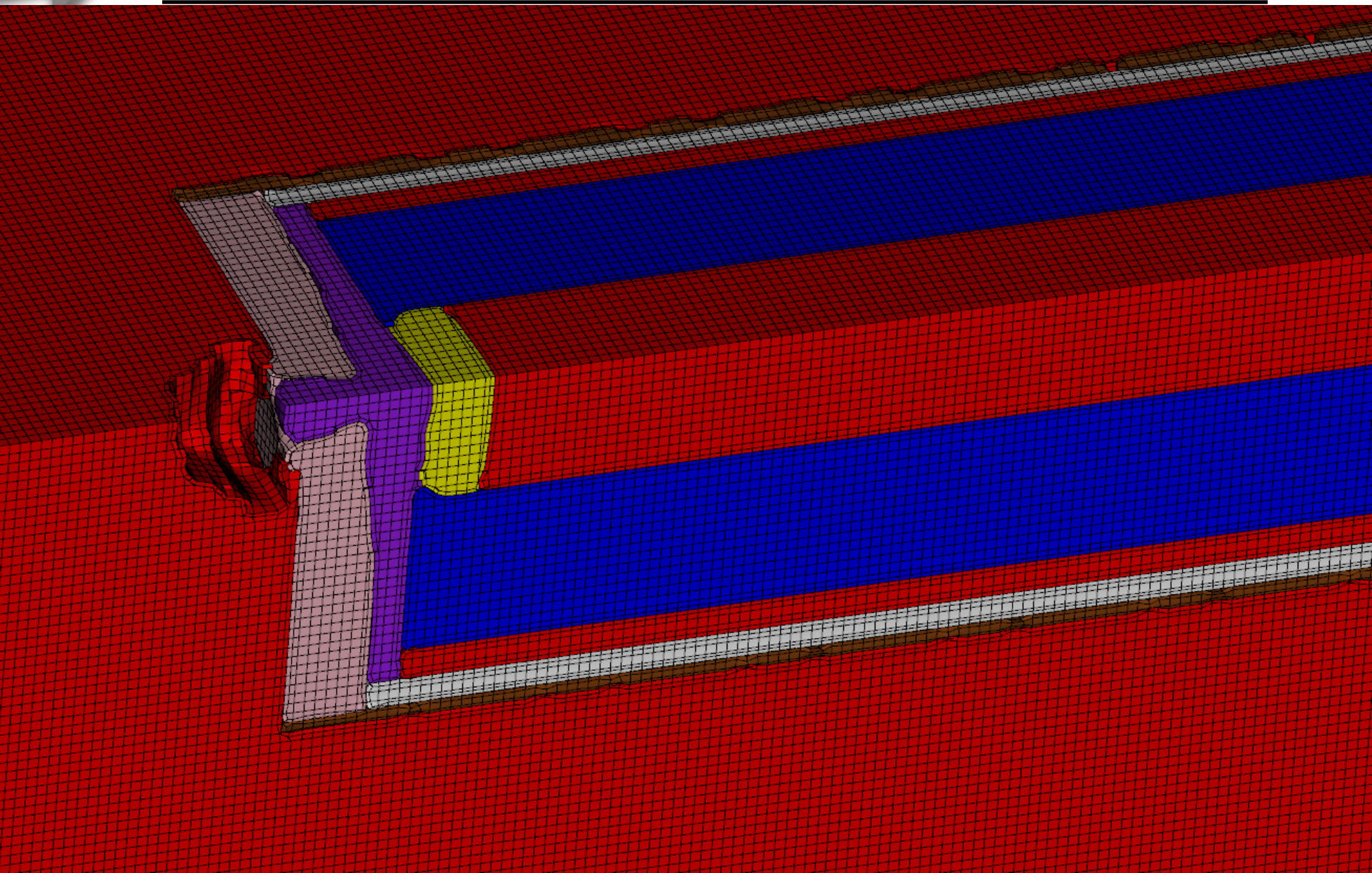
Conforming Mesh Between Multiple Materials and Void on 8 procesors



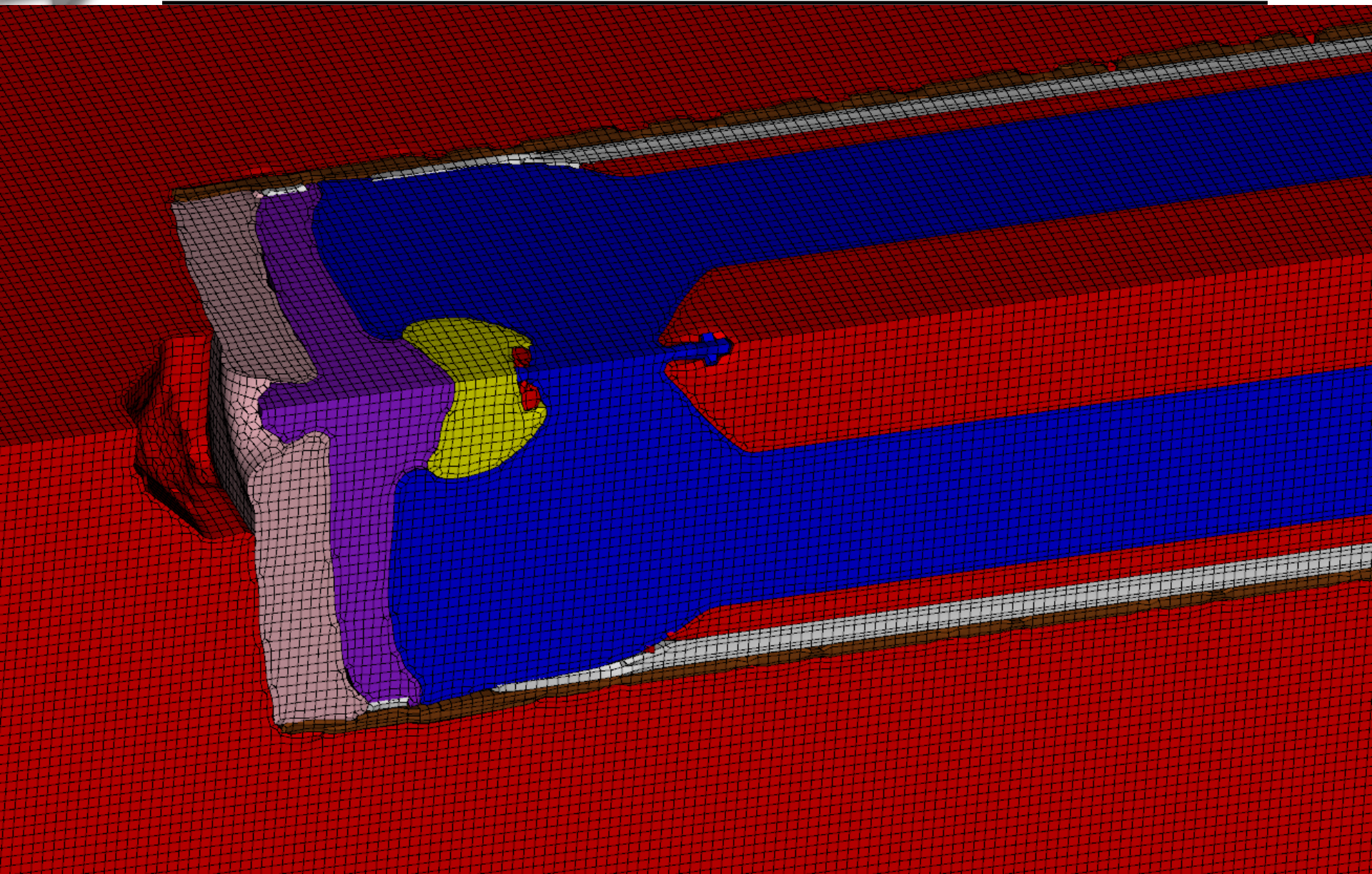
CTH Pipe Bomb Explosion



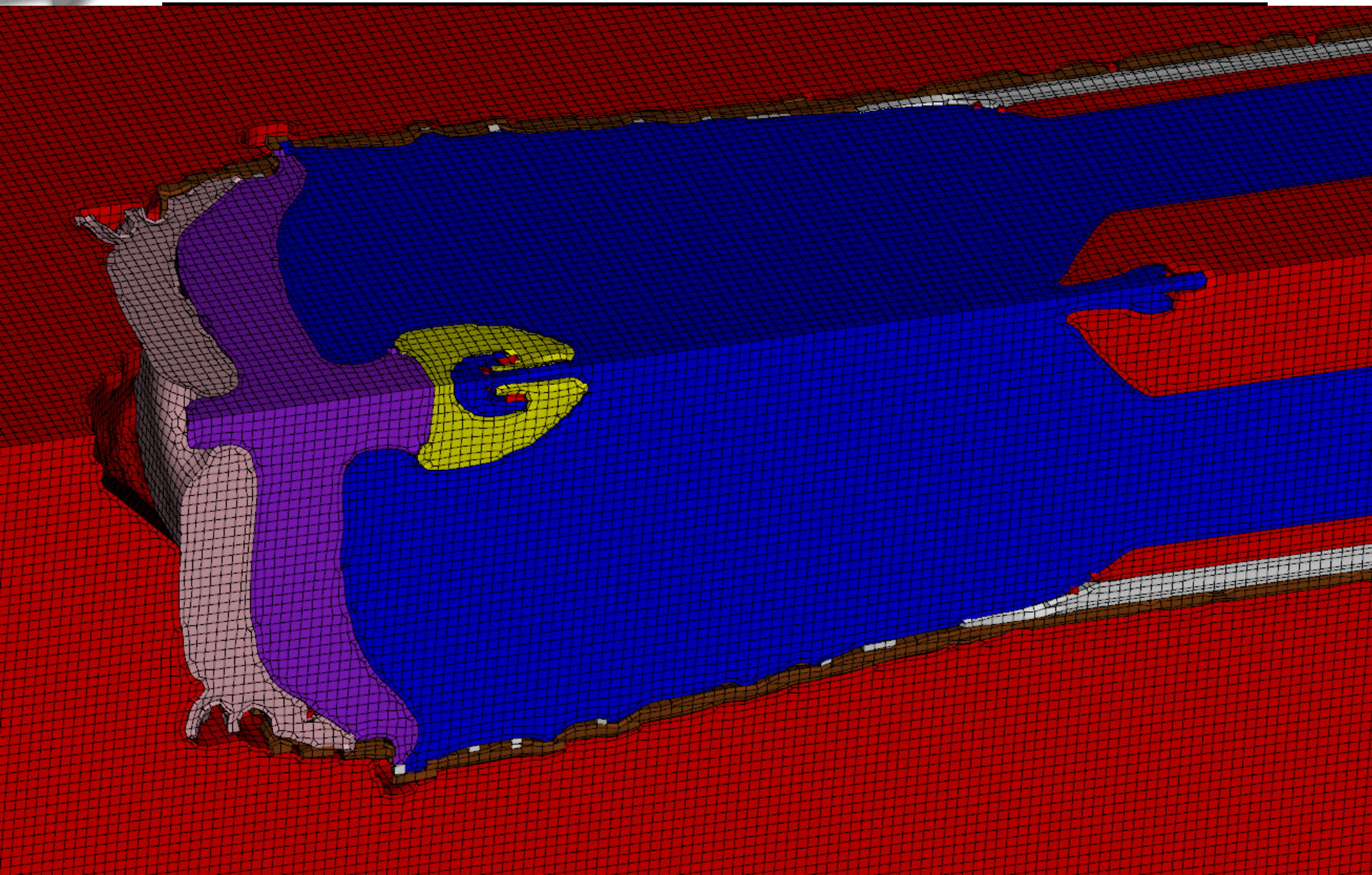
CTH Pipe Bomb Explosion



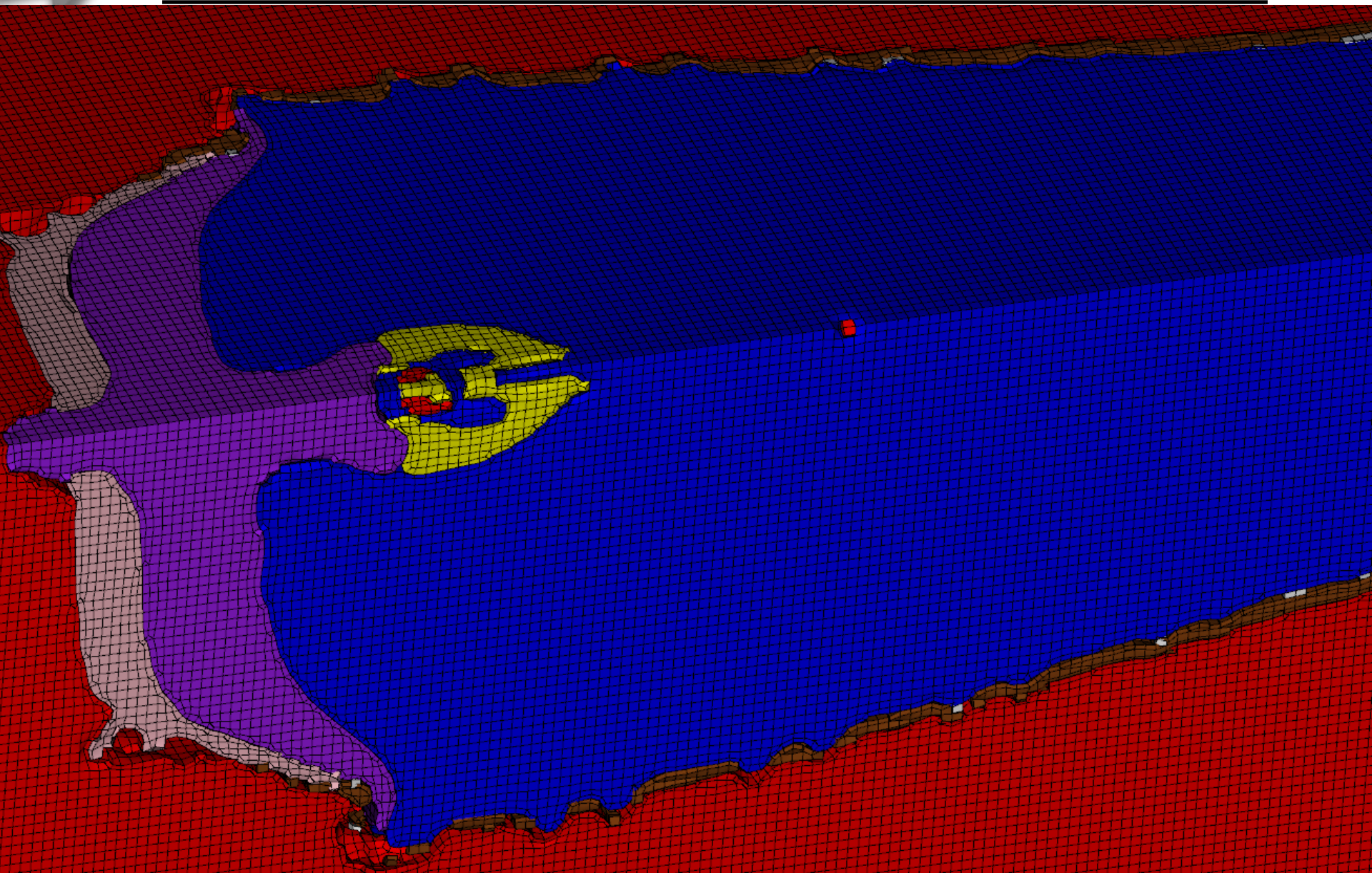
CTH Pipe Bomb Explosion

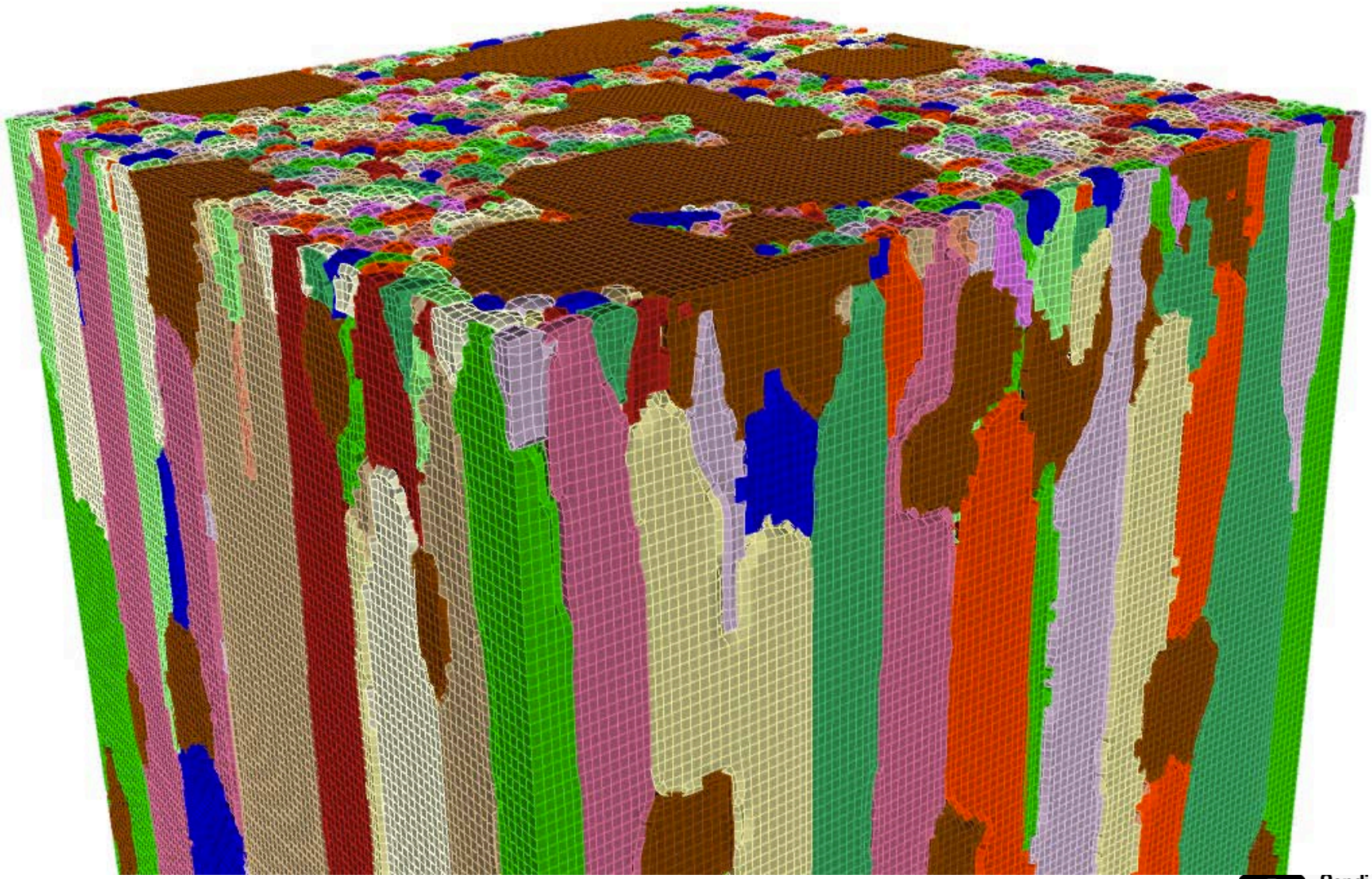


CTH Pipe Bomb Explosion



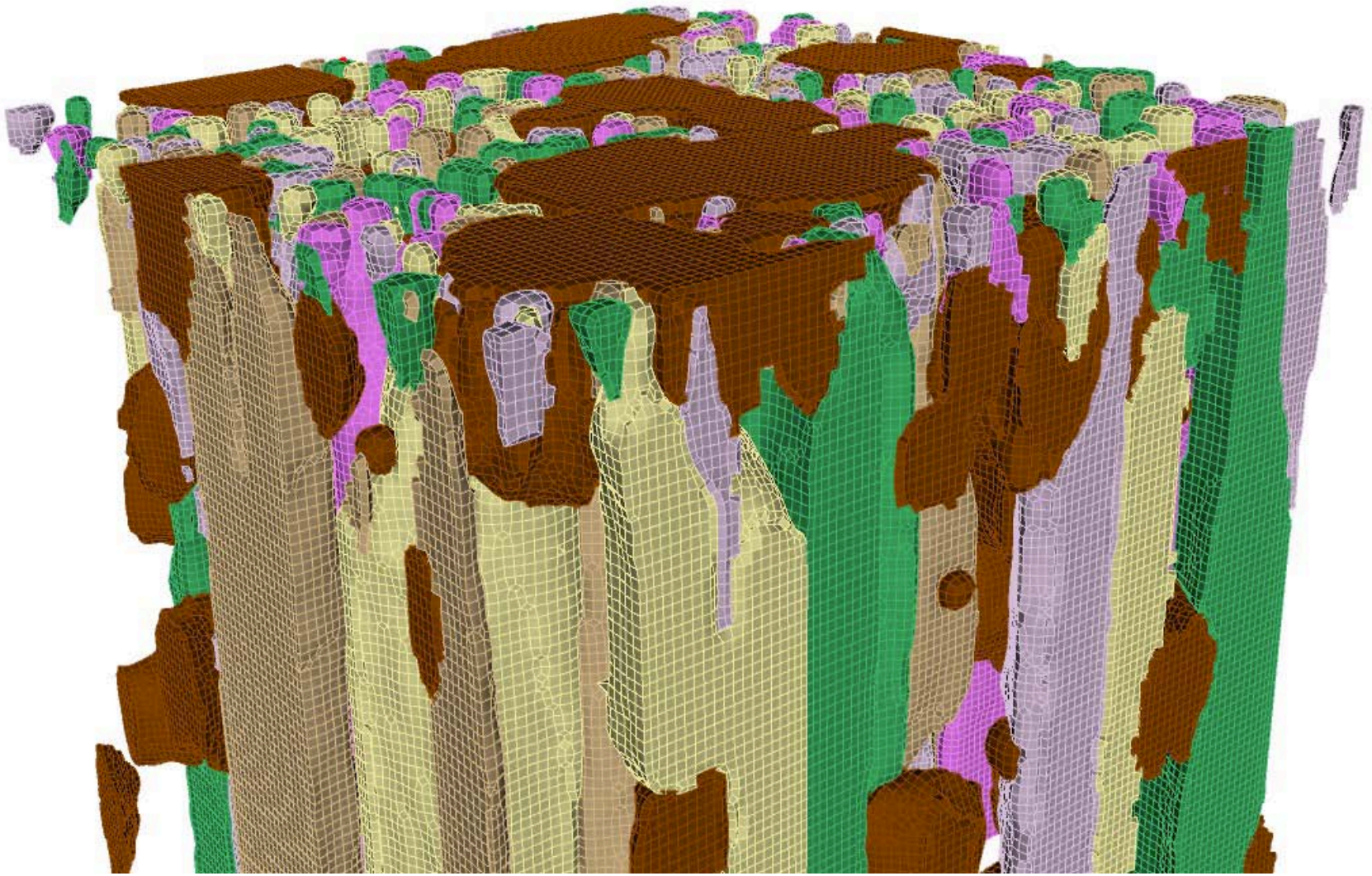
CTH Pipe Bomb Explosion





LOCKHEED MARTIN

Sandia National Laboratories

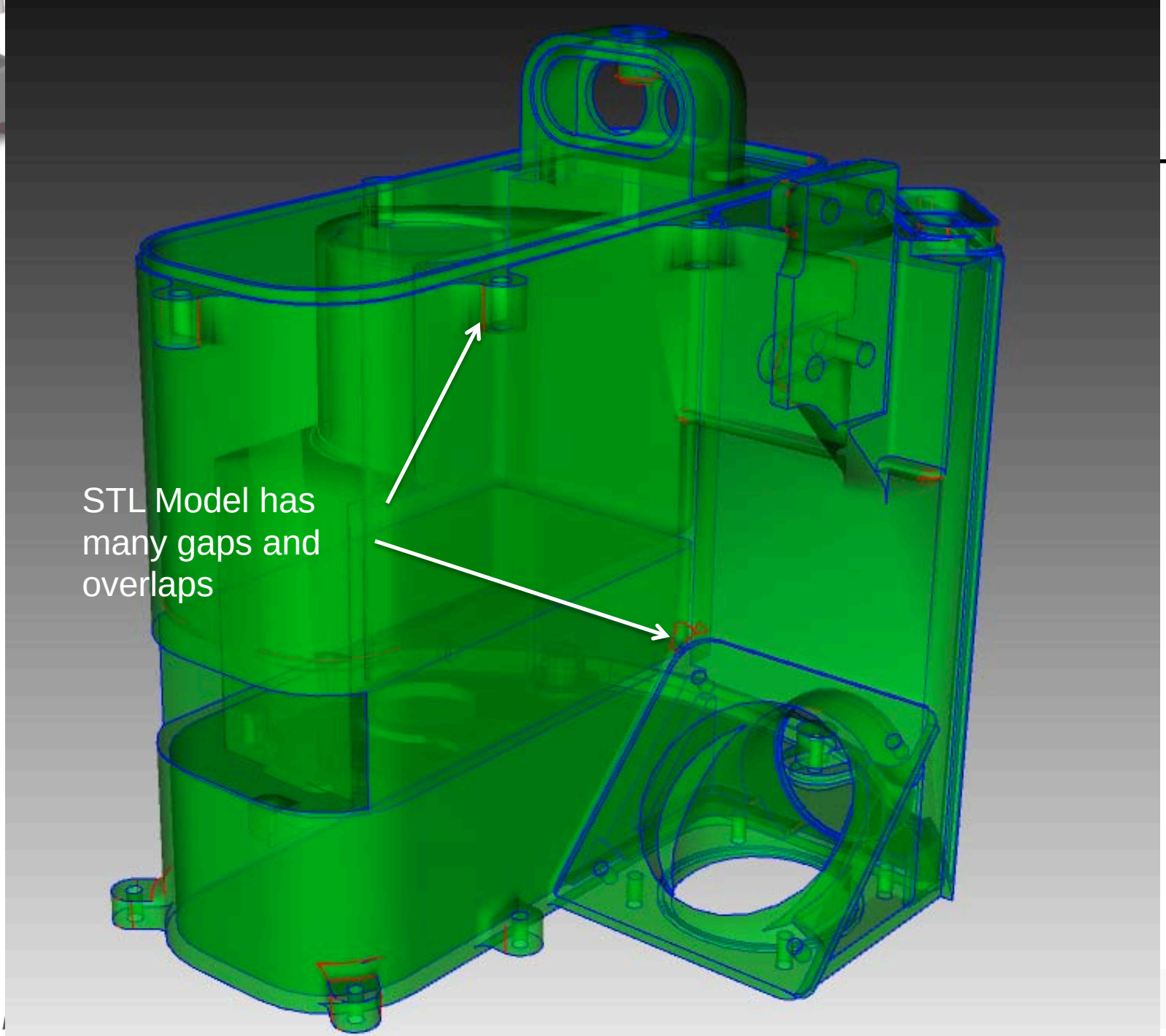


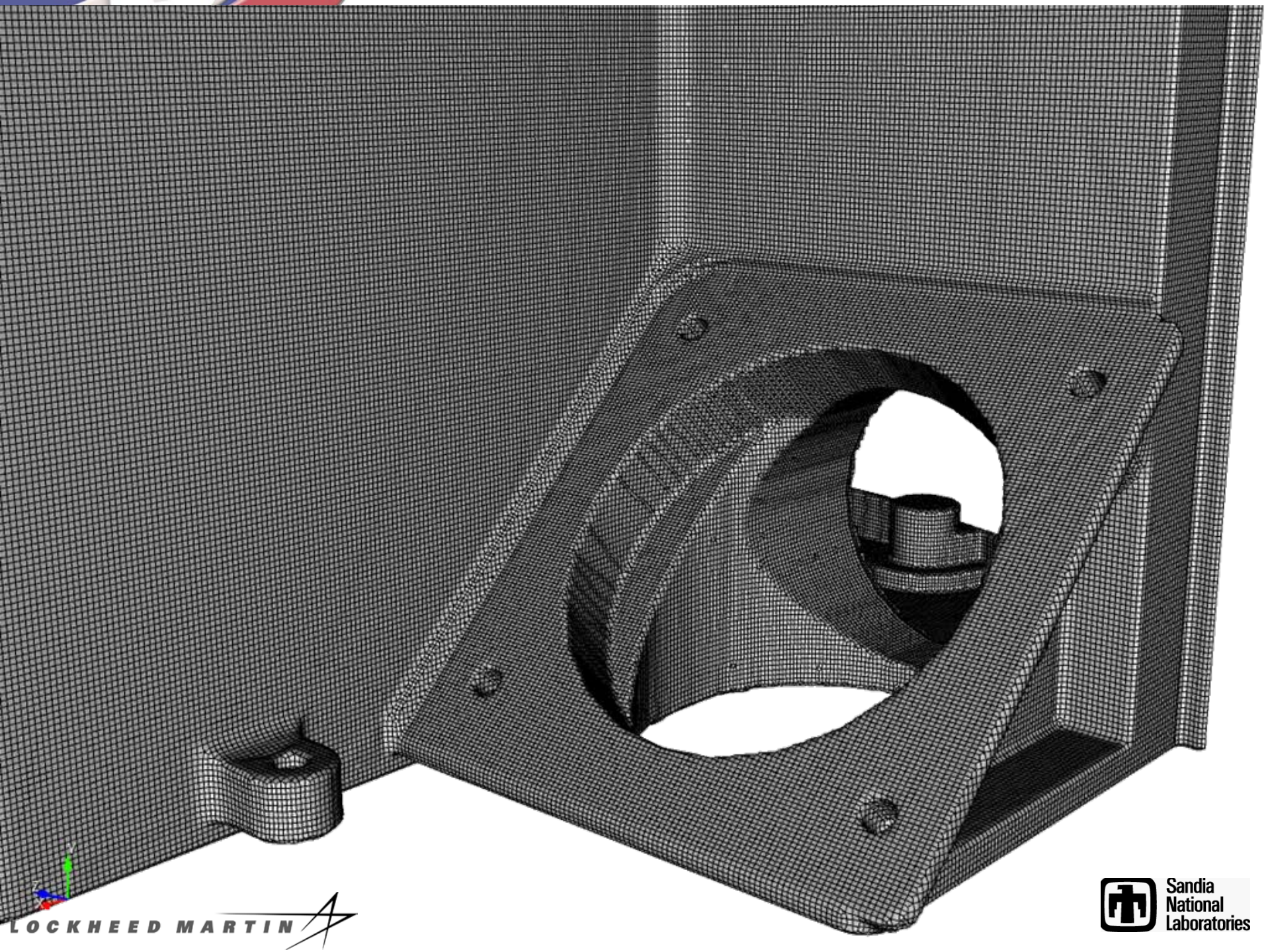
LOCKHEED MARTIN

Sandia National Laboratories

LOCKHEED MARTIN

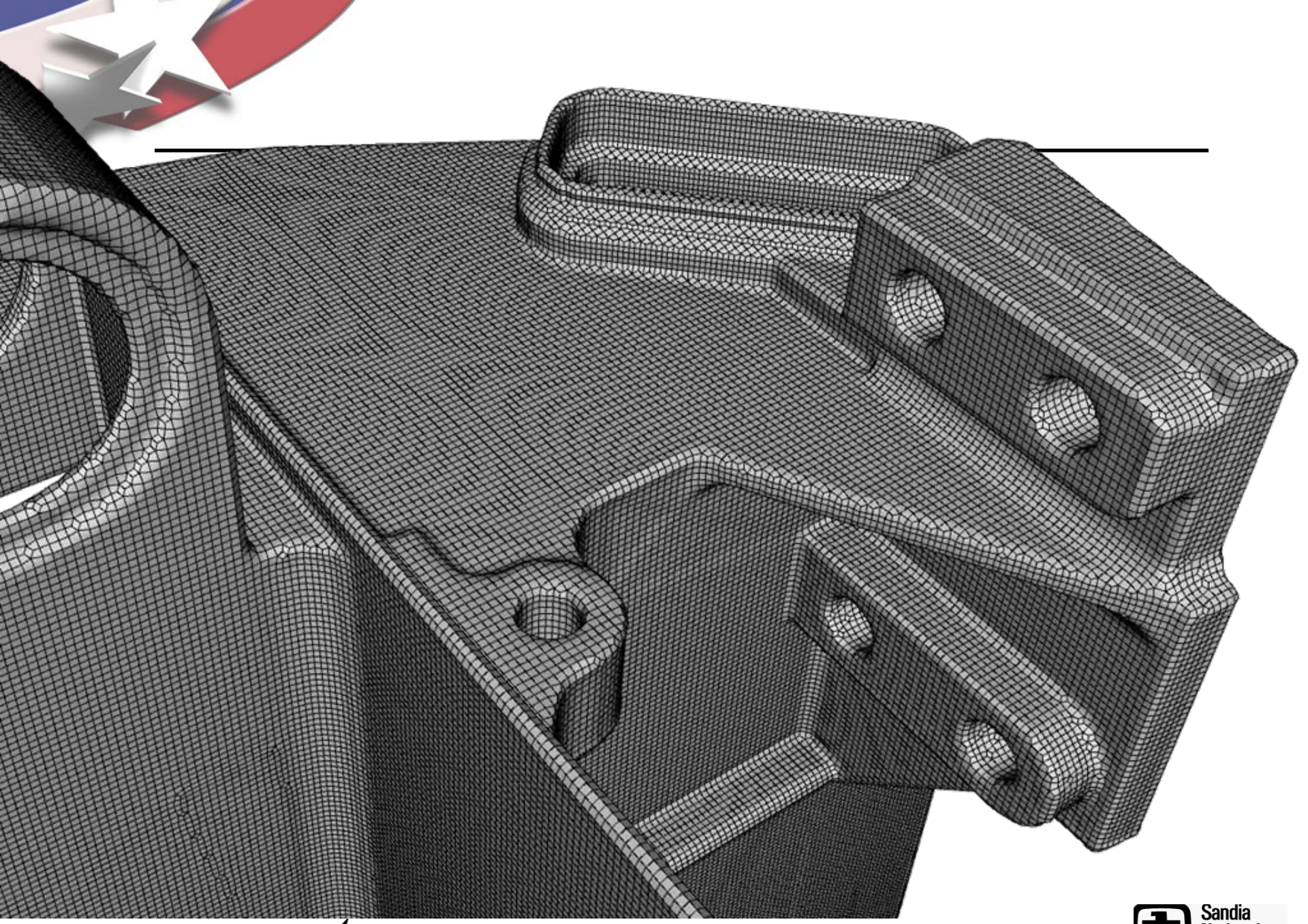
Sandia National Laboratories


STL Model has
many gaps and
overlaps



LOCKHEED MARTIN

Sandia National Laboratories



LOCKHEED MARTIN

Sandia National Laboratories