Challenges in Dynamically Evolving Meshes for Large-Scale Simulations

Presented by: Terry L. Wilmarth

Parallel Programming Laboratory
and
Center for Simulation of Advanced Rockets

The ParFUM Crew: Sayantan Chakravorty, Terry Wilmarth, Aaron Becker, Isaac Dooley, Gengbin Zheng, Celso Mendes, L. V. Kalé

Charm++ Workshop

18 April 2007

Overview

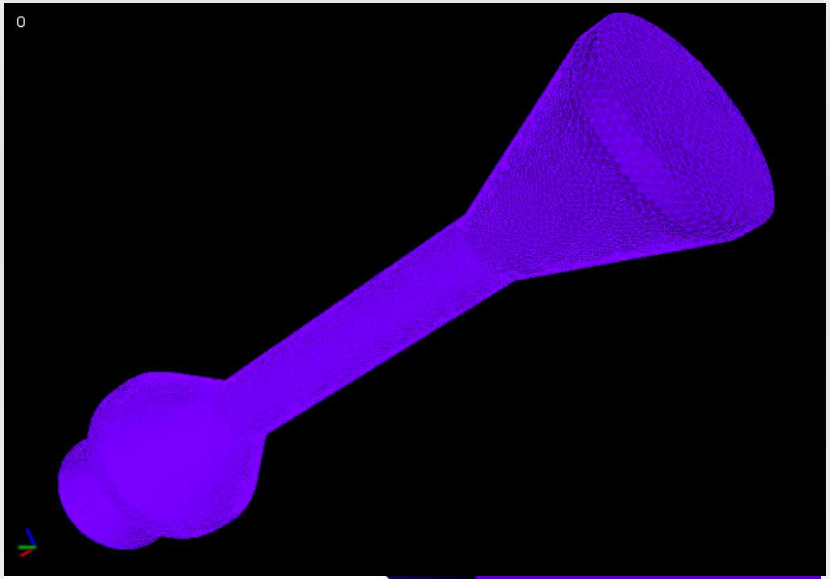
- Dynamic Meshing
  - Evolving Geometry
  - Mesh Adaptivity
  - Material changes
- ParFUM Approaches
  - Dynamic meshing
  - Load balancing
  - Solution Transfer
- ParFUM Status

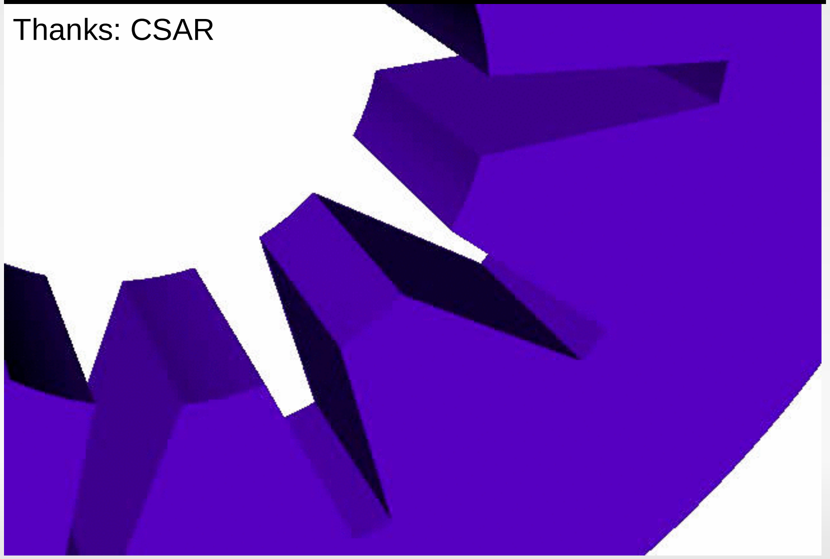
Dynamic Meshing

- Mesh-based simulations with dynamic, irregular behaviors:
  - Mesh geometry and/or topology changes over time
    - Mesh quality maintenance for evolving geometries
    - User-directed adaptivity for solution accuracy
  - Simulation behavior changes over time
    - Computational cost varies as material state changes
    - Simulation scale varies for regions and/or time periods

Evolving Geometry

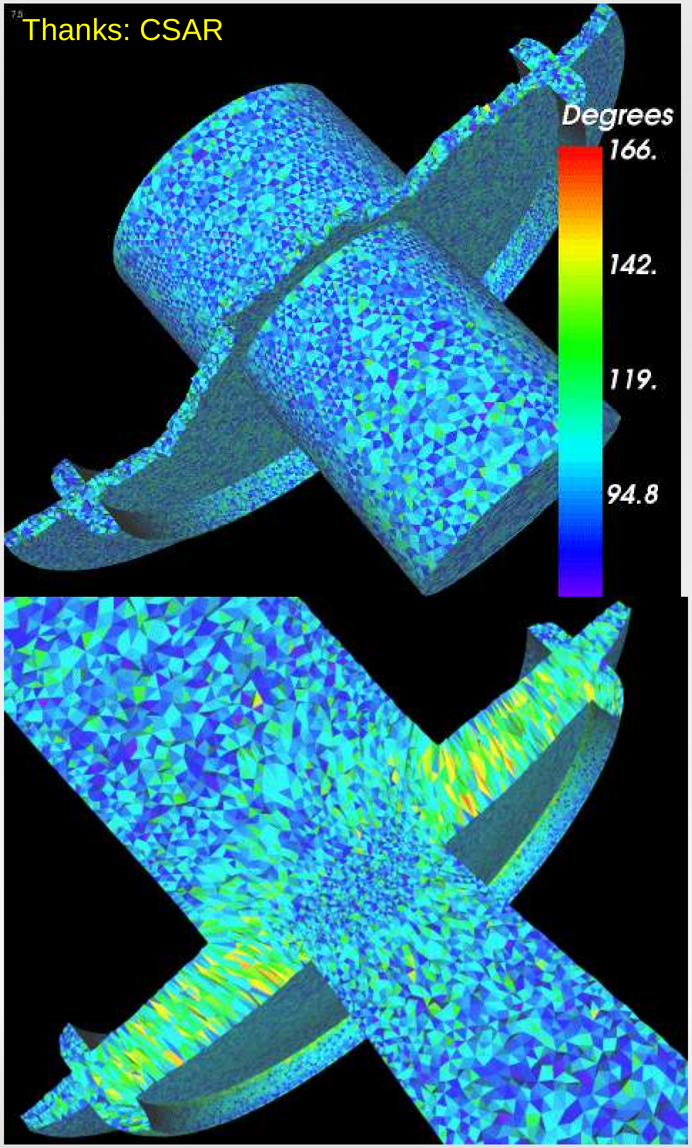
- Rocket simulations
  - Expanding gas domain
  - Burning solid fuel propellant domain
  - Pressure deformation




Thanks: CSAR

Evolving Geometry

- As geometry evolves, mesh quality degrades
  - Compromises accuracy of solution; hampers progress (inverted elements; 0-volume elements)
- Approaches:
  - Smoothing
  - Repair
  - Remeshing



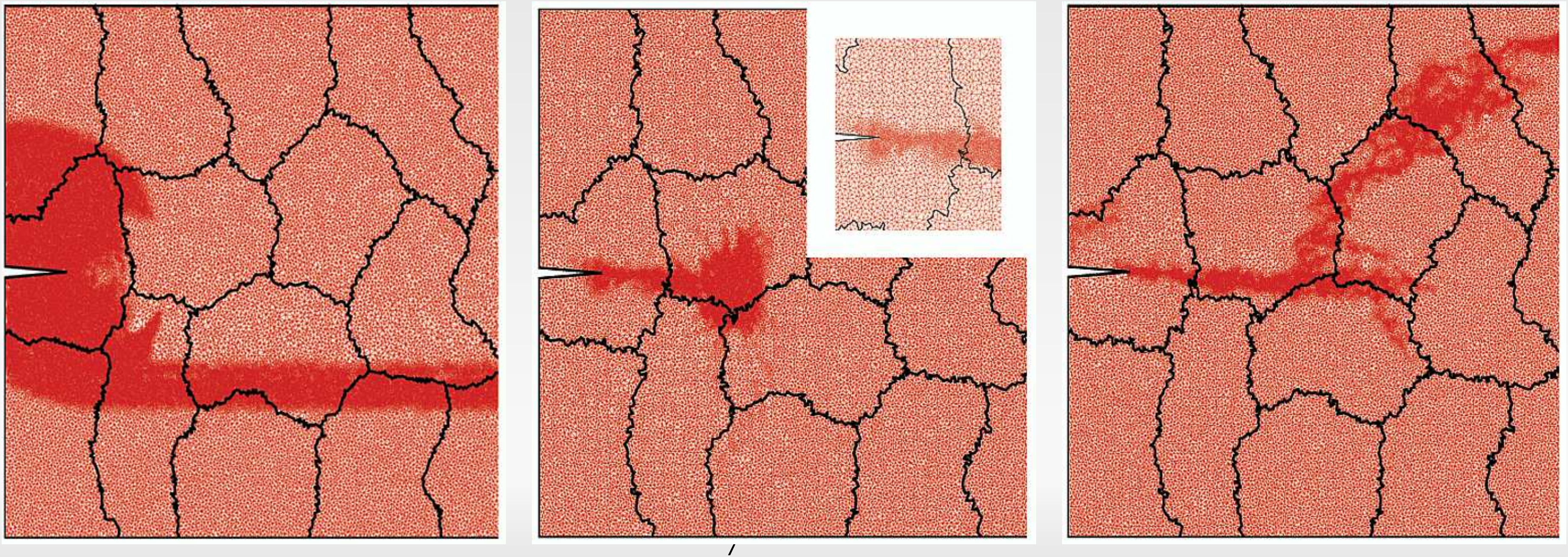
Thanks: CSAR

Mesh Adaptivity

- Mesh refinement and coarsening to accurately capture solution
  - Examples: Wave propagation, dynamic fracture simulation
  - Refinement in areas of interest, coarsening in rest of mesh
  - Refined area varies with passage of time
  - Induces load imbalance among mesh partitions

Mesh Adaptivity

- Example: dynamic failure simulation
  - lower boundary fixed, upper boundary pulled
  - downward propagation of 1D elastic wave is diffracted by notch, creating a region of high stress concentrations in the vicinity of the crack tip



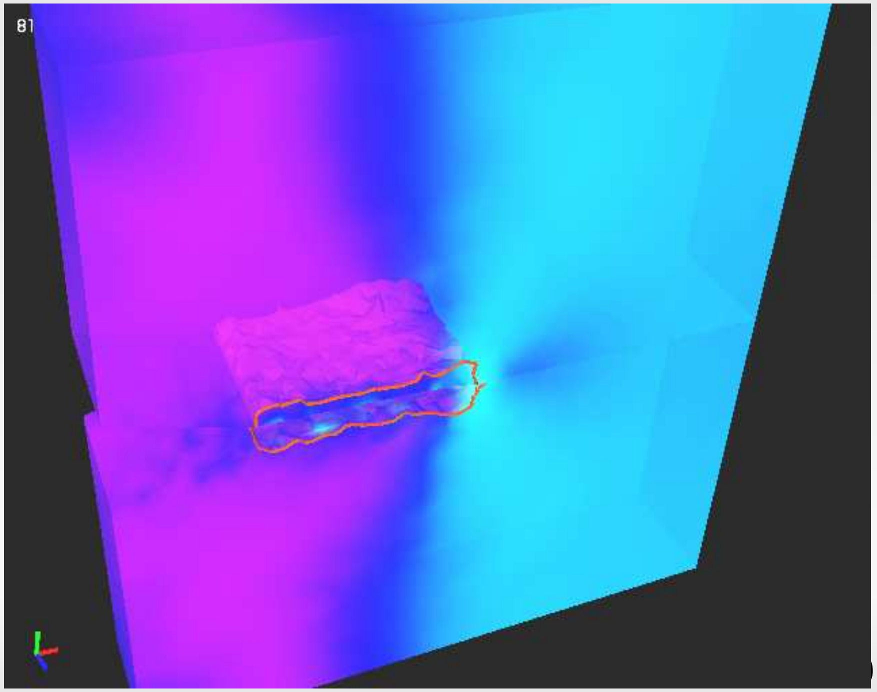
[In collaboration with Geubelle, Mangala]

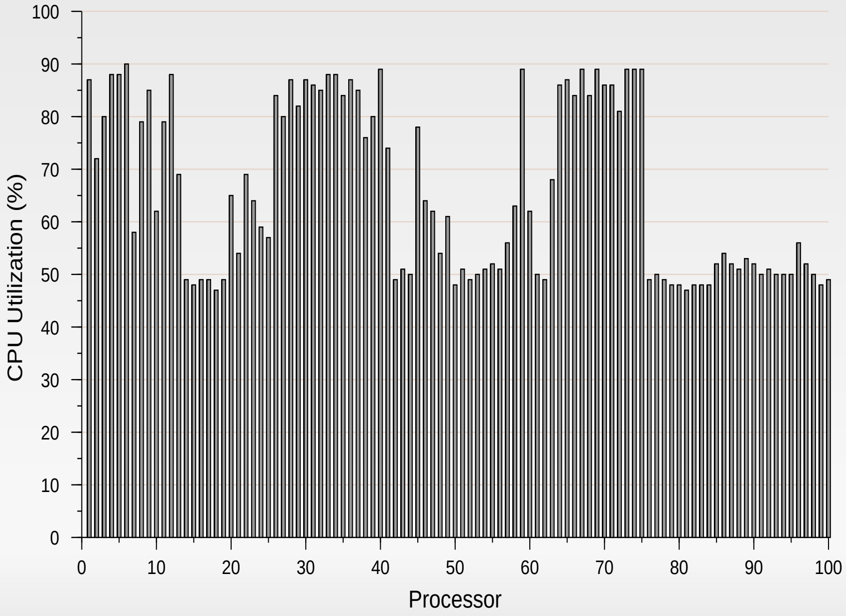
Computational cost changes

- Computational cost can change dynamically for a variety of reasons
  - Material properties of entities change
  - Cohesive finite elements added or activated in vicinity of propagating crack
- Example: *Fractography* solver
  - Domain entities transition from elastic to viscoplastic, incurring much higher computational cost for plastic material updates

Computational cost changes

- Example: *Fractography* solver
  - Dynamic fracture
  - High regional variation in computational cost creates severe load imbalance





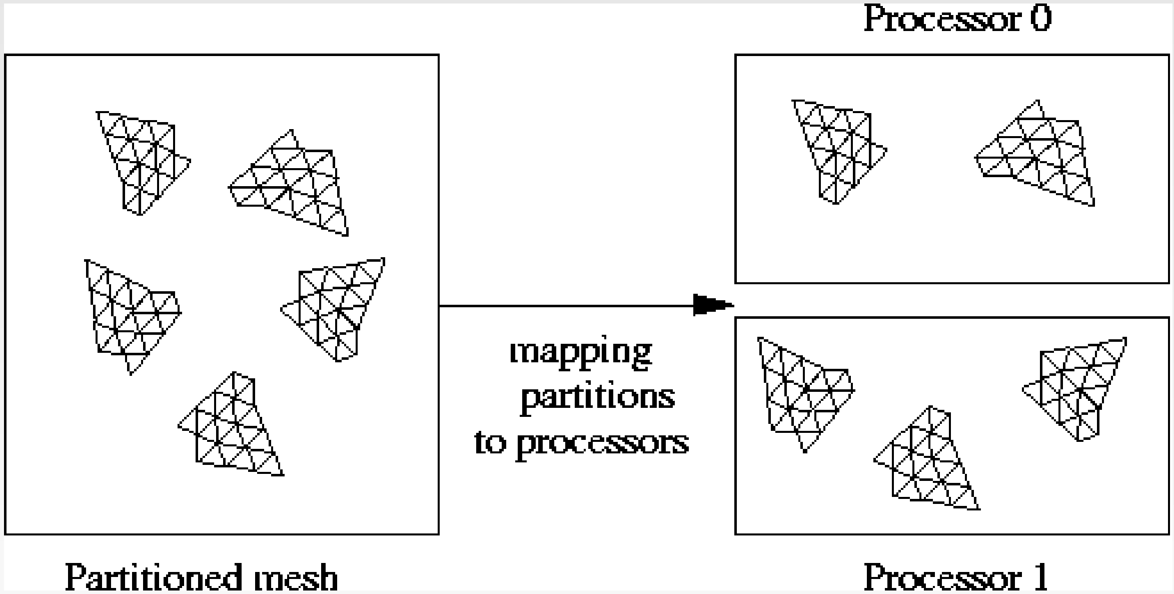
[In collaboration with Geubelle, Breitenfeld]

ParFUM

- Parallel Framework for Unstructured Meshes
  - Takes care of parallel communication, maintenance of ghost layers, partitioning, mesh adaptivity, load balancing, optimized communications, etc.
  - User focuses on development of solver
  - Efficient and scalable: communication/computation overlap and load balancing enabled by object-based virtualization

ParFUM

- Object-based virtualization in ParFUM:
  - Mesh is over-decomposed into N partitions where N >> P, the number of physical processors



ParFUM

- Object-based virtualization enables the Charm++ run-time system to migrate partitions in order to balance load
  - Extremely beneficial to dynamic simulations
- Mesh adaptivity capabilities of ParFUM for user-directed adaptivity, as well as mesh quality improvements
- Load balancing for dynamic unstructured mesh-based simulations
- Solution transfer capabilities

ParFUM: Approaches to Mesh Adaptivity

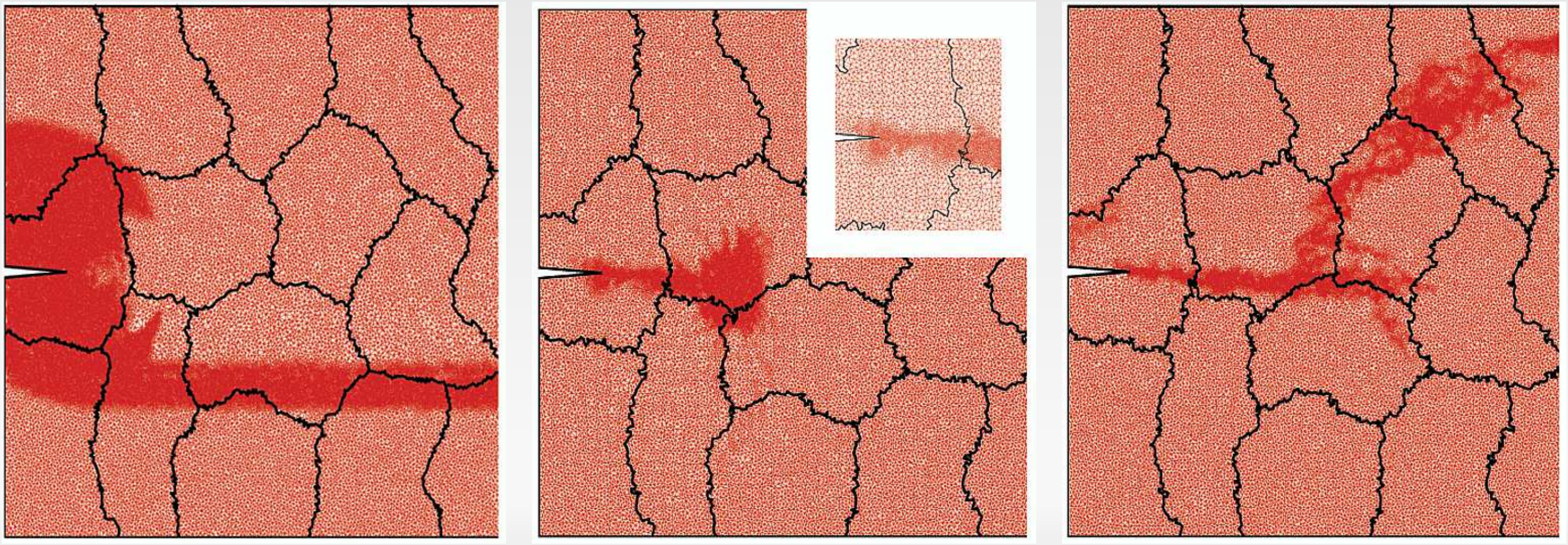
- Incremental Adaptivity
  - produce a completely coherent mesh between each primitive mesh modification
  - useful for applications which require minor modifications to the mesh structure as part of the solver
- Example: Space-time discontinuous Galerkin
  - mesh patch computation interleaved with mesh modifications in the solver

ParFUM: Approaches to Mesh Adaptivity

- Bulk Adaptivity
  - refinement and coarsening as directed by user to capture solution accurately
  - many mesh modifications are performed in bulk, coherent mesh returned at end of adaptivity
- Example: Dynamic fracture simulation
  - refinement in regions of interest (cracktip, propagating wave fronts, etc.)

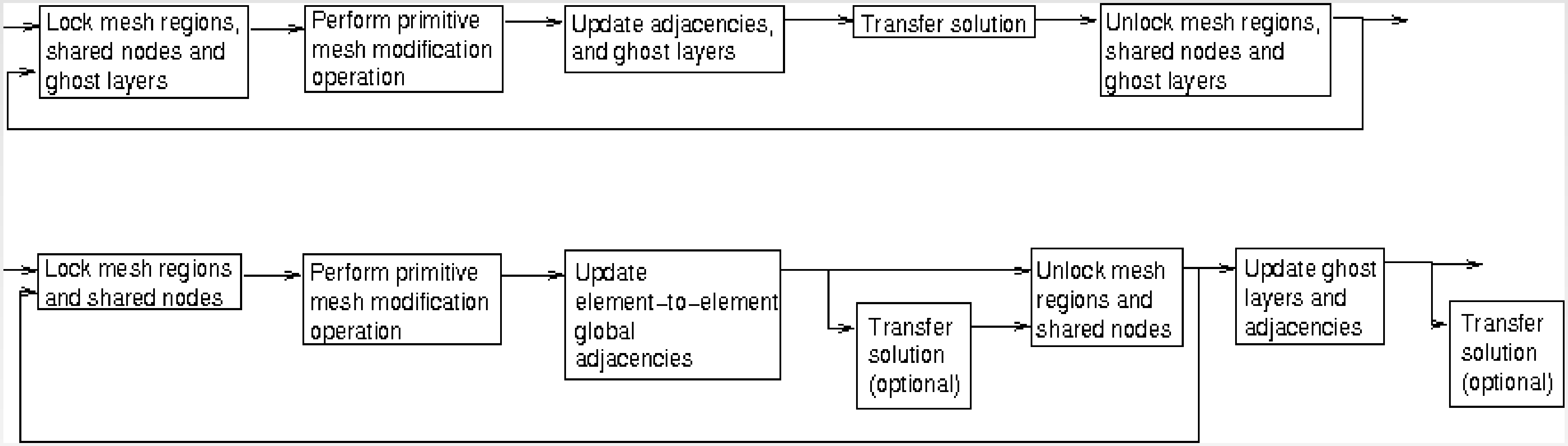
ParFUM: Approaches to Mesh Adaptivity

- Lower boundary fixed, upper boundary pulled
- Downward propagation of 1D elastic wave is diffracted by notch, creating a region of high stress concentrations in the vicinity of the crack tip



ParFUM: Approaches to Mesh Adaptivity

Incremental Adaptivity



Lock mesh regions, shared nodes and ghost layers

Perform primitive mesh modification operation

Update adjacencies, and ghost layers

Transfer solution

Unlock mesh regions, shared nodes and ghost layers

Lock mesh regions and shared nodes

Perform primitive mesh modification operation

Update element-to-element global adjacencies

Transfer solution (optional)

Unlock mesh regions and shared nodes

Update ghost layers and adjacencies

Transfer solution (optional)

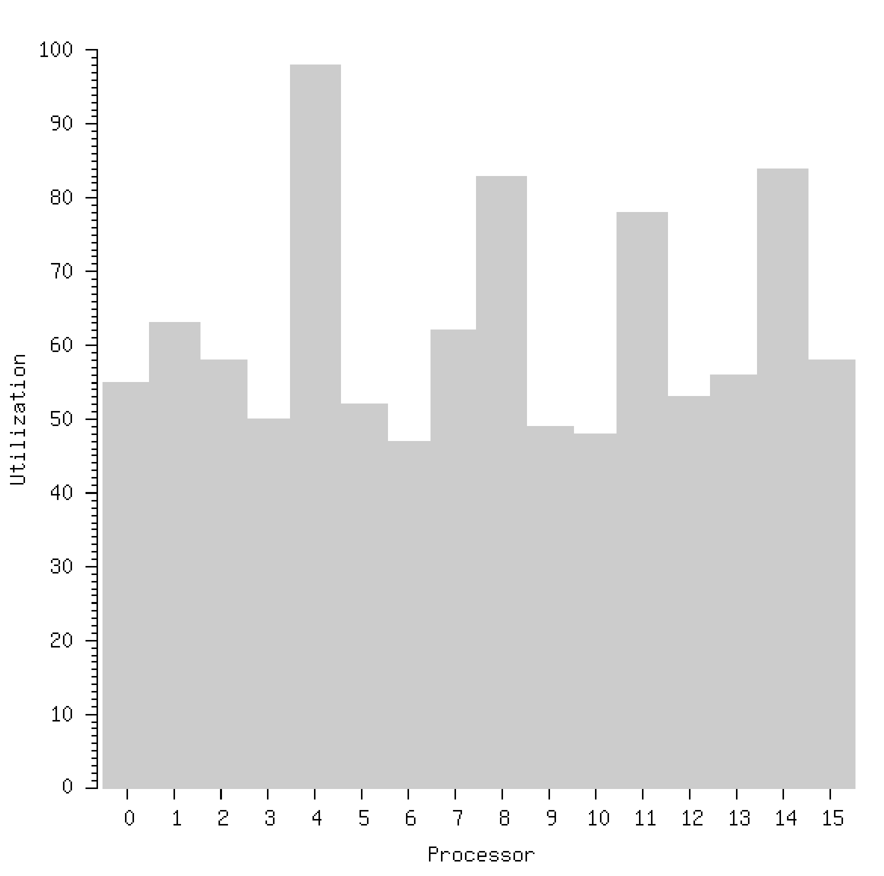
Bulk (Periodic) Adaptivity

ParFUM: Load Balancing for Dynamic Simulations

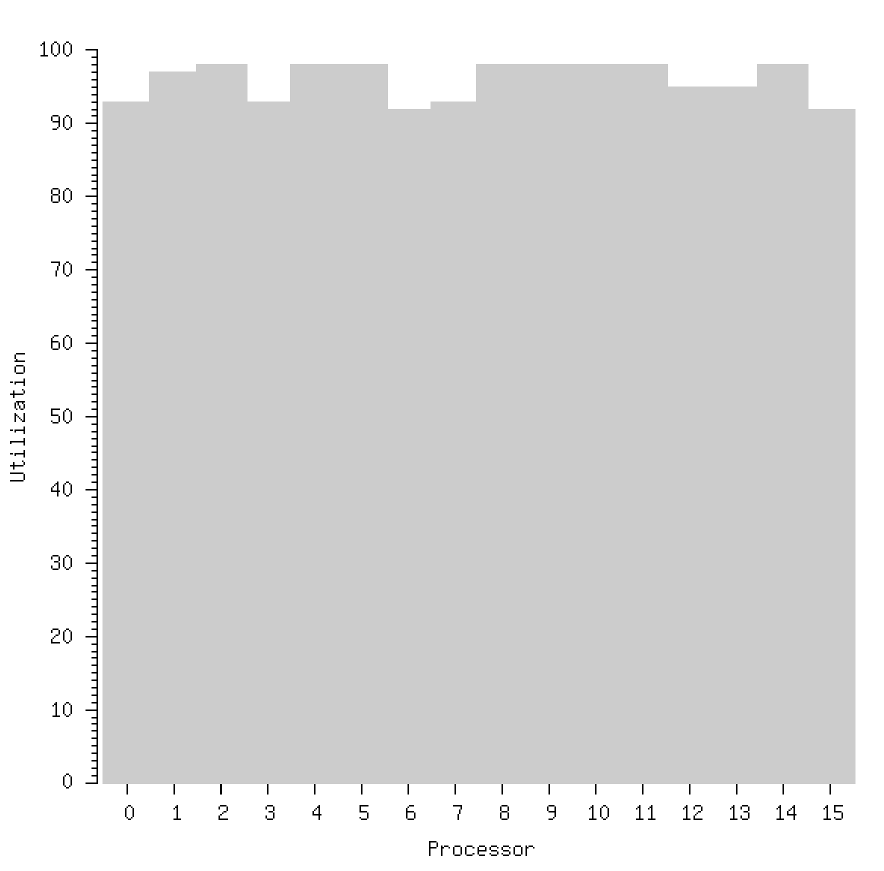
- Imbalance on partitions occurs during:
  - adaptivity: some regions are refined, some coarsened, some left alone
  - post-adaptivity: variations in quantities of mesh entities per partition
  - material transition: resulting in higher computational cost on some entities
- Load per partition does not persist over time

ParFUM: Load Balancing for Dynamic Simulations

- Measurement-based load balancing for post-adaptivity computation



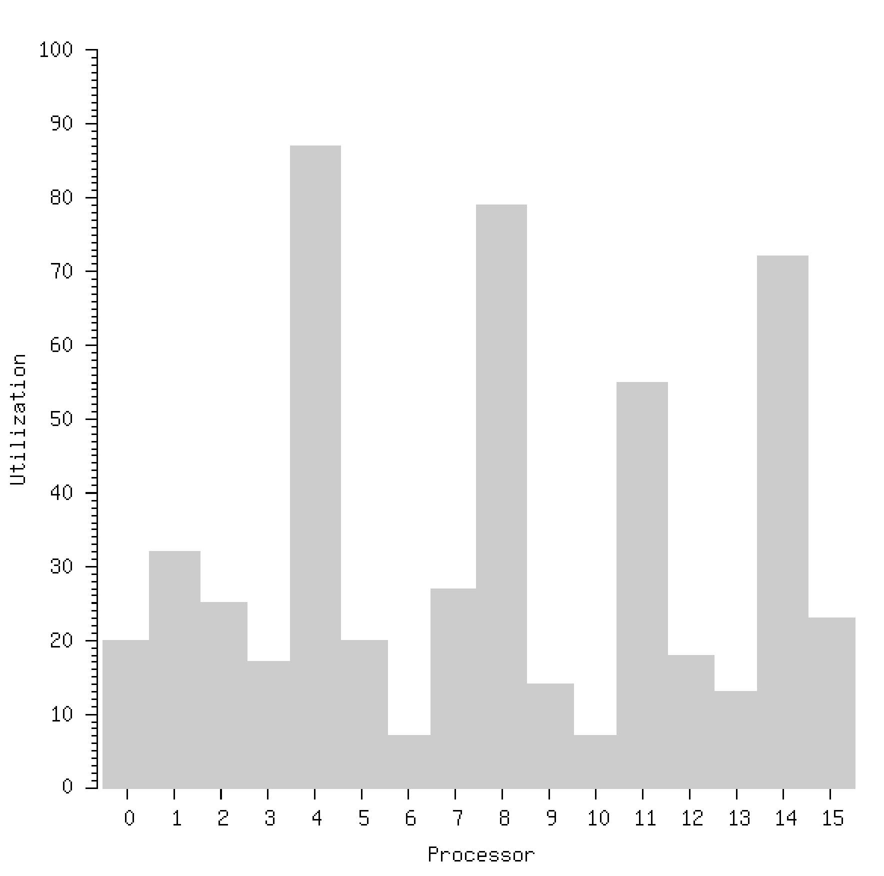
Before



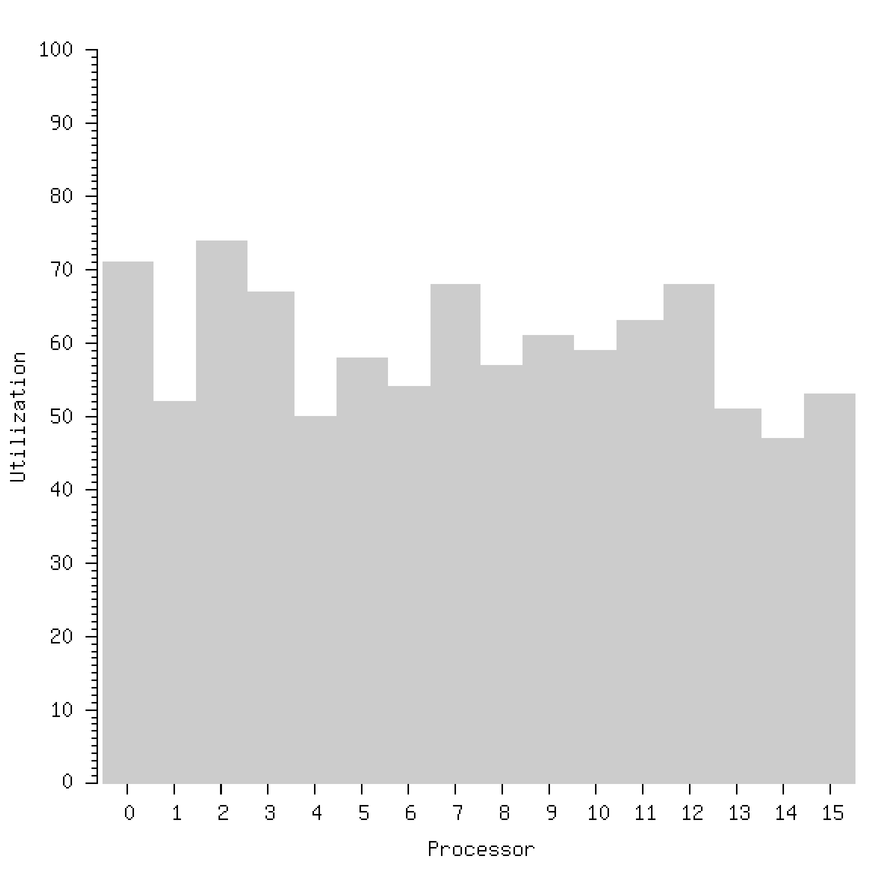
After

ParFUM: Load Balancing for Dynamic Simulations

- Pre-balancing for mesh adaptivity (refinement)



Before



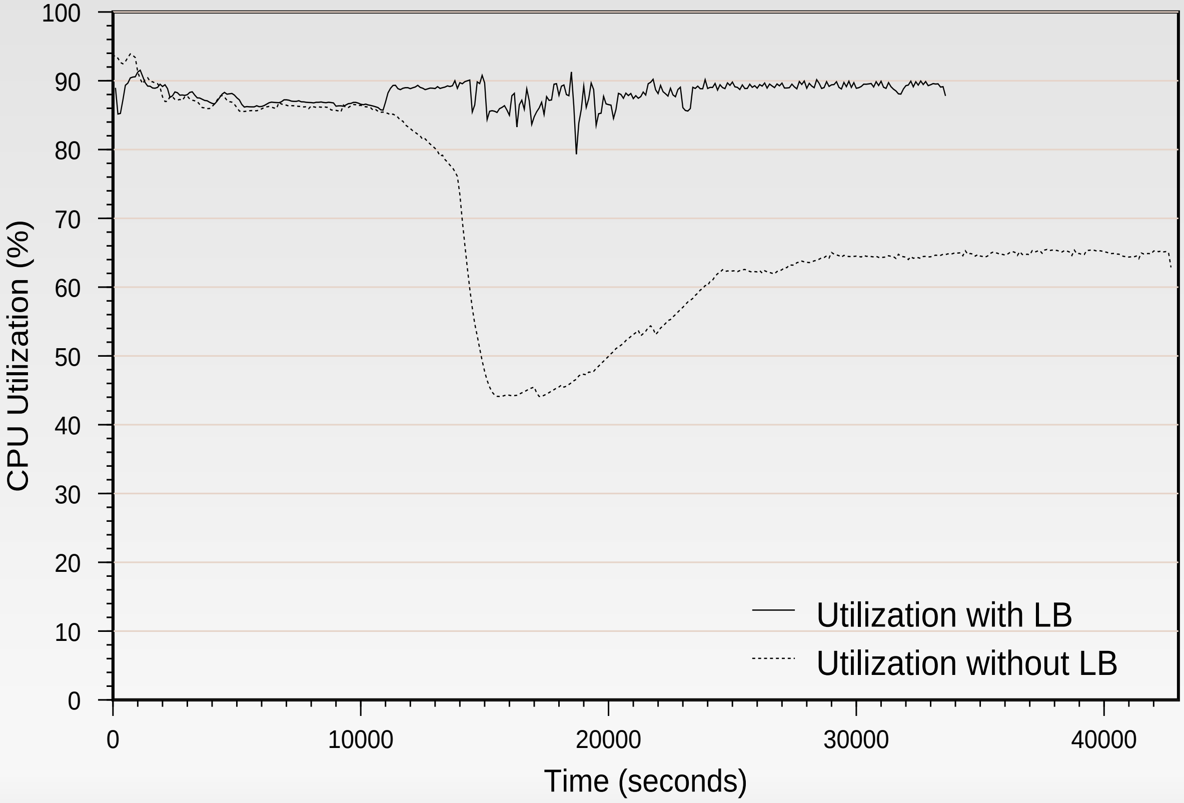
After

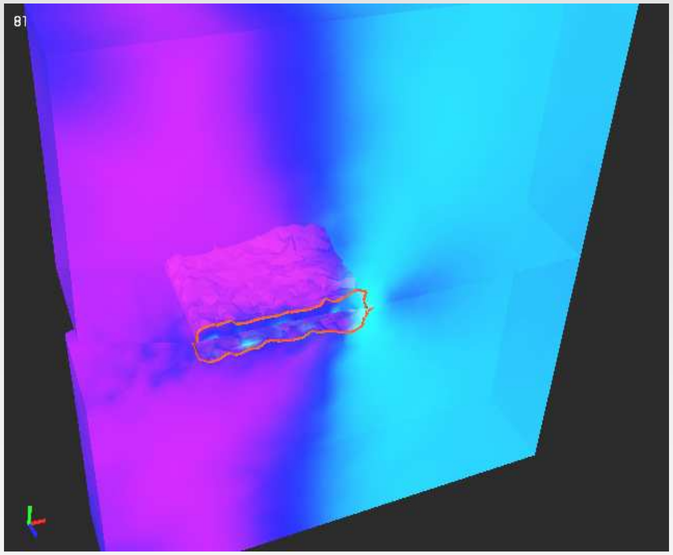
ParFUM: Load Balancing for Dynamic Simulations

- Extreme differences in mesh discretization may overwhelm initial partitioning into N>>P
- Dynamic repartitioning: increase or decrease number of partitions to optimize performance
- Not a true repartitioning, but rather operations to split and join existing partitions, as needed

ParFUM: Load Balancing for Dynamic Simulations

- Measurement-based load balancing for *Fractography*





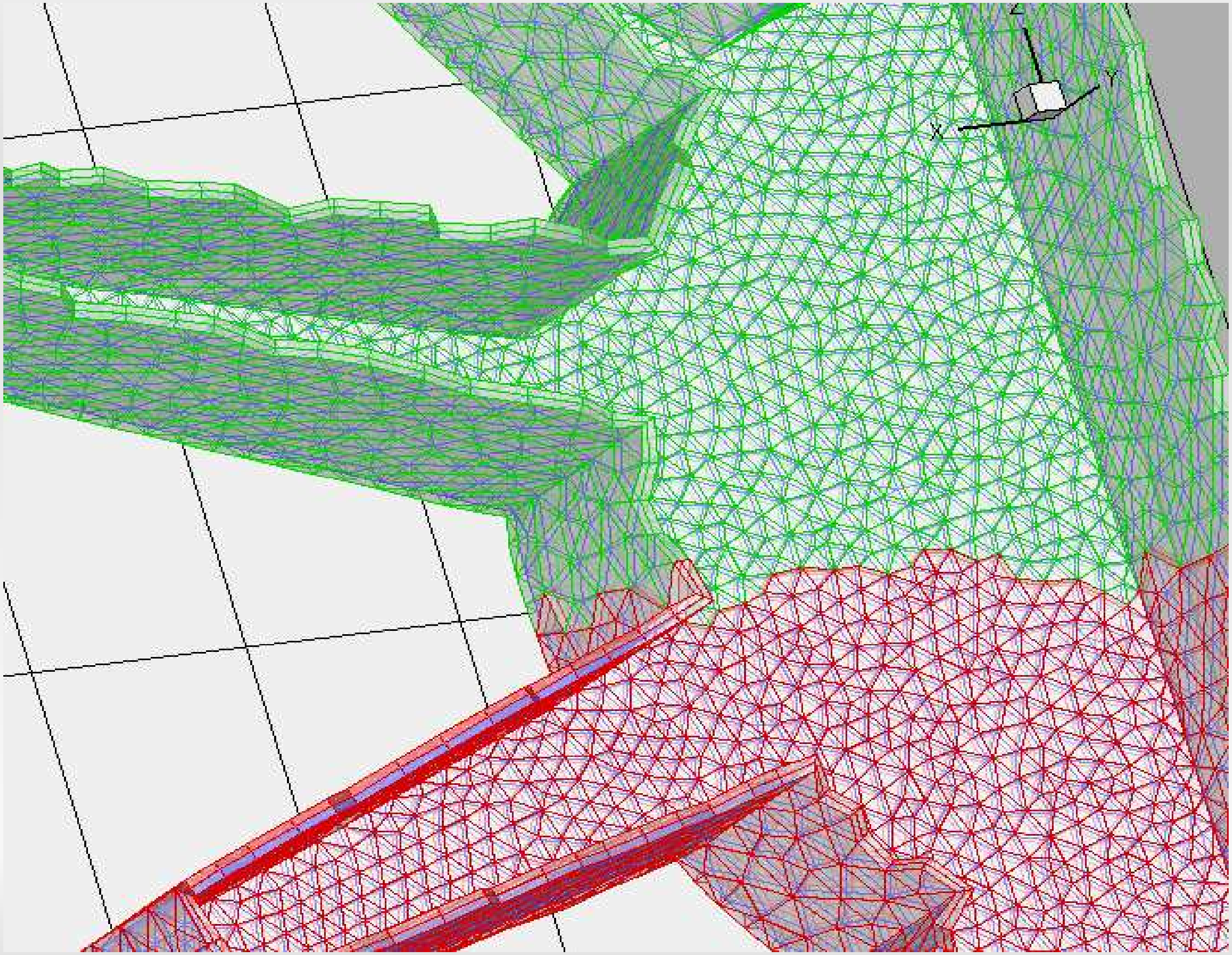
ParFUM: Mesh Quality Improvement

- Mesh Quality: many measures; min/max dihedral angles, aspect ratio, R/r, etc.
- Smoothing: fine adjustment of node coordinates to improve mesh quality
- Local Repair: mesh modification (for example, edge flip operations) in regions of low quality
- Remeshing: regeneration of entire mesh from a model

ParFUM: Solution Transfer

- Incremental: for each primitive mesh modification, determine new solution on new or altered mesh entities
  - Framework for user-specified solution updates
- Bulk: when bulk mesh modification is completed, transfer entire solution from previous mesh to modified mesh
  - Overlays new mesh with extruded old mesh
  - Charm++ collision detection determines volume-weighted transfers of cell data

ParFUM: Solution Transfer



ParFUM: Current Status

COMPLETE

- 2D incremental refinement and coarsening
- 2D bulk refinement
- 2D smoothing
- 2D local repair
- incremental transfer
- bulk volume transfer
- pre-balancing

IN PROGRESS

- 3D bulk refinement
- 2D and 3D bulk coarsening
- bulk surface transfer
- dynamic repartitioning

FUTURE

- advanced load balancing techniques
- contact
- mesh generation?