Parallel Unstructured Mesh Infrastructure

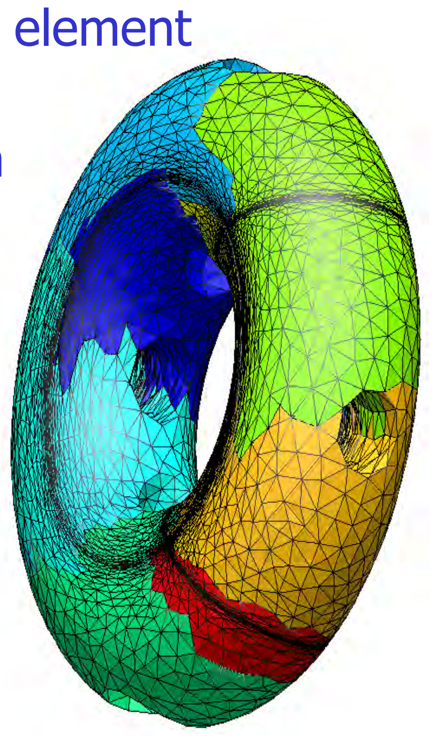
FASTMath Team Members: Dan Ibanez, E. Seegyoung Seol, Gerrett Diamond, Cameron W. Smith, Qiukai Lu, Mark S. Shephard
Scientific Computation Research Center, Rensselaer Polytechnic Institute

The PUMI toolkit supports massively parallel adaptive unstructured mesh applications. It also supports in-memory integration of adaptive mesh control through APIs. Current in-memory integration with PHASTA, Proteus, Nektar++, and FUN3D for fluid flow analysis, ACE3P for electromagnetics analysis, M3D-C1 for MHD analysis of fusion plasmas, and Albany for multiphysics analysis.

PUMI

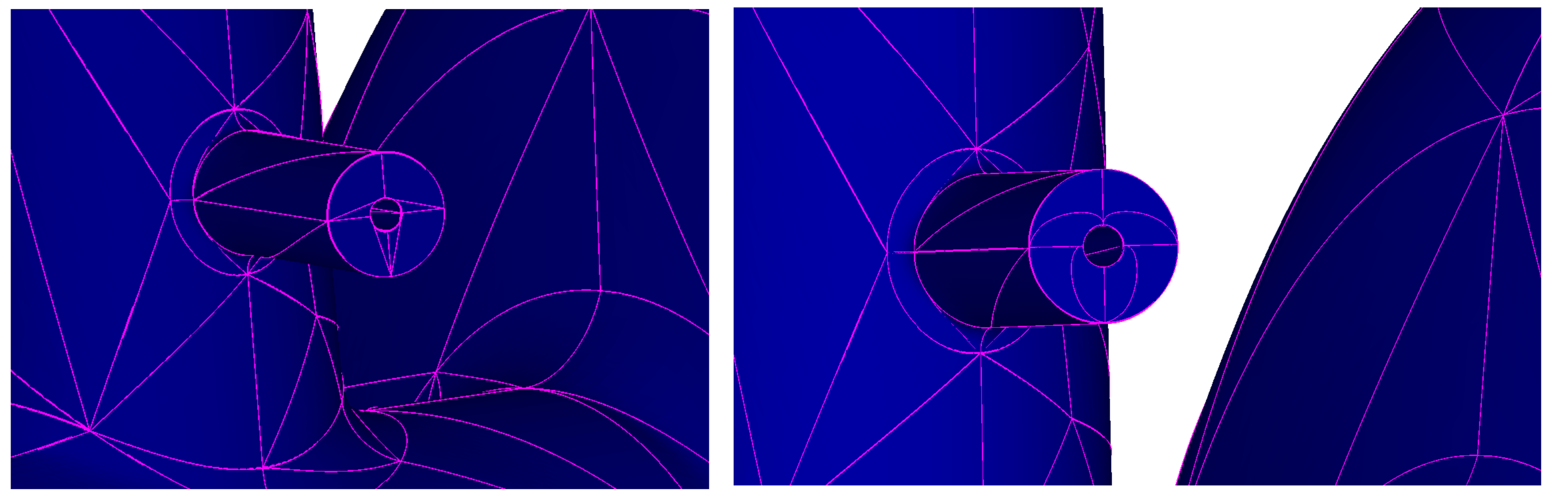
PUMI is a set of C/C++ libraries for representing and manipulating parallel unstructured meshes for large-scale finite element simulations. Key components include:

- PCU – a parallel communication library built on MPI, supporting sparse data exchange
- MDS – a compact mesh data structure, combining modifiability and array storage
- APF – a common mesh and field API enabling dependency inversion
- GMI – a common geometric model API, also serving dependency inversion of CAD kernels
- ParMA – a set of highly scalable partitioning tools



Curved Meshes for High-order Methods for SLAC

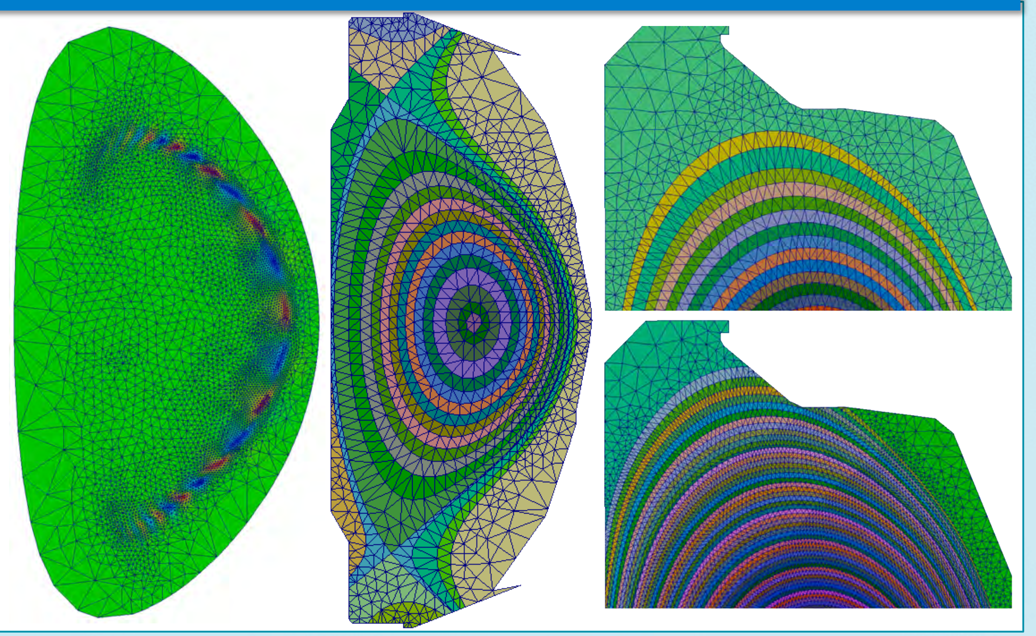
Complex curved geometry can be better approximated by high-order shape functions for surface faces. With proper corrections to maintain a positive Jacobian. Meshes by SLAC ACE3P code.



Support for PPPL Fusion Codes

PUMI supporting

- Mesh infrastructure for M3D-C1
- Mesh generation needs of XGC and M3D-C1
- Mesh adaptation needs of M3D-C1

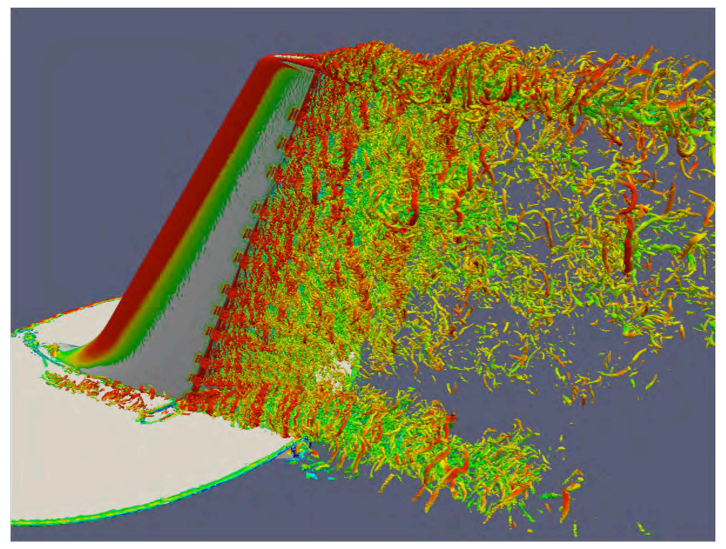


Performance Portability & Scaling

Distributed Memory Machines

PUMI scales well on machines like the BlueGene/Q, and can use threads if they communicate

For example, we generated a 92 billion element mesh for PHASTA on ALCF Mira using 256K MPI ranks and 8 threads per rank

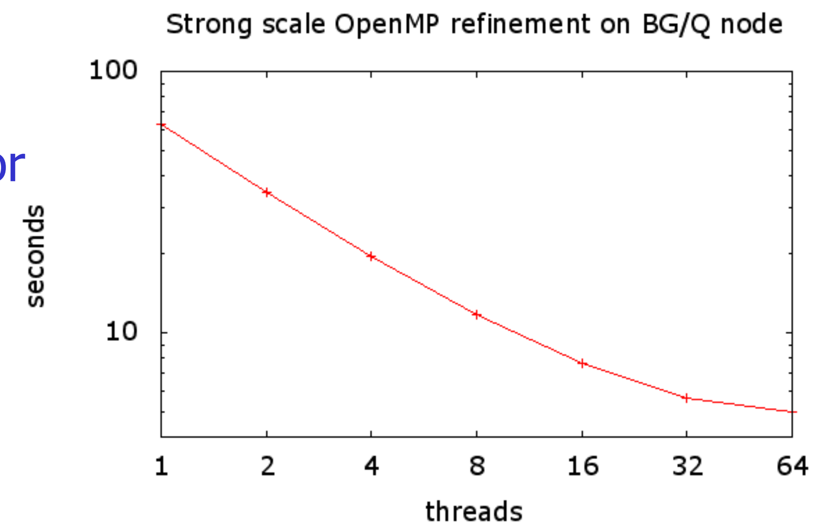


Accelerator Machines

PUMI scales well on TACC Stampede (Intel Xeon Phis), but thread communication and file IO currently not well supported by hardware

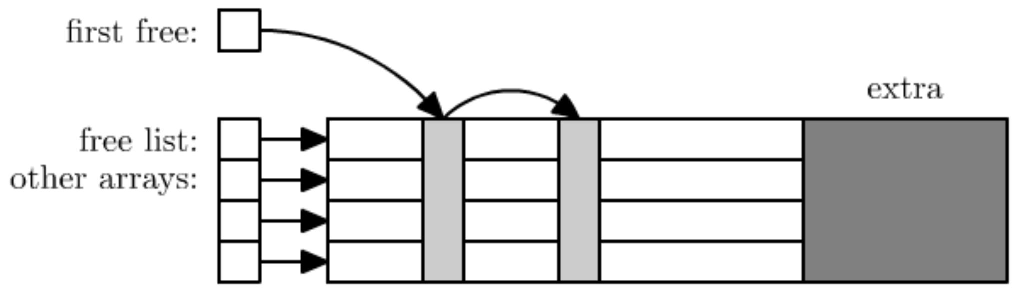
Research continues into mesh modification with OpenMP & CUDA parallelism for accelerator systems.

Performance is possible, but portability unlikely due to different models between MPI and OpenMP

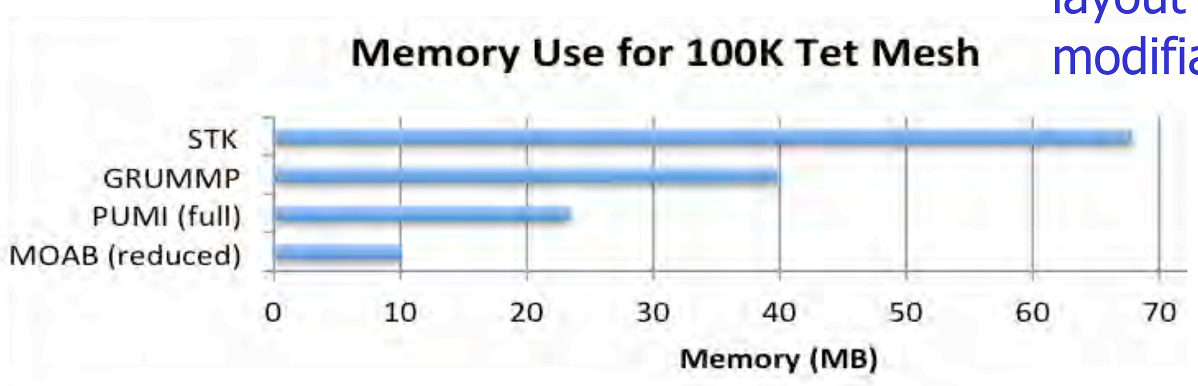


Modifiable Array Structure

PUMI has an array storage implementation (MDS) which efficiently supports entity-level modification, while being more memory efficient than other implementations capable of general adaptation.

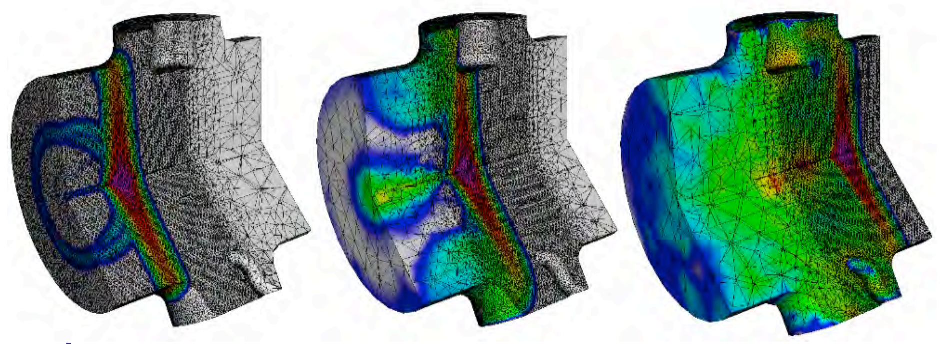


This structure can add/remove entities while using the same memory cost and layout as other non-modifiable structures

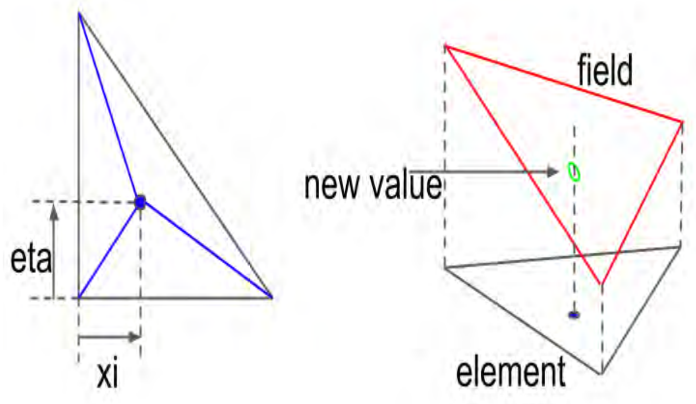


Fields

APF is the interface to PUMI. It allows dynamic adding and removing of scalar, vector, and tensor fields.

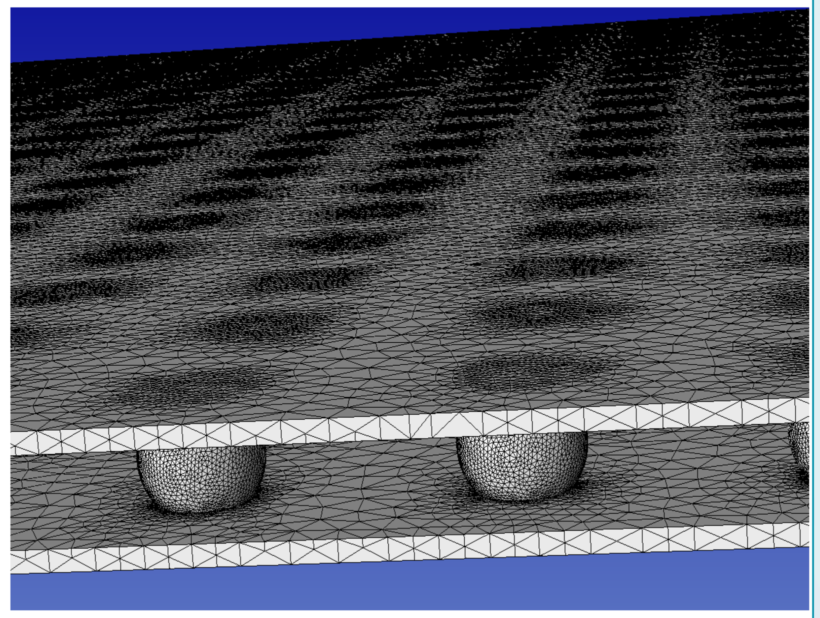


APF also has an extensible shape function system. Built-in functions include Lagrange, Bezier, and Hierarchic. General shape function queries enable correct high-order solution transfer. Supports operations on fields.



APF is designed to integrate with solvers. It associates material and boundary conditions to a CAD model, lays out field data for solvers, and simplifies linear system assembly by providing scalable numbering and node sharing tools.

Mesh and fields can be output for visualization to Paraview.

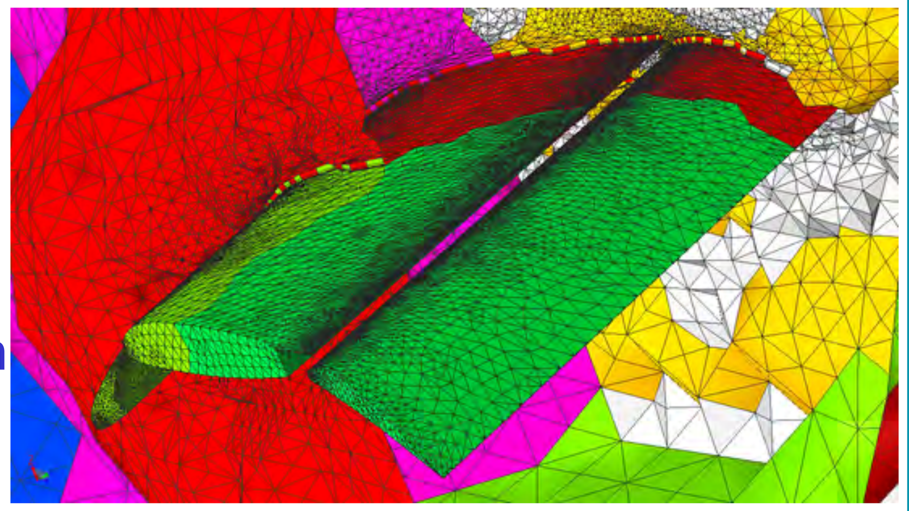


Partitioning

PUMI includes scalable mesh partitioners in the ParMA component, which have been applied to mulltibillion element meshes on up to ¾ million cores.

Recent work includes

- element weights for mixed meshes that balance memory use.
- Hybrid combination with graph and geometry based partitioners.



More Information: http://github.com/SCOREC/core or contact Dan Ibanez, RPI, ibaned@rpi.edu