Simulation-Driven Optimization of High-Order Meshes by the Target-Matrix Optimization Paradigm

SIAM Conference on Computational Science and Engineering, Spokane, WA

CASC
Center for Applied Scientific Computing

February 26, 2019

Vladimir Tomov
V. Dobrev, P. Knupp, Tz. Kolev, K. Mittal and R. Rieben

LLNL-PRES-768119
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 mesh representation

Positions are represented by a high-order finite element function

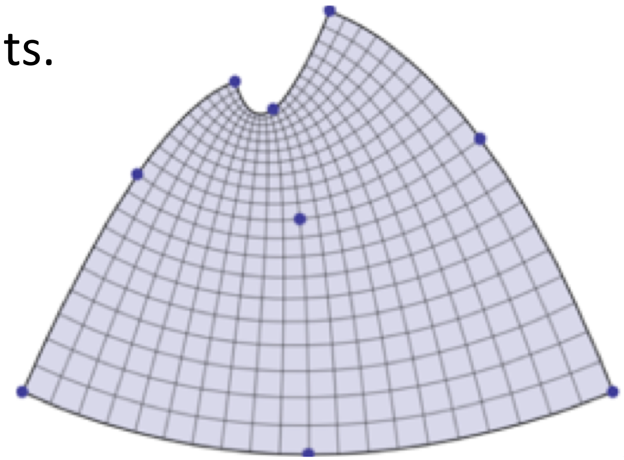
- Mesh positions are discretized by a position vector and a FE basis:

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

- $\{\bar{w}_i\}_1^{N_E}$ spans Q_k for quadrilateral / hexahedral elements.
- $\{\bar{w}_i\}_1^{N_E}$ spans P_k for triangular / tetrahedral elements.
- Reference -> physical Jacobian is given by the basis functions' gradients:

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

- To optimize the curved mesh, we move its nodes by changing $\boldsymbol{x}$. Topology is preserved.



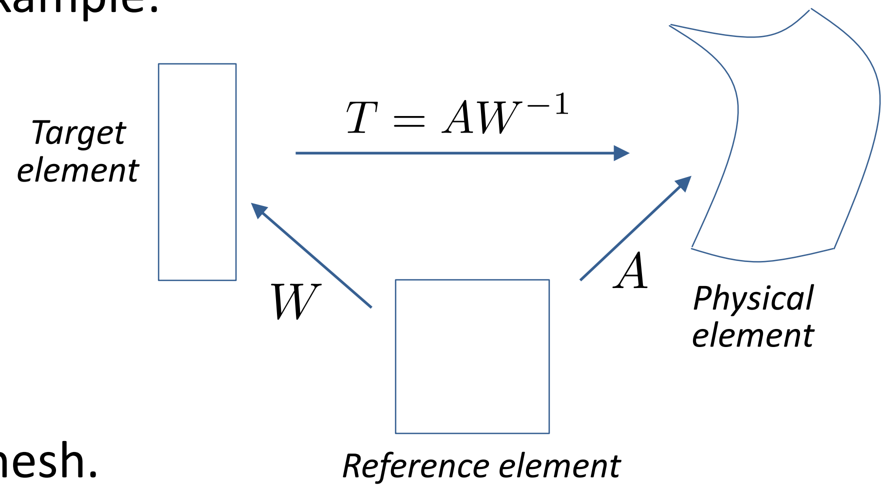
Example of a single Q_2 element

Target-Matrix Optimization Paradigm (TMOP)

TMOP is extended to high-order curved meshes

- Target construction: the user defines ideal target elements by specifying the target Jacobians W. 2D example:

$$W = \begin{pmatrix} W_{11} & 0 \\ 0 & W_{22} \end{pmatrix}, W_{11} > 0, W_{22} > 0$$



- TMOP combines the information contained in the Jacobians A and W.

- Taken at every quadrature point in the mesh.

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

- Combinations of W and $\mu(T)$ control various properties of the physical element.

P. Knupp, “Introducing the target-matrix paradigm for mesh optimization by node movement”, Engineering with Computers, 28(4):419-429, 2012.

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 combinations.
  SH+SZ, SH+AL, SZ+AL, SH+SZ+AL.
  $$\mu_7(T) = |T - T^{-t}|^2 \qquad \mu_{14}(T) = |T - I|^2$$
- Explicit combinations.
  $$\mu(T) = \alpha\mu_i(T) + (1-\alpha)\mu_j(T)$$

P. Knupp, “Algebraic mesh quality metrics”, SIAM J. Sci. Comp., 23(1):193-218, 2001.

Objective functions and nonlinear optimization

- 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 $\partial F(x) \,/\, \partial \boldsymbol{x} = 0$.
  - Efficient parallel implementation in contrast to derivative-free methods.

- Requires calculation of $\partial \mu(T) \,/\, \partial T$ and $\partial^2 \mu(T) \,/\, \partial T^2$ for all metrics.

- Mesh displacements can be limited by adjusting F:

$$F(x) := \cdots + \sum_{E \in \mathcal{M}} \int_{E_t} \frac{(x - x_0)^2}{d^2}$$

Problem-dependent physical distance specified by the user

- Other capabilities: stressing certain regions, region-dependent optimization by forming weighted combinations of integrals, untangling …

Combinations of metrics and limiting terms

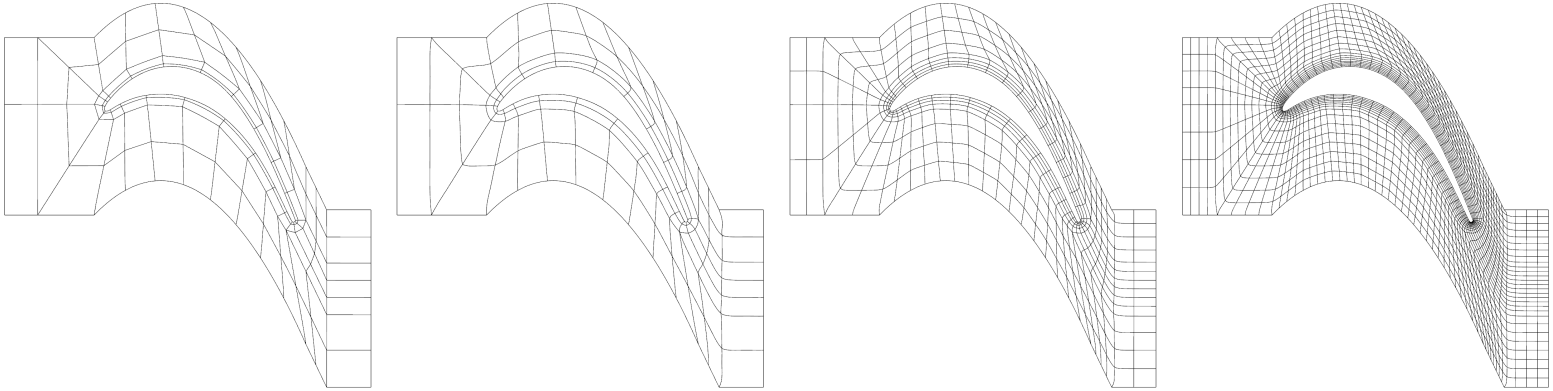
All terms are normalized relative to the unit value

- Users expect similar displacement under mesh refinement / change of units.
- Users expect reasonable, O(1) adjustable constants for each problem.

$$F(x) = \alpha \frac{1}{n} \frac{\sum_{E(x)} \int_{E_t} \mu_{i_1}(T)}{\sum_{E(x_0)} \int_{E_t} \mu_{i_1}(T_0)} + \dots \beta \frac{1}{n} \frac{\sum_{E(x)} \int_{E_t} \mu_{i_n}(T)}{\sum_{E(x_0)} \int_{E_t} \mu_{i_n}(T_0)} + \varepsilon \frac{\sum_E \int_{E_t} \frac{(x-x_0)^2}{d^2}}{\sum_E \int_{E_t} 1}$$

Behavior under refinement for fixed d 4th order mesh, Shape optimization

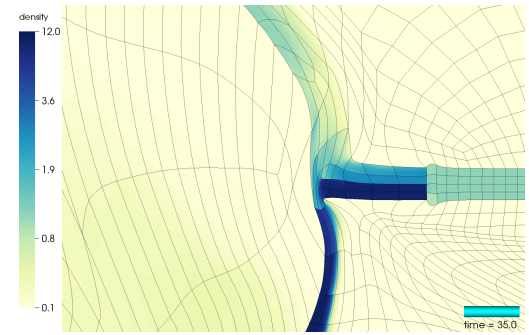
Refs	Final F	μ part	Limiting part	Max displacement
0	0.7548	0.6915	0.0632	0.0216
1	0.7542	0.6913	0.0629	0.0217
2	0.7541	0.6912	0.0628	0.0217



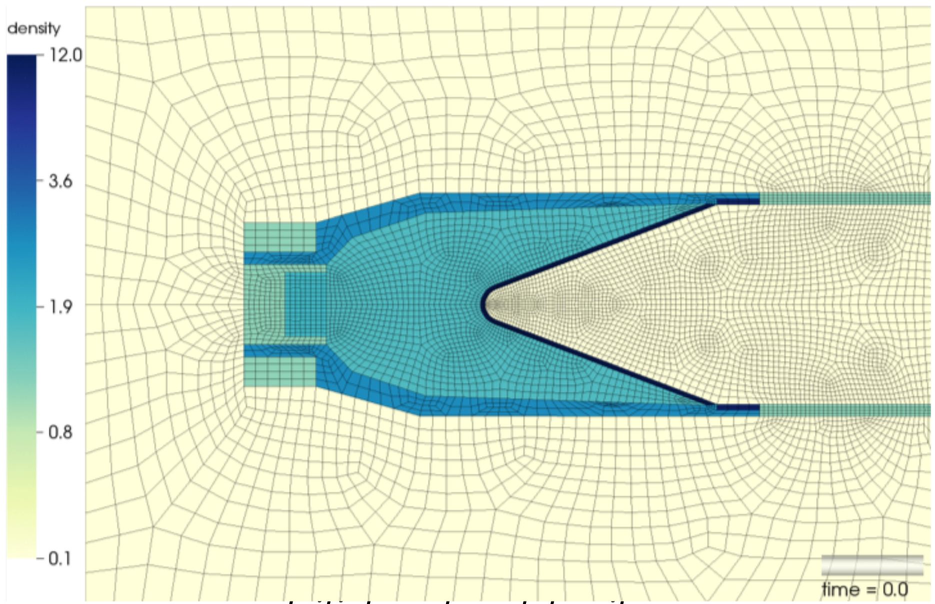
Feature preservation and local optimization

- Local optimization by using space-dependent d. No mesh motion for $d \to 0$.
- Applicable when the starting mesh has certain desirable features.

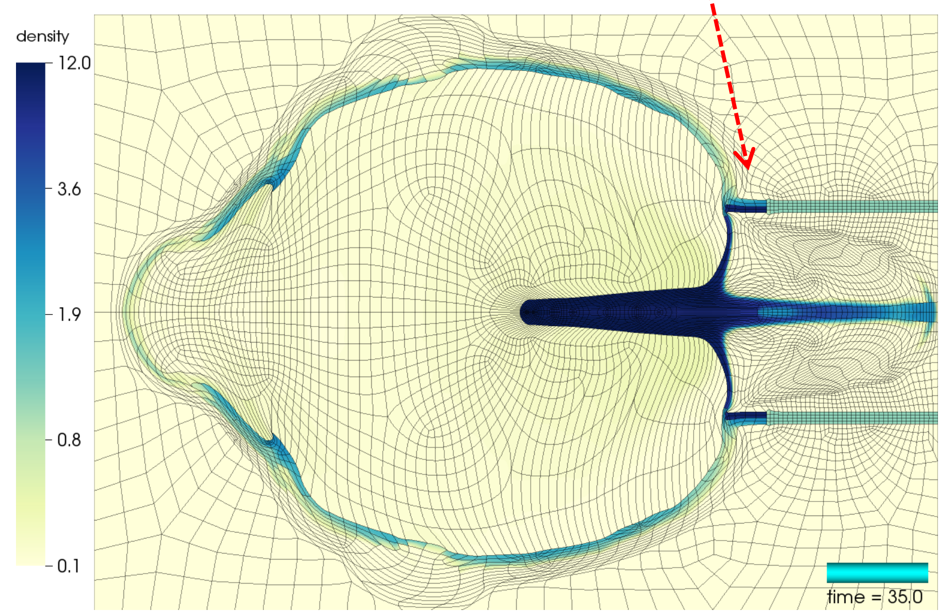
$$\sum_E \int_{E_t} \frac{(x - x_0)^2}{d^2}$$



Shaped charge simulation on an unstructured NURBS mesh, Shape+Size optimization



Initial mesh and density



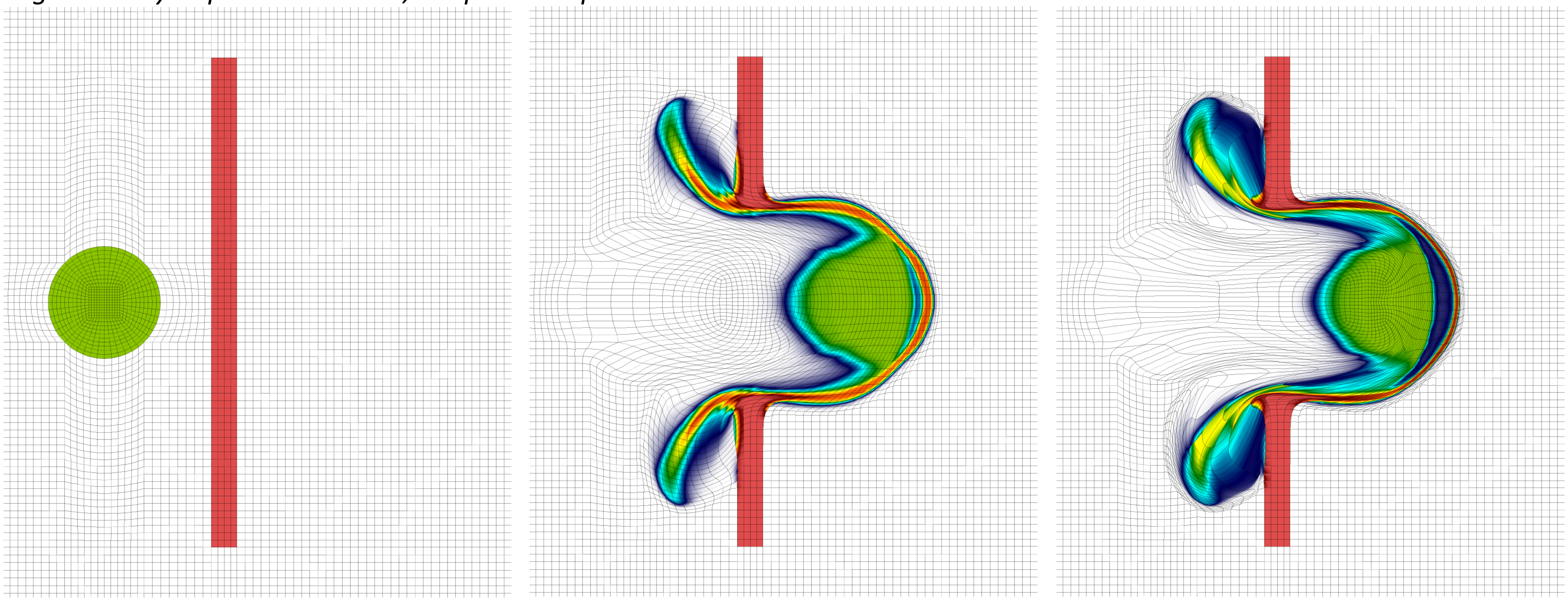
Mesh and density at final time

Feature preservation in moving mesh applications

$$\sum_E \int_{E_t} \frac{(x - x_0)^2}{d^2}$$

- Optimization can be adapted to the problem dynamics. Example: $d \sim \alpha$[mesh displacement] in ALE simulations.
- Allows tradeoffs between mesh quality feature preservation.

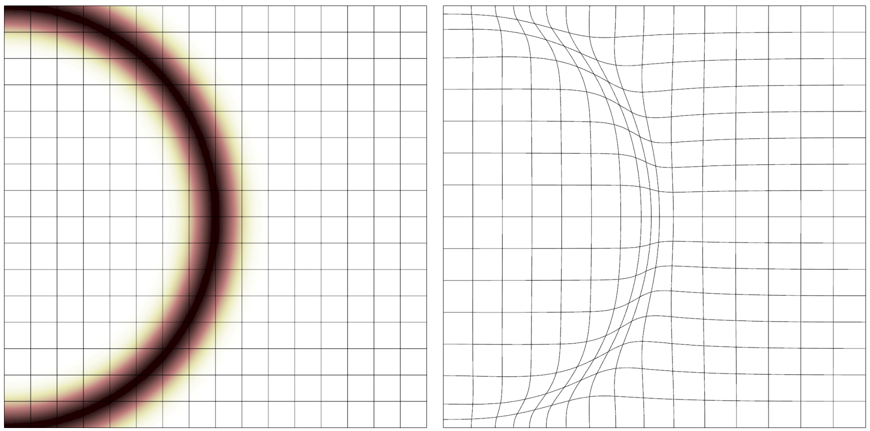
High-velocity impact simulation, Shape+Size optimization



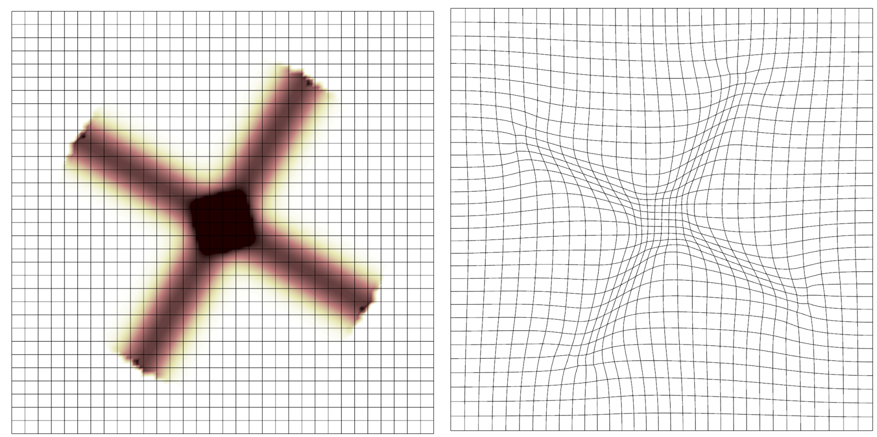
Initial condition | *Higher d – better mesh / more diffusion* | *Lower d – worse mesh / less diffusion*

Simulation-driven high-order mesh optimization
Optimization objectives are specified by simulation fields

- Goal: adapt the mesh, by node movement, to a given feature (r-adaptivity). Examples: material interfaces, known error estimates, regions of interest.



Adaptivity function on the initial mesh *Adapted mesh (size adaptation)*



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

P. Greene, S. Schofield, R. Nourgaliev, "Dynamic mesh adaptation for front evolution using DG based condition number relaxation", Journal of Computational Physics (335):664-687, 2017.
P. Váchal, P.-H. Maire, "Discretizations for weighted condition number smoothing on general unstructured meshes", Computers & Fluids 46(1):479-485, 2011.

Adaptivity process

Adaptivity requires additional user input and three extra steps

1. Choose the optimization goal (user).
   Jacobians are determined by a set of geometric parameters. In 2D:

   volume = ζ, orientation angle = θ, skew angle = ϕ, aspect ratio = ρ.

   Preserve / Improve / Equidistribute / Abstain option for each (PIE-A decision).

2. Define the geometric parameters (user).
   Transform data available in the simulation, e.g., material positions.

3. Construct targets (software).
   Form the final target Jacobians given the geometric parameter values. In 2D:

$$X = \sqrt{\zeta} \begin{pmatrix} \frac{1}{\sqrt{\rho}} \cos\theta & \sqrt{\rho}\cos(\theta+\phi) \\ \frac{1}{\sqrt{\rho}} \sin\theta & \sqrt{\rho}\sin(\theta+\phi) \end{pmatrix}$$

4. Choose appropriate mesh quality metric (user or software).

5. Optimize the final objective function (software).

Adaptivity field values on intermediate meshes

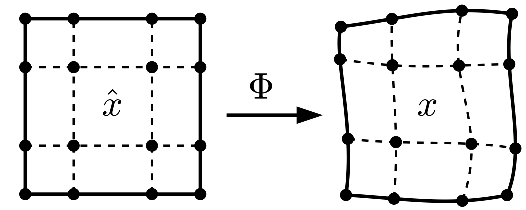
The values of $\eta_0(x_0)$ are transferred on different meshes

- Iterative solvers use a series of intermediate meshes to reach convergence.

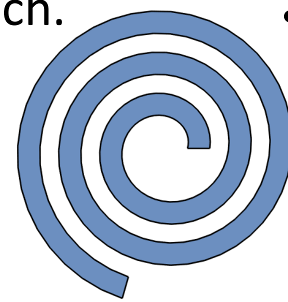
- Method 1: physical -> logical space interpolation.
  - Iterate over a set of candidate elements.
  - Invert the reference -> physical map for each.

$$\bar{x_0}^{n+1} = \bar{x_0}^n + A^{-1}(\bar{x_0}^n)\left[x - \Phi_{E_0}(\bar{x_0}^n)\right]$$

  - Challenging parallel implementation.



3rd order transformation



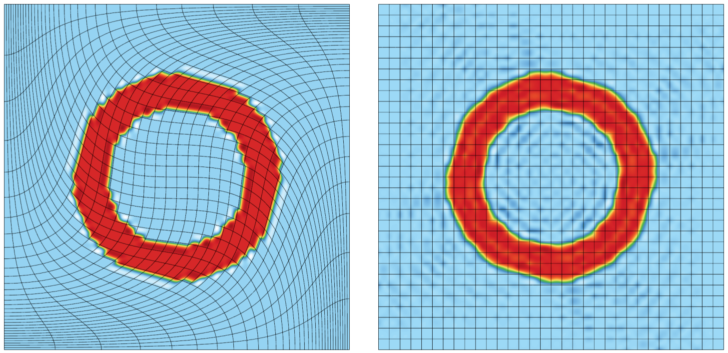
20th order element. MFEM's FindPoints routine succeeds on 90% of the test points.

- Method 2: advection remap.
  - Define pseudo-time τ and mesh velocity u.

$$\tau \in [0,1], \quad u = x - x_0$$

$$\frac{d\eta}{d\tau} = u \cdot \nabla \eta, \quad \eta(x_0, 0) = \eta_0(x_0)$$

  - CG advection, no monotonicity treatment.



CG remap of a 3rd order field on 3rd order mesh

Differentiation of adaptive target matrices

Major difference compared to geometry-based optimization

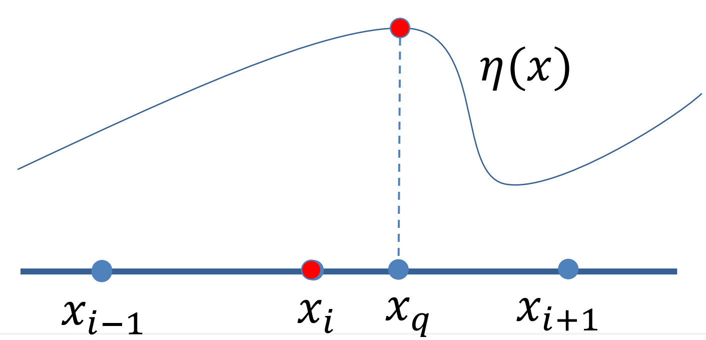
- As W depends on η, derivative-based solvers need its derivatives in $\boldsymbol{x}$.

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

$$\frac{\partial \mu(T)}{\partial x_{ij}} = \left(\nabla \bar{w}_i W^{-1}(\eta(x)) - \underbrace{A(x) W^{-1} \frac{\partial W}{\partial \eta} \left(\nabla \eta \cdot \frac{\partial x}{\partial x_{ij}} \right) W^{-1}}_{\textit{New term due to r-adaptivity}} \right) : \frac{\partial \mu(T)}{\partial T}$$

$$A(\boldsymbol{x}) = \sum_i \boldsymbol{x}_i \nabla \bar{w}_i(\bar{x}) \,, \quad x = \sum_i \boldsymbol{x}_i \bar{w}_i(\bar{x}) \,, \quad \eta(\boldsymbol{x}) = \sum_i \eta_i w_i(\boldsymbol{x}) \,, \quad \left[\frac{\partial \mu}{\partial T} \right]_{kl} = \frac{\partial \mu}{\partial T_{kl}}$$

- The above derivative is still an approximation. It doesn't consider transfer errors.



Adapted target matrices

The target matrices depend on the adaptivity function

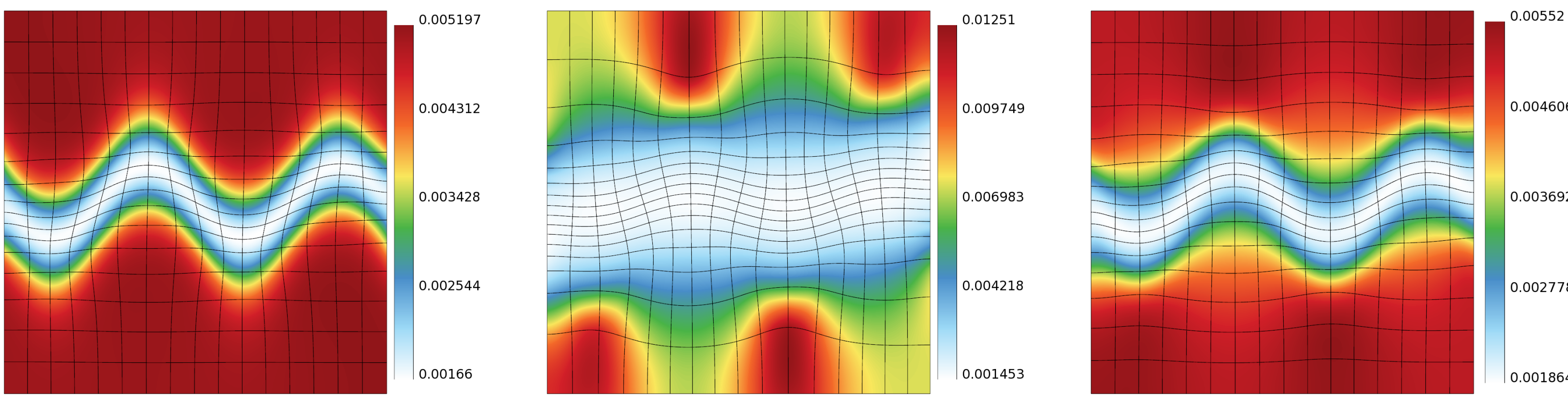
- The adaptivity function can influence any of the target components.

$W =$ [volume] [orientation] [skew] [aspect ratio].

- 2D Size adaptation: $W_1(x) = [\eta(x)s + (1-\eta(x))\alpha s]^{1/2} I\,, \quad \mu_7(T) = |T - T^{-t}|^2$

- Aspect ratio: $W_2(x) = \begin{pmatrix} \eta(x)\beta + (1-\eta(x)) & 0 \\ 0 & 1 \end{pmatrix}, \quad \mu_{58}(T) = \dfrac{|T^tT|^2}{\det(T)^2} - 2\dfrac{|T|^2}{\det(T)} + 2$

- Size+Aspect ratio: $W_3(x) = \begin{pmatrix} \sqrt{\gamma s} & 0 \\ 0 & \eta(x)\frac{\sqrt{\gamma s}}{\gamma} + (1-\eta(x))\sqrt{\gamma s} \end{pmatrix}, \quad \mu_9(T) = \det(T)|T - T^{-t}|^2$

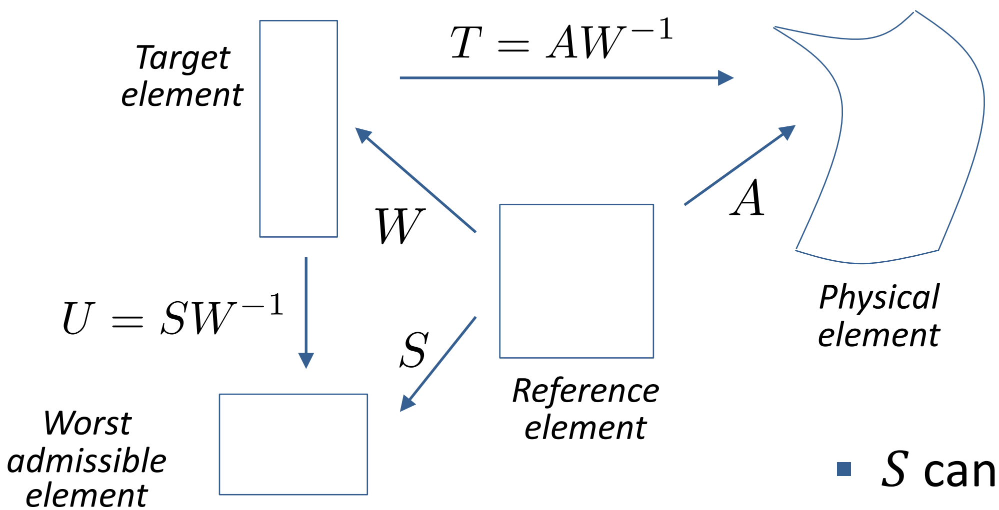


TMOP-based ALE triggers

User-defined admissible local quality can trigger the ALE step

- Moving mesh simulations need to perform ALE steps periodically.
  - Too few - deterioration of mesh quality can cause simulation failure.
  - Too frequent - affects performance and accuracy (no physics).

- The user defines admissible local quality.
  Jacobian S of the reference -> *worst admissible* element transformation.



- ALE step is triggered whenever

$$\mu(T) > \mu(U)$$

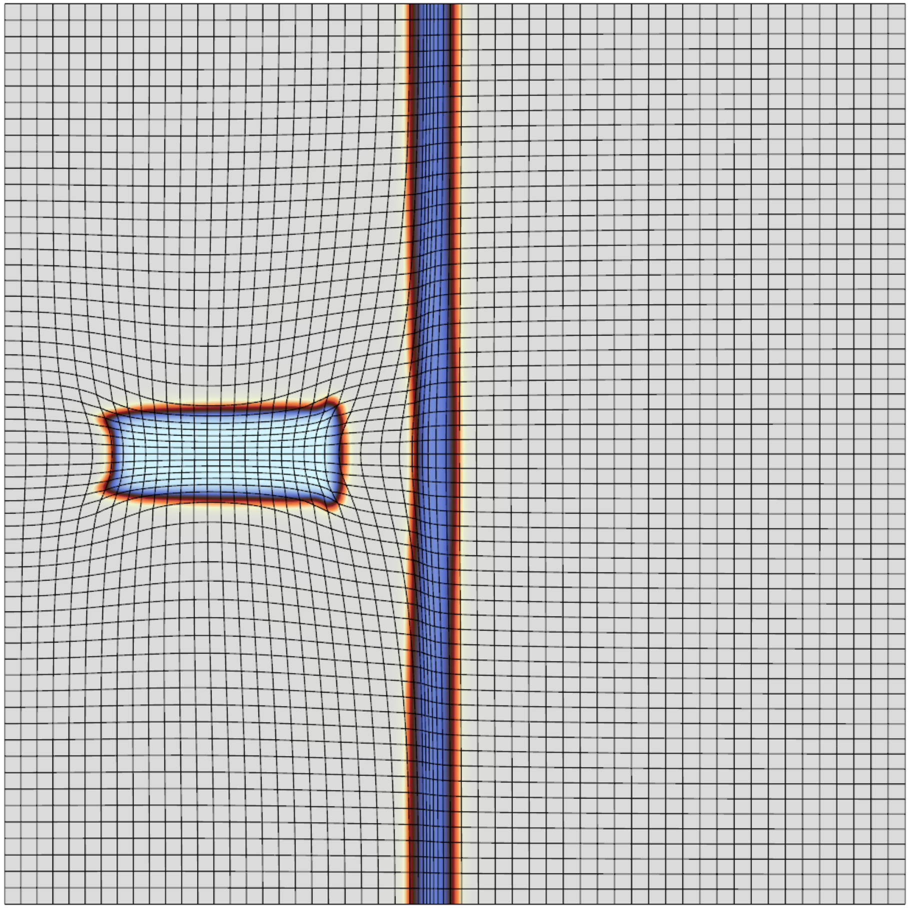
- Example: $S = \begin{pmatrix} 1 & 0 \\ 0 & S_{22} \end{pmatrix}$

- S can be adapted through η.

$S =$ [volume] [orientation] [skew] [aspect ratio].

Example of r-adaptivity in an ALE production code

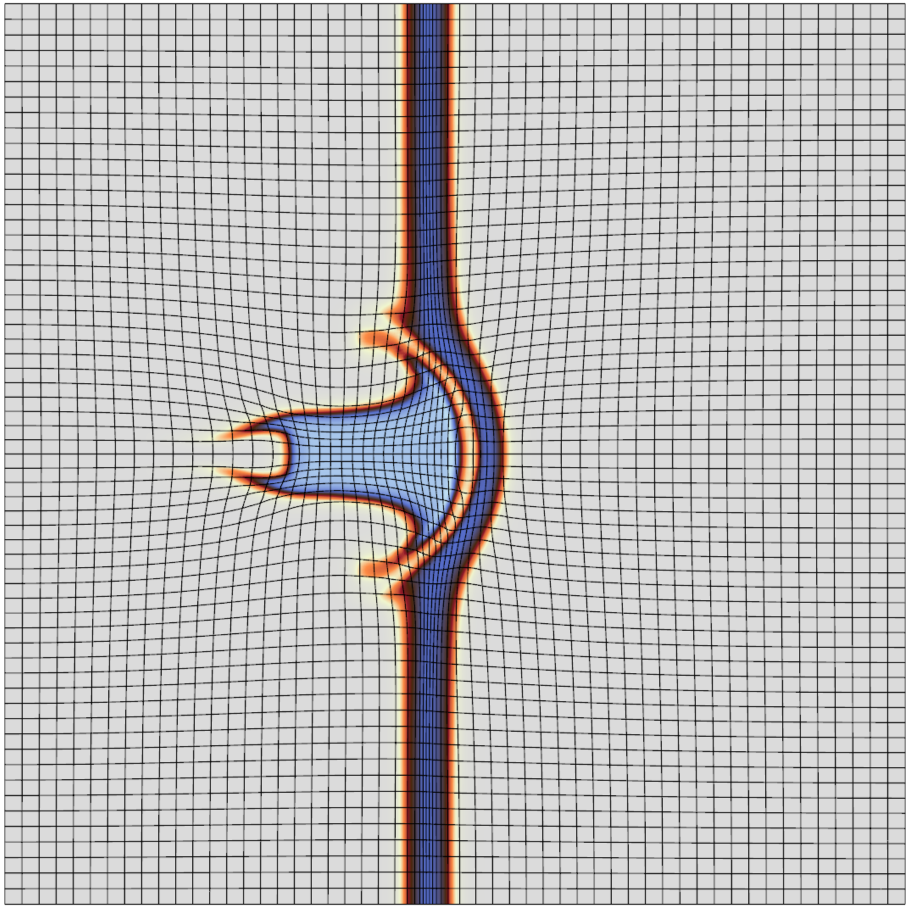
- 2D gas impact - the mesh adapts to the positions of the flyer and the wall.



t = 0.5

Example of r-adaptivity in an ALE production code

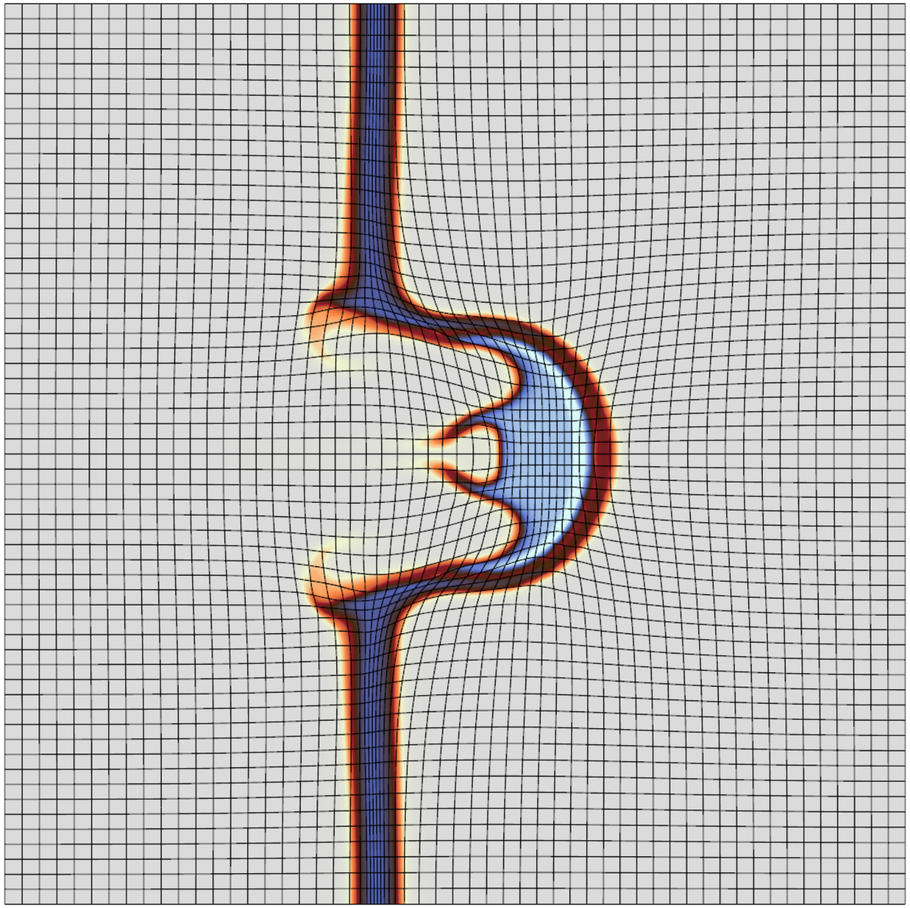
- 2D gas impact - the mesh adapts to the positions of the flyer and the wall.



t = 2.5

Example of r-adaptivity in an ALE production code

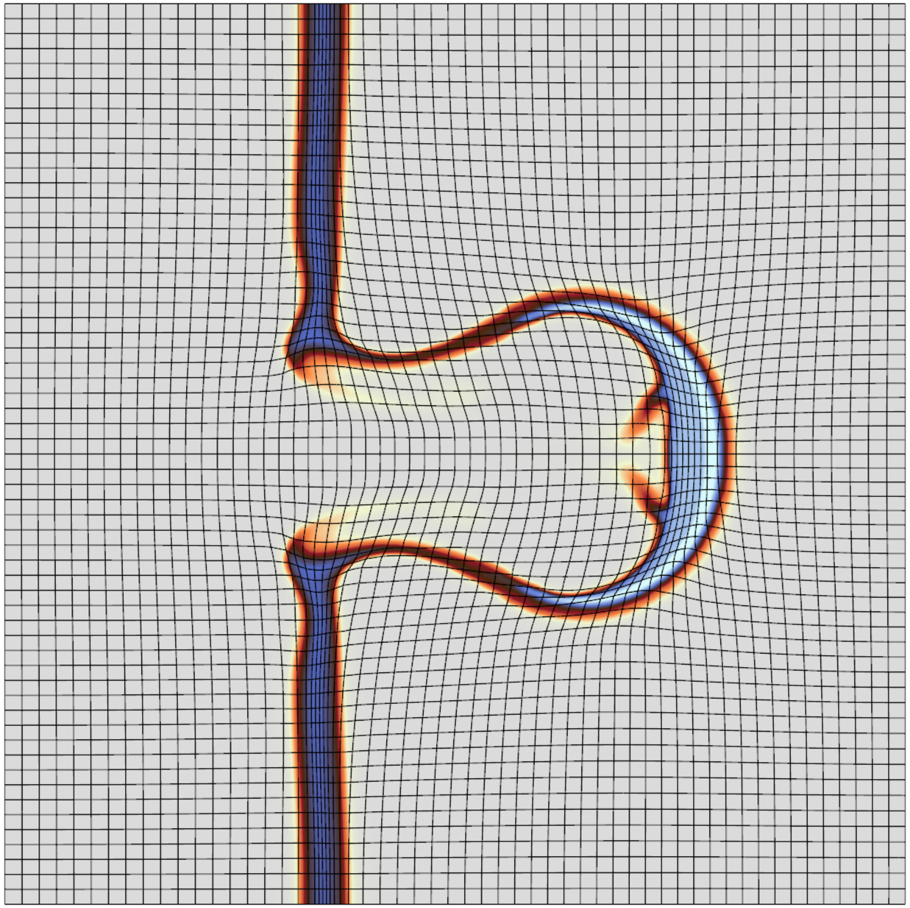
- 2D gas impact - the mesh adapts to the positions of the flyer and the wall.



t = 5

Example of r-adaptivity in an ALE production code

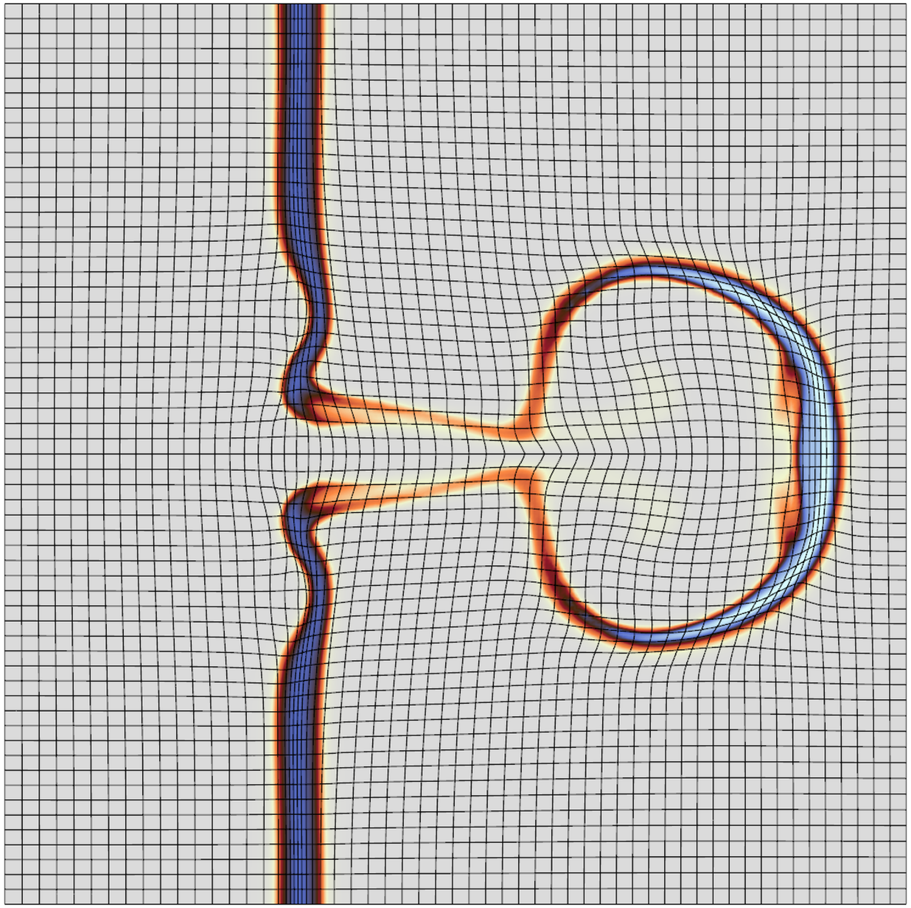
- 2D gas impact - the mesh adapts to the positions of the flyer and the wall.



t = 7.5

Example of r-adaptivity in an ALE production code

- 2D gas impact - the mesh adapts to the positions of the flyer and the wall.

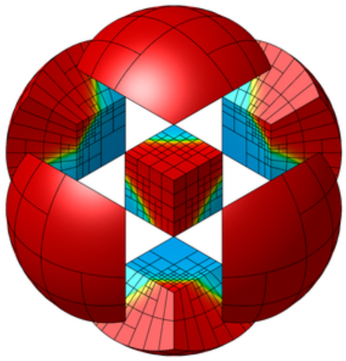


t = 10

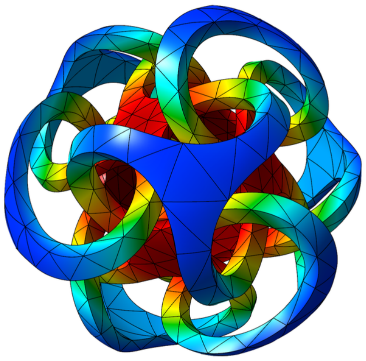
MFEM open source implementation

- All presented methods are (or will be) available in MFEM.

- MFEM contains **12** 2D metrics, **7** 3D metrics, all metric derivatives, **6** target construction methods.



mfem.org



glvis.org

- User interface provided by the ***mesh_optimizer*** and ***pmesh_optimizer*** miniapps.
  - Choice of target construction / quality metric / adaptivity fields / parameters.
  - Visualization through GLVis.

V. Dobrev, P. Knupp, Tz. Kolev, K. Mittal, V. Tomov, “The target-matrix optimization paradigm for high-order meshes”, SISC, 2018.
V. Dobrev, P. Knupp, Tz. Kolev, V. Tomov, “Towards simulation-driven optimization of high-order meshes by the target-matrix optimization paradigm”, IMR 2018 Proceedings.

Summary and future work

- General TMOP framework for improving quality of high-order curved meshes.
  - Point-wise quality metrics + target constructions.
  - Sub-element control over shape / size / alignment.

- Extension to r-adaptivity based on discrete adaptivity fields.
  - Interpolation / advection to obtain values of η on intermediate meshes.
  - Additional derivative terms in the nonlinear solvers.

- ALE triggers and results, availability through MFEM.

- Future work:
  - General interface for the adaptivity process.
  - Preservation of discrete surfaces during optimization.
  - Optimization of non-conforming meshes.
  - Combination of TMOP and AMR adaptivity (hr-adaptivity).
  - Improved nonlinear solvers, physical interpolation, general TMOP components.

CASC

Center for Applied Scientific Computing

Lawrence Livermore National Laboratory