Unstructured Mesh Technologies for Fusion Simulation Codes

M.S. Shephard, G. Diamond, M. Hakimi, G. Perumpilly, O. Sahni, E.S. Seol, C.W. Smith, W.R. Tobin, A. Truszkowska, E.S. Yoon*, C. Zhang

Rensselaer Polytechnic Institute

In Collaboration with

M.W. Beall, B.R. Downie, R. Nastasia and S. Tendulkar

Simmetrix Inc.

V. Dobrev, T. Kolov, M.L. Stowell
Lawrence Livermore National Laboratory

D. Ibanez
Sandia National Laboratories

And with Multiple SciDAC Fusion Partnerships

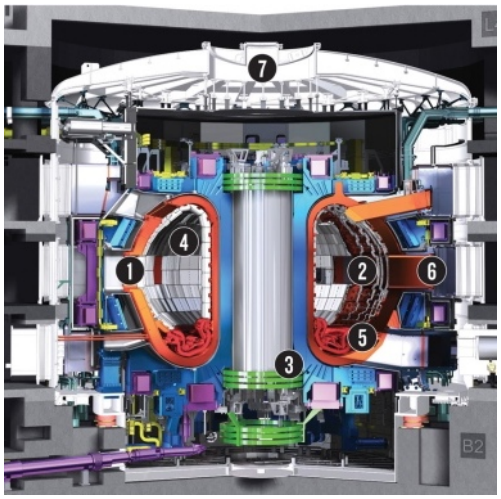
*Now on the faculty of Ulsan National Institute of Science and Technology, Korea

FASTMATH | Lawrence Livermore National Laboratory | Simmetrix Inc. | Sandia National Laboratories | Rensselaer

Background / Selected Developments

Why unstructured meshes for fusion simulations?

- High fidelity simulations must accurately represent fusion device geometry
- Codes have specific meshing requirements
- Unstructured meshes deal with any geometry



Areas of Developments

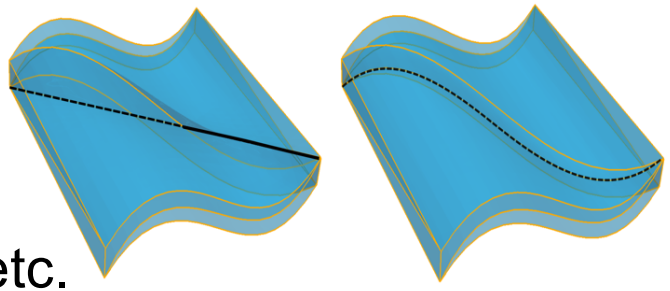
- Mesh generation/adaptation advances for fusion SciDACs
- Support of M3D-C1 extended MHD code for core plasma
- Adaptive RF simulation workflows
- Infrastructure for unstructured mesh PIC simulations
- Status of XGCm – mesh-based version of XGC edge plasma
- Status of GITRm – mesh-based GITR impurity transport

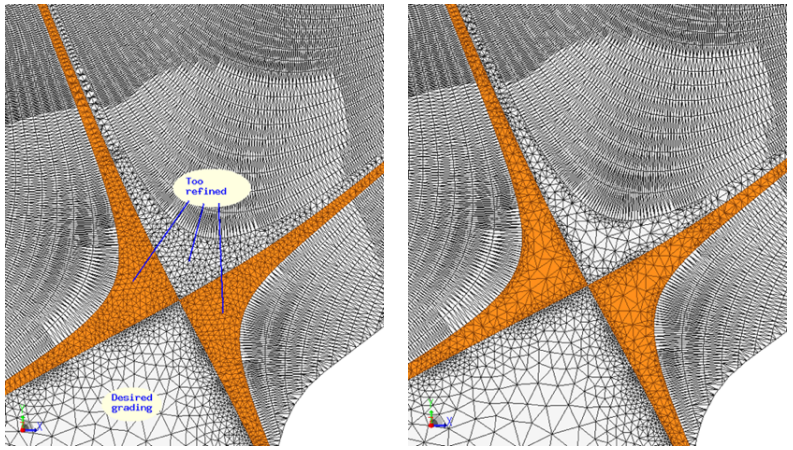
Last year overviewed technologies, this year focus on advances

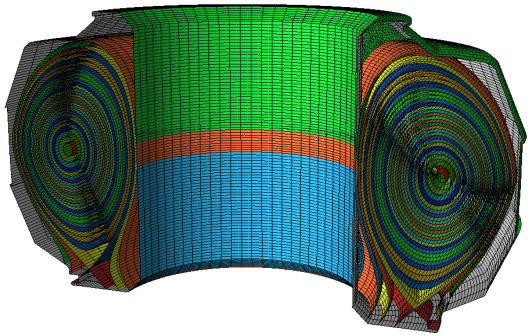
Fusion Geometry and Meshing Developments

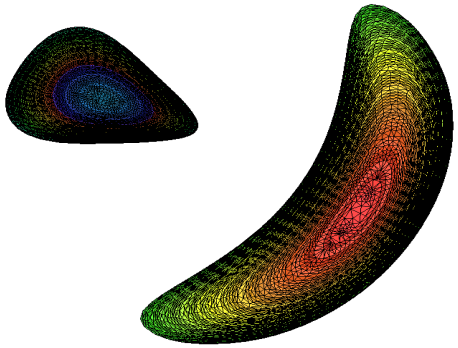
Working directly with Simmetrix

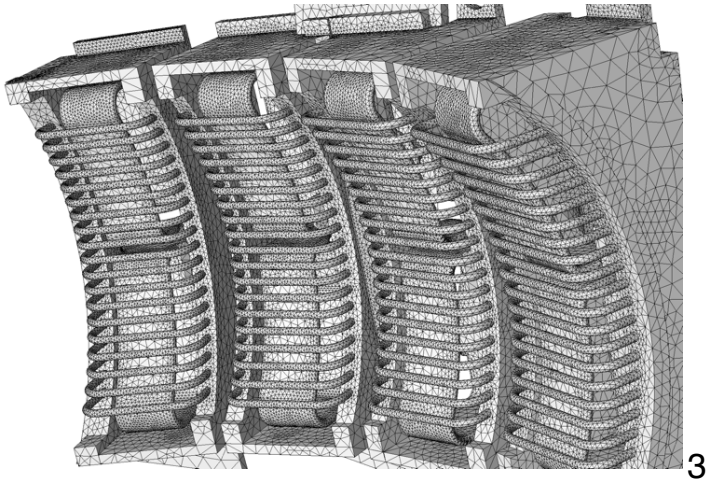
- Tokamak geometry and meshing
  - Reordering mesh for better memory access
  - ITER, DIII-D, Alcator C-MOD, NSTX, KSTAR, etc.
  - EFIT physics geometry
  - Improved mesh quality at x-point and at wall/flux curve interactions
- Graphical user interface enhanced, user documentation being written
- Curved mesh generation
- Higher order curved mesh adaptation
- Investigation of stellalator meshing











Adaptive RF Simulation Workflow

Workflow steps

1. Obtain and clean-up antenna CAD models
2. Combine antenna, reactor and physics geometries
3. Associate analysis attributes
4. Automatic mesh generation
5. MFEM finite element analysis
6. Estimate discretization errors. If below tolerance terminate
7. Adapt mesh and return to step 5

Steps 1-3 are interactive
Steps 4-7 are automated

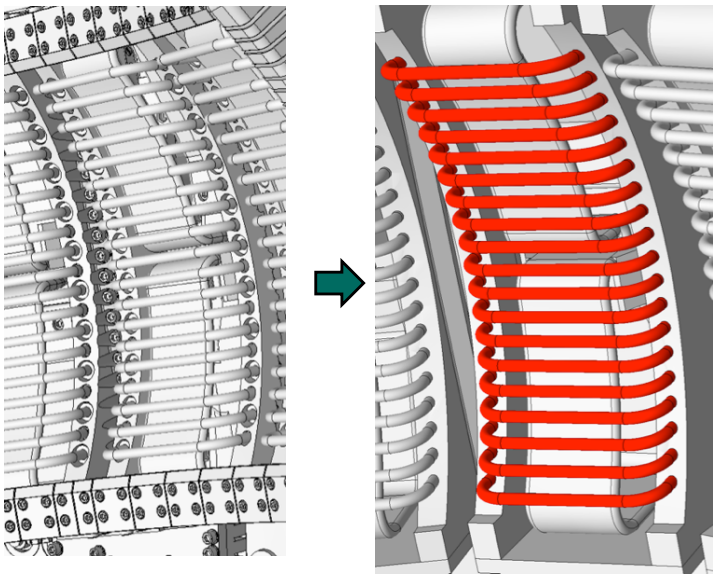
Tools used

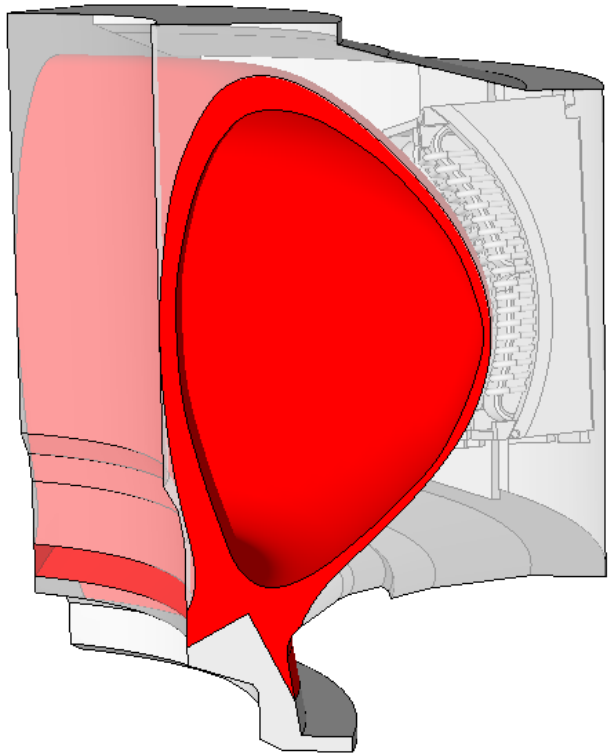
- Geometry – Simmetrix SimModeler, CAD systems, EFIT
- Mesh generation – Simmetrix MeshSim
- Mesh services and curved mesh adaptation – PUMI
- High-order finite element analysis – MFEM
- RF simulations – PetraM
- Error estimator – PUMI

Defeaturing CAD Models and Combining Geometry

Tools used: CAD systems, SimModeler

- Antenna defeatured using functions in the SimModeler GUI
  - Fast elimination of large numbers of small features
  - Simplified geometry can be added
- Reactor cross section extruded
- EFIT data used to construct flux surface
- Model components combined in SimModeler
  - Region identification
  - Preparation of analysis geometry



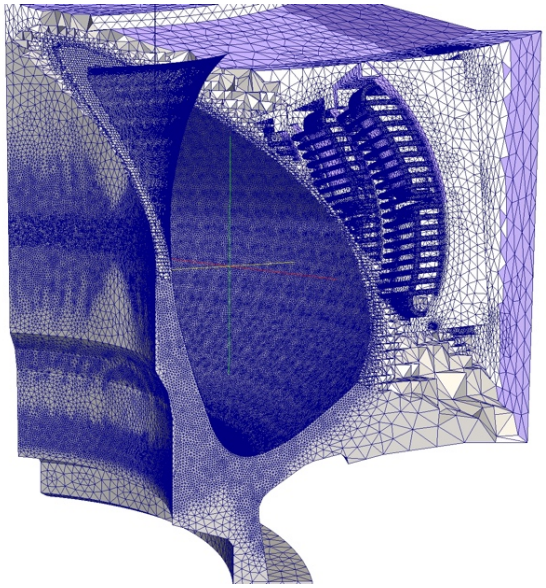


Curved Mesh Generation

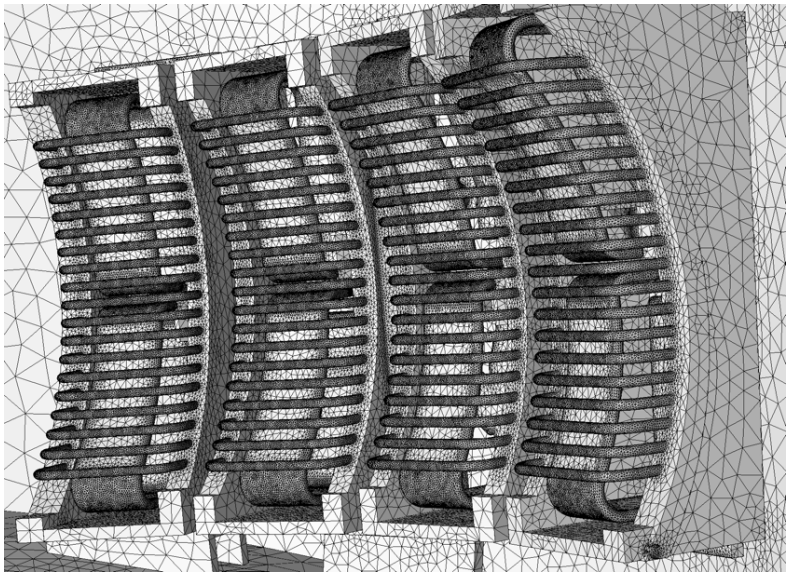
Tools used: MeshSim, GeomSim, Adapt

Functionalities needed

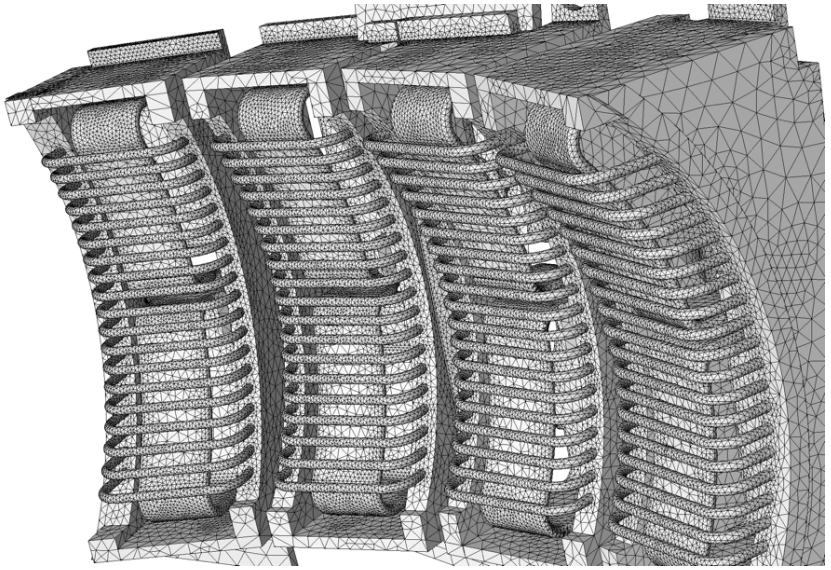
- Mesh controls on analysis geometry
- Automatic mesh generation
- Linear or quadratic geometry mesh
- Curved mesh entity inflation (initial version – further efforts required)



8M element mesh with refined SOL



8M element linear mesh



2.5M element quadratic mesh

Integration of PUMI with MFEM

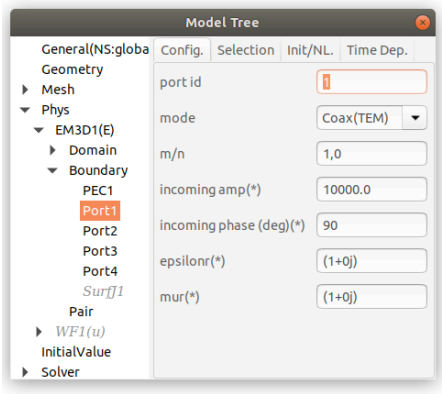
Tools used: PUMI, PetraM, MFEM

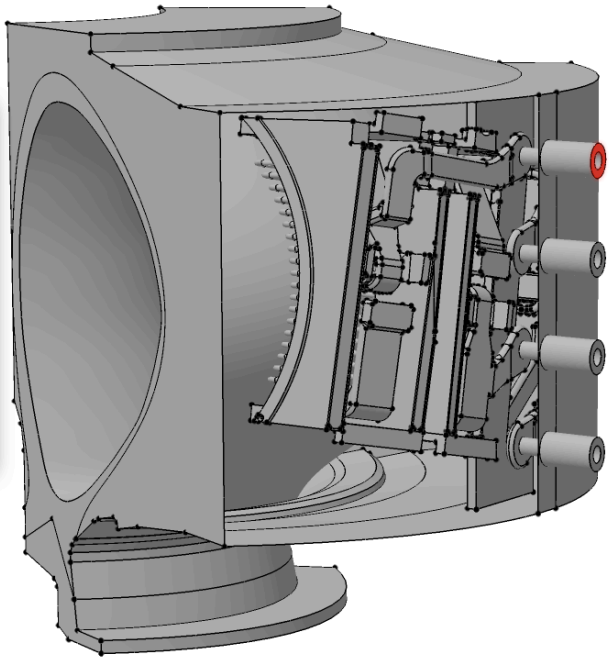
Functionalities needed

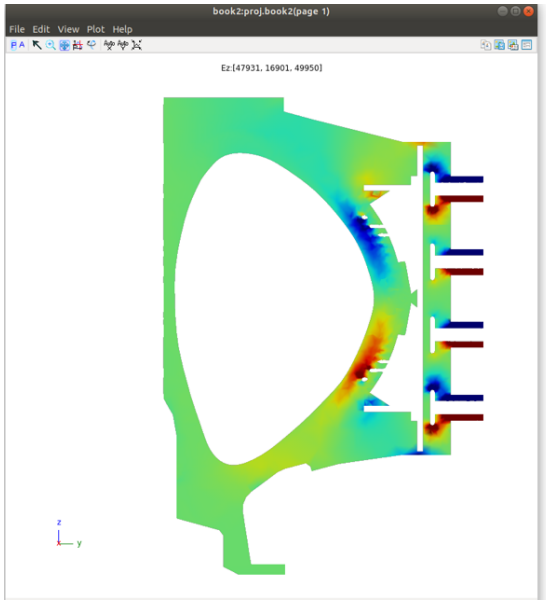
- Parallel meshes into MFEM
- Link with PetraM

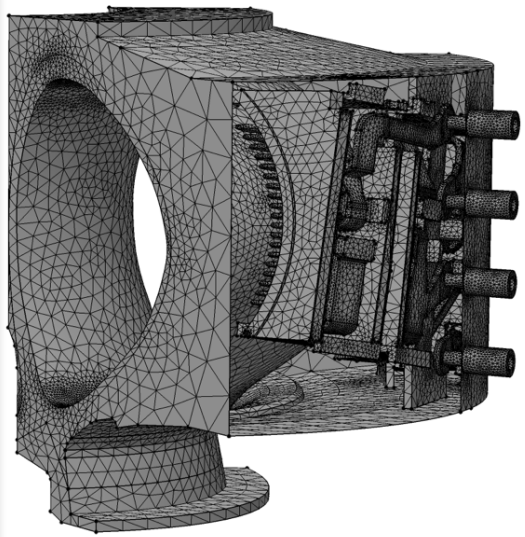
Current Status:

- In-memory integration of parallel PUMI with MFEM
- Integration with PetraM (some file transfer – in-memory underway)





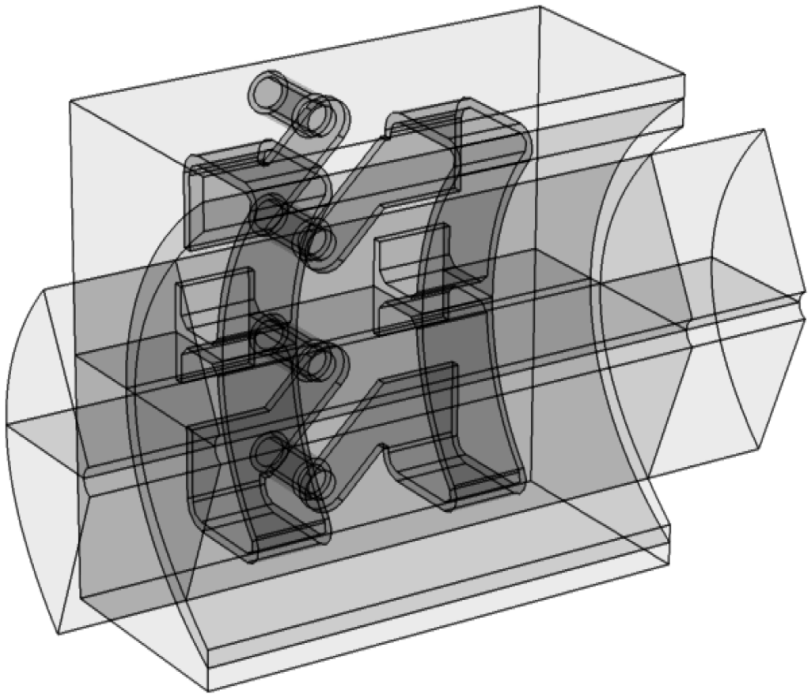


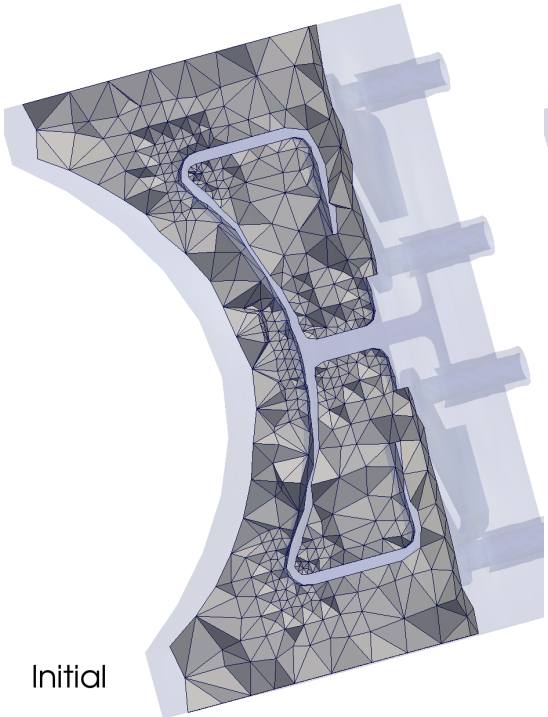


Adaptive Mesh Control

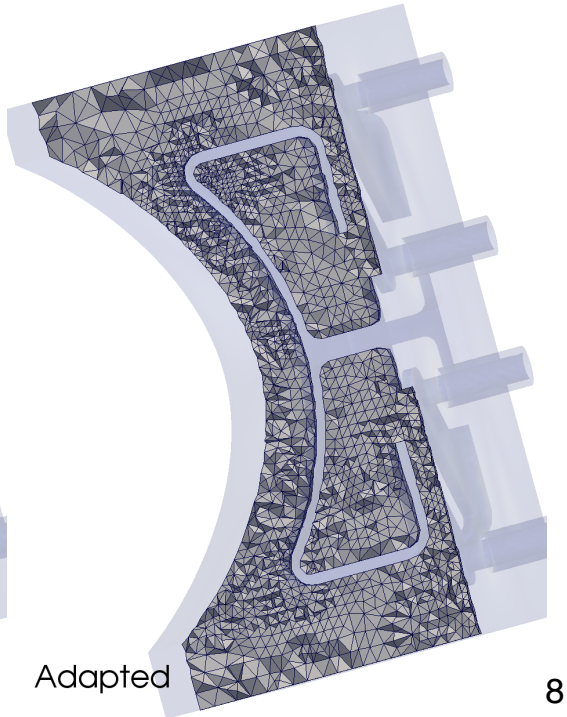
Tools used: PetraM, MFEM, PUMI/MeshAdapt, GeomSim

- Patch recovery error indicator
  - A version specific to Nedelec elements is under development
- Integrated with the PetraM/MFEM/PUMI/MeshAdapt
- Example: D-Port Antenna for C-MOD
  - 2-straps, driven and 80MHz
  - Core plasma is partial





Initial

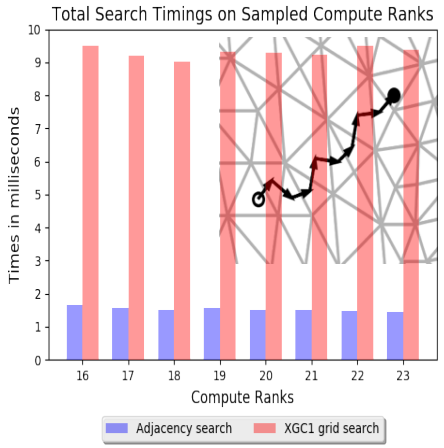


Adapted

Parallel Unstructured Mesh for Particle in Cell

Multiscale applications combining particle and continuum

- Fusion SciDACs want this for complex geometries

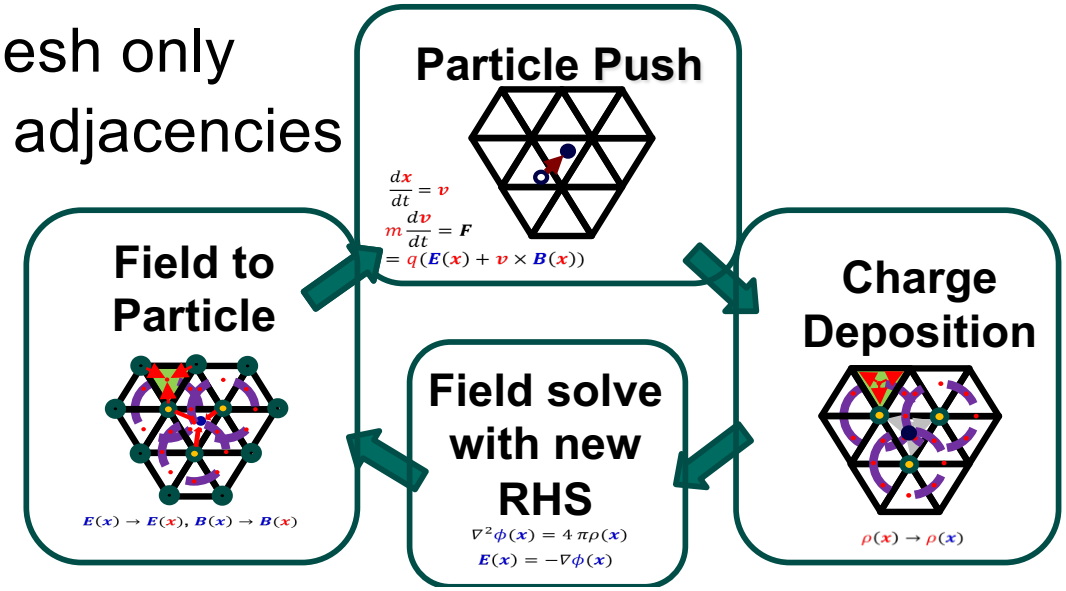


Approach used in current PIC codes

- Entire mesh data on each process
- Particles point to mesh – need grid search/sort
- Scalable wrt number particles but not wrt mesh

Mesh-based approach being developed

- Mesh is key structure – can be distributed
- Particles accessed through mesh only
- Particle search through mesh adjacencies
- Effective coupling to mesh-based PDE solvers



Parallel Unstructured Mesh for Particle in Cell

Mesh distributed in PICparts defined by

- Set of non-overlapping parts
- Plus sufficient buffer to keep particles on part in a push

Original version employed element level linked buffer

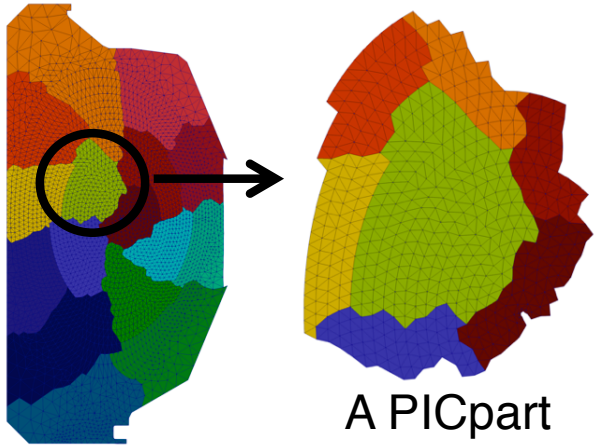
- Substantial memory overhead
- Being used in initial version of XGCm edge plasma – to be swapped out for modified one

Modified approach focused on more compact data and GPU execution

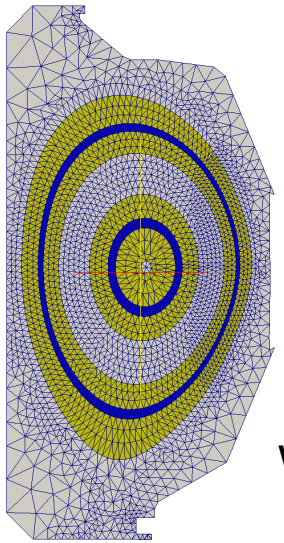
- PICpart is a "part" and sufficient neighboring parts
- Memory requirements reduced
- Being used in GITRm impurity transport

Both approaches support

- Migration of particles between pushes
- Dynamic load balancing



A PICpart



Two PICparts defined in terms of whole parts

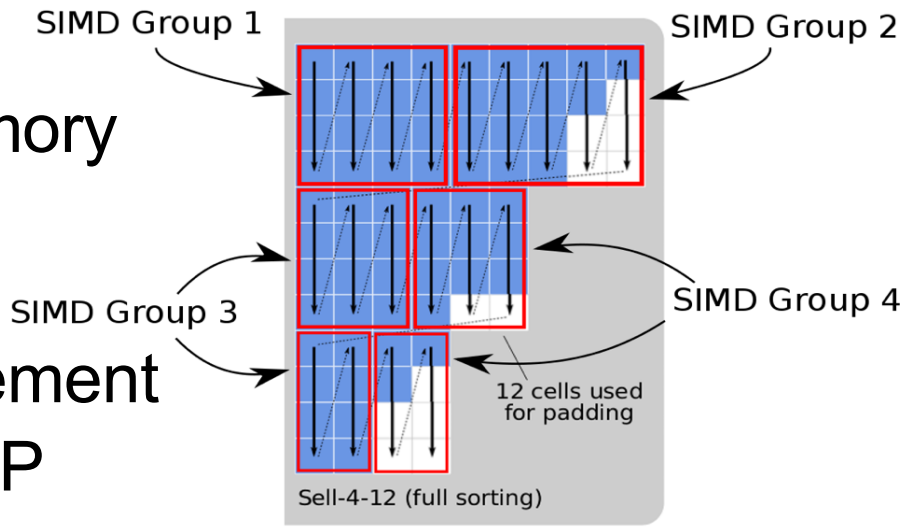
Mesh and Particle Data Structures

Mesh data structure: Omega

- GPU enabled mesh topology data structure
- BFS-like algorithms for effective local serial performance
- Space filling curves to support parallelization
- On-node OpenMP or CUDA parallelism using Kokkos

Particles structures critical for GPU performance

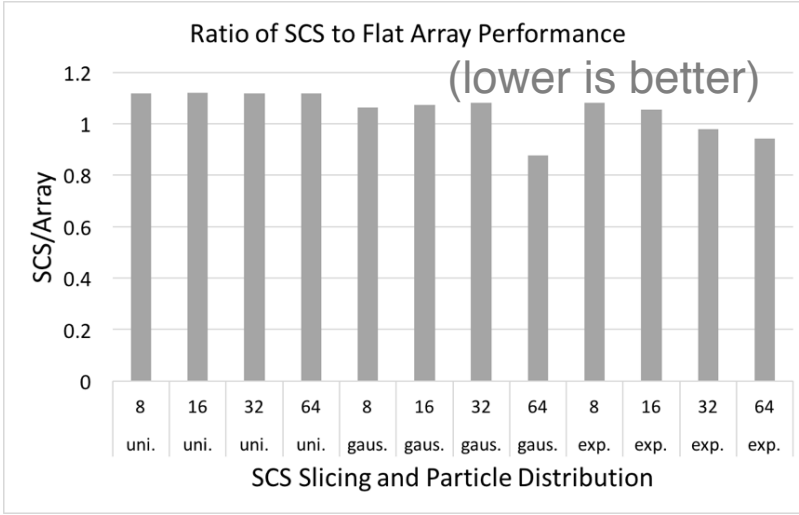
- Particles related to elements
  - Flat arrays use excessive memory due to variable numbers of particles/element
- Sell-C-σ defined for particles/element
- Coordinating with the COPA ECP particle co-design center



Testing of New Data Structures

Compute random particle position using parent 'element' field

- Proxy array for scalar element field
- 134M particles and 131,072 elements
- SCS is within 20% of flat array performance, faster with exp. dist.

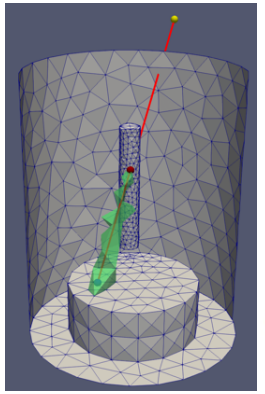


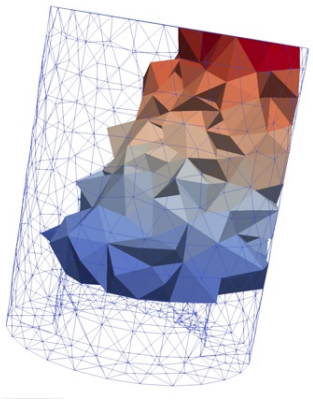
GPU: GTX 1080ti Code: C++ and Kokkos

Basic PIC test

- Executed 20 push, search, rebuild iterations in PISCES mesh (~6k tets) on a GPU
- Particles initialized off of bottom inset face
- One process and one GPU on RPI DCS System – node P9 (x2) and V100 (x4)

	no sorting	full sorting
ptcls (Ki)	time (s)	time (s)
128	2.298661	3.642041
256	2.895464	3.415048
512	3.79263	3.851178
1024	4.972283	4.090044
2048	7.089673	4.389198
4096	11.578984	4.799475





(left) Timing results. (center) Path of a particle through mesh. (right) Elements colored by iteration when particles enter; 0=blue, 20=red.

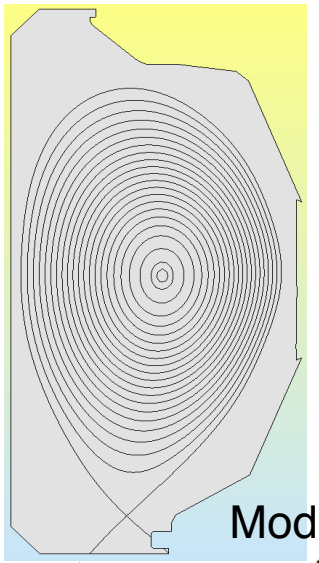
PUMIpic for XGCm Gyrokinetic Code

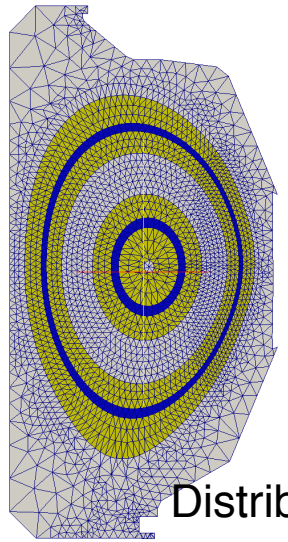
XGC uses a 2D poloidal plane mesh considering particle paths

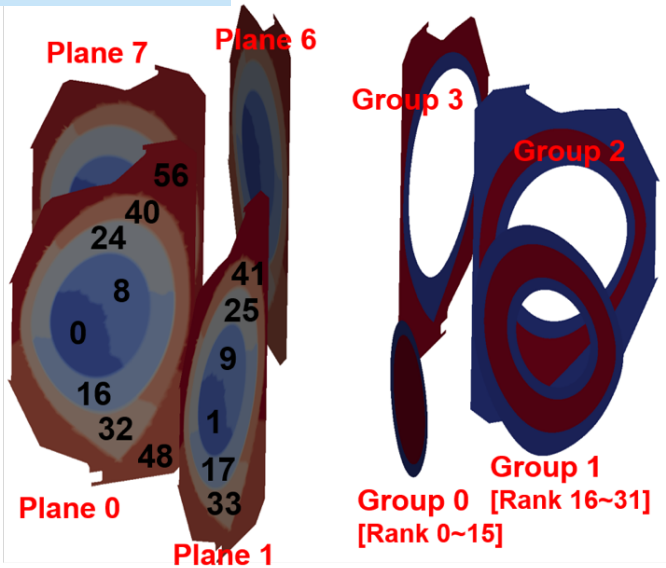
- Mesh distribution takes advantage of physics defined model/mesh
- Separate parallel field solve on each poloidal plane

Current implementation based on original PICpart definition

- Mesh distribution, adjacency search, particle migration, field transfers and parallel field solve implemented
- Dynamic load balancing approach defined
- Grouping structure introduced for effective parallel distribution of particles over repeated PICparts



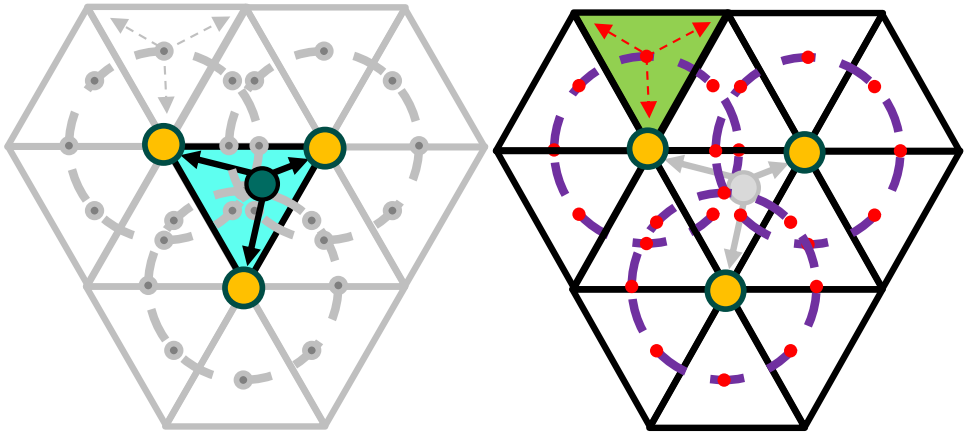




Two-level partition for solver (left) and particle push (right)

Example XGCm Operation: Field Transfers

- XGC gyro-average scheme for Charge-to-Mesh
  - Pre-computed gyro-ring weight functions
  - Scattering marker particle weights to vertices (left figure) → scattering gyro-ring samples of each "vertex" to vertices of element that contain the sample (right figure)
  - Scattering factors in the latter process pre-calculated once at setup phase with uniform gyro-radius grid
- Must do communication for sums
- Particle values depend on fields on bounding poloidal planes
- Mesh level reductions and field association use PUMI
- XGCm safe zone needed to be set so gyro-ring with max. radius is on process



Charge-to-Mesh

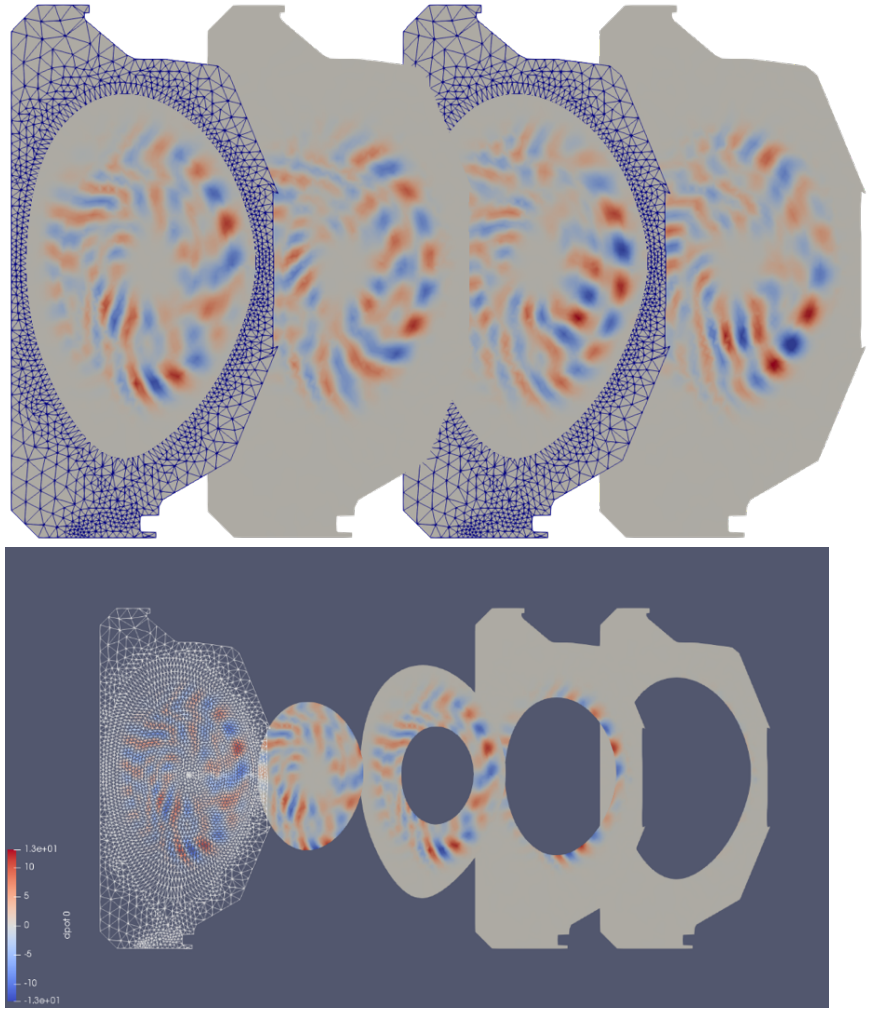
XGCm Edge Plasma Code based on PUMIpic

Implementing XGC physics and numerics

- Since data structures are changed code being rewritten in C++

Status

- Based on original PUMI structures – GPU focused structures will be integrated later
- Mesh/particle interaction operations
- Mesh solve
- Ion and electron push (including subcycling)
- Initial δf simulations
- Performance evaluation and improvement underway



Snapshot of electrostatic potential fluctuation (a) at toroidal angle ζ=0,π/2,π,3π/2 from left to right and (b) in local domain of each group at ζ=0

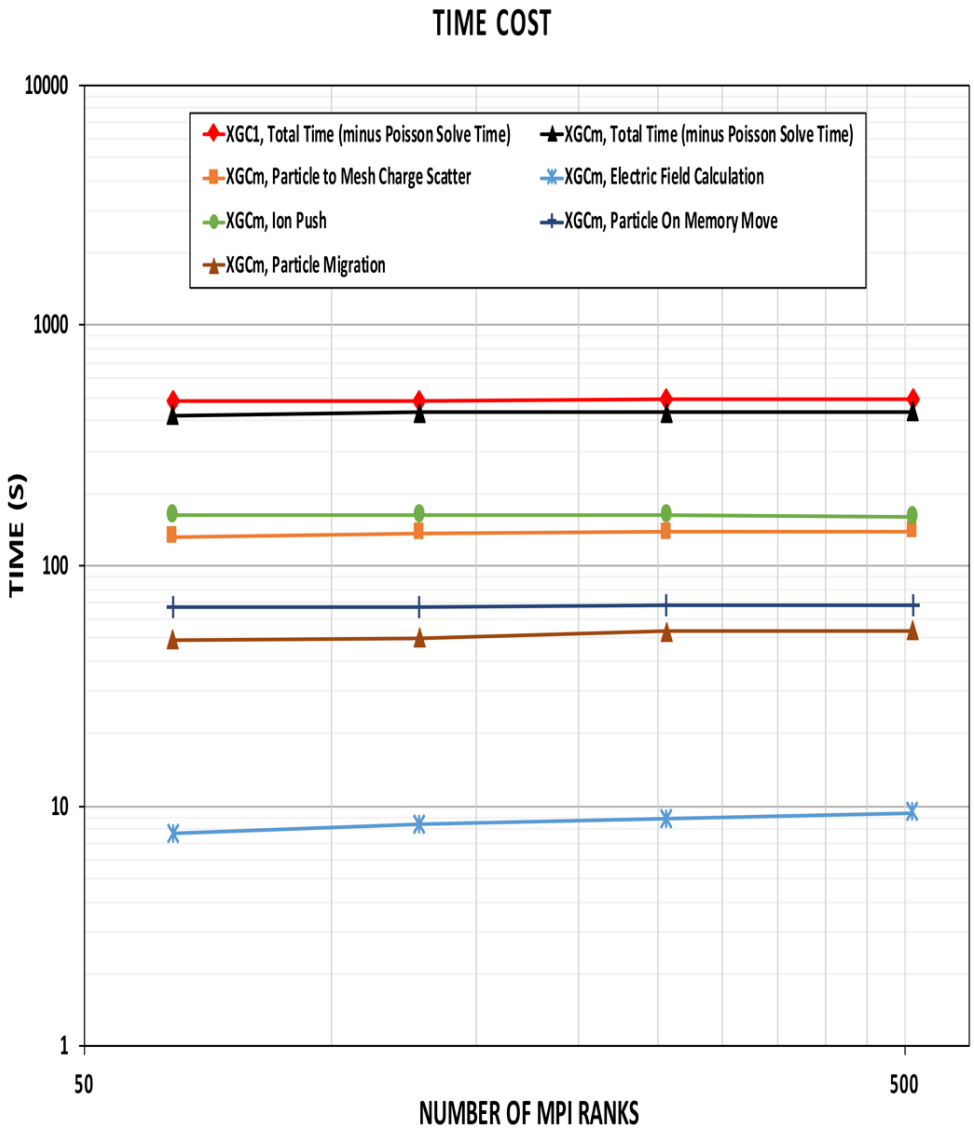
Initial XGCm Test Problem Results

Small δf test case – 6.2K elements, 8 poloidal planes

- 6201 mesh elements (triangles)
- 300k particles/rank, 64-512 ranks, 1-8 Cori KNL nodes
- Total timings (minus FE solve) equivalent to XGC1

Medium δf test case – 127K elements, 8 poloidal planes

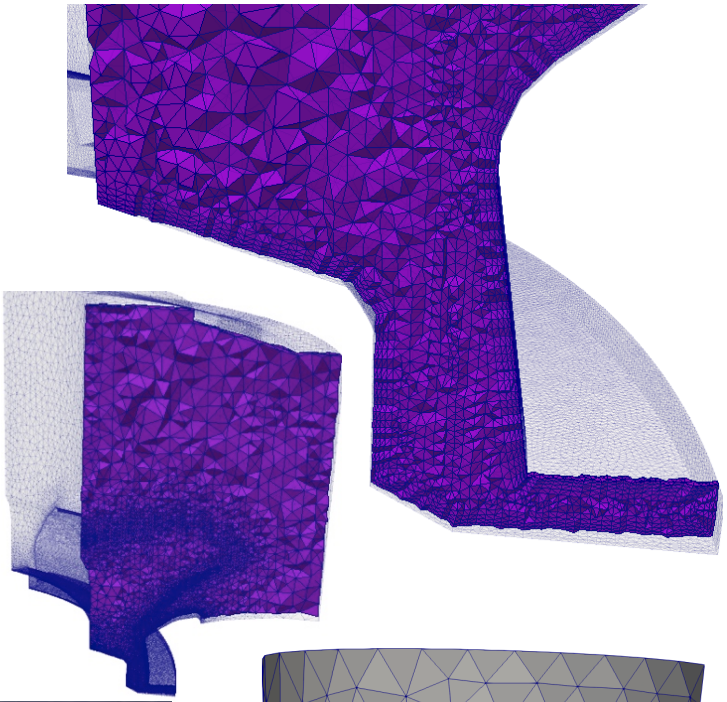
- 300k particles/rank, 256-1024 ranks, 4-16 Cori KNL nodes
- Push operation 25% faster than XGC (key and encouraging)
- Currently total time greater than XGC – need new mesh/particle structures to eliminate slow steps

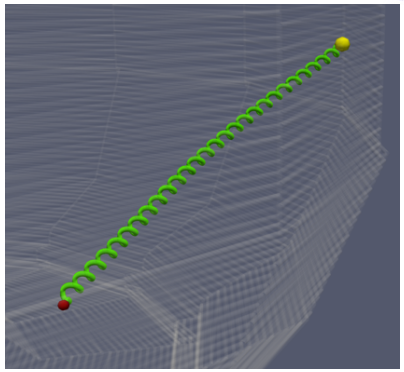


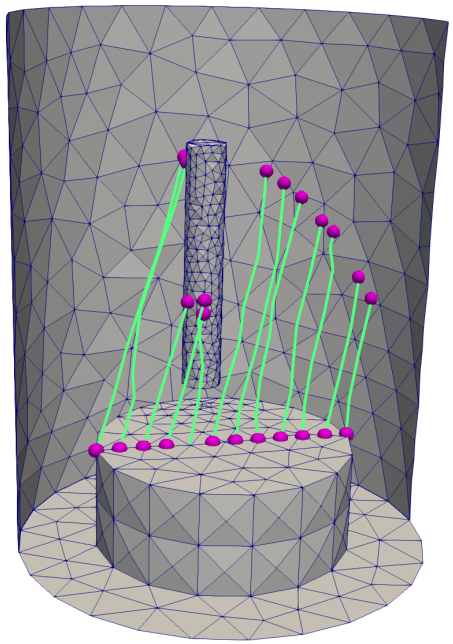
Mesh Based GITR: GITRm

Operates of 3D graded meshes

- 3D adjacency search
  - Effective on graded meshes
- Wall intersection implemented
  - Employ mesh entity classification for efficient operation
  - Elements where distance tracking preprocessed to know when needed
- GITR Boris move implemented
- Omega mesh and new particle data structures
- Initial tests, including GPU execution, underway
- Working with U. Tenn and ORNL on to define appropriate verification tests







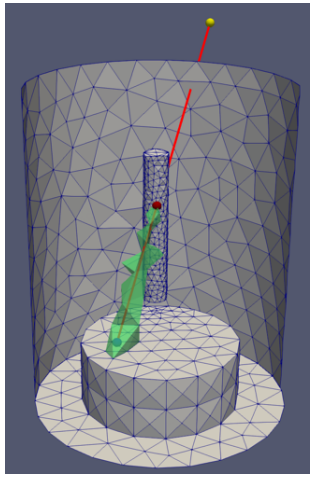
Mesh-based GITR: GITRm – Preliminary Tests

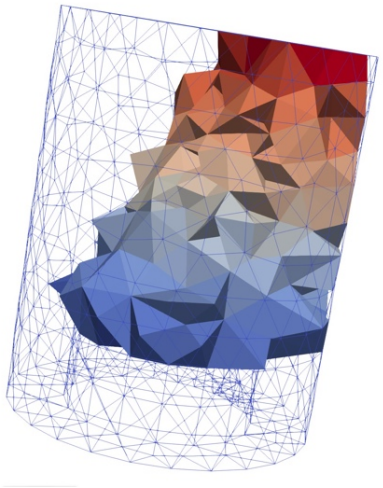
Executed 20 push, search, rebuild iterations on PISCES mesh (~6k tets) on a GPU

- Results comparable to those executed on background grids

Developments for larger particle counts and multiple processes:

- Bulk communication layer w/MPI wrapper for MPI, CUDA aware MPI, and possibly NCCL (initial version in testing)
- Remaining and new physics
- Parallel PICpart creation (done)





	no sorting	full sorting
ptcls (Ki)	time (s)	time (s)
128	2.298661	3.642041
256	2.895464	3.415048
512	3.79263	3.851178
1024	4.972283	4.090044
2048	7.089673	4.389198
4096	11.578984	4.799475

(left) Path of a particle through mesh. (center) Elements colored by iteration when particles enter; 0=blue, 20=red. (right) Timing results

Closing Remarks

- Key advances in the last year:
  - Simulation workflow for RF simulations with full geometric complexity
  - Unstructured mesh infrastructure for PIC calculations – emphasis on structures suitable for GPUs
  - XGCm edge plasma code timings are encouraging
  - GITRm code uses new GPU ready mesh and particle structures
  - Initial computations being produced
- We are interested in helping additional fusion, and other, SciDACs on unstructured mesh related needs
  - Working on specific meshing tool for hPIC code to support sheath/presheath simulations
  - Very preliminary discussions of alternative FE technologies for core plasma disruption simulations
- Additional FASTMath Fusion Partnership activities
  - Conservative solver for runaway electron kinetics (SCREAM)
  - Scalable MHD solver (TDS)

Poster on FASTMath activities with SciDAC Fusion Partnerships on Thursday

Support Acknowledgements

- "Unstructured Mesh Technologies for Fusion Simulation Codes", contract # DE-SC0018275, to support fusion SciDACs including:
  - Center for Tokamak Transient Simulations, Steve Jardin, PPPL
  - Partnership Center for High-fidelity Boundary Plasma Simulation, C.S. Chang, PPPL
  - Center for Integrated Simulation of Fusion Relevant RF Actuators, Paul Bonoli, MIT
  - Plasma Surface Interactions: Predicting the Performance and Impact of Dynamic PFC Surfaces, Brian Wirth, ORNL
- Subcontract from LLNL as part of "Frameworks, Algorithms and Scalable Technologies for Mathematics (FASTMath) SciDAC Institute", contract # DE-AC52-07-NA27344
- Subcontract from Simmetrix on their DOE SBIR "Unstructured Mesh Technologies for Massively Parallel Simulation and Data Analysis of Magnetically Confined Plasmas", Phase IIb SBIR contract # DE-SC0013919