MFEM Capabilities for High-Order Mesh Optimization

MFEM Community Workshop

CASC
Center for Applied Scientific Computing

October 20, 2021

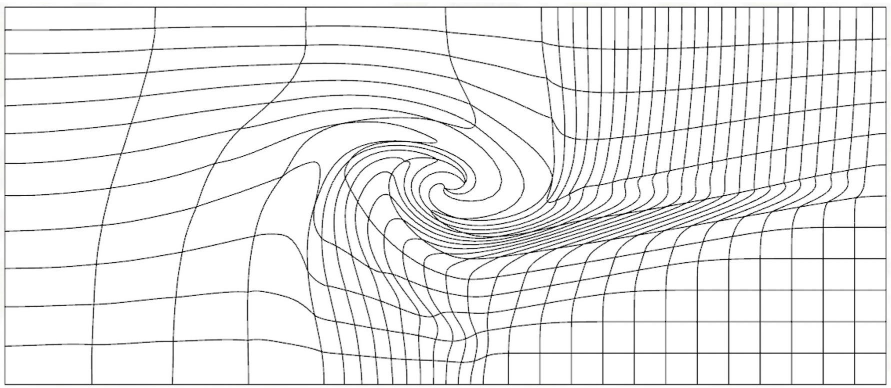
V. Tomov on behalf of the MFEM team

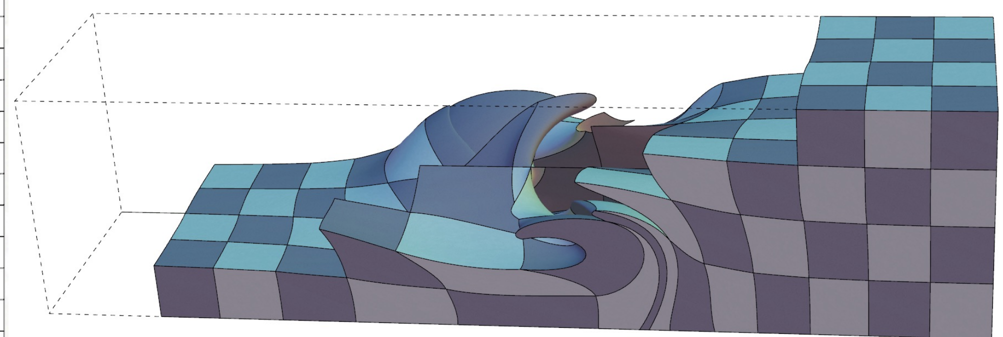
LLNL-PRES-828130
This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC

Lawrence Livermore National Laboratory

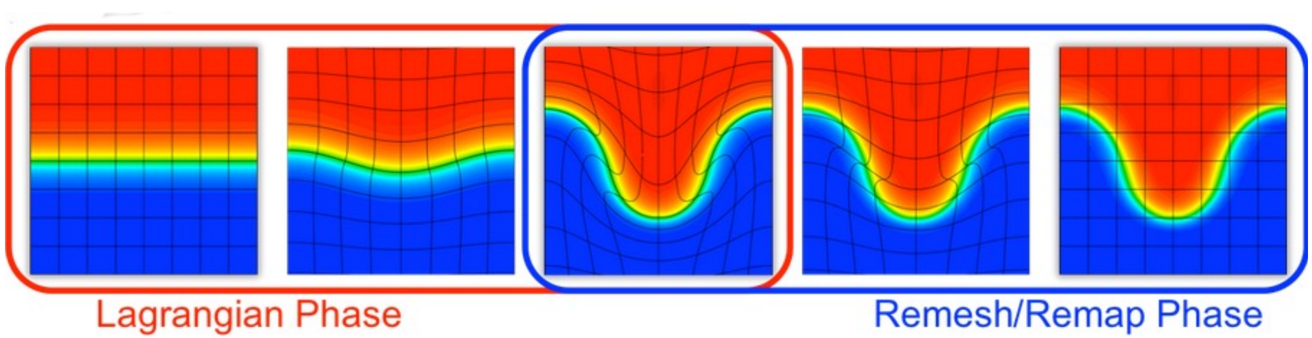
Initially motivated by an ALE application

- Mesh deformation in Lagrangian simulations.





- ALE procedure -->

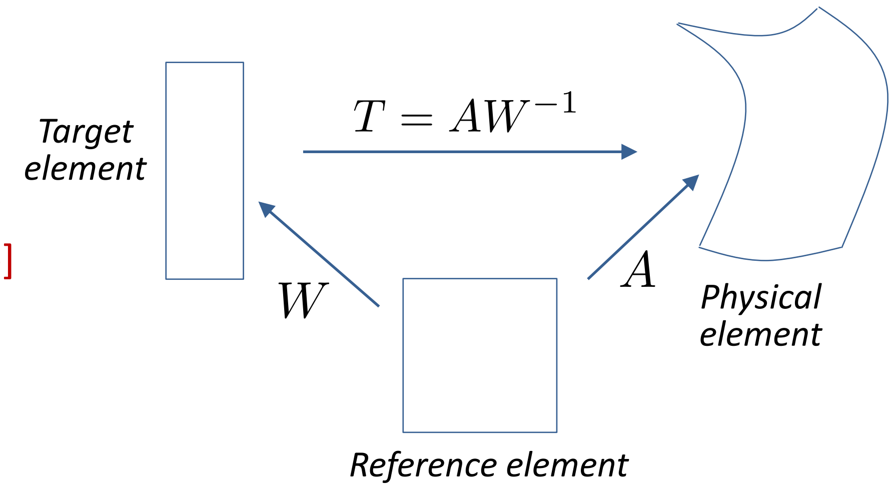


Main Directions

- Variational formulation based on FE; minimize geometry-based operations.
  - Use MFEM's FE and MPI infrastructure.
  - Use MFEM's GPU / device infrastructure.
- Curved meshes with arbitrary order elements.
  - Generality w.r.t. order, dimension, element type.
- Node movement and preservation of topology (not a mesh generator).
- Adaptivity to discrete fields / preservation of discrete features.
- Untangling, support for AMR meshes, HR-adaptivity.
- Surface fitting, tangential relaxation (in progress).

Approach – Target-Matrix Optimization Paradigm

- Target construction - the user defines ideal target elements.
  - W includes adaptivity information.
  $W =$ [volume] [orientation] [skew] [aspect ratio]



- The Jacobian T is used to define the local mesh quality measure $\mu(T)$.

- We minimize a global integral over the target elements:

$$F(x) := \sum_{E \in \mathcal{M}} \int_{E_t} \mu(T(x_t)) dx_t = \sum_{E \in \mathcal{M}} \sum_{x_q \in E_t} w_q \det(W(\bar{x}_q)) \, \mu(T(x_q))$$

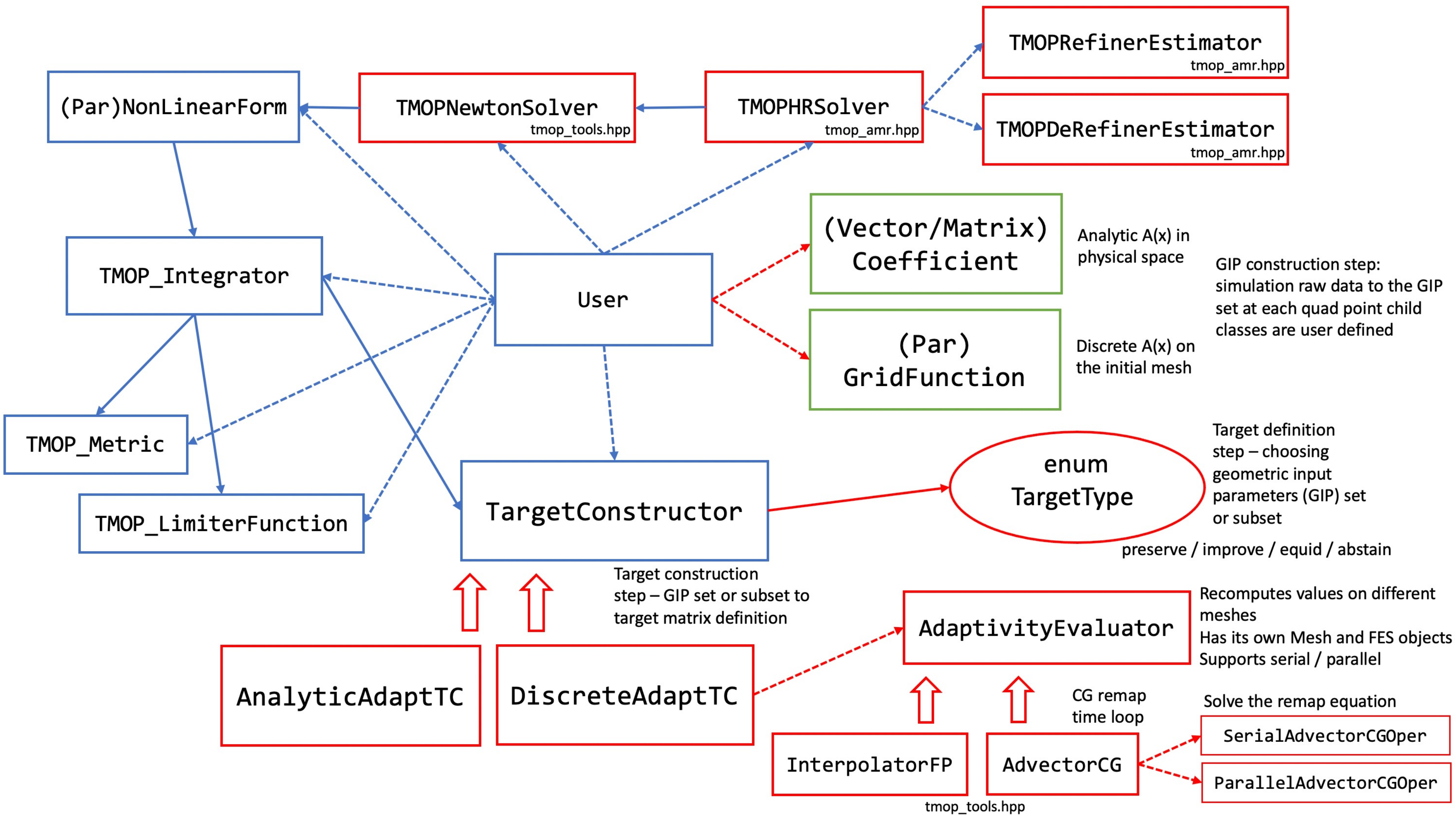
- Default option: Newton's method (+ line search) to solve $\partial F(x) \,/\, \partial \boldsymbol{x} = 0$.

Implementation overview

$$F(x) := \sum_{E \in \mathcal{M}} \int_{E_t} \mu(T(x_t)) dx_t = \sum_{E \in \mathcal{M}} \sum_{x_q \in E_t} w_q \det(W(\bar{x}_q))\, \mu(T(x_q))$$

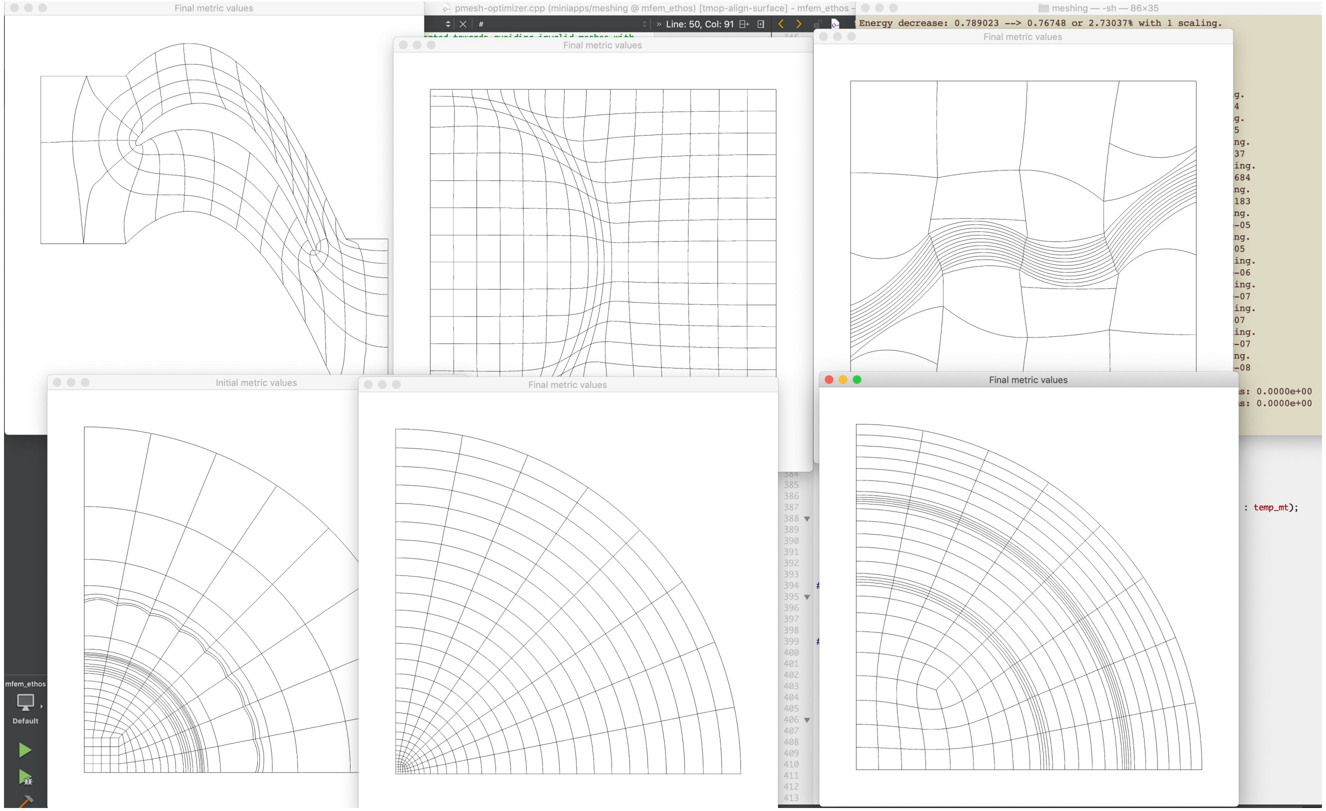
- class TargetConstructor – computation of W based on user inputs.
- class TMOP_Metric – computation of $\mu(T)$ and $\partial\mu/\partial T$ derivatives.
- class TMOP_Integrator – assembly of all TMOP integrals.
- class TMOP_NewtonSolver – solves the nonlinear problem $\partial F(x)\, /\, \partial \boldsymbol{x} = 0$.

Class diagram of the TMOP module



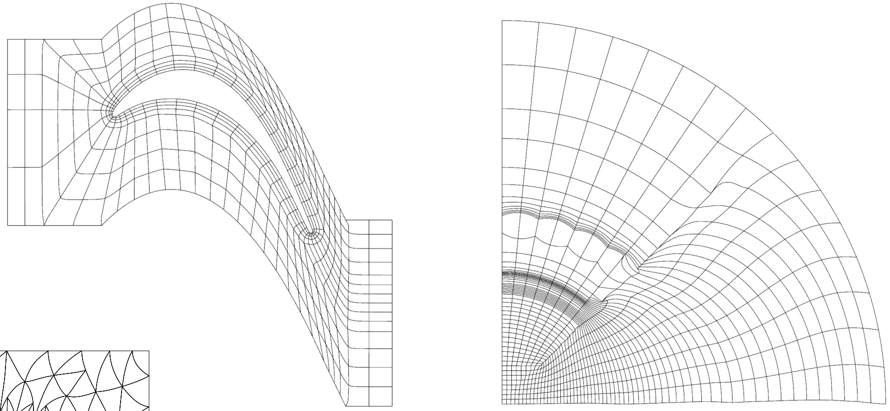
Demo of the mesh-optimizer miniapp

- Mesh optimization miniapp: /miniapps/meshing/(p)mesh_optimizer.cpp



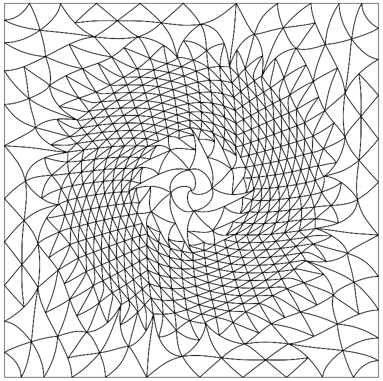
Advanced features

- Limiting of node displacements.
  - space-dependent flexibility



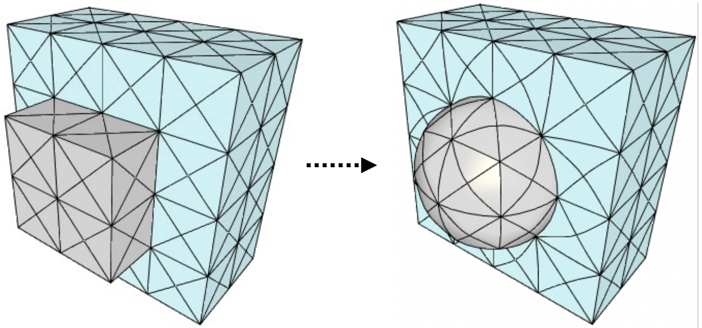
$$F(x) = F_\mu + \sum_E \int_{E_t} \frac{(x - x_0)^2}{d^2}$$

- hr-adaptivity.



- Interface fitting / tangential relaxation.

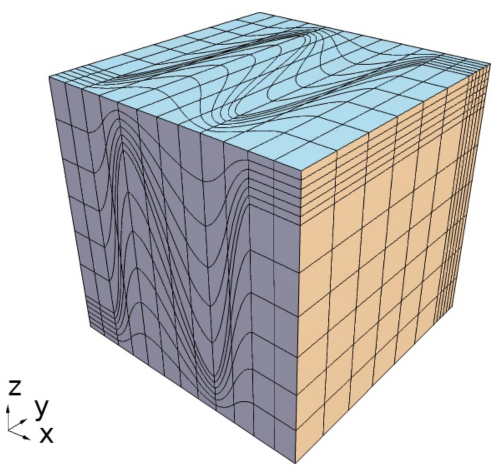
$$F(x) = F_\mu + w_\sigma \int_\Omega \bar{\sigma}(x)^2$$

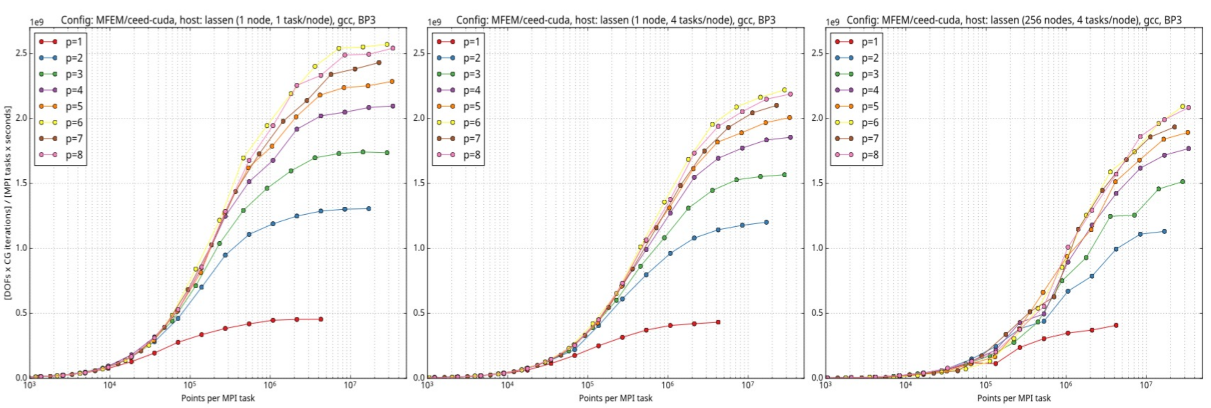


Performance – partial assembly and GPUs

- The major functionality supports partial assembly and GPU execution.
  - tensor contractions to evaluate Jacobians and all integrals.
  - matrix-free action of the Hessian.

- Performance benchmark – optimize the Kershaw mesh.





1 GPU ***4 GPUs*** ***1024 GPUs***

Optimized kernels for MPI buffer packing/unpacking on the GPU
GPU-aware MPI ready

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

CASC

Center for Applied Scientific Computing

Lawrence Livermore National Laboratory