Conforming & Nonconforming Adaptivity for Unstructured Meshes

Presented to
ATPESC 2017 Participants

Tzanio Kolev (LLNL) & Mark Shephard (RPI)

Q Center, St. Charles, IL (USA)
Date 08/07/2017

U.S. DEPARTMENT OF ENERGY | Office of Science

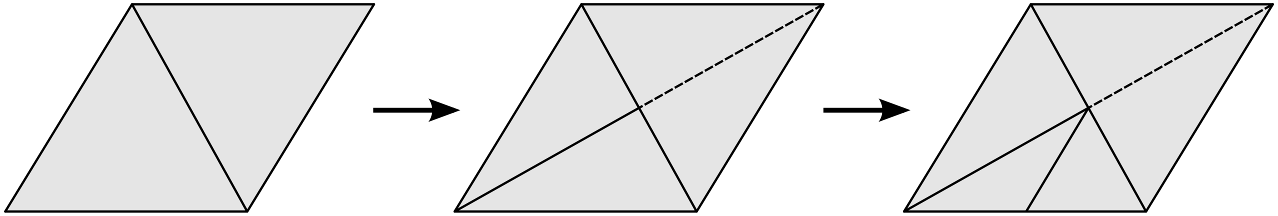
ATPESC Numerical Software Track

EXASCALE COMPUTING PROJECT

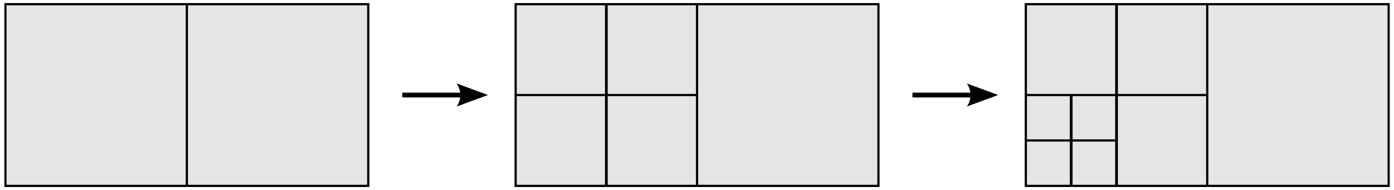
Argonne NATIONAL LABORATORY · BERKELEY LAB · Sandia National Laboratories · Rensselaer · SMU

Conforming & Nonconforming Mesh Refinement

- Conforming refinement



- Nonconforming refinement



  - Natural for quadrilaterals and hexahedra

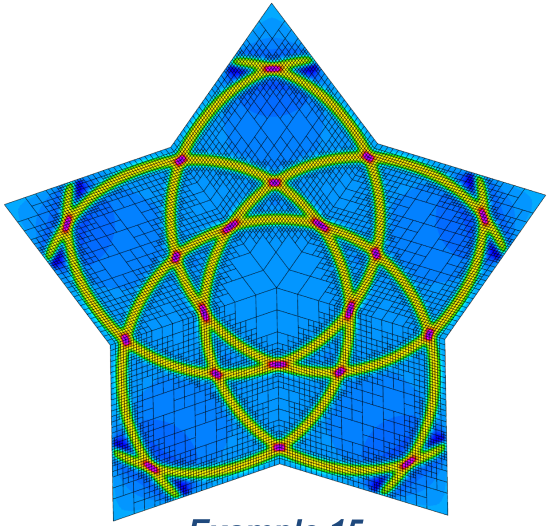
MFEM's unstructured AMR infrastructure

Adaptive mesh refinement on library level:

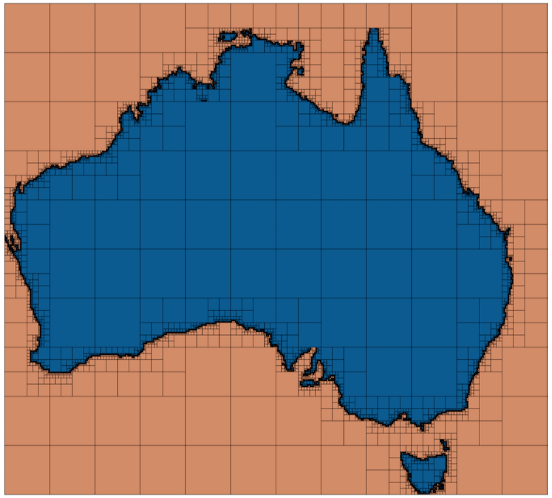
- Conforming local refinement on simplex meshes
- ***Non-conforming refinement for quad/hex meshes***
- h-refinement with fixed p

General approach:

- any high-order finite element space, H1, H(curl), H(div), …, on any high-order curved mesh
- 2D and 3D
- arbitrary order hanging nodes
- anisotropic refinement
- derifenement
- serial and parallel, including parallel load balancing
- independent of the physics (easy to incorporate in applications)



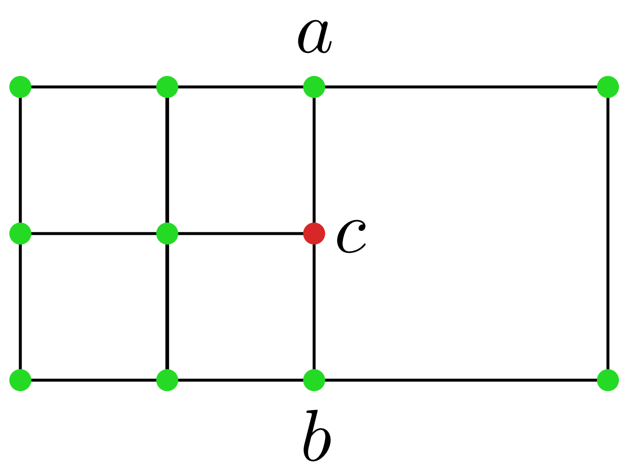
Example 15



Shaper miniapp

Nonconforming meshes – H1 finite elements

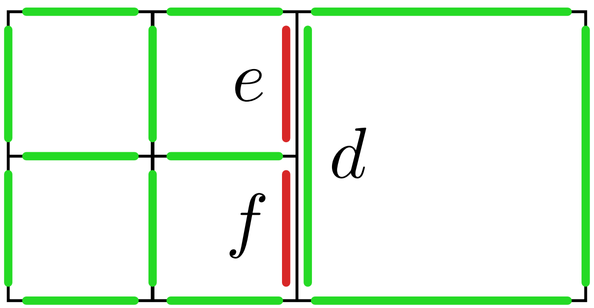
- Finite element space cut along coarse-fine interfaces (solution discontinuous)
- Define constrained FE space with some degrees of freedom (DOFs) eliminated
- Simple example: linear H^1 (continuous) elements



Constraint: $c = (a + b)/2$

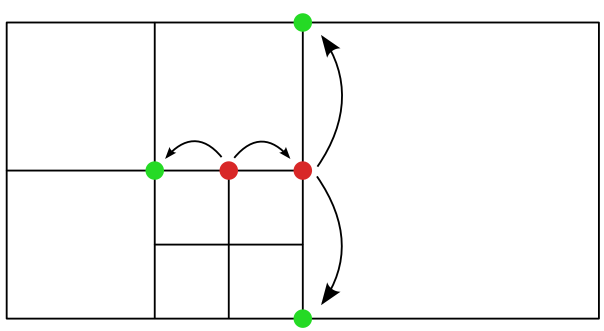
More general nonconforming constraints

H(curl) elements



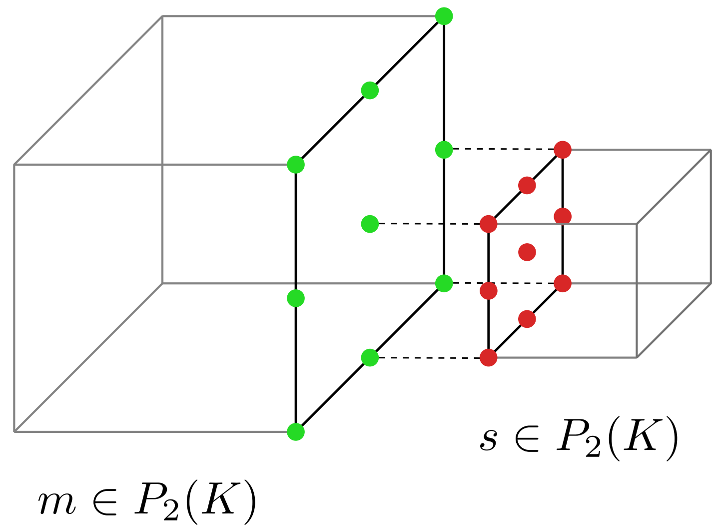
Constraint: $e = f = d/2$

Indirect constraints



More complicated in 3D…

High-order elements



Constraint: local interpolation matrix

$$s = Q \cdot m, \quad Q \in \mathbb{R}^{9\times 9}$$

Nonconforming variational restriction

- General constraint:

$$y = Px, \quad P = \begin{bmatrix} I \\ W \end{bmatrix}.$$

x – conforming space DOFs,
y – nonconforming space DOFs (unconstrained + slave),

$$\dim(x) \leq \dim(y)$$

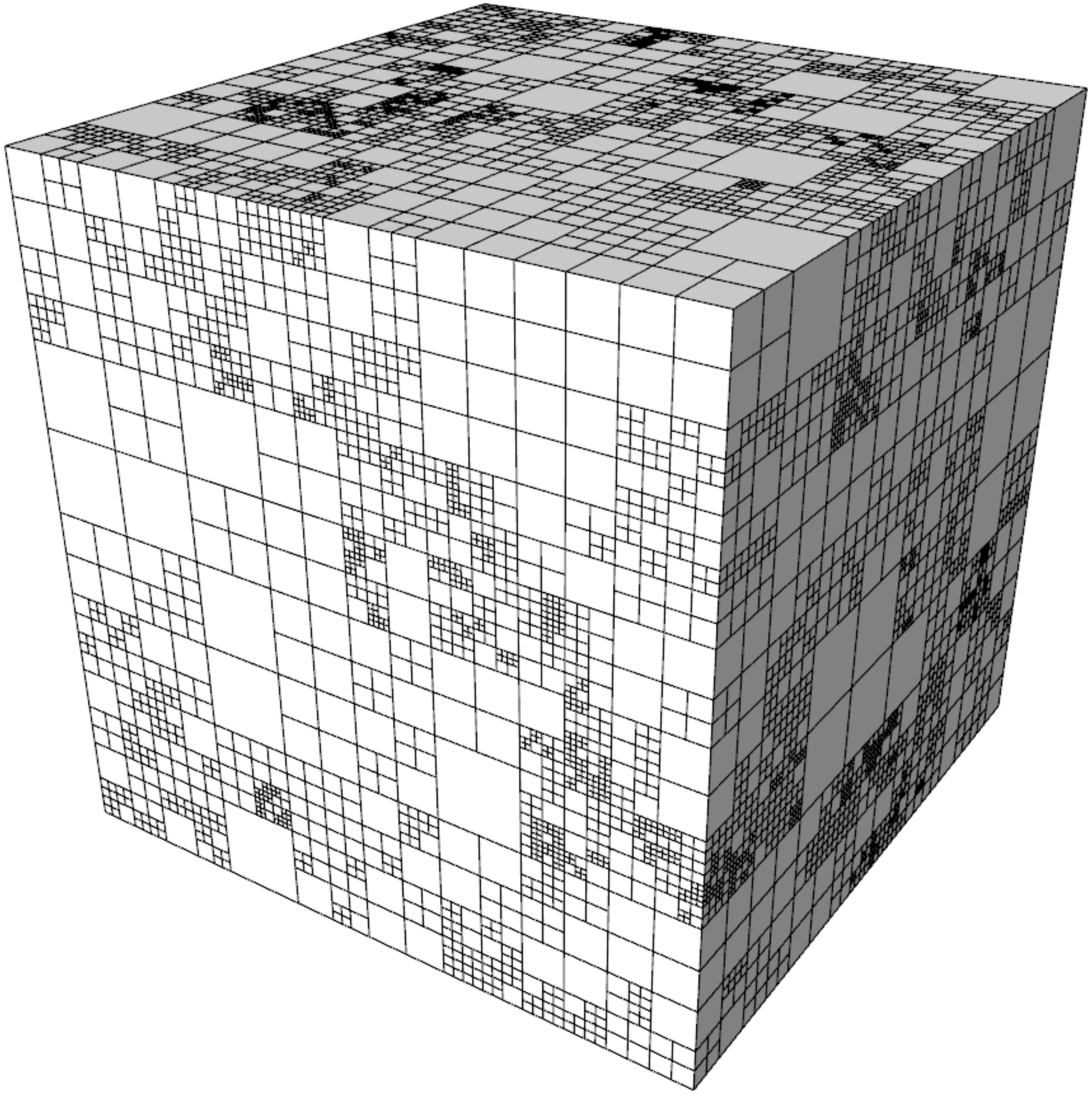
W – interpolation for slave DOFs

- Constrained problem:

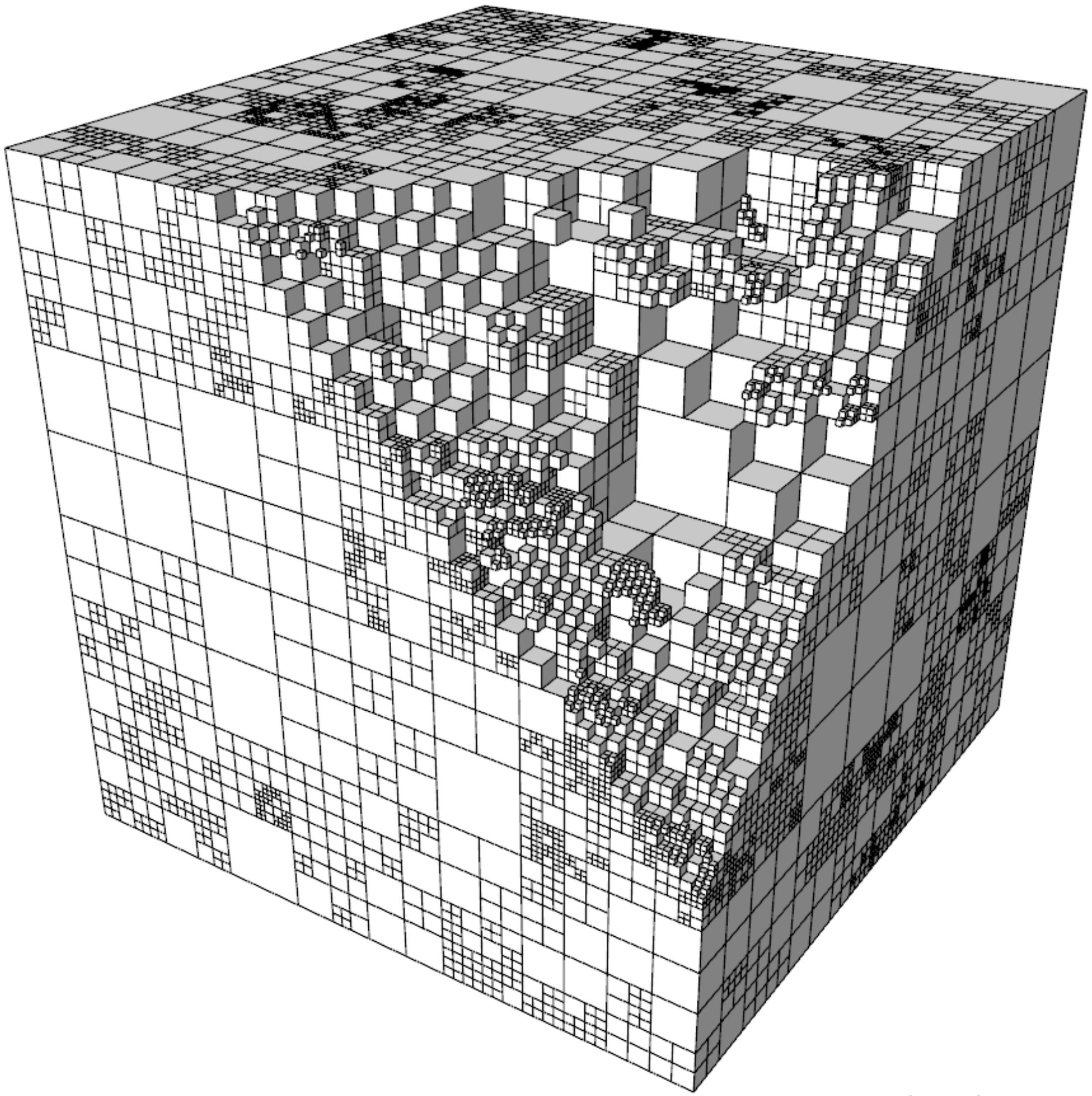
$$P^T A P x = P^T b,$$

$$y = Px.$$

Nonconforming variational restriction

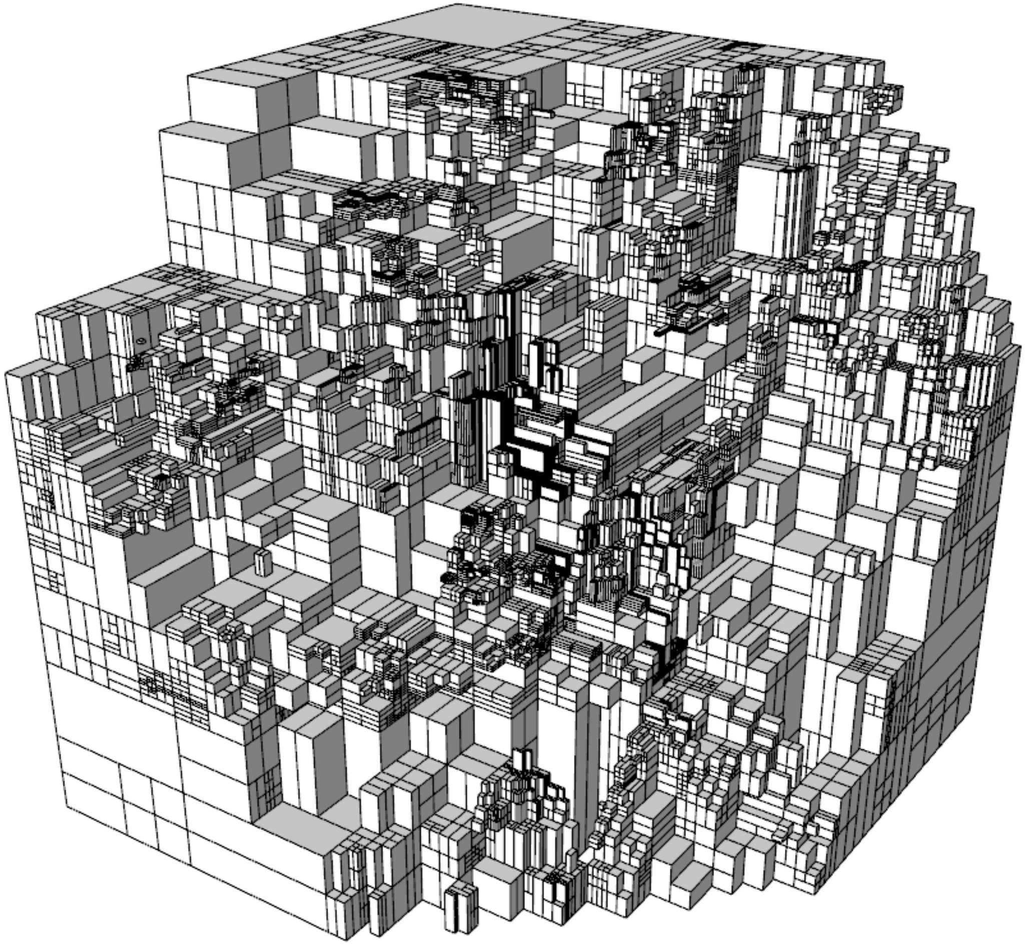


Nonconforming variational restriction



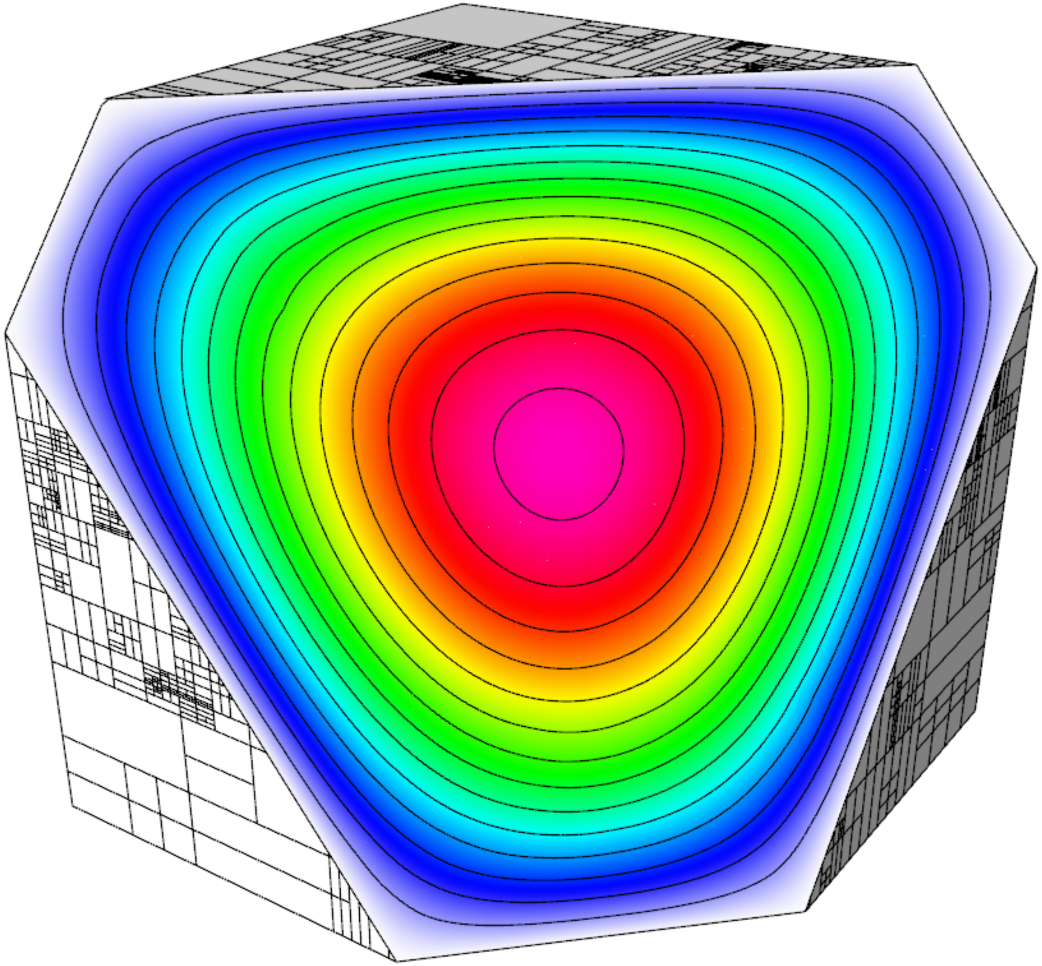
Regular assembly of A on the elements of the (cut) mesh

Nonconforming variational restriction



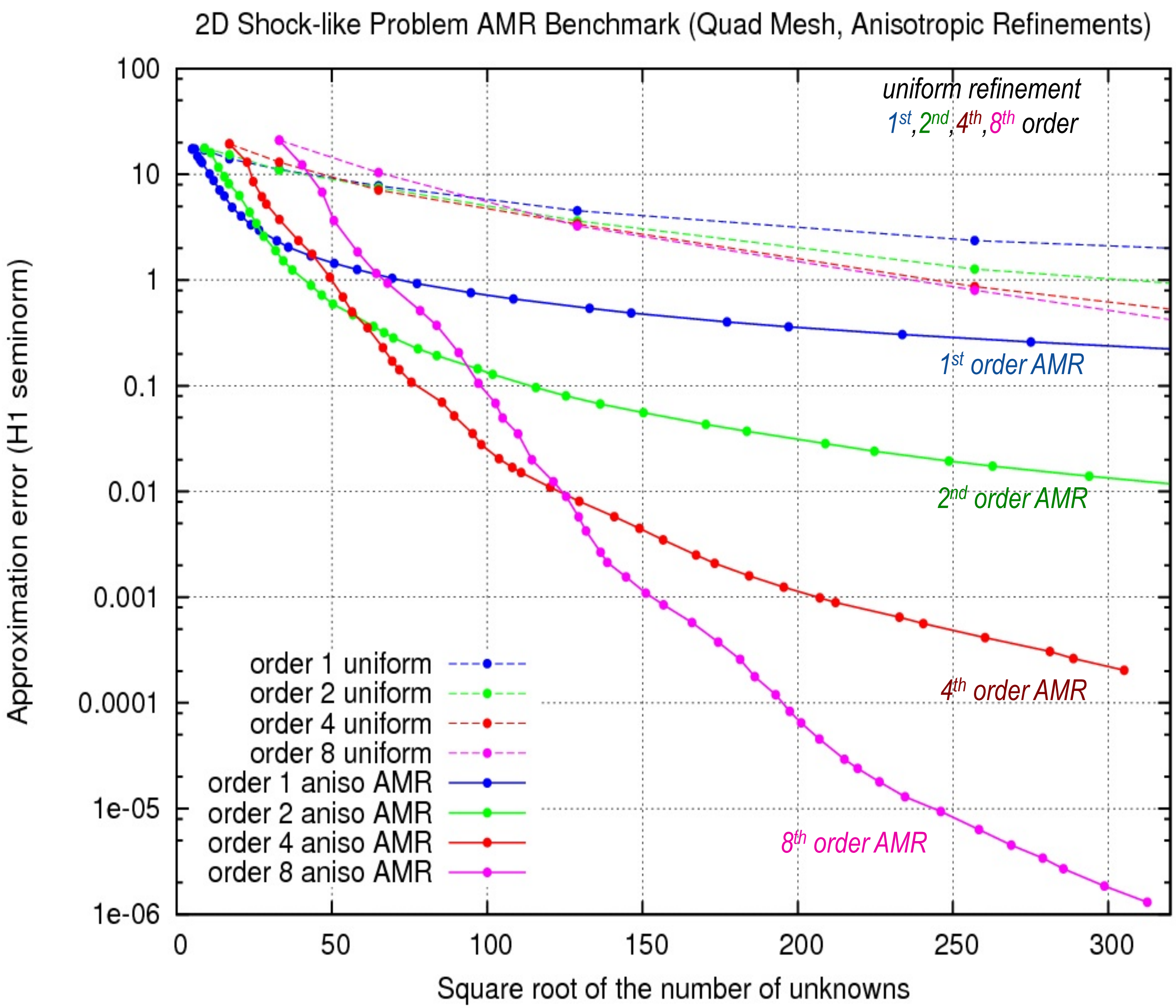
Regular assembly of A on the elements of the (cut) mesh

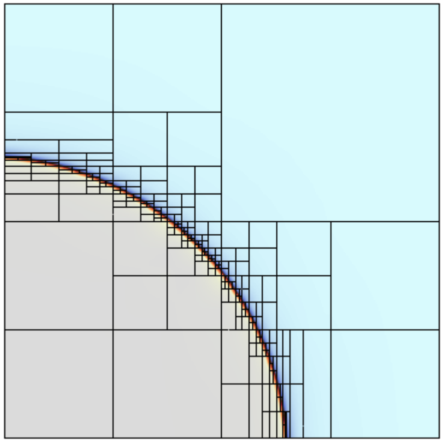
Nonconforming variational restriction

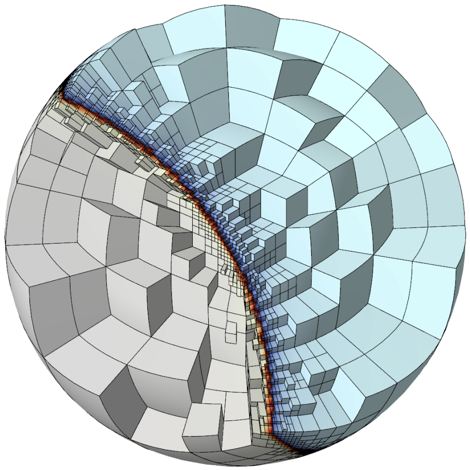


Conforming solution $y = P\,x$

AMR = smaller error for same number of unknowns



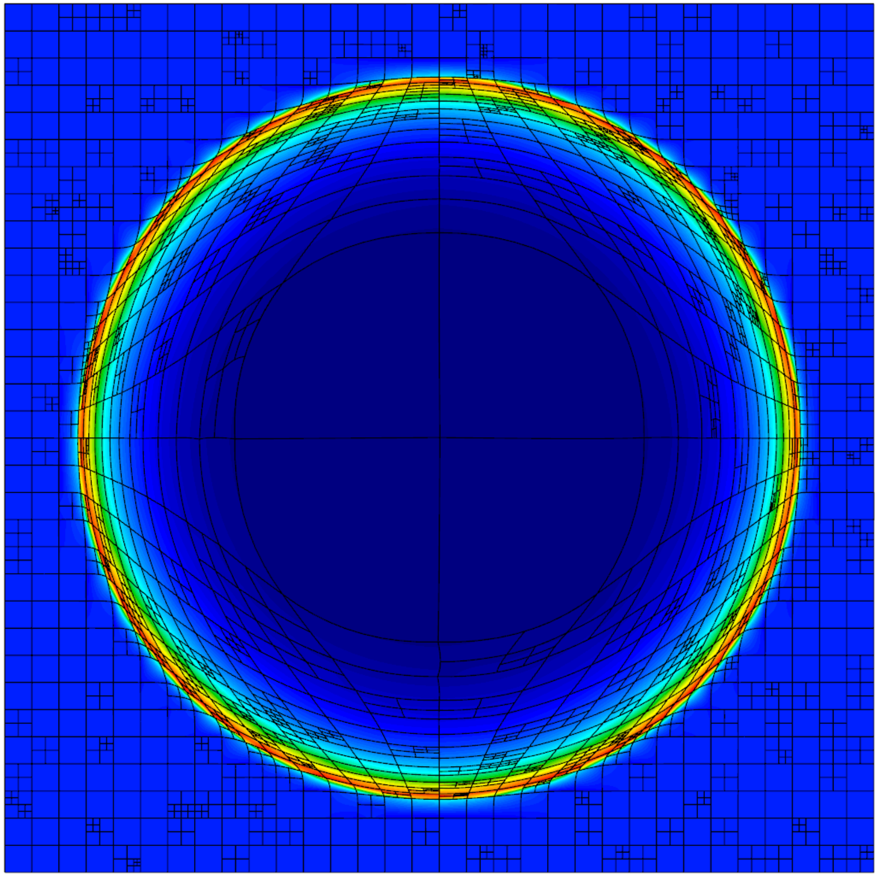




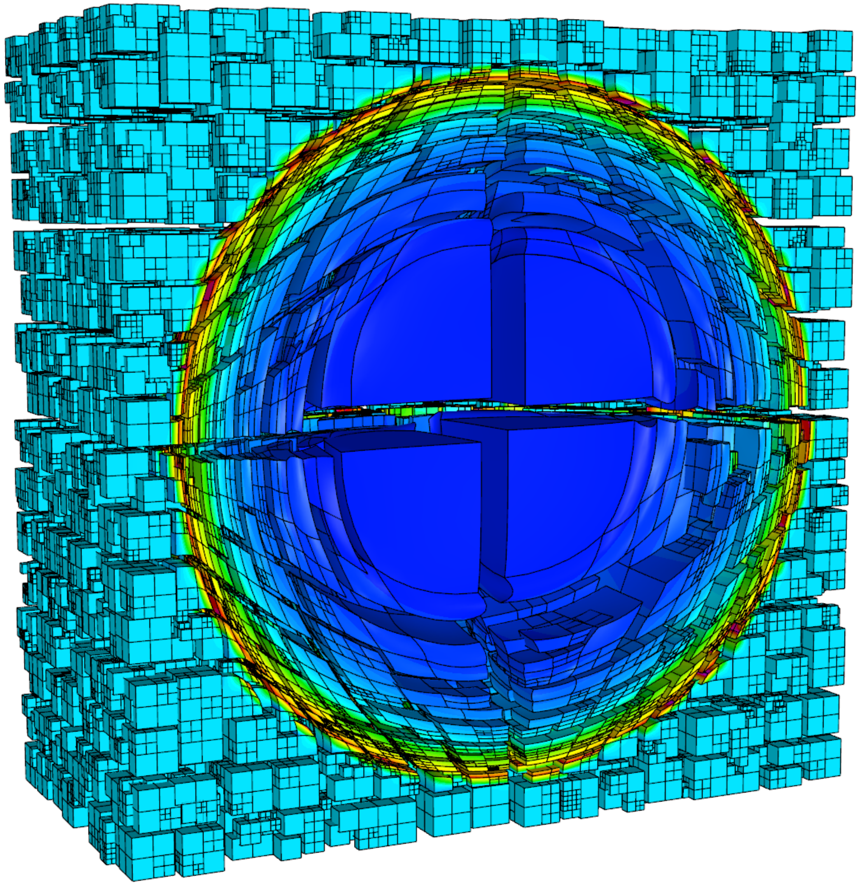
Anisotropic adaptation to shock-like fields in 2D & 3D

Static parallel refinement, Lagrangian Sedov problem

8 cores, random non-conforming ref.

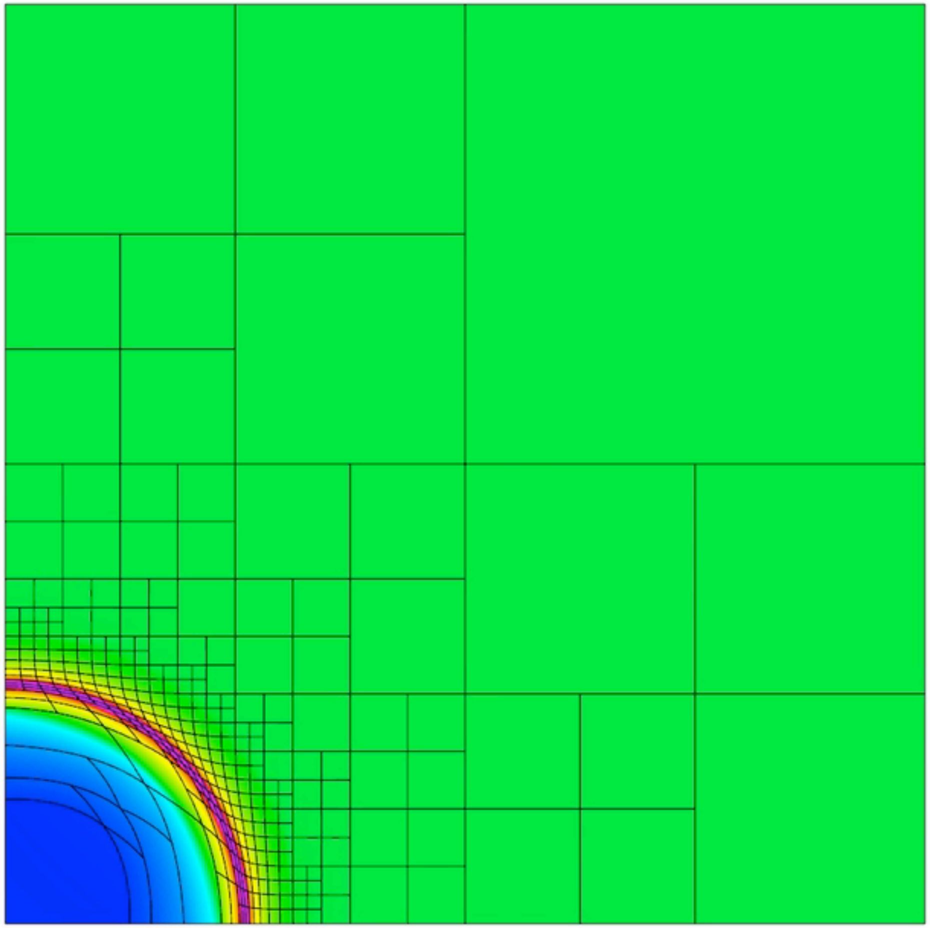


4096 cores, random non-conforming ref.

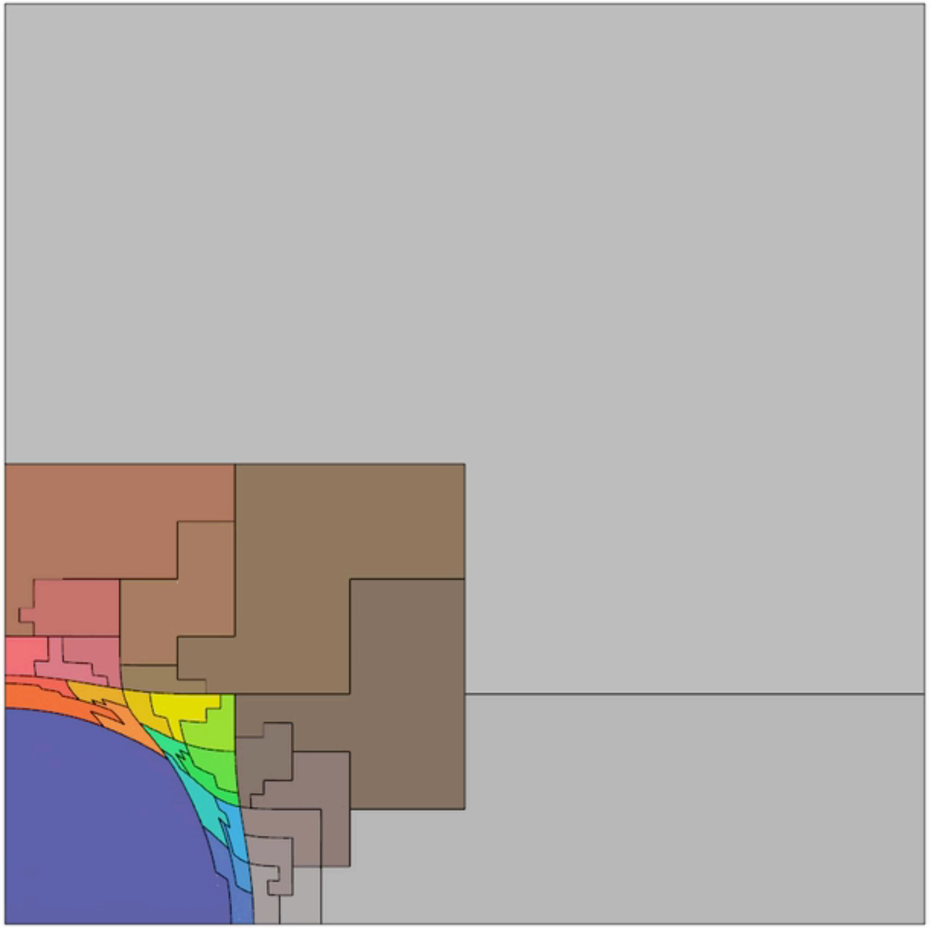


Shock propagates through non-conforming zones without imprinting

Initial Lagrangian dynamic AMR results



Adaptive, viscosity-based refinement and derefinement. 2nd order Lagrangian Sedov



Parallel load balancing based on space-filling curve partitioning, 16 cores

Integration of Geometry-Based Adaptive Simulation into MFEM

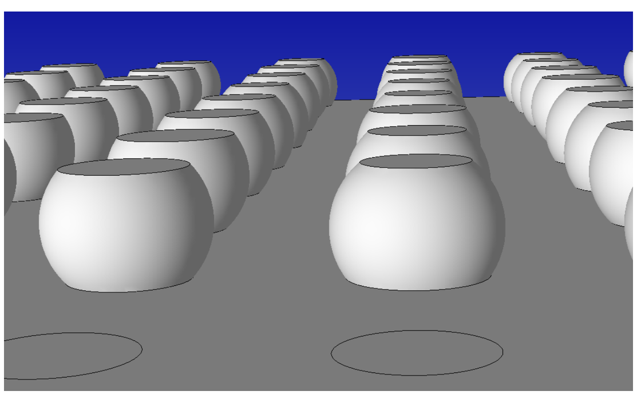
Goal: Fully automated parallel adaptive simulations from general problem definition consisting of:

- A complete geometric domain definition
- The physical parameters defined in terms of that model
- List of desired output fields and level of accuracy desired

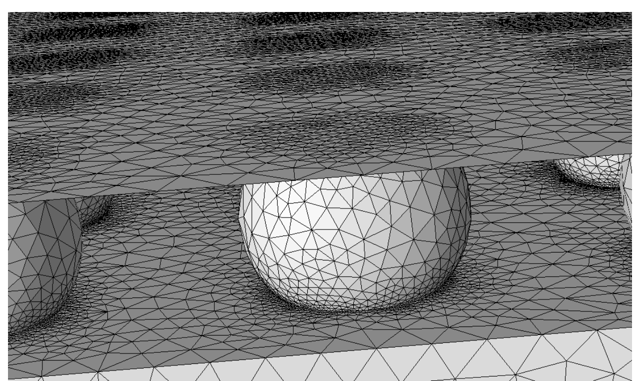
Proper interaction with geometry must be addressed

This is of even greater importance for high order methods

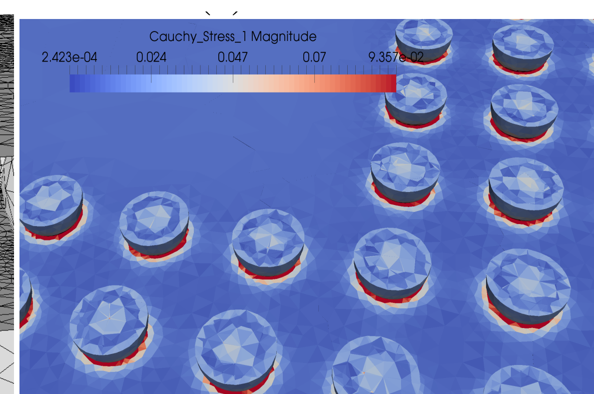
CAD Geometry



Mesh



Solution field

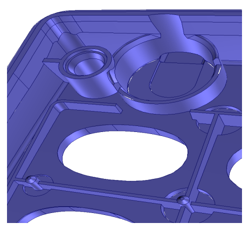


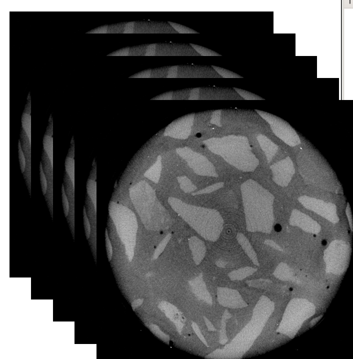
Geometry-Based Adaptive Simulation

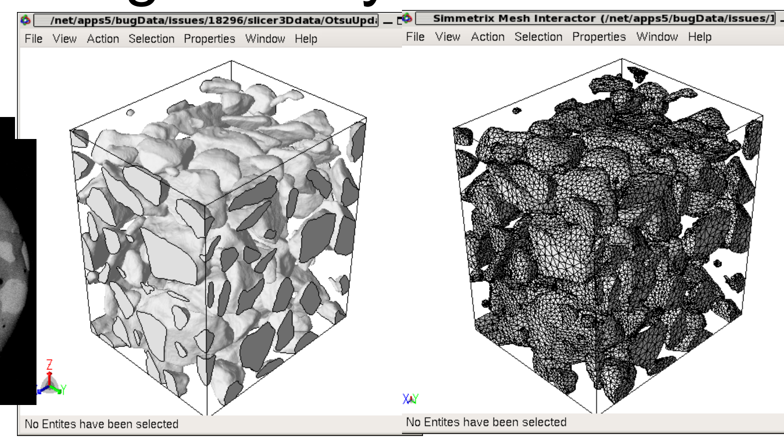
- Element geometric approximation order must be consistent with order of basis function
- Mesh generators for graded coarse curved meshes needed
- Curved mesh adaptation
- Mesh generation, mesh adaptation and element routines (for Jacobian calculation) need to interact with geometry

Typical geometry sources:

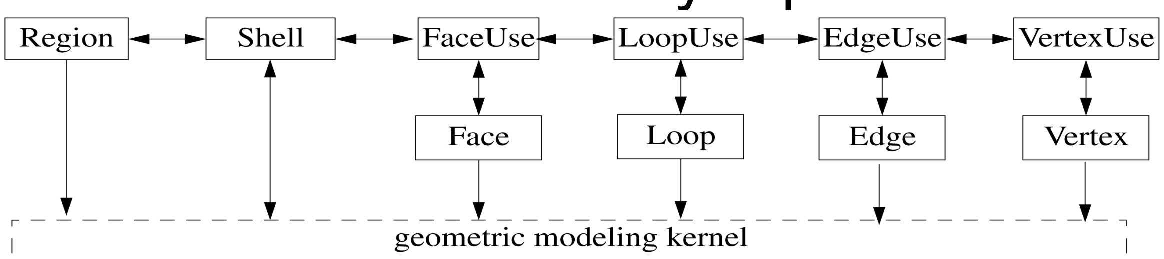
- CAD Geometric Models
- Image data
- Mesh Models

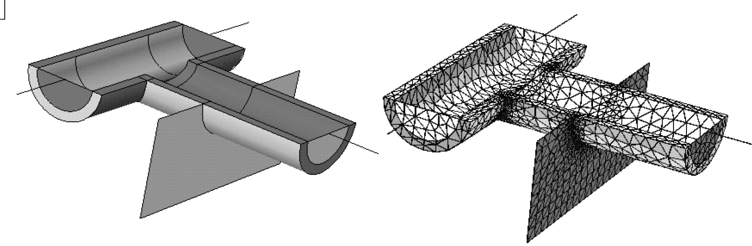






Non-manifold boundary representations provide ideal abstraction





Mesh/Model Relationship

Critical to simulation processes

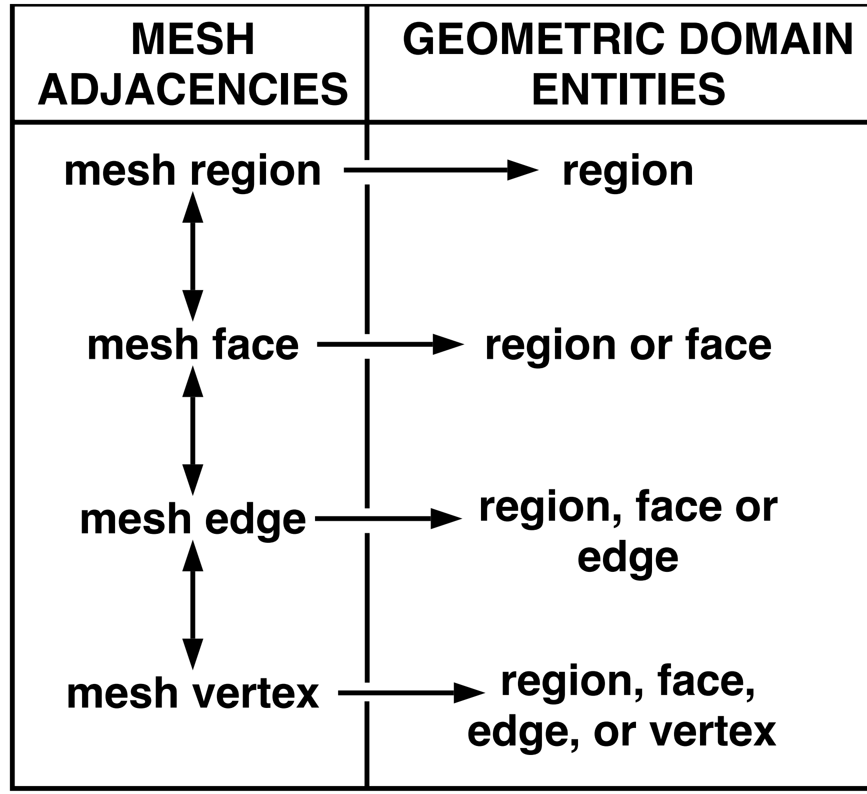
Relationship is termed "classification"

- Mesh Classification: Unique association of a mesh entity, $M_i^{d_i}$, to a geometric model entity, $G_j^{d_j}$, where $d_i<d_j$ is denoted by

  $$M_i^{d_i} \sqsubset G_j^{d_j}$$

  $\sqsubset$ indicates the left-hand entity (or set) represents a portion of the right-hand entity in the discretization

  Multiple $M_i^{d_i}$ classified on a $G_j^{d_j}$

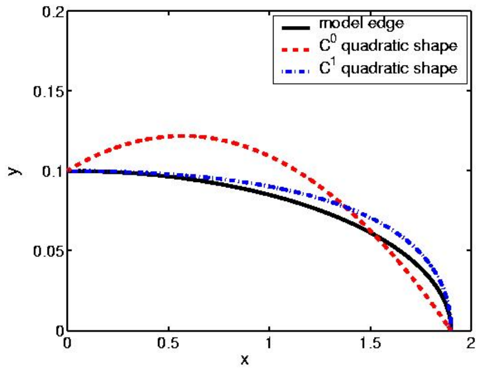


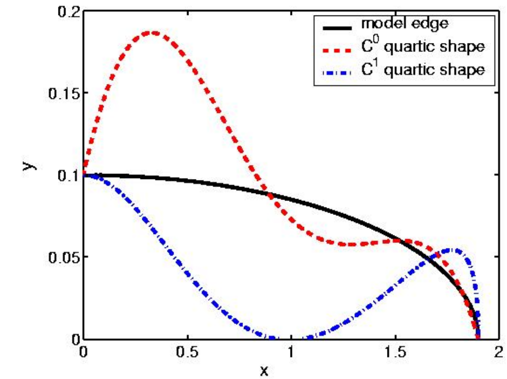
- Boundary mesh entities are identified in terms of their classifications
- Classification critical to supporting adaptive simulations and high level problem definitions

Curved Mesh Representations

There are multiple options for curved mesh representation

- Interpolation methods
  - Sensitive to interpolating points – not easy to select for curved CAD domains



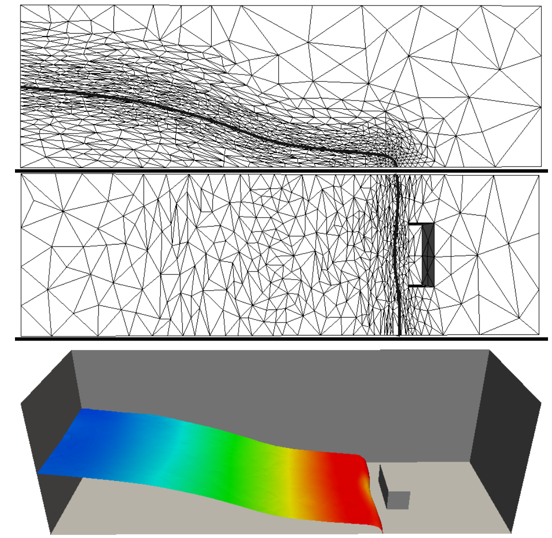


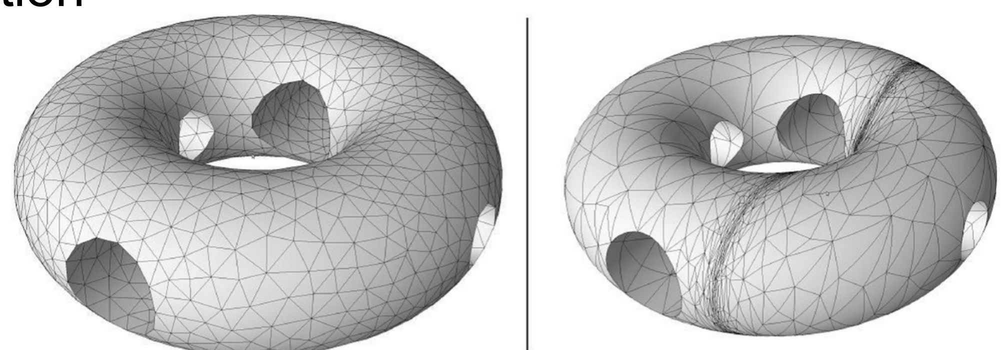
- Isogeometric
  - Requires basically a one-to-one correspondence between model and mesh entity parametric spaces – strongly constrains mesh configuration and not possible in general since CAD systems support Boolean operations that trim portions of parametric spaces in arbitrary ways and CAD systems do not parameterize volumes
- Employ geometric approximation of sufficient order for curved boundary representation – several options
  - Bezier polynomials are convenient
  - Rational polynomials possible – but unless properly aligned – still have approximation plus the cost of dealing with the rational functions

PUMI-MFEM Integration

Goal: Effective in-memory integration of PUMI based parallel mesh adaptation tools into MFEM

- **PUMI** - Parallel Unstructured Mesh Infrastructure
- **Key** components to be integrated into **MFEM** are:
  - Distributed mesh – interprocess communication, migration of mesh entities, remote read only copies
  - Link to the geometry and attributes
  - Mesh adaptation & mesh motion
  - Curved-geometry mesh adaptation
  - Dynamic load balancing
- Mesh adaptation driven by anisotropic mesh metric field





(a) (b)

Argonne NATIONAL LABORATORY BERKELEY LAB Sandia National Laboratories Rensselaer SMU

PUMI Software Pointers

Resources for PUMI:

- Overview: scorec.rpi.edu/pumi/
- Design, concepts, and applications: (TOMS journal paper) scorec.rpi.edu/REPORTS/2014-9.pdf
- Intro and user's guide: scorec.rpi.edu/pumi/pumi_intro.pdf, scorec.rpi.edu/pumi/PUMI.pdf
- APIs: scorec.rpi.edu/~seol/scorec/doxygen/
- Build instructions: github.com/SCOREC/core/wiki/General-Build-instructions
- Nightly regression: my.cdash.org/index.php?project=SCOREC
- Much more: github.com/SCOREC/core/wiki

Recent PUMI advances (its running on the latest Phi’s at Argonne and NERSC, there is also a GPU version):

- Thesis on array-based implementation using manycore & GPUs: scorec.rpi.edu/reports/view_report.php?id=710
- See Ibanez or Smith 2015 - 2017 papers: scorec.rpi.edu/reports/

MFEM/PUMI Integration: Tools Used

Tools currently being used/available in this process

- Curved mesh generation – Simmetrix has high quality quadratic mesh generation capabilities
- Tool to inflate the order of elements with Beziers to order 6 on fixed mesh topologies (Usually no problem for good quadratic curved meshes – but not guaranteed)
- Tools to take straight sided meshes and curve them – this includes mesh topology modification since that is needed in the majority of cases (curving often yields invalid elements)
- PUMI to manage the meshes in parallel
- Curved mesh adaptation based on mesh modification
- ParMA for mesh partition improvement

MFEM/PUMI Integration

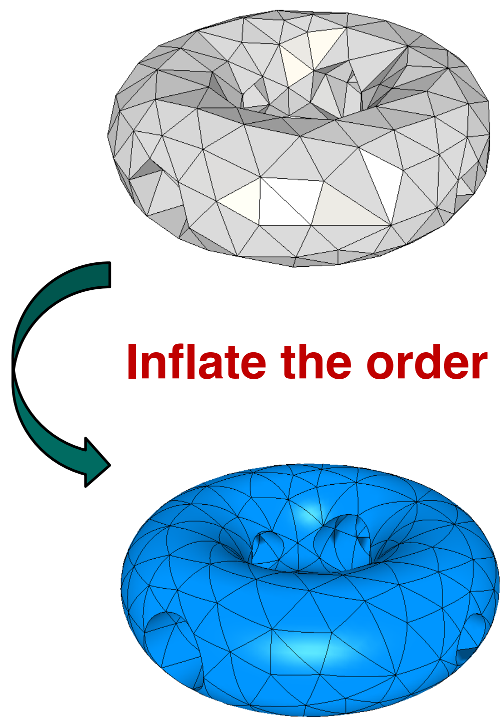
- Partition the mesh w load balancing. The mesh is classified on CAD model (PUMI)
- Curve the mesh up to order 6 (PUMI)
- Load the parallel mesh into MFEM format (MFEM-PUMI)
- Solve (MFEM)
- Transfer field to PUMI (MFEM-PUMI)
- AMR: error estimate, refine/coarse, snap, dynamic load balance (PUMI)
- Load new mesh to MFEM w updated field (MFEM-PUMI)

Mesh

PUMI → MFEM

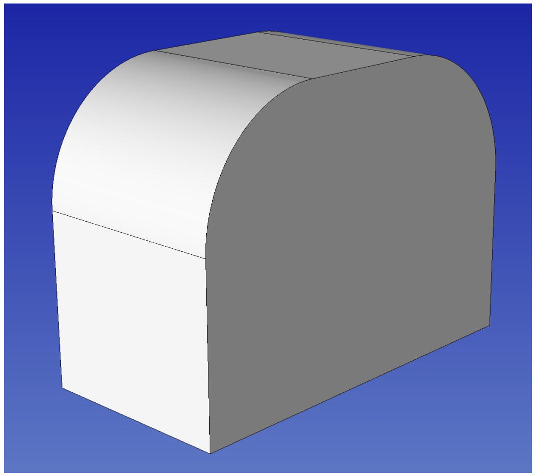
Field

PUMI ↔ MFEM

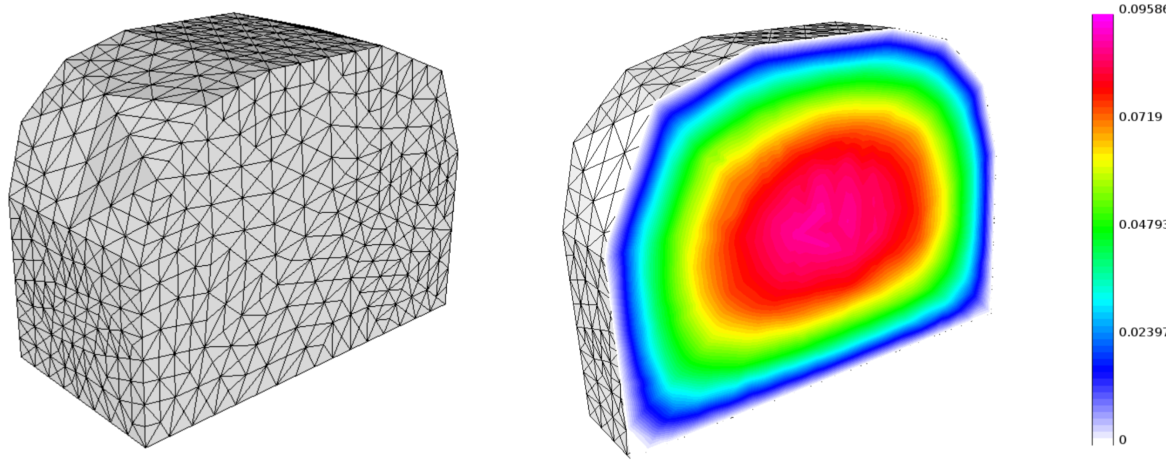


MFEM/PUMI Integration Simple Example

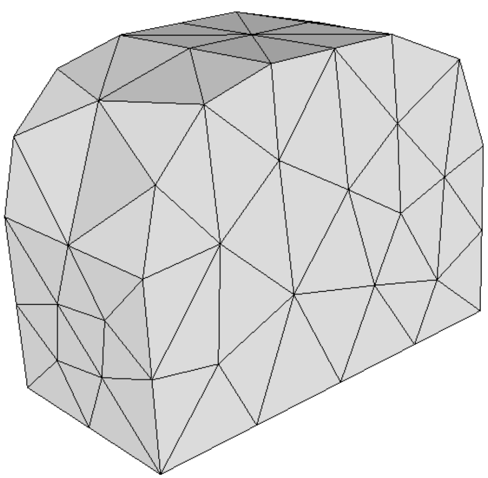
Simple example: Poisson + homogenous Dirichlet BCs.



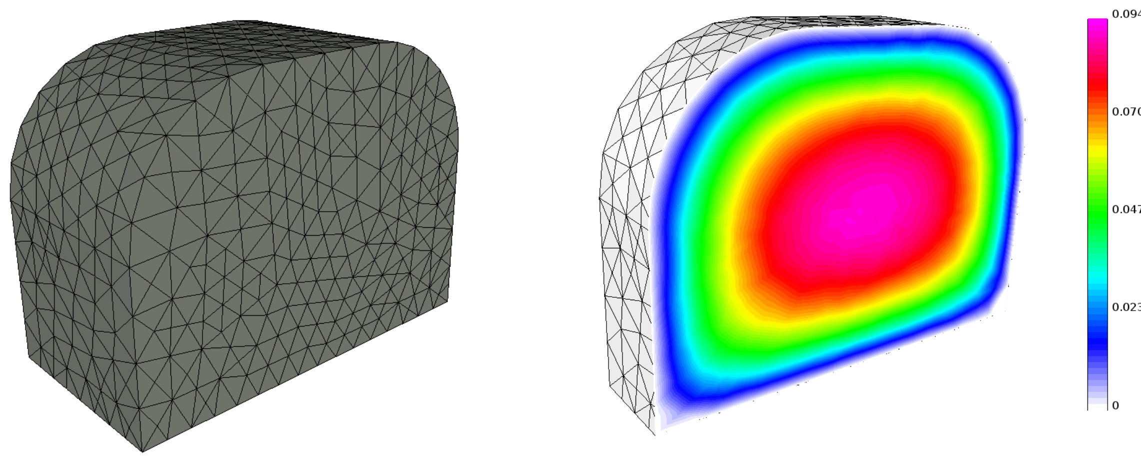
CAD Model



Refine against initial mesh geometry



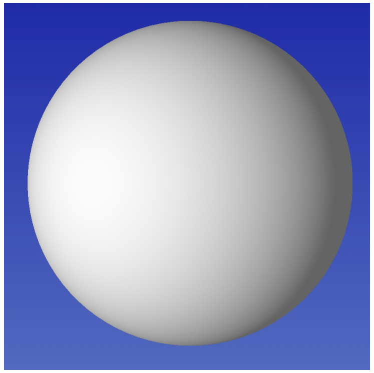
Initial Mesh



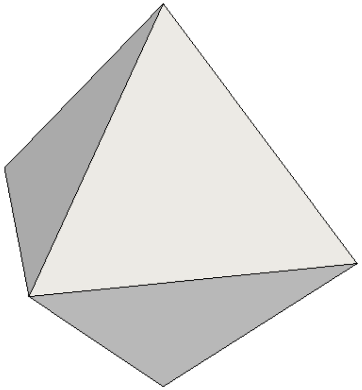
Refine against CAD using classification

Importance of Geometric Approximation

Solving same problem on a sphere meshed with 8 elements



Geometry

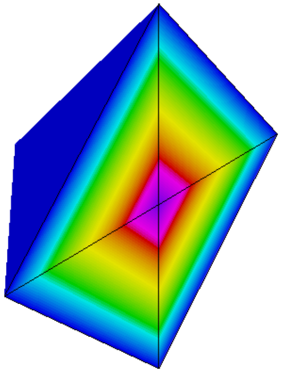


Linear mesh

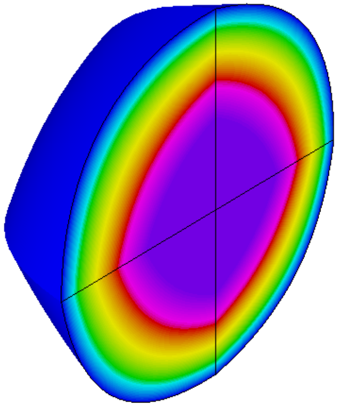
"G" : Geometric order

"P" : FE polynomial order

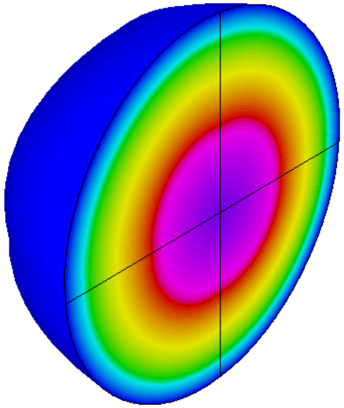
Analytical solution at center : 0.4166



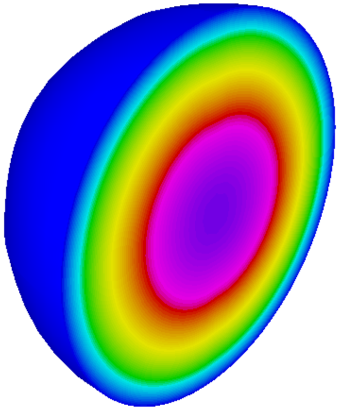
H1_3D_P1_G1



H1_3D_P2_G2



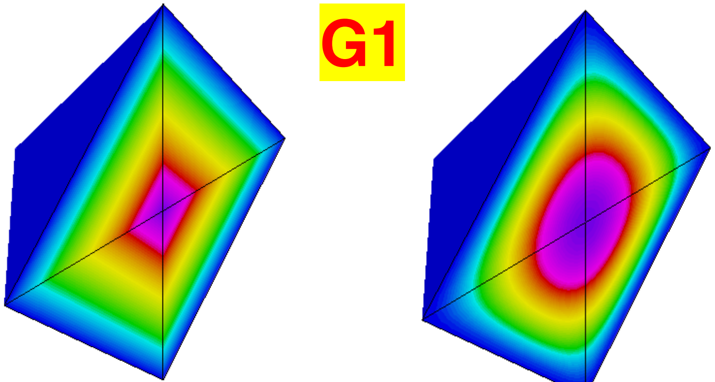
H1_3D_P3_G3



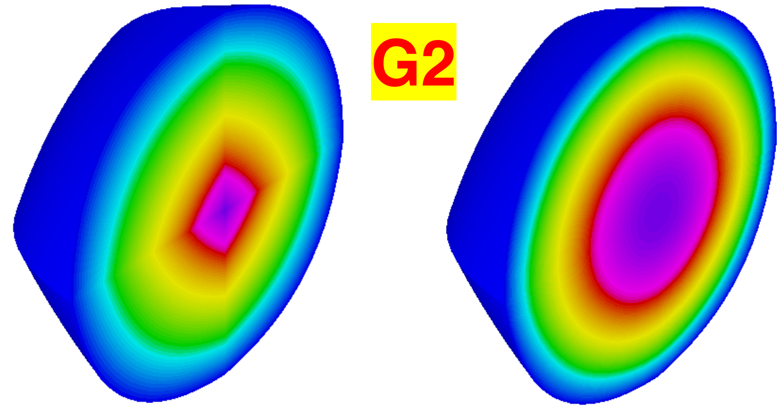
H1_3D_P4_G4

	P1G1	P2G2	P3G3	P4G4	P8G4
Center T	0.2083	0.2955	0.4214	0.4174	0.4156

Importance of Geometric Approximation



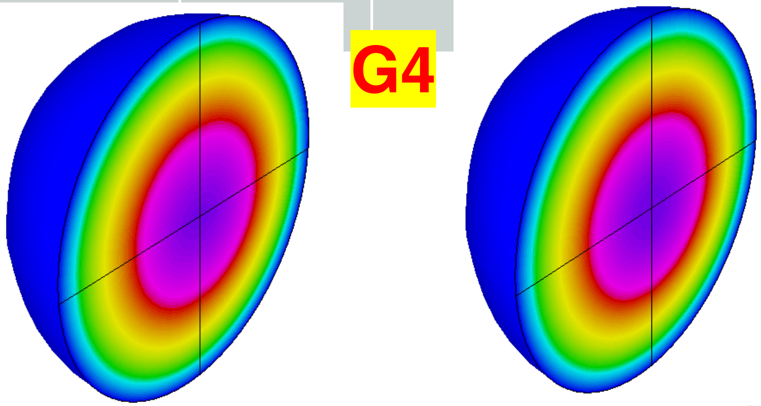
H1_3D_P1_G1 H1_3D_P8_G1



H1_3D_P1_G2 H1_3D_P8_G2

P1G1	P2G1	P3G1	P4G1	P8G1
0.2083	0.1389	0.1770	0.1773	0.1757

P1G2	P2G2	P3G2	P4G2	P8G2
0.5983	0.2955	0.3739	0.3701	0.3713



H1_3D_P3_G4 H1_3D_P8_G4

Analytical solution: 0.4166

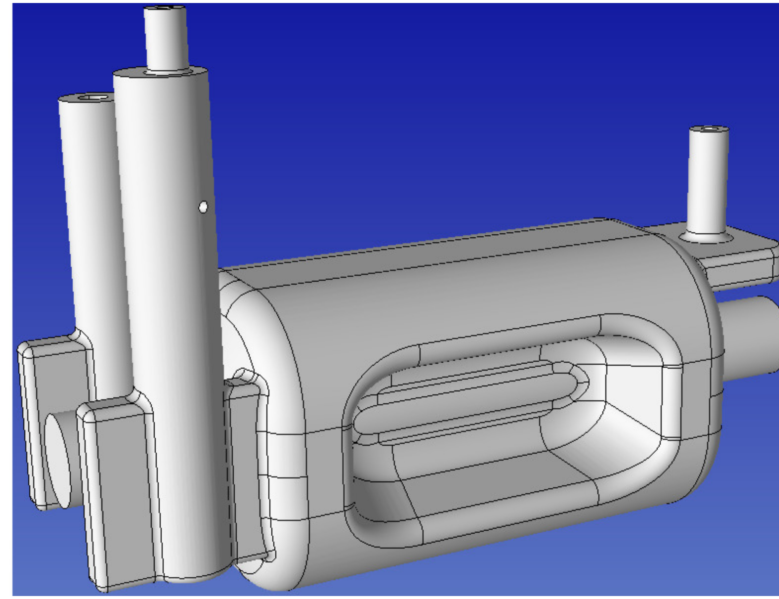
P1G4	P2G4	P3G4	P4G4	P8G4
0.2083	0.3471	0.4189	0.4174	0.4156

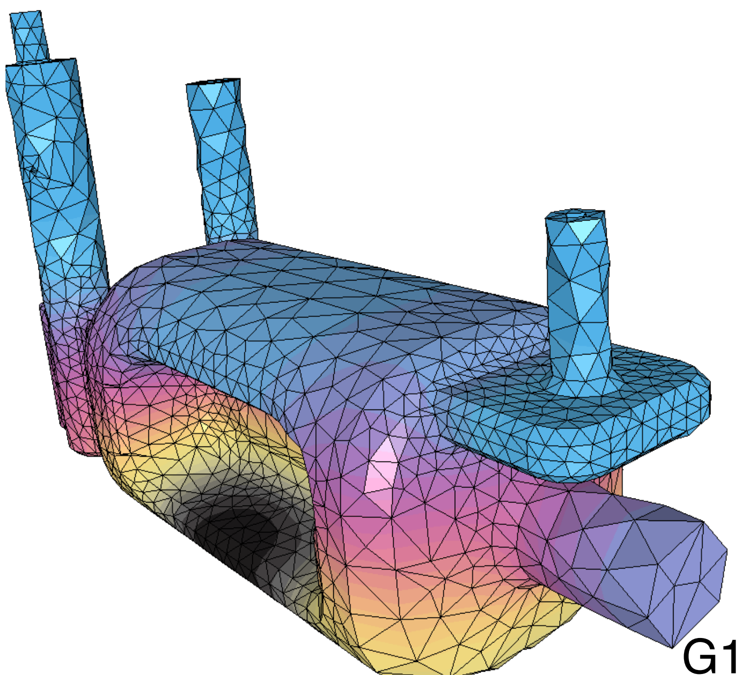
MFEM/PUMI Integration Example

Transient Nonlinear heat equation + Neumann BC.

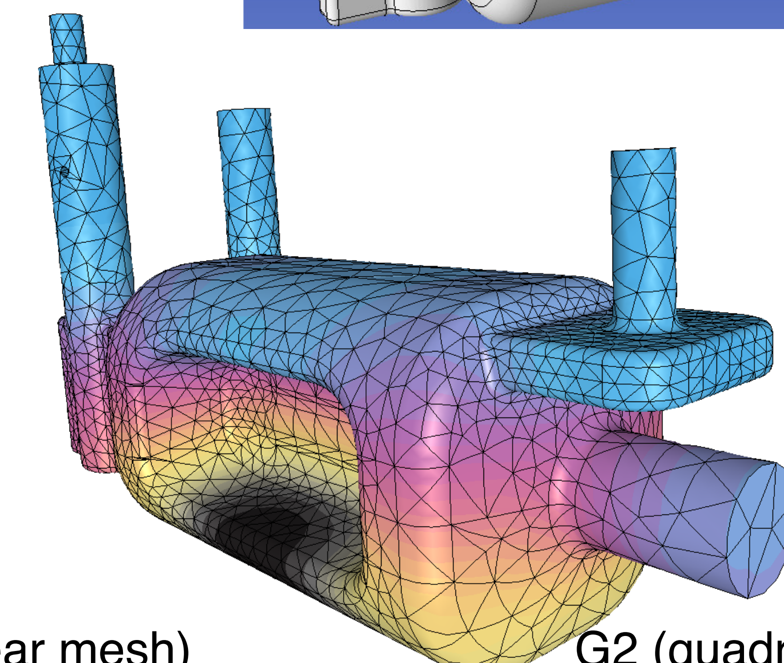
$$
\begin{cases}
\partial T/\partial t = \nabla.(\alpha T+\varepsilon)\nabla T & \text{in } \Omega \\
\partial T/\partial t = 0 & \text{in } \Gamma
\end{cases}
$$

Initial T : Lower 5, upper 1.





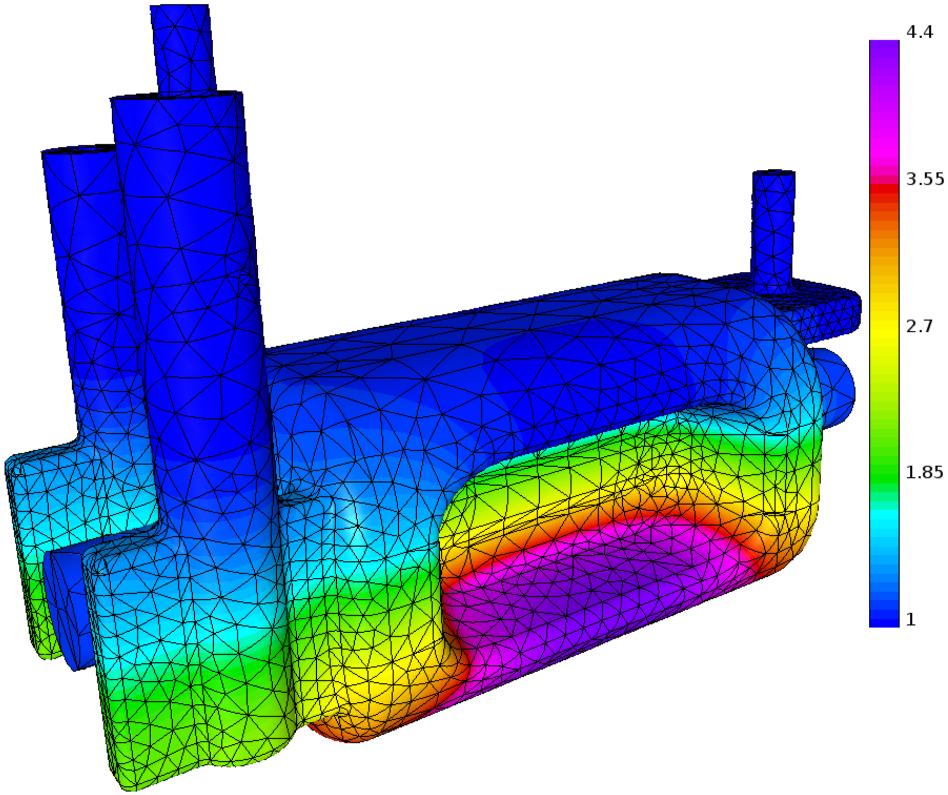
G1 (linear mesh)



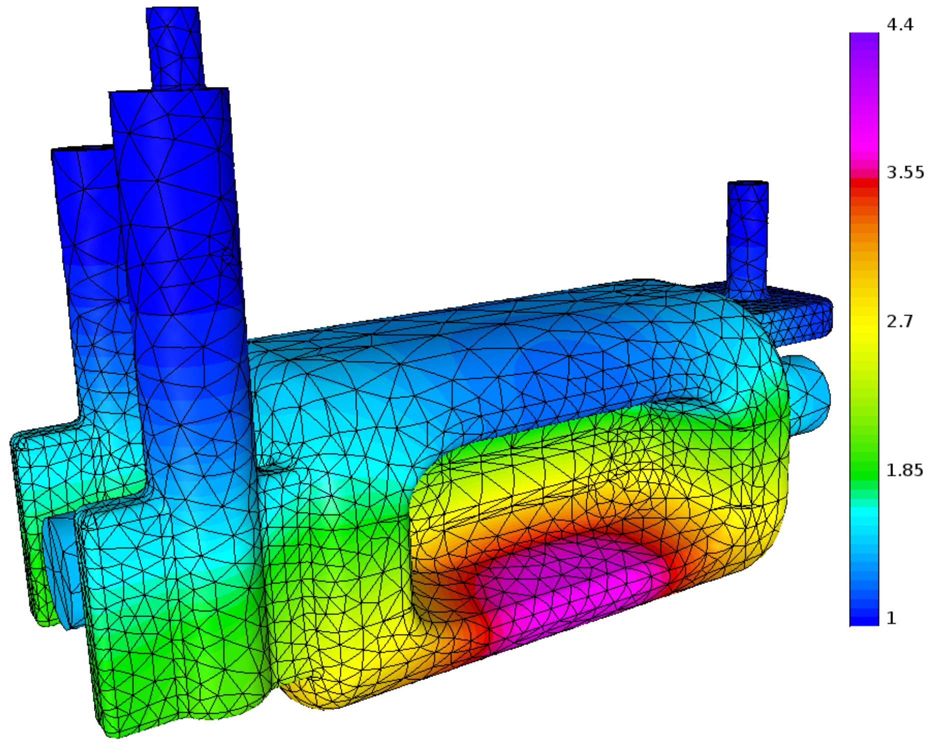
G2 (quadratic)

MFEM/PUMI Integration Example

G2_H2



T = 0.003



T = 0.006