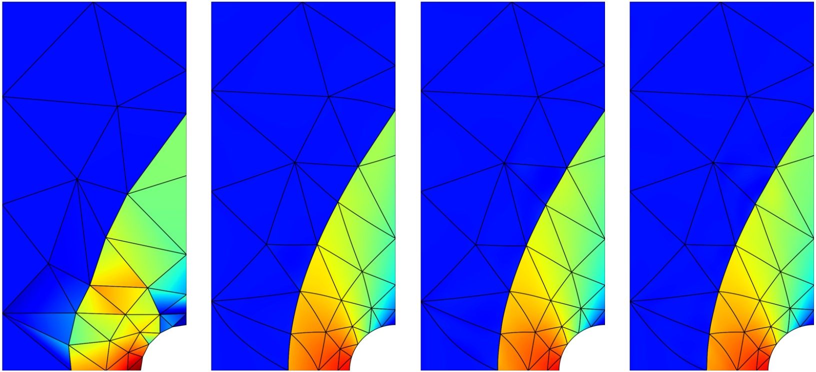
A robust, high-order implicit shock tracking method for high-speed flows

Matthew J. Zahr
Aerospace and Mechanical Engineering
University of Notre Dame

Computational Hypersonics Research Lab Seminar
University of Minnesota
May 18, 2021

Collaborators: Tianci Huang, Charles Naudet, Andrew Shi, Per-Olof Persson

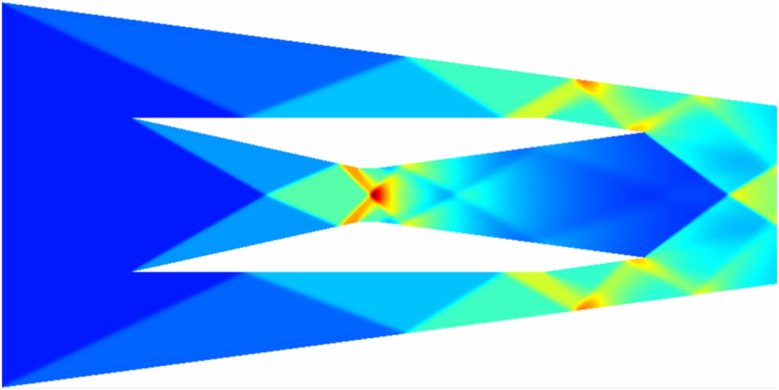
Why high-order tracking: Accurate solutions on coarse meshes



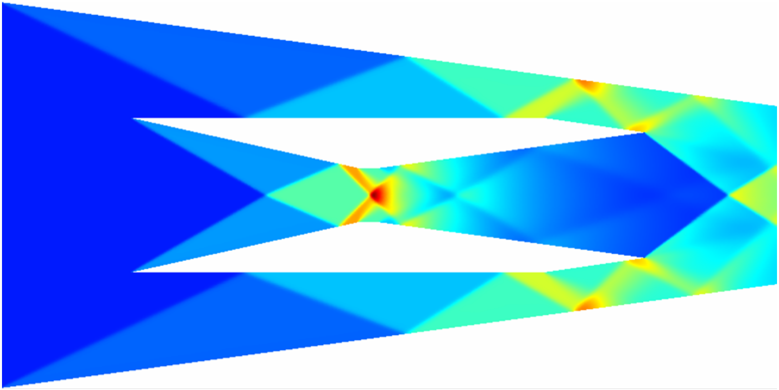
Density of supersonic flow ($M = 2$) past a cylinder using implicit shock tracking with $p = 1$ to $p = 4$ (left to right) DG discretization.

Key observation: High-order tracking enables accurate resolution of 2D supersonic flow with 48 elements; the error in the stagnation enthalpy is $\mathcal{O}(10^{-4})$ for $p = 2$ (1152 DoF).

Why not tracking: Difficult for complex discontinuity surfaces



Why not tracking: Difficult for complex discontinuity surfaces

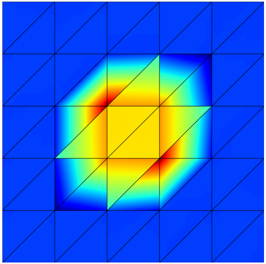


Implicit shock tracking

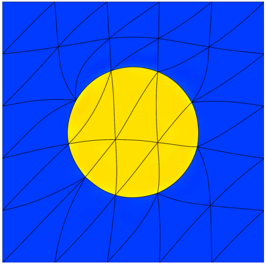
Aims to overcome the difficulty of explicitly meshing the unknown shock surface, e.g., HOIST [Zahr, Persson; 2018], MDG-ICE [Corrigan, Kercher, Kessler; 2019]

Implicit tracking for stable, high-order resolution of discontinuities

Goal: Align element faces with (unknown) discontinuities to perfectly capture them and approximate smooth regions to high-order



Non-aligned



Discontinuity-aligned

High-order implicit shock tracking (HOIST)[1]

- Discontinuous Galerkin discretization: inter-element jumps, high-order
- Discontinuity-aligned mesh: solution of optimization problem constrained by the discrete PDE $\implies$ **implicit tracking**
- Full space solver that converges the solution and mesh simultaneously to ensure solution of PDE never required on non-aligned mesh

[1] [Zahr, Persson; 2018], [Zahr, Shi, Persson; 2020]

Discontinuous Galerkin discretization of conservation law

Inviscid conservation law:

$$\nabla \cdot F(U) = 0 \quad \text{in } \Omega$$

Element-wise finite-dimensional weak form of conservation law:

$$r^K_{h,p'}(U_{h,p}) := \int_{\partial K} \psi^+_{h,p'} \cdot \mathcal{H}(U^+_{h,p}, U^-_{h,p}, n) \, dS - \int_K F(U_{h,p}) : \nabla \psi_{h,p'} \, dV,$$

where $\mathcal{V}_{h,p'}$ is the test space, $\mathcal{V}_{h,p}$ is the trial space, $\mathcal{H}$ is the numerical flux function, h is element size, and p/p' is the polynomial degree.

Introduce basis for polynomial spaces to obtain discrete residuals

$$\boldsymbol{r}(\boldsymbol{u}, \boldsymbol{x}) \quad (p' = p), \qquad \boldsymbol{R}(\boldsymbol{u}, \boldsymbol{x}) \quad (p' = p + 1),$$

where $\boldsymbol{u}$ is the discrete state vector and $\boldsymbol{x}$ are the coordinates of the mesh nodes.

Implicit shock tracking: constrained optimization formulation

We formulate the problem of tracking discontinuities with the mesh as the solution of an optimization problem constrained by the discrete PDE (DG discretization)

$$\begin{aligned} \underset{\boldsymbol{u},\boldsymbol{x}}{\text{minimize}} \quad & f(\boldsymbol{u},\boldsymbol{x}) := \frac{1}{2}\left\|\boldsymbol{F}(\boldsymbol{u},\boldsymbol{x})\right\|_2^2 \\ \text{subject to} \quad & \boldsymbol{r}(\boldsymbol{u},\boldsymbol{x}) = \boldsymbol{0}. \end{aligned}$$

The objective function *balances* tracking and mesh quality

$$\boldsymbol{F}(\boldsymbol{u},\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{R}(\boldsymbol{u},\boldsymbol{x}) \\ \kappa \boldsymbol{R}_{\text{msh}}(\boldsymbol{x}) \end{bmatrix}$$

$\boldsymbol{r}(\boldsymbol{u},\boldsymbol{x}) = \boldsymbol{0}$ (DG equation), $\boldsymbol{u}$ (discrete state vector), $\boldsymbol{x}$ (coordinates of mesh nodes)

$\boldsymbol{R}$ (tracking term): penalizes the DG residual in the *enriched test space*

$\boldsymbol{R}_{\text{msh}}$ (mesh term): accounts for the distortion of each high-order element

κ: mesh distortion penalization parameter

Implicit shock tracking: sequential quadratic programming solver

Define $\boldsymbol{z} = (\boldsymbol{u}, \boldsymbol{x})$ and use interchangeably. To solve the optimization problem, we define a sequence $\{\boldsymbol{z}_k\}$ updated as

$$\boldsymbol{z}_{k+1} = \boldsymbol{z}_k + \alpha_k \Delta \boldsymbol{z}_k.$$

Implicit shock tracking: sequential quadratic programming solver

Define $\boldsymbol{z} = (\boldsymbol{u}, \boldsymbol{x})$ and use interchangeably. To solve the optimization problem, we define a sequence $\{\boldsymbol{z}_k\}$ updated as

$$\boldsymbol{z}_{k+1} = \boldsymbol{z}_k + \alpha_k \Delta \boldsymbol{z}_k.$$

The step direction $\Delta \boldsymbol{z}_k$ is defined as the solution of the quadratic program (QP) approximation of the tracking problem centered at $\boldsymbol{z}_k$

$$\begin{aligned} \underset{\Delta \boldsymbol{z} \in \mathbb{R}^{N_{\boldsymbol{z}}}}{\text{minimize}} \quad & \boldsymbol{g}_{\boldsymbol{z}}(\boldsymbol{z}_k)^T \Delta \boldsymbol{z} + \frac{1}{2} \Delta \boldsymbol{z}^T \boldsymbol{B}_{\boldsymbol{z}}(\boldsymbol{z}_k, \hat{\boldsymbol{\lambda}}(\boldsymbol{z}_k)) \Delta \boldsymbol{z} \\ \text{subject to} \quad & \boldsymbol{r}(\boldsymbol{z}_k) + \boldsymbol{J}_{\boldsymbol{z}}(\boldsymbol{z}_k) \Delta \boldsymbol{z} = \boldsymbol{0}, \end{aligned}$$

where

$$\boldsymbol{g}_{\boldsymbol{z}}(\boldsymbol{z}) = \frac{\partial f}{\partial \boldsymbol{z}}(\boldsymbol{z})^T, \quad \boldsymbol{J}_{\boldsymbol{z}}(\boldsymbol{z}) = \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{z}}(\boldsymbol{z}), \qquad \boldsymbol{B}_{\boldsymbol{z}}(\boldsymbol{z}, \boldsymbol{\lambda}) \approx \frac{\partial^2 \mathcal{L}}{\partial \boldsymbol{z} \partial \boldsymbol{z}}(\boldsymbol{z}, \boldsymbol{\lambda}),$$

$$\mathcal{L}(\boldsymbol{z}, \boldsymbol{\lambda}) = f(\boldsymbol{z}) - \boldsymbol{\lambda}^T \boldsymbol{r}(\boldsymbol{z}) \qquad \text{(Lagrangian)}$$

$$\hat{\boldsymbol{\lambda}}(\boldsymbol{z}) = \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{z})^{-T} \frac{\partial f}{\partial \boldsymbol{u}}(\boldsymbol{z})^T \qquad \text{(Lagrange mulitplier estimate)}$$

Implicit shock tracking: sequential quadratic programming solver

The solution of the quadratic program leads to the following linear system

$$\begin{bmatrix} \boldsymbol{B}_{\boldsymbol{u}\boldsymbol{u}}(\boldsymbol{z}_k, \hat{\boldsymbol{\lambda}}(\boldsymbol{z}_k)) & \boldsymbol{B}_{\boldsymbol{u}\boldsymbol{x}}(\boldsymbol{z}_k, \hat{\boldsymbol{\lambda}}(\boldsymbol{z}_k)) & \boldsymbol{J}_{\boldsymbol{u}}(\boldsymbol{z}_k)^T \\ \boldsymbol{B}_{\boldsymbol{u}\boldsymbol{x}}(\boldsymbol{z}_k, \hat{\boldsymbol{\lambda}}(\boldsymbol{z}_k))^T & \boldsymbol{B}_{\boldsymbol{x}\boldsymbol{x}}(\boldsymbol{z}_k, \hat{\boldsymbol{\lambda}}(\boldsymbol{z}_k)) & \boldsymbol{J}_{\boldsymbol{x}}(\boldsymbol{z}_k)^T \\ \boldsymbol{J}_{\boldsymbol{u}}(\boldsymbol{z}_k) & \boldsymbol{J}_{\boldsymbol{x}}(\boldsymbol{z}_k) & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \Delta \boldsymbol{u}_k \\ \Delta \boldsymbol{x}_k \\ \boldsymbol{\eta}_k \end{bmatrix} = - \begin{bmatrix} \boldsymbol{g}_{\boldsymbol{u}}(\boldsymbol{z}_k) \\ \boldsymbol{g}_{\boldsymbol{x}}(\boldsymbol{z}_k) \\ \boldsymbol{r}(\boldsymbol{z}_k) \end{bmatrix},$$

where

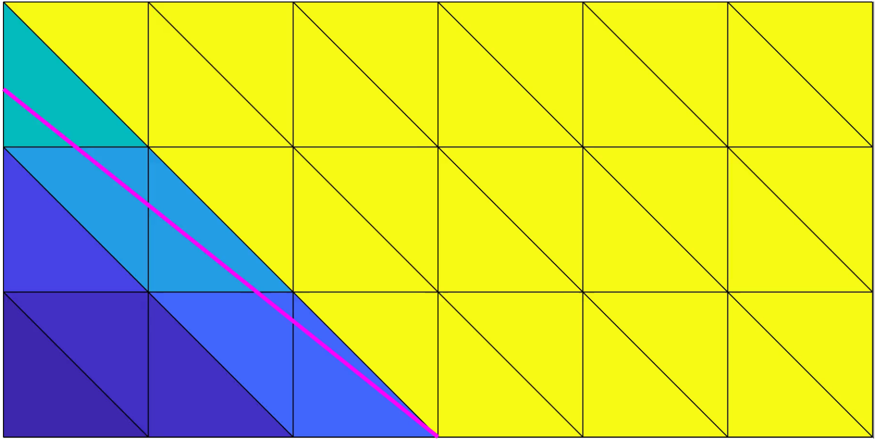
$$\boldsymbol{g}_{\boldsymbol{u}}(\boldsymbol{z}) = \frac{\partial f}{\partial \boldsymbol{u}}(\boldsymbol{z})^T, \quad \boldsymbol{J}_{\boldsymbol{u}}(\boldsymbol{z}) = \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{z}), \quad \boldsymbol{g}_{\boldsymbol{x}}(\boldsymbol{z}) = \frac{\partial f}{\partial \boldsymbol{x}}(\boldsymbol{z})^T, \quad \boldsymbol{J}_{\boldsymbol{x}}(\boldsymbol{z}) = \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{x}}(\boldsymbol{z}),$$

the approximate Hessian of the Lagrangian is taken as

$$\begin{aligned} \boldsymbol{B}_{\boldsymbol{u}\boldsymbol{u}}(\boldsymbol{z}, \boldsymbol{\lambda}) &= \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{u}}(\boldsymbol{z})^T \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{u}}(\boldsymbol{z}), \quad \boldsymbol{B}_{\boldsymbol{u}\boldsymbol{x}}(\boldsymbol{z}, \boldsymbol{\lambda}) = \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{u}}(\boldsymbol{z})^T \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}}(\boldsymbol{z}), \\ \boldsymbol{B}_{\boldsymbol{x}\boldsymbol{x}}(\boldsymbol{z}, \boldsymbol{\lambda}) &= \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}}(\boldsymbol{z})^T \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}}(\boldsymbol{z}) + \gamma \boldsymbol{D}, \end{aligned}$$

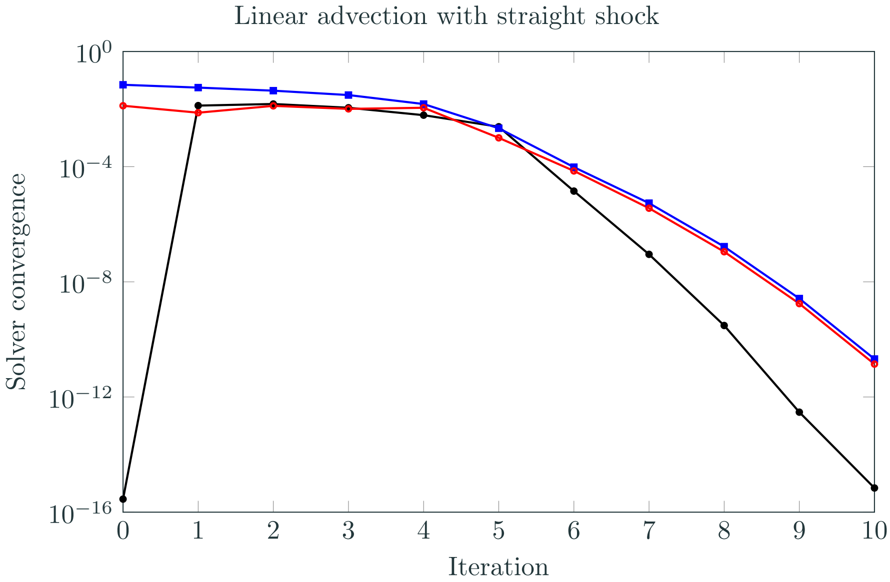
and $\boldsymbol{\eta}_k$ are the Lagrange multipliers of the QP and $\boldsymbol{D}$ is a mesh regularization matrix (linear elasticity stiffness).

Linear advection (2D), straight shock



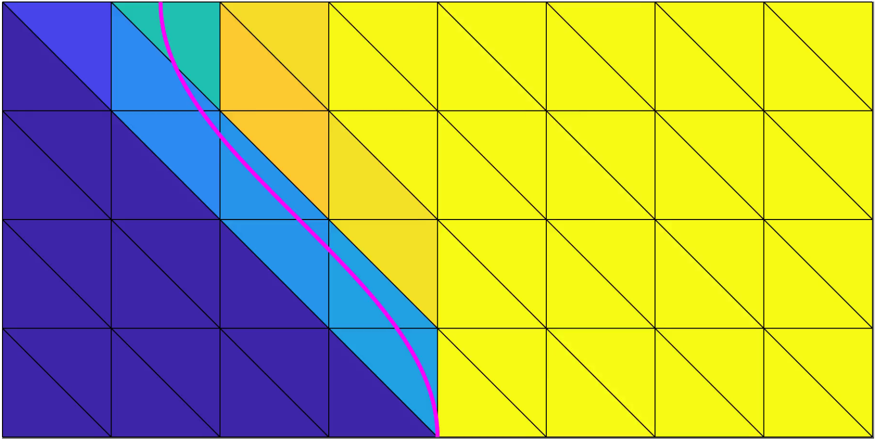
$p = 0$ space for solution, $q = 1$ space for mesh

Newton-like convergence when solution lies in DG subspace



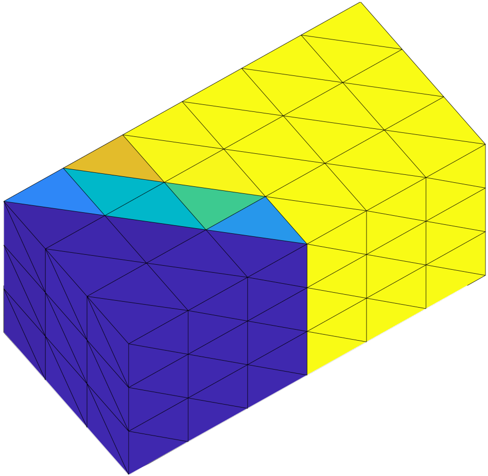
$\|\boldsymbol{r}(\boldsymbol{z})\|$ (—●—), $\|\boldsymbol{R}(\boldsymbol{z})\|$ (—■—), $\left\|\nabla_{\boldsymbol{x}}\mathcal{L}(\boldsymbol{z},\hat{\boldsymbol{\lambda}}(\boldsymbol{z}))\right\|$ (—○—)

Linear advection (2D), trigonometic shock



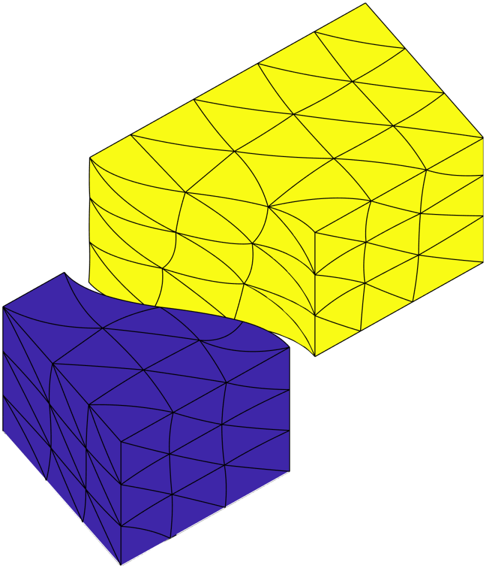
$p = 0$ space for solution, $q = 2$ space for mesh

Linear advection (3D), trigonometric shock



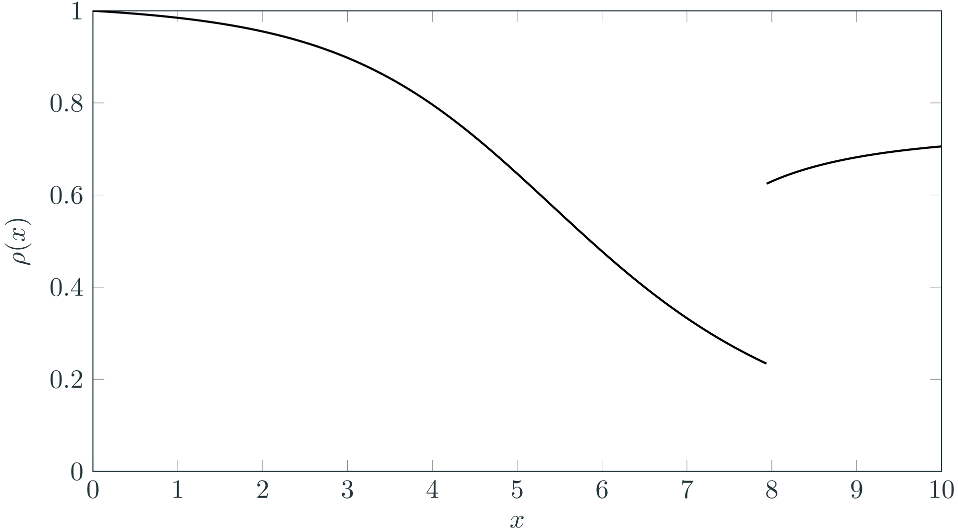
$p = 0$ space for solution, $q = 2$ space for mesh

Linear advection (3D), trigonometric shock



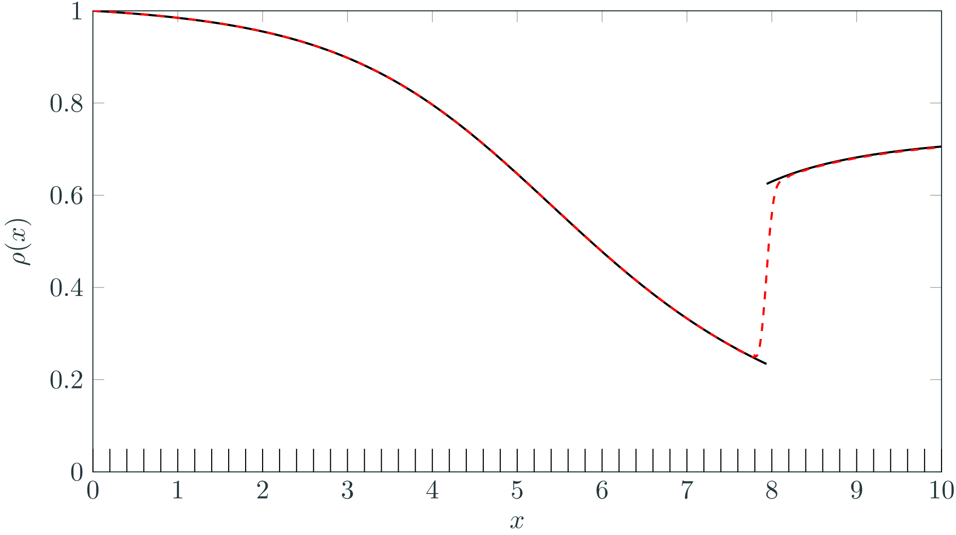
$p = 0$ space for solution, $q = 2$ space for mesh

Inviscid flow through area variation: HOIST vs capturing ($p = 4$)



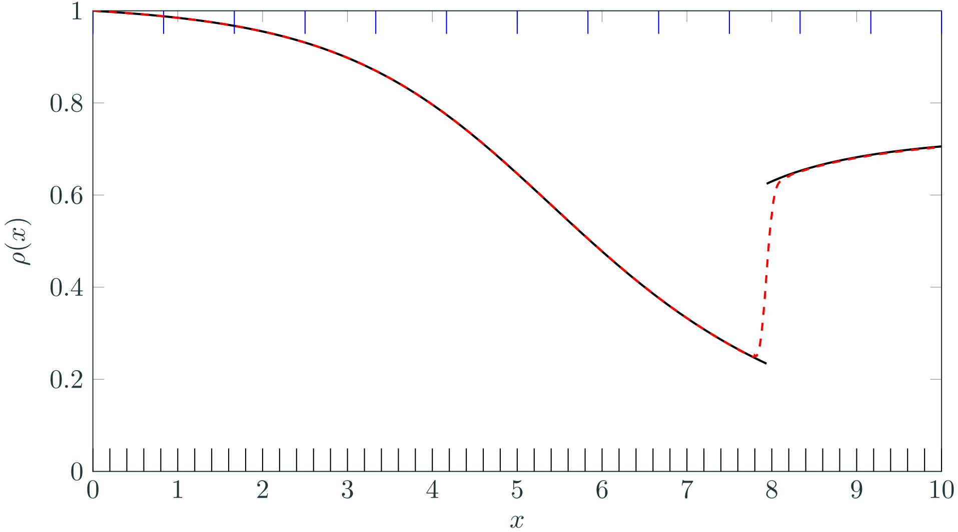
Exact solution (——), shock capturing (- - -), HOIST (······)

Inviscid flow through area variation: HOIST vs capturing ($p = 4$)



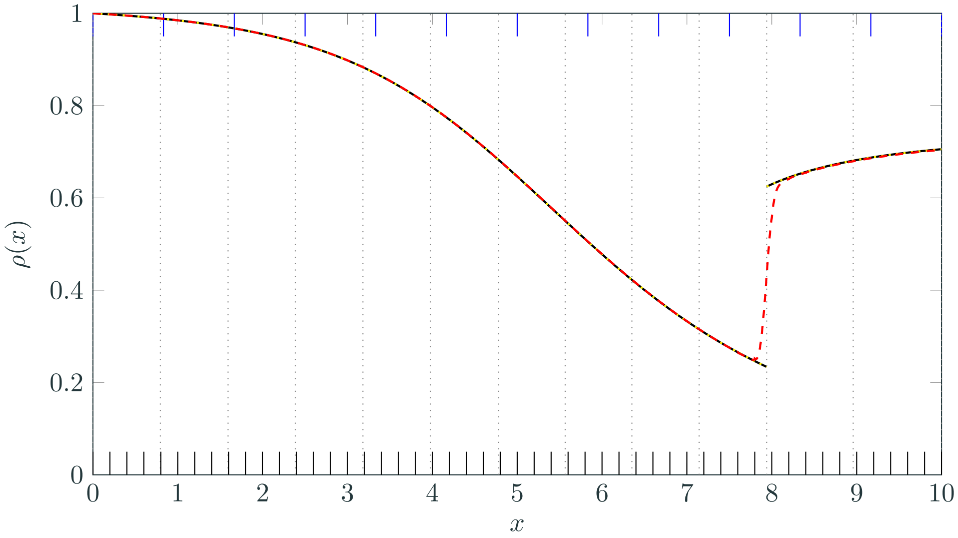
Exact solution (——), shock capturing (- - -), HOIST (······)

Inviscid flow through area variation: HOIST vs capturing ($p = 4$)



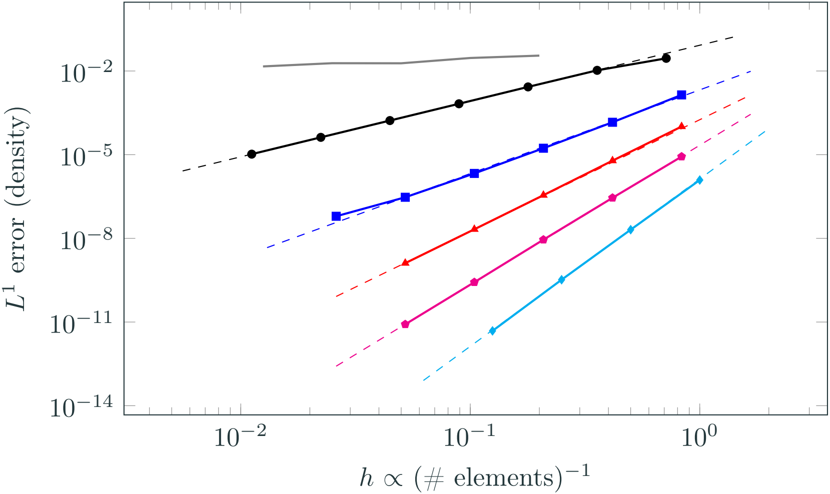
Exact solution (——), shock capturing (- - -), HOIST (⋯⋯)

Inviscid flow through area variation: HOIST vs capturing ($p = 4$)



Exact solution (——), shock capturing (- - -), HOIST (······)

Inviscid flow through area variation: h-convergence



Shock capturing: $p = 4$ (—); HOIST: $p = 1$ (—●—), $p = 2$ (—■—), $p = 3$ (—▲—), $p = 4$ (—⬟—), $p = 5$ (—◆—); dashed line indicates optimal convergence rate ($\mathcal{O}(h^{p+1})$)

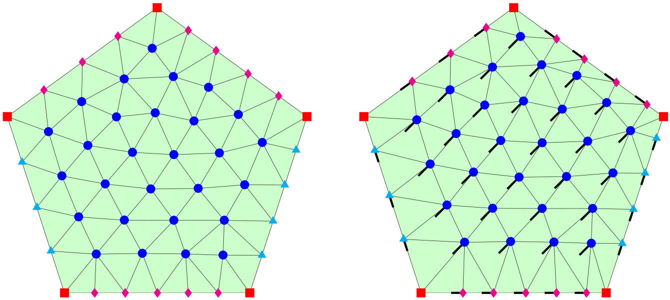
Observation: Shock capturing limited to first-order convergence rate; HOIST achieves optimal convergence rates ($\mathcal{O}(h^{p+1})$) and high accuracy per DoF

Construction of admissible mesh motion

Cannot directly optimize nodal coordinates ($\boldsymbol{x}$) without changing the domain; instead, construct mapping that guarantees mesh conforms to the domain boundaries from a collection of unconstrained degrees of freedom ($\boldsymbol{y}$) and directly optimize $\boldsymbol{y}$

$$\boldsymbol{x} = \boldsymbol{\phi}(\boldsymbol{y}) \qquad \Longrightarrow \qquad \begin{array}{ll} \underset{\boldsymbol{u},\boldsymbol{y}}{\text{minimize}} & f(\boldsymbol{u}, \boldsymbol{\phi}(\boldsymbol{y})) \\ \text{subject to} & \boldsymbol{r}(\boldsymbol{u}, \boldsymbol{\phi}(\boldsymbol{y})) = \boldsymbol{0} \end{array}$$

- Planar boundaries: $\boldsymbol{\phi}$ automatically constructed from normals

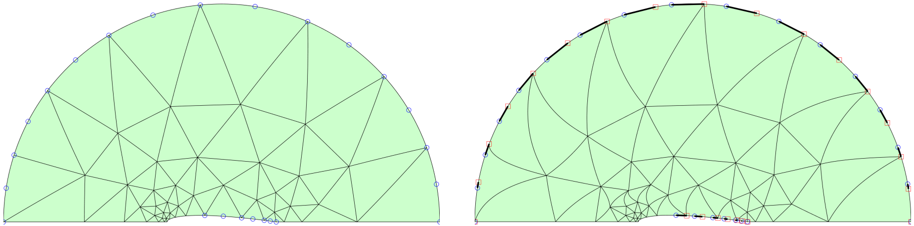


Construction of admissible mesh motion

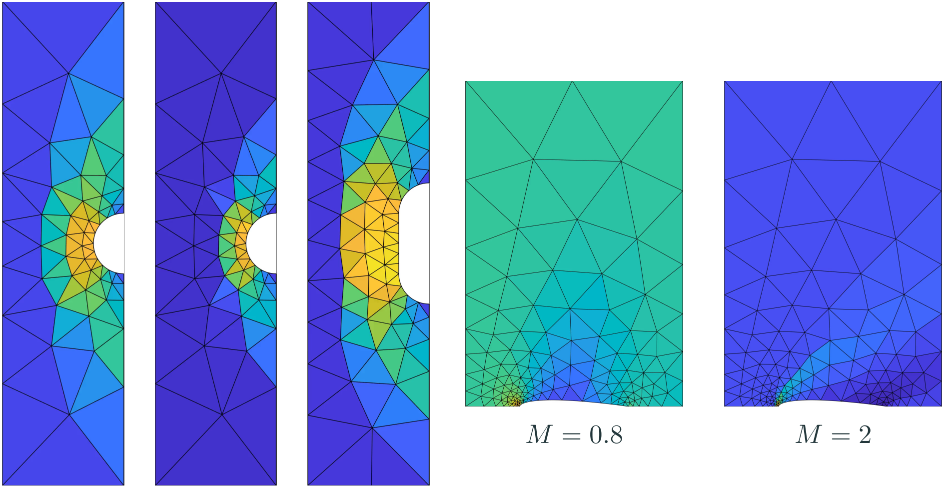
Cannot directly optimize nodal coordinates ($\boldsymbol{x}$) without changing the domain; instead, construct mapping that guarantees mesh conforms to the domain boundaries from a collection of unconstrained degrees of freedom ($\boldsymbol{y}$) and directly optimize $\boldsymbol{y}$

$$\boldsymbol{x} = \boldsymbol{\phi}(\boldsymbol{y}) \qquad \Longrightarrow \qquad \begin{aligned} \underset{\boldsymbol{u},\boldsymbol{y}}{\text{minimize}} \quad & f(\boldsymbol{u}, \boldsymbol{\phi}(\boldsymbol{y})) \\ \text{subject to} \quad & \boldsymbol{r}(\boldsymbol{u}, \boldsymbol{\phi}(\boldsymbol{y})) = \boldsymbol{0} \end{aligned}$$

- Planar boundaries: $\boldsymbol{\phi}$ automatically constructed from normals
- Curved boundaries: $\boldsymbol{\phi}$ defined from the analytical expression for the surface



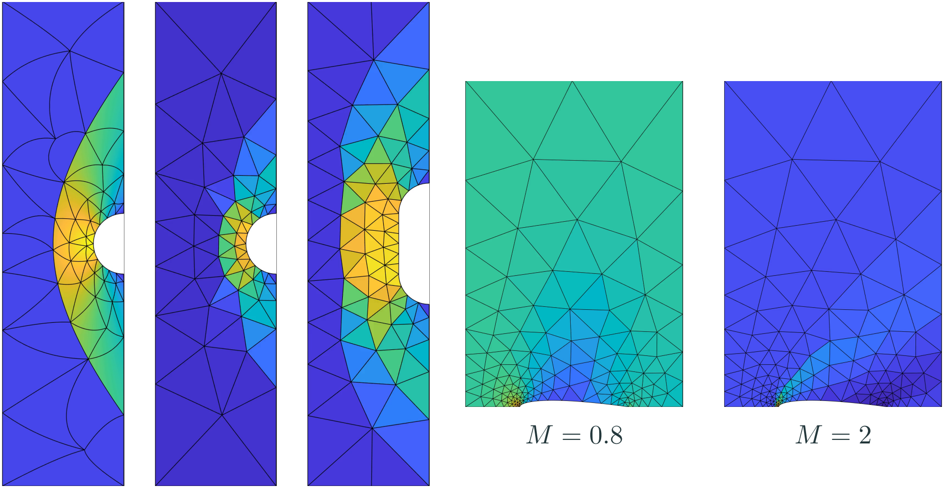
Implicit shock tracking for simple 2D compressible flows



$M = 2$ $M = 3$ $M = 4$ $M = 0.8$ $M = 2$

Observation: Quickly tracks bow shocks, shocks attached to curved boundaries, and secondary shocks; high-order elements curve to approximate curvature in shock surface; high-quality solutions on coarse high-order meshes ($\mathcal{O}(100)$ elements).

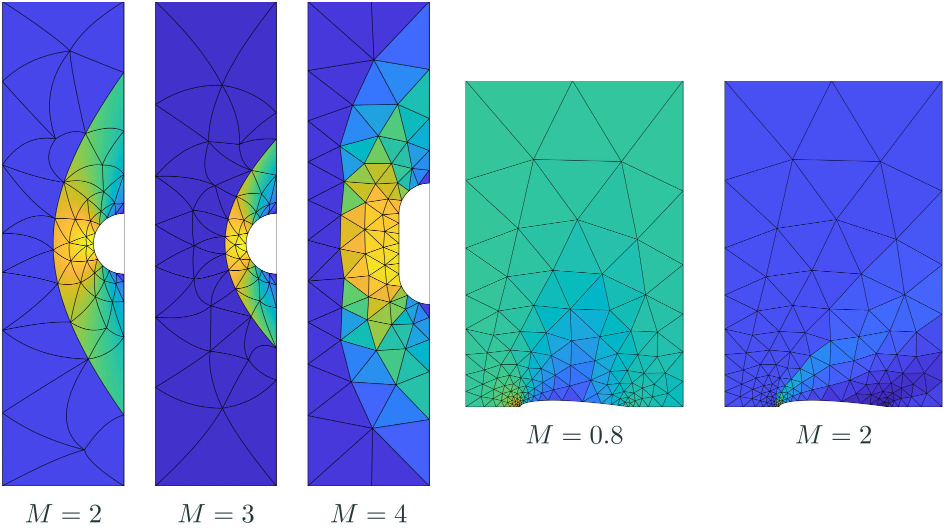
Implicit shock tracking for simple 2D compressible flows



$M = 2$ $M = 3$ $M = 4$ $M = 0.8$ $M = 2$

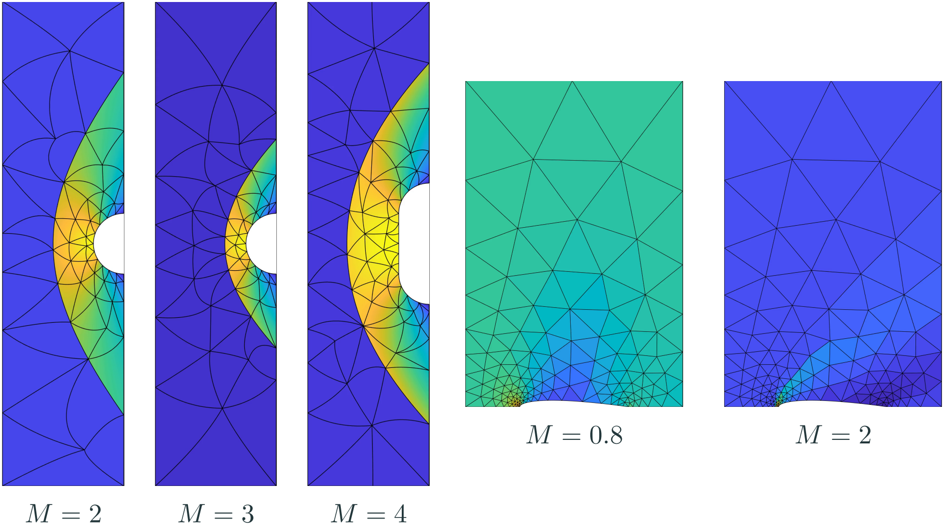
Observation: Quickly tracks bow shocks, shocks attached to curved boundaries, and secondary shocks; high-order elements curve to approximate curvature in shock surface; high-quality solutions on coarse high-order meshes ($\mathcal{O}(100)$ elements).

Implicit shock tracking for simple 2D compressible flows



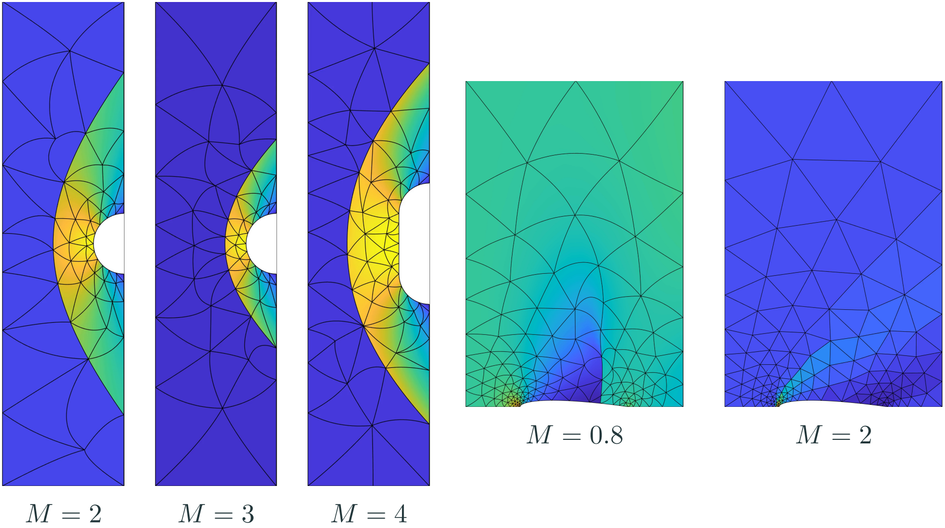
Observation: Quickly tracks bow shocks, shocks attached to curved boundaries, and secondary shocks; high-order elements curve to approximate curvature in shock surface; high-quality solutions on coarse high-order meshes ($\mathcal{O}(100)$ elements).

Implicit shock tracking for simple 2D compressible flows



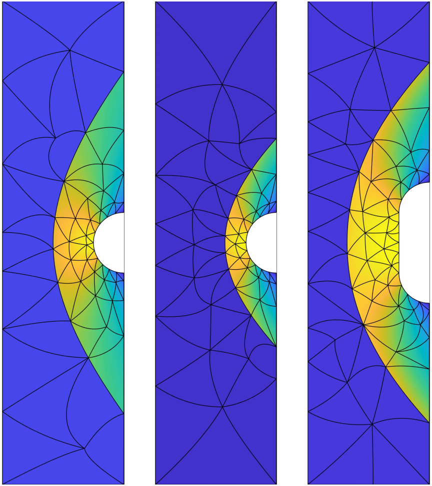
Observation: Quickly tracks bow shocks, shocks attached to curved boundaries, and secondary shocks; high-order elements curve to approximate curvature in shock surface; high-quality solutions on coarse high-order meshes ($\mathcal{O}(100)$ elements).

Implicit shock tracking for simple 2D compressible flows

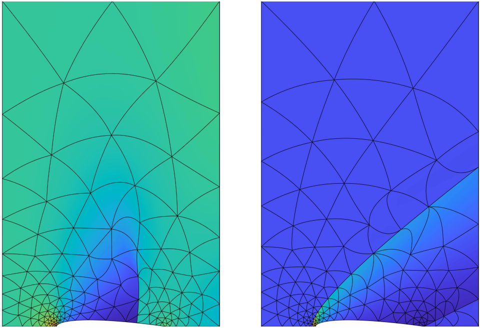


Observation: Quickly tracks bow shocks, shocks attached to curved boundaries, and secondary shocks; high-order elements curve to approximate curvature in shock surface; high-quality solutions on coarse high-order meshes ($\mathcal{O}(100)$ elements).

Implicit shock tracking for simple 2D compressible flows



$M = 2$ $M = 3$ $M = 4$



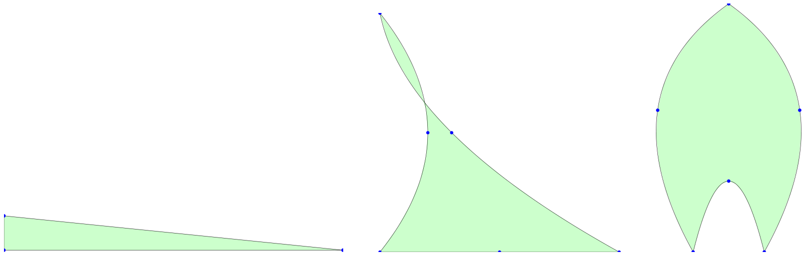
$M = 0.8$ $M = 2$

Observation: Quickly tracks bow shocks, shocks attached to curved boundaries, and secondary shocks; high-order elements curve to approximate curvature in shock surface; high-quality solutions on coarse high-order meshes ($\mathcal{O}(100)$ elements).

Practical considerations: element collapse

Despite measures to keep mesh well-conditioned, best option can be to *remove* element from the mesh

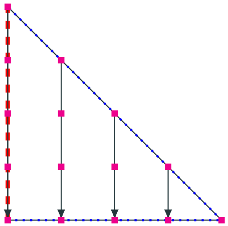
- Tag elements for removal based on volume, quality, edge length

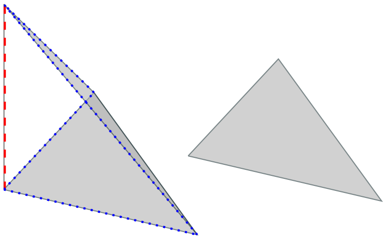


Practical considerations: element collapse

Despite measures to keep mesh well-conditioned, best option can be to *remove* element from the mesh

- Tag elements for removal based on volume, quality, edge length
- Collapse shortest edge: well-defined for simplices of any order in any dimension

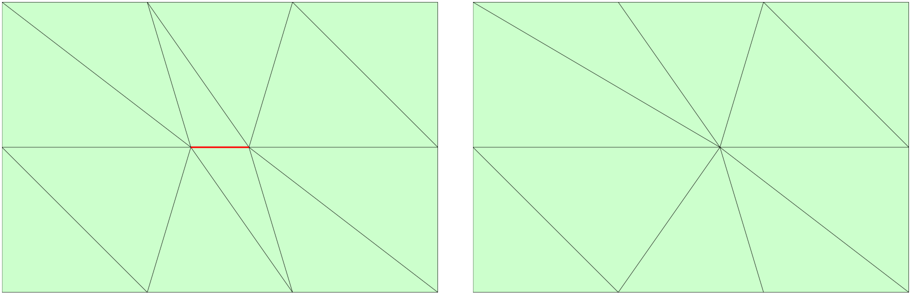




Practical considerations: element collapse

Despite measures to keep mesh well-conditioned, best option can be to *remove* element from the mesh

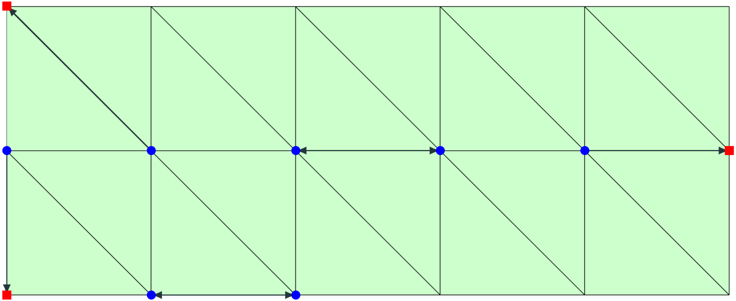
- Tag elements for removal based on volume, quality, edge length
- Collapse shortest edge: well-defined for simplices of any order in any dimension
- Remove all zero-volume elements



Practical considerations: element collapse

Despite measures to keep mesh well-conditioned, best option can be to *remove* element from the mesh

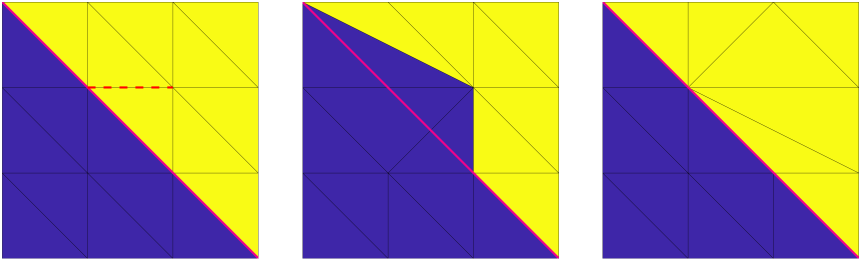
- Tag elements for removal based on volume, quality, edge length
- Collapse shortest edge: well-defined for simplices of any order in any dimension
- Remove all zero-volume elements
- Must preserve boundaries



Practical considerations: element collapse

Despite measures to keep mesh well-conditioned, best option can be to *remove* element from the mesh

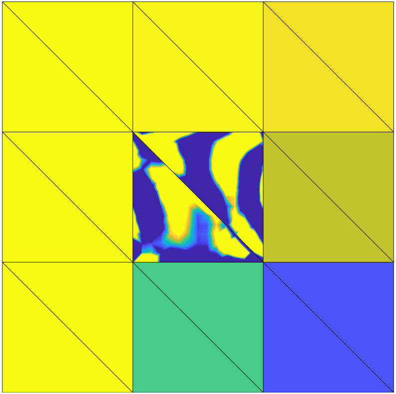
- Tag elements for removal based on volume, quality, edge length
- Collapse shortest edge: well-defined for simplices of any order in any dimension
- Remove all zero-volume elements
- Must preserve boundaries and shocks



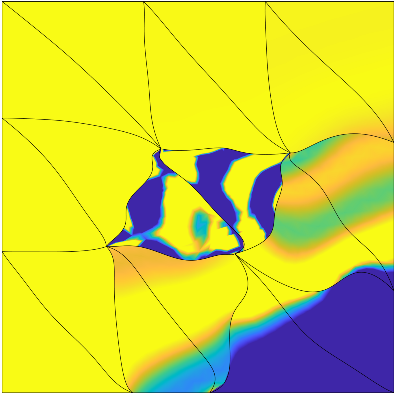
before collapse — ignore shock — shock-aware

Practical considerations: solution re-initialization

- High-order solutions can become oscillatory, which leads to poor SQP steps (requiring many line search iterations)



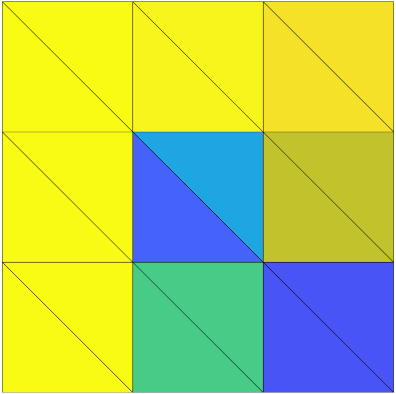
before SQP step (without re-init)



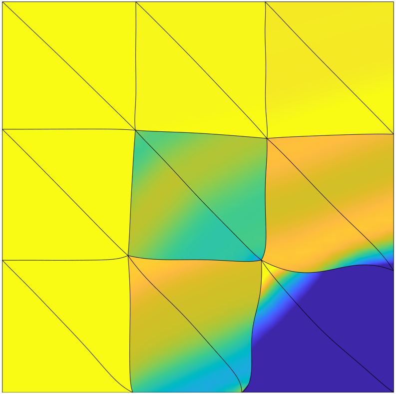
after SQP step (without re-init)

Practical considerations: solution re-initialization

- High-order solutions can become oscillatory, which leads to poor SQP steps (requiring many line search iterations)
- Overcome by replacing element-wise solution with the element-wise average (oscillatory element identified using Persson-Peraire indicator)



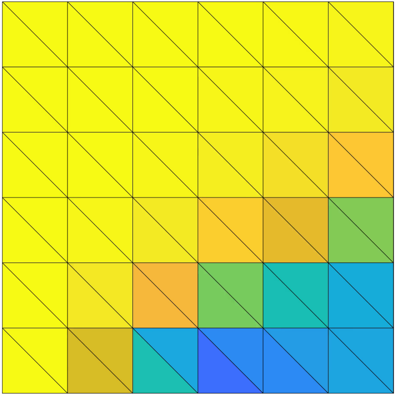
before SQP step (with re-init)



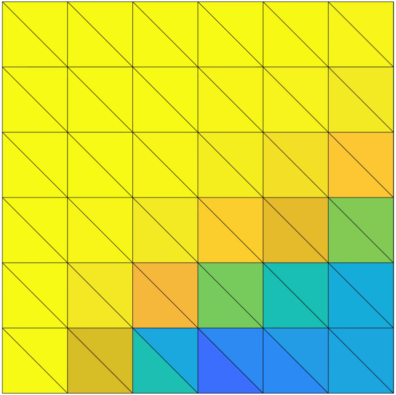
after SQP step (with re-init)

Practical considerations: solution re-initialization

- High-order solutions can become oscillatory, which leads to poor SQP steps (requiring many line search iterations)
- Overcome by replacing element-wise solution with the element-wise average (oscillatory element identified using Persson-Peraire indicator)
- Without re-initialization, must hope oscillatory elements get collapsed



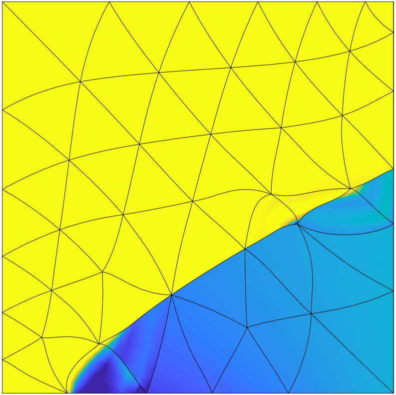
without re-initialization



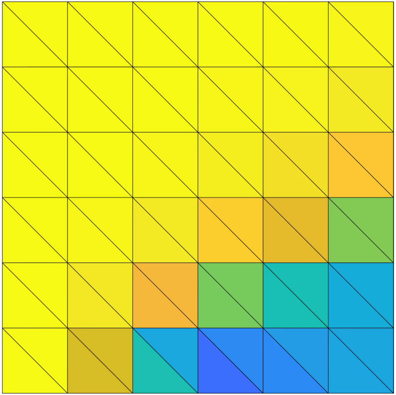
with re-initialization

Practical considerations: solution re-initialization

- High-order solutions can become oscillatory, which leads to poor SQP steps (requiring many line search iterations)
- Overcome by replacing element-wise solution with the element-wise average (oscillatory element identified using Persson-Peraire indicator)
- Without re-initialization, must hope oscillatory elements get collapsed



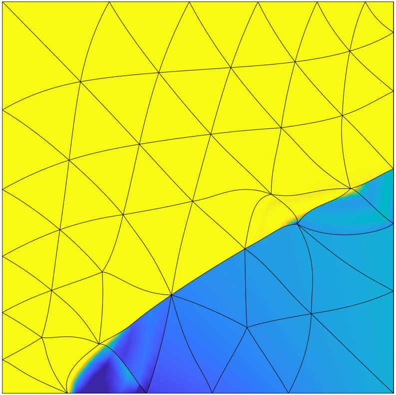
without re-initialization



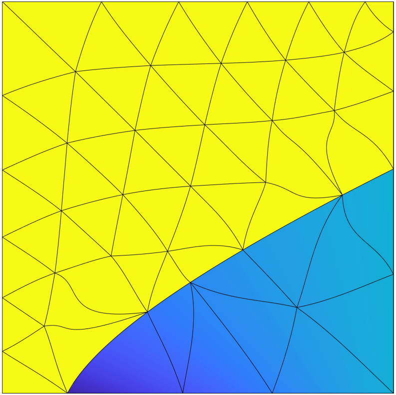
with re-initialization

Practical considerations: solution re-initialization

- High-order solutions can become oscillatory, which leads to poor SQP steps (requiring many line search iterations)
- Overcome by replacing element-wise solution with the element-wise average (oscillatory element identified using Persson-Peraire indicator)
- Without re-initialization, must hope oscillatory elements get collapsed



without re-initialization

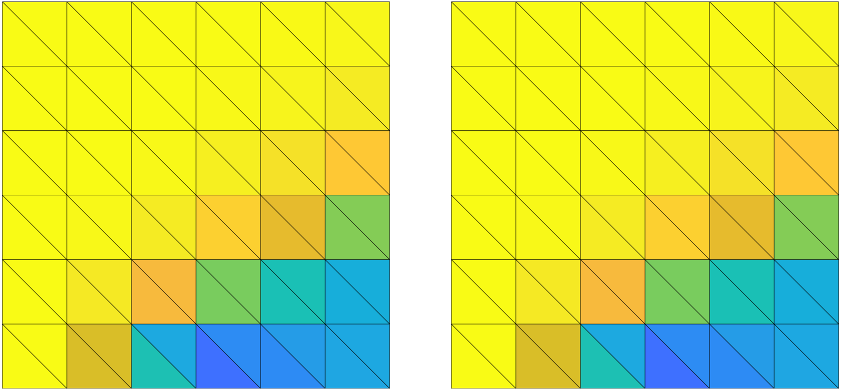


with re-initialization

Practical considerations: initialization

Robustness measures reduce sensitivity of solvers to initialization of $\boldsymbol{u}$, $\boldsymbol{x}$.

- $\boldsymbol{x}_0$: directly from mesh generation
- $\boldsymbol{u}_0$: DG($p = 0$) solution on mesh $\boldsymbol{x}_0$
- homotopy in p no longer required

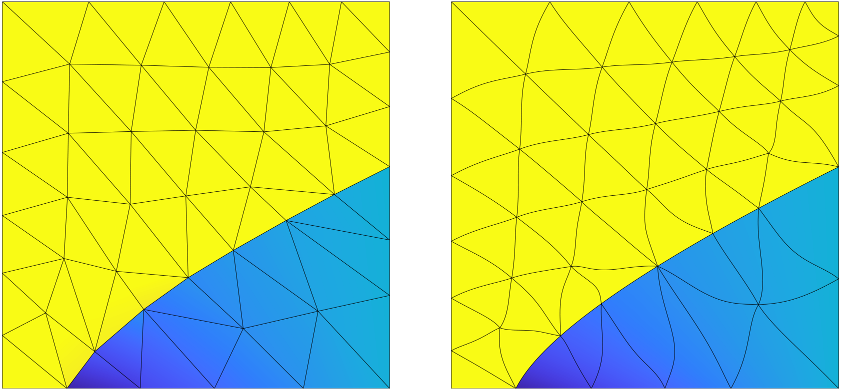


Reference mesh, $p = 0$ DG solution

Practical considerations: initialization

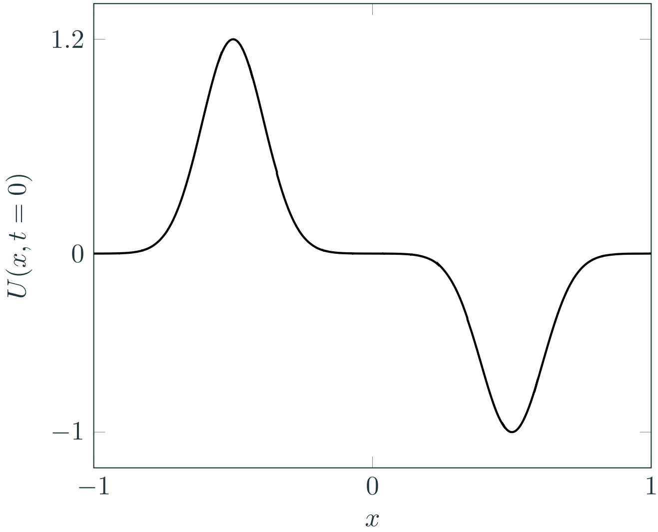
Robustness measures reduce sensitivity of solvers to initialization of $\boldsymbol{u}$, $\boldsymbol{x}$.

- $\boldsymbol{x}_0$: directly from mesh generation
- $\boldsymbol{u}_0$: DG($p = 0$) solution on mesh $\boldsymbol{x}_0$
- homotopy in p no longer required

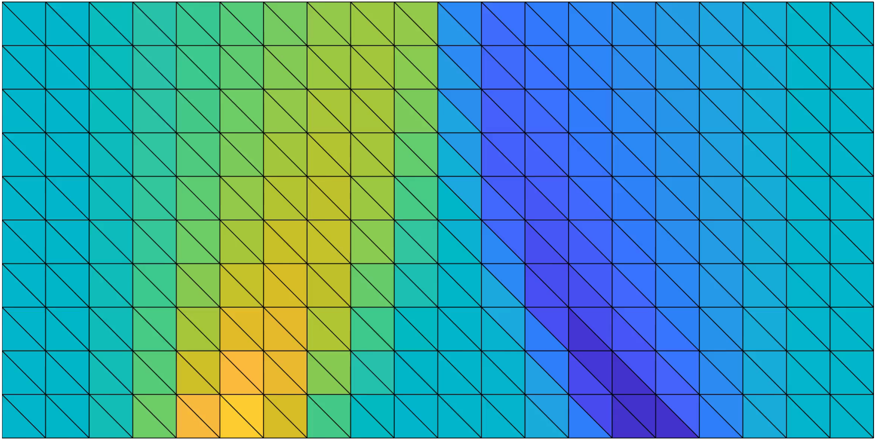


$p = 1$ (*left*) and $p = 4$ (*right*) tracking solution

Burgers' equation, shock formation and intersection



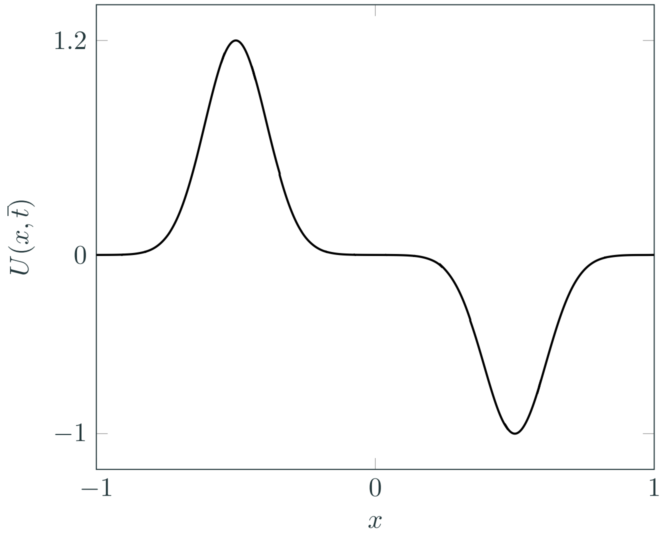
Burgers' equation, shock formation and intersection (space-time)



$p = q = 3$

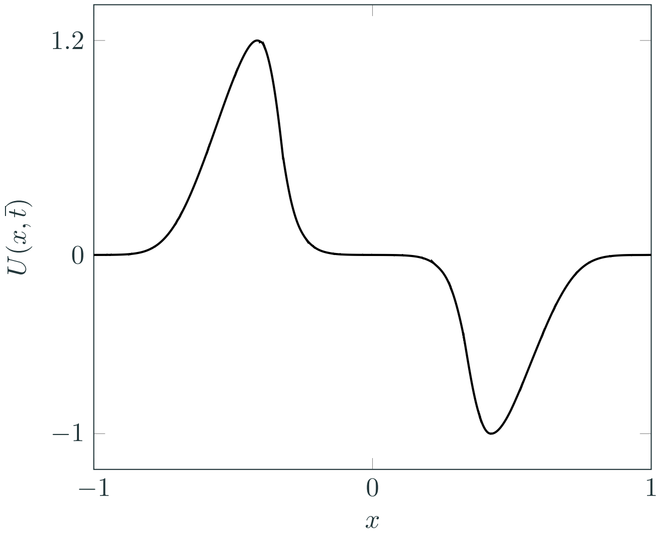
Observation: Triple point where shocks merge is tracked. Insufficient resolution to fully capture shock formation; approximate with discontinuity.

Burgers' equation, shock formation and intersection (time slices)



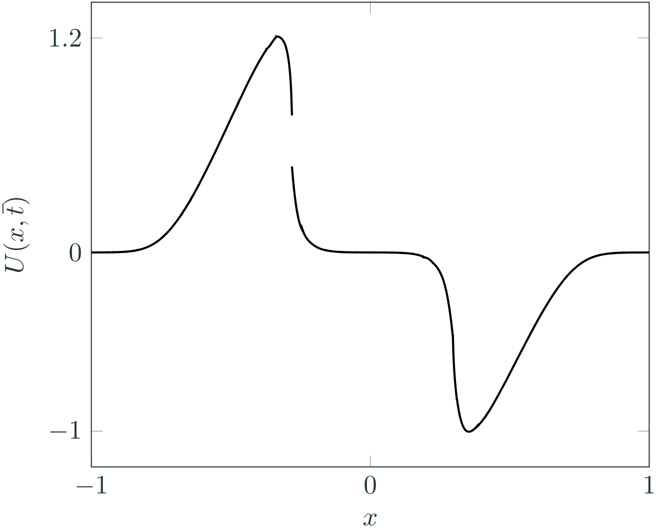
Observation: Triple point where shocks merge is tracked. Insufficient resolution to fully capture shock formation; approximate with discontinuity.

Burgers' equation, shock formation and intersection (time slices)



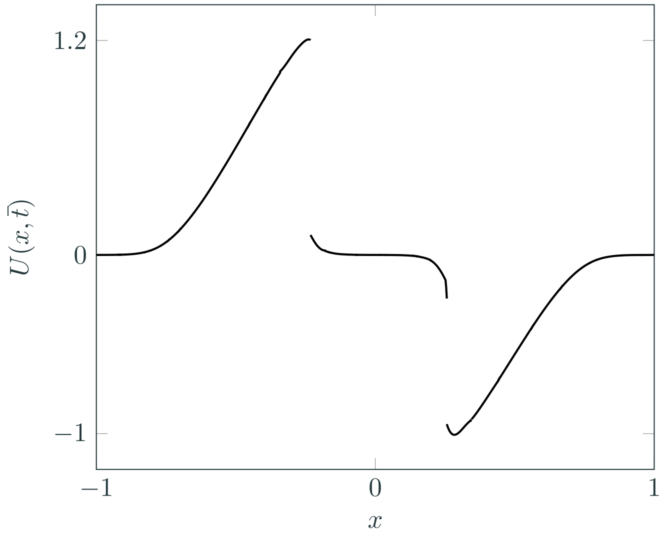
Observation: Triple point where shocks merge is tracked. Insufficient resolution to fully capture shock formation; approximate with discontinuity.

Burgers' equation, shock formation and intersection (time slices)



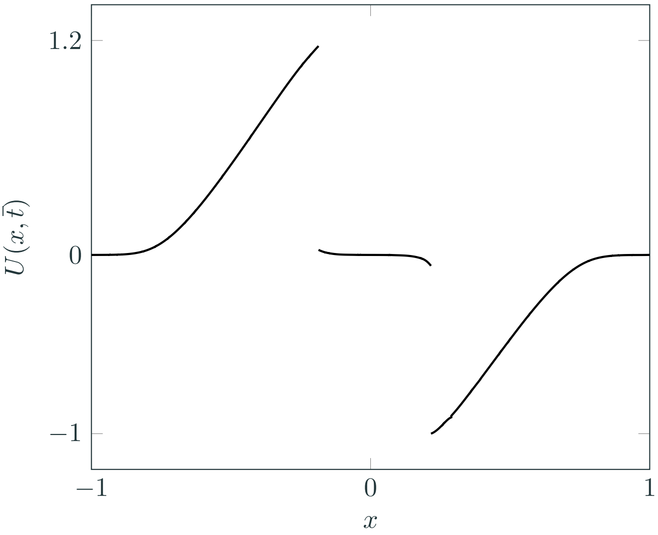
Observation: Triple point where shocks merge is tracked. Insufficient resolution to fully capture shock formation; approximate with discontinuity.

Burgers' equation, shock formation and intersection (time slices)



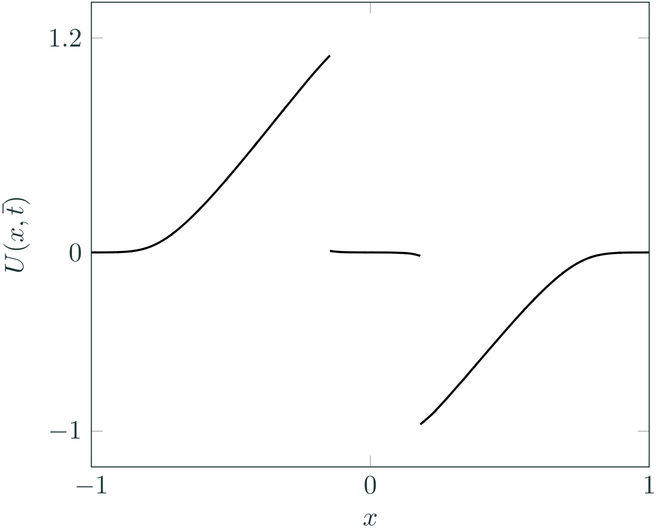
Observation: Triple point where shocks merge is tracked. Insufficient resolution to fully capture shock formation; approximate with discontinuity.

Burgers' equation, shock formation and intersection (time slices)



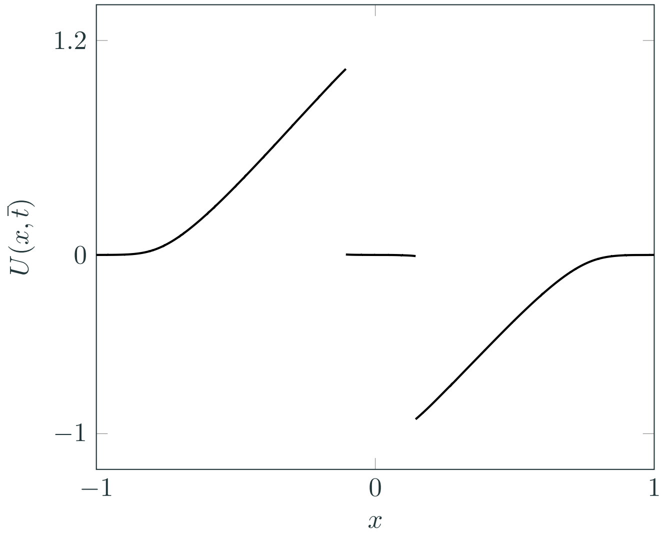
Observation: Triple point where shocks merge is tracked. Insufficient resolution to fully capture shock formation; approximate with discontinuity.

Burgers' equation, shock formation and intersection (time slices)



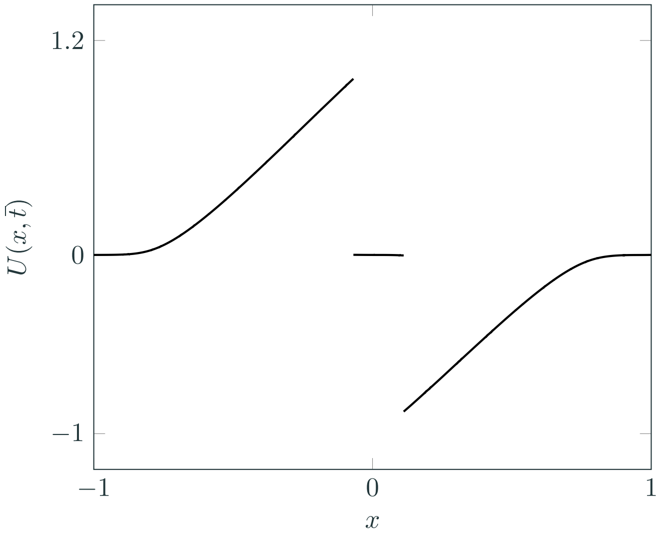
Observation: Triple point where shocks merge is tracked. Insufficient resolution to fully capture shock formation; approximate with discontinuity.

Burgers' equation, shock formation and intersection (time slices)



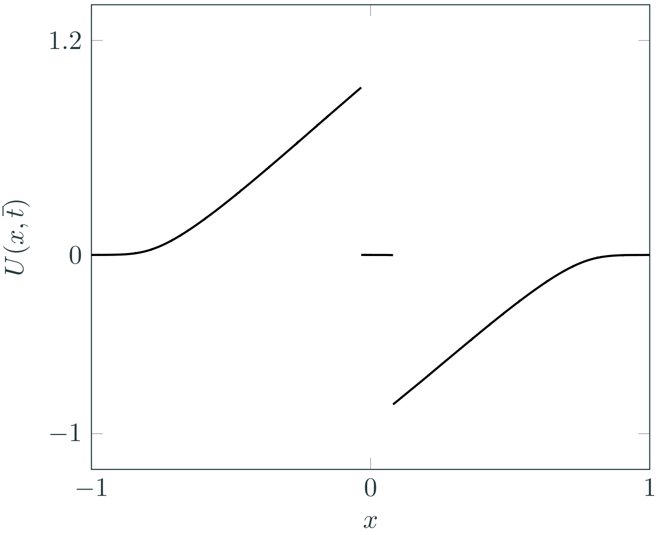
Observation: Triple point where shocks merge is tracked. Insufficient resolution to fully capture shock formation; approximate with discontinuity.

Burgers' equation, shock formation and intersection (time slices)



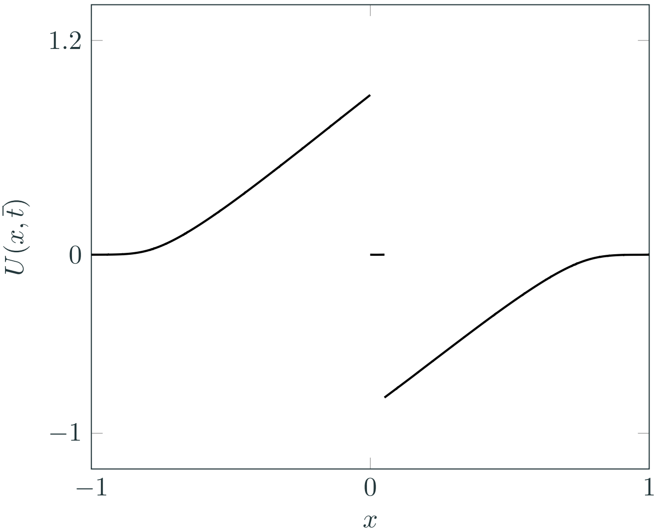
Observation: Triple point where shocks merge is tracked. Insufficient resolution to fully capture shock formation; approximate with discontinuity.

Burgers' equation, shock formation and intersection (time slices)



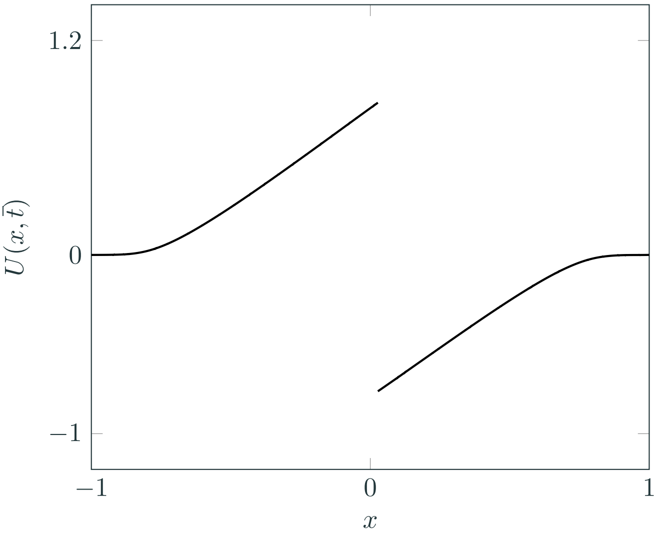
Observation: Triple point where shocks merge is tracked. Insufficient resolution to fully capture shock formation; approximate with discontinuity.

Burgers' equation, shock formation and intersection (time slices)



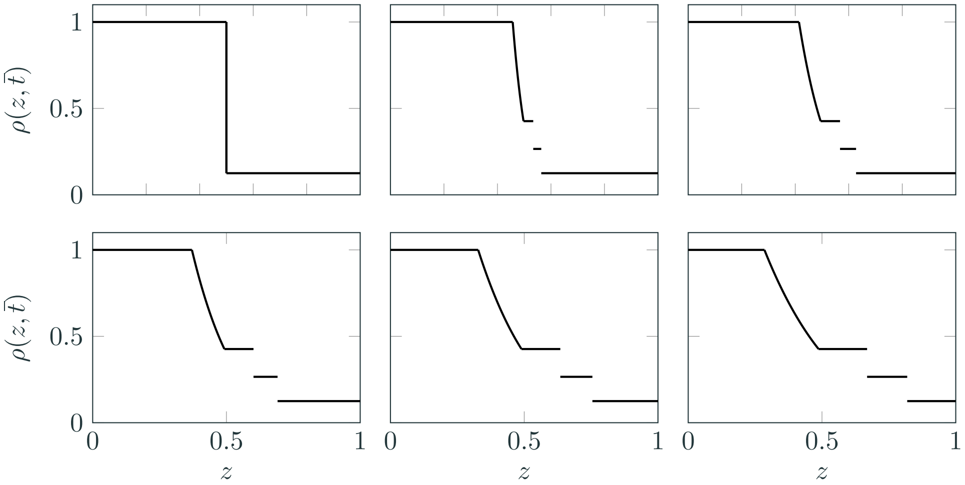
Observation: Triple point where shocks merge is tracked. Insufficient resolution to fully capture shock formation; approximate with discontinuity.

Burgers' equation, shock formation and intersection (time slices)

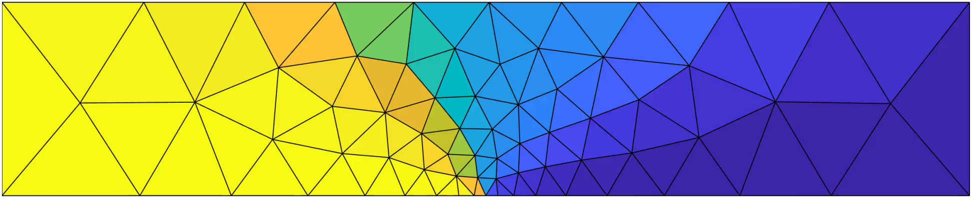


Observation: Triple point where shocks merge is tracked. Insufficient resolution to fully capture shock formation; approximate with discontinuity.

Unsteady, inviscid flow, space-time: Sod shock tube



Unsteady, inviscid flow, space-time: Sod shock tube



$p = 2,\ q = 1$

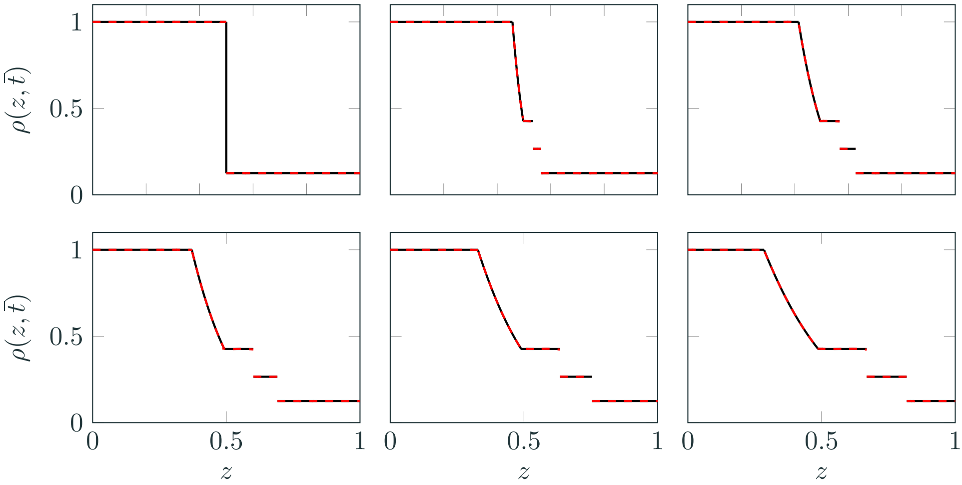
Observation: Tracks multiple features including discontinuities and derivative jumps; stronger features "easier" to track (track earlier in process).

Unsteady, inviscid flow, space-time: Sod shock tube

$p = 2$, $q = 1$

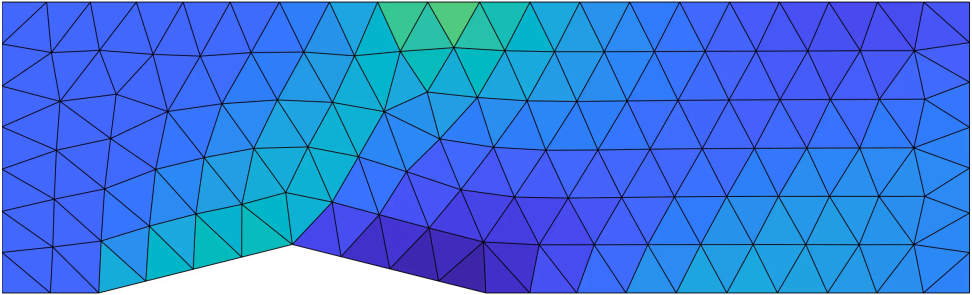
Observation: Tracks multiple features including discontinuities and derivative jumps; stronger features "easier" to track (track earlier in process).

Unsteady, inviscid flow, space-time: Sod shock tube



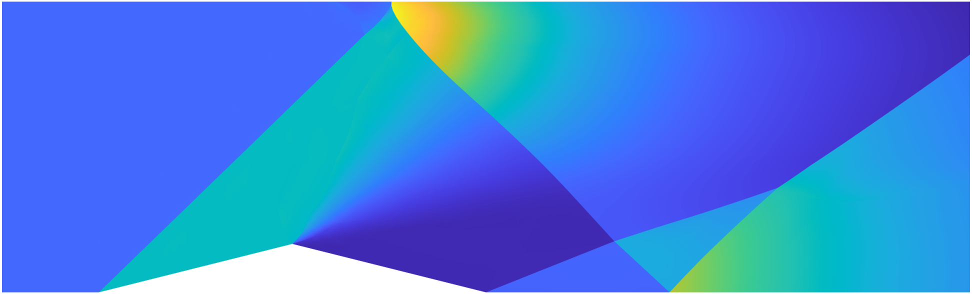
Exact (——), $p = 2$ $(q = 1)$ tracking (- - -)

Observation: Tracks multiple features including discontinuities and derivative jumps; stronger features "easier" to track (track earlier in process).



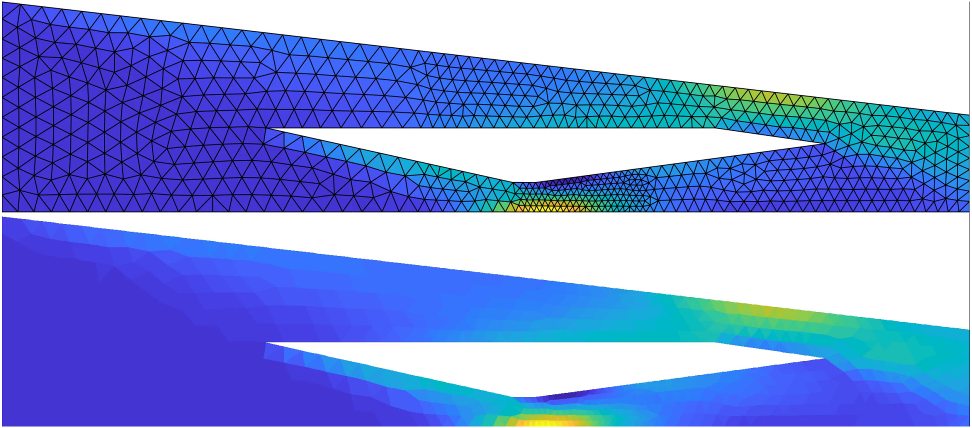
$p = q = 2$

2D Supersonic flow: $M = 2$ flow over diamond



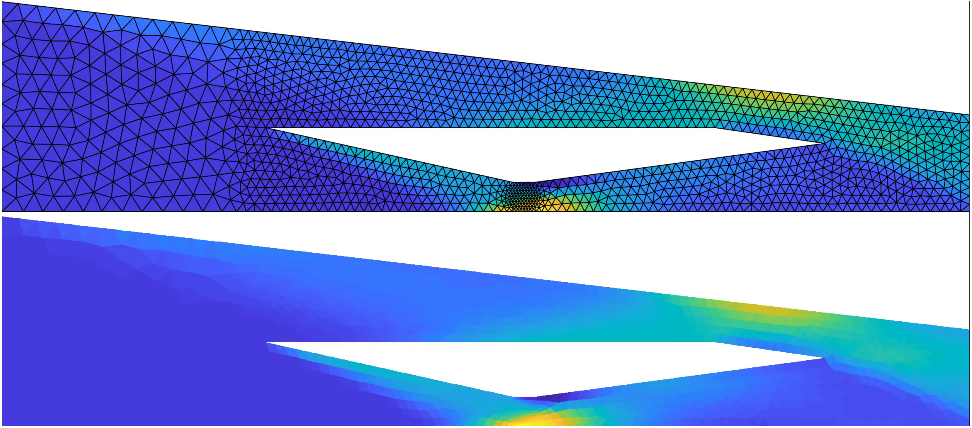
$p = q = 2$

2D Hypersonic flow: $M = 5$ flow through scramjet



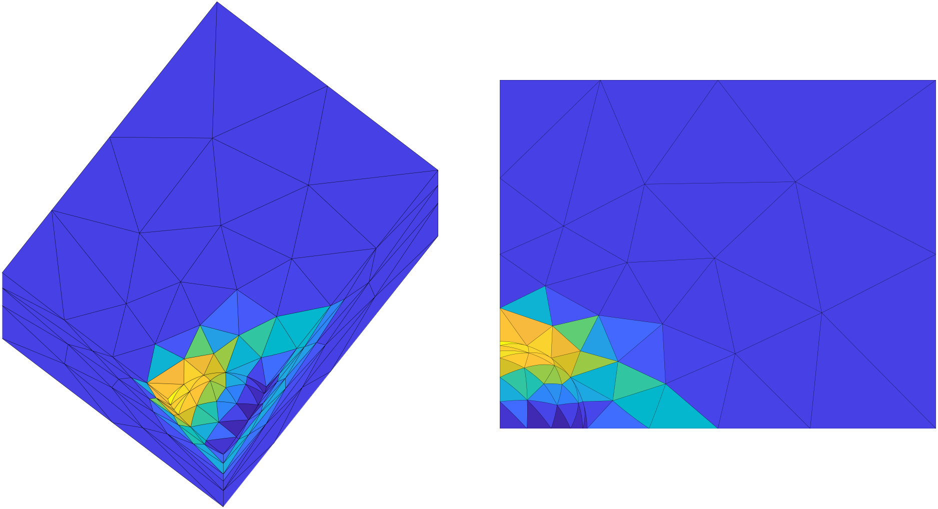
Coarse mesh, $p = q = 2$

2D Hypersonic flow: $M = 5$ flow through scramjet



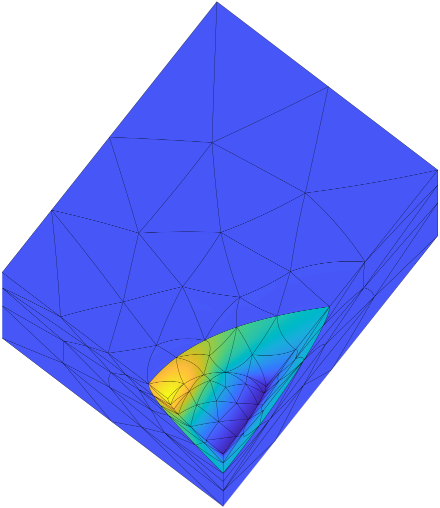
Fine mesh, $p = q = 2$

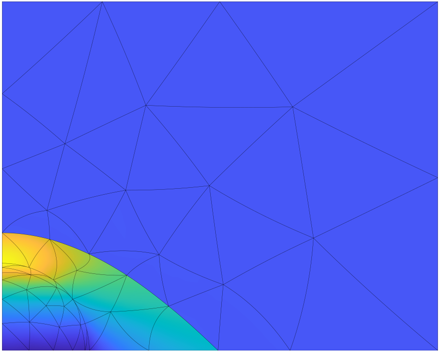
3D Supersonic flow: $M = 2$ flow over sphere



$p = q = 2$

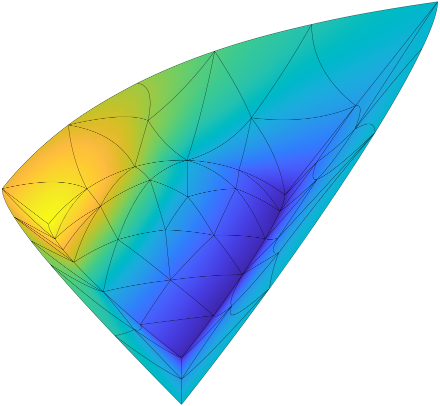
3D Supersonic flow: $M = 2$ flow over sphere

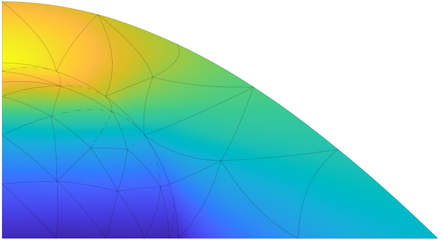




$$p = q = 2$$

3D Supersonic flow: $M = 2$ flow over sphere

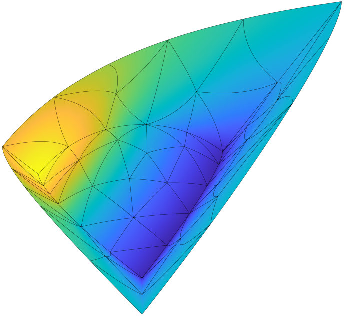


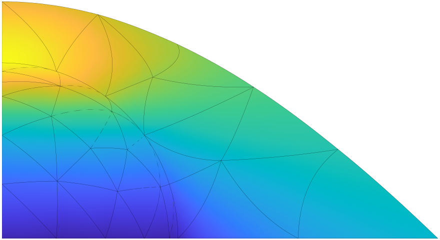


$p = q = 2$

High-order, implicit shock tracking

- **Implicit tracking:** formulate tracking as optimization problem over $(\boldsymbol{u}, \boldsymbol{x})$
- Highly *accurate solutions* on coarse meshes, *optimal convergence* rates
- High-order methods exaggerate accuracy benefits of tracking discontinuities
- Traditional barrier to tracking (explicitly meshing unknown discontinuity surface) replaced with solving constrained optimization problem





Research to make implicit tracking competitive for hypersonics

- Viscous conservation laws

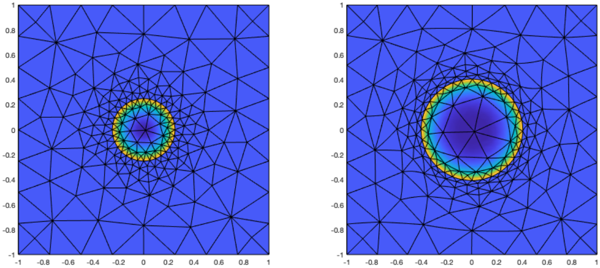
$$\begin{array}{ll} \underset{\boldsymbol{u},\boldsymbol{x}}{\text{minimize}} & \dfrac{1}{2}\left\|\boldsymbol{R}(\boldsymbol{u},\boldsymbol{x})\right\|_2^2 + \dfrac{\kappa^2}{2}\left\|\boldsymbol{R}_{\text{msh}}(\boldsymbol{x})\right\|_2^2 \\ \text{subject to} & \boldsymbol{r}(\boldsymbol{u},\boldsymbol{x}) = \boldsymbol{0} \end{array}$$

Research to make implicit tracking competitive for hypersonics

- Viscous conservation laws
- Time-dependent problems:

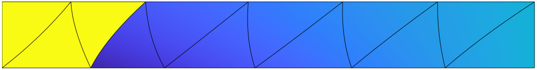
Research to make implicit tracking competitive for hypersonics

- Viscous conservation laws
- Time-dependent problems: method of lines,



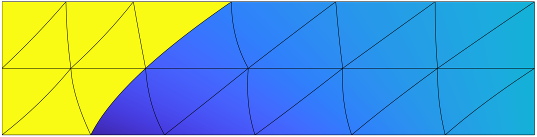
Research to make implicit tracking competitive for hypersonics

- Viscous conservation laws
- Time-dependent problems: method of lines, slab-based space-time



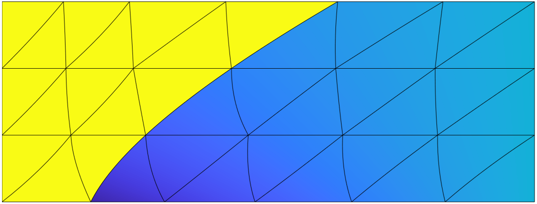
Research to make implicit tracking competitive for hypersonics

- Viscous conservation laws
- Time-dependent problems: method of lines, slab-based space-time



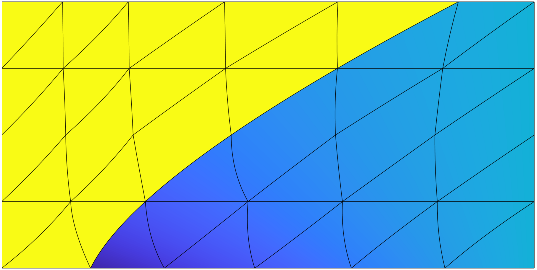
Research to make implicit tracking competitive for hypersonics

- Viscous conservation laws
- Time-dependent problems: method of lines, slab-based space-time



Research to make implicit tracking competitive for hypersonics

- Viscous conservation laws
- Time-dependent problems: method of lines, slab-based space-time



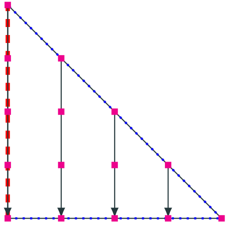
Research to make implicit tracking competitive for hypersonics

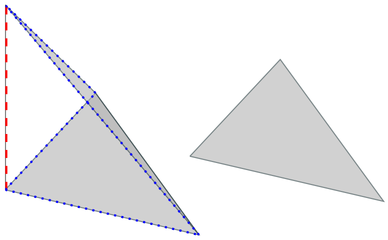
- Viscous conservation laws
- Time-dependent problems: method of lines, slab-based space-time
- Scalable linear system solver

$$\begin{bmatrix} \boldsymbol{B}_{\boldsymbol{uu}}(\boldsymbol{z}_k, \hat{\boldsymbol{\lambda}}(\boldsymbol{z}_k)) & \boldsymbol{B}_{\boldsymbol{ux}}(\boldsymbol{z}_k, \hat{\boldsymbol{\lambda}}(\boldsymbol{z}_k)) & \boldsymbol{J}_{\boldsymbol{u}}(\boldsymbol{z}_k)^T \\ \boldsymbol{B}_{\boldsymbol{ux}}(\boldsymbol{z}_k, \hat{\boldsymbol{\lambda}}(\boldsymbol{z}_k))^T & \boldsymbol{B}_{\boldsymbol{xx}}(\boldsymbol{z}_k, \hat{\boldsymbol{\lambda}}(\boldsymbol{z}_k)) & \boldsymbol{J}_{\boldsymbol{x}}(\boldsymbol{z}_k)^T \\ \boldsymbol{J}_{\boldsymbol{u}}(\boldsymbol{z}_k) & \boldsymbol{J}_{\boldsymbol{x}}(\boldsymbol{z}_k) & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \Delta \boldsymbol{u}_k \\ \Delta \boldsymbol{x}_k \\ \boldsymbol{\eta}_k \end{bmatrix} = - \begin{bmatrix} \boldsymbol{g}_{\boldsymbol{u}}(\boldsymbol{z}_k) \\ \boldsymbol{g}_{\boldsymbol{x}}(\boldsymbol{z}_k) \\ \boldsymbol{r}(\boldsymbol{z}_k) \end{bmatrix}$$

Research to make implicit tracking competitive for hypersonics

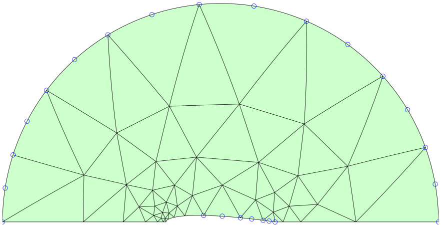
- Viscous conservation laws
- Time-dependent problems: method of lines, slab-based space-time
- Scalable linear system solver
- Edge collapses for hypercube elements; degenerate elements

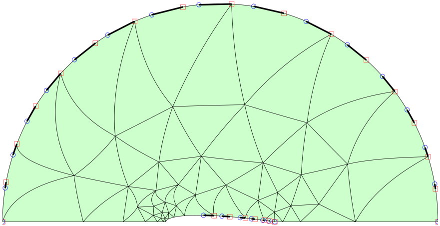




Research to make implicit tracking competitive for hypersonics

- Viscous conservation laws
- Time-dependent problems: method of lines, slab-based space-time
- Scalable linear system solver
- Edge collapses for hypercube elements; degenerate elements
- Automatically slide nodes along curved boundaries from CAD or mesh





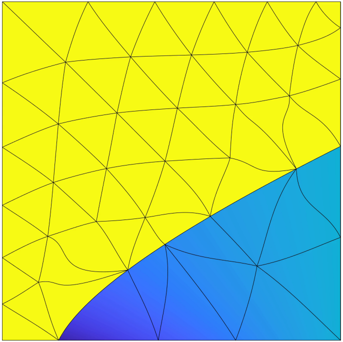
Research to make implicit tracking competitive for hypersonics

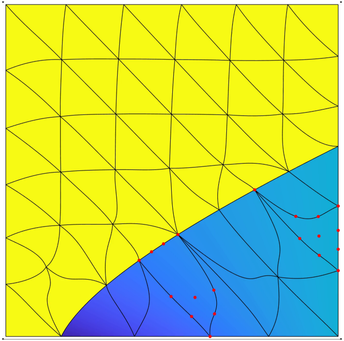
- Viscous conservation laws
- Time-dependent problems: method of lines, slab-based space-time
- Scalable linear system solver
- Edge collapses for hypercube elements; degenerate elements
- Automatically slide nodes along curved boundaries from CAD or mesh
- Integrate approach with second-order finite volume method

$$\begin{aligned} \underset{\boldsymbol{u},\boldsymbol{x}}{\text{minimize}} \quad & \frac{1}{2}\left\|\boldsymbol{R}(\boldsymbol{u},\boldsymbol{x})\right\|_2^2 + \frac{\kappa^2}{2}\left\|\boldsymbol{R}_{\text{msh}}(\boldsymbol{x})\right\|_2^2 \\ \text{subject to} \quad & \boldsymbol{r}(\boldsymbol{u},\boldsymbol{x}) = \boldsymbol{0} \end{aligned}$$

Research to make implicit tracking competitive for hypersonics

- Viscous conservation laws
- Time-dependent problems: method of lines, slab-based space-time
- Scalable linear system solver
- Edge collapses for hypercube elements; degenerate elements
- Automatically slide nodes along curved boundaries from CAD or mesh
- Integrate approach with second-order finite volume method
- Hybrid shock tracking/capturing approach (e.g., only track bow shock)





Perspective: artificial viscosity vs. implicit tracking

	artificial viscosity	**implicit tracking**[2]
Strong shocks	control	easier
Complex shock structures	control	harder
Nonlinear solver	PTC/Newton	SQP
Parameter tweaking	formulation	solver
Linearization	$\partial_{\boldsymbol{u}}, \partial_{\nu}$	$\partial_{\boldsymbol{u}}, \partial_{\boldsymbol{x}}$
Mesh generation	control	easier
Geometry	only high-order mesh	geometry required
Linear solver	ILU+GMRES	?
Cost per element	control	higher
Cost per iteration	control	higher
Mesh fineness	control	coarser
Overall cost	control	?

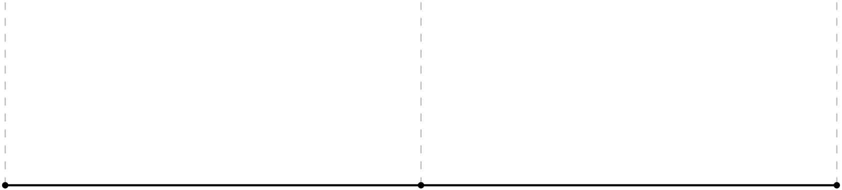
[2] e.g., HOIST, MDG-ICE

Corrigan, A., Kercher, A., and Kessler, D. (2019).
A moving discontinuous Galerkin finite element method for flows with interfaces.
International Journal for Numerical Methods in Fluids, 89(9):362–406.

Zahr, M. and Persson, P.-O. (2018).
An optimization-based approach for high-order accurate discretization of conservation laws with discontinuous solutions.
Journal of Computational Physics, 365:105–134.

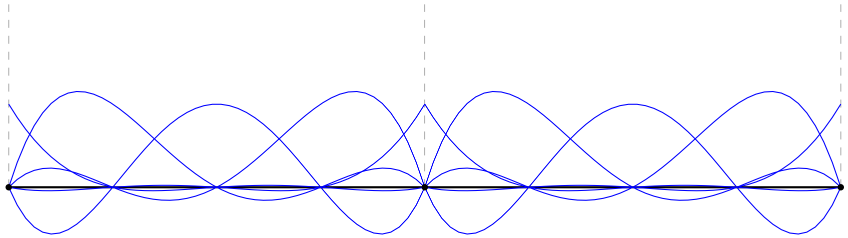
Zahr, M., Shi, A., and Persson, P.-O. (2020).
Implicit shock tracking using an optimization-based discontinuous galerkin method.
Journal of Computational Physics.

Numerical methods for resolving shocks



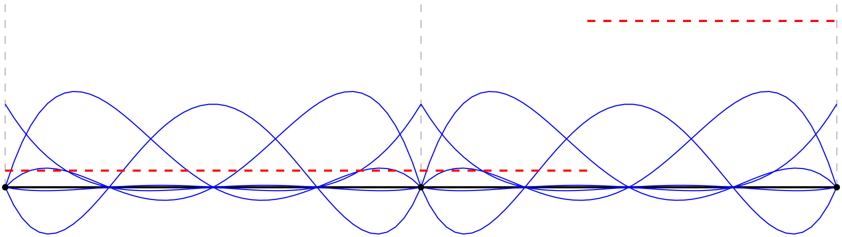
Fundamental issue: approximate discontinuity with polynomial basis

Numerical methods for resolving shocks



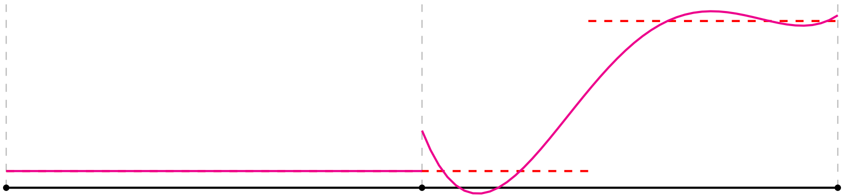
<u>Fundamental issue</u>: approximate discontinuity with polynomial basis

Numerical methods for resolving shocks



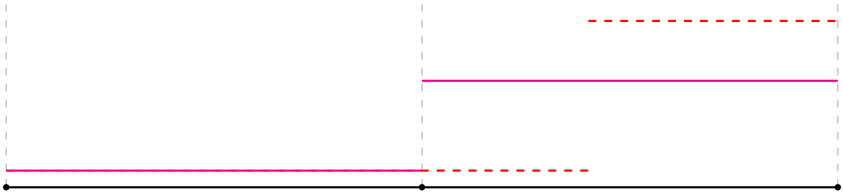
Fundamental issue: approximate discontinuity with polynomial basis

Numerical methods for resolving shocks



Fundamental issue: approximate discontinuity with polynomial basis

Numerical methods for resolving shocks

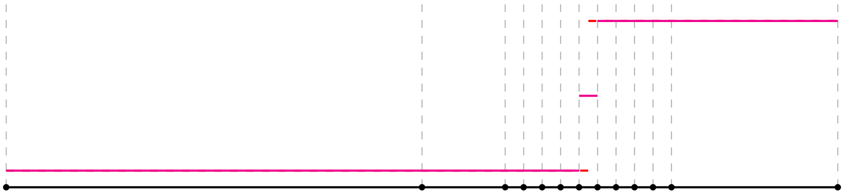


Fundamental issue: approximate discontinuity with polynomial basis

Exising solutions: **limiting**, artificial viscosity

Drawbacks: order reduction, local refinement

Numerical methods for resolving shocks

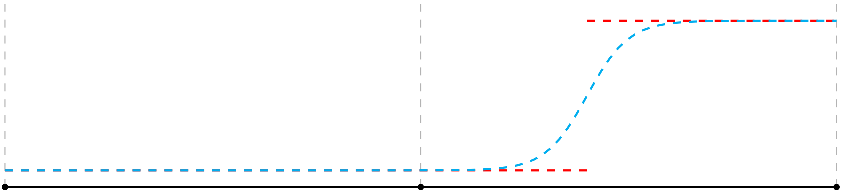


Fundamental issue: approximate discontinuity with polynomial basis

Exising solutions: **limiting**, artificial viscosity

Drawbacks: order reduction, local refinement

Numerical methods for resolving shocks

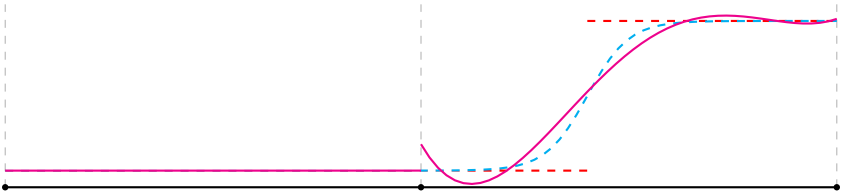


Fundamental issue: approximate discontinuity with polynomial basis

Exising solutions: limiting, **artificial viscosity**

Drawbacks: order reduction, local refinement

Numerical methods for resolving shocks

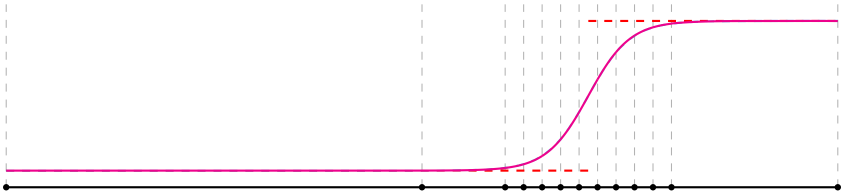


Fundamental issue: approximate discontinuity with polynomial basis

Exising solutions: limiting, **artificial viscosity**

Drawbacks: order reduction, local refinement

Numerical methods for resolving shocks

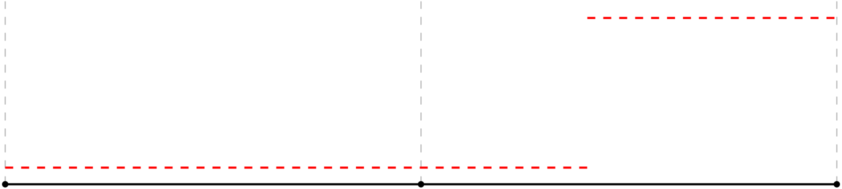


Fundamental issue: approximate discontinuity with polynomial basis

Exising solutions: limiting, **artificial viscosity**

Drawbacks: order reduction, local refinement

Numerical methods for resolving shocks



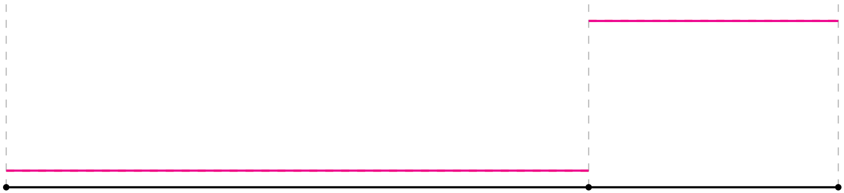
Fundamental issue: approximate discontinuity with polynomial basis

Exising solutions: limiting, artificial viscosity

Drawbacks: order reduction, local refinement

Shock tracking/fitting: align features of solution basis with features in the solution using optimization formulation and solver

Numerical methods for resolving shocks



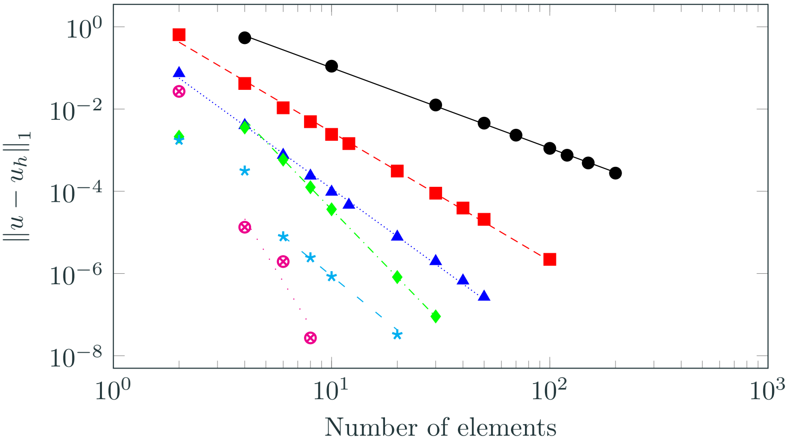
Fundamental issue: approximate discontinuity with polynomial basis

Exising solutions: limiting, artificial viscosity

Drawbacks: order reduction, local refinement

Shock tracking/fitting: align features of solution basis with features in the solution using optimization formulation and solver

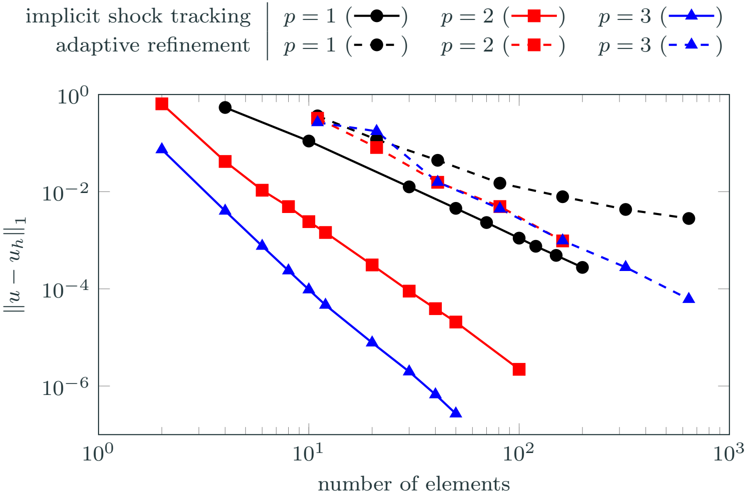
Why tracking: Recover optimal $\mathcal{O}(h^{p+1})$ convergence rates



Convergence of implicit shock tracking (Burgers' equation) with polynomial degrees $p = 1$ (●), $p = 2$ (■), $p = 3$ (▲), $p = 4$ (♦), $p = 5$ (∗), $p = 6$ (⊗).

Key observation: Optimal convergence rates ($\mathcal{O}(h^{p+1})$) attainable, even for discontinuous solutions.

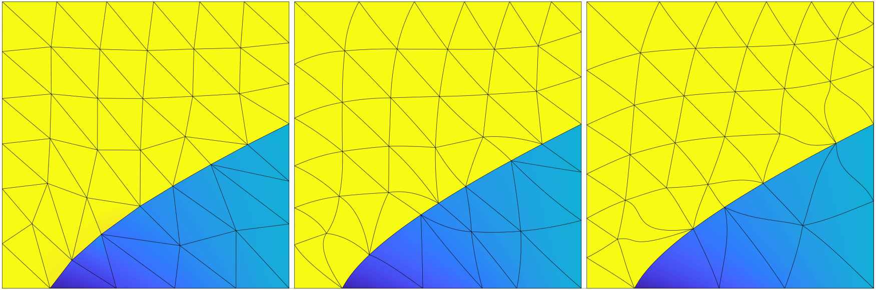
Why high-order tracking: Benefits more dramatic than low-order



Convergence of implicit shock tracking (Burgers' equation): implicit shock tracking (solid) vs. adaptive mesh refinement (dashed).

Key observation: Accuracy improvement of tracking approach relative to (specialized) adaptive mesh refinement is more exaggerated for high-order approximations: $\mathcal{O}(10^1)$ for $p = 1$ and $\mathcal{O}(10^6)$ for $p = 3$.

Burgers' equation, accelerating shock



$p = q = 1$ $\quad$ $p = q = 2$ $\quad$ $p = q = 3$

Burgers' equation, accelerating shock: h convergence

Convergence of solution error (E_u) along line $x = 0.8$ and shock surface error (E_Γ)

p	q	$\lvert\mathcal{E}_h\rvert$	h	E_u	$m(E_u)$	E_Γ	$m(E_\Gamma)$
1	1	38	1.45e-01	2.72e-02	-	2.32e-03	-
1	1	152	7.25e-02	7.18e-03	1.92	1.09e-03	1.09
1	1	598	3.66e-02	1.91e-03	1.93	1.93e-04	2.53
1	1	2392	1.83e-02	4.69e-04	2.03	3.92e-05	2.30
2	2	38	1.45e-01	5.68e-03	-	4.83e-05	-
2	2	152	7.25e-02	9.64e-05	5.88	2.70e-07	7.48
2	2	608	3.63e-02	6.36e-06	3.92	1.20e-08	4.49
2	2	2432	1.81e-02	8.66e-07	2.88	7.70e-10	3.96
3	3	32	1.58e-01	1.57e-03	-	2.06e-05	-
3	3	128	7.91e-02	1.62e-05	6.60	3.37e-07	5.93
3	3	512	3.95e-02	4.37e-07	5.21	5.90e-09	5.84
3	3	2040	1.98e-02	3.31e-08	3.73	1.87e-10	5.00

Observation: Optimal convergence rates ($\mathcal{O}(h^{p+1})$) obtained for solution error; faster rates obtained for shock surface.