EPSI Meshing Developments

Ben Ball[1], Robert Hager[2], Seung-Hoe Ku[2], Mark S. Shephard[1] and Fan Zhang[1]

[1]Rensselaer Polytechnic Institute, [2]Princeton Plasma Physics Laboratory

Role of Mesh in Edge Physics Simulations

Must capture behaviors at multiple scales

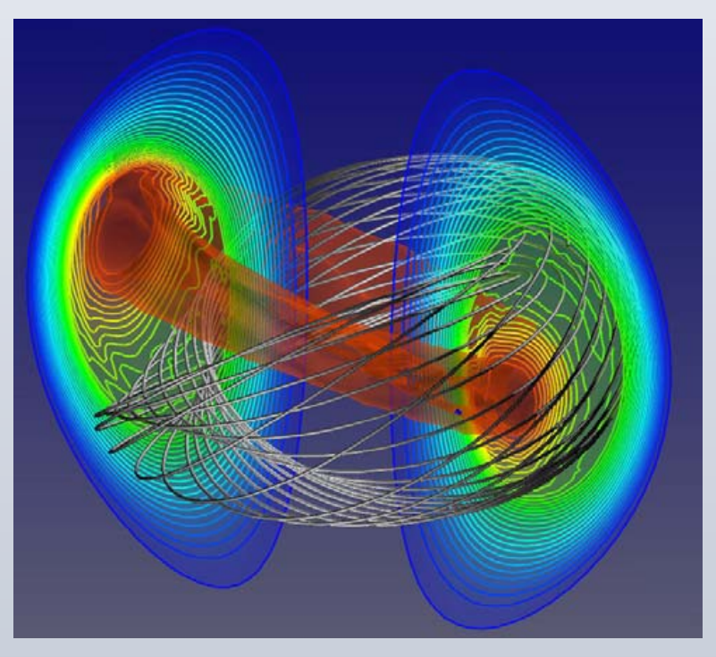
- Employ overlapping two scale approach to solve gyrokinetic Vlasov equations
  - o Tracking of ions and electrons through particle equations of motion
  - o Gyrokinetic Poisson equation numerically solved at the reactor scale to define fields that push particles
  - o Must account for coupling of fields and particles

XGC code is a highly parallel code that implements this multiscale simulation

- Particle motion is updated using Runge-Kutta 4 step iteration or 3rd-order Predictor-Corrector
- Reactor fields solved using an unstructured mesh finite element method
- Mesh and particles must be properly coupled at each step in the simulation

Requirements Driving Meshing Efforts

Critical to account for the physics of the particle motion in the models and associated numerical solution methods



- Motion primarily confined to be along field lines of the magnetic field (on which the magnetic flux is constant)
- Motion of particles in toroidal direction much faster than in the poloidal plane

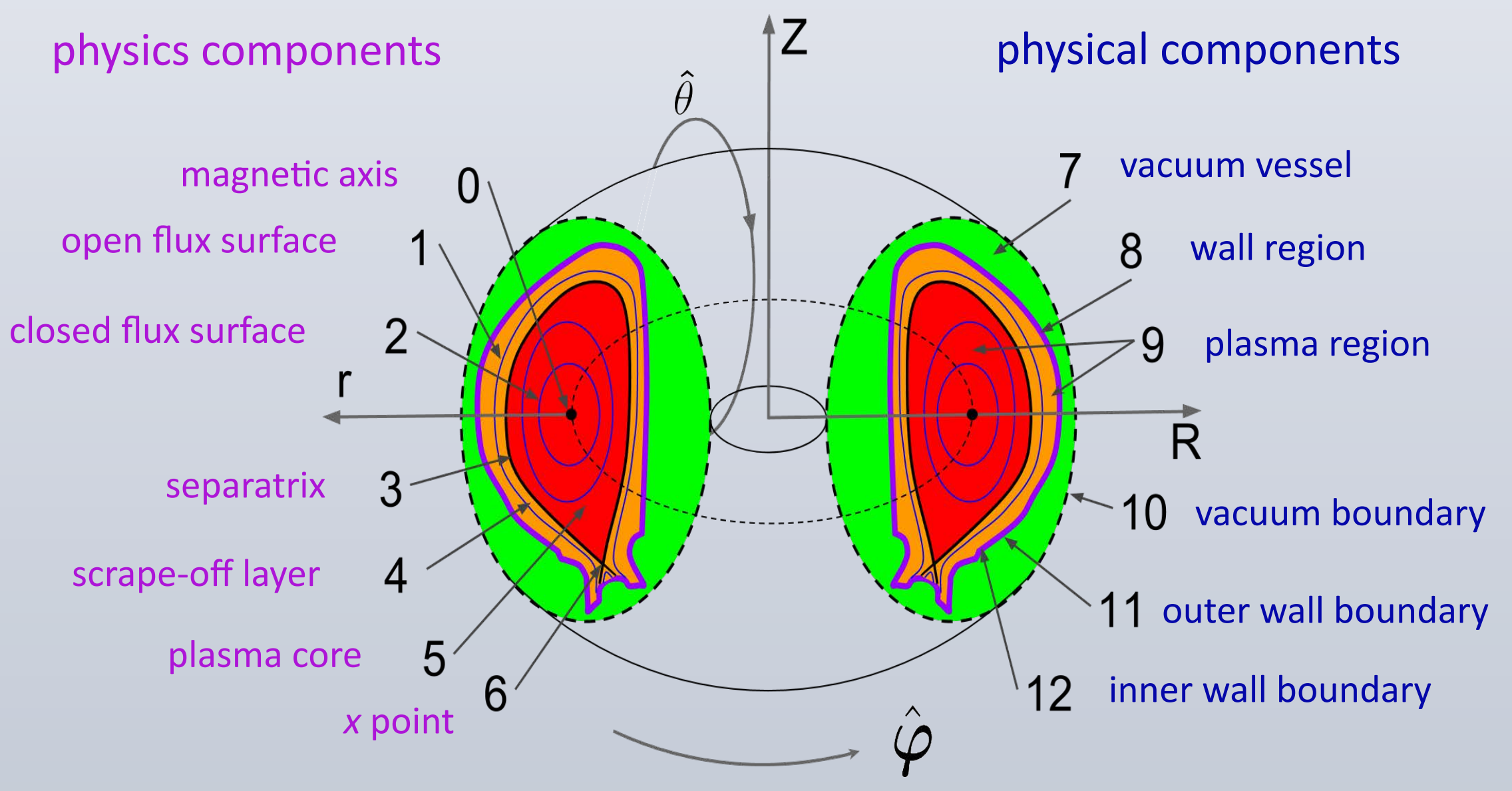
Effective numerical models account for the physics of the particle motion – leads to a set of mesh requirements

- Primary mesh is a 2D mesh resolving the poloidal plane
  - o Most effective to carefully align mesh to follow magnetic field lines in the critical regions
  - o Projection of particle motion on the poloidal plane controls mesh vertex placement on flux curves
  - o Coupling of particle and mesh methods, and accuracy of mesh method dictate mesh fineness and gradation, as well as need for element shape control
- Computing Issues
  - o Need to ensure scalability as more complex reactors are considered (ITER) and memory per core decreases
  - o Drives need for parallel mesh

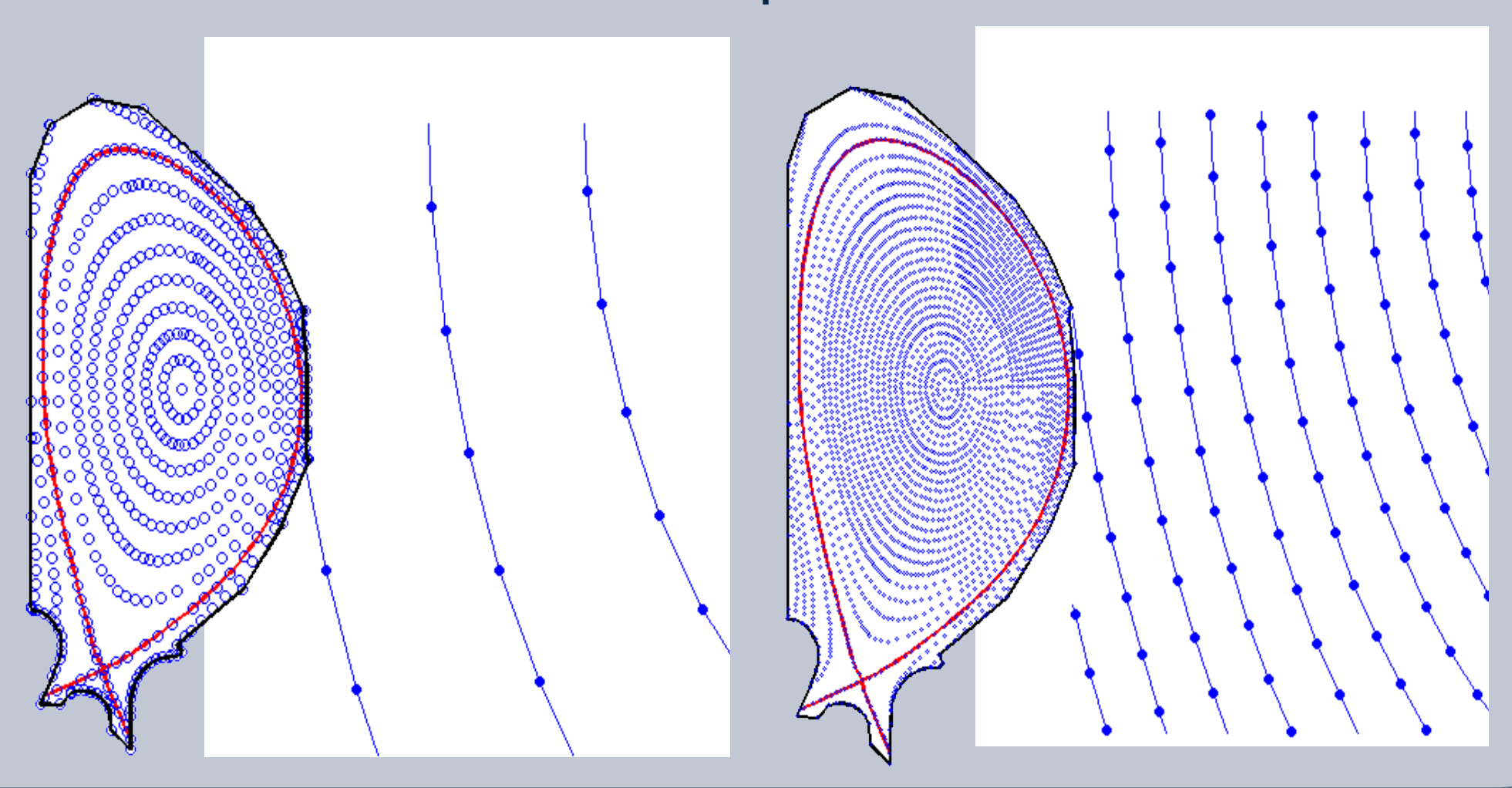
Geometry Definition

Automatic meshing requires unique geometry representation

- Boundary representation required
- Meshing requirements dictate geometric model meshed includes combination of physical and physics entities
- Flux surfaces are key physics component required
  - o Require mesh vertices specifically placed on flux curves



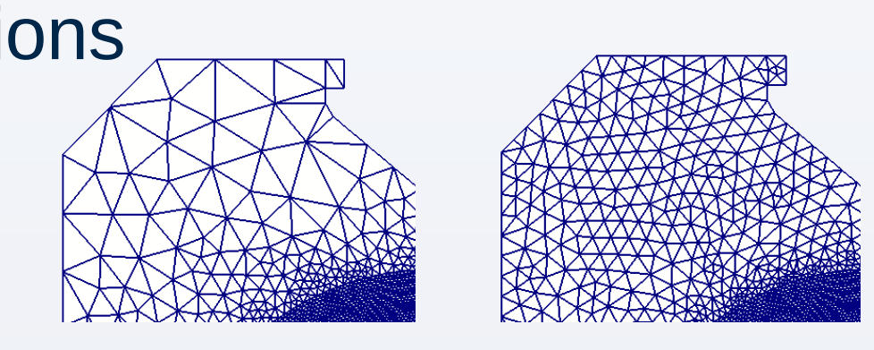
- Constant flux curves constructed from data measured on a uniform grid
  - o Input flux (ψ) values are obtained from experimental data on a coarse uniform grid
  - o Interpolative splines are constructed from this data to give continuous and smooth ψ values
  - o Curves are then constructed to be perpendicular to the gradient of ψ and parallel to the magnetic field, meaning each curve has a constant ψ value



Meshing Generation Method

Combines multiple meshing functions

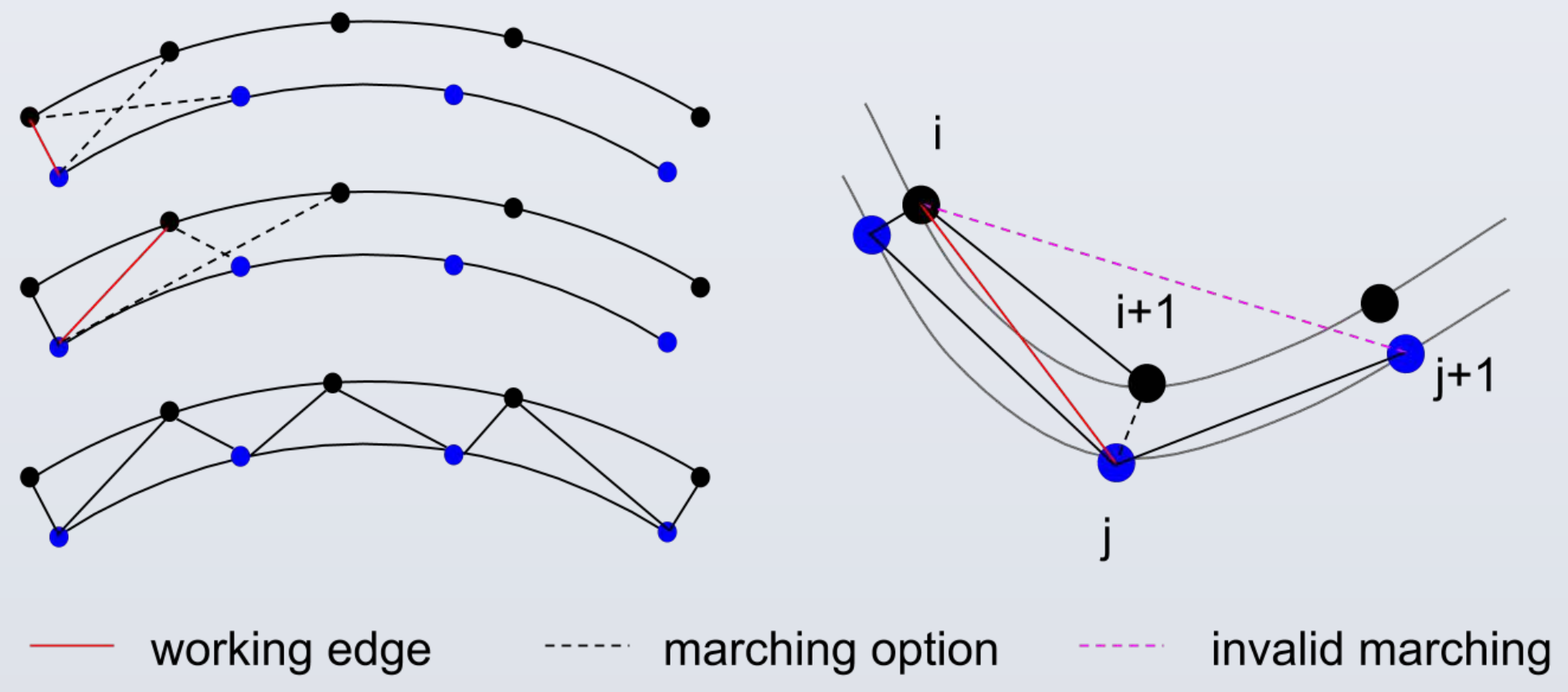
- General triangulation
  - o Generate well shaped, graded mesh near reactor wall where geometry is complex



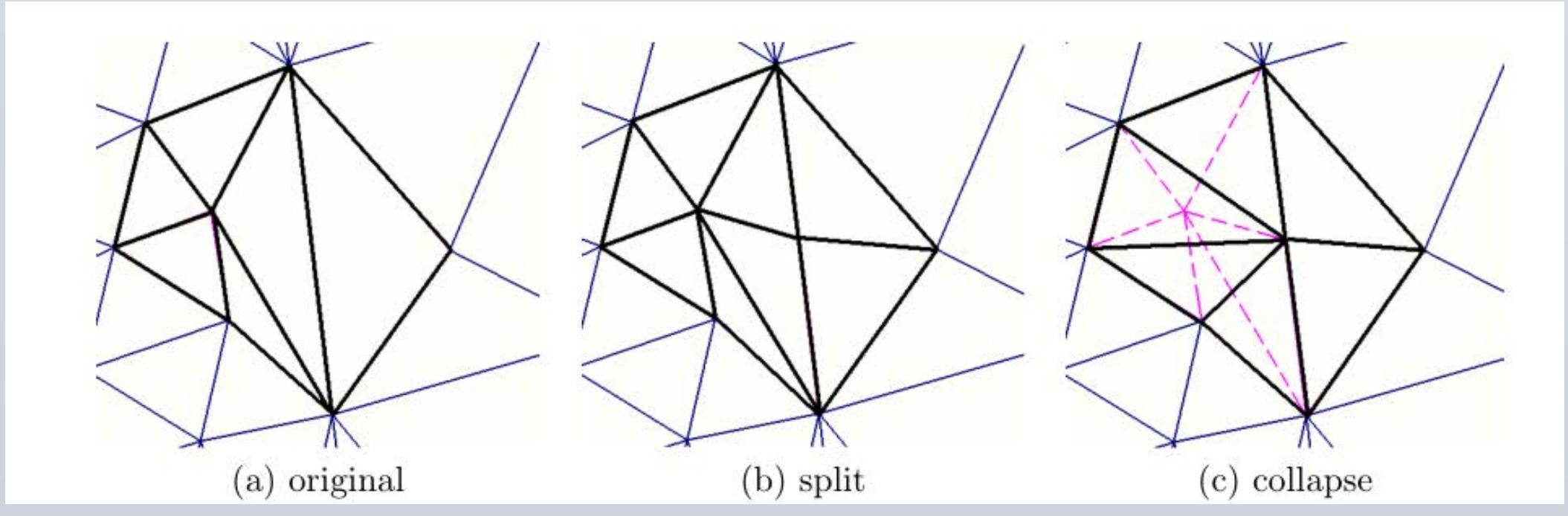
Strongly graded to coarse wall mesh

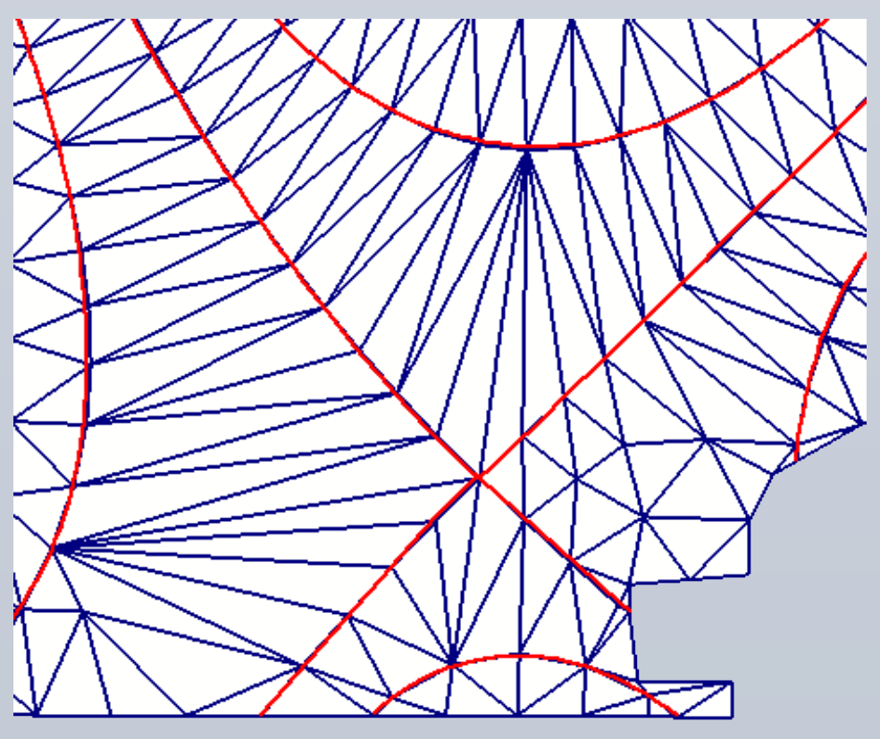
Graded to a finer wall mesh

- One-element layered meshing procedure
  - o Require depth of one element between critical flux curves
  - o Given constant flux curves with vertices placed as required by the vertex to follow particles, a direct algorithm used to construct best shaped elements

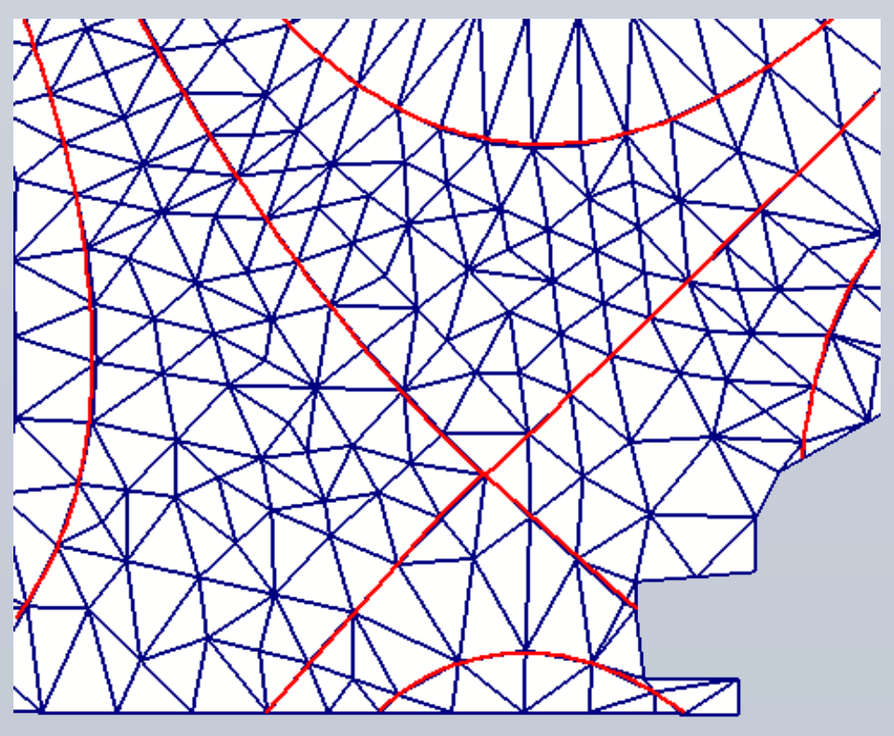


- Mesh Control near x-point
  - o Near x-point the one-element procedure between flux surfaces produces poorly shaped elements
    - – Such element can degrade the local solution
    - – Local physics is such that maintaining the one-element deep layer is not critical
  - o Apply local general mesh modifications to control element shapes





initial layered mesh at x point

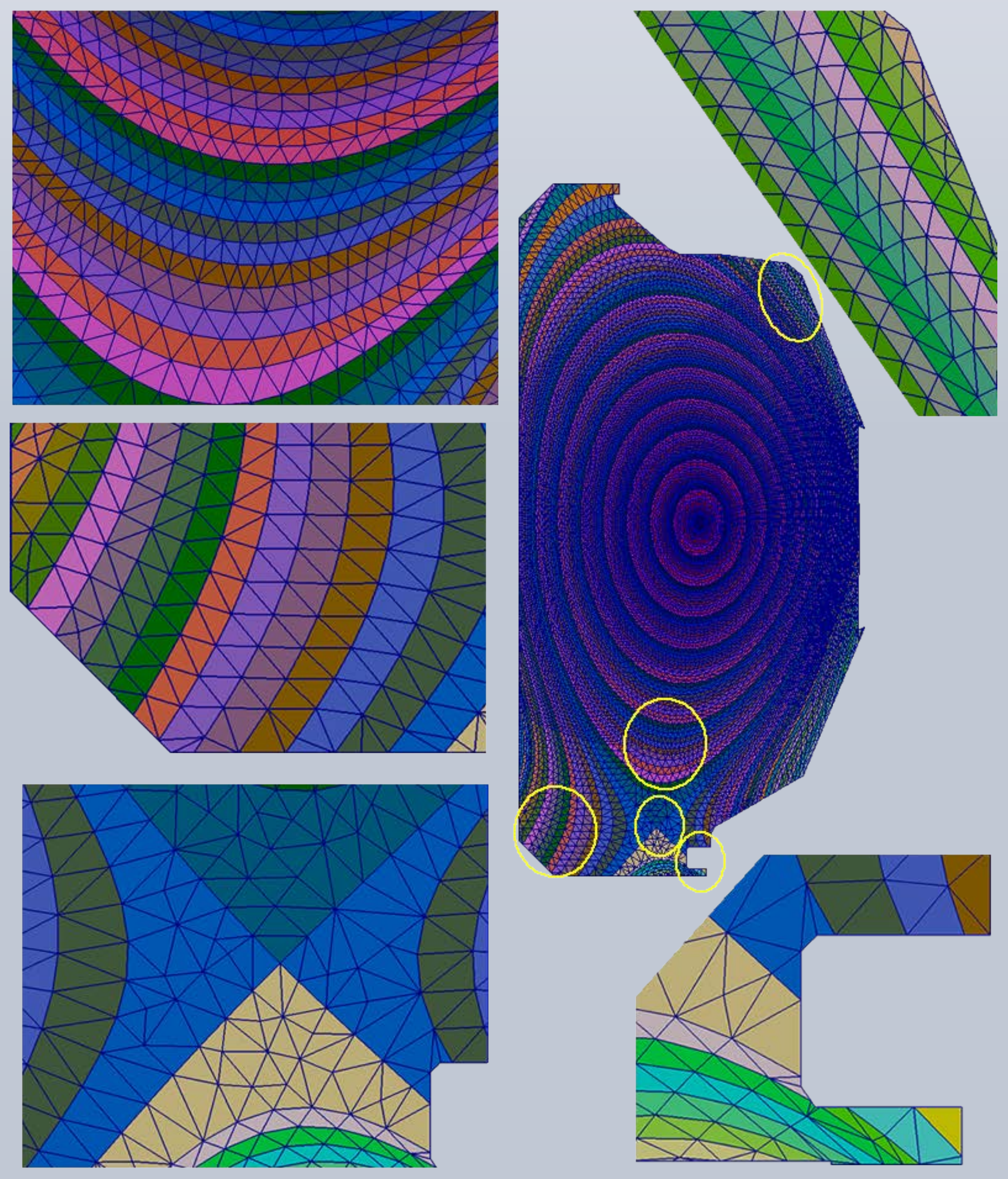


mesh at x-point after mesh modification

Mesh Generation Example

Comparison

- Original method took >11 hours of combined human and machine time
  - o Mesh still did not have desired quality in all locations
- Run time (human plus machine) for new procedure reduced by a factor of >60 to under 10 minutes for finest mesh generated (10^7 elements)
- New procedure will support dealing with new and more complex reactor configurations (e.g., ITER)



Parallelization of Mesh with Particles

Current XGC code

- Entire mesh exists on every process
- All values associated with the mesh are stored on every process
  - o Requires global communication

New distributed code being developed

- Connects with SCOREC's Parallel Unstructured Mesh Interface (PUMI) for mesh representation
  - o Mesh and associated information is partitioned among processes
  - o Particles to be associated with mesh regions
  - o Most communication is local
  - o PUMI partition model for interprocess communication

Parallelization Algorithms

XGC has three main interactions with the mesh

Particle search

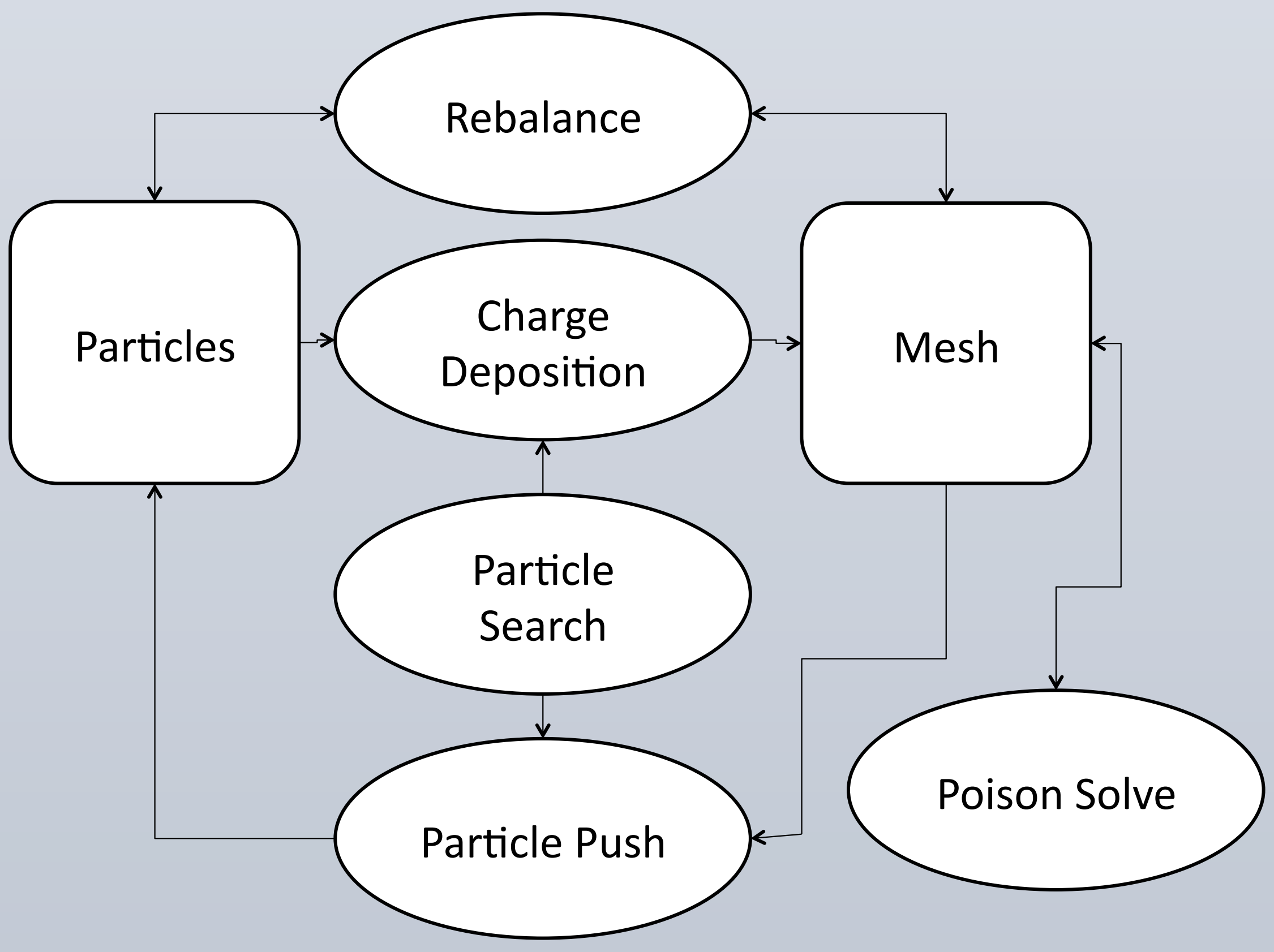
- Used to locate the element a particle is in
- Used mainly during particle push and charge deposition
- In the new distributed code, this is implemented using mesh adjacency information
  - o Each particle remembers the element where it was last located
  - o An evaluation of area coordinates within that element indicates the next element to search
  - o If the next element is on a different partition, the particle is queued to be transferred

Rebalancing

- Is now driven by the mesh
- Requires particles to follow their containing element as it moves
  - o To remember where each particle should be sent and to match it back to it's containing element on arrival, a temporary map is created

Poisson Solve

- Field equations solved on finite element mesh



Rebalance

Particles

Charge Deposition

Mesh

Particle Search

Poison Solve

Particle Push

Progress on Parallel Mesh Implementation

Implemented thus far

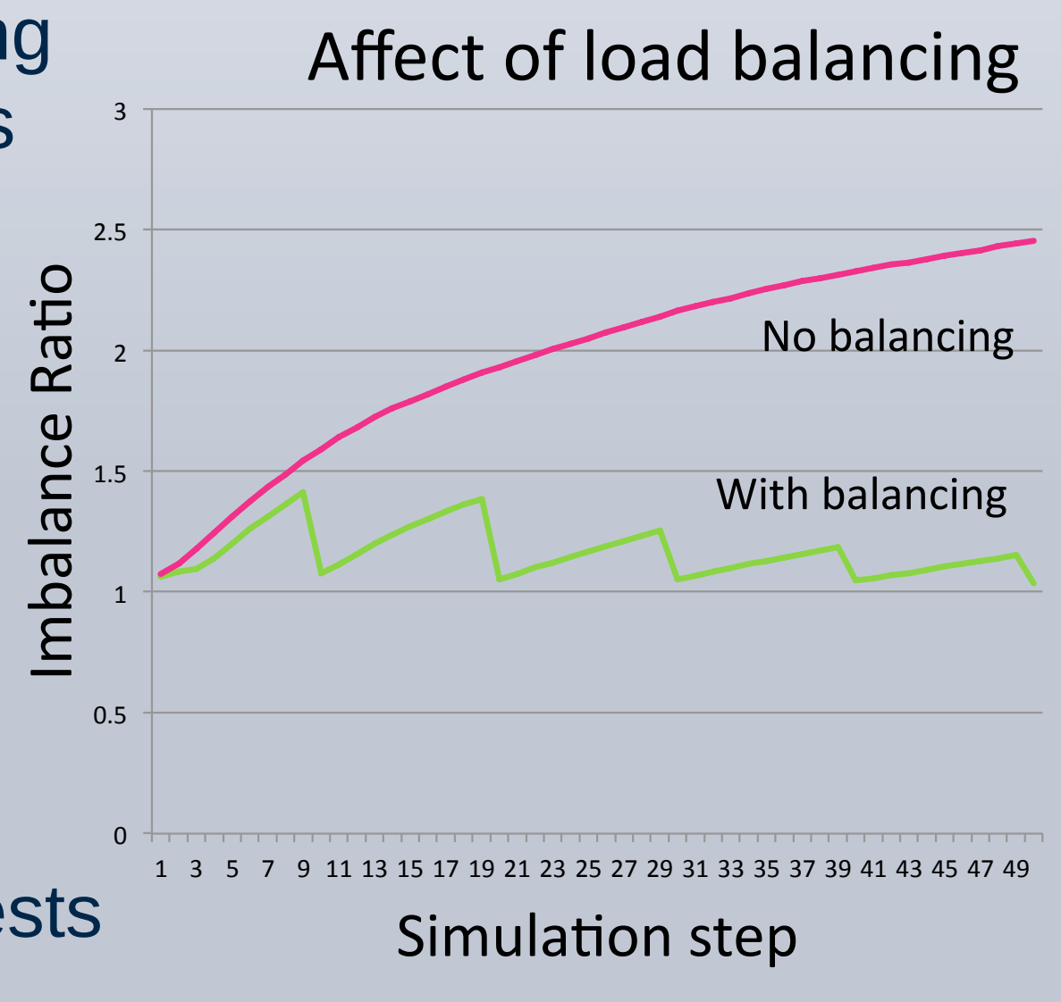
- Initialization and starting distribution of mesh and particles
- Search for particles in mesh
- Deposition of charge from particles to mesh, and associated mesh-oriented data storage
- Particle movement, including transfer between processes when required
- Periodic rebalancing of mesh and particles to prevent load imbalance

Developments underway

- Integration of testing of full set of particle operations
- Component performance tests

Remaining developments

- Integration with finite element field solve
- Scaling and performance improvements



Affect of load balancing