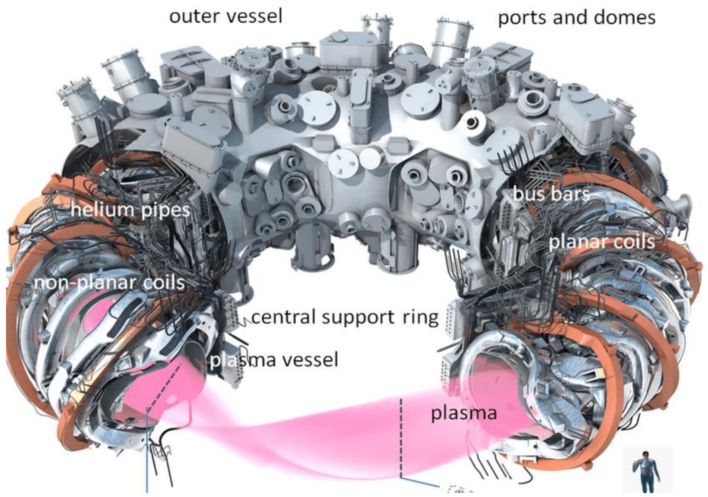
Tools to Support Geometry and Meshing Needs for Fusion Energy System Simulation Codes

M.S. Shephard, J.S. Merson, O. Sahni, C.W. Smith, B.J. Sturdevant, D.D. Nath, U. Riaz, F.H. Hasan, A.E. Castillo-Crooke, A.Y. Joshi
Scientific Computation Research Center, Rensselaer Polytechnic Institute

M.W. Beall, O. Klaas, R. Nastasia, S. Tendulkar
Simmetrix, Inc

December 12, 2025

Geometry and Meshing



Critical to ensuring that fusion energy systems will succeed

- Integration of simulations that account for the levels of geometric fidelity needed, particularly for plasma facing components
- Employ CAD technologies to define the analysis geometries – effort needed to coordinate with the CAE to provide the needed tools
- Unstructured mesh methods only reasonable means to address simulation workflows with complex geometries

When using the full set of available simulation automation technologies

- The generation and control of good meshes given a properly defined analysis geometry is possible using available tools
- The generation of the proper analysis geometry is a challenge

There is a Simmetrix/RPI/LLNL ASCR SBIR addressing some of these geometry construction needs including

- Automatic clean-up of CAD models to ensure meshes can be automatically generated
- Geometry and meshing for stellarators
- Working with SciDACs and fusion energy companies on supporting simulation workflows needing complex analysis geometries

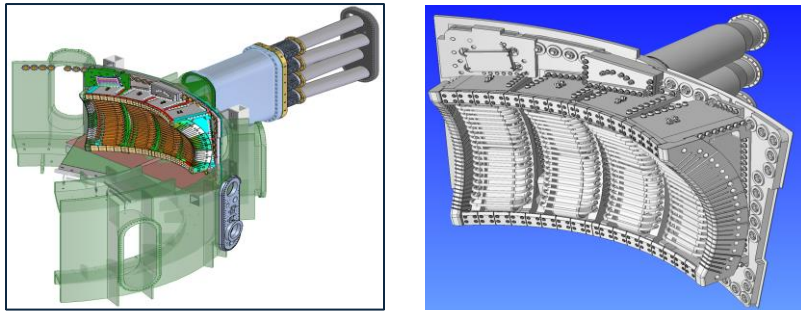
Example Analysis Geometry Construction

Desired properties of the analysis geometry for RF and impurity transport simulations

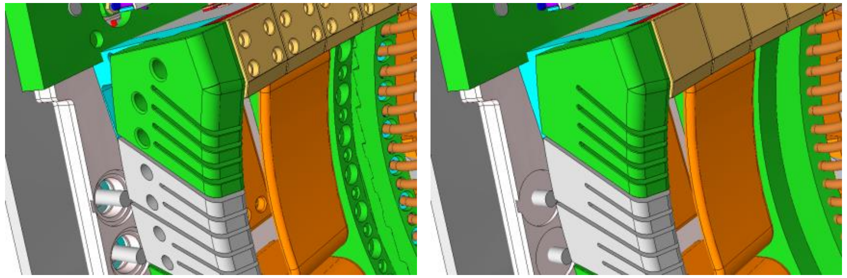
- Represents the physical domain to the level of geometric fidelity required
- Can include “physics” geometry of importance to the simulation process (example flux curves)
- Can be constructed from multiple input sources including CAD models, functions on grids, image data, mesh data, etc.
- Is a complete geometry in a form that can support the application of automatic mesh generation

Approach

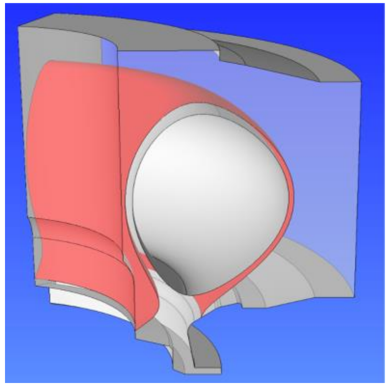
- Employ tools that support the combination of the various forms of input geometry into produce complete boundary representations as supported by CAD systems
- The Simmetrix GeomSim Core and SimModeler support these operations



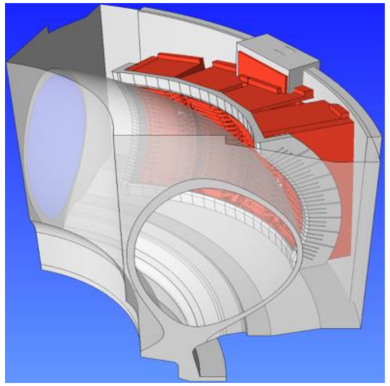
Extracting antenna assembly geometry



Efficient elimination of unwanted components



Add Physics geometry



Assembled Geometry

Example of applying Simmetrix analysis geometry construction technologies

Automatic Mesh Generation

Desired properties of the mesh generators

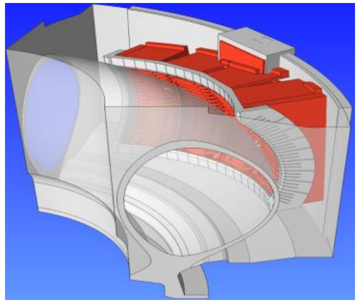
- Produce optimal meshes for the specific analysis package
- Can be automatically generated given analysis geometry and mesh control information defined on the entities in that model
- Are well designed to provide the desired level of accuracy

A problem

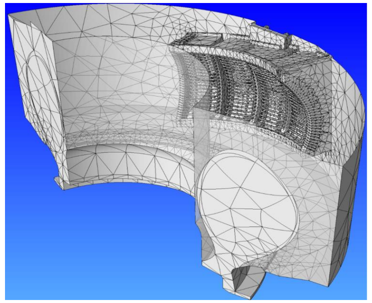
- There are no tools for the automatic construction of structured grids – typical mode of operation is to use pre-defined block layouts for each specific geometric configuration

Approach

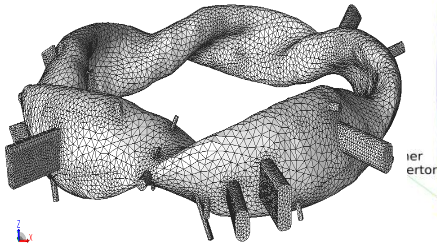
- Focus on use of unstructured mesh technologies and support interactions with structured meshes when coupling codes
- Employ automatic mesh generation procedures such as Simmetrix MeshSim or opensource tools such as GMSH
- Provide geometry construction and meshing tools building on CAD and automatic mesh generators for specific applications
- Employ adaptive mesh control to ensure mesh optimality when possible

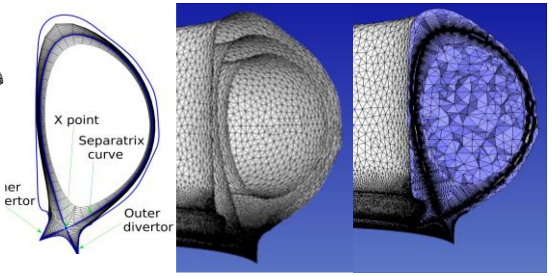


Analysis geometry

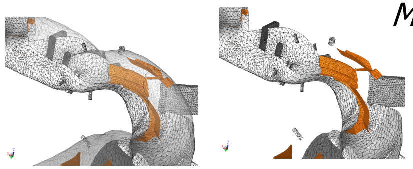


High-order curved mesh

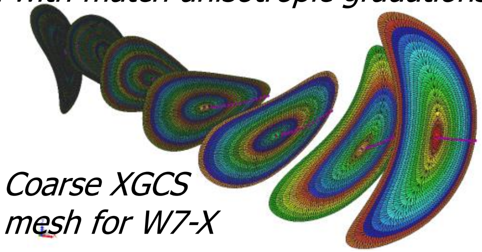




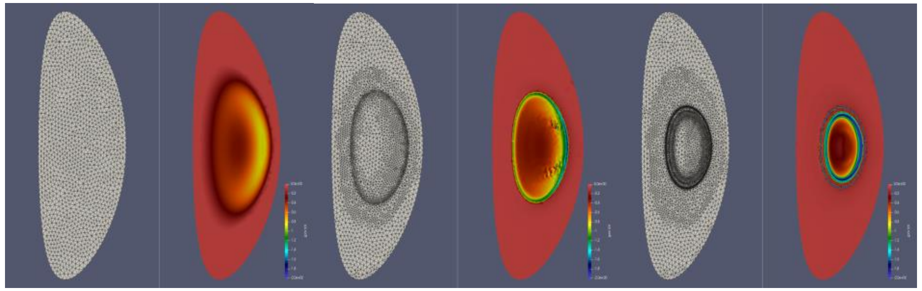
Mesh with match anisotropic gradations



Coarse 3D mesh for W7-X



Coarse XGCS mesh for W7-X



Adapted meshes for M3D-C1 pellet injection simulation

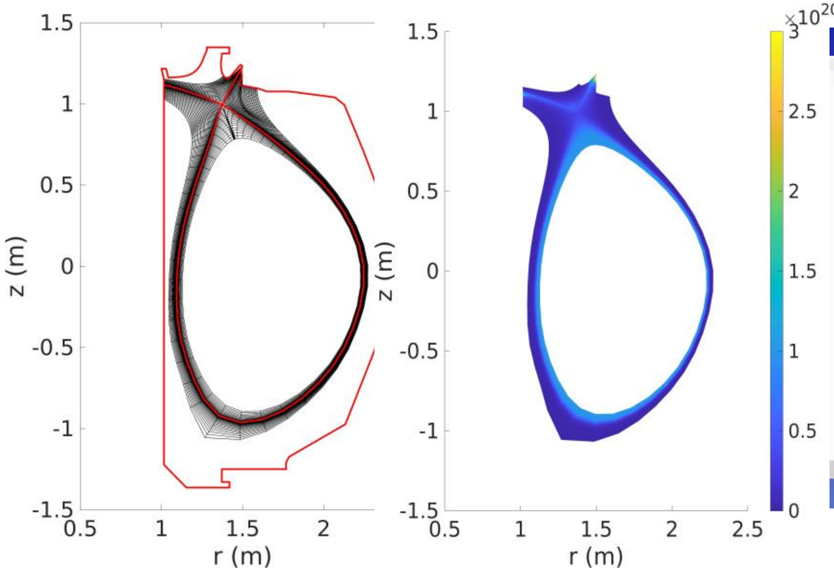
Matching Mesh with Coupled Meshes

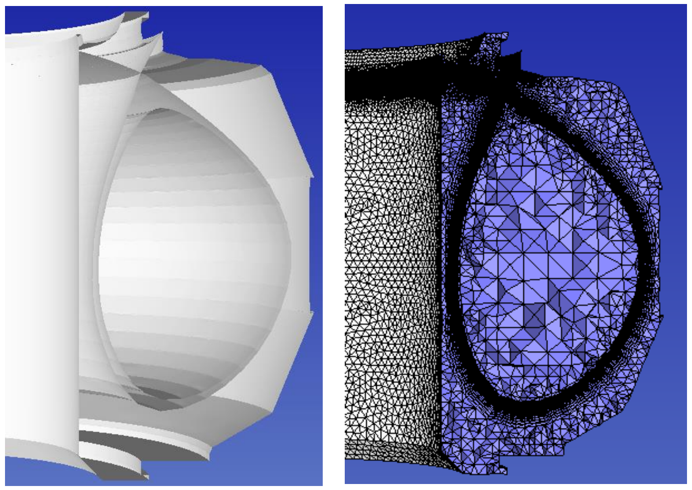
Many simulation require field coupling:

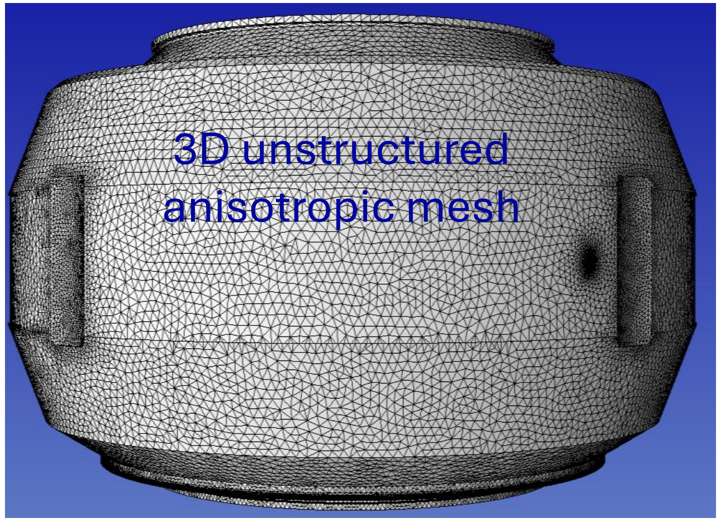
- Maintaining solution accuracy requires accurate field transfer.
- Accurate field transfer enhanced by matching mesh size and anisotropy.

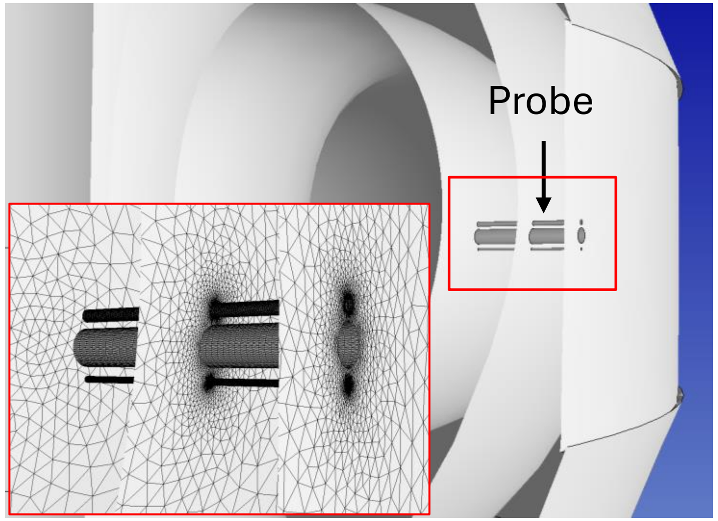
Example: GITRm requires background field from an edge plasma code:

- Edge plasma code mesh uses is block structured, field following, anisotropic and attached to separatrix.
- Include separatrix and selected flux surfaces in GITRm analysis geometry.
- Set mesh type in that area to boundary layer and match mesh size and anisotropy.
- Able to deal with full range of 3D geometry.







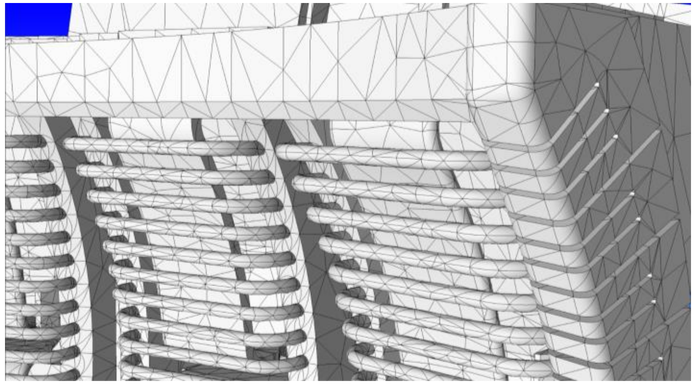


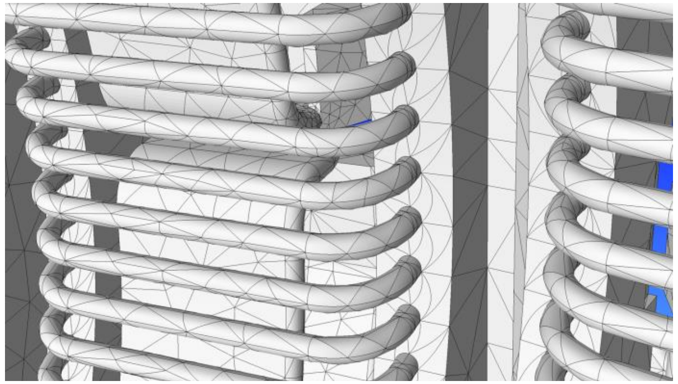
Splitting open flux curves faces

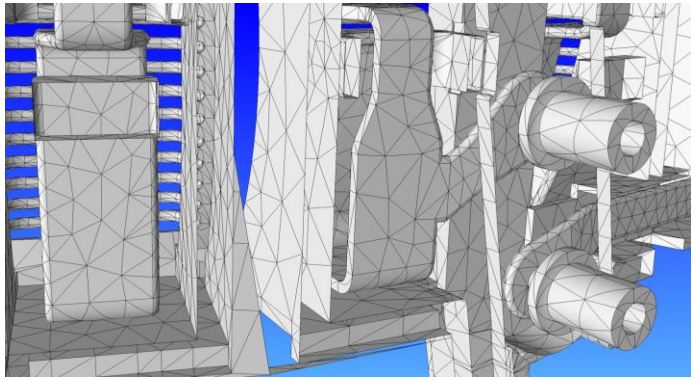
Mesh Generation to Support RF Simulation

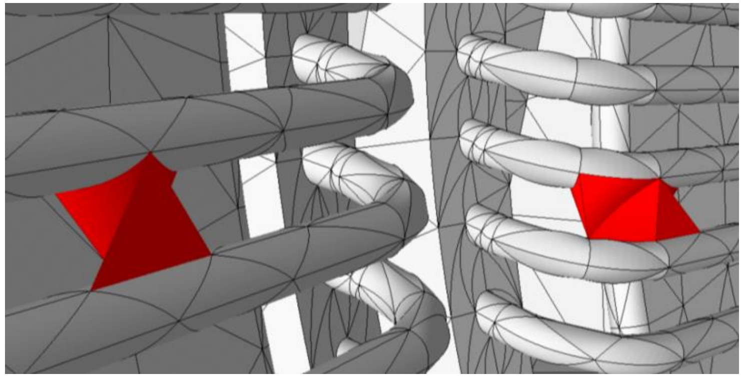
Time-harmonic Maxwell's equations solved using high-order Nédélec finite elements.

- High-order elements required generation of coarse curves meshes.
- Simulation done using MFEM using up to 5^{th} order elements.
- Curved meshes automatically generated using Simmetrix automatic mesh generation tools.

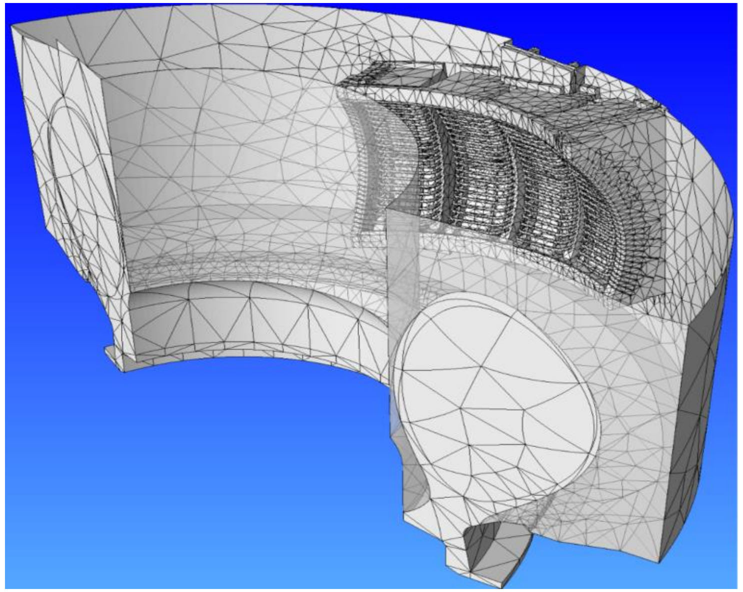


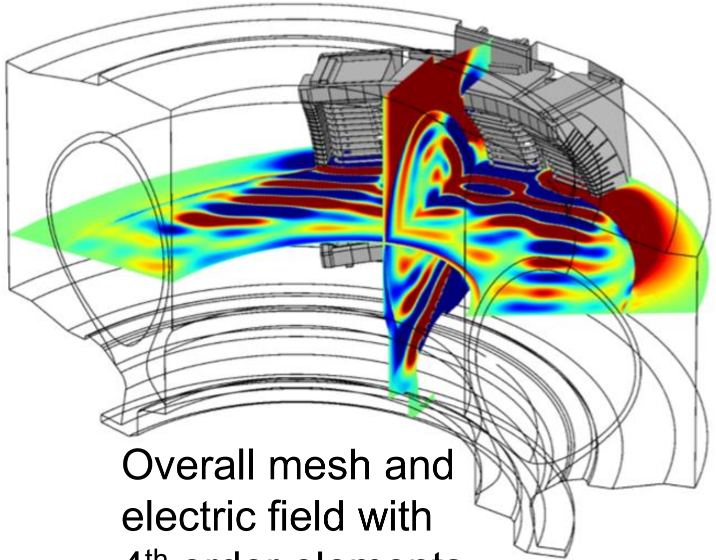






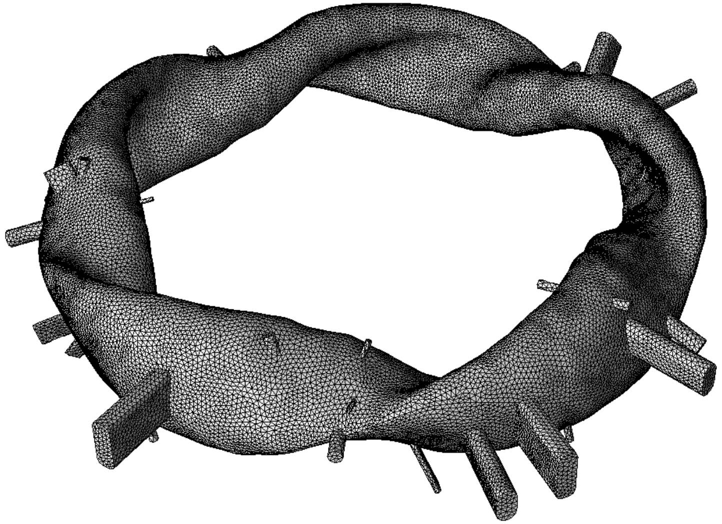
Close-up views of coarse curved mesh





Overall mesh and electric field with 4^{th} order elements

Current Efforts Related to Stellarator Geometry

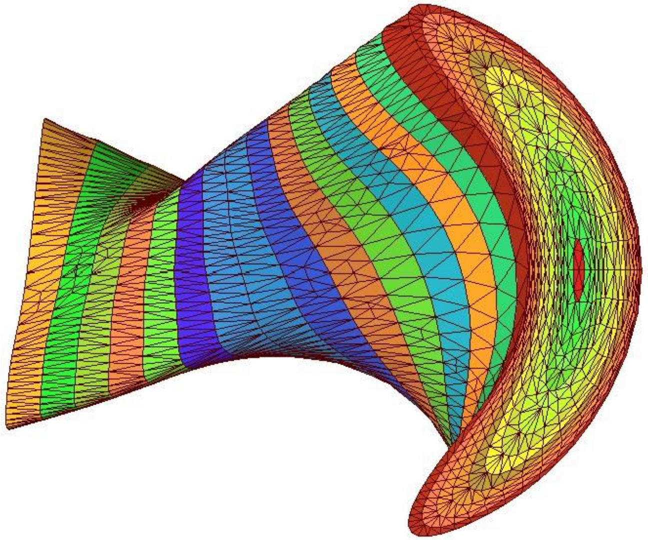


Sources of geometry information

- “Closed” magnetic flux surfaces: VMEC and DESC
- Outside the VMEX and DESC last flux surface is an open question
- Kisslinger geometry
- W7-X CAD files from MAX Planck Institute for Plasma Physics (IPP) https://www.ipp.mpg.de/5440442/cad_druck

Currently working on

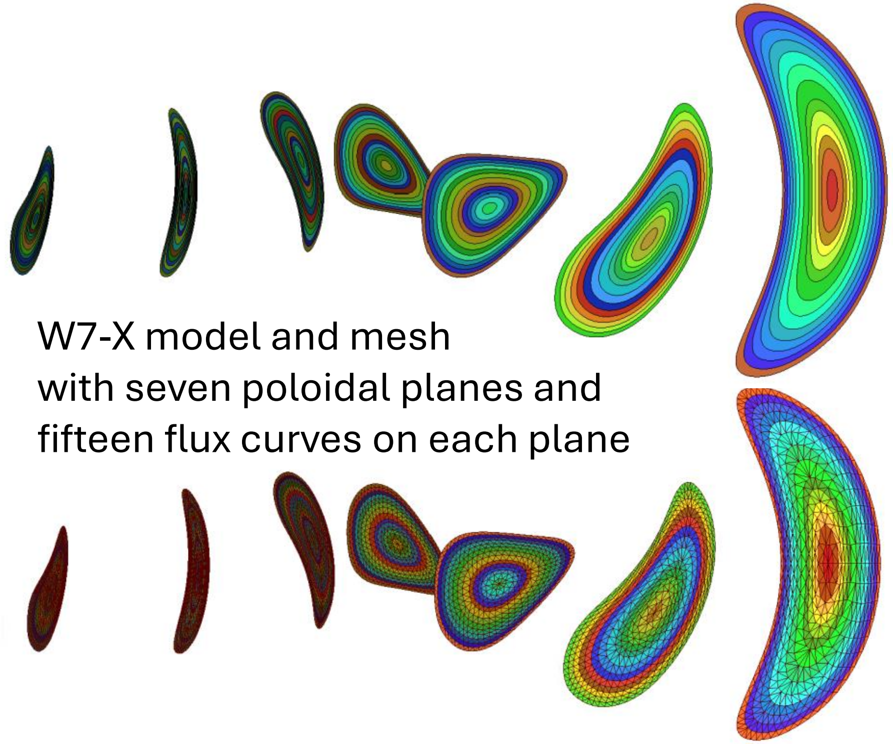
- Investigating using M3D-C1 to for magnetic flux field outside the last VMEX data
- Made CAD model, with ports added, from Kisslinger plate file geometry
- Support of island divertor design optimization
- XGCS point placement and meshes
- An initial interaction with Chris Hansen to provide meshes



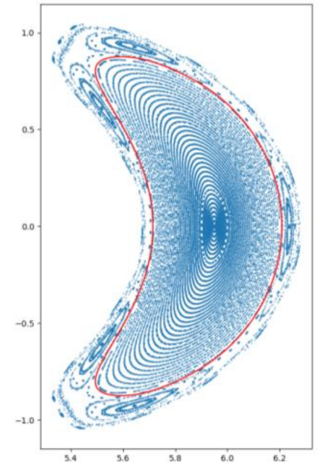
Want to develop more interactions with stellarator experts to provide geometry and meshing capabilities

Stellarator Meshing

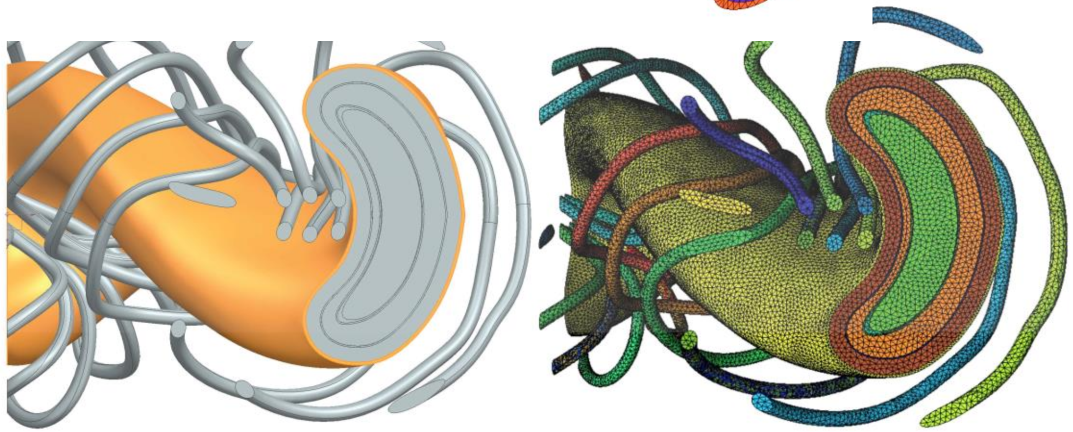
- For XGCS
  - For 1nside the last closed flux curve
  - Model geometry directly evaluated from VMEC Fourier coefficients
  - Field following points placed on flux curves
  - One element deep between flux curves
  - Outside the last closed flux curve
  - Use magnetic geometry from BMW outside LCFS
  - Challenges: Extracting flux field from magnetic fields in BMW, matching value on last closed flux curve
- Blankets and coils
  - Parameterized geometry from ParaStell
  - Automatic mesh generations including multiple elements through thin sections



W7-X model and mesh
with seven poloidal planes and
fifteen flux curves on each plane

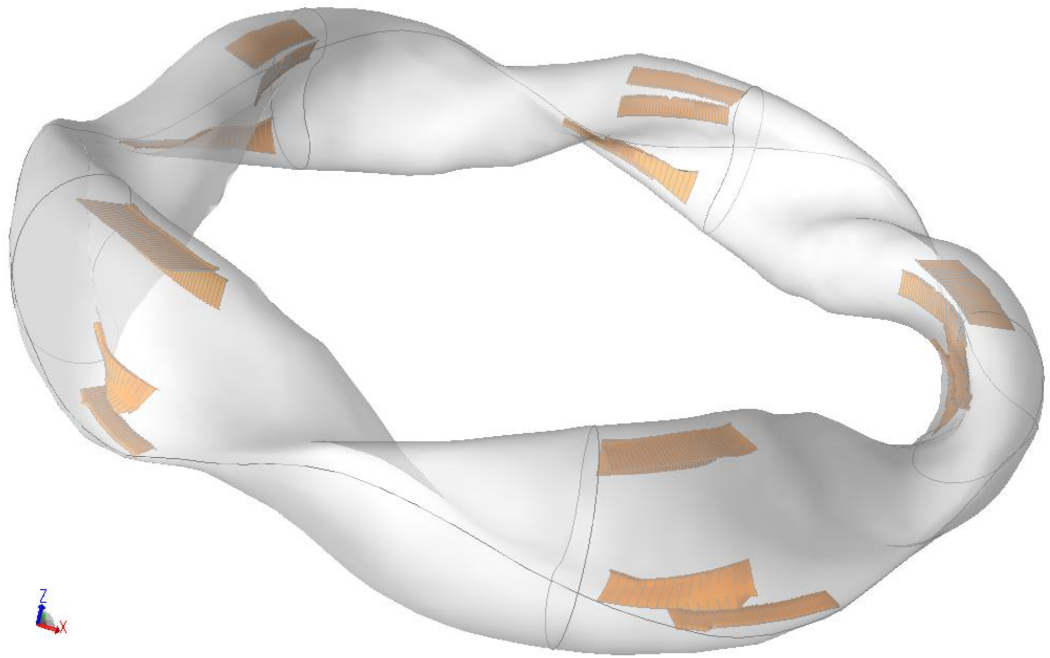


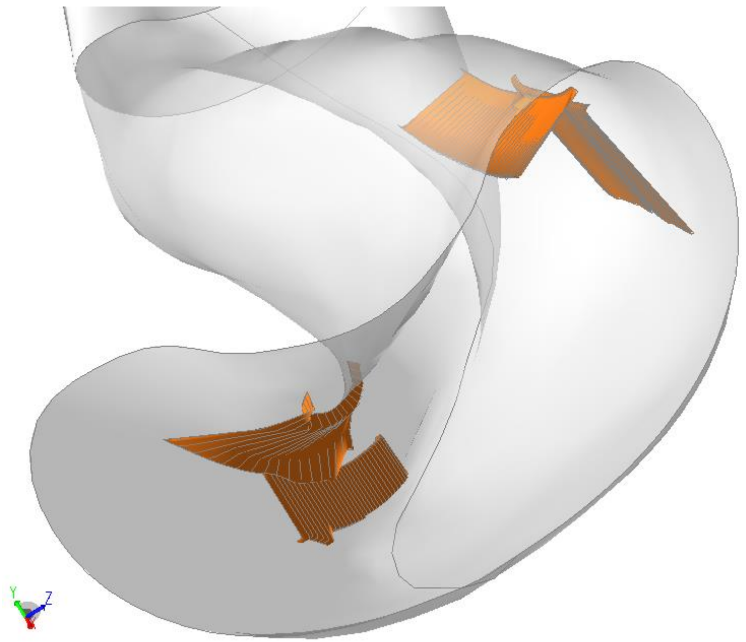
Poincare plot
shows close
match between
VMEC and BMW

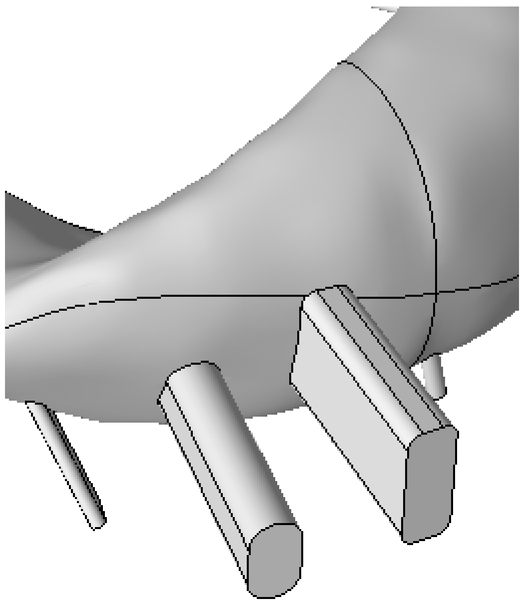


ParaStell Geometry and automatically generated mesh

CAD model of Kisslinger plate file geometry



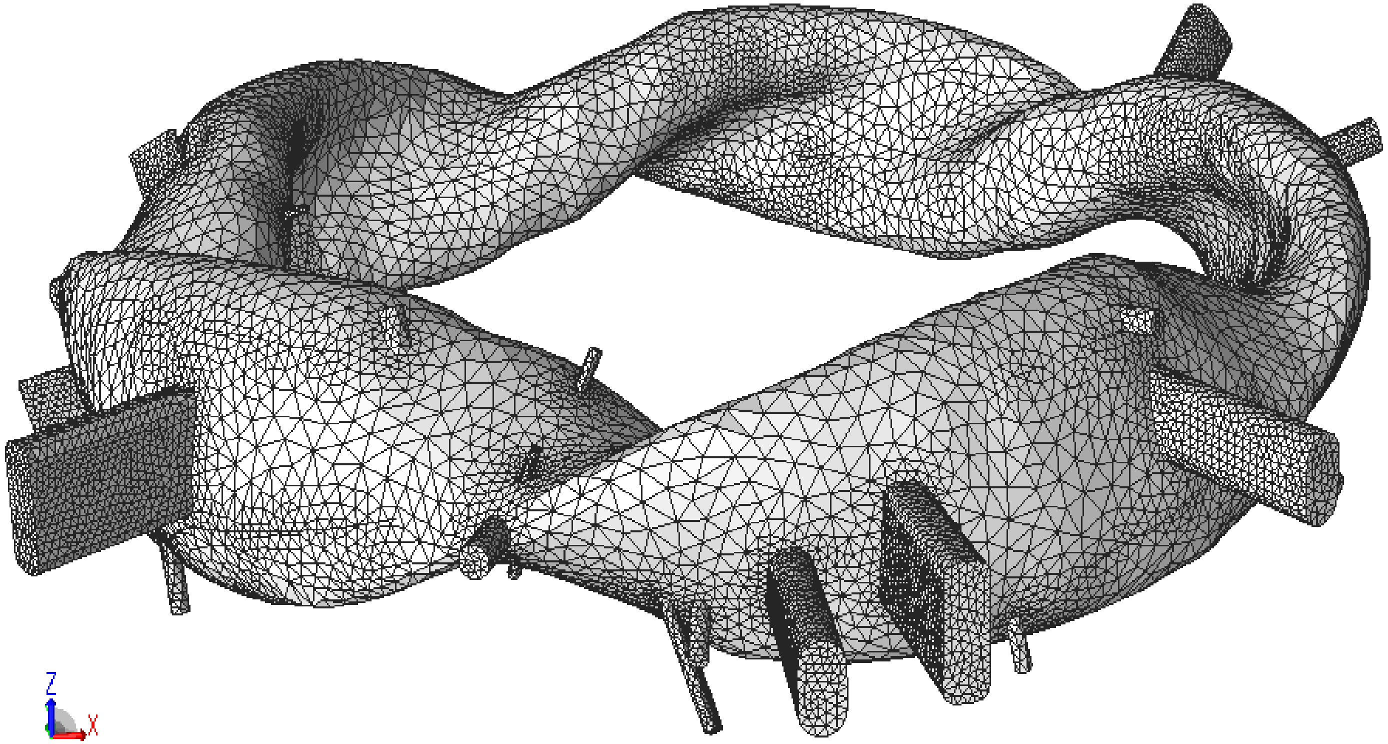


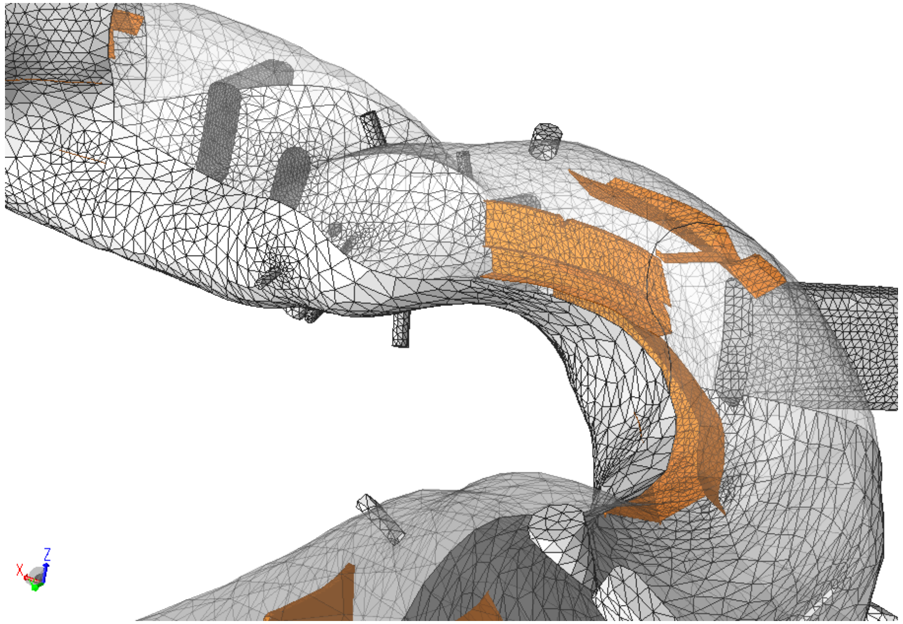


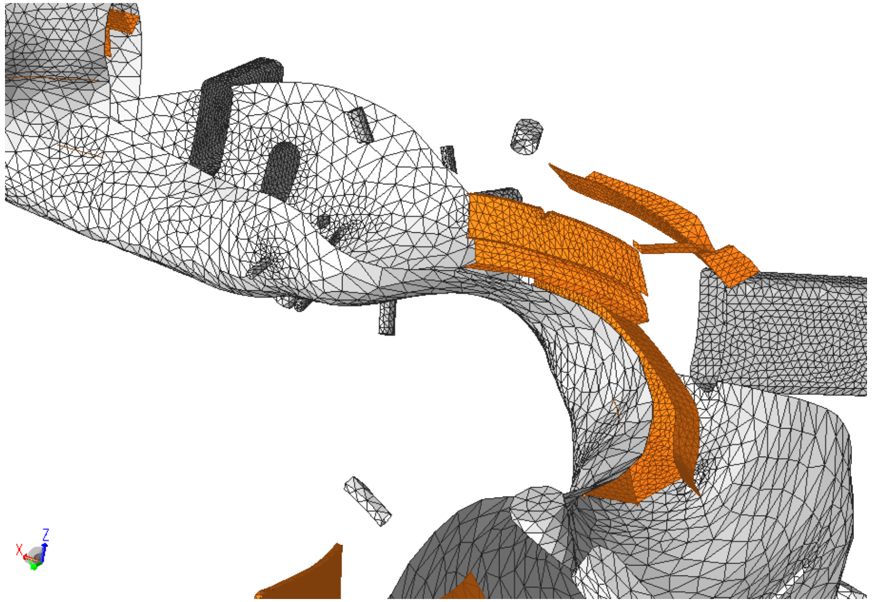
Four steps in CAD model construction

1. Outer surface created by lofting through given cross-sections resulting surface is lumpy – due to noise in cross section curves
2. Surfaces fitted to island divertors
3. Non-manifold solid created by filling the volume while maintaining the island divertor surface faces
4. Port geometry defined and unioned into CAD model based on ports seen in CAD models

Volume Mesh of the Non-manifold CAD Model







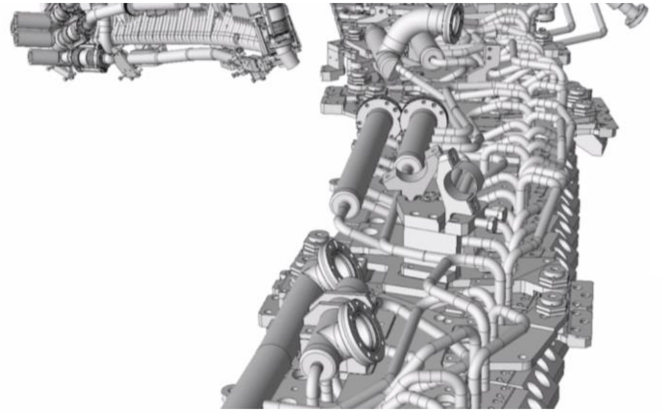
Above: Exterior view of the volume mesh

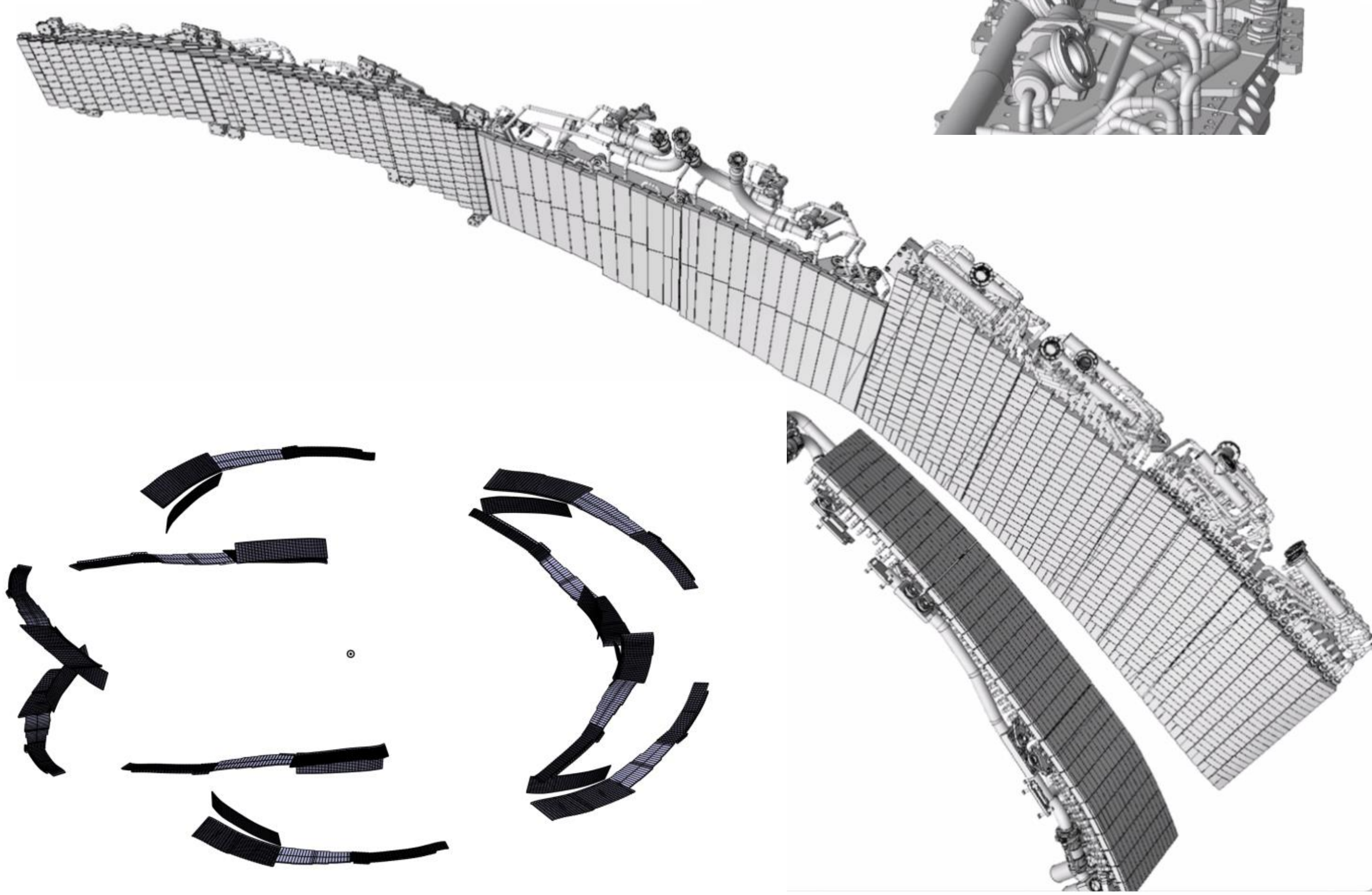
Right: Cut-away views of the mesh showing only the surface triangles from the volume mesh on select model surfaces

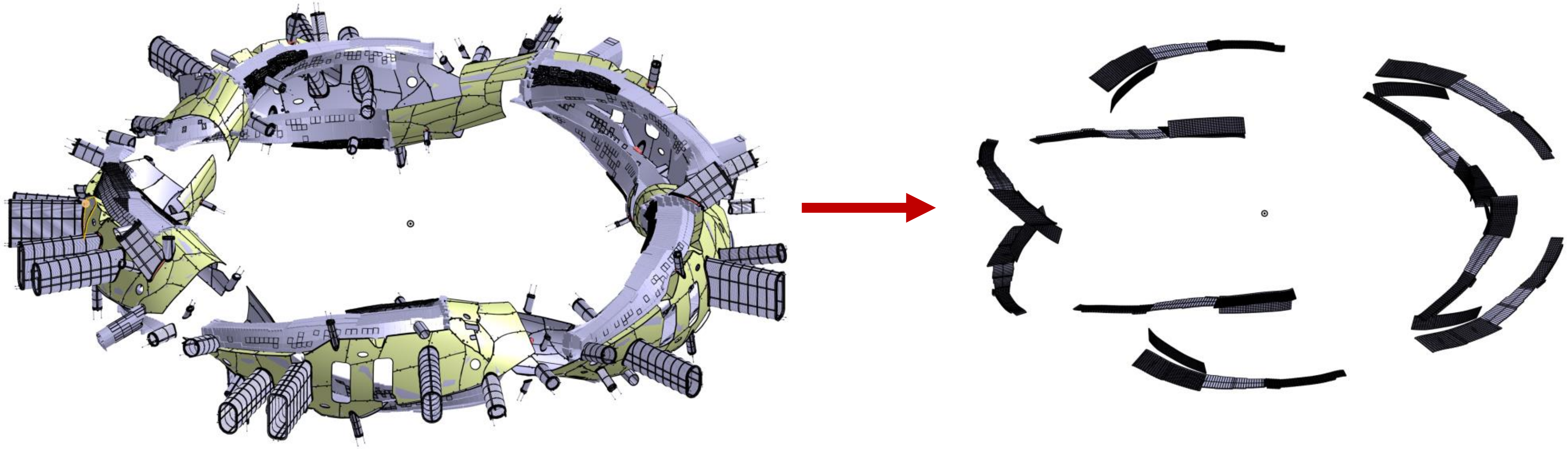
Where do we get the W7-X Island divertor geometry?

Interacting with Heinke Frerichs on Divertor Design Optimization

- Most reliable would be the CAD Divertor CAD files from IPP
- The images to the right are what is in those CAD files – files has thousands of parts
- Just need the divertor files
- Hard to extract from this CAD file
- Found a version in one of the other CAD files







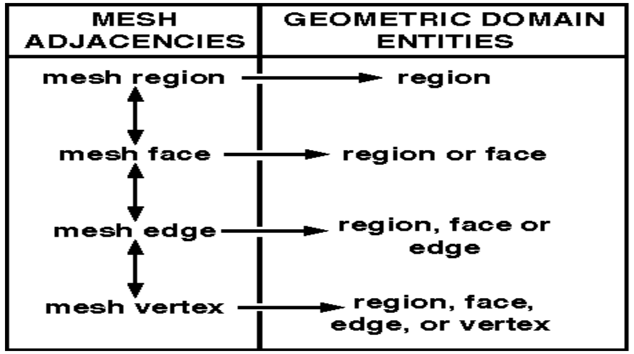
Infrastructures to Support Unstructured Mesh Operations

Desired properties of unstructured mesh infrastructure

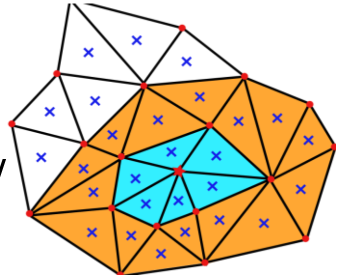
- Complete mesh infrastructures that efficiently supports any mesh-based operation
- API driven for in-memory integration into exascale codes
- Execute at scale on CPU's and GPU's

Approach

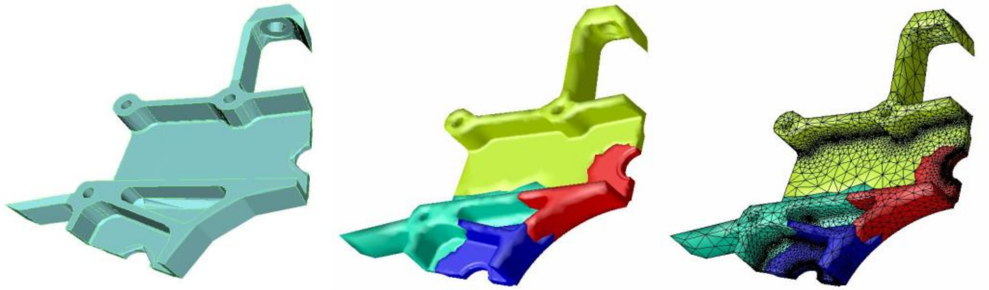
- Complete mesh infrastructures base on topology linked to analysis geometry via classification
  - Fully distributed to support MPI operations
  - PUMI - CPU version
  - Omega_h – GPU version
  - O(1) adjacency queries
  - Anisotropic mesh adaptation
  - Dynamic load balancing of meshes, particles etc.
  - Support a full range of solution transfer methods
  - Used as the infrastructure in M3D-C1 code
  - Provides mesh topology information to XGC code



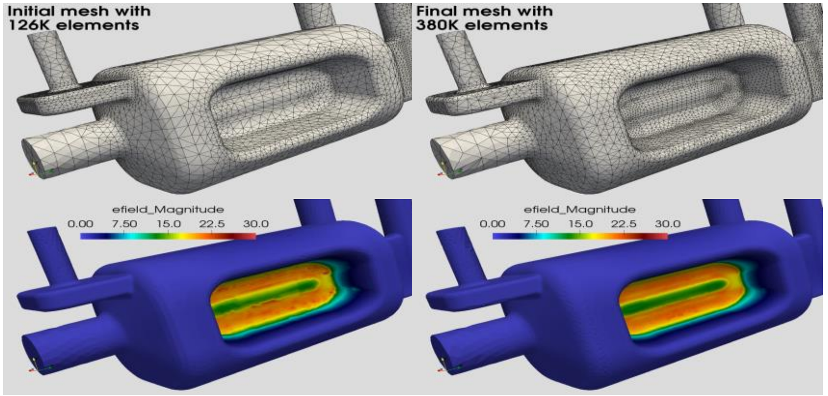
Mesh topology to support operations and relation to analysis geometry



Patch-based solution transfer

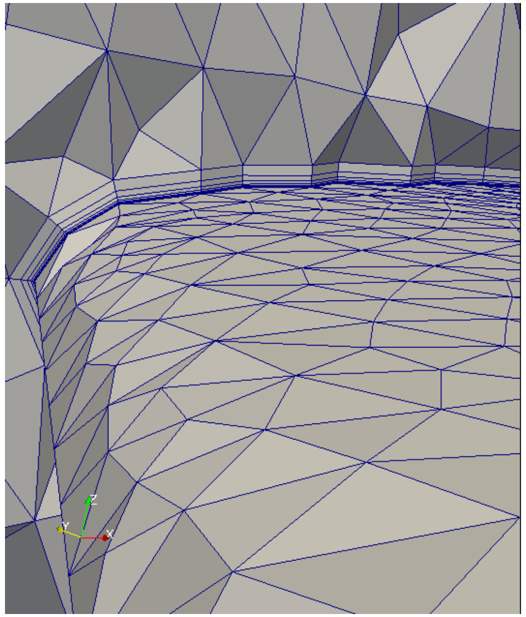


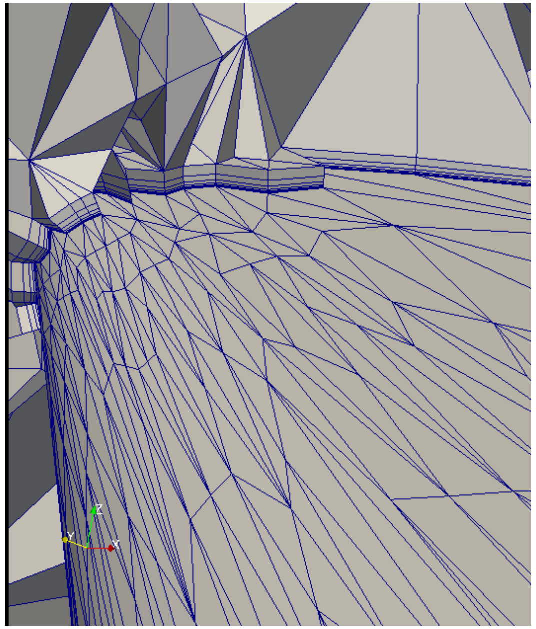
geometric model *partition model* *partitioned mesh*

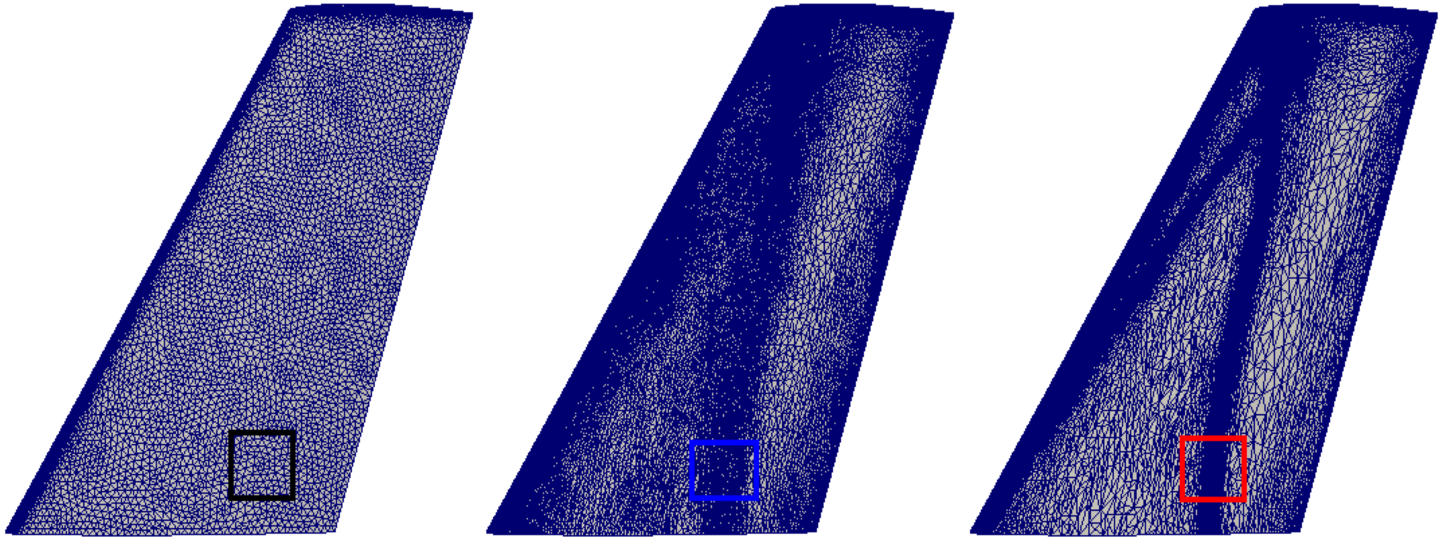


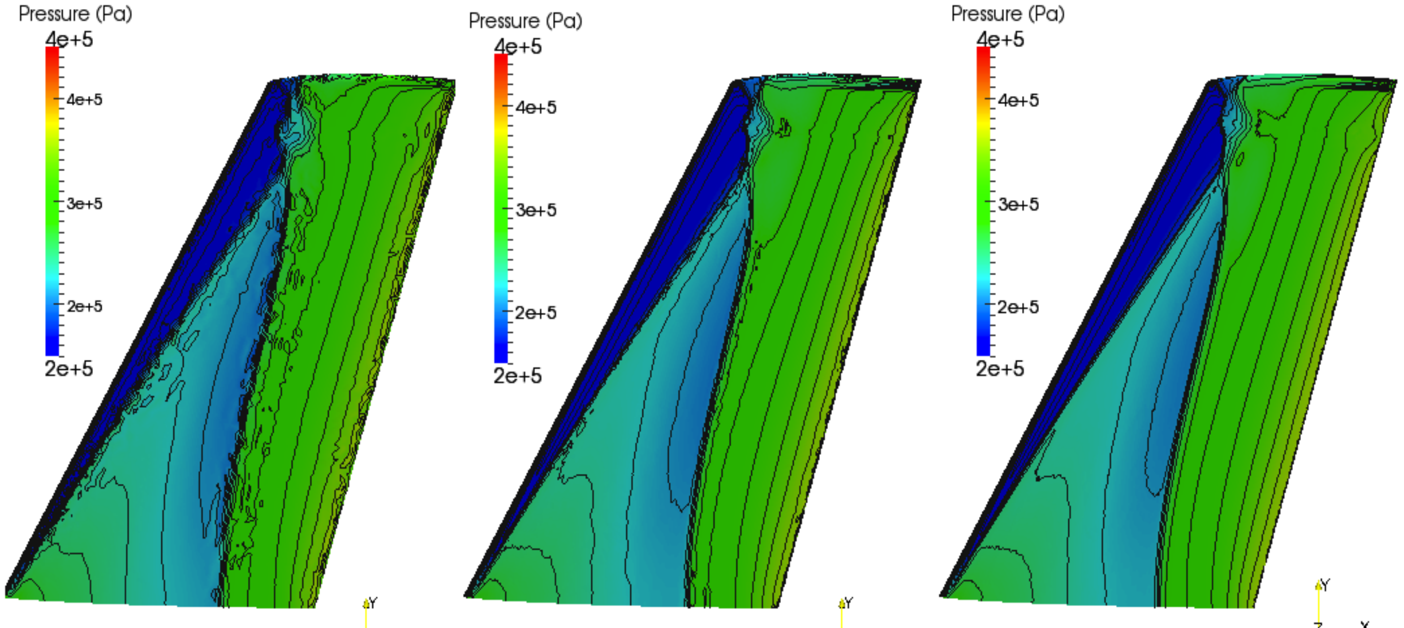
Mesh adaptation for an Accelerartor

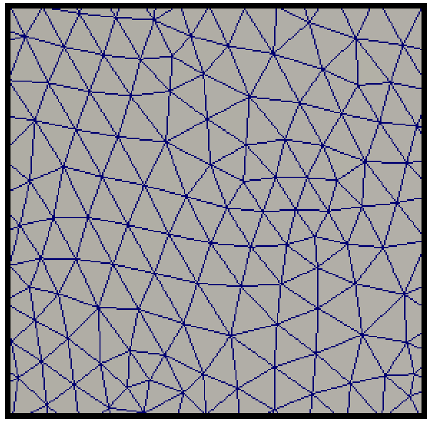
Anisotropic Mesh Adaptation



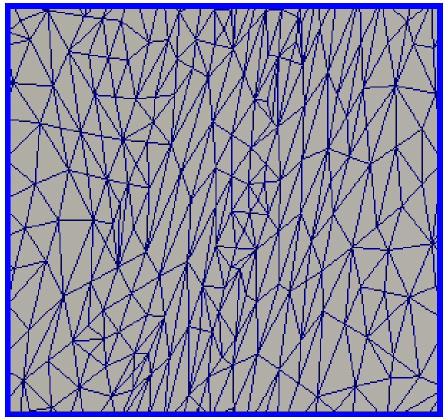




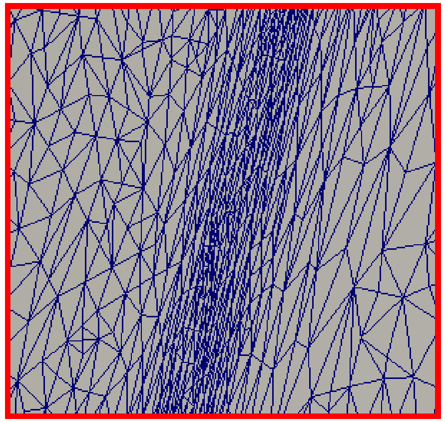




Initial: LEV0

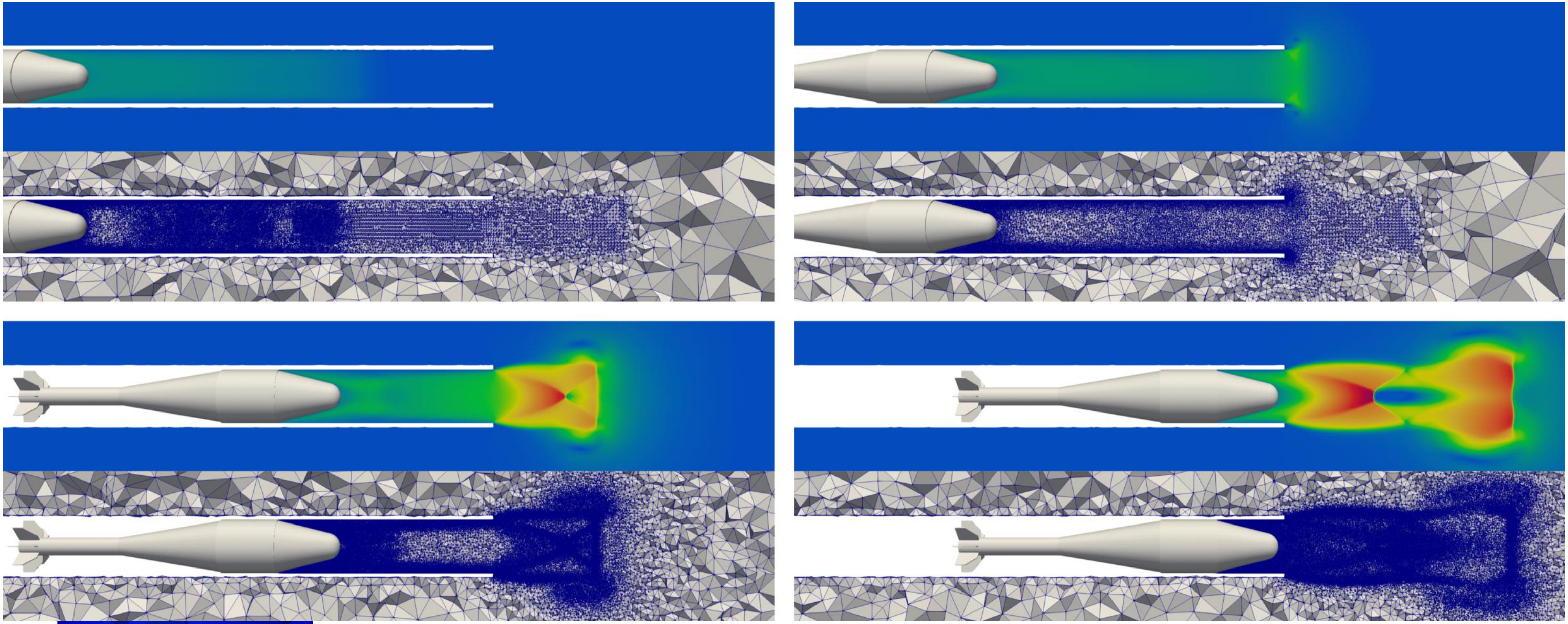


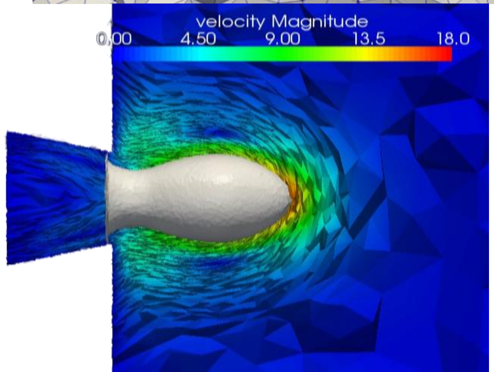
Adapted: LEV1



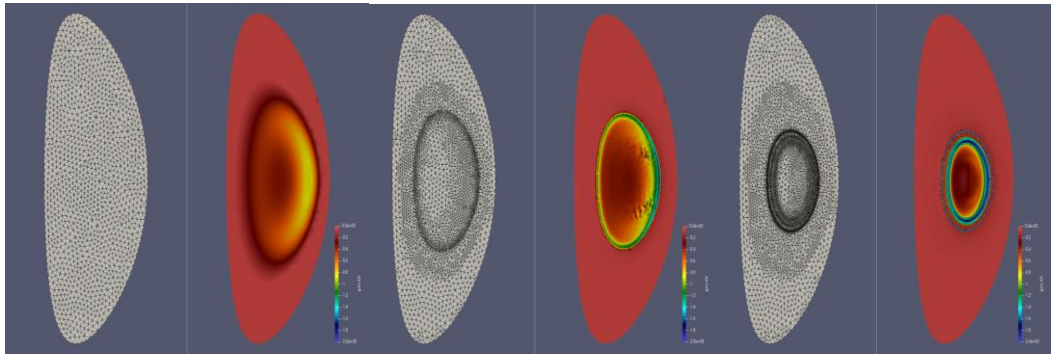
Adapted: LEV2

Time Dependent and Evolving Geometry Adaptation

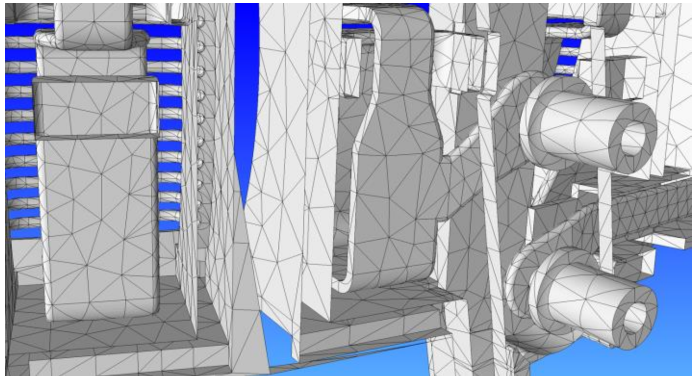


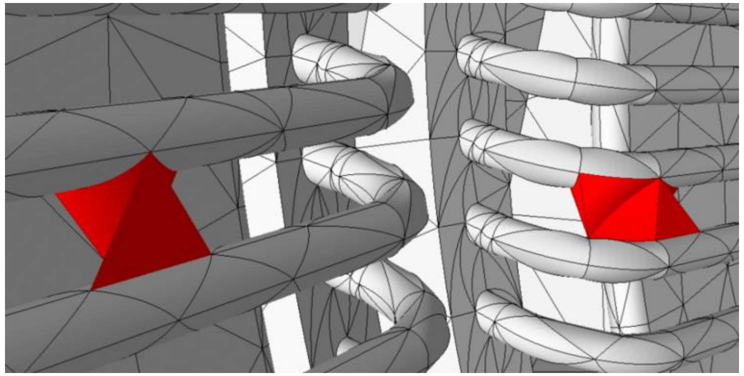


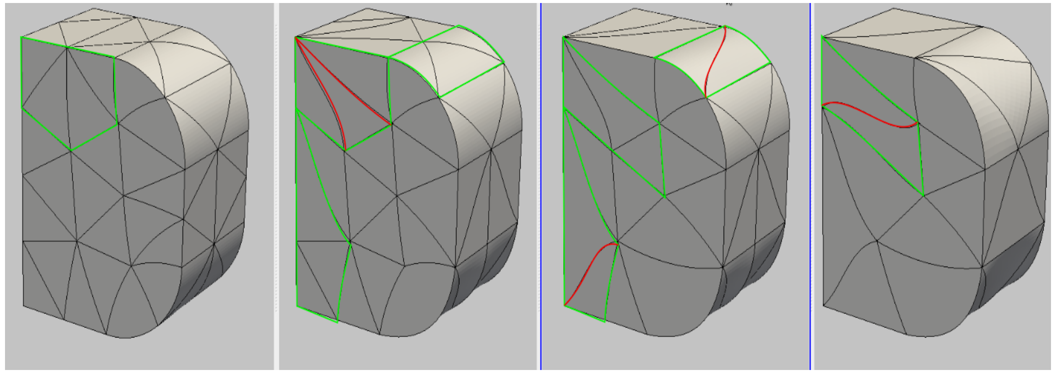
Mesh adaptation for accurate M3D-C1 simulations

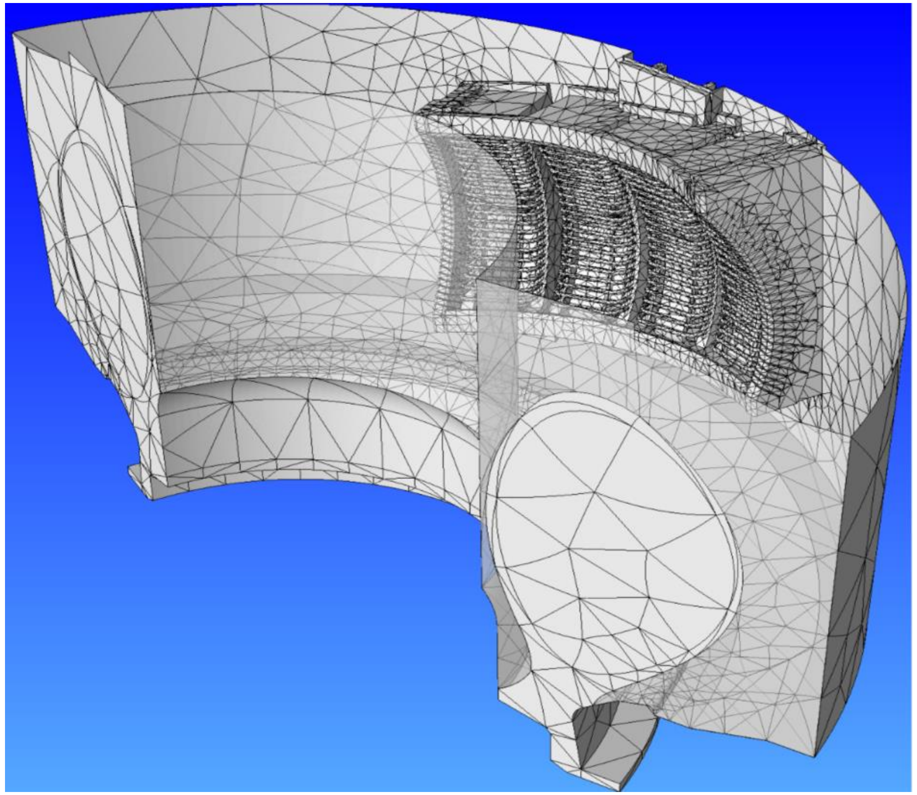


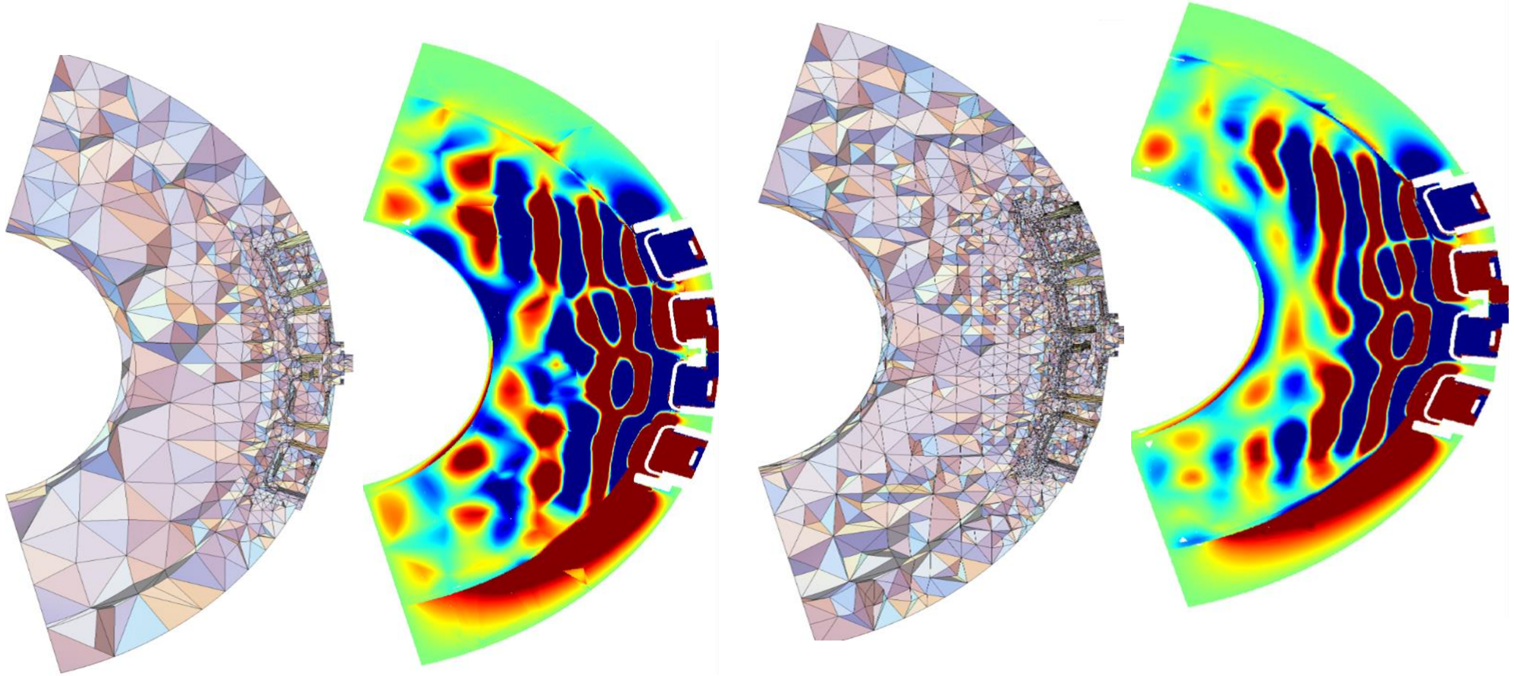
Tokamak RF Simulations with High-Order Elements in MFEM









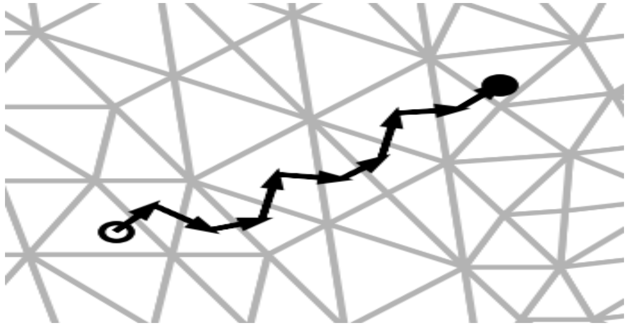


Initial Mesh Adapted Mesh

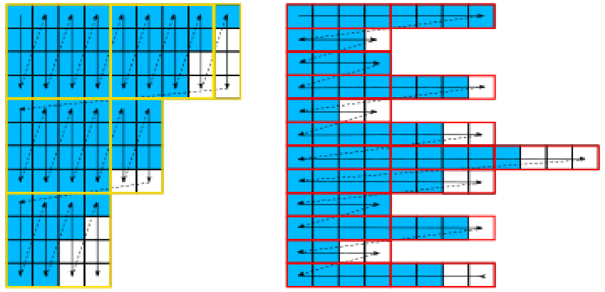
Infrastructures to Support Unstructured Mesh Operations

Approach (continued)

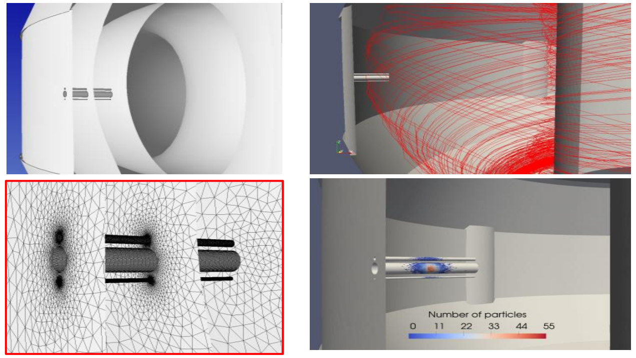
- Infrastructure for scalable execution of particle simulations over general domains (PUMI-Particle)
  - Distributed data structures for mesh (Omega_h) and particles
  - Alternative particle data structures to support different particle per cell loads and variable particle distributions
  - Full GPU based execution at scale on Frontier and Perlmutter
  - Efficient algorithm for multi-species cases
  - Dynamic load balancing of particles
  - API's for particle/mesh and particle/geometry interactions for PIC, material point method, neutronics tallies, etc.
- Applications to date:
  - Global impurity transport code (GITRm) – developed in collaboration with ORNL, used in physics studies by GA and ORNL
  - Neutronics Tallies (PUMI-Tally) – Insertion into OpenMC yielded a 19x speed-up, adjacency search methods moved into OpenMC/XDG
  - To support efficient neutral particle transport, developing a version with the atomic physics from DEGAS2 included (Under development)



Adjacency-based search

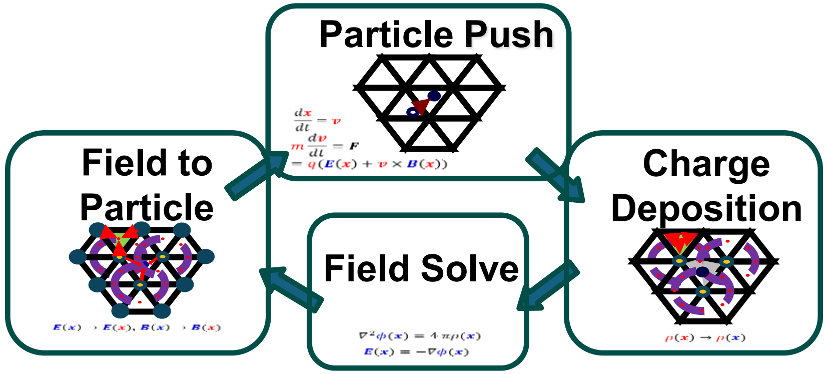


Particle data structures



DIII-D case with collector probes

Unstructured Mesh Infrastructure Particle Methods



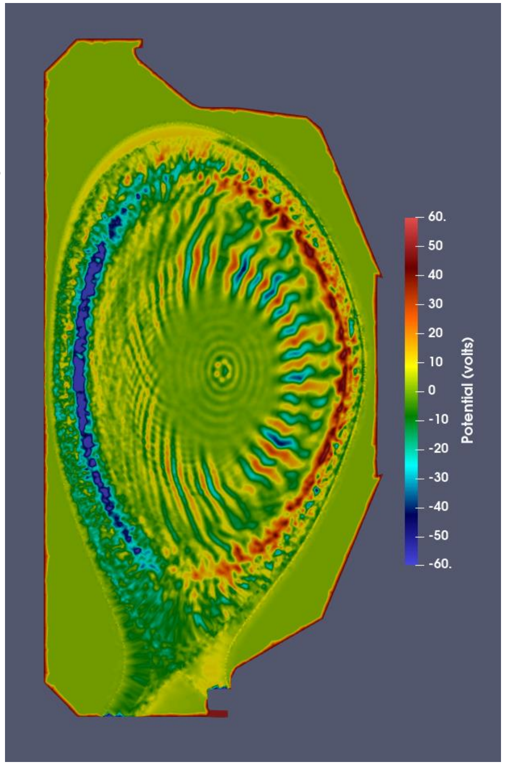
Supports applications combining particle and continuum

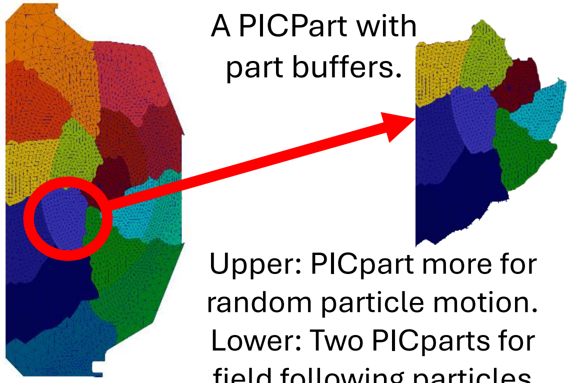
PUMI-Particle mesh-based approach

- Mesh can be distributed (Omega_h)
- Multiple particle data structures matched to needs of applications all using same API:
  - Particles related to elements (3 options)
  - Particles per PICpart
- Particle search through mesh adjacencies
- Effective coupling to mesh-based PDE solvers
- Fully executes all PIC and particle boundary operations on GPUs
- Dynamic load balancing supported

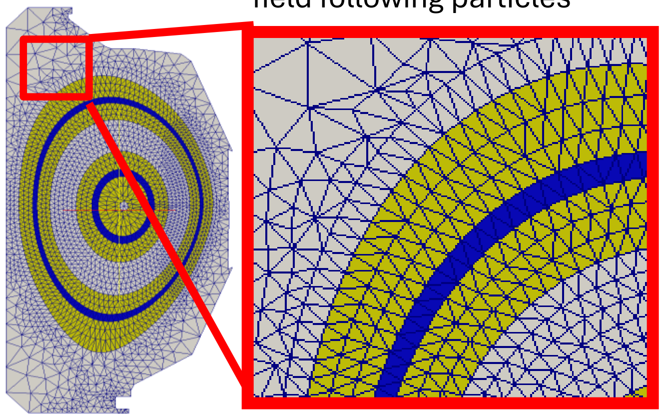
Unstructured mesh particle codes

- Impurity transport code – GITRm
- Tally calculations for neutronics demonstrated in OpenMC
- Neutral transport code with physics models used in DEGAS 2 (in early development)





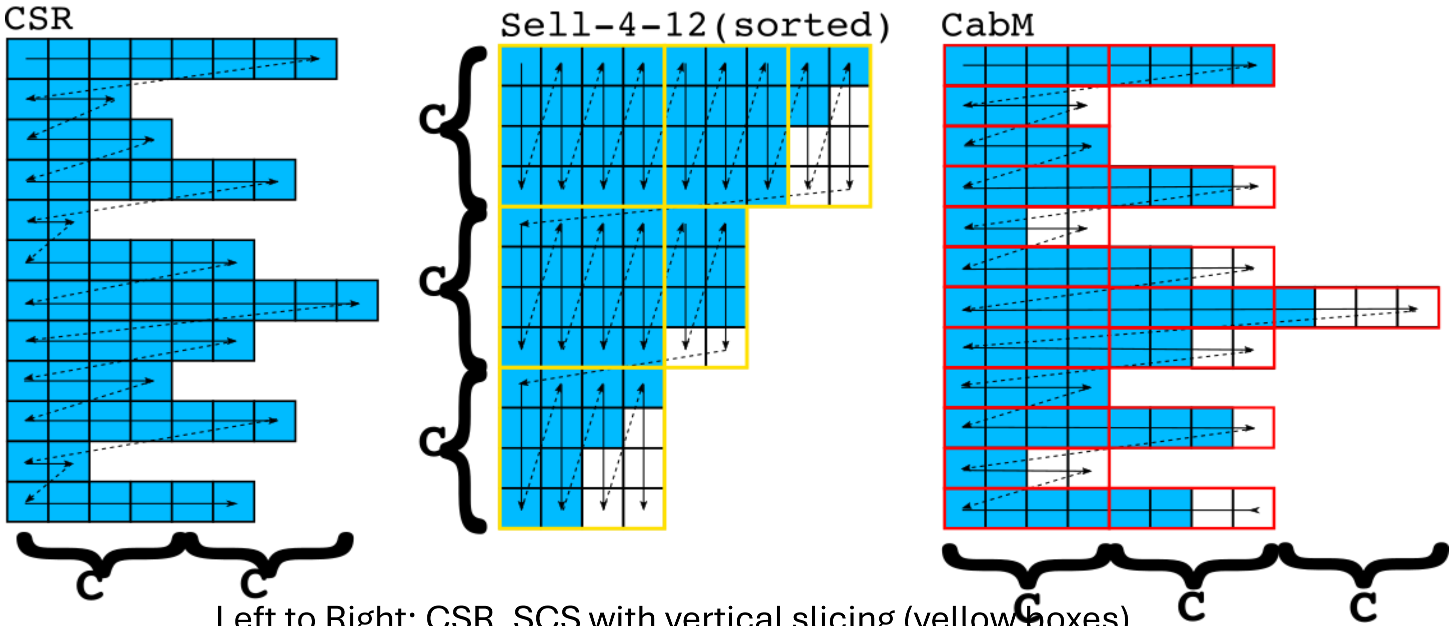
Upper: PICpart more for random particle motion. Lower: Two PICparts for field following particles



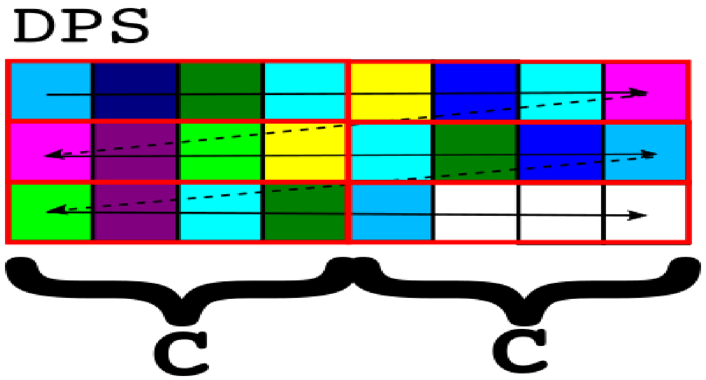
github.com/SCOREC/pumi-pic

Particle Data Structures

- Grouped by Element
  - CSR
    - Layout: 1D arrays of particle data.
    - Pros: Simple, fast push.
    - Cons: Performance.
  - Sell-C-σ (SCS)
    - Layout: rotated and sorted CSR, a row has the particles of an element.
    - Pros: Fast push, architecture aware.
    - Cons: Complexity
  - CabanaM (CabM)
    - Layout: AoSoA with CSR indexing layer over SoAs to group particles by element.
    - Pros: Fast push, simpler vs SCS, architecture aware.
    - Cons: Indexing layer overhead, breaks core Cabana functionality.
- DPS – particles **not** grouped by element:
  - Layout: AoSoA with particle-to-element 'pointer'.
  - Pros: Simpler than CabM, no cost to maintain grouping, good for low particle density per element, architecture aware.
  - Cons: Worse push performance for high particle density per element.



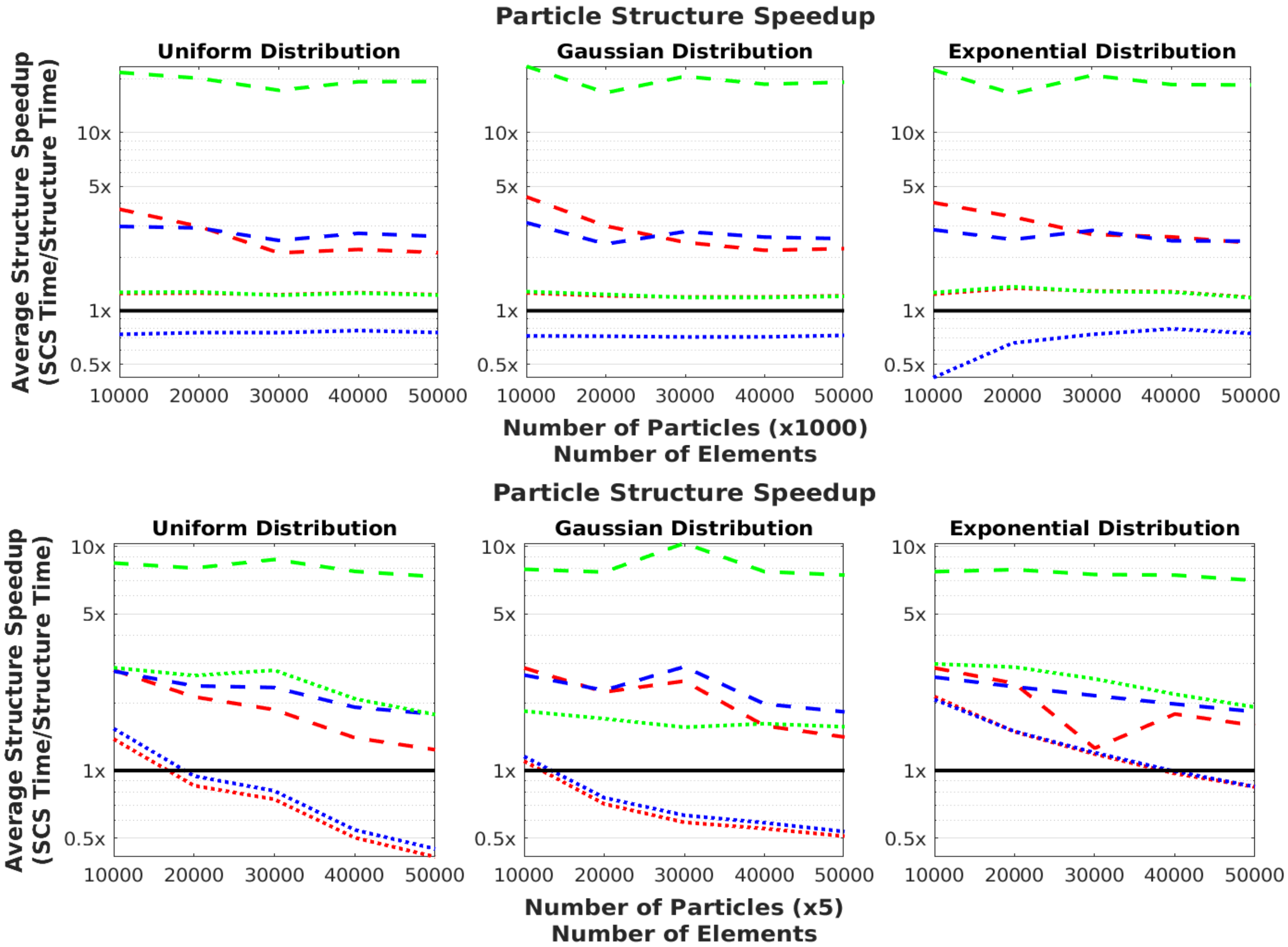
Left to Right: CSR, SCS with vertical slicing (yellow boxes), CabM with SOAs (red boxes). C is a team of threads.

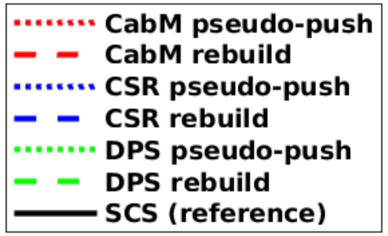


DPS structure without grouping by mesh element. Each color represents a 'pointer' to the element the particle exists within. The white cells are padding. C is the size of each SoA in the AoSoA.

Particle Data Structures: Performance Testing

- Testing procedure
  - Two NVIDIA A100 (Perlmutter)
  - Uniform, Gaussian, and exponential particle distribution
  - 10 iterations of pseudo-push and rebuild
    - *Pseudo-push:* per particle operations on its data
    - *Rebuild: assign* particles to new parent element
  - 20 doubles (160B) stored per particle
- Summary
  - At low density, DPS pseudo-push is fastest: 1.5-3 times faster than SCS.
  - At high density, pseudo-push for AoSoA based CabM and DPS is 10-20% faster than SCS.
  - DPS rebuild is about 8-10 times faster than SCS – less data movement.
  - All structures use second, 5% larger, particle structure to avoid allocations during rebuild and migrate.



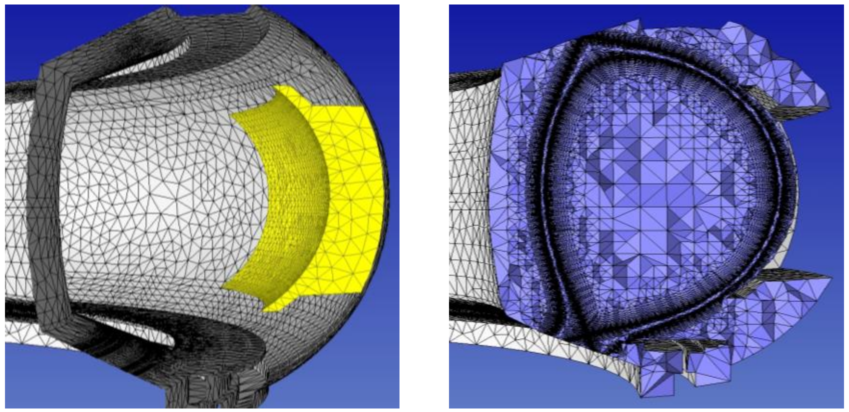


Particle structure performance relative to SCS for high (top) and low (bottom) density tests. **Higher is better.**

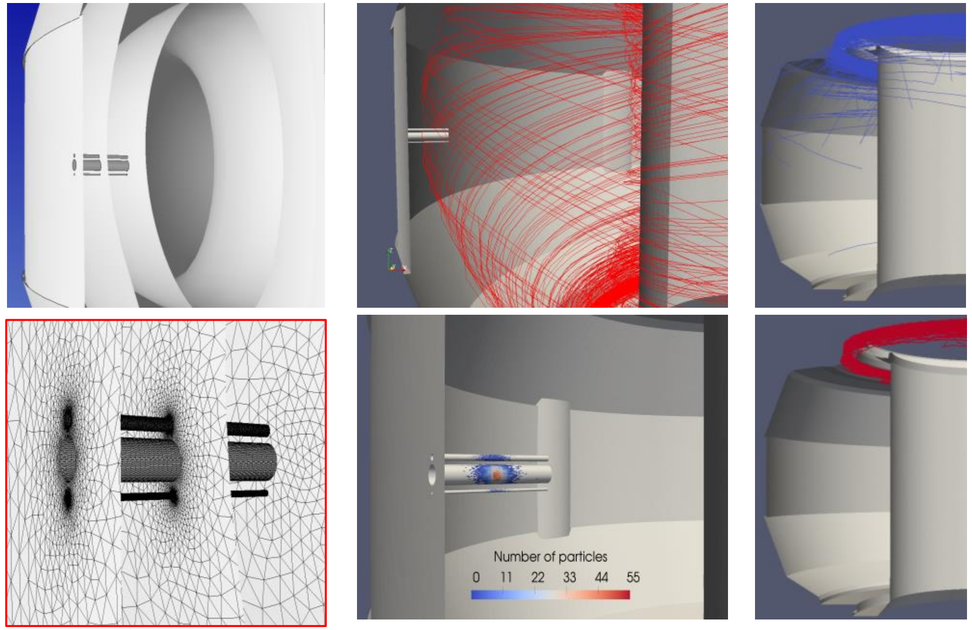
GITRm Impurity Transport Code

GITRm is a fully 3D GPU-enabled PIC code

- Supports physics operators needed for fusion plasmas
- Employs anisotropic graded meshes to effectively represent plasma fields
- Parallel particle initialization on distributed meshes for multi-GPUs
- Dynamic load balancing to account for evolution of particle distribution
- Highly effective distance to boundary to deal with near wall physics modeling
- Supports coupling with hPIC2 for sheath electric field.
- Multi-component/multi-species capabilities using highly effective parallel implementation
- 3D unstructured mesh-based post-processing capabilities are provided to support domain experts and physicists



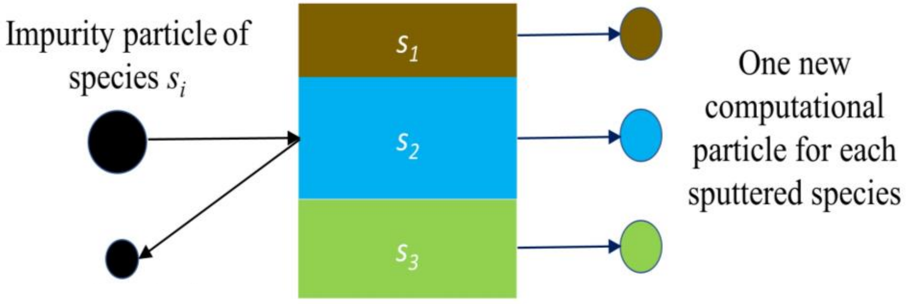
WEST with antennas and limiters

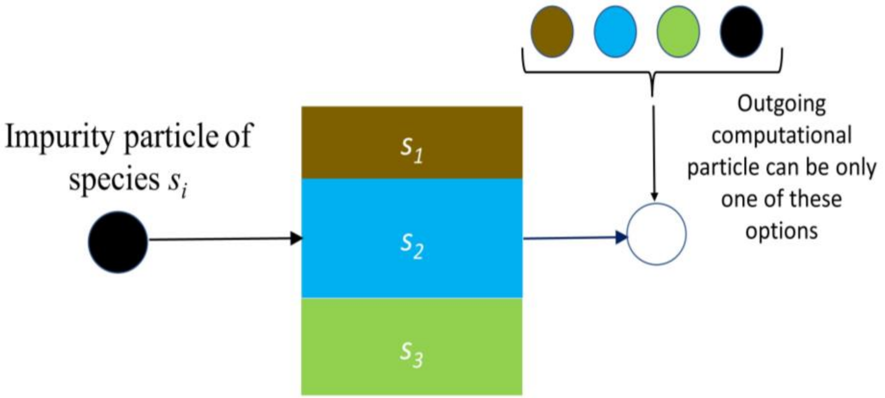


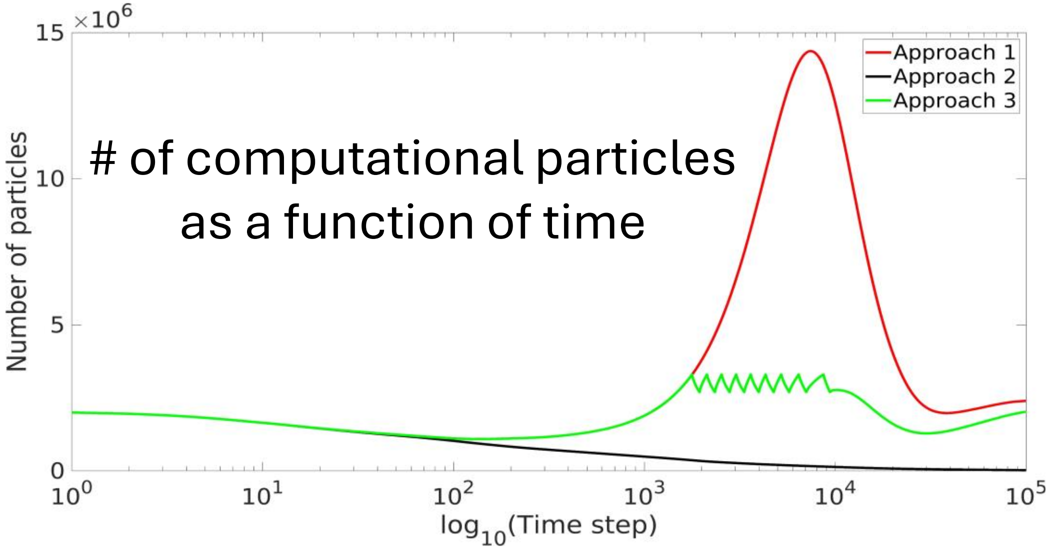
DIII-D case with collector probes (left and middle) and multiple species (right)

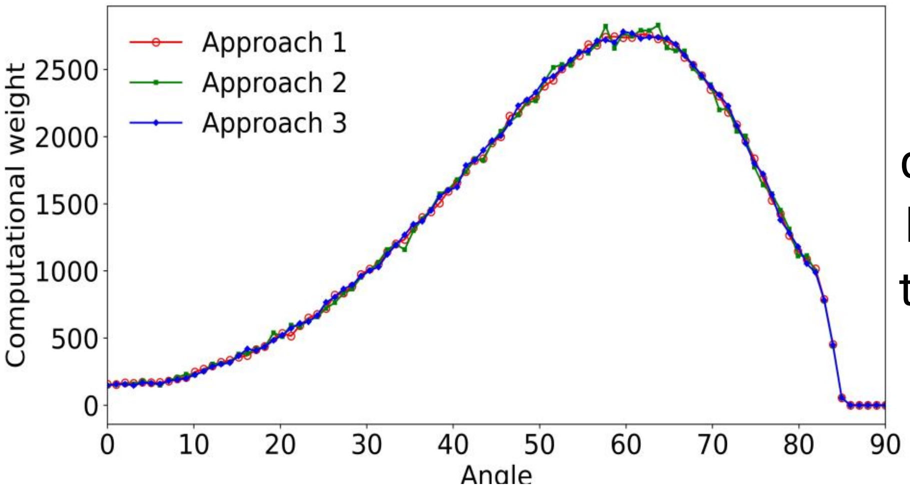
Multi-Species Support in GITRm

- Interaction of particle with a surface can result in deposition/removal of the particle and/or creation of new particles
- Two basic options
  - 0 (absorbed) or 1 particle produced
  - 0 to several particles produced
- Combined option can maintain control on the number of particles





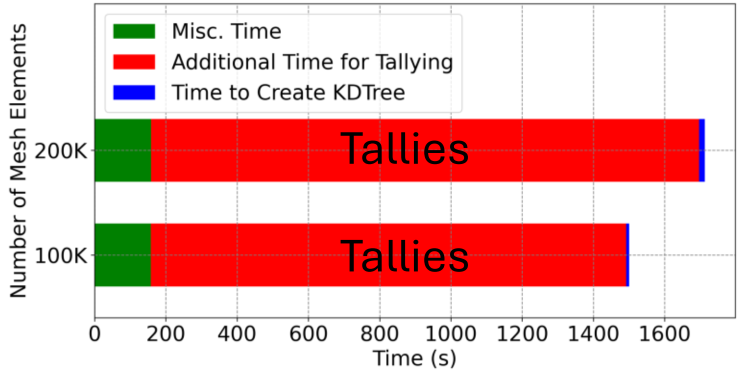


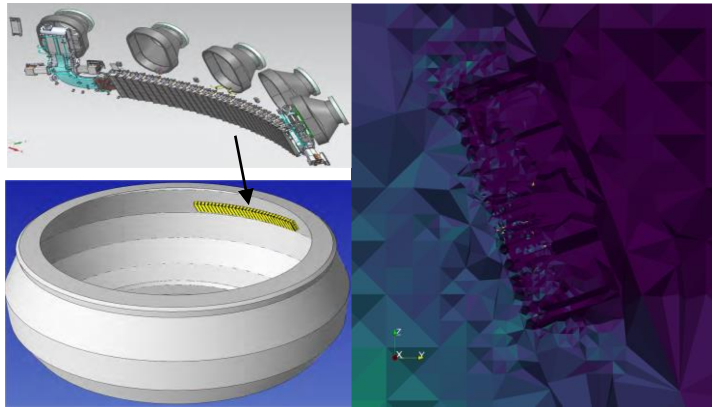


Energy angle distributions of particles hitting and sputtered off the wall (top and bottom images respectively)

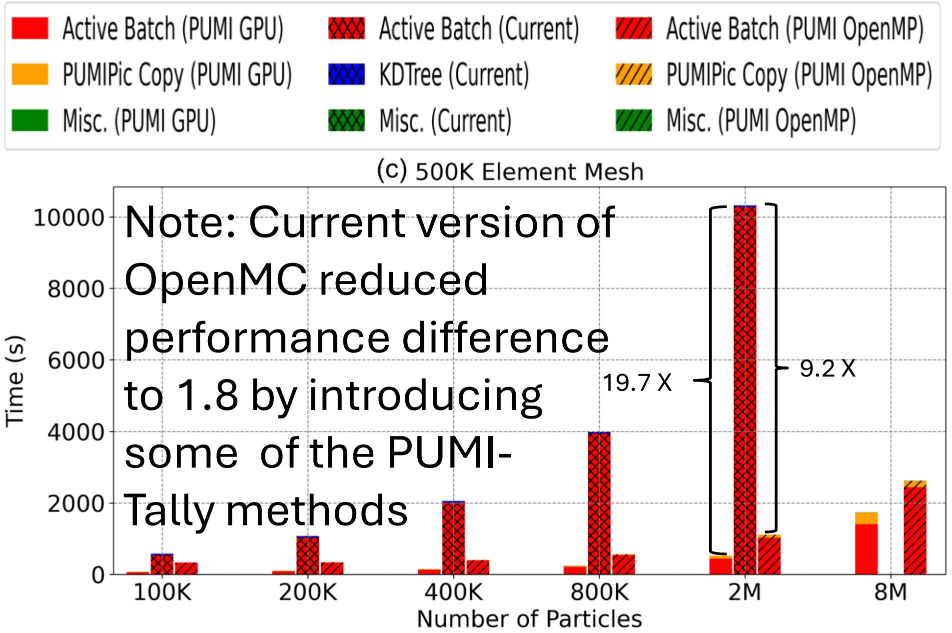
PUMI-Tally: Accelerating Unstructured Meshes for OpenMC

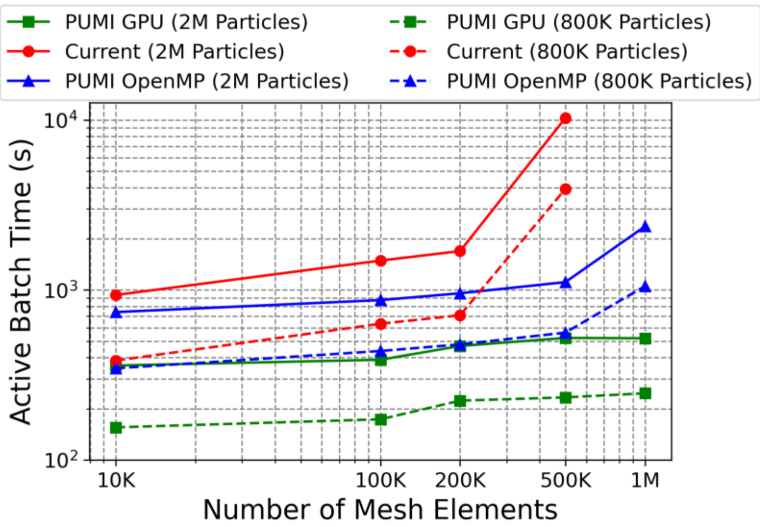
- Tallies are most significant cost for unstructured-mesh based simulations
- PUMI-Tally approach:
  - Use minimally modified CPU-Only OpenMC for physics (e.g., cross section evaluation)
  - Use PUMIPic with user defined search functors on GPU for adjacency search and classification information
  - PUMI-Tally avoids need to compile OpenMC with GPU enabled compilers.





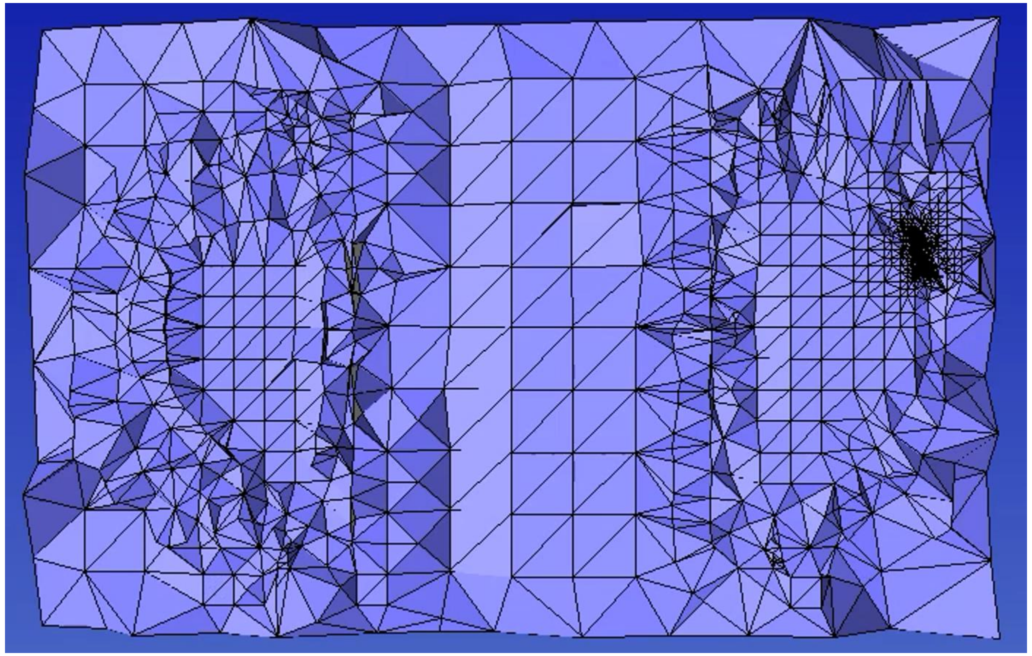
Neutronics simulation on Helicon RF Antenna in DIII-D Reactor Vessel





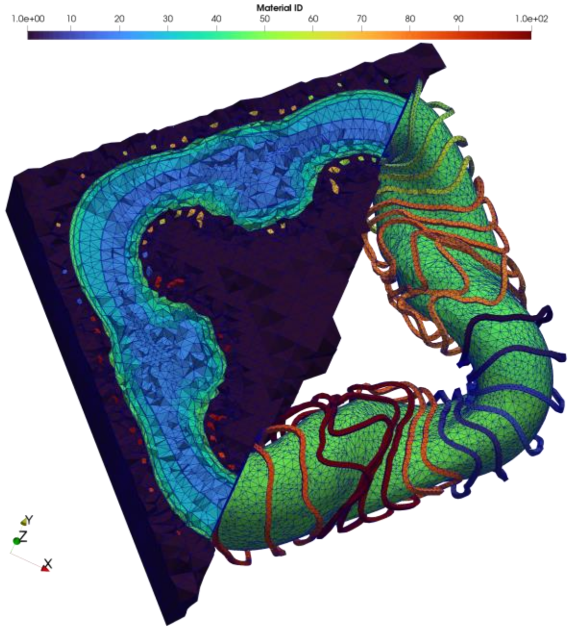
- Simulations on Perlmutter
  - 1 NVIDIA A100
  - 128 Threads of 2 AMD EPYC 7763 CPUs

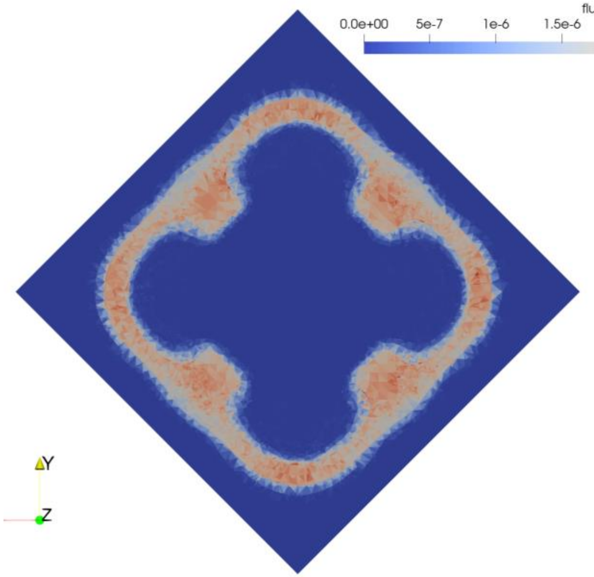
PUMI-Tally: Accelerating Unstructured Meshes for OpenMC



Preliminary Results on fusion systems

- Strongly graded 3.9M element mesh to resolve helicon antenna geometry – no performance loss when using graded meshes
- Application on ParaStell CAD model of the WISTELL-D stellarator

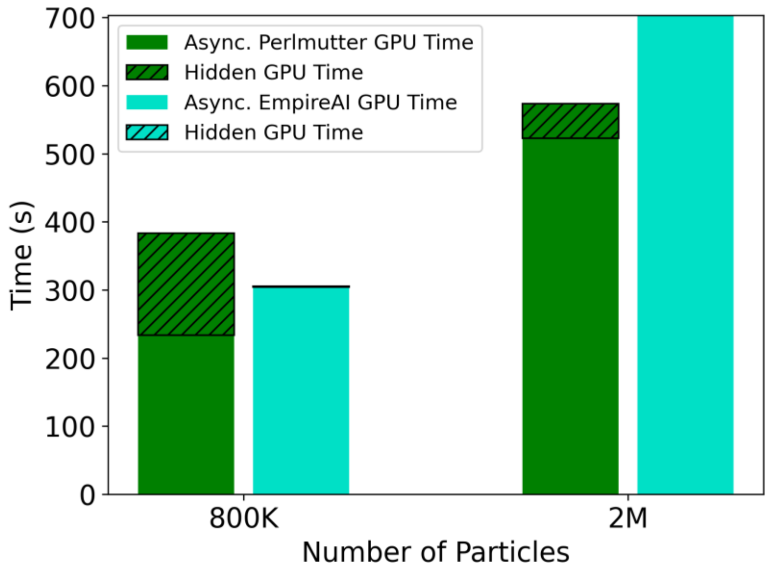




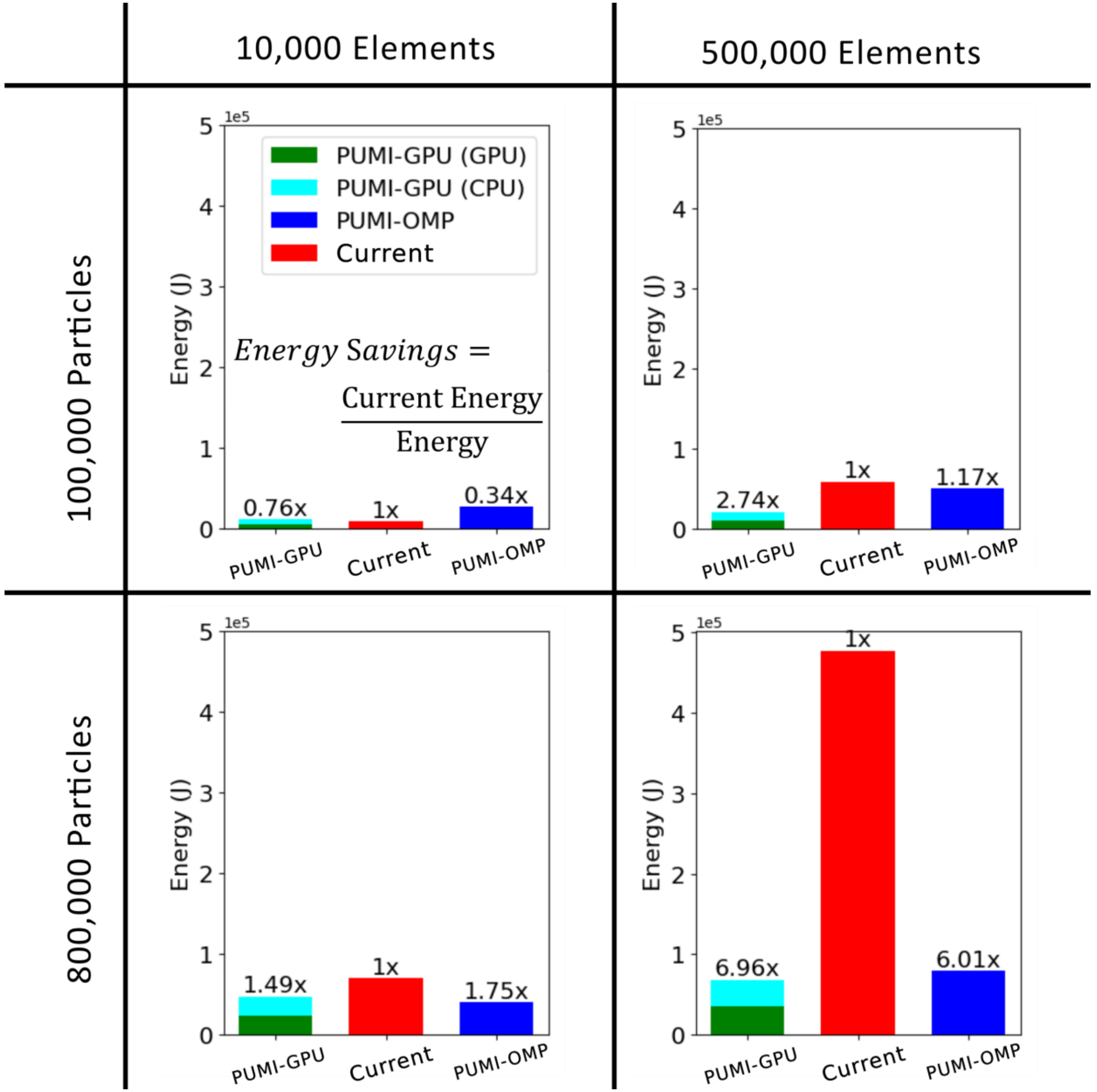
Fuad Hasan, Cameron W. Smith, Mark S. Shephard, R. Michael Churchill, George J. Wilkie, Paul K. Romano, Patrick C. Shriwise, and Jacob S. Merson. 2025. GPU Acceleration of Monte Carlo Tallies on Unstructured Meshes in OpenMC with PUMI-Tally.
https://doi.org/10.48550/arXiv.2504.19048

PUMI-Tally: Accelerating Unstructured Meshes for OpenMC

- Demonstrated ~7X less energy usage on large models.
- Slight energy penalty on small models.
- Takes advantage of asynchronous operations on GPU.



Simulations on Perlmutter
- 1NVIDIA A100
- 128 Threads of 2 AMD EPYC 7763 CPUs



Closing Remarks

Complex geometry needs to be considered for a number of fusion energy system simulations

- Unstructured mesh methods provide an opportunity to automate the creation of the meshes needed – unstructured mesh methods increasingly demonstrating ability to address fusion physics – the extreme anisotropy of key physics is still a challenge
- Using automatic mesh generation requires the ability to effectively define the analysis geometry – although there are good tools available, this area need further consideration
- Unstructured mesh infrastructure tools developed in the DOE SciDAC program can ease the complexity of applying unstructured mesh methods to more fusion energy simulation needs
- Contact Mark Shephard (shephard@rpi.edu) if any of the tools discussed in this presentation might help address your simulation needs