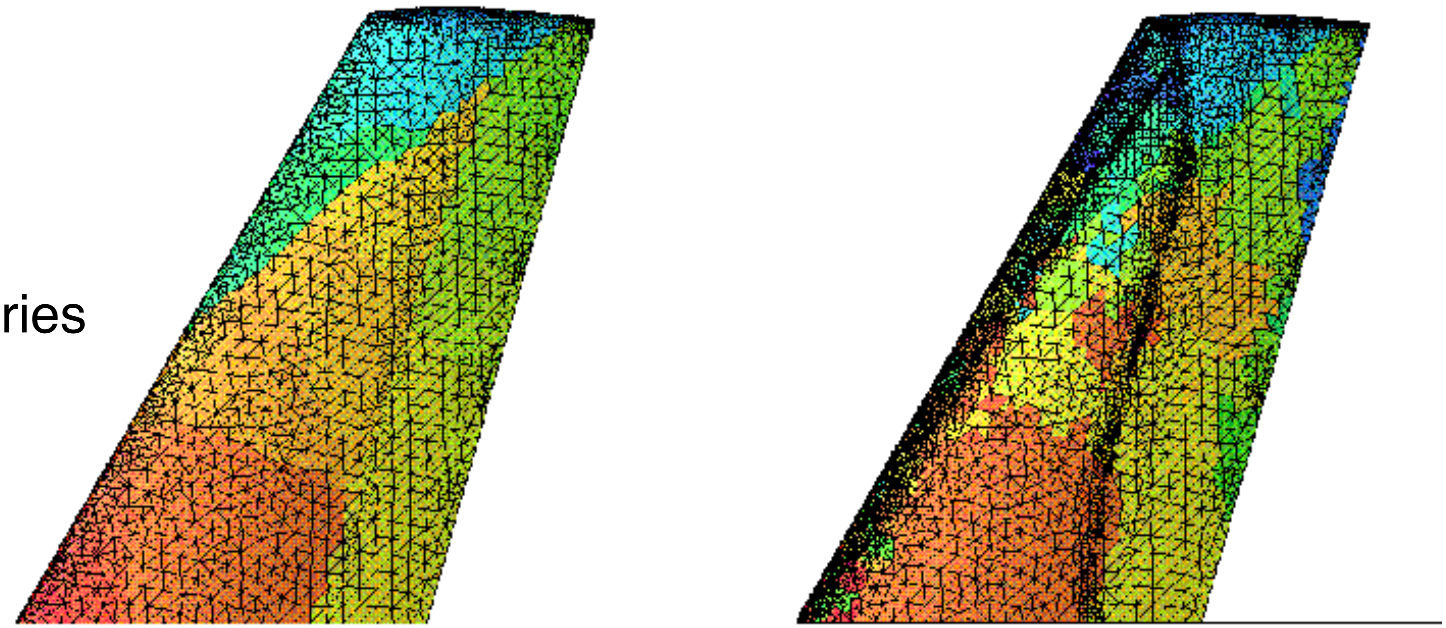
Parallel Mesh Adaptation

Mesh needs to be refined and coarsened during the adaptive analysis process

Mesh modifications in parallel on already partitioned mesh

Tools include:

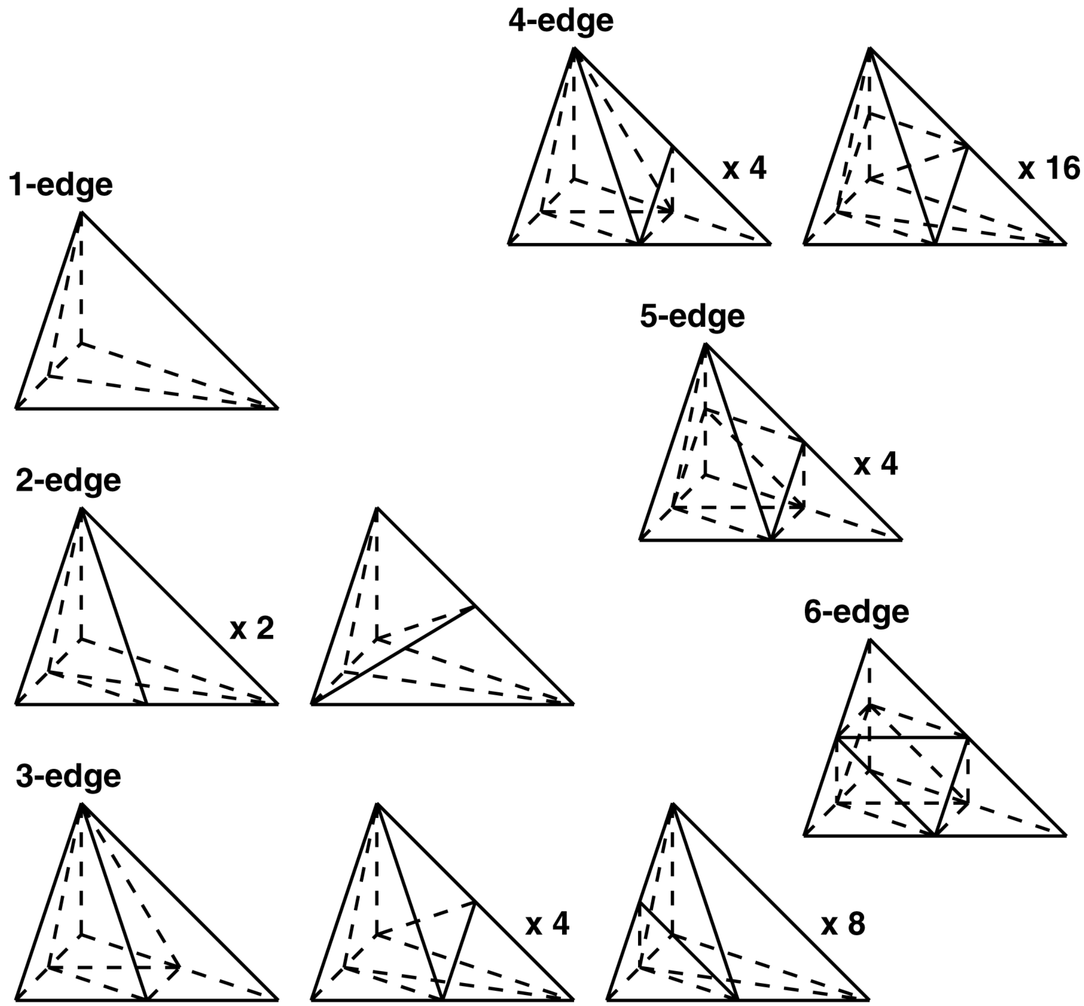
- Refinement
- Coarsening
- Swapping
- Snapping to curved boundaries

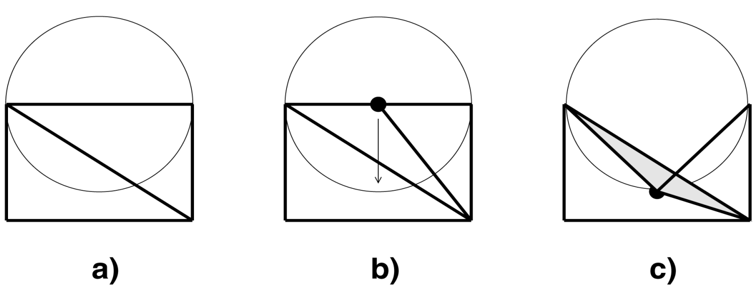


Mesh Refinement

Parallelization issues

- Diagonal selection when two edges marked on a face - template to ensure consistency across partition boundary
- Snapping to boundary for curved domains - can require swaps, etc.
- Rest of subdivision parallelizes easily
- Synchronize the partition data when done



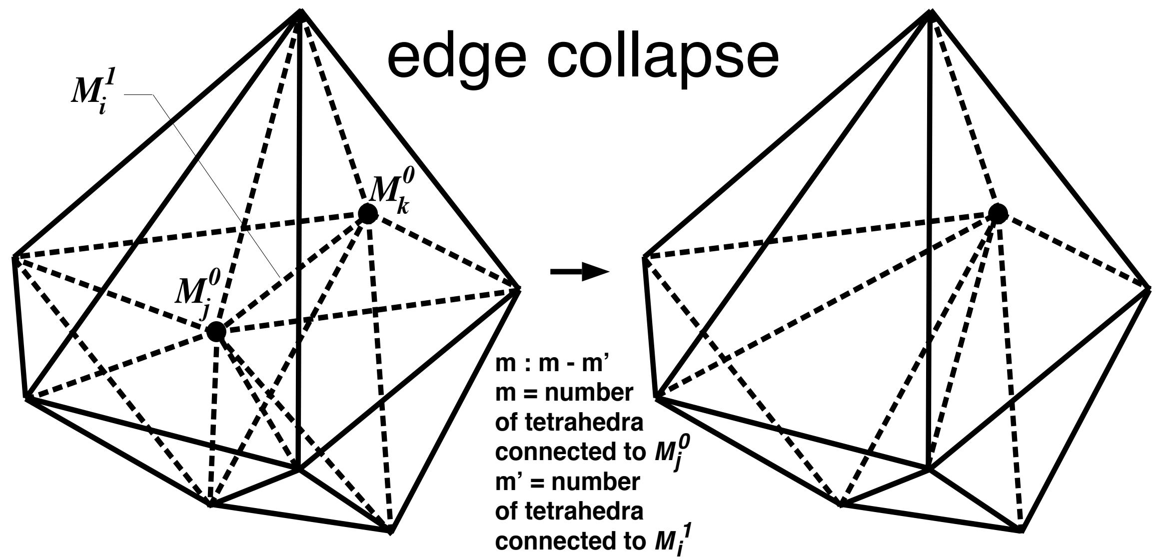


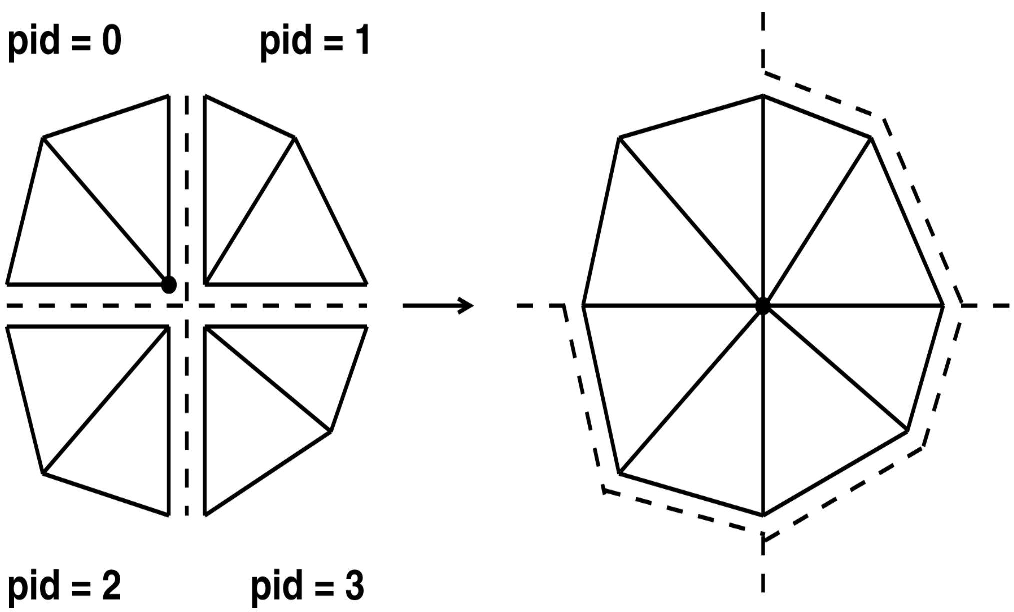
Mesh Coarsening and Swapping

Parallelization

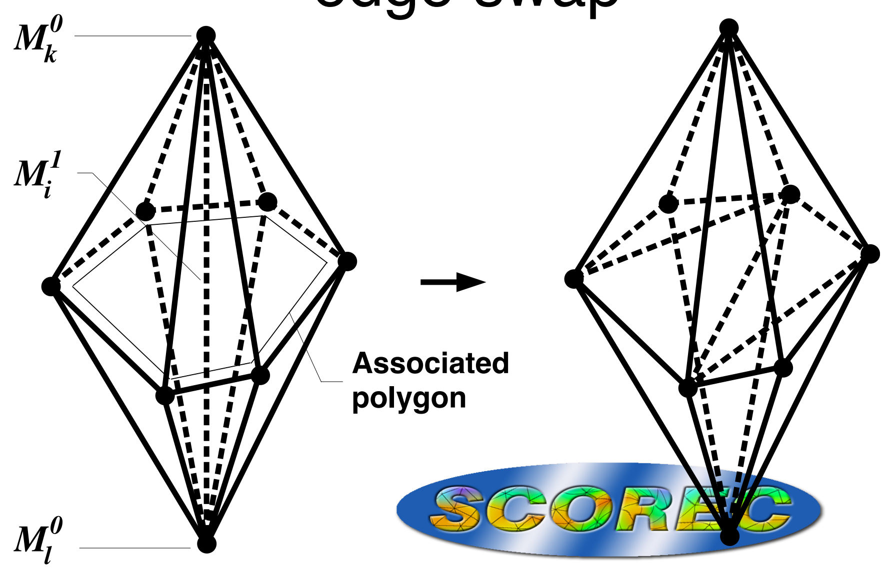
- On processor - performed
- Off processor - held
- Migration to get operation on processor

edge collapse



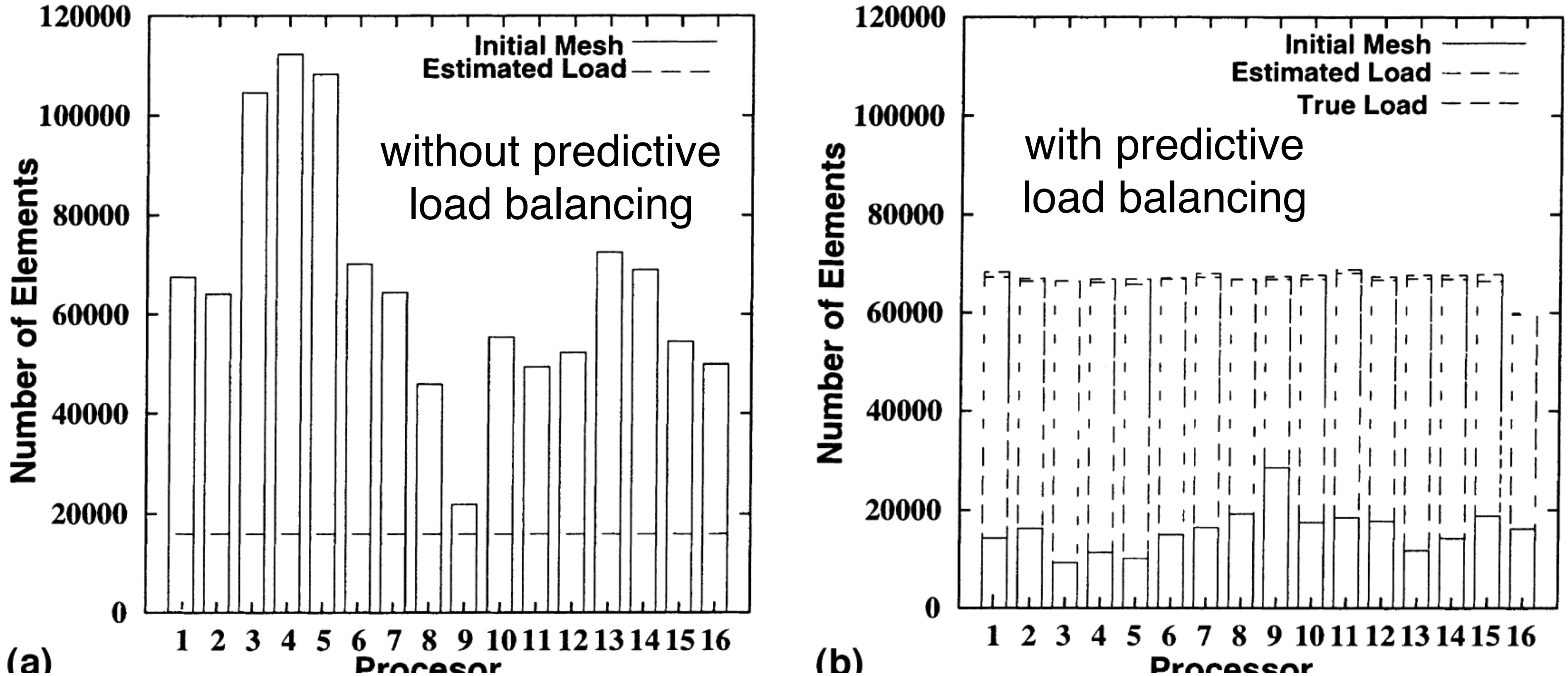


edge swap

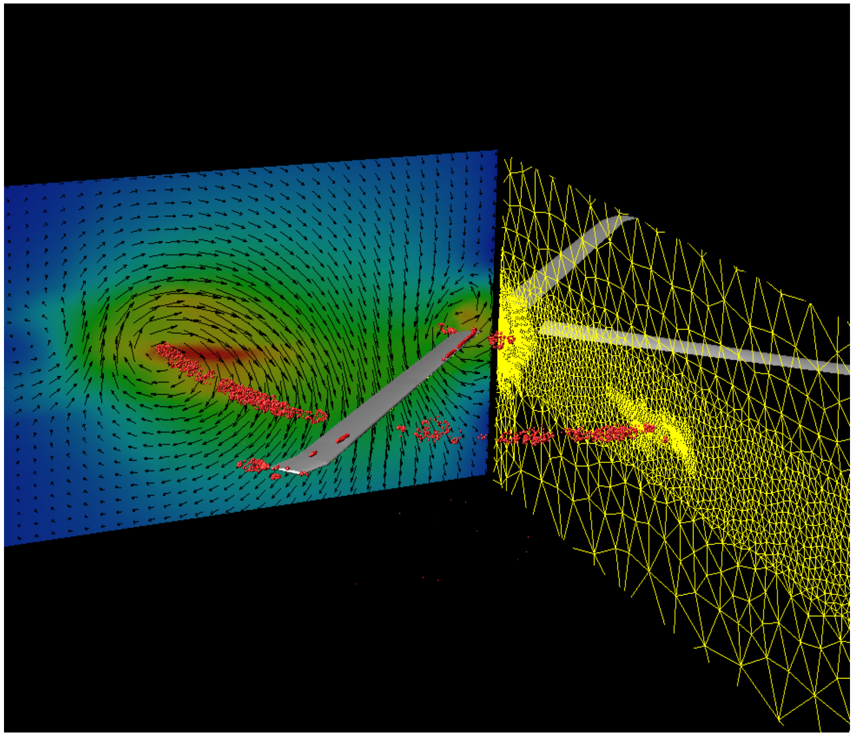


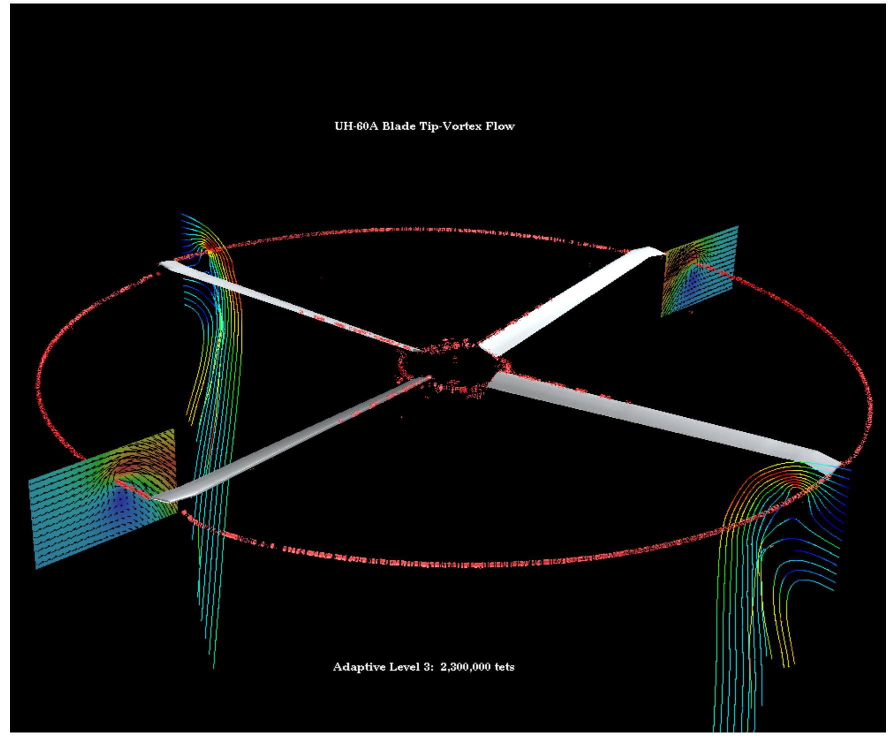
Predictive Load Balancing

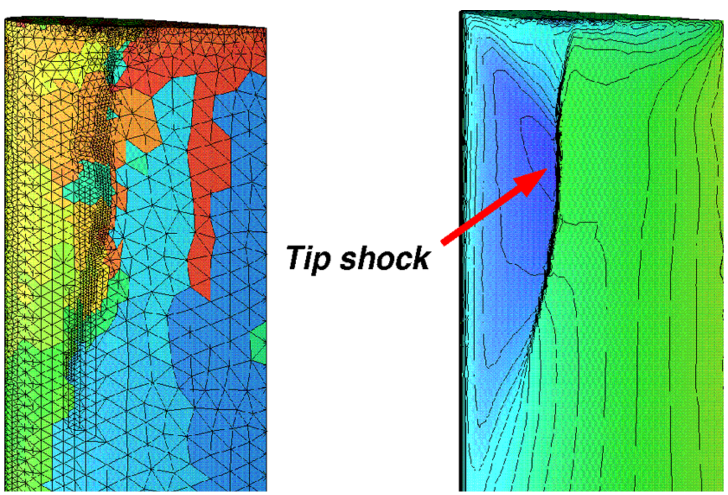
- Refinement of mesh before load balancing can lead to memory problems
- Employ predictive load balancing to avoid the problem
  - Assign weights based on what will be refined
  - Apply dynamic load balancing
  - Refinement
  - May want to do some local migration



Parallel Automated Adaptive Analysis







additional flow examples

Parallel Mesh Generation

All mesh generation steps operate in parallel

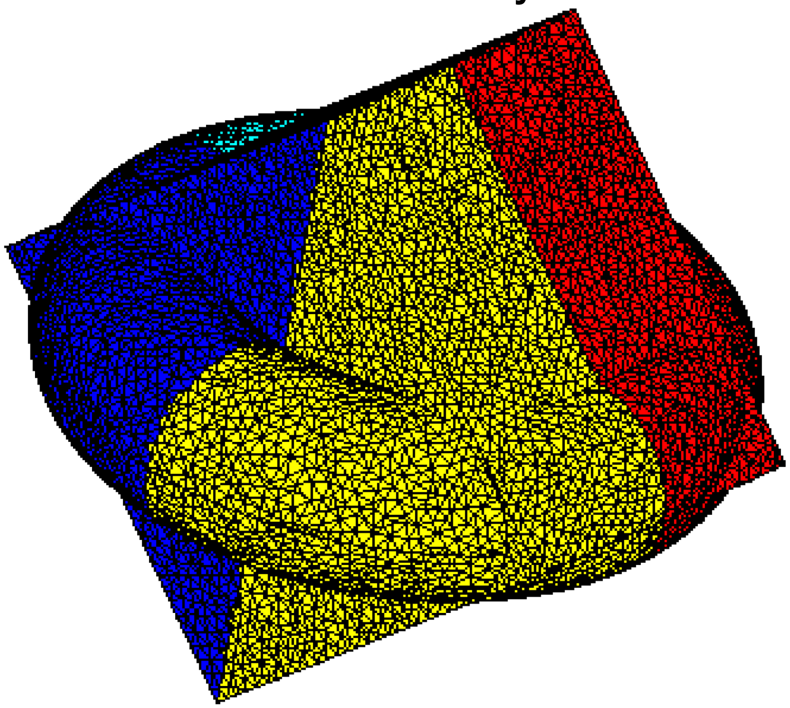
Meshes starting from solid model

Both structures created by the mesh generator are distributed

- Octree - used for mesh control, localizing searches, interior templates
- Mesh - topological hierarchy distributed and controlled by RPM

Mesh generation steps

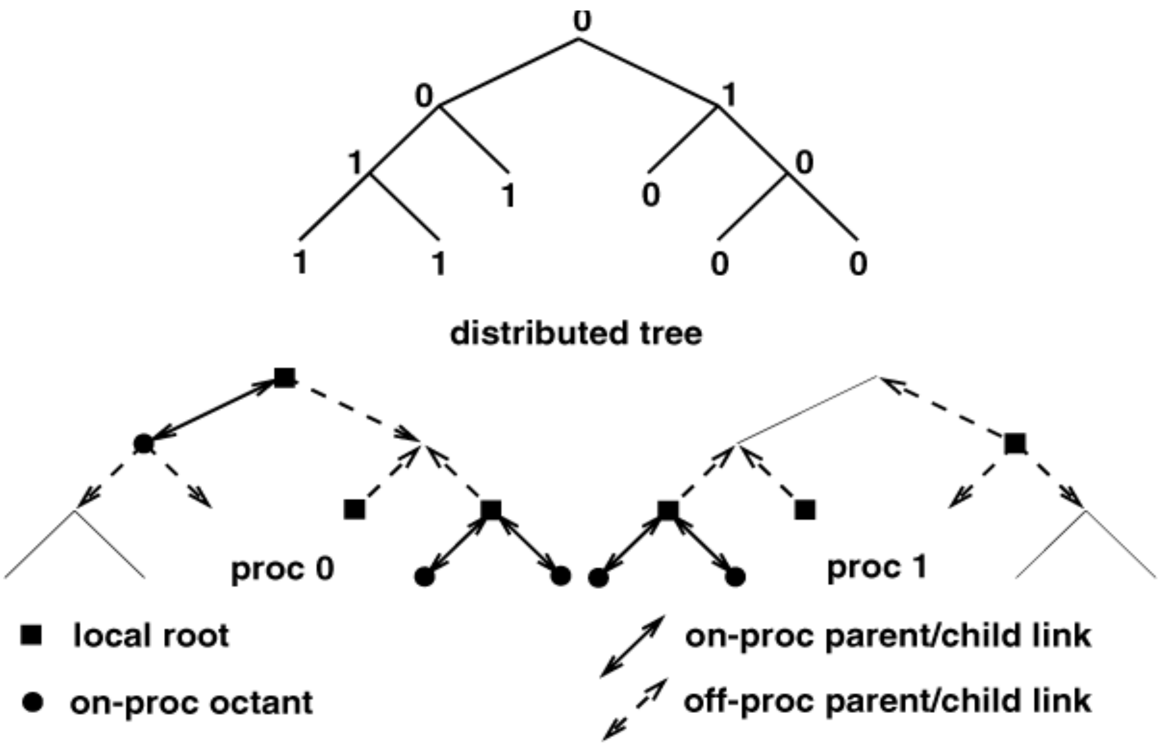
- Surface mesh generation
- Octree refinement
- Template meshing of interior octants
- Meshing boundary octants

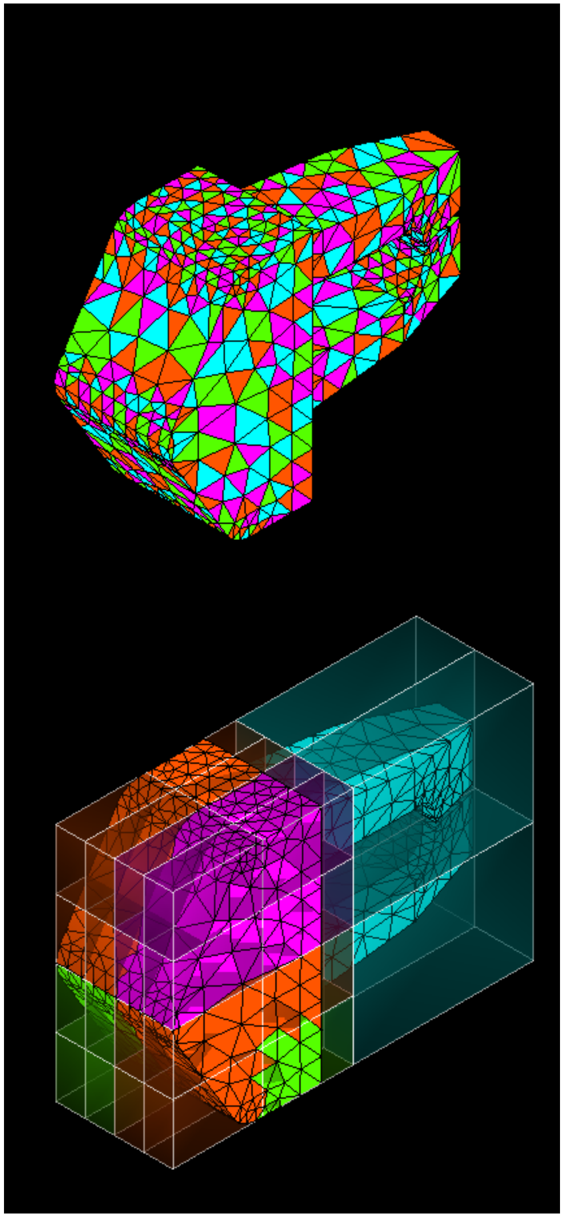


Parallel Distributed Octree

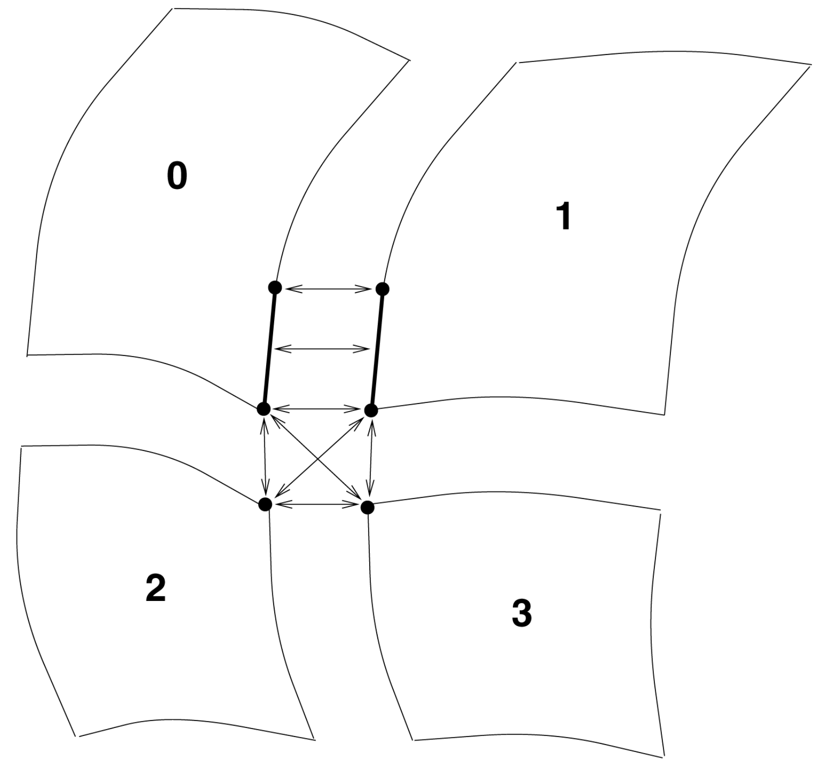
Octree structure distributed to processors

- Parents can point of off processor children
- Local roots and local root list
- Pointers to equal or larger face neighbors - with tree built maintaining one level difference this avoids tree traversal - O(1) to find neighbor
- Mesh related to the tree





Surface Mesh Generation

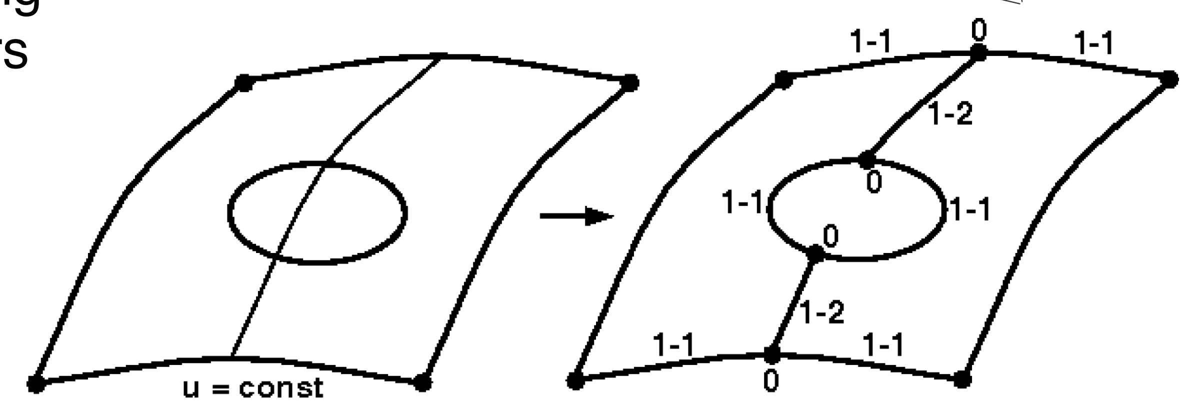


Key features

- Delaunay-type insertion on surfaces distributed to processors
- Faces can be split if needed to ensure scalability
- Boundary mesh entity links (edges and vertices) built

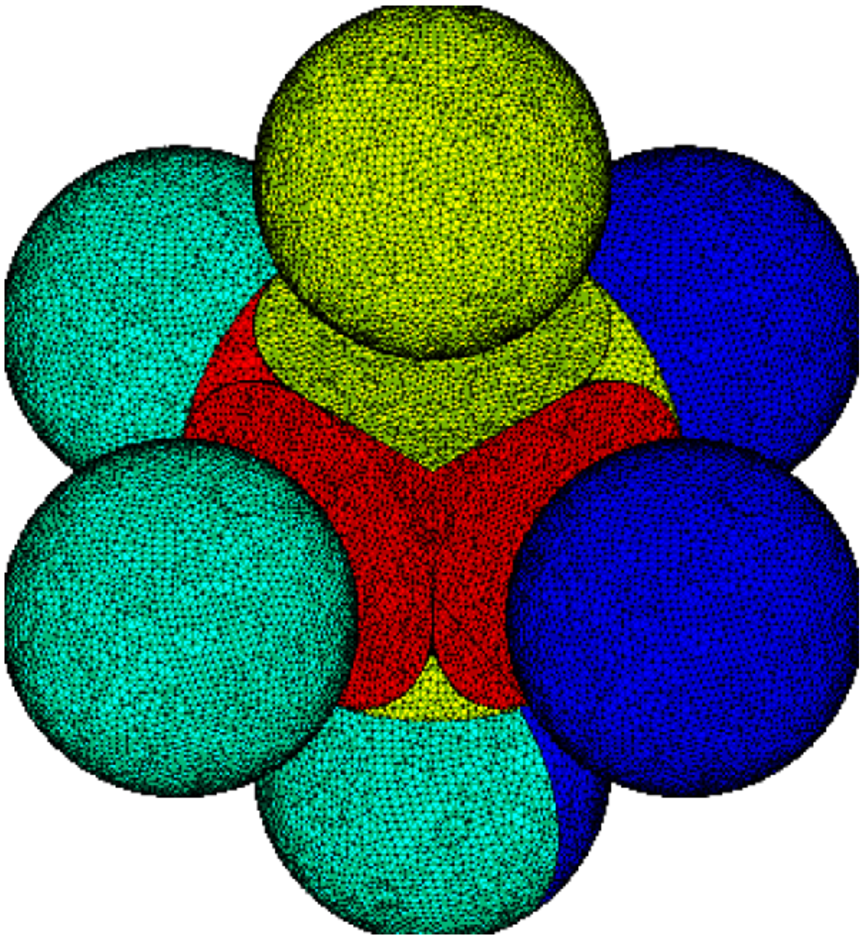
Reasonable speedups:

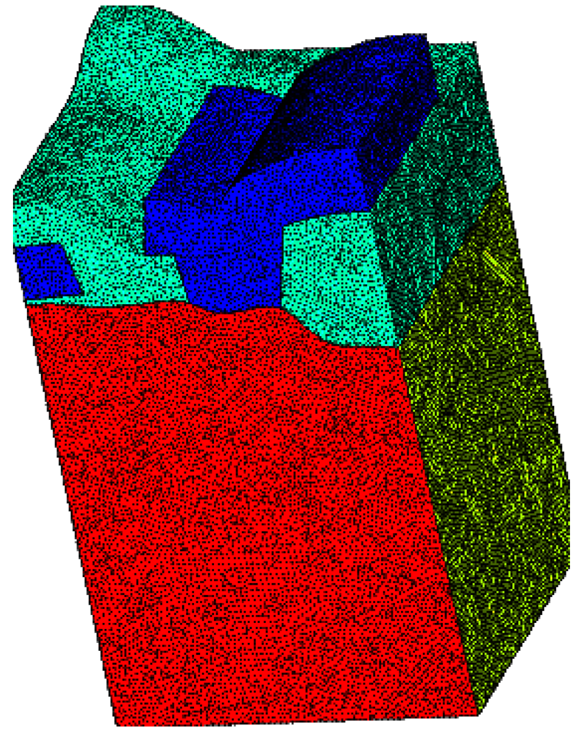
- Approx. 1.5 each doubling the number of processors
- Tests to 32 processors



Parallel Surface Mesh Generation

Examples

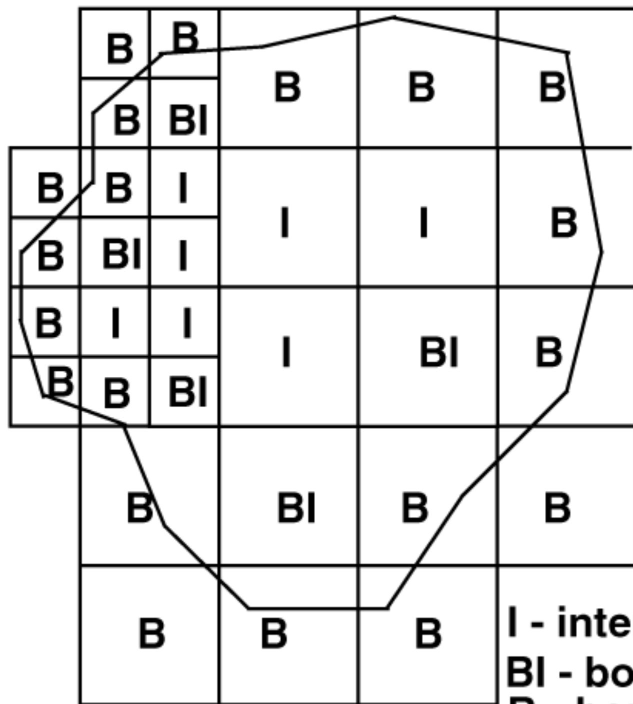


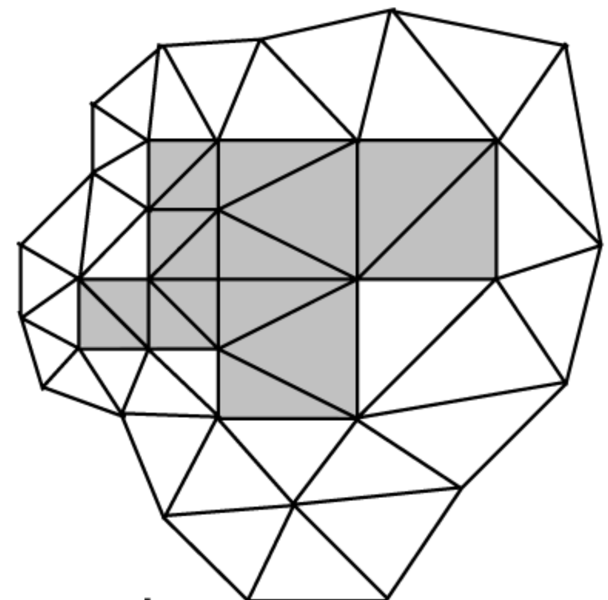


Parallel Volume Meshing

Given a distributed surface mesh, steps include:

- Build distributed tree (may be partly done)
- Classify octants
- Template meshing of interior octants
- Partition boundary octants
- Mesh on processor boundary regions
- Repartition to mesh partition boundaries
  - face
  - edge
  - vertex
- Repartition for next operation (an analysis step)





I - interior quadrant
BI - boundary like interior quadrant
B - boundary quadrant

Quadtree with boundary edges

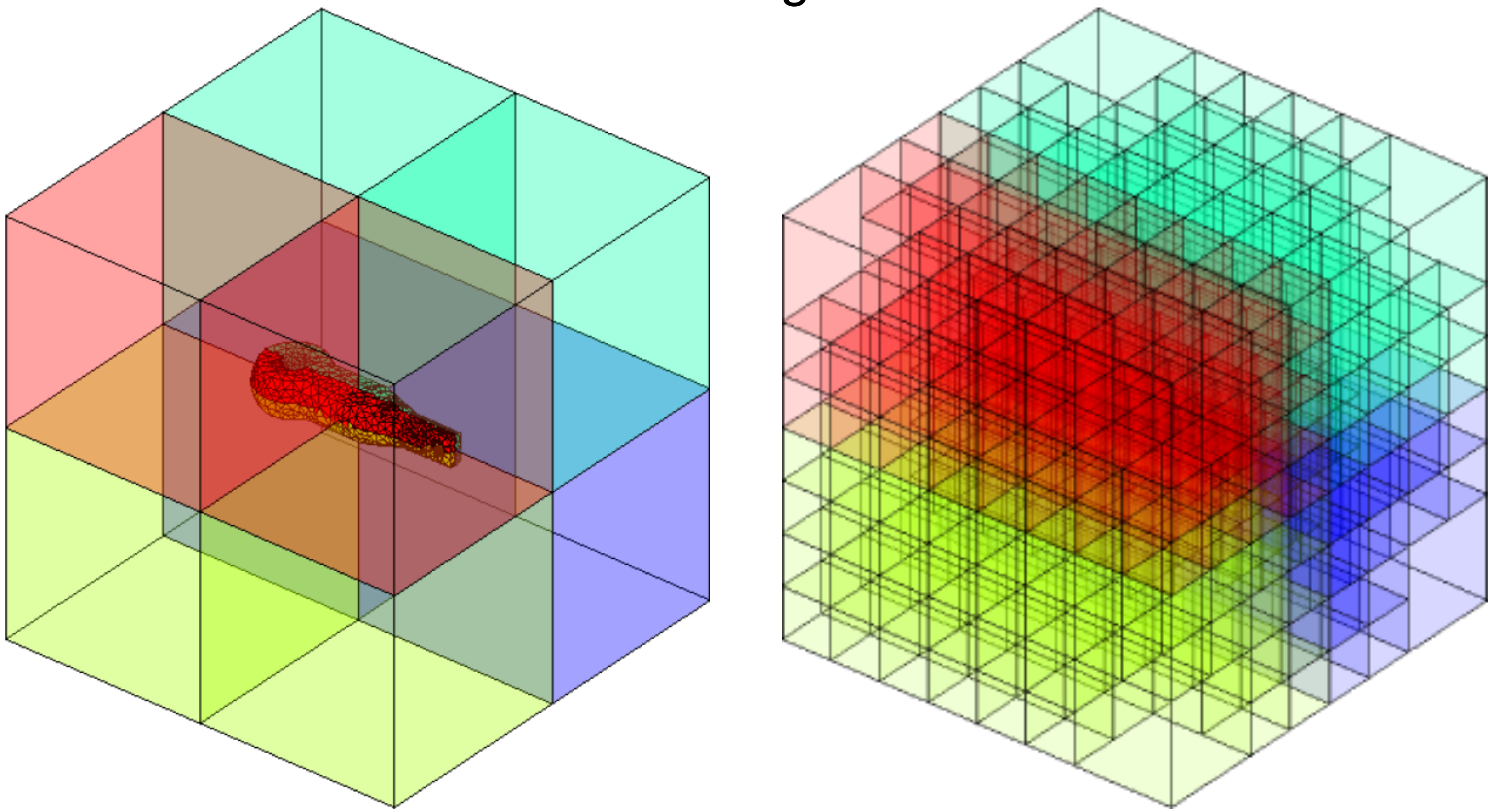
Unsmoothed mesh

Octree Building

Build from surface mesh and gradation control

Distributed from start

One level difference enforced during construction



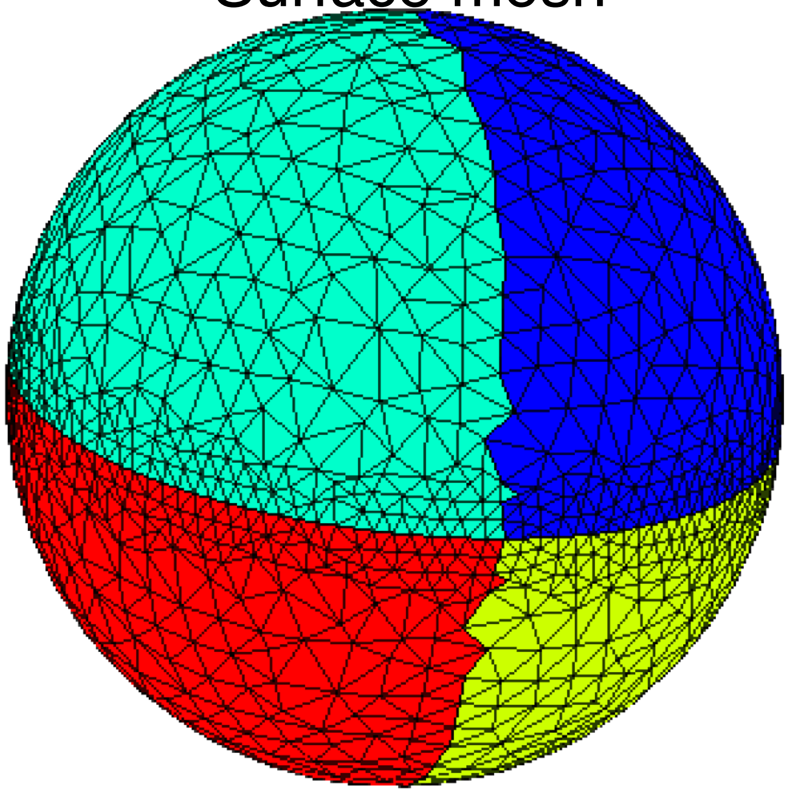
Template Meshing of Interior Octants

Templates ensure mesh matching even with one level difference

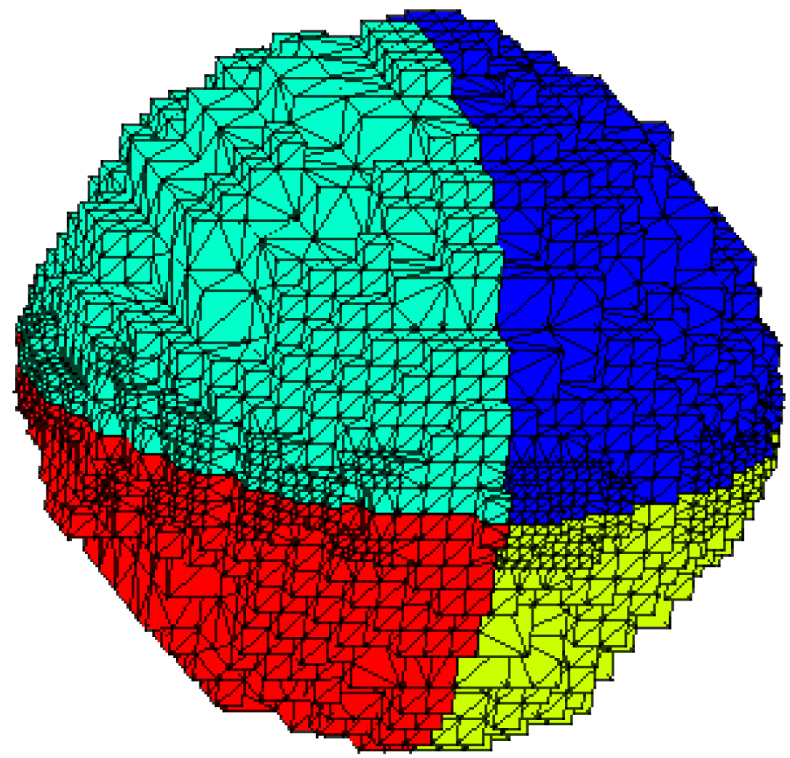
Interior octants done in parallel

One synchronization step to coordinate partition pointers

Surface mesh



Interior octant mesh



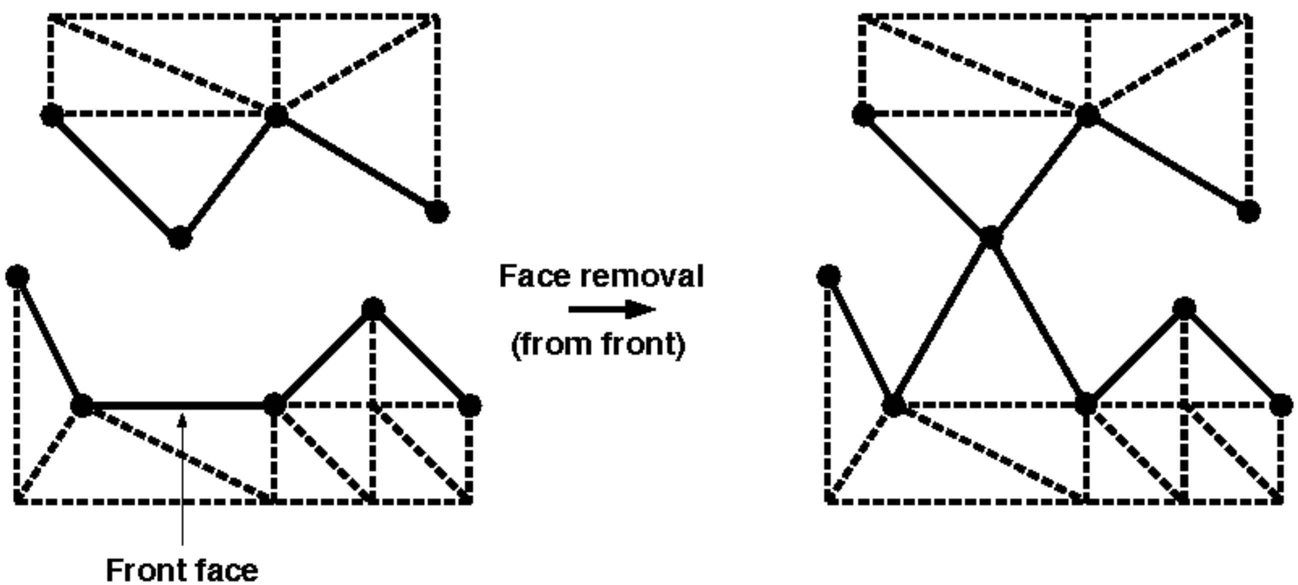
Boundary Octant Meshing

Face removal procedure used to connect surface mesh to interior octants

Candidate face neighbors obtained from octree

Candidates ordered based on normal heuristics

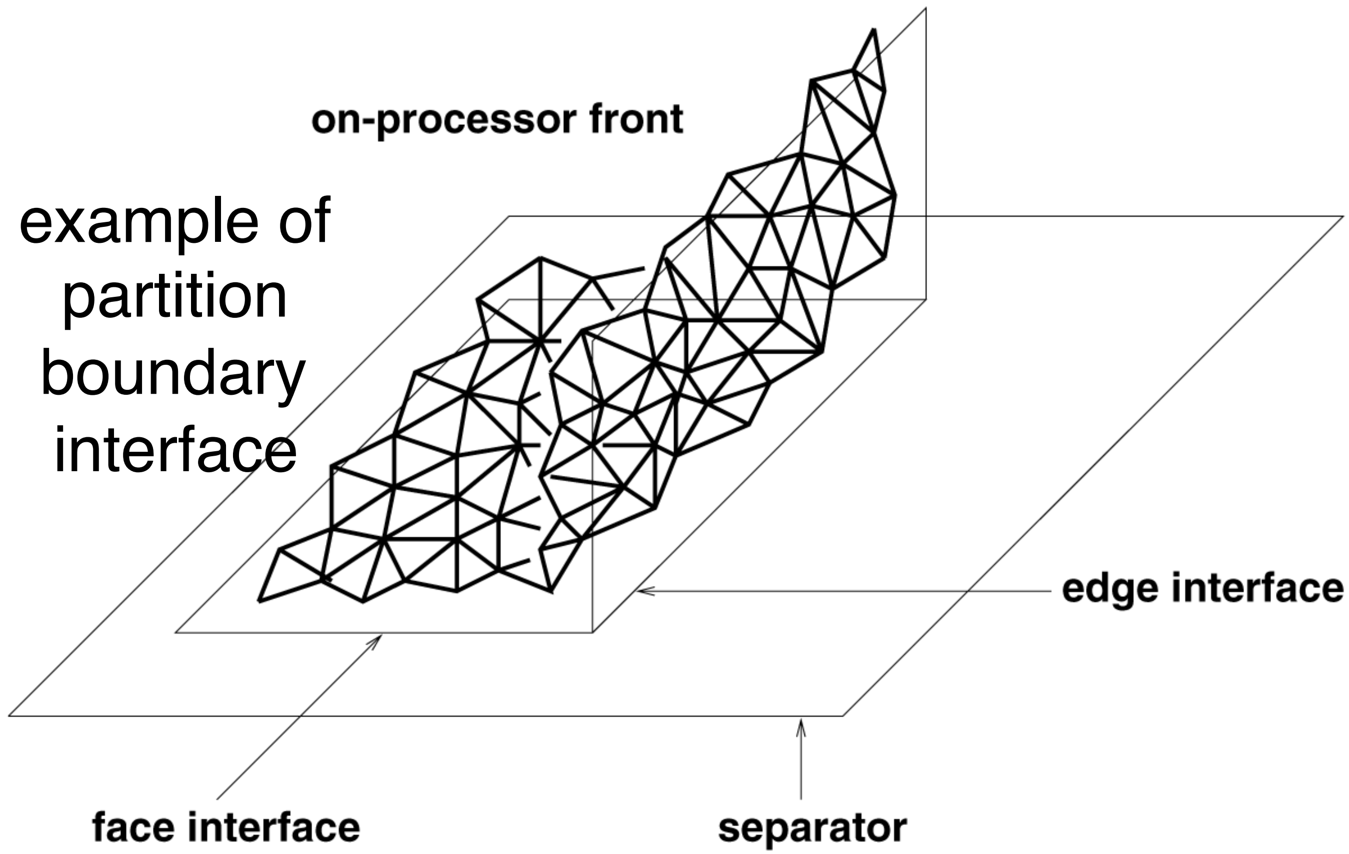
If neighbor information not on processor, delay operation



Boundary Octant Meshing

After first set of face removals the regions at interprocessor boundaries remain unmeshed

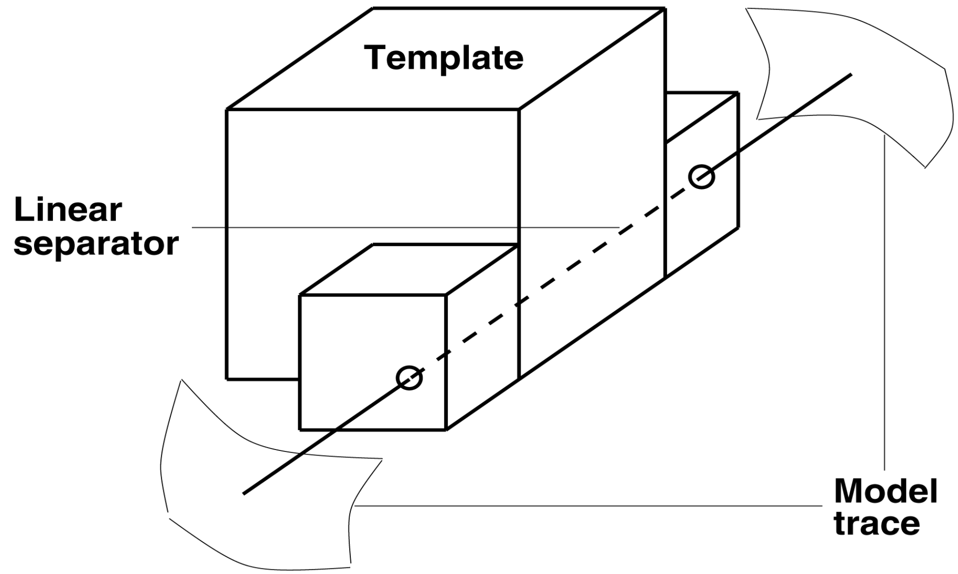
Can complete in parallel using three repartitions

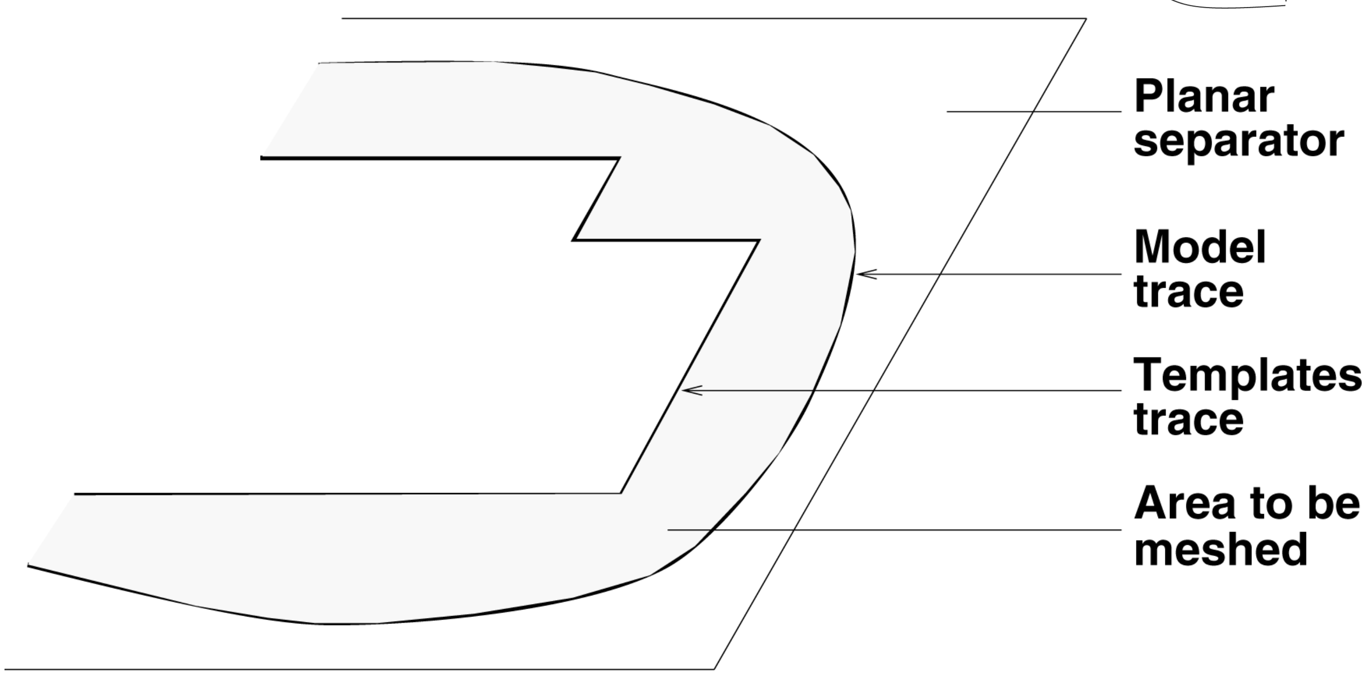


Boundary Octant Meshing

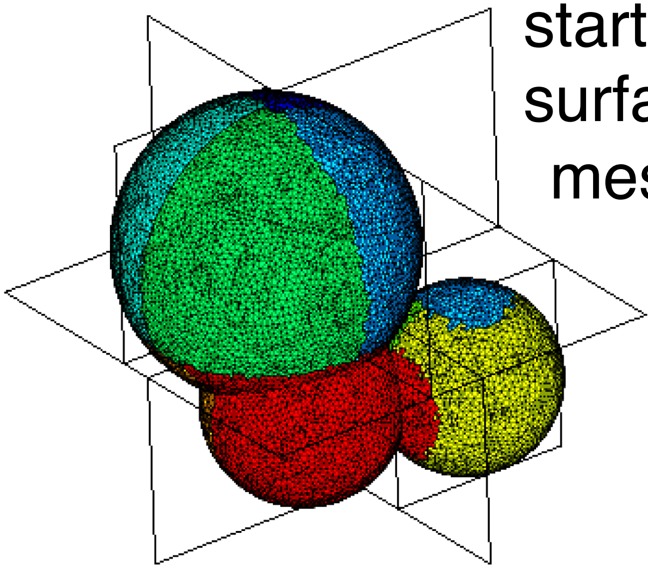
Template meshing procedure has already meshed the portions of the partition face edge and vertex interfaces interior to the model

Partition interfaces appear a dimension lower

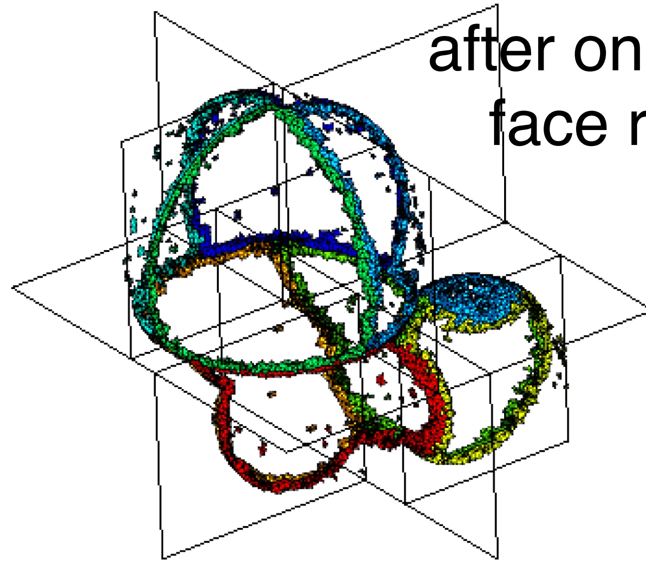




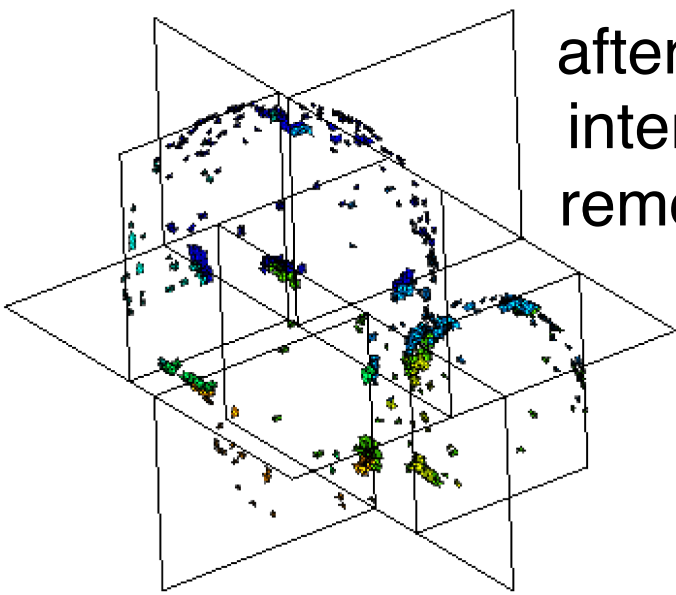
Boundary Octant Meshing Steps



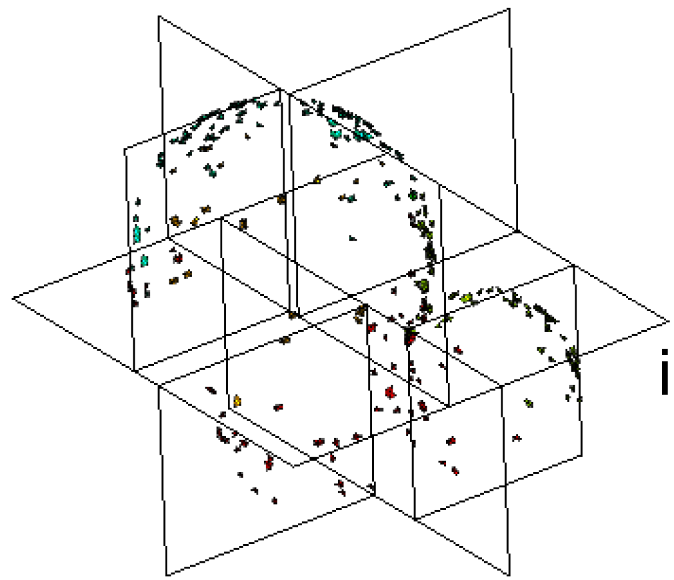
starting surface mesh



after on-processor face removals



after face interface removals



after edge interface removals - completed in next step