Fracture and Fragmentation of Simplicial Finite Element Meshes using Graphs

Alejandro Mota[1], Jaroslaw Knap[2]

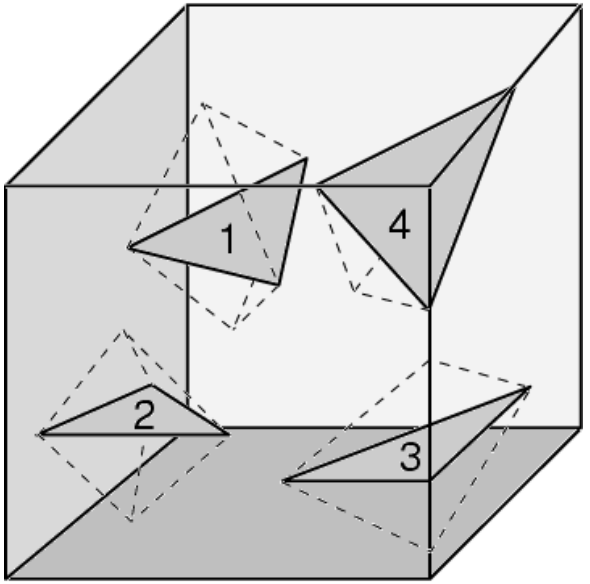
[1]Sandia National Laboratories, Livermore CA
[2]Army Research Laboratory, Aberdeen MD

CAT Workshop
Santa Fe, NM, August 29, 2009

Sandia National Laboratories

Existing Fracture Algorithm

Implemented using a 4-way linked-list data structure.

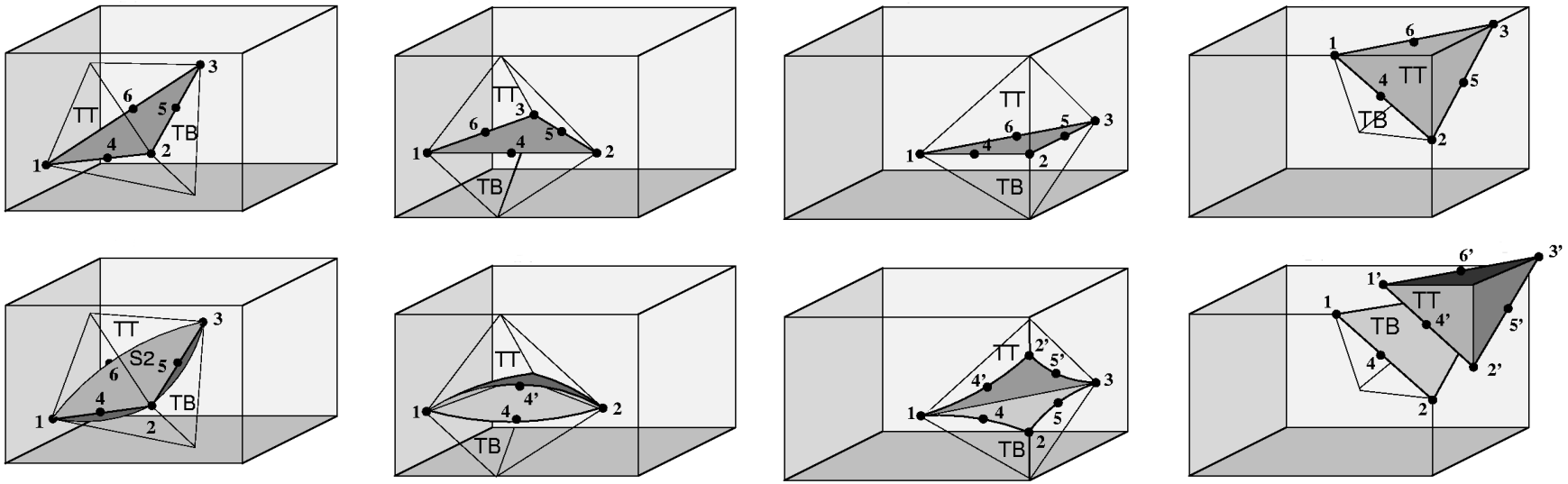


Pandolfi and Ortiz (1999)

Sandia National Laboratories

Fragmentation Cases

Change of topology according to the number of boundary sides of the triangle to be duplicated:



0 sides 1 side 2 sides 3 sides

Sandia National Laboratories

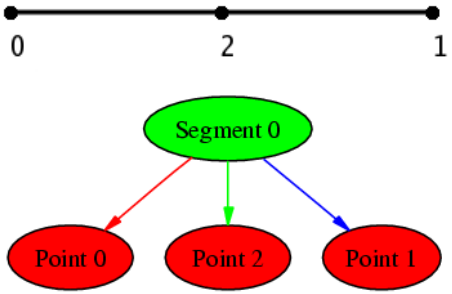
Motivation

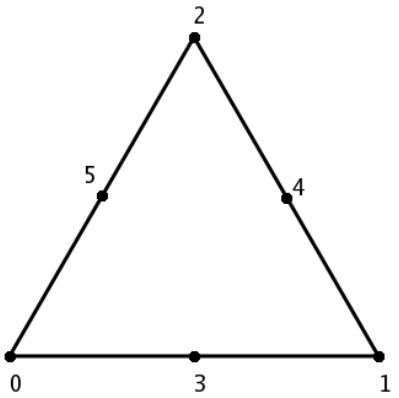
- Parallel 3D fracture and fragmentation.
- Parallel 3D contact.
- Same topology representation for both.
- Reuse for 2D if possible.
- Better performance for large meshes.
- Independence of interpolation scheme.
- Correctness.

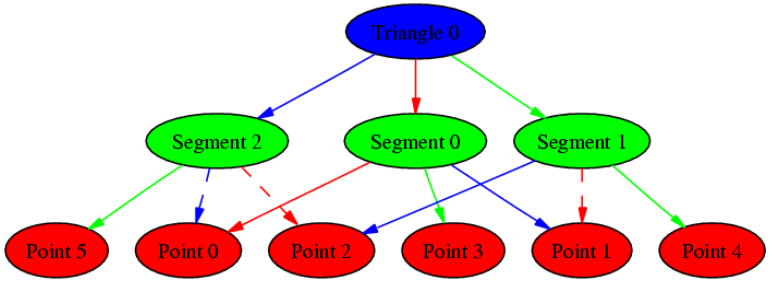
Sandia National Laboratories

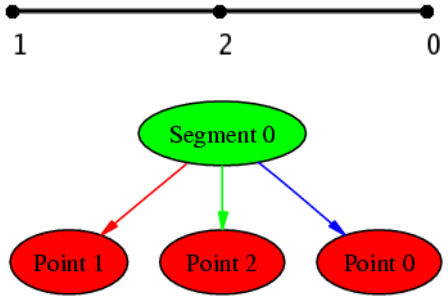
Simplices as Graphs

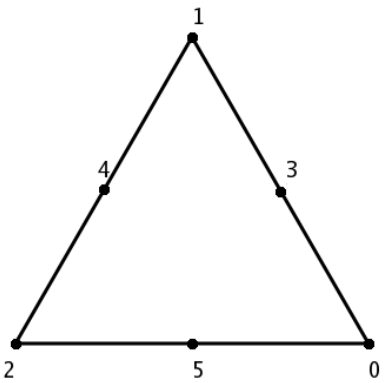
Simplices are graphs vertices, and colored edges represent connectivity and orientation. Graphs are directed.

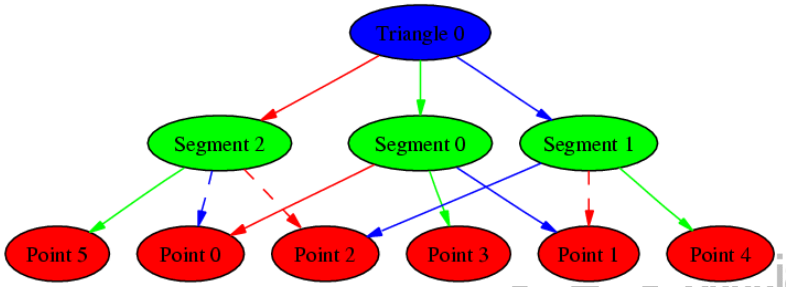








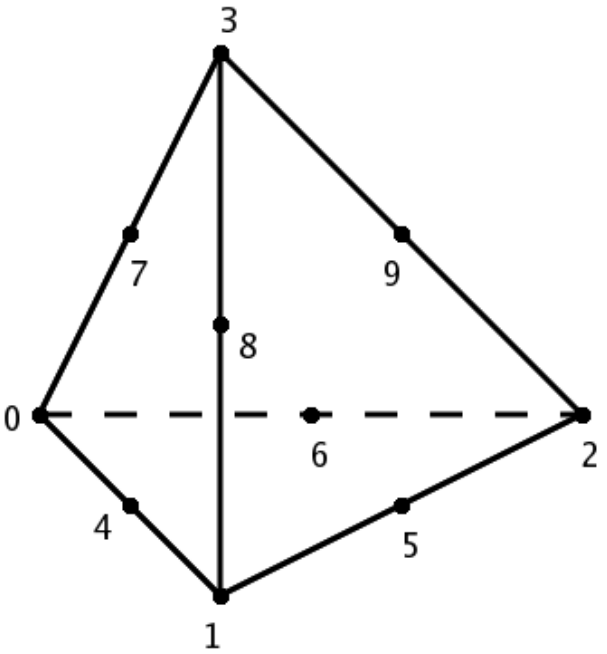


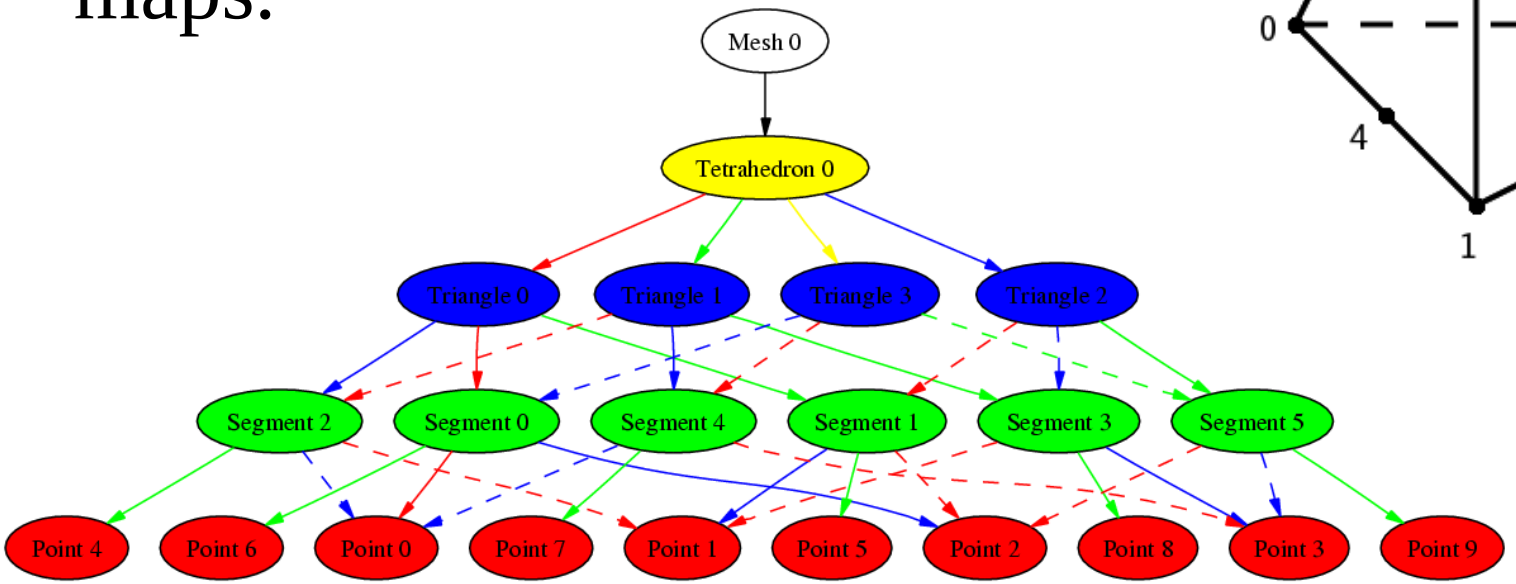


Sandia National Laboratories

FE Meshes as Graphs

- Building of graph is top-down.
- Vertices are shared.
- Original orientation kept by color maps.





Sandia National Laboratories

FE Meshes as Graphs

Simplex:

$$\sigma = [x_0, \ldots, x_n]$$

Face operator:

$$d_i(\sigma) := [x_0, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n]$$

Incidence number:

$$[\sigma^p, \sigma^{p-1}] := \begin{cases} 0 & \text{if } \sigma^p \cap \sigma^{p-1} = \emptyset \\ 1 & \text{if } \sigma^{p-1} = d_i(\sigma^p) \\ -1 & \text{if } \sigma^{p-1} = -d_i(\sigma^p) \end{cases}$$

Graph:

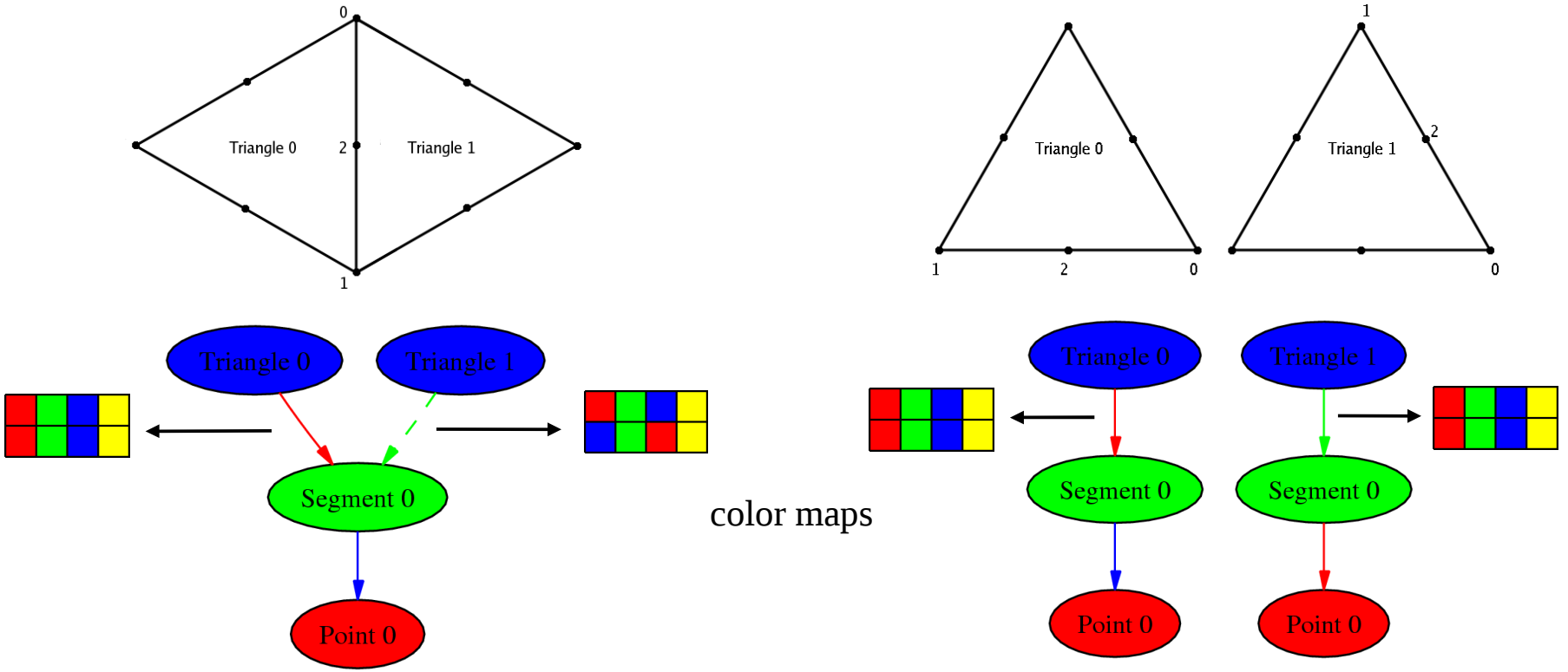
$$V = \{v | v = f(\sigma) \in \mathbb{N}, \sigma \in K\}$$

$$E = \{e | e = (u, v), u = f(\sigma^p) \in V, v = f(\sigma^{p-1}) \in V, \sigma^p \in K, \sigma^{p-1} \in K, [\sigma^p, \sigma^{p-1}] \neq 0\}$$

National Laboratories

Local Ordering as Color Maps

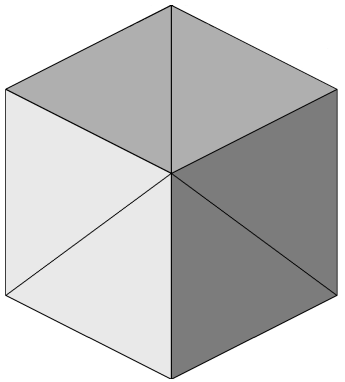
Triangle 0 sees point 0 as the blue point in segment 0. Triangle 1 sees it as red.

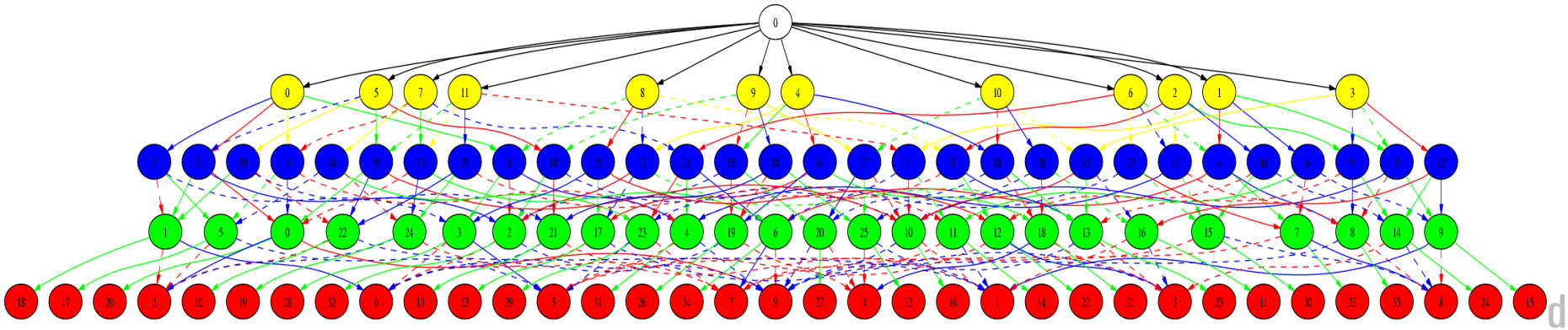


Sandia National Laboratories

Cube Mesh as Graph

Graphs are quite complex even for meshes with a few elements.

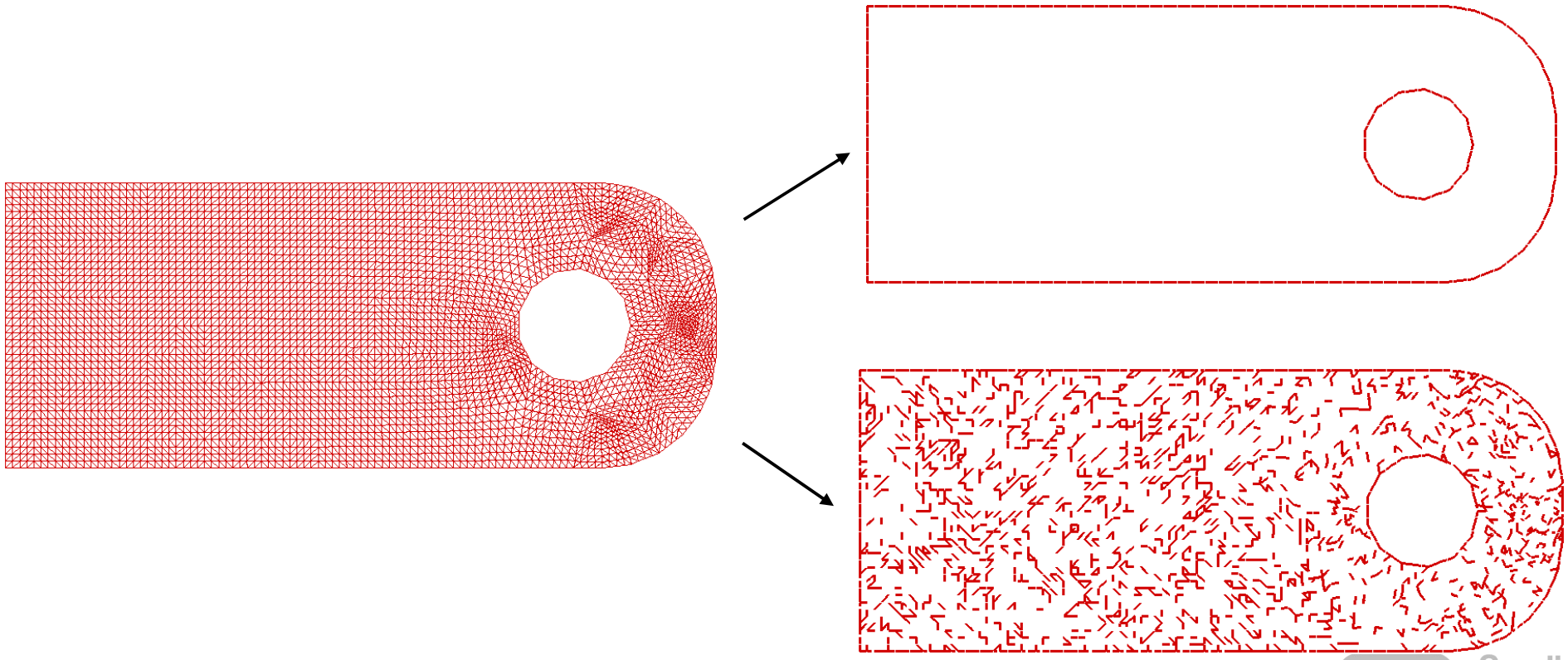




dia
National
Laboratories

Plate with Hole Mesh

Same graph representation works for 2D. Easy to extract useful features.



Sandia National Laboratories

Simplex Graph Properties

- Simplices are vertices.
- Edges represent connectivity.
- Directed graph.
- Built top-down.
- Edge color is local order of n-simplex wrt to (n+1)-simplex.
- An n-simplex may be shared by many (n+1)-simplices.

Sandia National Laboratories

Simplex Graph Properties (2)

- Edge color map modifies color of edge directly below if target vertex is shared.
- Color maps are linear maps akin to reflections and rotations, or single and double permutations.
- Graph depth represents the dimension of the mesh. Works for N dimensions.
- Connectivity array can always be recovered from graph.

Sandia National Laboratories

Graph Fracture Algorithm

- Mark open triangles, segments, and corner points.
- Build segment subgraphs.
- Clone open triangles.
- Split segment vertices.
- Build point subgraphs.
- Split point vertices.

Sandia National Laboratories

Graph Fracture Algorithm

Algorithm 1. SPLIT(G, U, n, i) Split articulation points.

Require: $U \subset V_i$, $i \leqslant n-1$

for all $v \in U$ **do**
  if $i < n-2$ **then**
    SPLIT$(G, D^-(v), n, i+1)$
  else
    CLONE$(G, D^-(v), n)$
  end if
  $G'' \leftarrow G'(v) \setminus v$ // Check whether v is an articulation point
  for all $j \in \{2, \ldots, N(G'')\}$ **do**
    $Y \leftarrow \{u | u \in G''_j,\ g(u) = i+1\}$
    $V' \leftarrow \{V', z\}$ // Split the vertex in the subgraph and graph
    for all $u \in Y$ **do**
      $E' \leftarrow E' \setminus (u, v)$
      $E' \leftarrow \{E', (u, z)\}$
    end for
  end for
end for

Algorithm 2. CLONE(G, U, n) Duplicate fractured interface simplices.

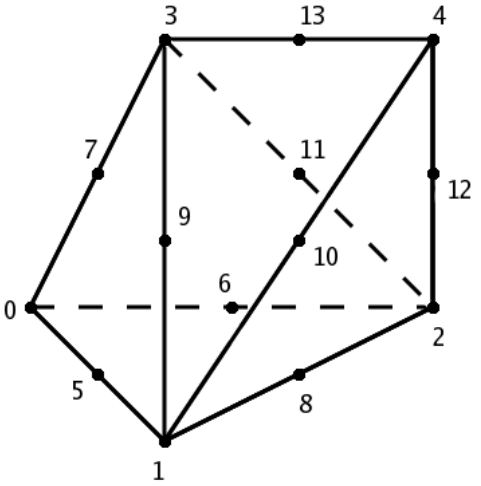
Require: $U \subset V_{\mathrm{I},n-1}$

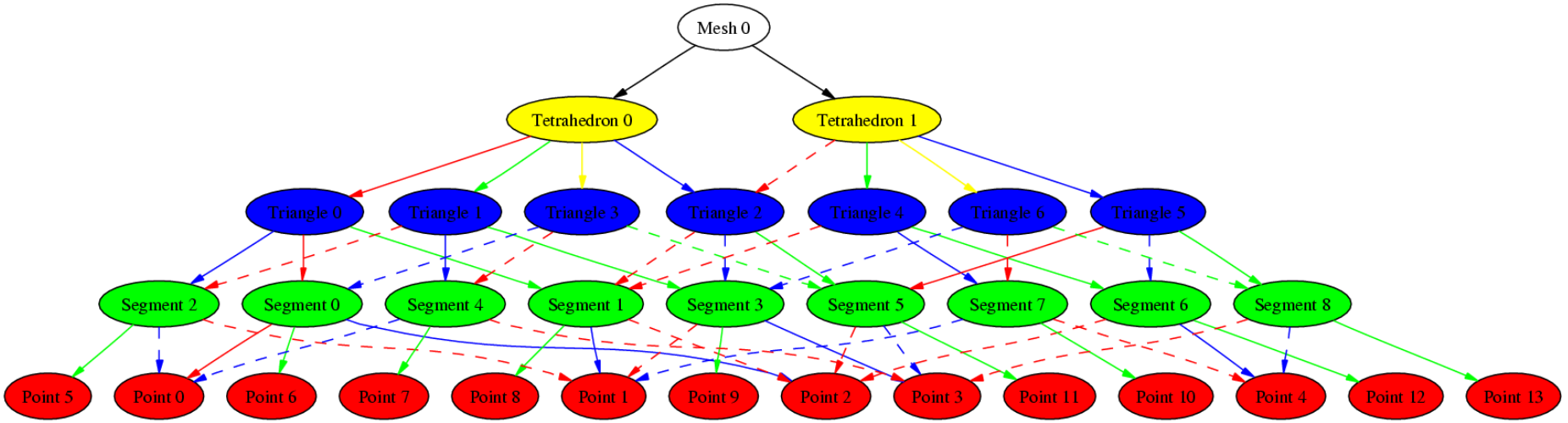
for all $v \in U$ **do**
  if $v \in V_{\mathrm{F},n-1}$ **then**
    $Y \leftarrow D^-(v)$ // Note that $d^-(v) = 2$, hence $|Y| = 2$
    $u_1 \leftarrow u \in Y$ s.t. $[f^{-1}(u_1), f^{-1}(v)] = 1$
    $u_2 \leftarrow u \in Y$ s.t. $[f^{-1}(u_2), f^{-1}(v)] = -1$
    $V \leftarrow \{V, w\}$
    $E \leftarrow E \setminus (u_2, v)$
    $E \leftarrow \{E, (u_2, w)\}$
    for all $z \in D^+(v)$ **do**
      $E \leftarrow \{E, (w, z)\}$
    end for
  end if
end for

Sandia National Laboratories

Two-Tetrahedra Mesh

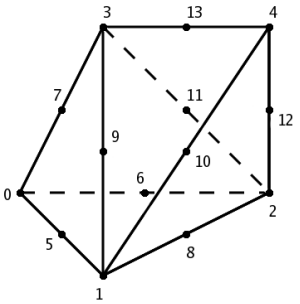
- Single shared triangle.
- Open simplices are triangle 2, segments 1,3,5 and points 1,2,3.



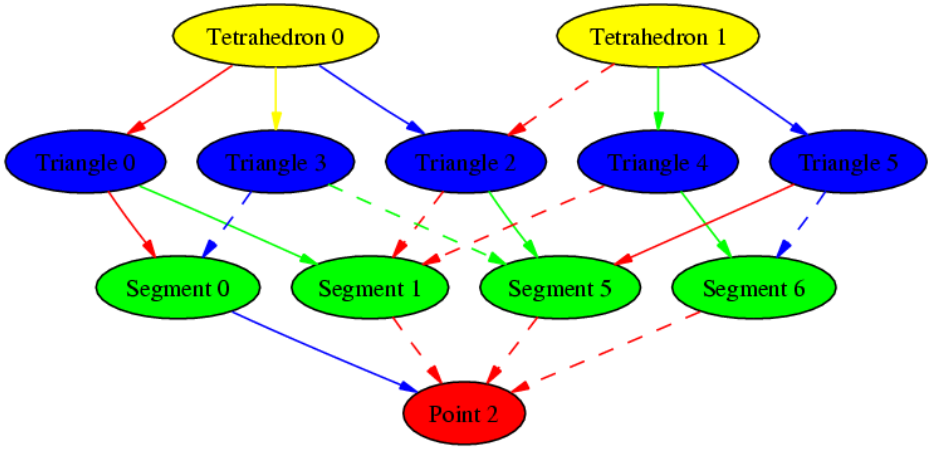


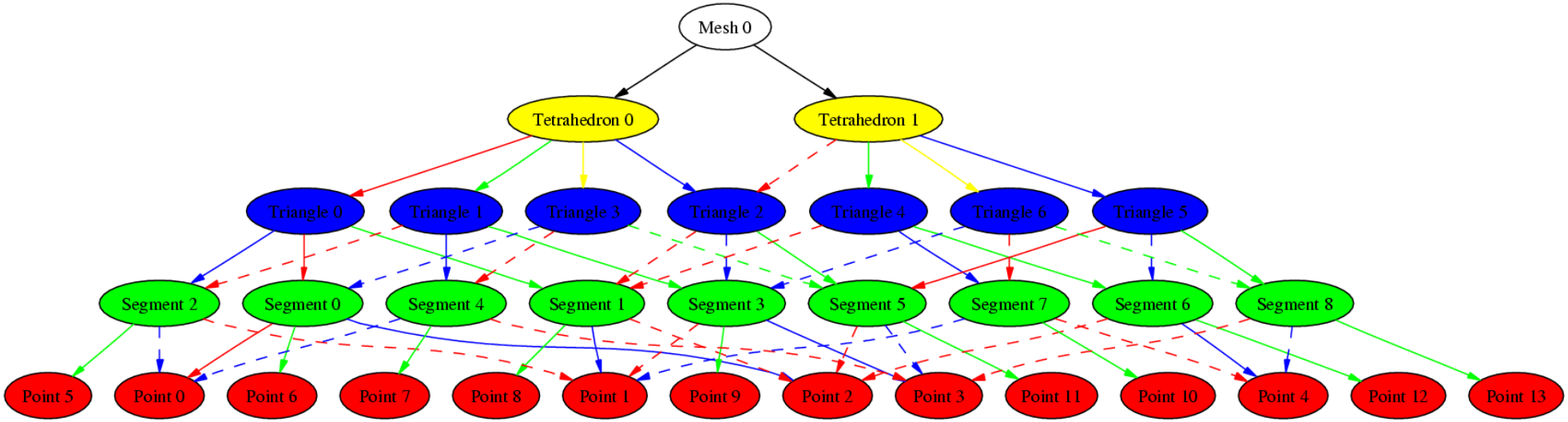
Sandia National Laboratories

Fracture Algorithm (1)



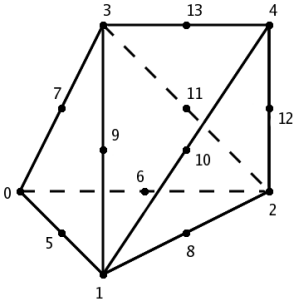
First selected open point is 2, so process open segments 1 and 5 attached to it.



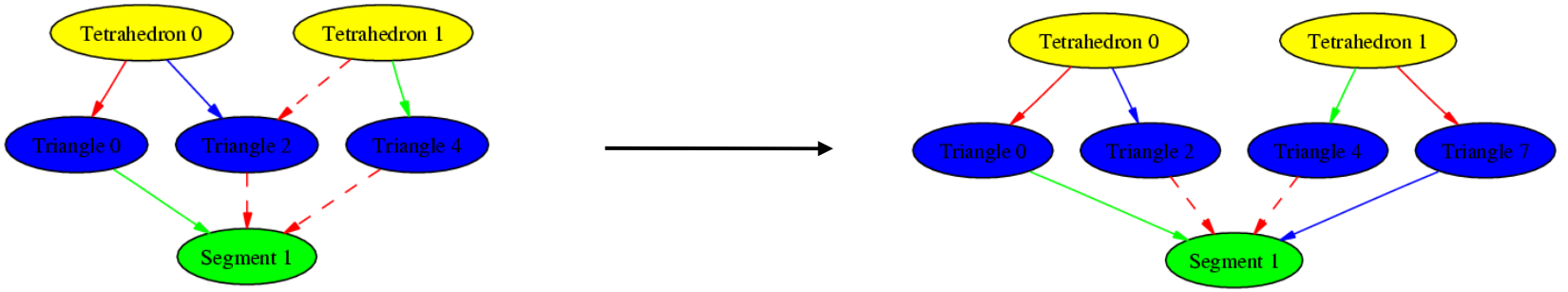


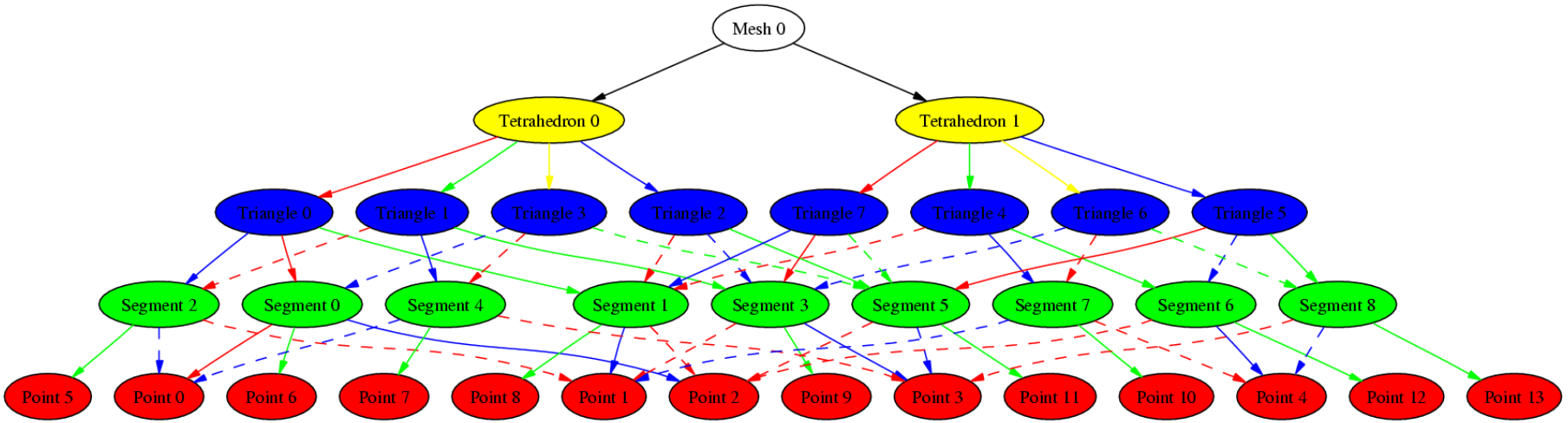
Sandia National Laboratories

Fracture Algorithm (2)



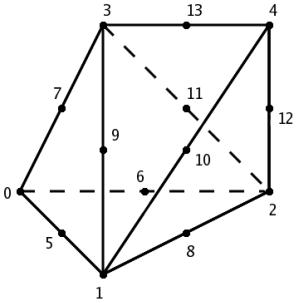
Extract subgraph for segment 1, which clones open triangle 2, creating triangle 7.



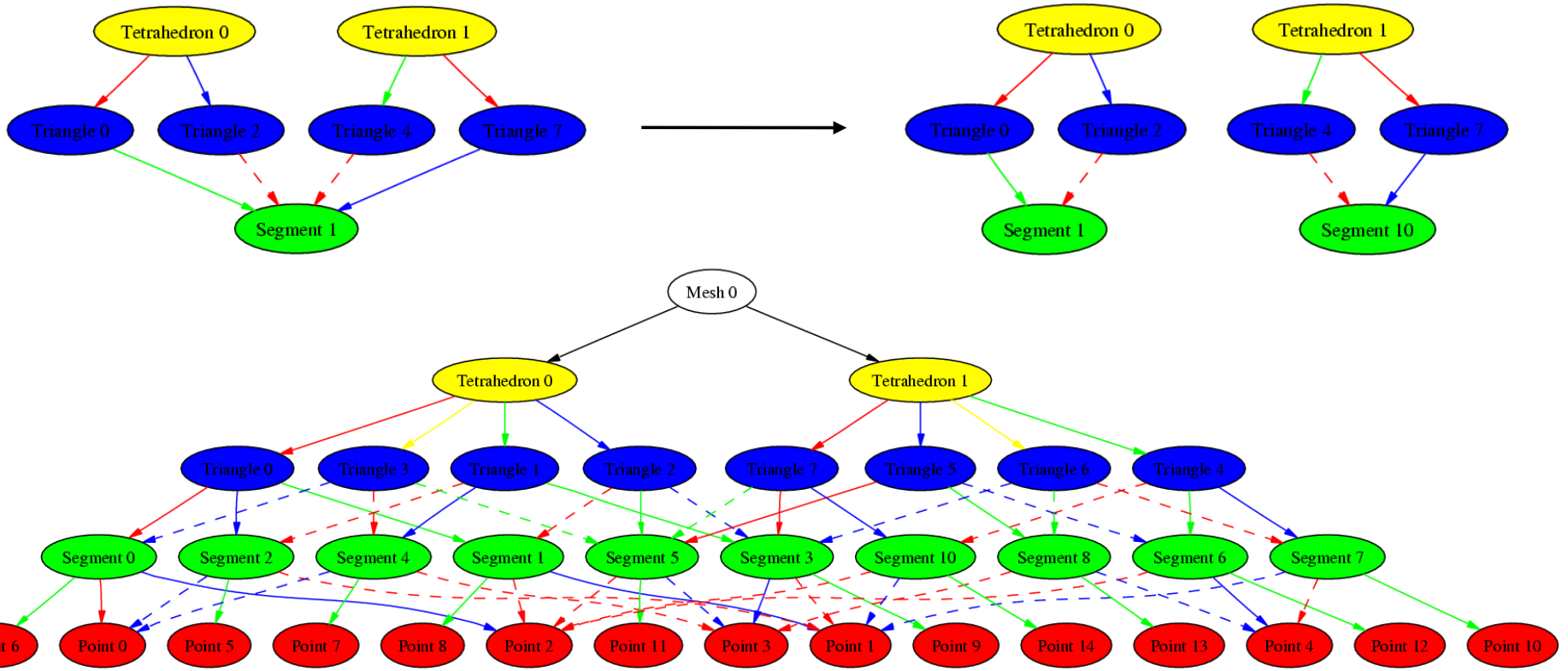


dia
National
Laboratories

Fracture Algorithm (3)

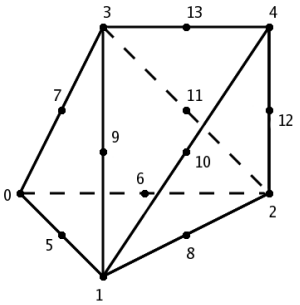


There are 2 branches in segment 1's subgraph, so the segment is split.

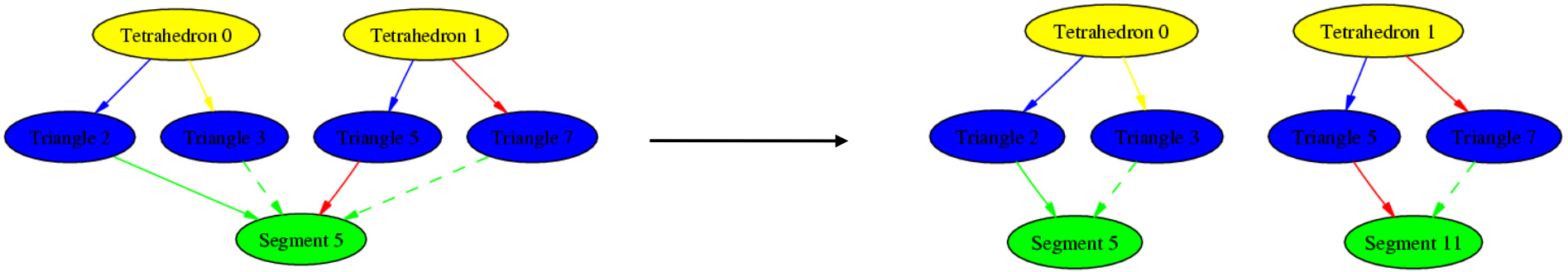


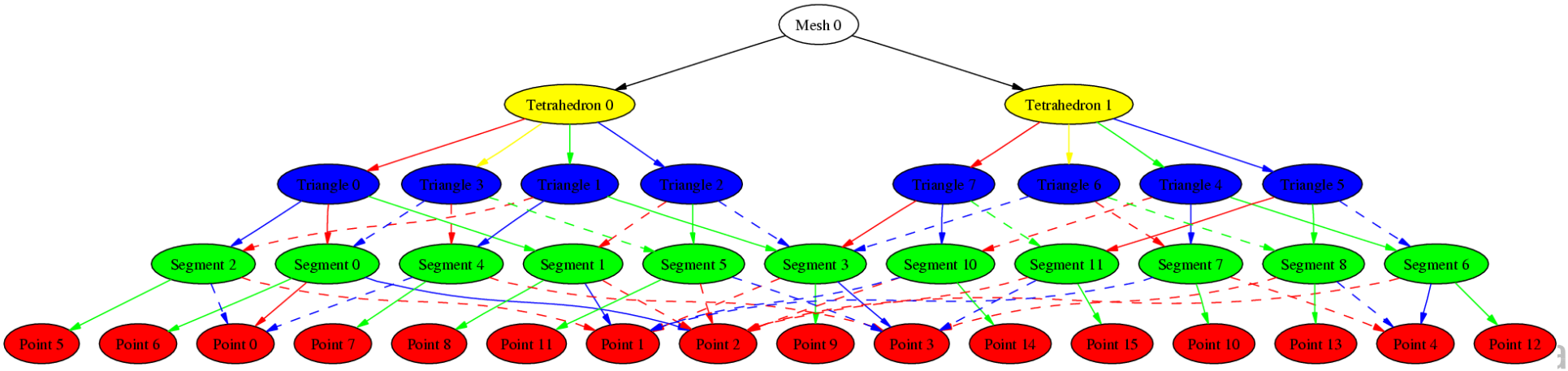
a National Laboratories

Fracture Algorithm (4)



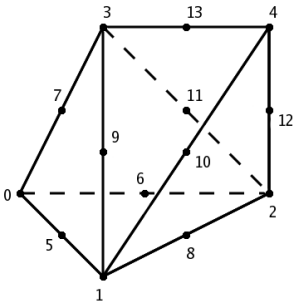
Next is segment 5, now attached to 0 open triangles.
Its subgraph has 2 branches.



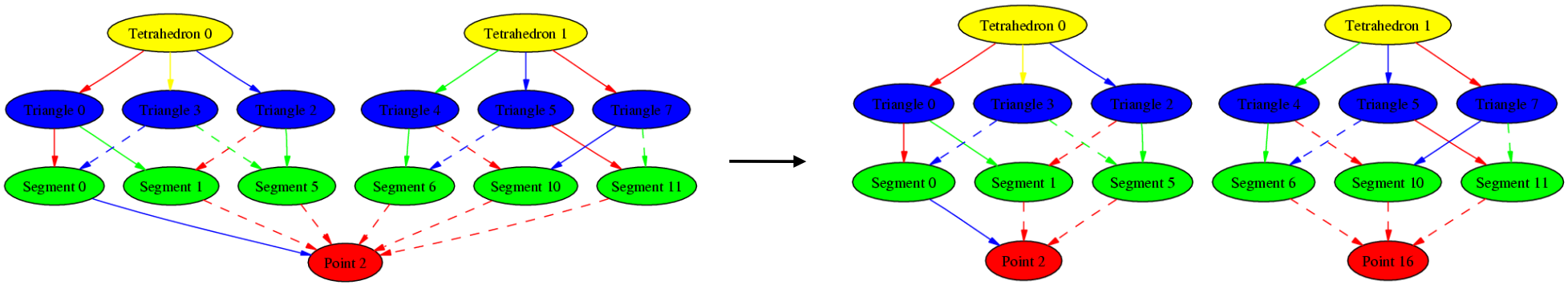


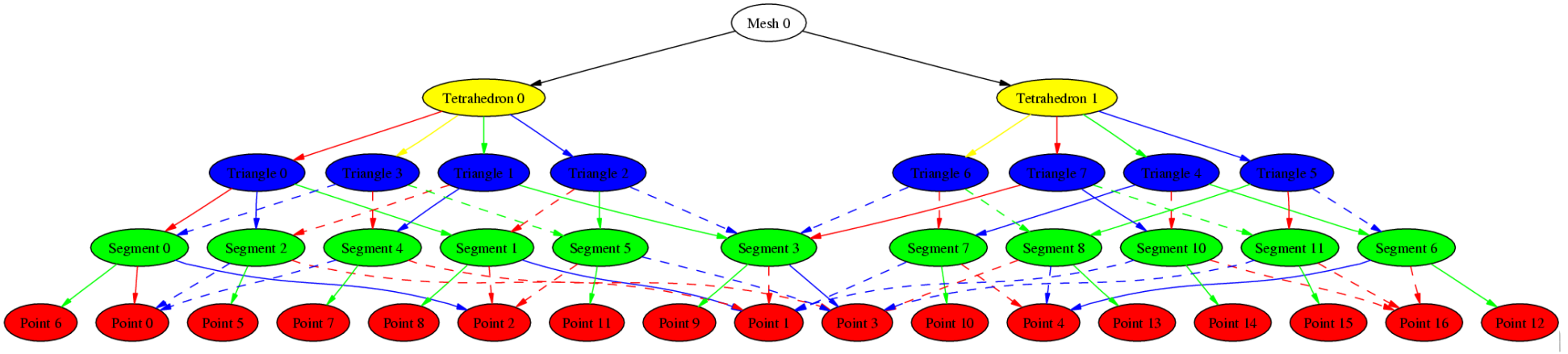
al
Laboratories

Fracture Algorithm (5)

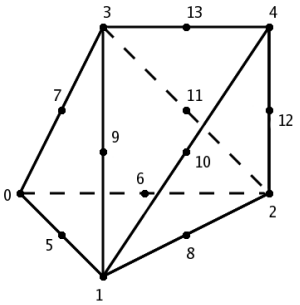


Both open segments for point 2 were split, its subgraph has 2 branches now.

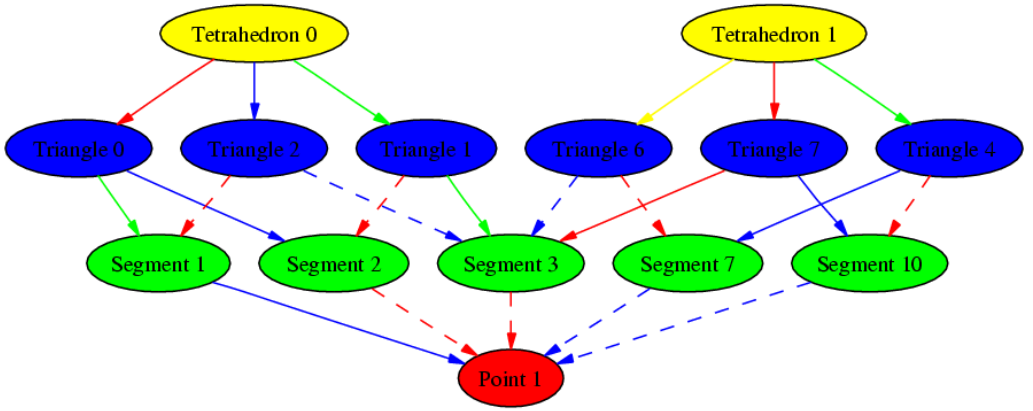


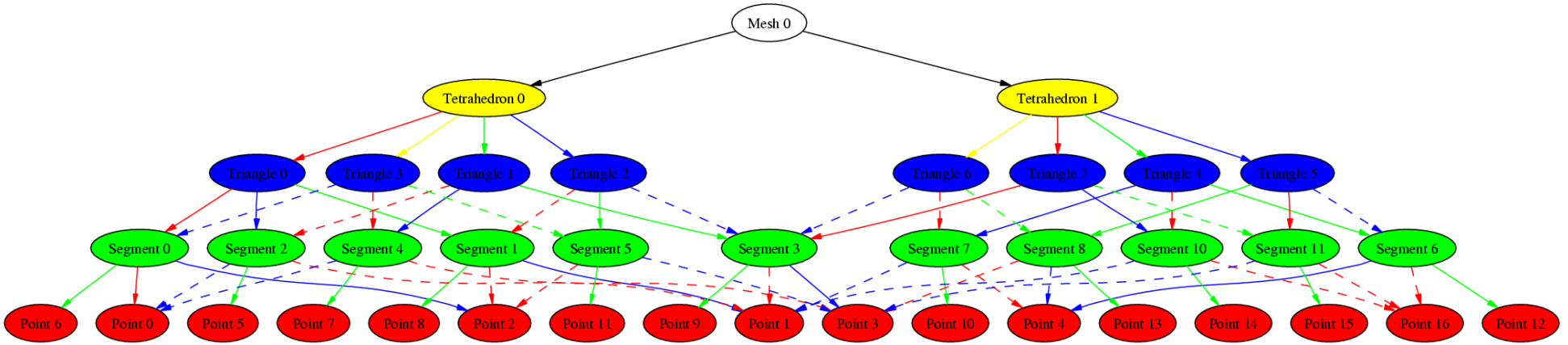


Fracture Algorithm (6)



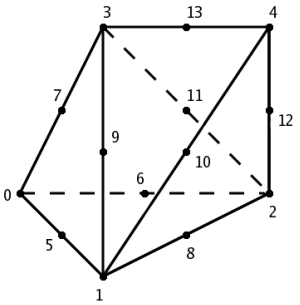
Next is point 1, with segment 3 the only remaining open segment.



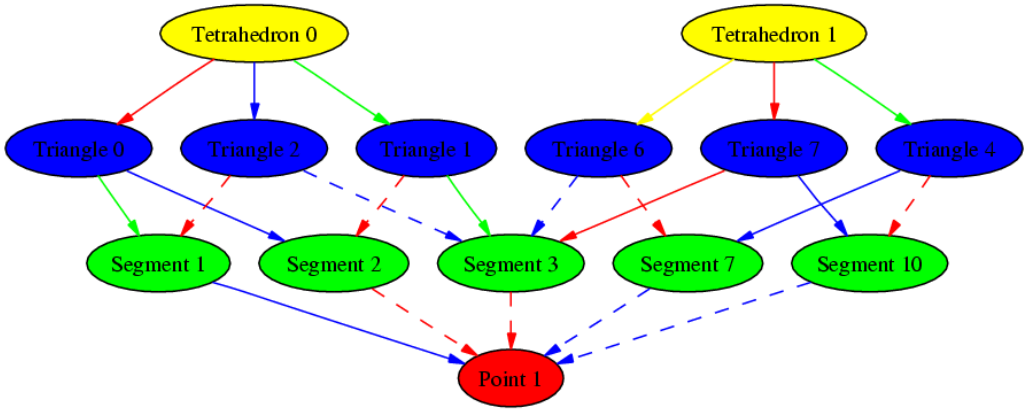


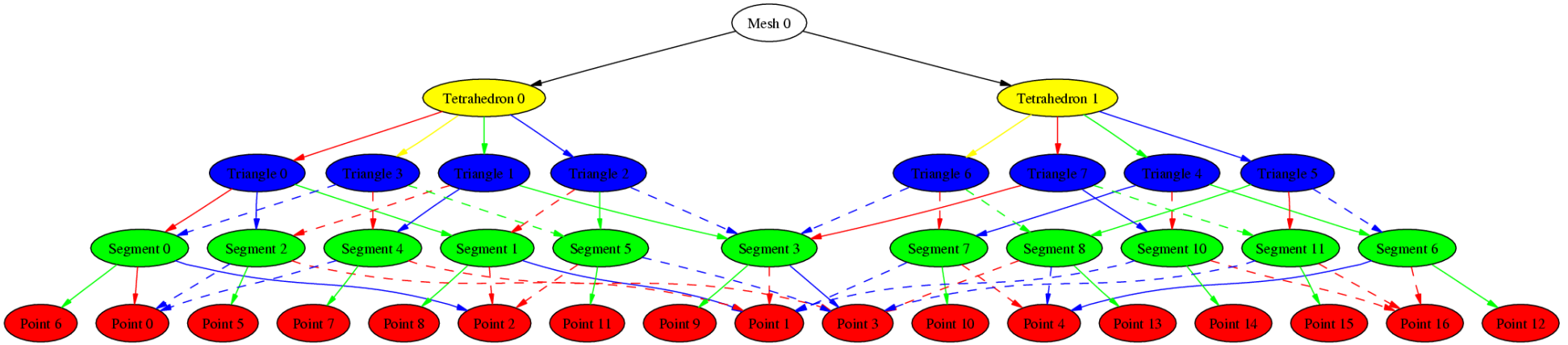
Sandia National Laboratories

Fracture Algorithm (6)



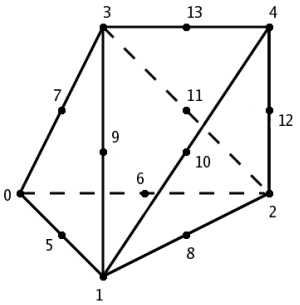
Next is point 1, with segment 3 the only remaining open segment.



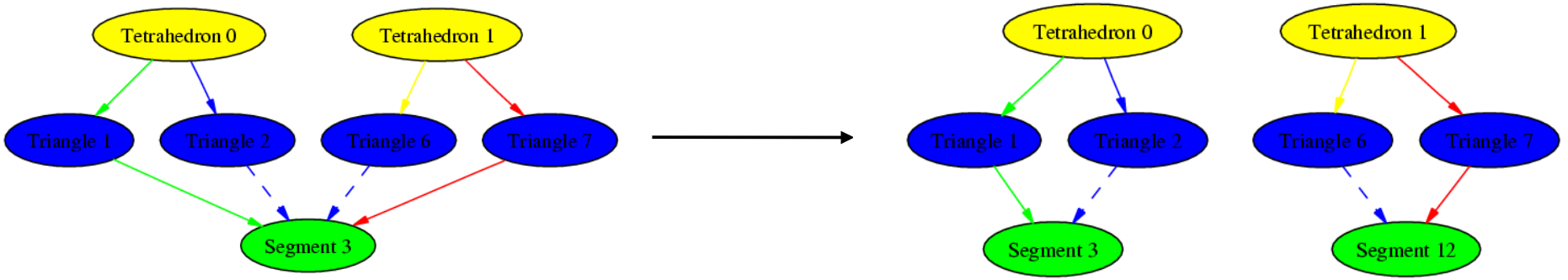


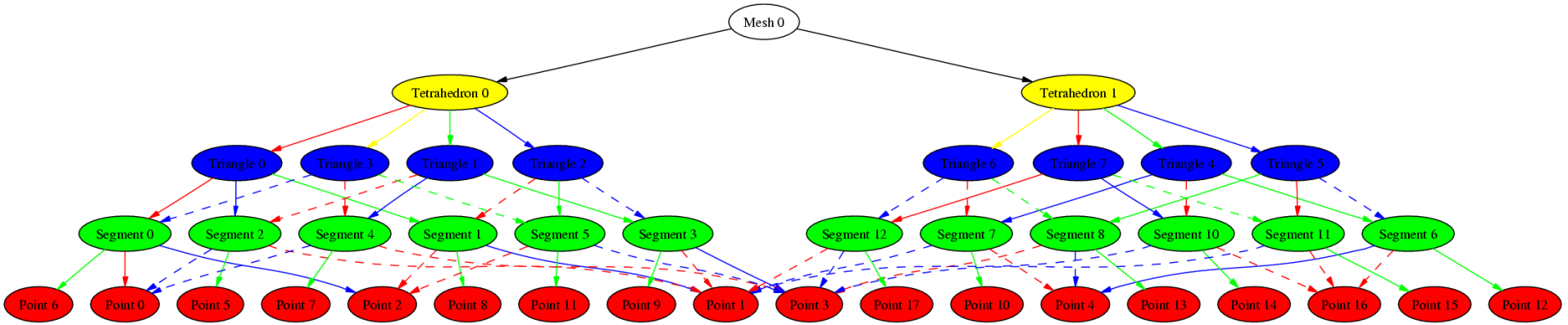
Sandia National Laboratories

Fracture Algorithm (7)



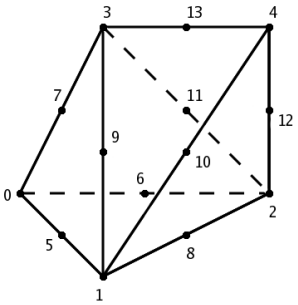
Segment 3 is split.



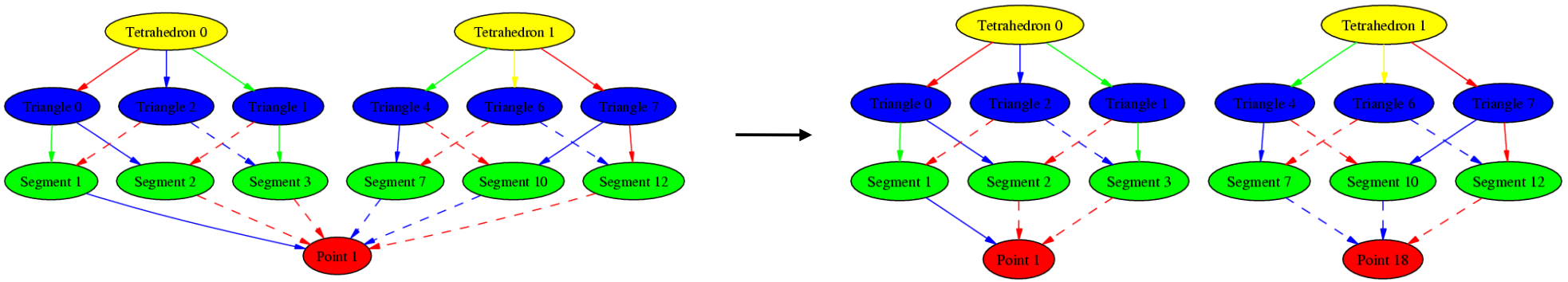


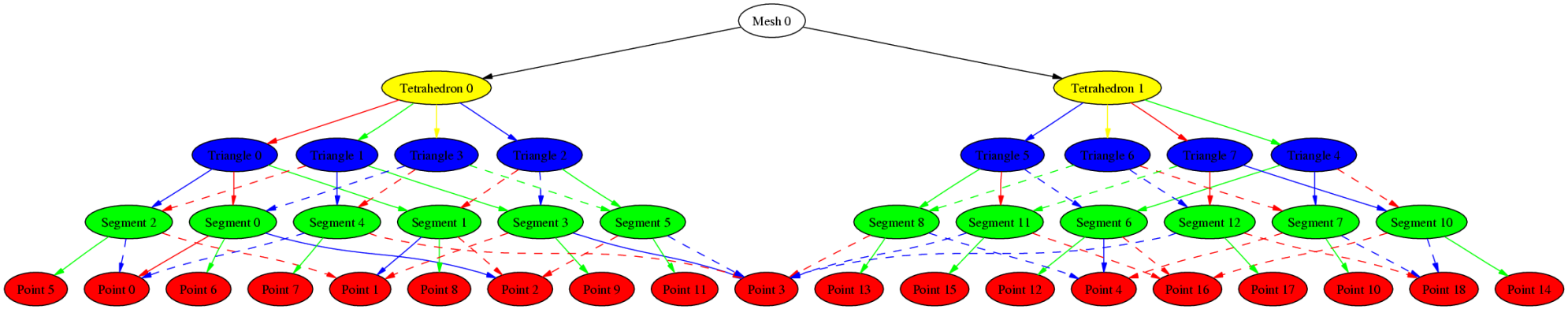
National Laboratories

Fracture Algorithm (8)



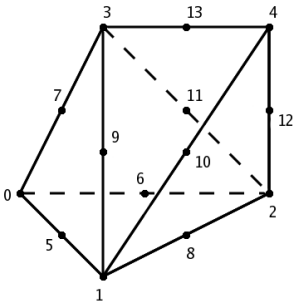
Point 1 is split.



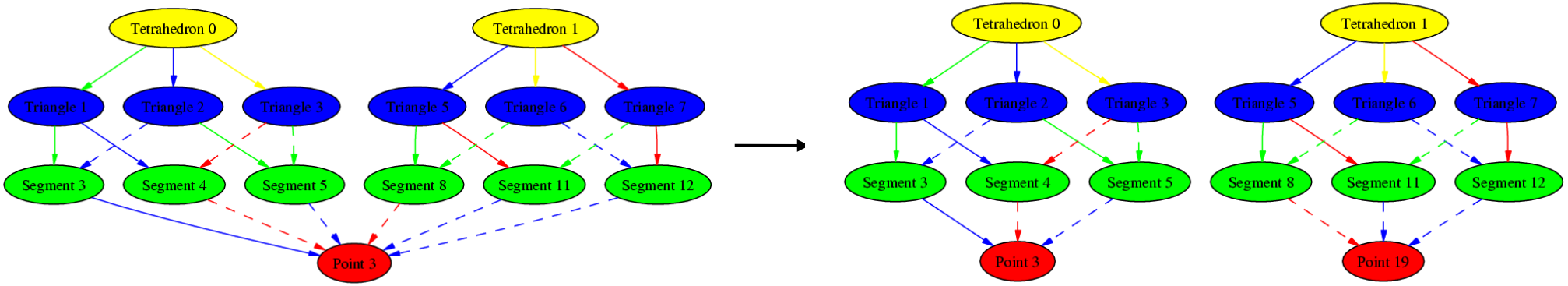


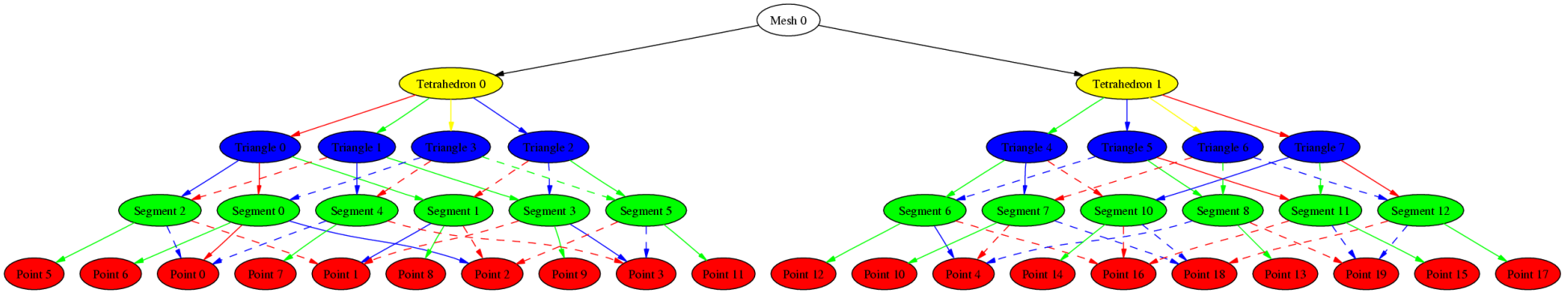
National Laboratories

Fracture Algorithm (9)



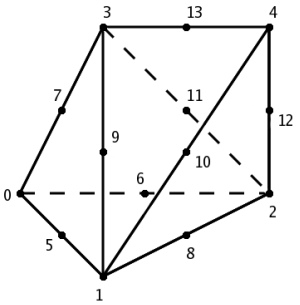
Finally point 3, with 0 open segments, is split, obtaining 2 separate tetrahedra.

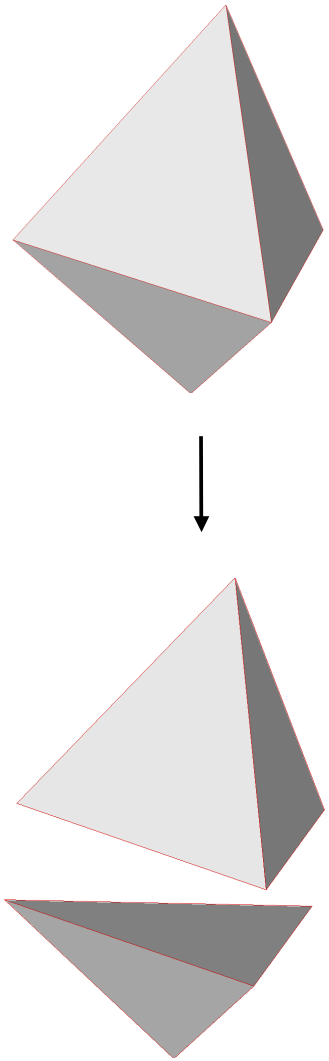




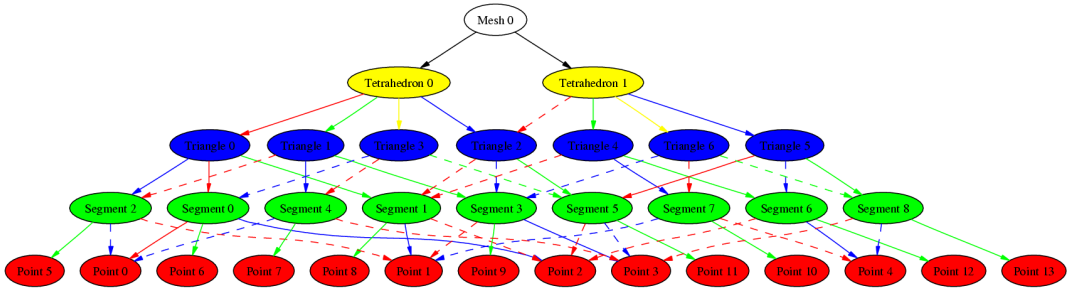
National Laboratories

Two-Tet Mesh and Graphs

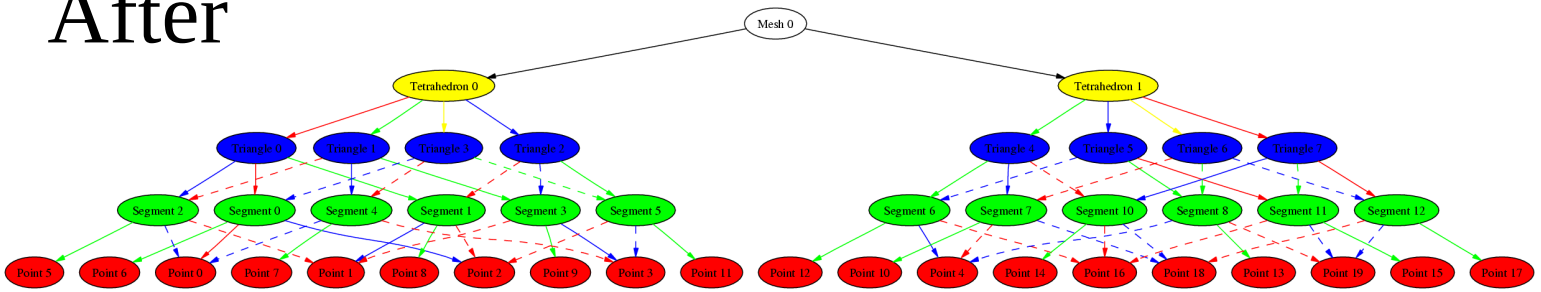




Before



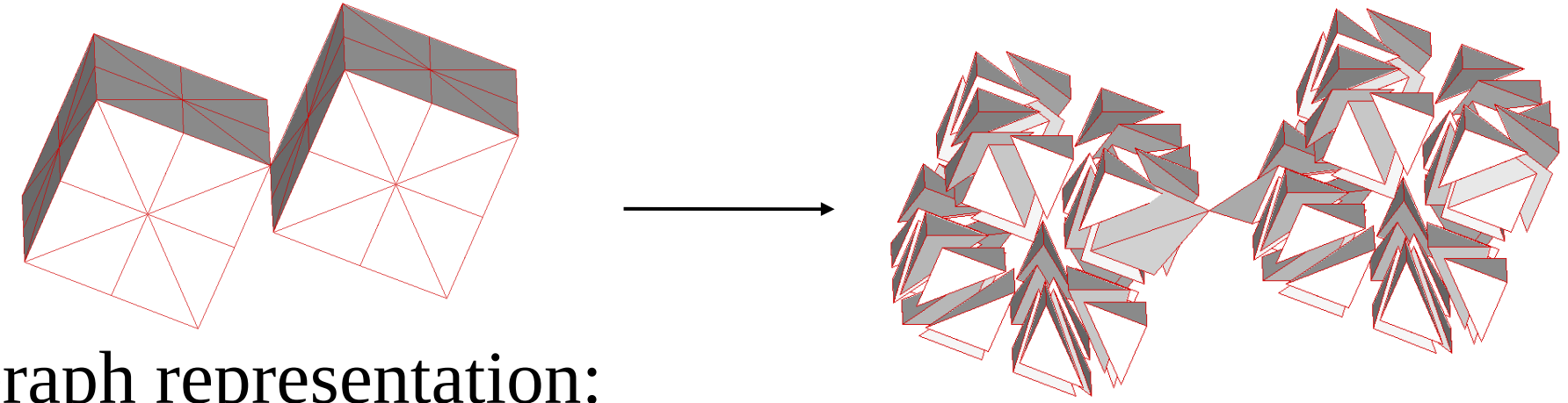
After



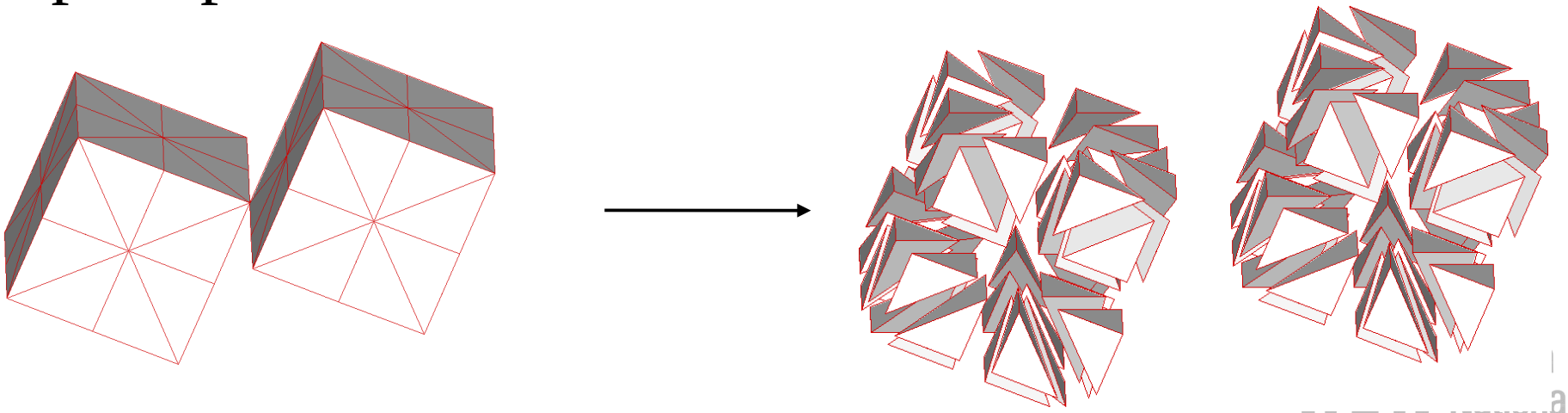
Sandia National Laboratories

Correctness – Point Cubes

Two cubes joined by a point. Fracture all internal triangles. Current algorithm:

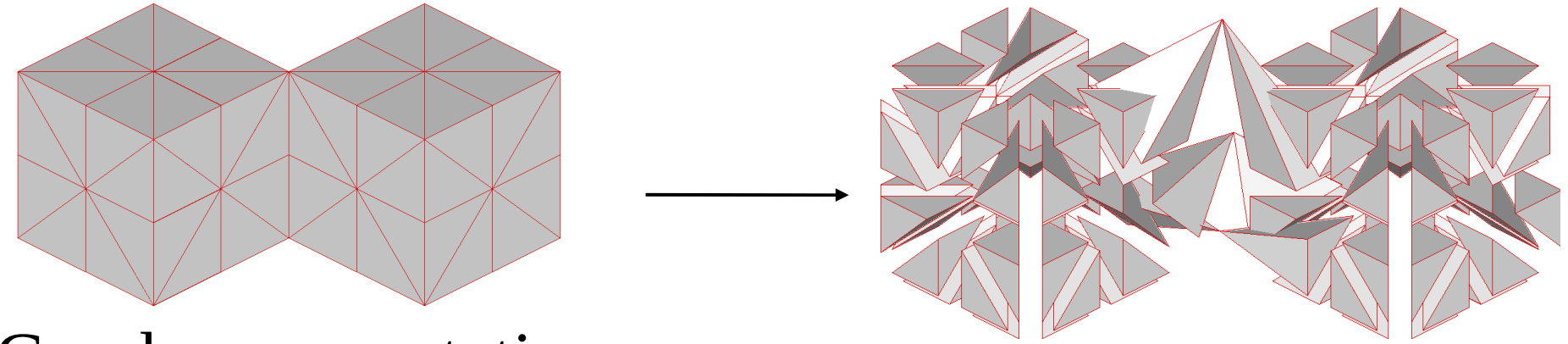


Graph representation:

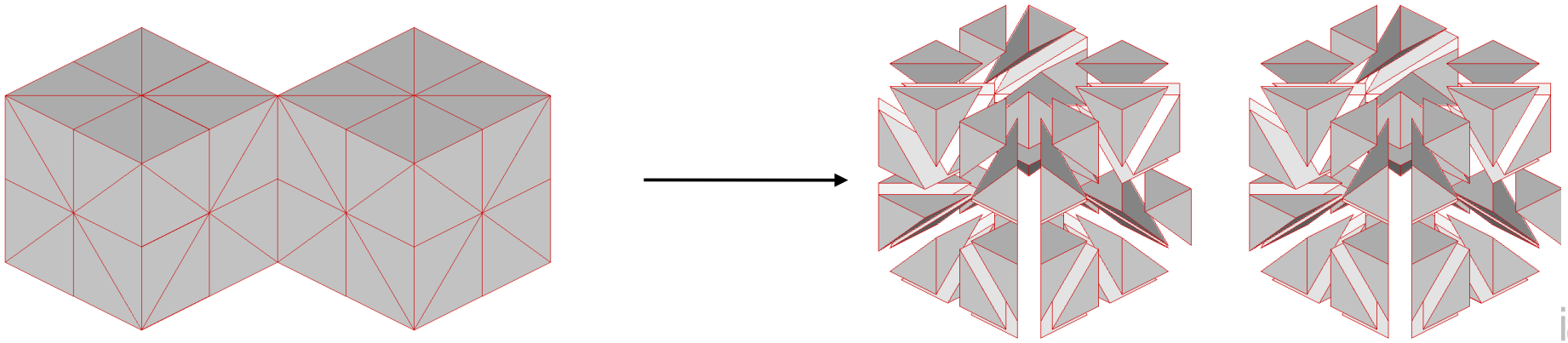


Correctness – Edge Cubes

Two cubes joined by an edge. Fracture all internal triangles. Current algorithm:



Graph representation:



ia National Laboratories

Graph Fracture Properties

- Subgraphs greatly simplify fracture.
- Operations on subgraphs are mirrored on parent graph.
- Localized operations confined to subgraphs, essential for parallelization.
- Time complexity of initialization linear with number of elements.
- Time complexity of fracture linear with number of open simplices.

Sandia National Laboratories

Graph Fracture Properties (2)

- Non-manifold cases handled correctly.
- Significant reuse of code.
- Marking of open simplices is top-down.
- Building of subgraphs is bottom-up.
- Fracture is top-down.
- Works for both 2D and 3D.
- Recursive with each level in the graph.

Sandia National Laboratories

C++ Boost Graph Library

- Free peer-reviewed portable C++ source library.
- Works well with the C++ Standard Library.
- Generic, STL-like interface for manipulating and traversing graphs.
- Hides details of the graph data structure implementation.

Sandia National Laboratories

Performance Evaluation

- Reference fracture implementation is the one used in the ARES FE code.
- Tested 20 3D meshes of various sizes and geometries.
- Initialization time plotted as a function of the mesh size.
- Fracture time plotted as a function of triangles to fracture.

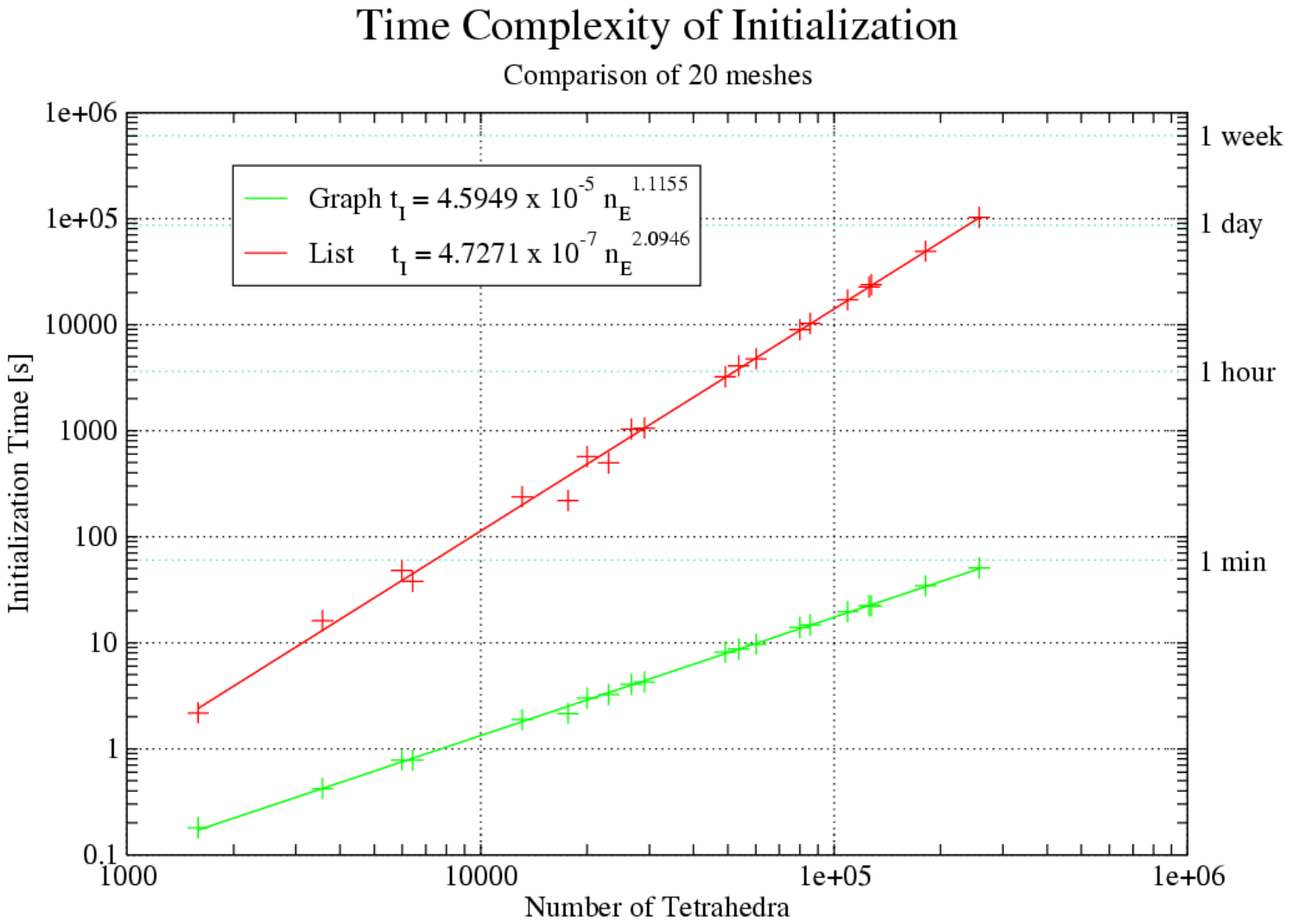
Sandia National Laboratories

Performance Evaluation (2)

- Improvements in C++ Boost library result in immediate performance gain.
- First implementation.
- Some performance tuning.
- No thorough profiling and extensive performance tuning yet.
- Tests performed using GCC on an Intel Xeon 1.5 GHz machine.

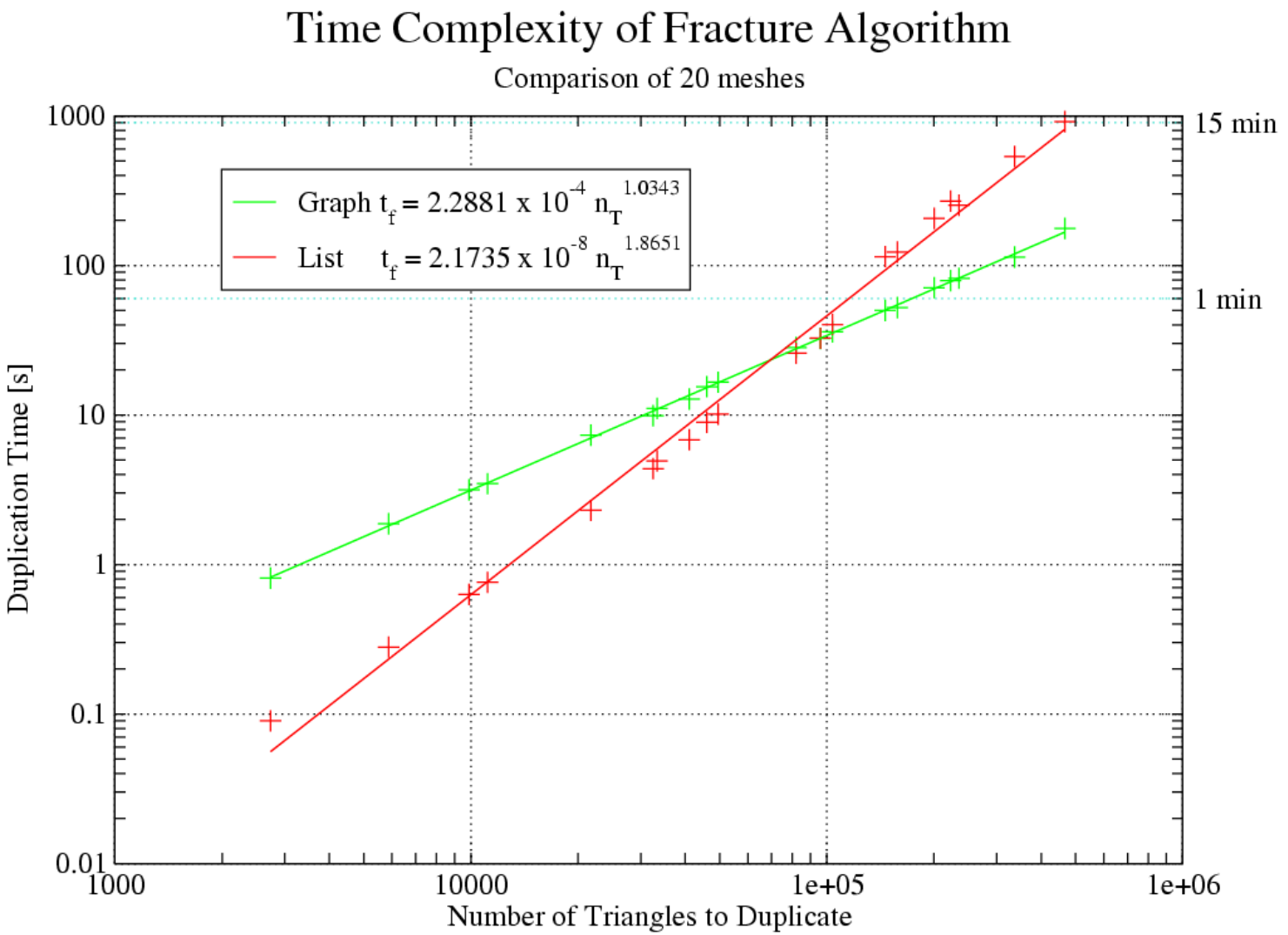
Sandia National Laboratories

Initialization Time



Sandia National Laboratories

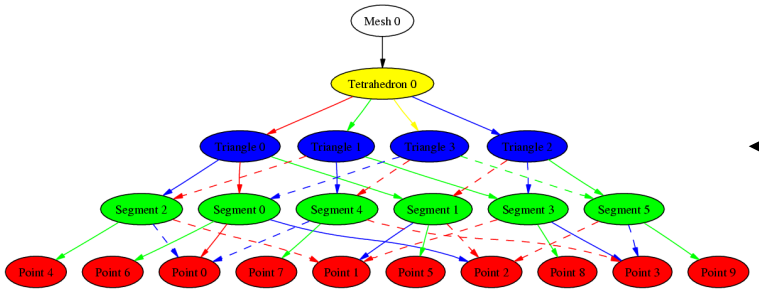
Fracture Time

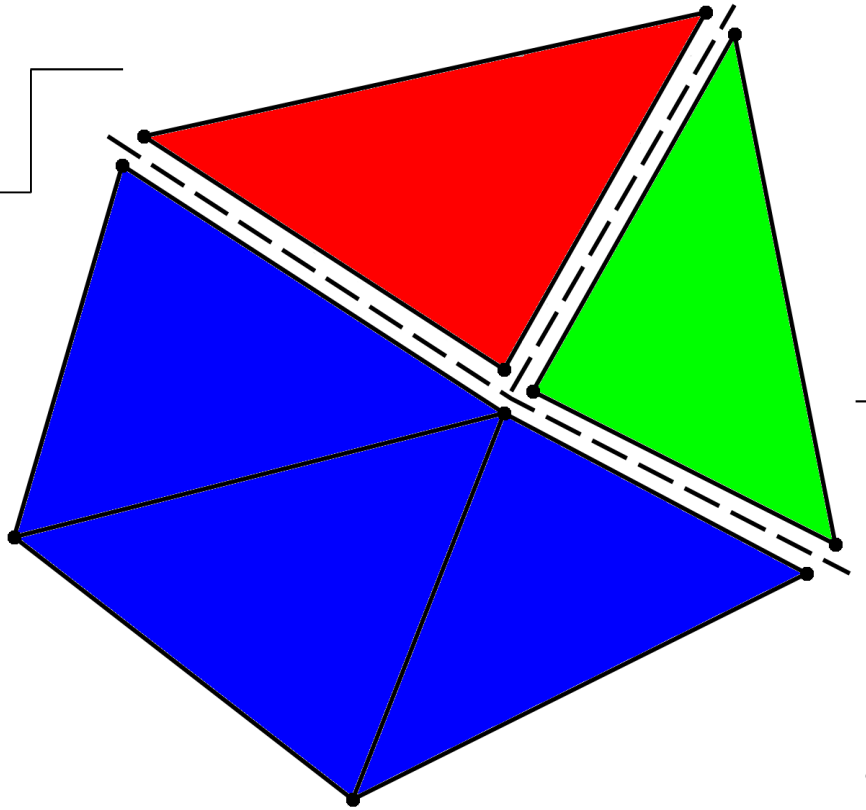


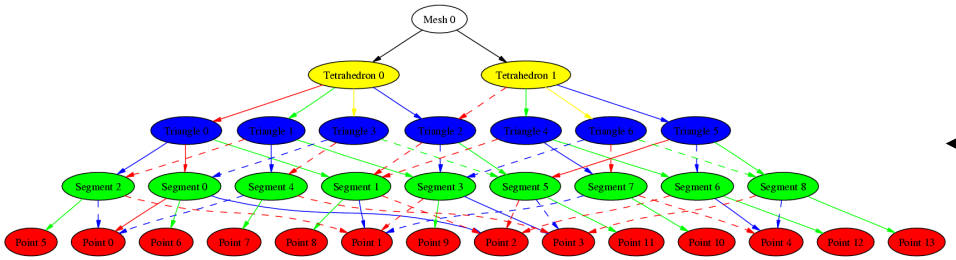
Sandia National Laboratories

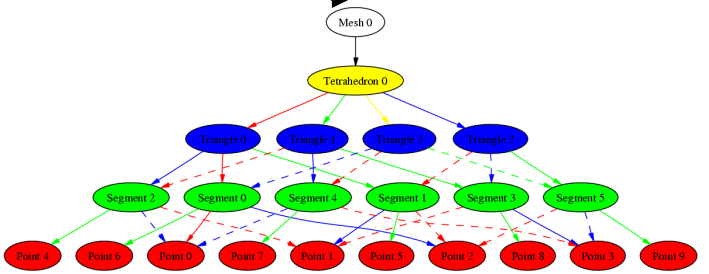
3D Graph Parallel Fracture

Partition mesh and create graph for each partition.



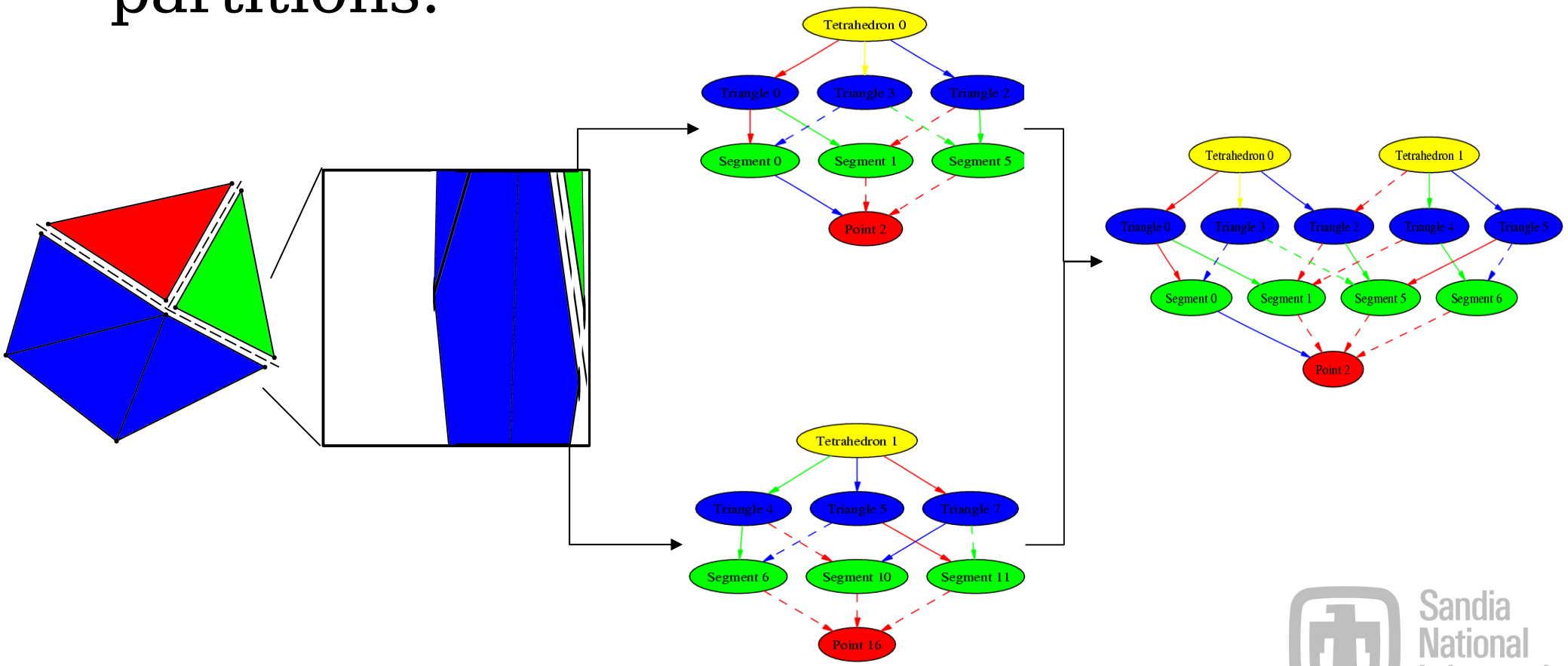






3D Graph Parallel Fracture

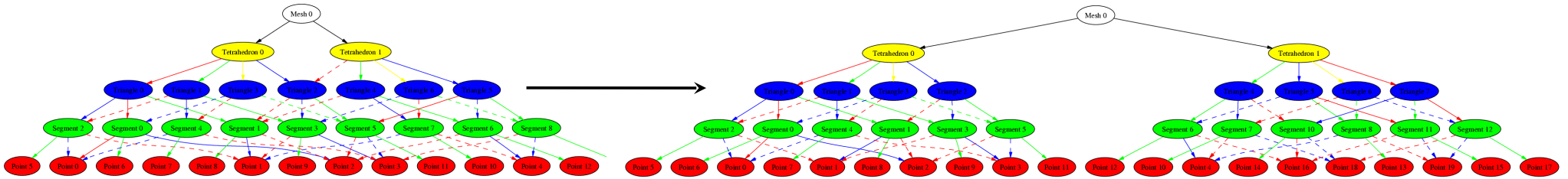
- On each fracture step mark open simplices.
- Partition-boundary points are missing parts of their subgraphs. Supplement from remote partitions.

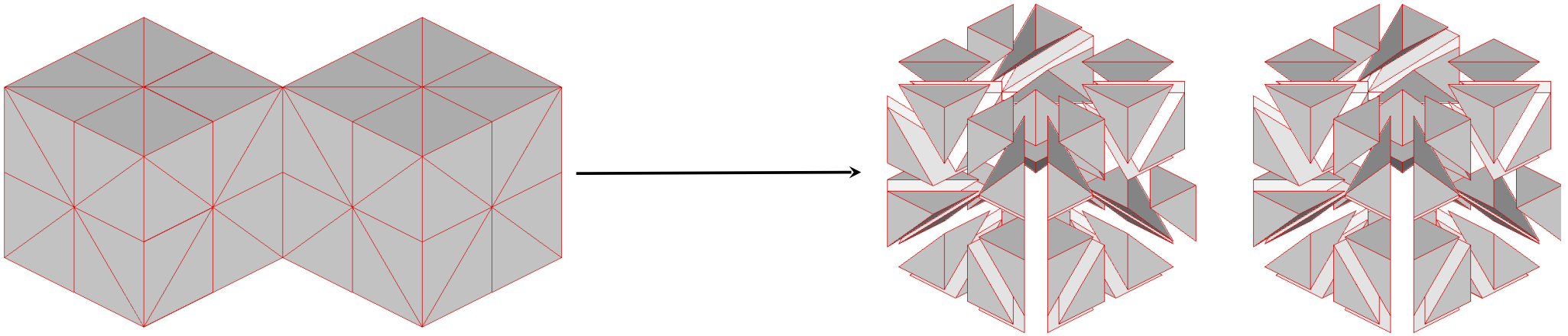


Sandia National Laboratories

3D Graph Parallel Fracture

Apply serial fracture algorithm to each partition.

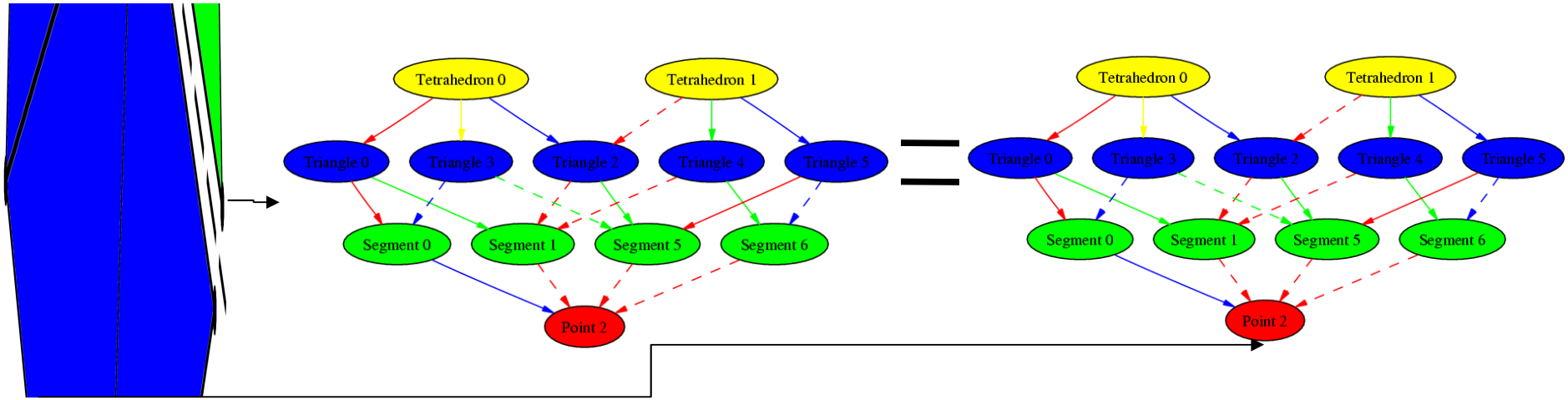




Sandia National Laboratories

3D Graph Parallel Fracture

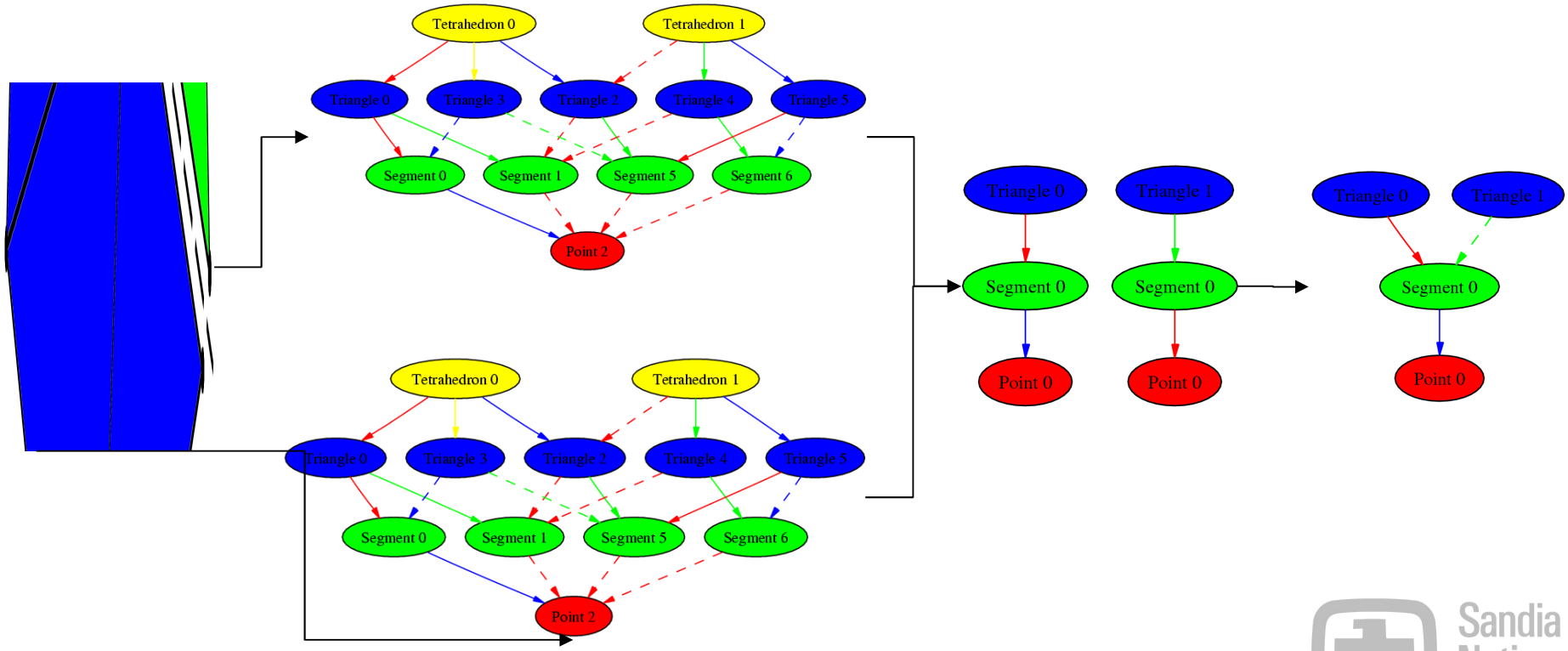
Subgraphs for partition-boundary points are in a consistent state across partition boundaries after serial fracture on each partition.



Sandia National Laboratories

3D Graph Parallel Fracture

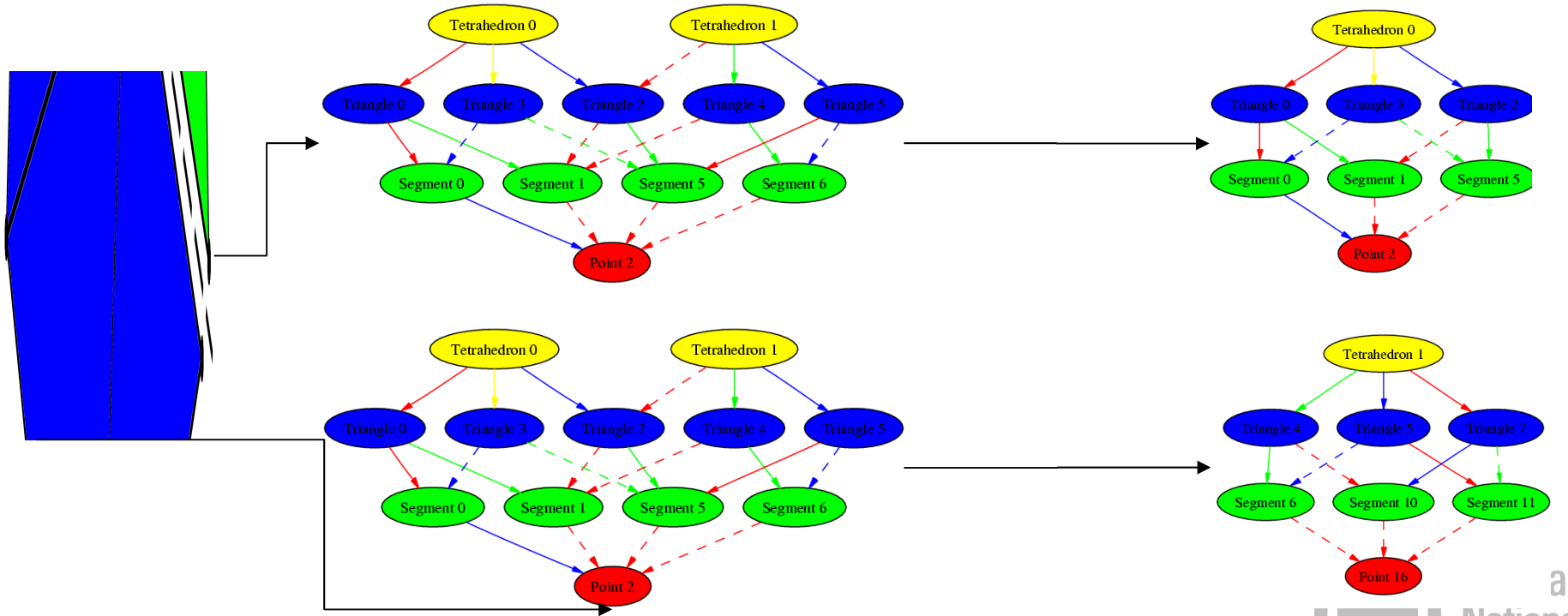
Exchange information between partition-boundary point subgraphs to identify and match newly created simplices across partition boundaries.



Sandia National Laboratories

3D Graph Parallel Fracture

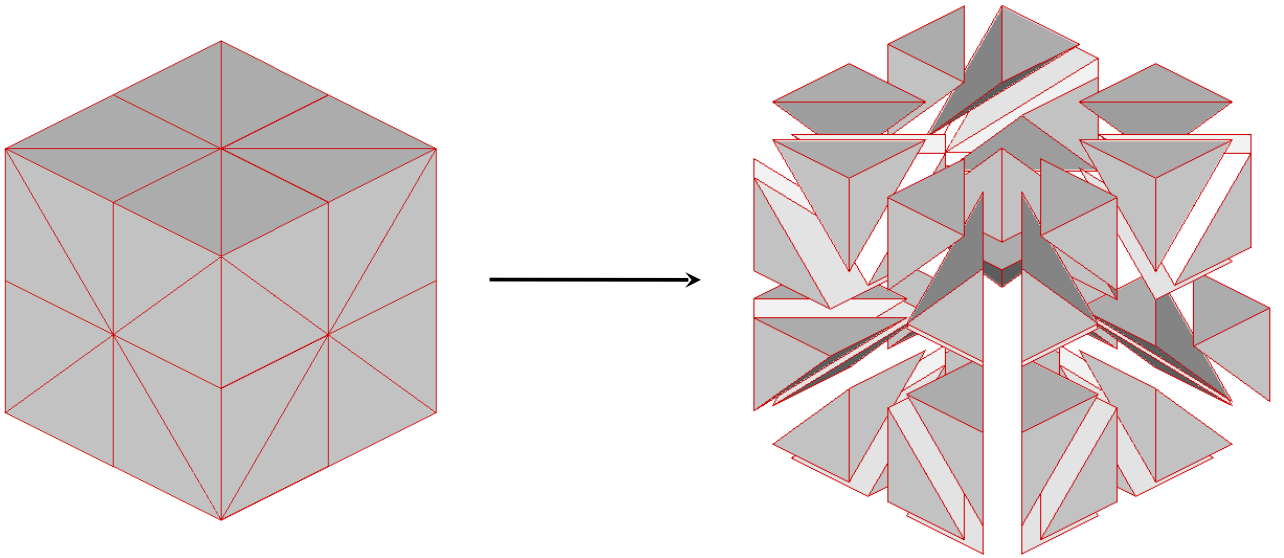
- Update boundary communicator.
- Discard the remote part of the graph for each partition and proceed with mechanics until next fracture step.



a National Laboratories

Testing

- Test on simple meshes yield the same results as serial fragmentation.
- The number of partitions is incremented until the partition algorithm (METIS) produces partitions with zero elements.

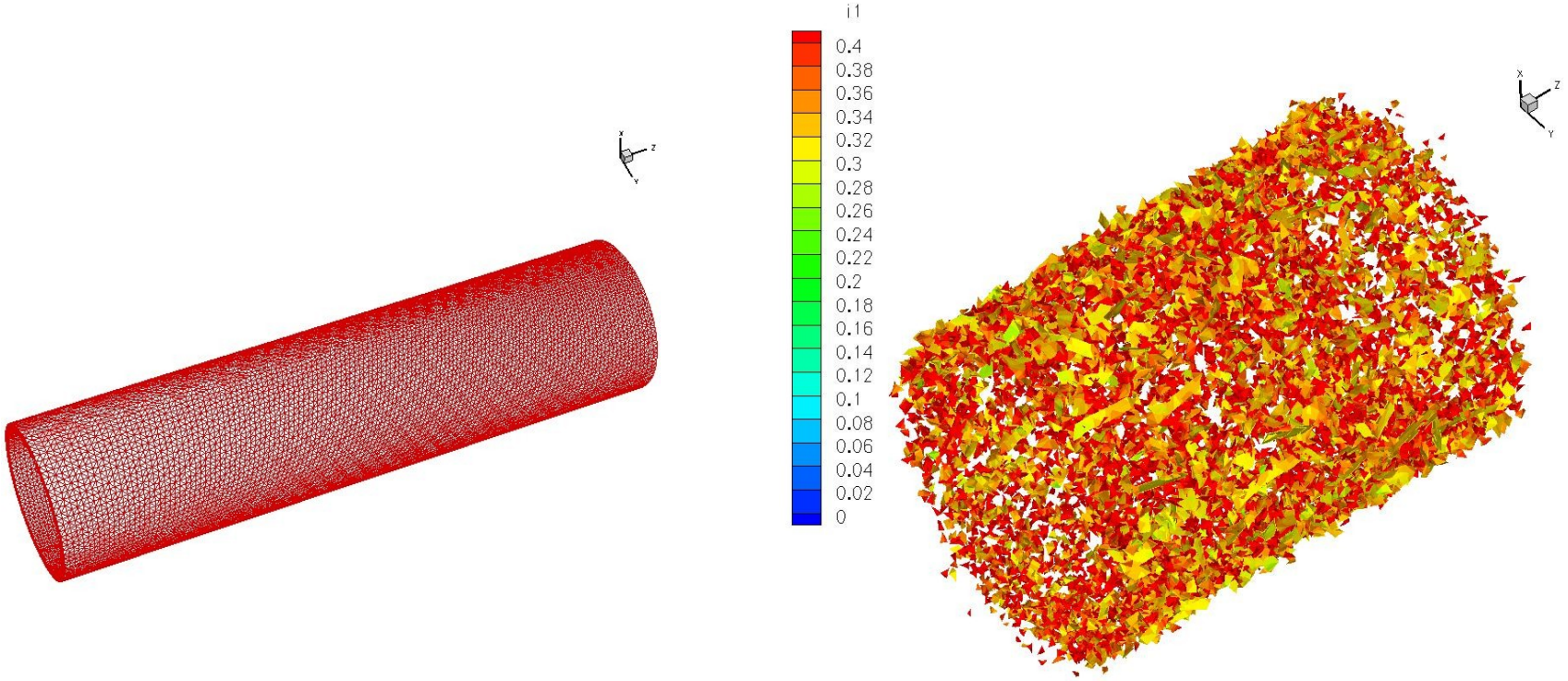


Cube example
2 to 14 partitions

Sandia National Laboratories

Testing

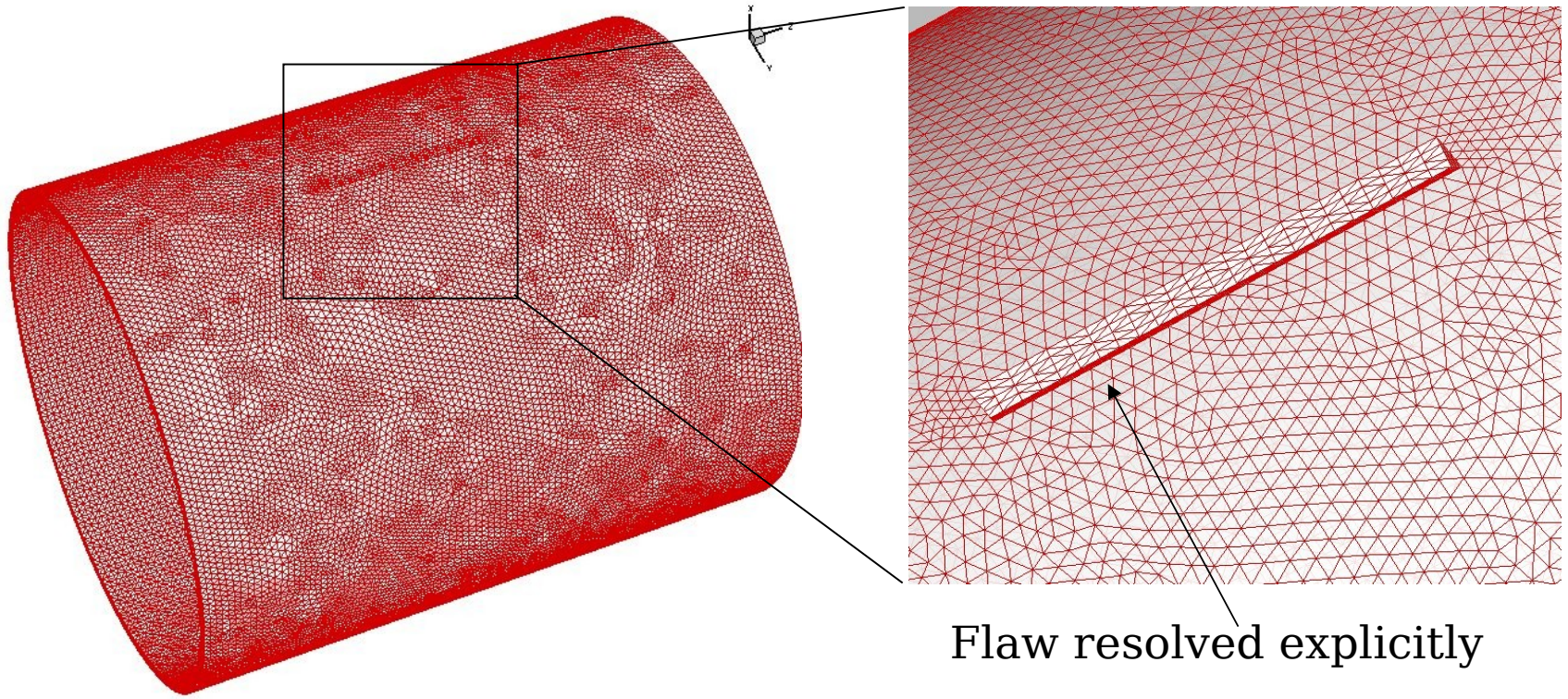
- Test on simple meshes yield the same results as serial fragmentation.



Tube example
2 to 8 partitions

Sandia National Laboratories

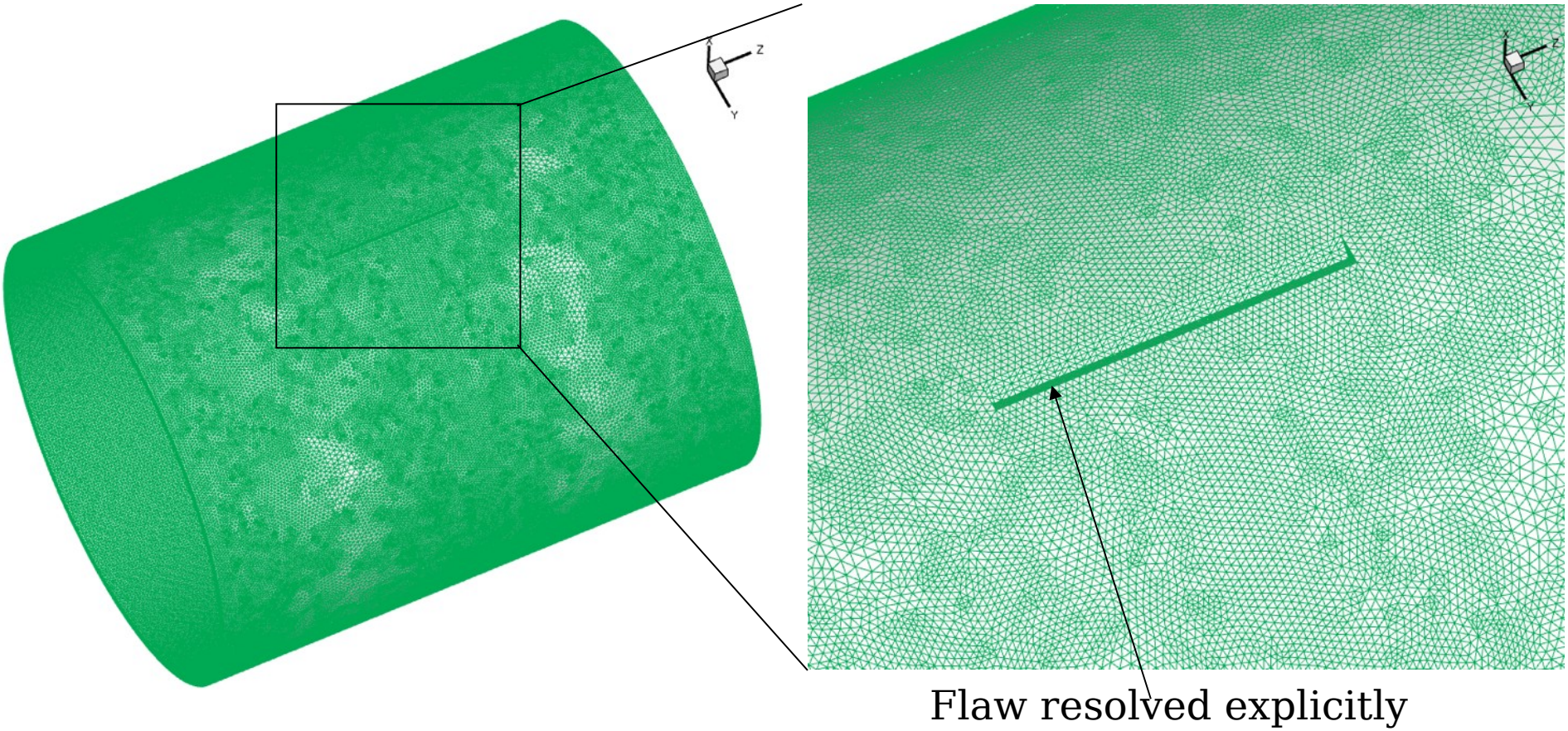
Coarse FE Mesh - Serial



Material model: Porous plasticity informed by QC
Interface model: Strain localization element
Failure criterion: critical plastic strain ~ 0.12
Mesh: 39150 elements, 78850 nodes
Machine: Single-processor Linux PC, ~1 week run time

Sandia National Laboratories

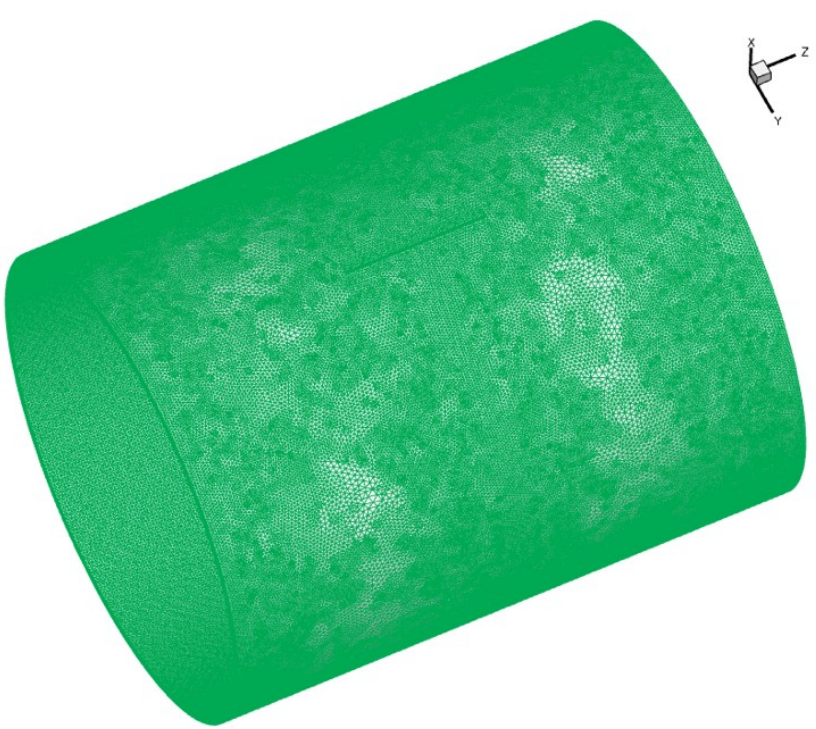
Fine FE Mesh - Parallel

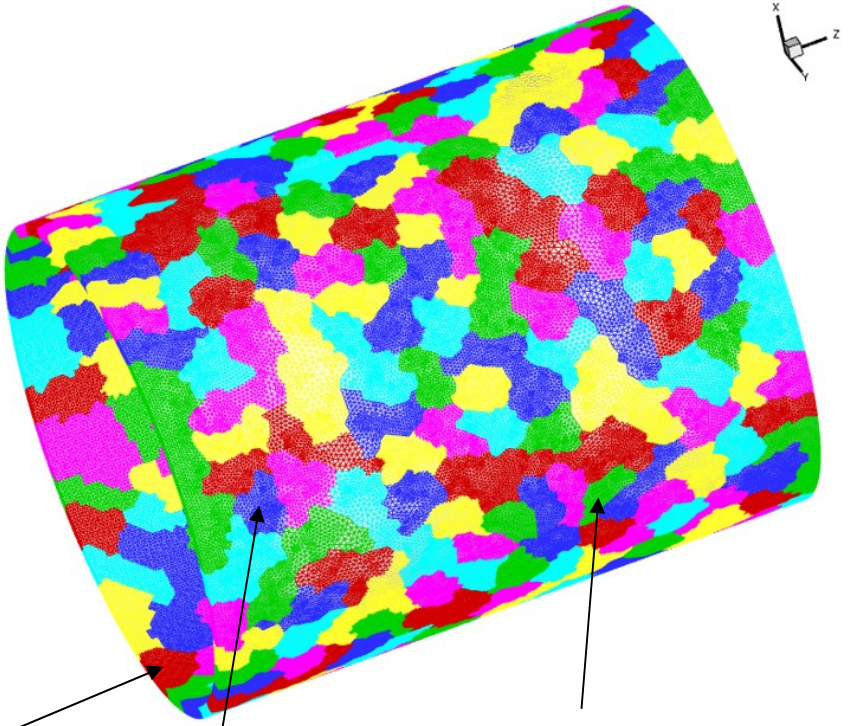


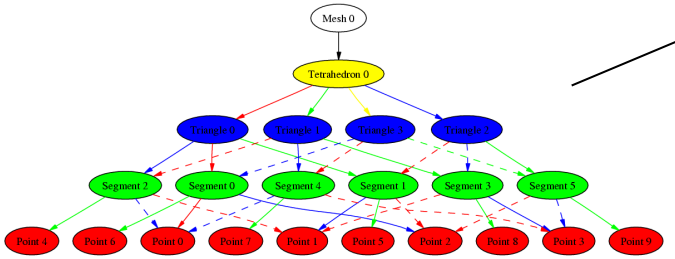
Material model: Porous plasticity informed by QC
Interface model: Strain localization element
Failure criterion: critical plastic strain ~ 0.12
Mesh: 220945 elements, 399998 nodes
Machine: LLNL ALC, 400 processors, 8 hours run time

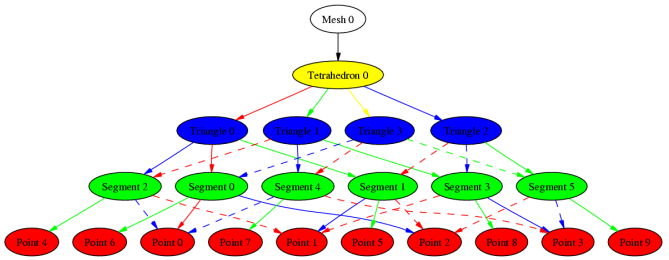
Sandia National Laboratories

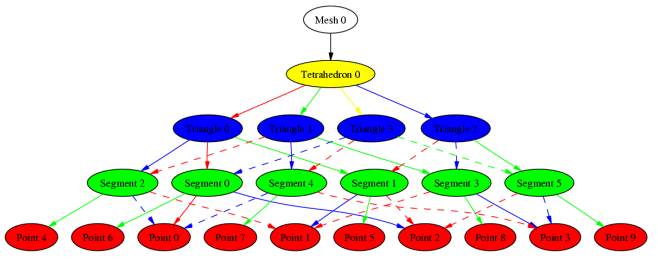
3D Graph Parallel Fracture







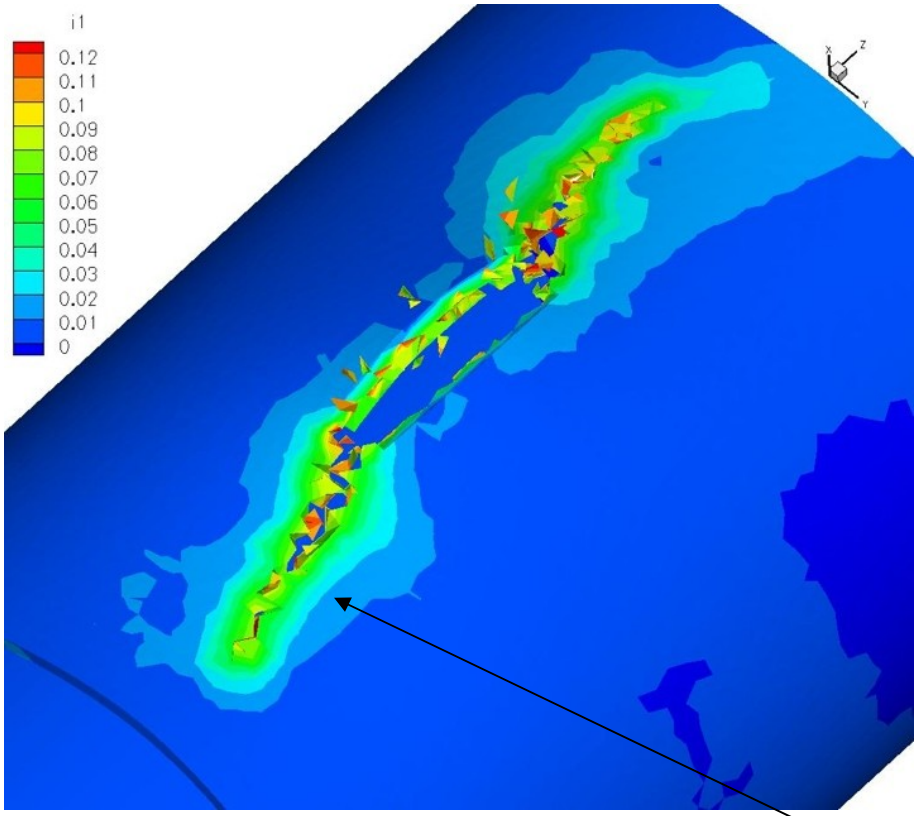


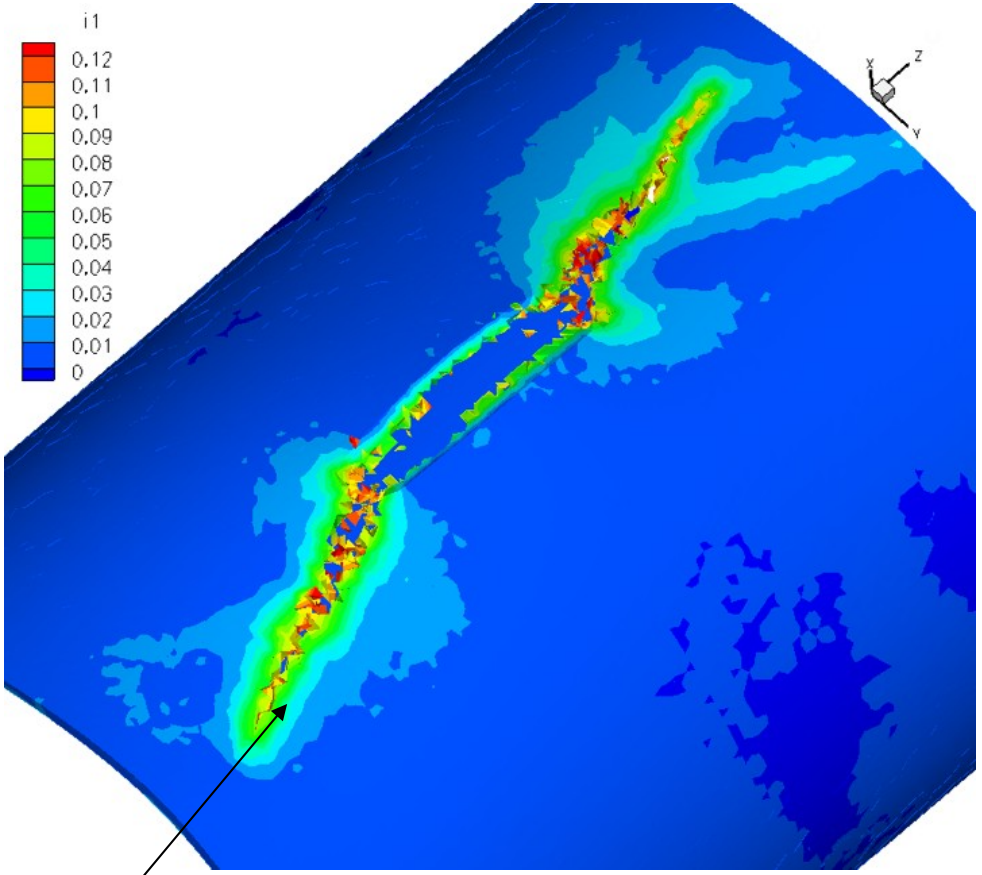


Partition mesh and create graph for each partition.

Sandia National Laboratories

Results: Coarse, Fine Meshes





Coarse Mesh - Serial

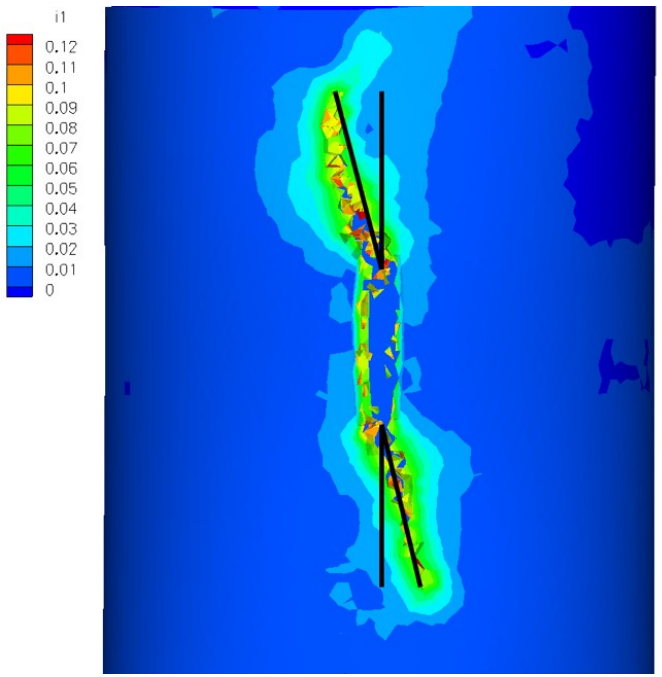
Kinked cracks

Fine Mesh - Parallel

Configuration at 75 μs
with initial torsion

Sandia National Laboratories

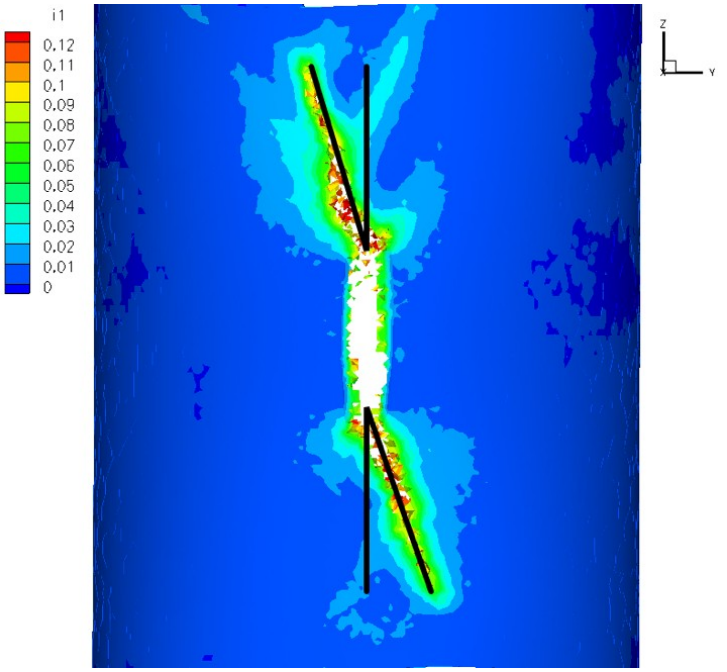
Crack Angles



Serial simulation
Peak pressure: 6.1 MPa
Crack kink angle ~ 14 degrees

Peak pressure [MPa]	Crack angle [degrees]
6.1	7
5.1	13
4.1	38
3.3	50

Experimental results
(Chao and Shepherd, 2004)



Parallel simulation
Peak pressure: 6.1 MPa
Crack kink angle ~ 16 degrees

Sandia National Laboratories

Conclusions

- Graphs allow fracture by simple, repetitive operations.
- The use of the Boost library reduced the work load significantly.
- Localized fracture operations suitable for parallel implementation.
- Time complexity reduced practically from quadratic to linear.
- Non-manifold cases handled correctly.

Sandia National Laboratories

Future Work

- Extend graph representation for CW complexes.
- Run very large fracture problems.
- Reuse graph mesh representation for serial and parallel contact.
- Replace connectivity arrays and use the graph for all FE computations.
- Mesh subdivision.

Sandia National Laboratories