Mesh Adaptivity for Scalable and Accurate High-Order Simulations

TETRAHEDRON VII
Workshop on Grid Generation for Numerical Computations
October 9-11, BSC-UPC, Barcelona, 2023

Barcelona, Oct 9-11, 2023

Tzanio Kolev

J. Camier, V. Dobrev, P. Knupp

K. Mittal, V. Tomov

LLNL-PRES-802003

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC

Lawrence Livermore National Laboratory

High-order finite elements are a good foundation for next-generation scalable multi-physics simulations

- **Large-scale parallel multi-physics simulations**
  - radiation diffusion
  - electromagnetic diffusion
  - compressible hydrodynamics

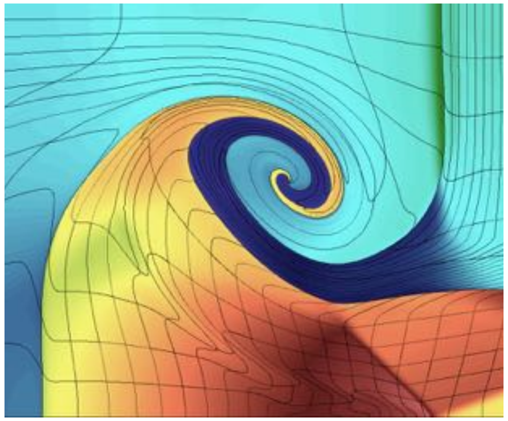
- **Finite elements naturally connect different physics**

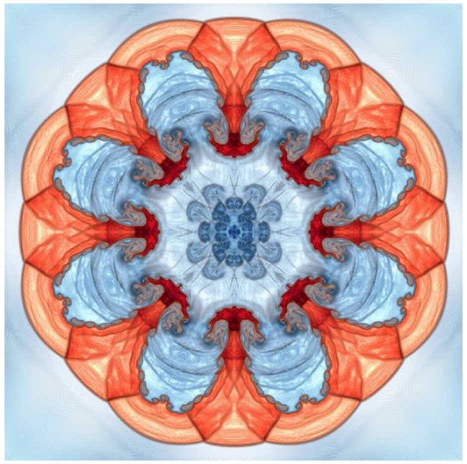
$$\begin{tikzcd} \boxed{\begin{array}{c} H(grad) \\ \text{“nodes”} \end{array}} \arrow[r, "\nabla"] & \boxed{\begin{array}{c} H(curl) \\ \text{“edges”} \end{array}} \arrow[r, "\nabla\times"] & \boxed{\begin{array}{c} H(div) \\ \text{“faces”} \end{array}} \arrow[r, "\nabla\cdot"] & \boxed{\begin{array}{c} L_2 \\ \text{“elems”} \end{array}} \end{tikzcd}$$

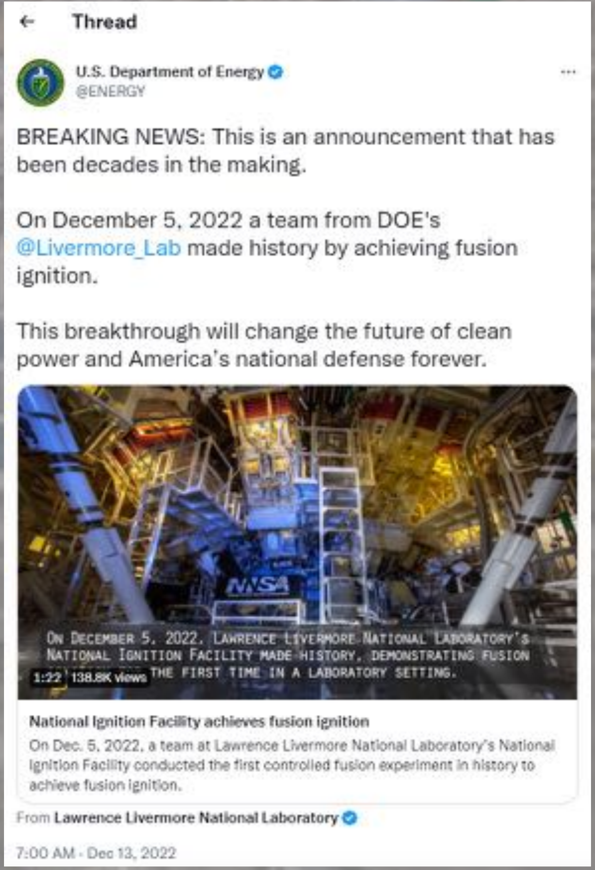
High-order kinematics	High-order MHD	High-order rad. diffusion	High-order thermodynamics

- **High-order finite elements on high-order meshes**
  - increased accuracy for smooth problems
  - sub-element modeling for problems with shocks
  - HPC utilization, FLOPs/bytes increase with the order

- **Need new (interesting!) R&D for full benefits**
  - meshing, discretizations, solvers, AMR, UQ, visualization, …


8th order Lagrangian simulation of shock triple-point interaction


Inertial Confinement Fusion


Thread

U.S. Department of Energy
@ENERGY

BREAKING NEWS: This is an announcement that has been decades in the making.

On December 5, 2022 a team from DOE's @Livermore_Lab made history by achieving fusion ignition.

This breakthrough will change the future of clean power and America's national defense forever.

On December 5, 2022, Lawrence Livermore National Laboratory's National Ignition Facility made history, demonstrating fusion ... the first time in a laboratory setting.

1:22 138.8K views

National Ignition Facility achieves fusion ignition

On Dec. 5, 2022, a team at Lawrence Livermore National Laboratory's National Ignition Facility conducted the first controlled fusion experiment in history to achieve fusion ignition.

From Lawrence Livermore National Laboratory

7:00 AM · Dec 13, 2022

lasers.llnl.gov

We model shock hydrodynamics with high-order FEM in both Lagrange and Remap ALE phases

Lagrange phase

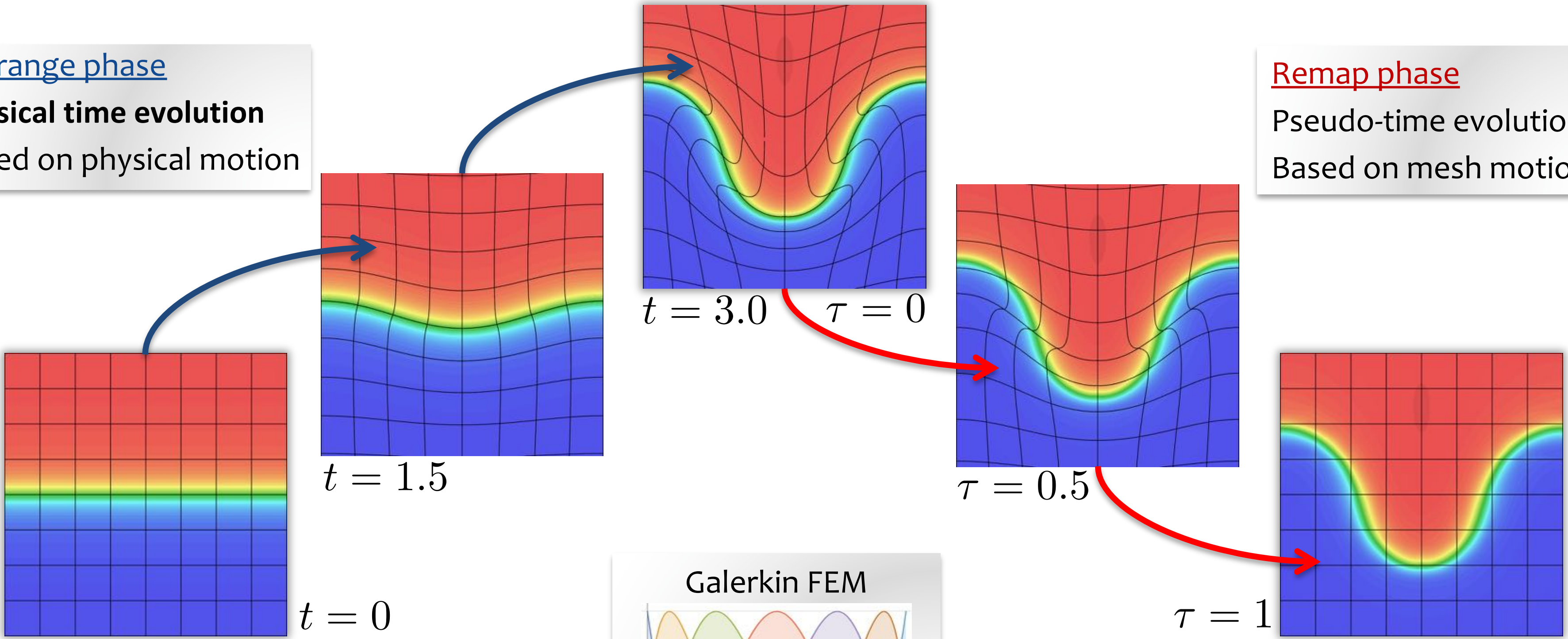
Physical time evolution

Based on physical motion

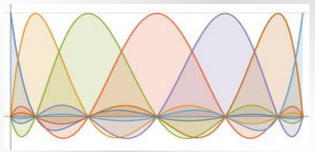
Remap phase

Pseudo-time evolution

Based on mesh motion

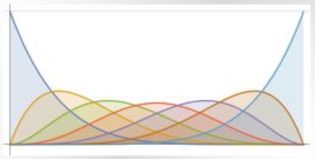


Galerkin FEM



Gauss-Lobatto basis

DG



Bernstein basis

Lagrangian phase ($\vec{c} = \vec{0}$)

Momentum Conservation:	$\rho \frac{\mathrm{d}\vec{v}}{\mathrm{d}t} = \nabla \cdot \sigma$
Mass Conservation:	$\frac{\mathrm{d}\rho}{\mathrm{d}t} = -\rho \nabla \cdot \vec{v}$
Energy Conservation:	$\rho \frac{\mathrm{d}e}{\mathrm{d}t} = \sigma : \nabla \vec{v}$
Equation of Motion:	$\frac{\mathrm{d}\vec{x}}{\mathrm{d}t} = \vec{v}$

Advection phase ($\vec{c} = -\vec{v}_m$)

Momentum Conservation:	$\frac{\mathrm{d}(\rho\vec{v})}{\mathrm{d}\tau} = \vec{v}_m \cdot \nabla(\rho\vec{v})$
Mass Conservation:	$\frac{\mathrm{d}\rho}{\mathrm{d}\tau} = \vec{v}_m \cdot \nabla\rho$
Energy Conservation:	$\frac{\mathrm{d}(\rho e)}{\mathrm{d}\tau} = \vec{v}_m \cdot \nabla(\rho e)$
Mesh velocity:	$\vec{v}_m = \frac{\mathrm{d}\vec{x}}{\mathrm{d}\tau}$

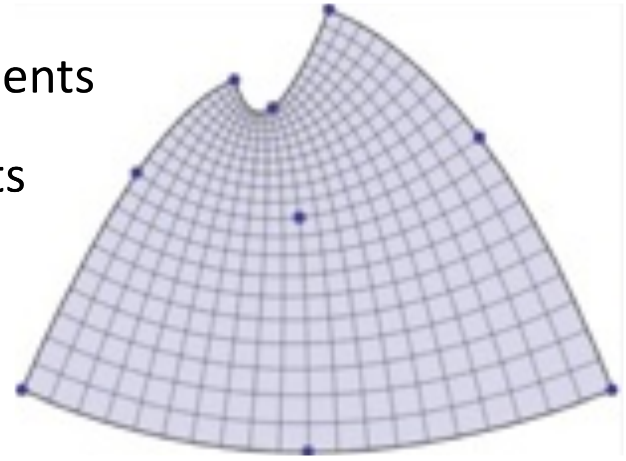
"High-order multi-material ALE hydrodynamics", SISC 2018

High–order mesh representation

- High-order mesh positions are discretized via position vector and a FE basis:

$$\boldsymbol{x} = (\boldsymbol{x}_1 \dots \boldsymbol{x}_N)^T, \quad x_q(\bar{x}_q) = \sum_{i=1}^{N} \boldsymbol{x}_i \bar{w}_i(\bar{x}_q)$$

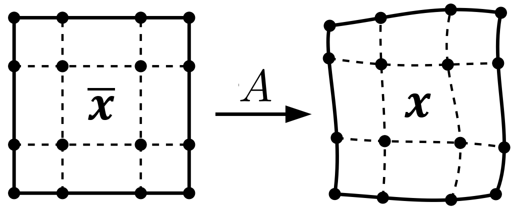
- $\{\bar{w}_i\}_1^{N_E}$ spans Q_k for quadrilateral / hexahedral elements
- $\{\bar{w}_i\}_1^{N_E}$ spans P_k for triangular / tetrahedral elements



Example of a single Q_2 element

- Reference -> physical Jacobian is given by the basis functions' gradients:

$$A_q(x) = \frac{\partial x_q}{\partial \bar{x}_q} = \sum_{i=1}^{N} \boldsymbol{x}_i [\nabla \bar{w}_i(\bar{x}_q)]^T$$



3rd order transformation

- Mesh is optimized, by node movement (changing $\boldsymbol{x}$)
- Topology is preserved.

Many papers by: ***Shephard***, ***Roca***, ***Geuzaine***, ***Shontz***, ***Johnen***, ***Persson***, ***Panozzo***

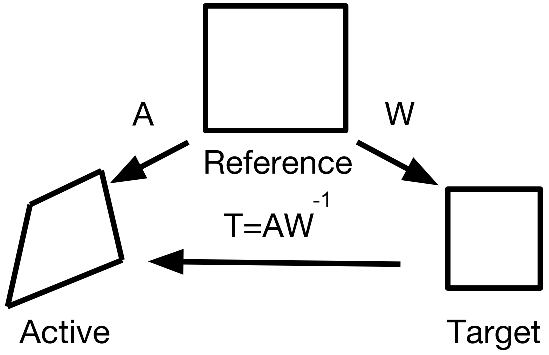
Target-Matrix Optimization Paradigm (TMOP)

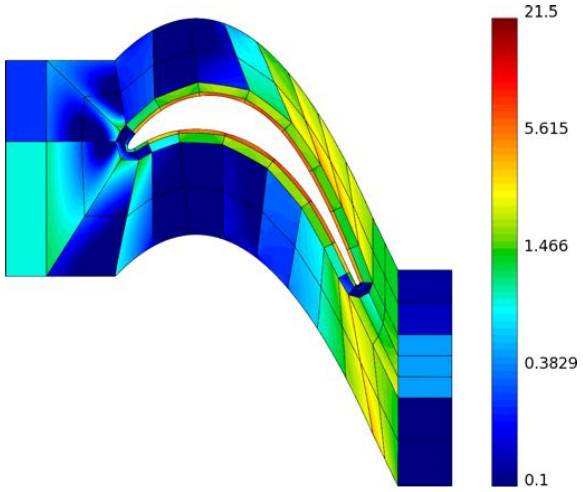
- *Target-Matrix Optimization Paradigm (TMOP)*
  - Extended P. Knupp's theory to high-order meshes.
- *Application-specific target elements, W*
  - Allow tailoring to different apps.
  - Examples: ideal, ideal + specified size.
- *Point-based mesh quality metric $\mu(T)$*
  - Can measure shape, size and alignment independently. computed on quadrature point level. Examples:

$$\mu_2^{shape} = \frac{|T|^2}{2\det(T)} - 1 \quad \mu_{55}^{size} = 0.5\,(\det(T) - 1)^2$$

- *Global quality functional and minimization*
  - Hessian-based methods need $\partial^2\mu/\partial T^2$.

$$\frac{\partial F(\boldsymbol{x})}{\partial \boldsymbol{x}} = 0\,, \quad \text{where} \quad F(x) = \sum_K \int_{K_t} \mu(T(x))$$





TMOP mesh quality metrics

- We have explored more than 60 metrics divided into 7 metric types
- Jacobian decomposition: $W =$ [volume] [orientation] [skew] [aspect ratio].
- Shape metrics – control over skew and aspect ratio.
  Minimized when A is a scaled rotation of W.

$$\mu_2(T) = 0.5\frac{|T|^2}{\det(T)} - 1$$

- Size metrics – control over volume.
  Minimized when $\det(A) = \det(W)$.

$$\mu_{77}(T) = 0.5\left(\det(T) - \frac{1}{\det(T)}\right)^2$$

- Alignment metrics – control over orientation and skew.
  Minimized when $A = \mathrm{W} * \mathrm{Diag}$.

$$\mu_{30}(A,W) = |\boldsymbol{a}_1||\boldsymbol{w}_1| - (\boldsymbol{a}_1 \cdot \boldsymbol{w}_1) + |\boldsymbol{a}_2||\boldsymbol{w}_2| - (\boldsymbol{a}_2 \cdot \boldsymbol{w}_2)$$

- Implicit + explicit combinations.
  SH+SZ, SH+AL, SZ+AL, SH+SZ+AL.

$$\mu_7(T) = |T - T^{-t}|^2 \qquad \mu_{14}(T) = |T - I|^2$$

$$\mu(T) = \alpha\mu_i(T) + (1-\alpha)\mu_j(T)$$

"Algebraic mesh quality metrics", Knupp, SISC, 2001
"17 criteria for evaluating Jacobian-based optimization metrics", Knupp, EngComp, 2023

TMOP geometric parameters

- **Extraction map:** given A, extract (volume, orientation, skew, aspect ratio)

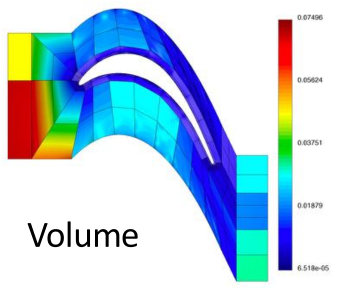
$$E(A_{2\,X\,2}) = (\upsilon, \theta, \phi, \rho)\;\epsilon\; P \;\subset\; R^4$$

- **Insertion map:** given the geometric parameters, build matrix A

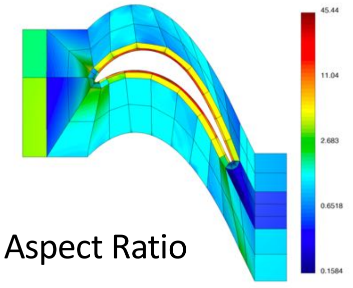
$$E^{-1}(\upsilon, \theta, \phi, \rho) = A = \sqrt{\frac{\upsilon}{\sin\phi}}\begin{pmatrix} \dfrac{\cos\theta}{\sqrt{\rho}} & \sqrt{\rho}\cos(\theta+\phi) \\ \dfrac{\sin\theta}{\sqrt{\rho}} & \sqrt{\rho}\sin(\theta+\;\phi) \end{pmatrix}$$

- Extraction map for $A_{3\,X\,3}$ exists as well. For valid A, mapping is one-to-one.

- Applications:
  - Target construction
  - Metric type classification
  - Geometric mesh quality visualization



Volume

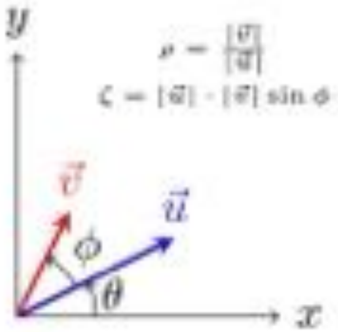


Aspect Ratio

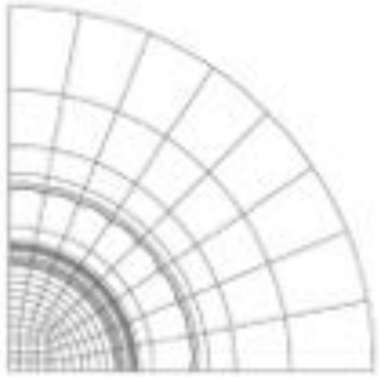
"Geometric parameters in the target matrix mesh optimization paradigm", Knupp, PDEAM, 2022

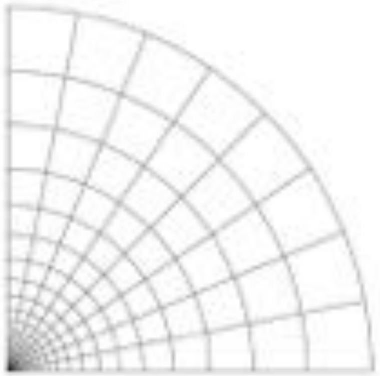
Geometric target construction

$$W = \sqrt{\frac{\zeta}{\sin\phi}} \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} 1 & \cos\phi \\ 0 & \sin\phi \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{\rho}} & 0 \\ 0 & \sqrt{\rho} \end{bmatrix}$$

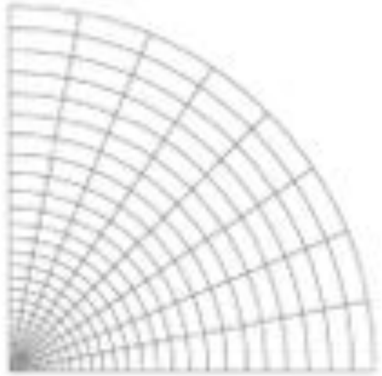


y
$\rho = \frac{|\vec{v}|}{|\vec{u}|}$
$\zeta = |\vec{u}| \cdot |\vec{v}| \sin\phi$
$\vec{v}$
$\vec{u}$
ϕ
θ
x

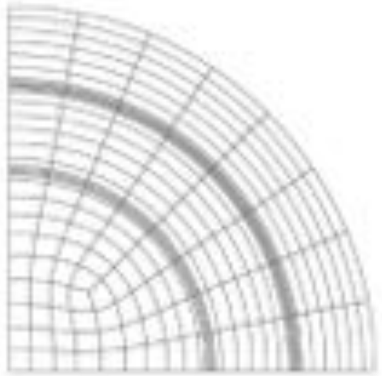

Original 2nd order mesh


Ideal shape + shape-metric.

$\phi = \frac{\pi}{2}, \rho = 1, \mu_{Sh}(T)$

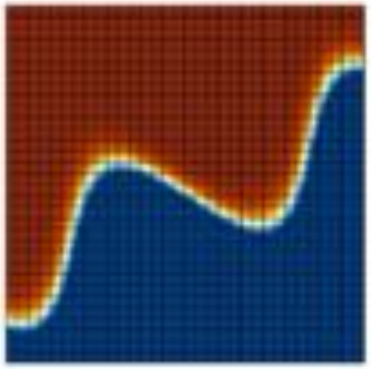

Ideal shape, equal size + shape-metric.

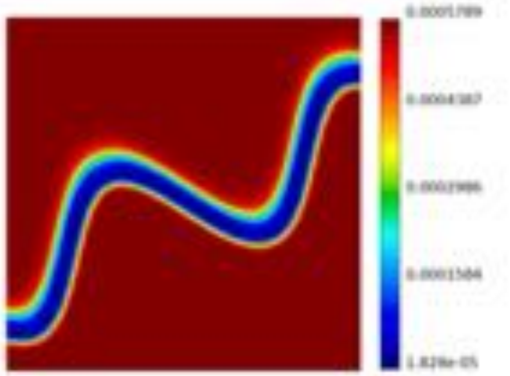
$\zeta = \frac{V}{N_E}, \phi = \frac{\pi}{2}, \rho = 1, \mu_{ShSz}(T)$


Ideal shape, spatially varying size + shape+size -metric.

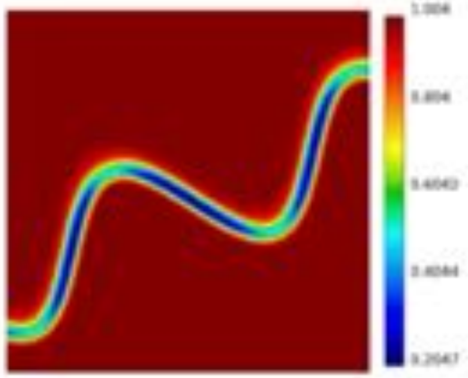
$\zeta(\mathbf{x}), \phi = \frac{\pi}{2}, \rho = 1, \mu_{ShSz}(T)$

Simulation-driven target construction


Simulation data material indicator (η)



Size - $\zeta \propto 1/|\nabla \eta|$



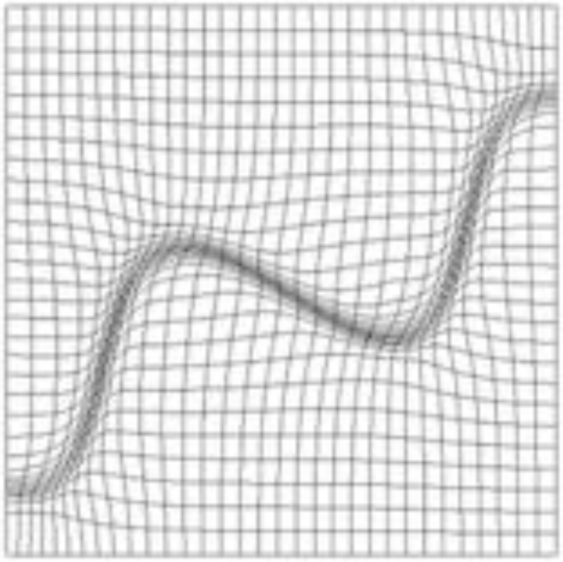
Aspect-Ratio - $\rho \propto |\eta_x/\eta_y|$

$$W = \sqrt{\frac{\zeta}{\sin\phi}} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & \cos\phi \\ 0 & \sin\phi \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{\rho}} & 0 \\ 0 & \sqrt{\rho} \end{bmatrix},$$

- $\phi = \frac{\pi}{2}$ for an ideal square.
- Use a Shape + Size polyconvex metric, $\mu_{80} = (1-\gamma)\mu_2 + \gamma\mu_{77}$.

$$\mu_2(T) = 0.5\frac{|T|^2}{\det(T)} - 1 \qquad \mu_{77}(T) = \frac{1}{2}(\tau - \frac{1}{\tau})^2$$

- Note: η must be remapped between and after Newton iterations.



Discrete field transfer between meshes

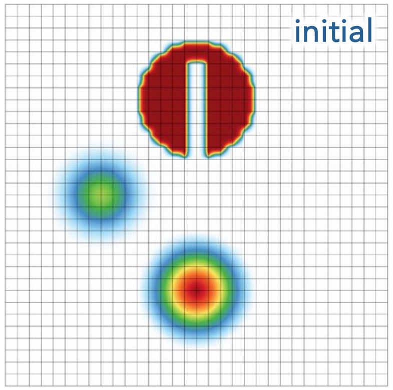
Major difficulty: discrete fields are defined only on the starting mesh.

We have two approaches to address this.

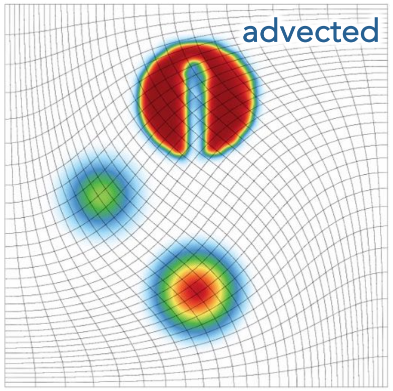
Advection PDE

- Transfer between meshes of same topology
- Galerkin-based formulation of advection
- Based on remap algorithm from high-order ALE
- Ensures conservation

$$\frac{\partial \eta}{\partial \tau} = \mathbf{u} \cdot \nabla \eta, \;\; \eta(\mathbf{x}_0, \tau = 0) = \eta_0(\mathbf{x}_0)$$



$\eta(\mathbf{x}_0, \tau = 0)$

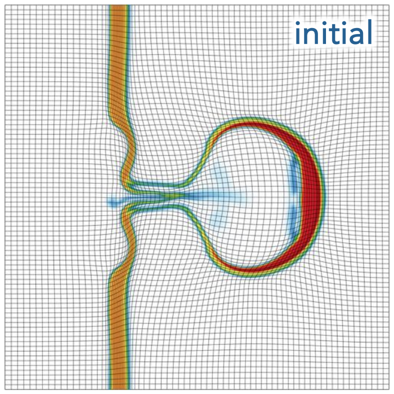


$\eta(\mathbf{x}, \tau = 1)$

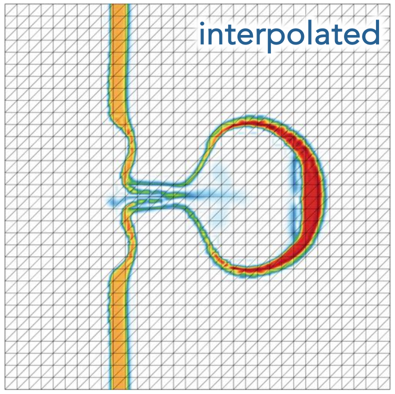
High-order interpolation

- Transfer between meshes of different topology
- Supports grid functions in the entire de Rham complex → L^2, H^1, H(*div*), H(*curl*)
- Supports non-conforming AMR meshes
- Based on *gslib*, part of ECP/CEED

$$\mathcal{M}_0, \;\; u_0(\mathcal{M}_0), \;\; \mathcal{M} \rightarrow u(\mathcal{M})$$



$\mathcal{M}_0$

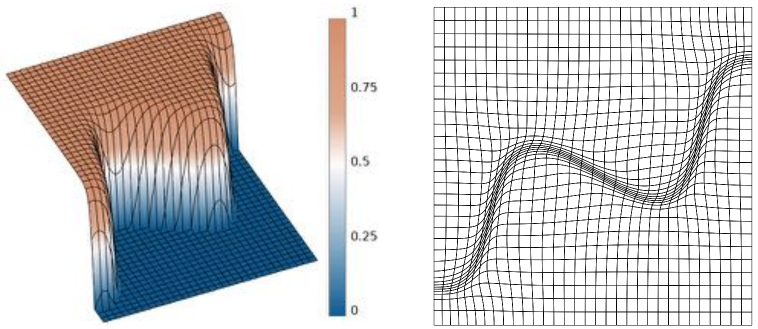


$\mathcal{M}$

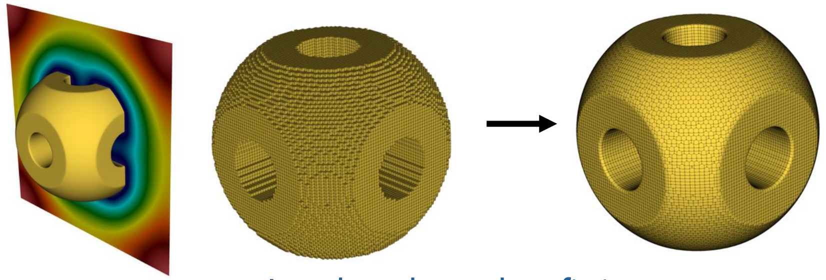
Simulation-driven mesh adaptivity

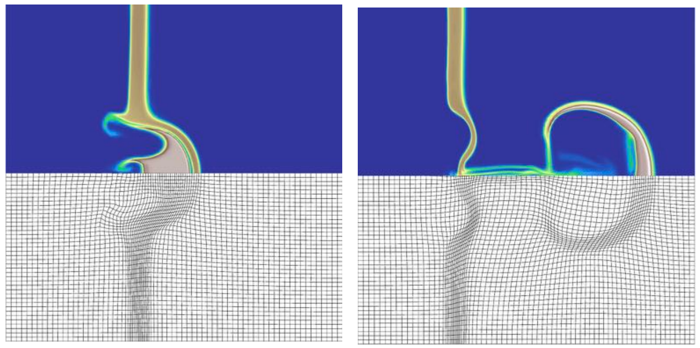
- ***Simulation-driven mesh optimization***
  - Driven by information provided by the simulation
  - Adapt to dynamic / discrete simulation features
  - Material interfaces, shocks, physics fields, etc.

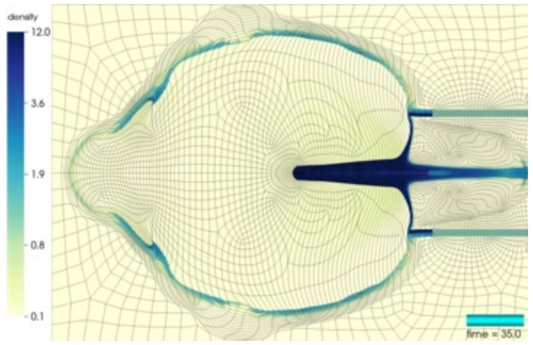
- ***Recent Advancements***
  - Application-specific targets
  - Foundational TMOP theory and algorithms
  - Discrete field transfer between meshes
  - Limiting and interface/boundary fitting
  - *hr*-adaptivity and *rp*-adaptivity



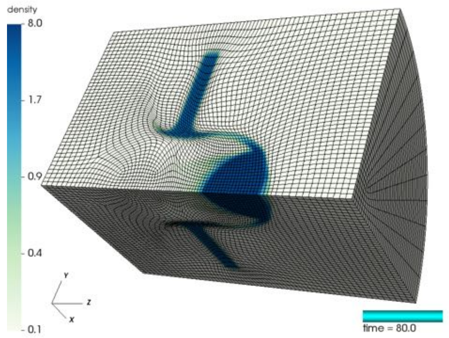
Size + aspect ratio adaptivity to a material interface


Level-set boundary fitting


Adaptivity to dynamic material positions



Moving mesh simulation

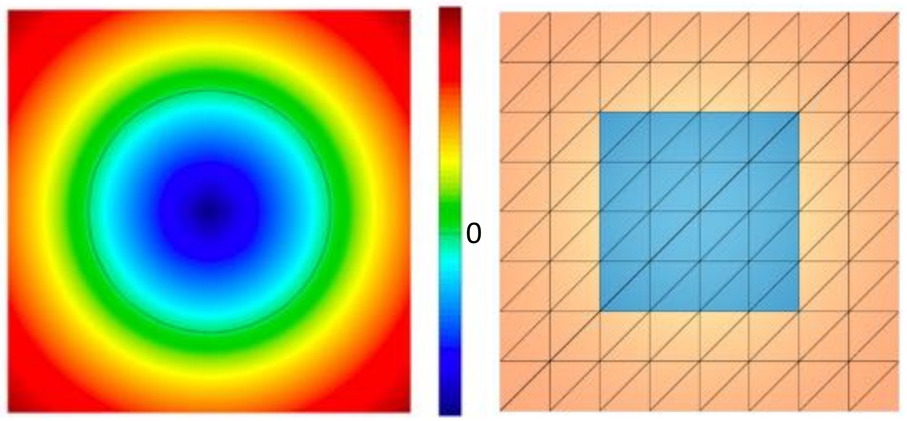


3D high-velocity ball impact

"hr-adaptivity for nonconforming high-order meshes with TMOP", Eng. Comp, 2021

Adaptive surface fitting

- Our approach for boundary and interface fitting is to fit the mesh to surface of interest given as the zero level set of a discrete function $\sigma(\mathbf{x})$, using a penalty-based formulation



$\sigma(\mathbf{x})$ describing target interface and mesh to be optimized

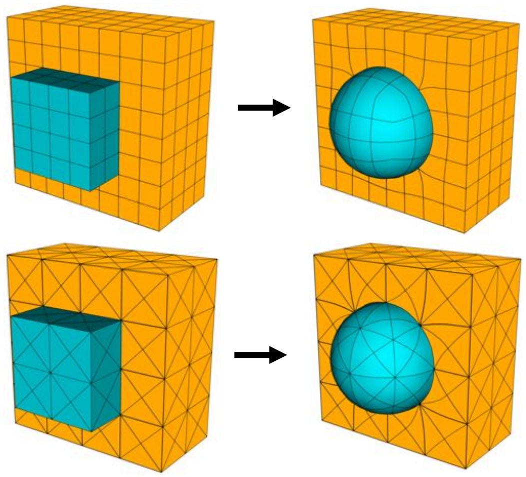
$$F(\mathbf{x}) = \sum_{E \in \mathcal{M}} \int_{E_t} \mu(T(x)) dx_t + w_\sigma \sum_{s \in \mathcal{S}} \sigma^2(x_s)$$

σ– Level set function *$\mathcal{S}$– Nodes marked for fitting* *w_σ – Penalization weight*

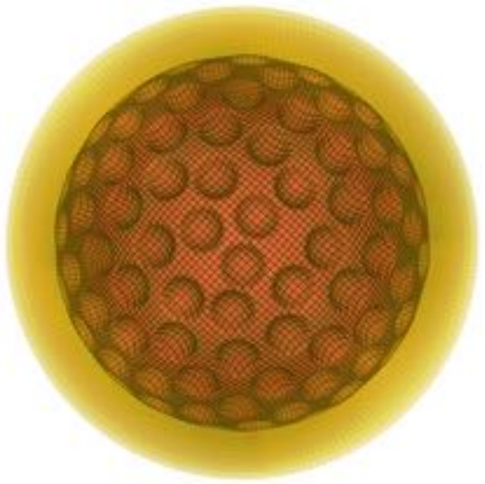
- Various ways to compute $\sigma(\mathbf{x})$, e.g. distance solver for *internal* interfaces
- Preserves topology, good shape quality and appropriate local size
- Mostly finite element operations: generality (dim, order), GPU and PA

Internal interface fitting

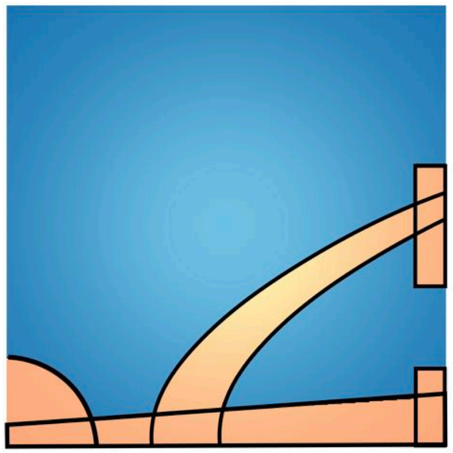
- Fitting of a simple internal interface



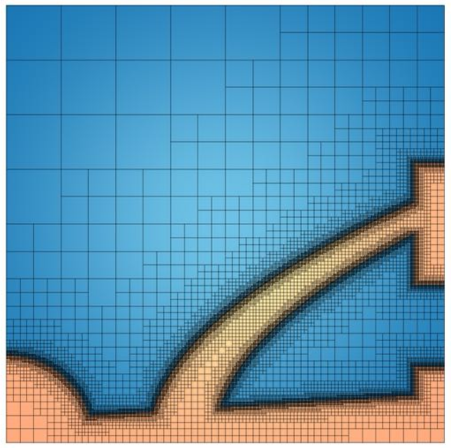
- Fitting of complex 3D interface



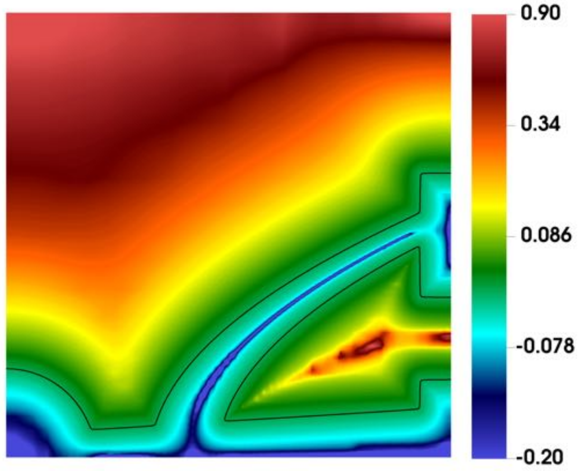
- We use geometric primitives to define the level set function for complex domains



Reactor spoke CSG decomposition



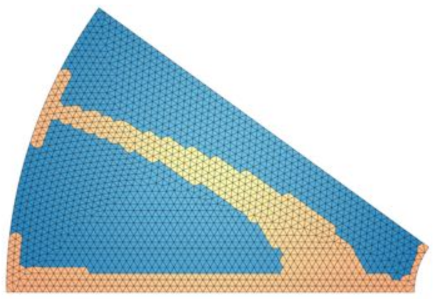
AMR around the zero level set



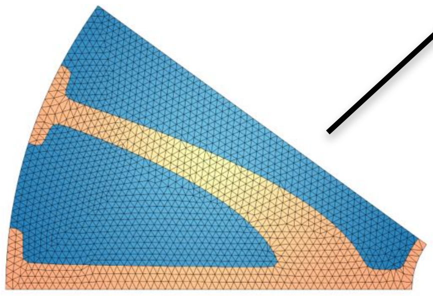
Distance function from zero level set

Interface fitting for a reactor design problem

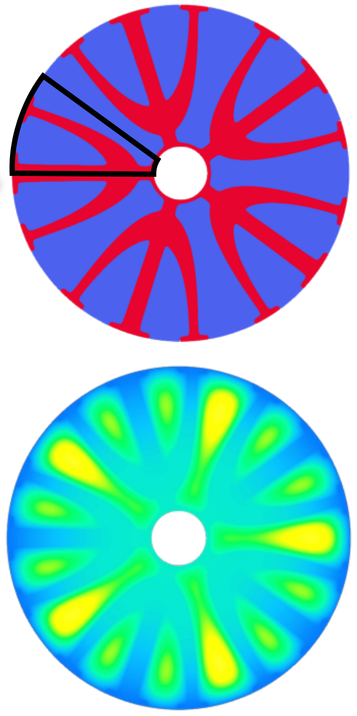
- Reactor design problem: Maximize the energy produced by the system (blue region) while keeping the volume of the aluminum fins fixed (*red* in plots below).
- We first generate a uniform mesh and optimize it to get an initial mesh to be used for the reactor design problem.



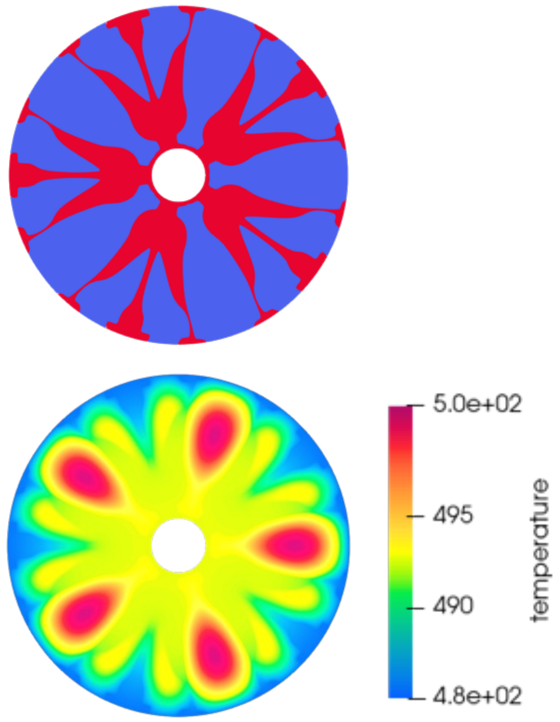
Initial mesh



Interface fitting mesh

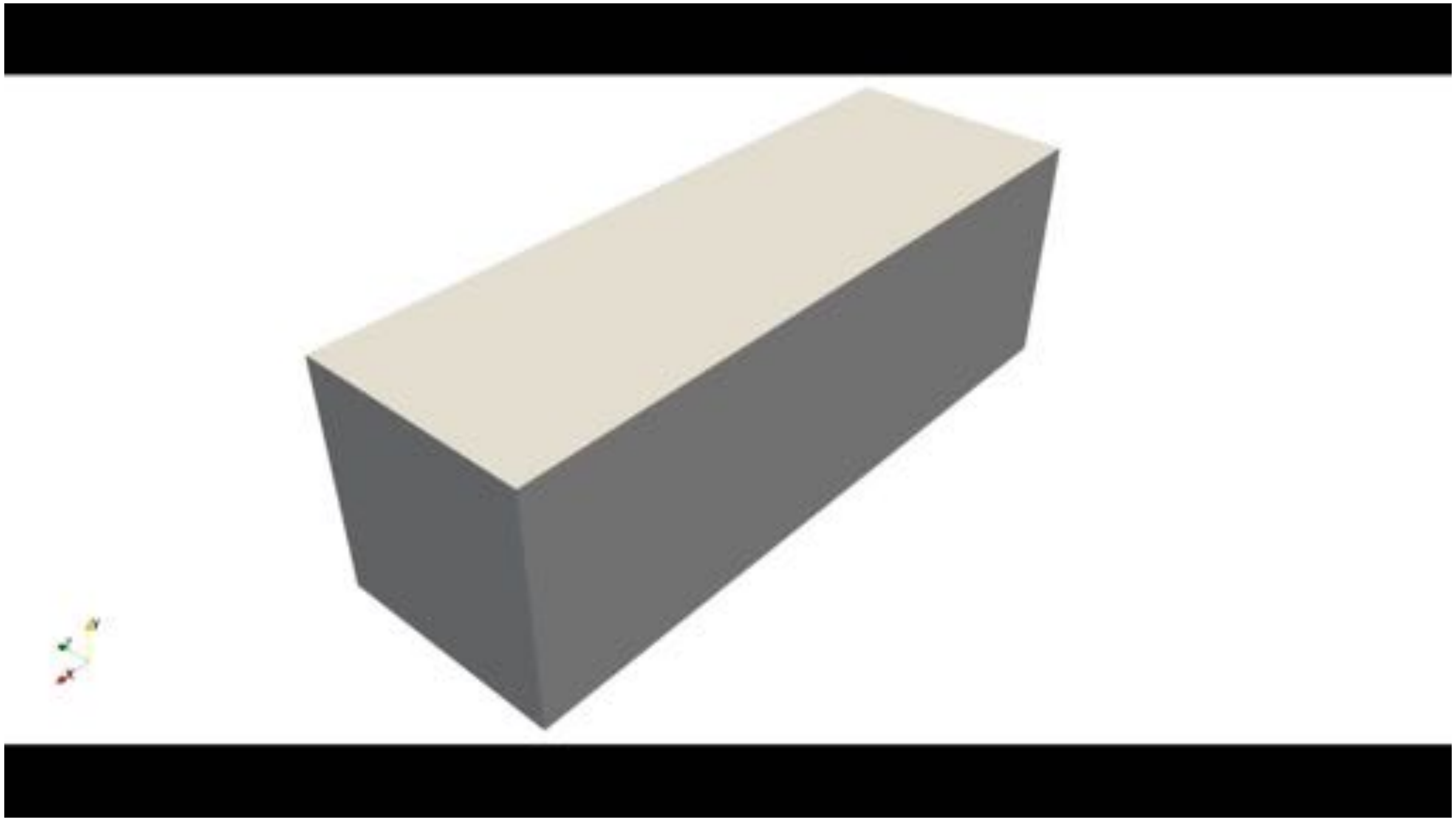


Initial fitted mesh



Fitted mesh optimized for energy production

Shape optimization using conformal meshes



Conformal meshing using r-adaptivity for shape optimization of a beam to maximize stiffness for a given mass constraint.

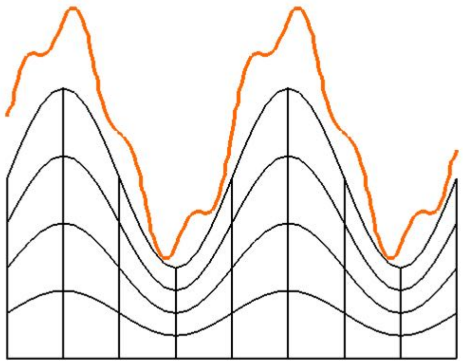
Boundary fitting

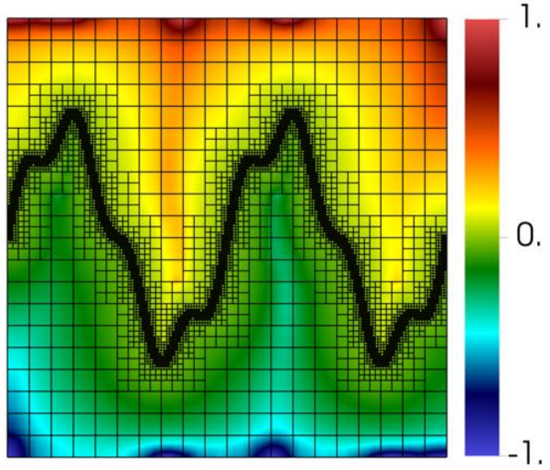
Using the mesh being optimized for representing $\sigma(\mathbf{x})$ results in a sub-optimal fit if

- The mesh does not have sufficient resolution around the zero level-set of $\sigma(\mathbf{x})$.
- The zero level-set of $\sigma(\mathbf{x})$ is outside the domain of the mesh.

We use a background mesh with AMR to ensure accuracy in $\sigma(\mathbf{x}_B)$ and its gradient

- Can also be used for tangential relaxation


Current mesh and target level set

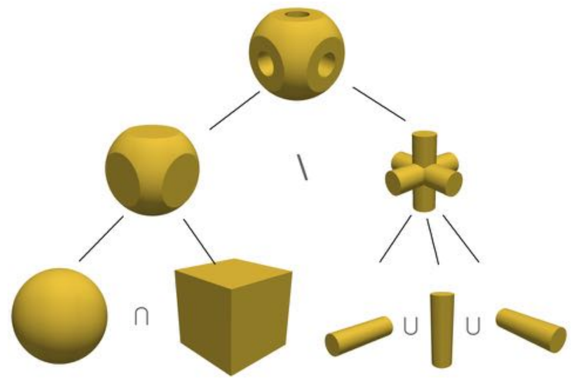


Level set on a background mesh

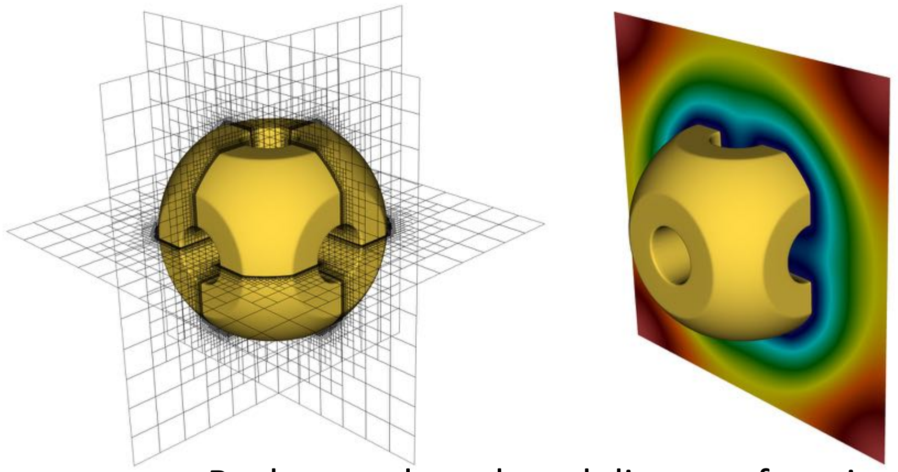
Level set is transferred from the background mesh to the current mesh with *gslib:*

$$\sigma(\mathbf{x}) = I(\mathbf{x}_B, \sigma(\mathbf{x}_B), \mathbf{x})$$

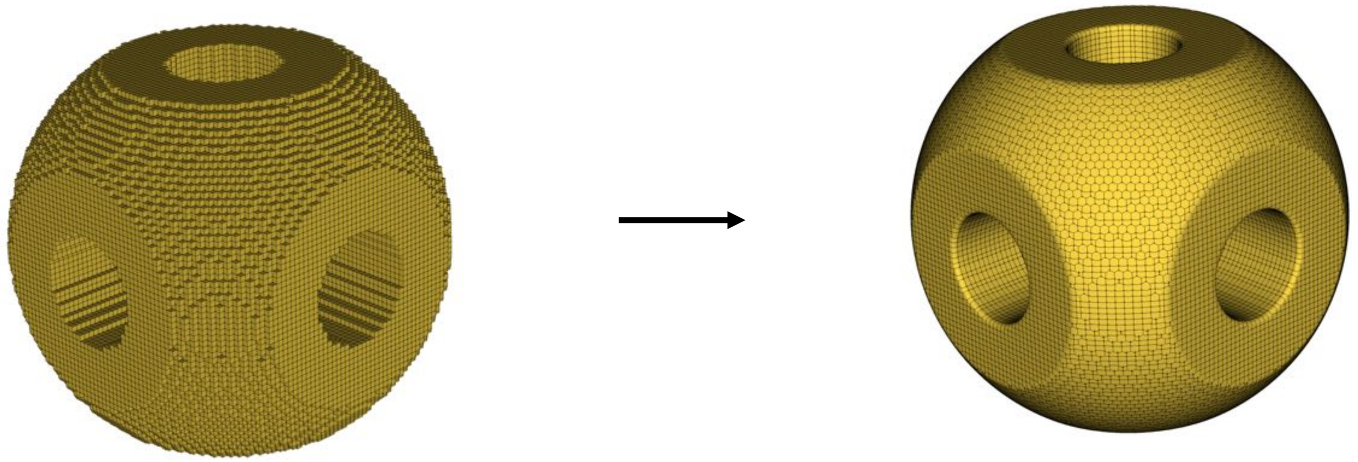
Boundary fitting for a complex 3D domain



CSG Tree for a curvilinear domain



Background mesh and distance function

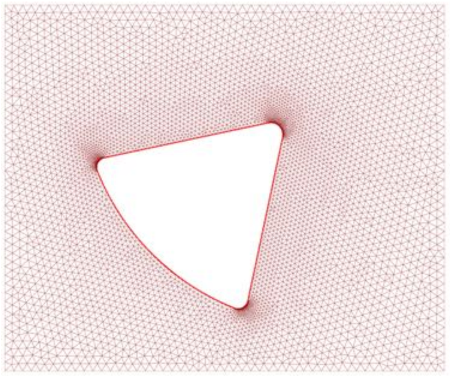


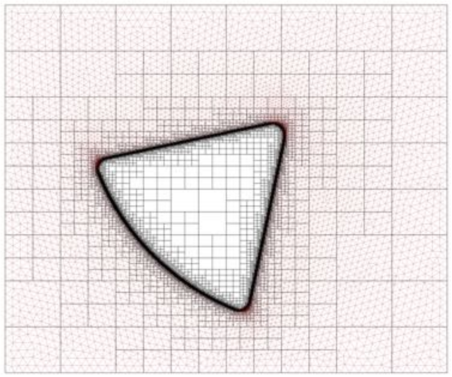
Uniform Cartesian (second-order) mesh trimmed and fit to the level set function.

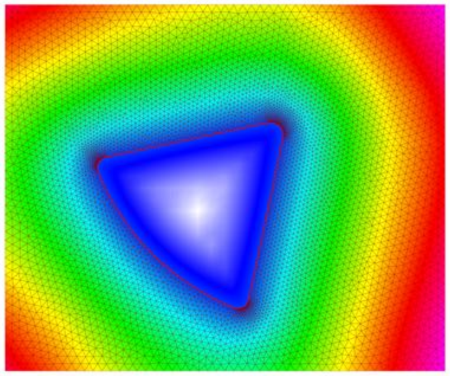
Mixed-order meshes using rp-adaptivity

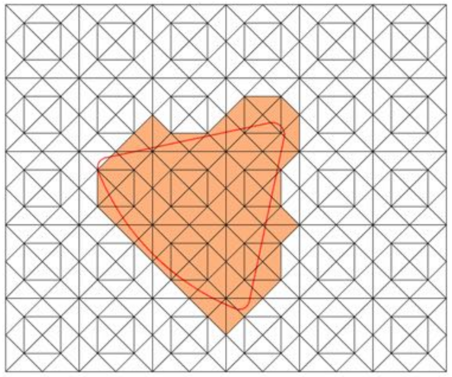
Use high-order elements in regions of low-curvature and lower-order elements elsewhere

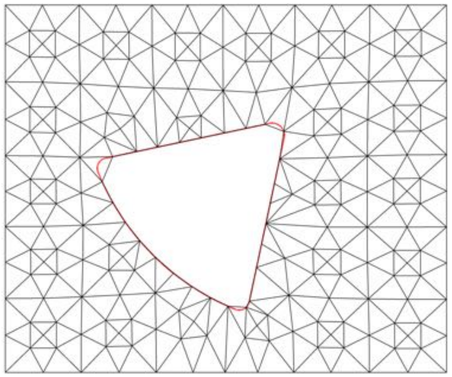
[Submitted to IMR 2024 in collaboration with Franck and Claire from CEA]

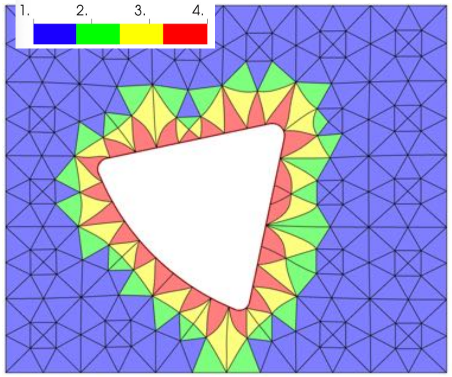

Dense low-order mesh


AMR background mesh

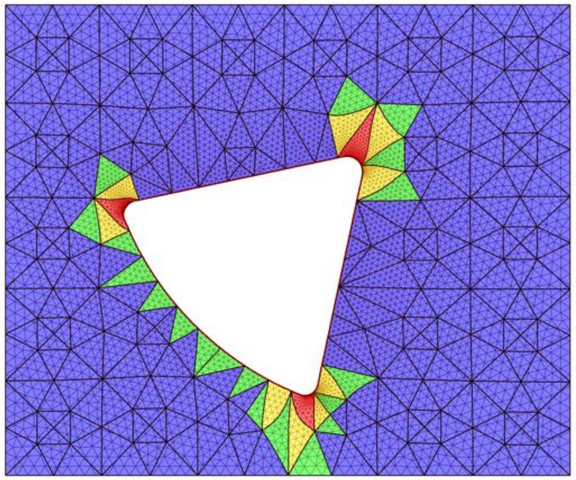

Extracted Level-Set

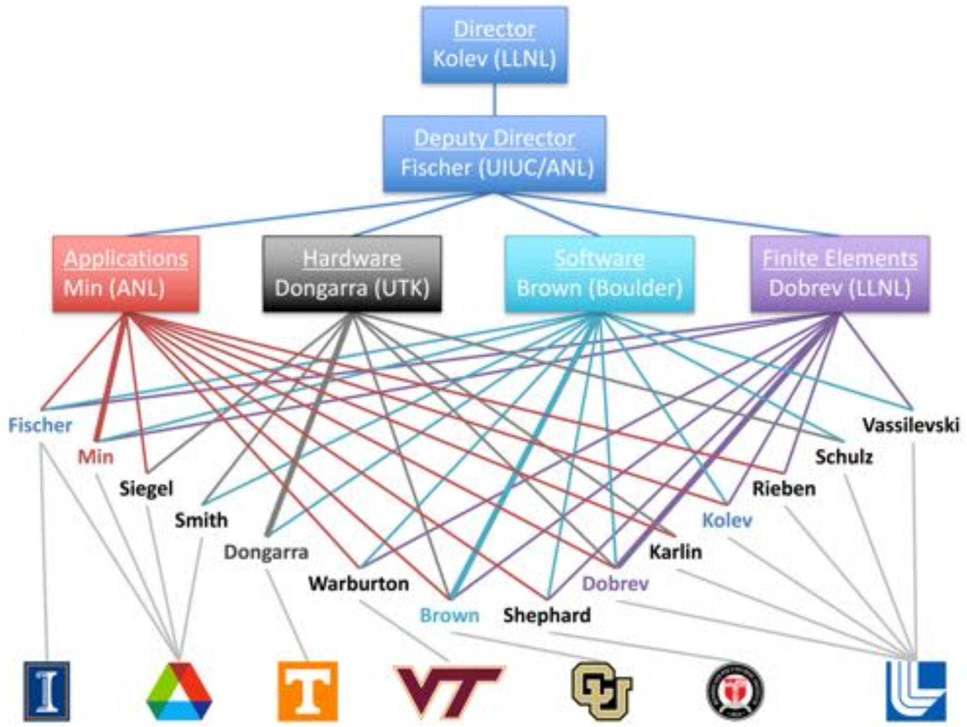

Coarse target mesh


Linear mesh aligned with the target surface.



Optimized mesh with high-order elements around interface.

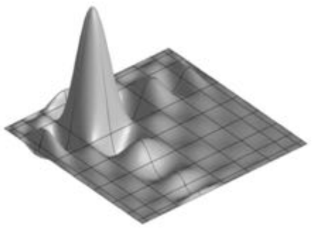

Final Coarse Mixed-Order Mesh compared with Dense Linear Mesh.

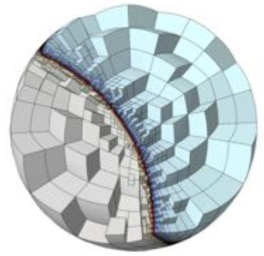


2 Labs, 5 Universities, 30+ researchers

- PDE-based simulations on **unstructured grids**
- **high-order** and **spectral** finite elements

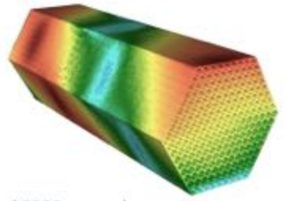
  ✓ *any order space on any order mesh* ✓ *curved meshes,*
  ✓ *unstructured AMR* ✓ *optimized low-order support*

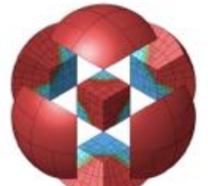

10th order basis function


non-conforming AMR, 2nd order mesh

- state-of-the art CEED **discretization libraries**

  ✓ *better exploit the hardware to deliver significant performance gain over conventional methods*
  ✓ *based on MFEM/Nek, low & high-level APIs*

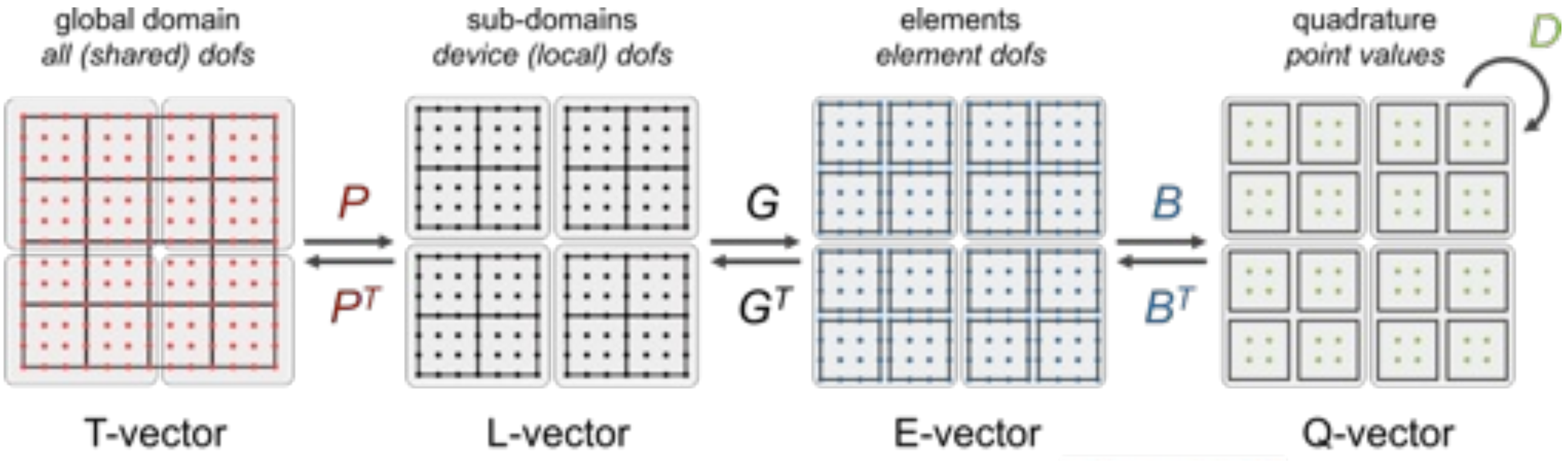

nek5000.mcs.anl.gov
High-performance spectral elements


mfem.org
Scalable high-order finite elements

Finite element operator decomposition

Finite element operator assembly/evaluation can be split into ***parallel***, ***mesh***, ***basis***, *and* *geometry/physics* *parts:*

$$\boxed{A = P^T G^T B^T D B G P}$$



✓ ***partial assembly*** = store only *D*, evaluate ***B*** (tensor-product structure)

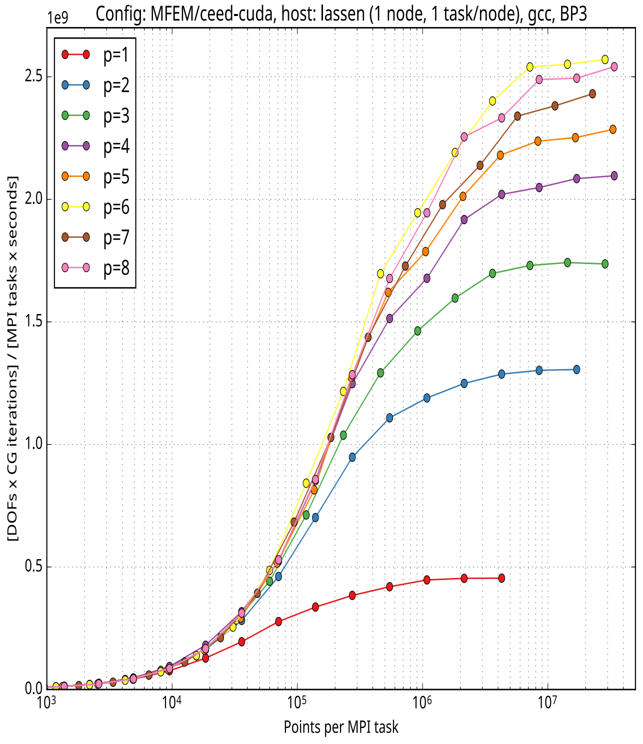
✓ better representation than ***A***: *optimal memory, near-optimal FLOPs*

✓ purely algebraic ✓ high-order *operator format* ✓ AD-friendly

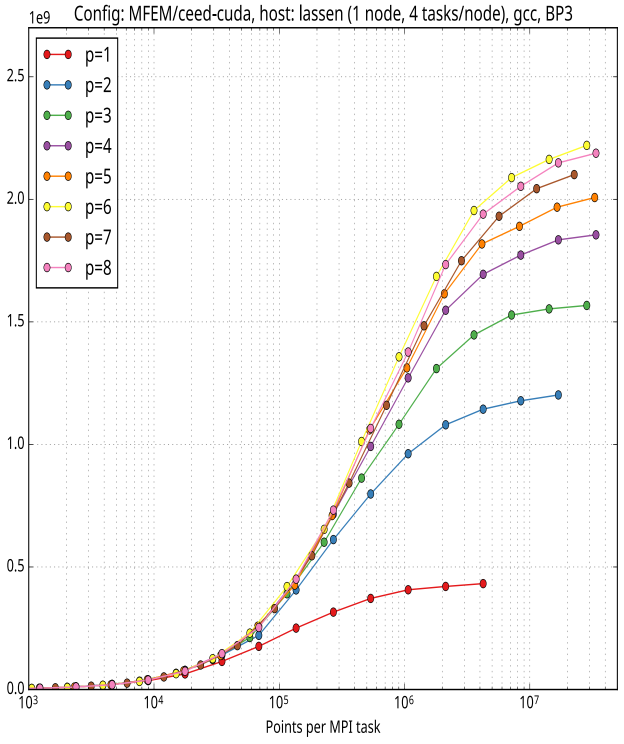
Lawrence Livermore National Laboratory

CASC

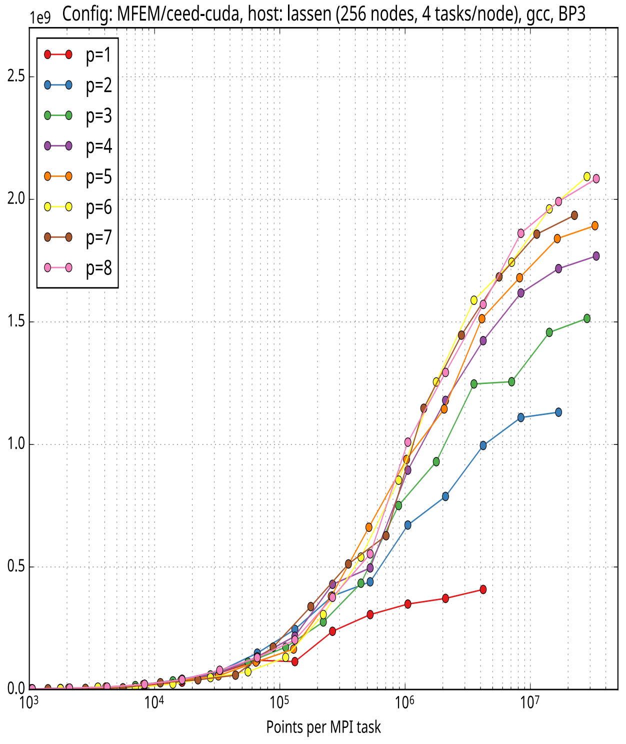
MFEM performance on multiple GPUs



1 GPU



4 GPUs



1024 GPUs

Single GPU performance: **2.6 GDOF/s**
Problem size: 10+ million

Best total performance: **2.1 TDOF/s**
Largest size: 34 billion

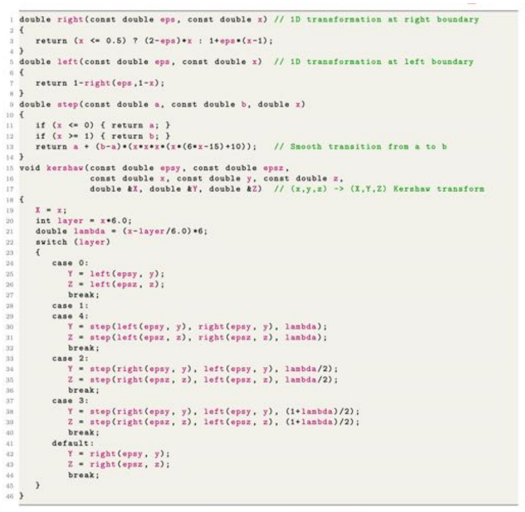
Optimized kernels for MPI buffer packing/unpacking on the GPU

Lawrence Livermore National Laboratory

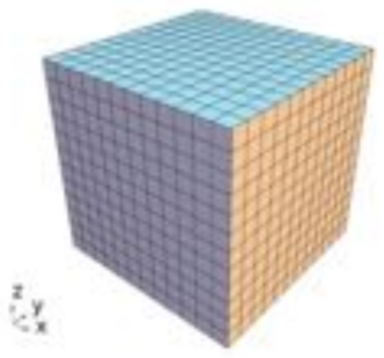
CASC

Kershaw benchmark

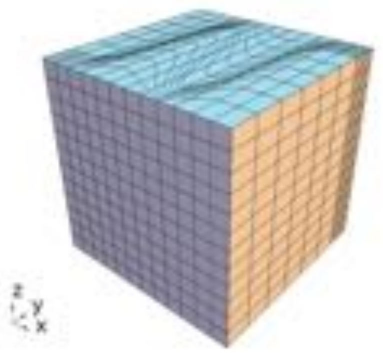
- Easy-to-setup benchmark for timing high-order mesh optimization.
- Two parameters $(\epsilon_y, \epsilon_z) \in (0,1]^2$ control the element deformation.
- In our tests, we use a $24 \times 24 \times 24$ mesh, 9 quadrature points in each direction in an element, and a shape metric $\mu_{303} = \frac{|T|^2}{3\tau^{2/3}} - 1$.



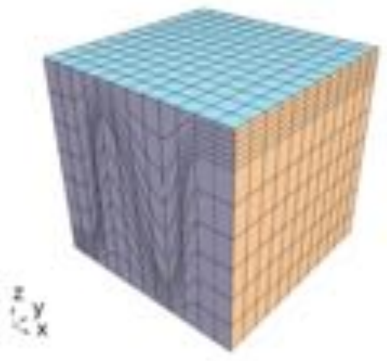
```
double right(const double eps, const double x) // 1D transformation at right boundary
{
   return (x <= 0.5) ? (2-eps)*x : 1+eps*(x-1);
}
double left(const double eps, const double x)  // 1D transformation at left boundary
{
   return 1-right(eps,1-x);
}
double step(const double a, const double b, double x)
{
   if (x <= 0) { return a; }
   if (x >= 1) { return b; }
   return a + (b-a)*(x*x*x*(x*(6*x-15)+10));   // Smooth transition from a to b
}
void kershaw(const double epsy, const double epsz,
             const double x, const double y, const double z,
             double &X, double &Y, double &Z)  // (x,y,z) -> (X,Y,Z) Kershaw transform
{
   X = x;
   int layer = x*6.0;
   double lambda = (x-layer/6.0)*6;
   switch (layer)
   {
      case 0:
         Y = left(epsy, y);
         Z = left(epsz, z);
         break;
      case 1:
      case 4:
         Y = step(left(epsy, y), right(epsy, y), lambda);
         Z = step(left(epsz, z), right(epsz, z), lambda);
         break;
      case 2:
         Y = step(right(epsy, y), left(epsy, y), lambda/2);
         Z = step(right(epsz, z), left(epsz, z), lambda/2);
         break;
      case 3:
         Y = step(right(epsy, y), left(epsy, y), (1+lambda)/2);
         Z = step(right(epsz, z), left(epsz, z), (1+lambda)/2);
         break;
      default:
         Y = right(epsy, y);
         Z = right(epsz, z);
         break;
   }
}
```



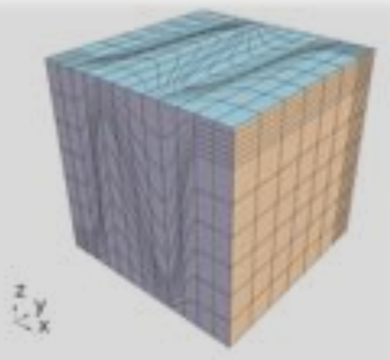
(a) $\varepsilon_y = \varepsilon_z = 1$



(b) $\varepsilon_y = 0.3,\ \varepsilon_z = 1$



(c) $\varepsilon_y = 1,\ \varepsilon_z = 0.3$

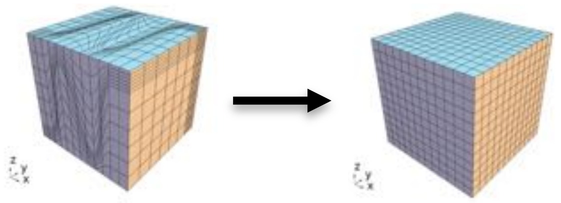


(d) $\varepsilon_y = \varepsilon_z = 0.3$

"Accelerating high-order mesh optimization using finite element partial assembly on GPUs", JCP, 2023

Kershaw timing and throughput results

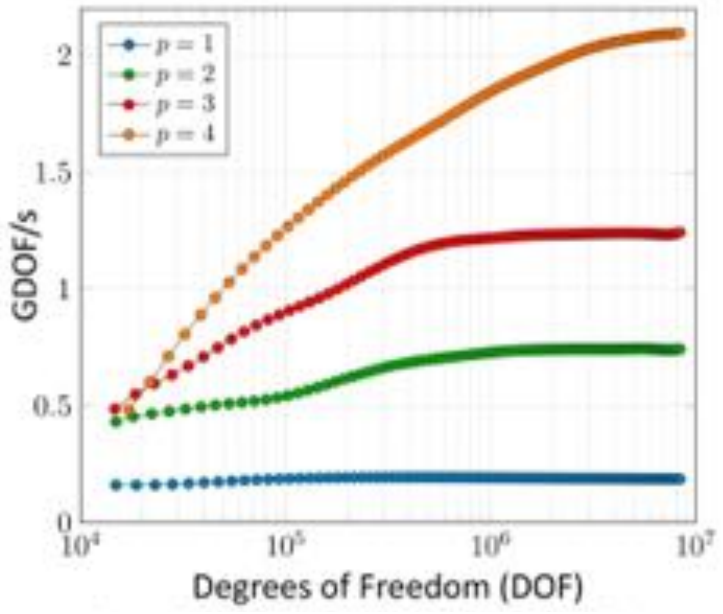
- Timing comparison on Lassen, a Livermore Computing supercomputer, for full- and partial-assembly on CPU vs partial-assembly on GPU.
  - CPU - 36 cores.
  - GPU - 4 CPU cores with 1 GPU per core.



	Time to solution (sec)			
	$p=1$	$p=2$	$p=3$	$p=4$
$\mathrm{CPU}^{\mathrm{FA}}$	2.9	31.1	489.6	2868.8
$\mathrm{CPU}^{\mathrm{PA}}$	18.0	41.0	128.5	298.0
$\mathrm{GPU}^{\mathrm{PA}}$	0.4	0.9	3.9	8.5
	Speedup ($\mathrm{GPU}^{\mathrm{PA}}$ vs $\mathrm{CPU}^{\mathrm{PA}}$)			
	42×	**43×**	**32×**	**35×**

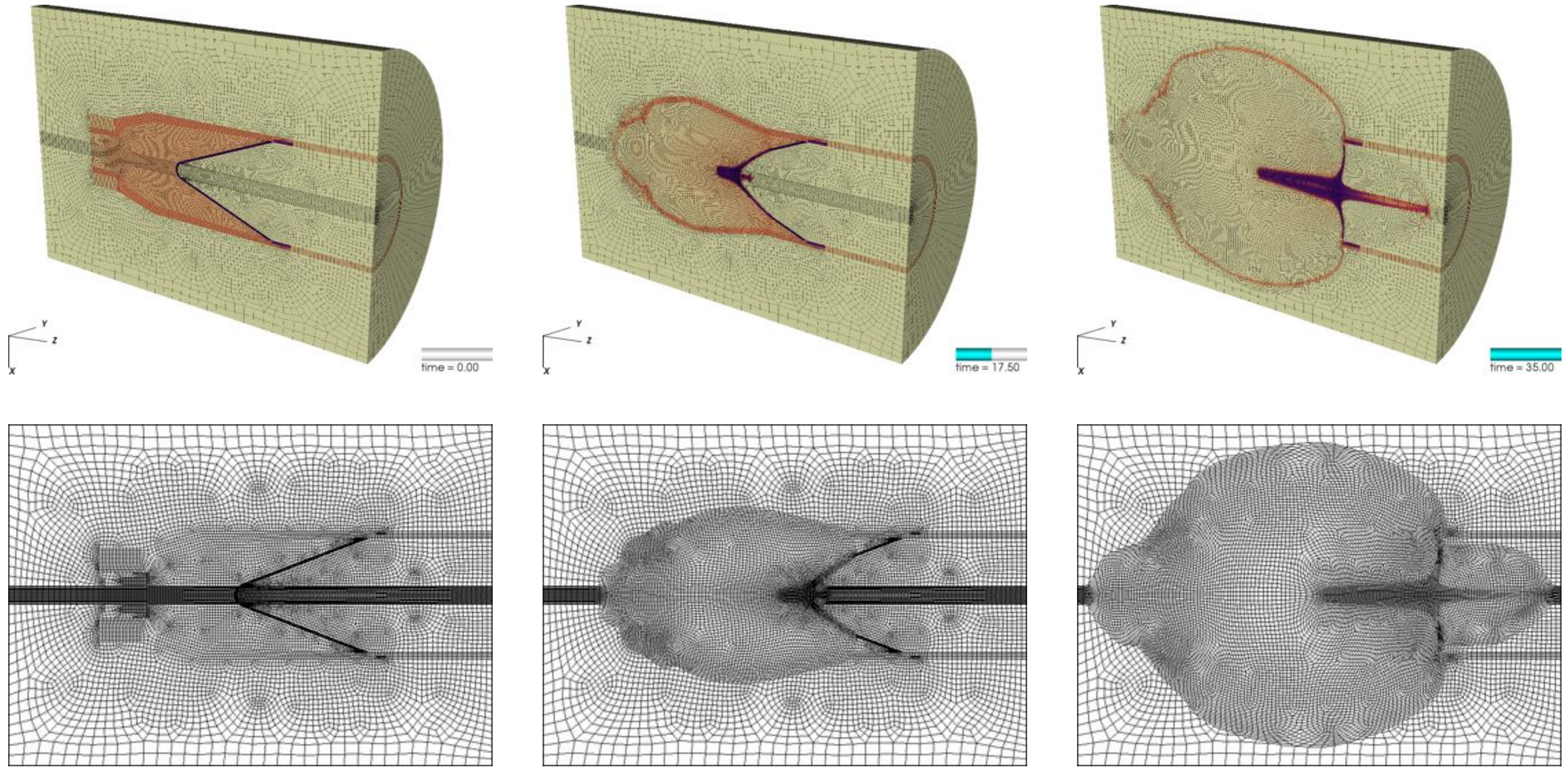
$\mathcal{O}(30\times)$ speed-up on GPUs versus CPUs

PA beneficial on CPUs for higher p



Throughput on NVIDIA V100

GPU acceleration for multi-material high-order ALE with solution-driven TMOP adaptivity



Density (top) and 2nd order mesh (bottom) for the ALE shaped charge GPU simulation.

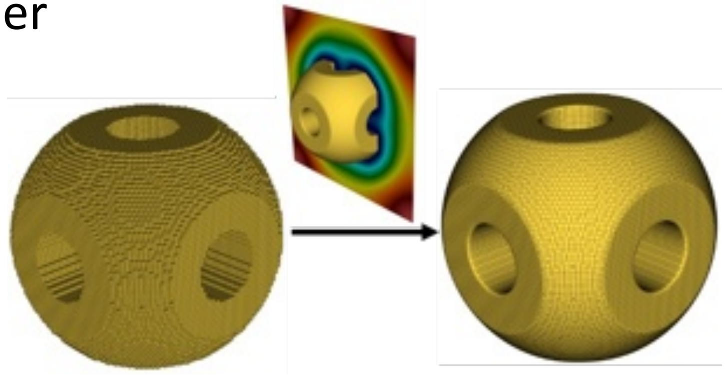
20x speed-up for TMOP step in the solver!

Lawrence Livermore National Laboratory

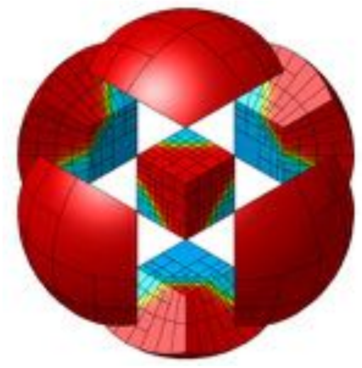
CASC

Summary

- Mesh adaptivity for HPC simulations on high-order meshes via TMOP
  - TMOP for high-order meshes
  - Adaptive surface fitting
  - GPU acceleration



- All presented methods are available in MFEM
  - MFEM contains **12** 2D metrics, **7** 3D metrics, all metric derivatives, **6** target constructions
  - **Mesh Optimizer** miniapp provides choice of target construction, quality metric, adaptivity fields and parameters, GLVis visualization.



mfem.org

- Papers and more information: llnl.gov/casc/projects/ethos

CASC

Lawrence Livermore National Laboratory

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.
LLNL-PRES-XXXXXX

Modular Finite Element Methods (MFEM)

Flexible discretizations on unstructured grids

- Triangular, quadrilateral, tetrahedral, hexahedral, prism; volume, surface and topologically periodic meshes
- Bilinear/linear forms for: Galerkin methods, DG, HDG, DPG, IGA, …
- Local conforming and non-conforming AMR, mesh optimization
- Hybridization and static condensation

High-order methods and scalability

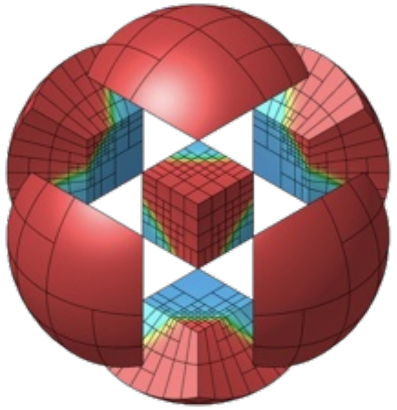
- Arbitrary-order H1, H(curl), H(div)- and L2 elements
- Arbitrary order curvilinear meshes
- MPI scalable to millions of cores + GPU accelerated
- Enables development from laptops to exascale machines.

Solvers and preconditioners

- Integrated with: HYPRE, SUNDIALS, PETSc, SLEPc, SUPERLU, VisIt, …
- AMG solvers for full de Rham complex on CPU+GPU, geometric MG
- Time integrators: SUNDIALS, PETSc, built-in RK, SDIRK, ...

Open-source software

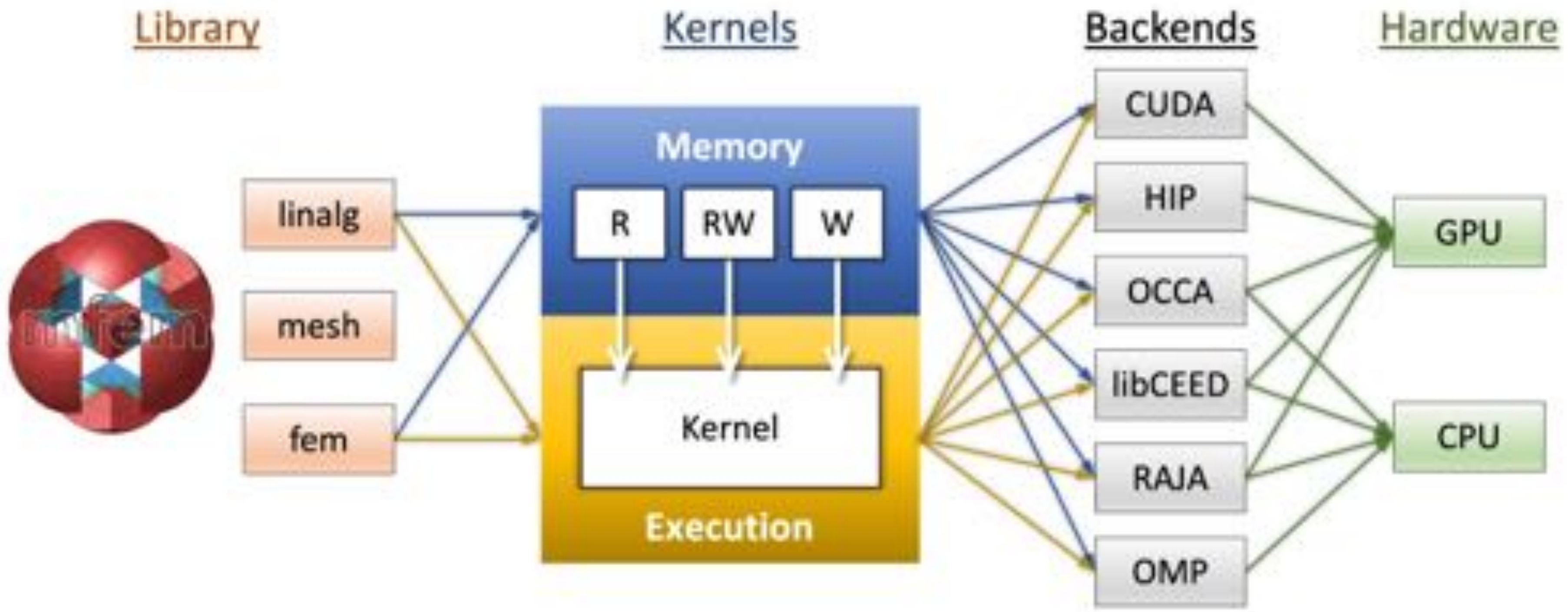
- Open-source (GitHub) with 114 *contributors*, 50 *clones/day*
- Part of FASTMath, ECP/CEED, xSDK, OpenHPC, E4S, …
- 75+ example codes & miniapps: mfem.org/examples



mfem.org
(v4.6, Sep/2023)

Device support in MFEM

MFEM support GPU acceleration in many linear algebra and finite element operations



- Several MFEM examples + miniapps have been ported with small changes
- Many kernels have a single source for CUDA, RAJA and OpenMP backends
- Backends are runtime selectable, can be mixed
- Recent improvements in CUDA, HIP, RAJA, SYCL, …