Optimization-Based Simulation of High-Speed Flows

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

Simulation-Based Optimization with Applications
ICERM, Brown University, Providence, RI
April 13–17, 2026

Students: Akshay Thakur, Huijing Dong
AFOSR FA9550-20-1-0236, FA9550-22-1-0002, FA9550-22-1-0004
ONR N00014-22-1-2299
NSF CBET-2338843

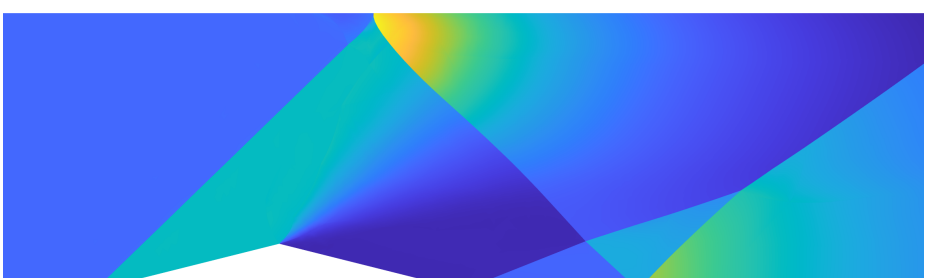
Overview

Introduction

What is shock tracking/fitting?

Shock tracking/fitting: Numerical methods that align the faces of the computational mesh with discontinuities (or steep gradients) in the solution to represent them perfectly with inter-element jumps.

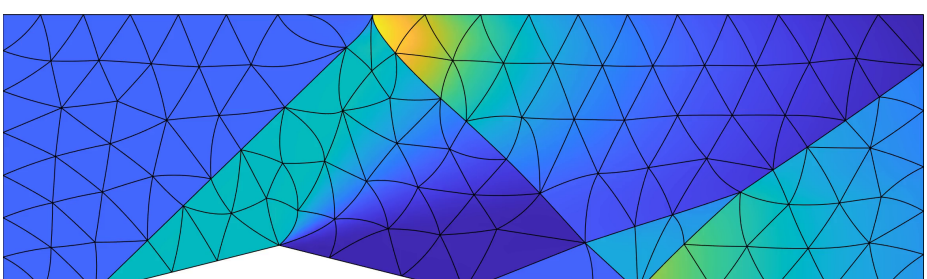
- High accuracy per degree of freedom
- Do not require additional stabilization near discontinuities (e.g., artificial viscosity, limiting)



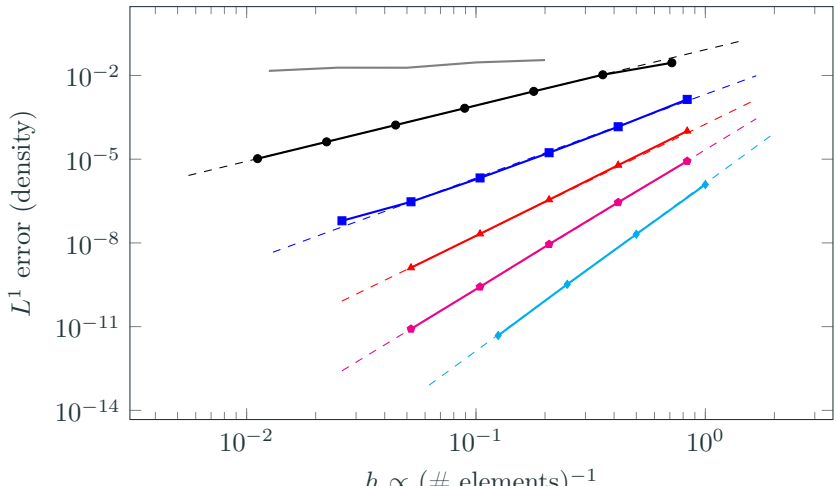
What is shock tracking/fitting?

Shock tracking/fitting: Numerical methods that align the faces of the computational mesh with discontinuities (or steep gradients) in the solution to represent them perfectly with inter-element jumps.

- High accuracy per degree of freedom
- Do not require additional stabilization near discontinuities (e.g., artificial viscosity, limiting)



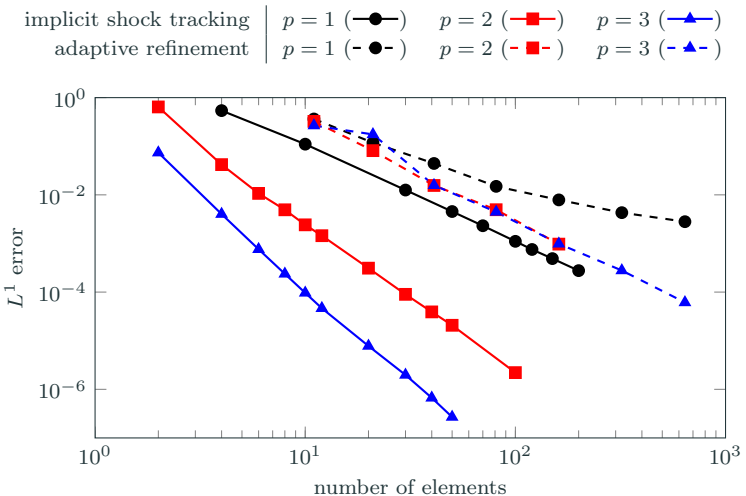
Why tracking: Recover optimal $\mathcal{O}(h^{p+1})$ convergence rates



Convergence of shock capturing $p = 4$ (——) vs. implicit shock tracking (inviscid nozzle flow) with polynomial degrees $p = 1$ (—●—), $p = 2$ (—■—), $p = 3$ (—▲—), $p = 4$ (—●—), $p = 5$ (—◆—); dashed line indicates optimal convergence rate ($\mathcal{O}(h^{p+1})$).

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

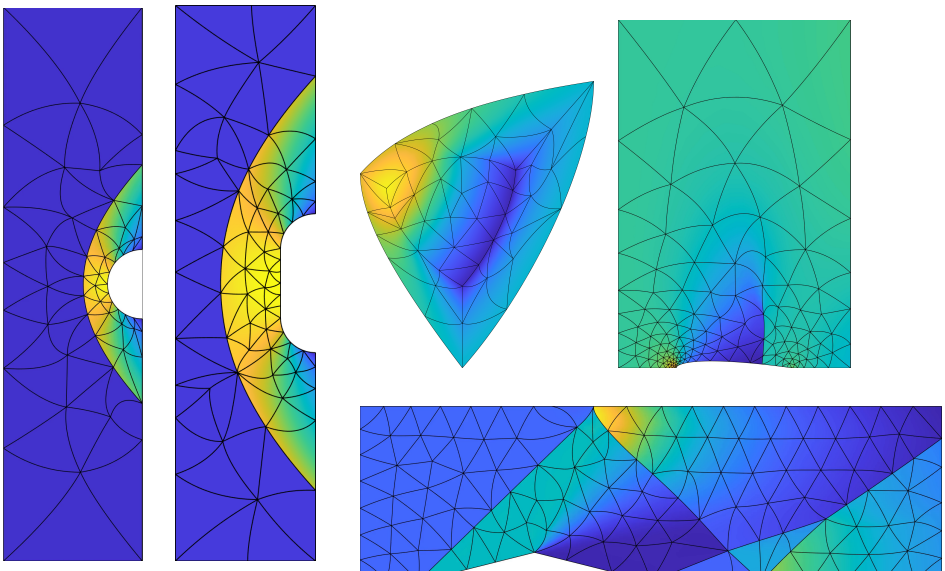
Why high-order tracking: Benefits more dramatic than low-order



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

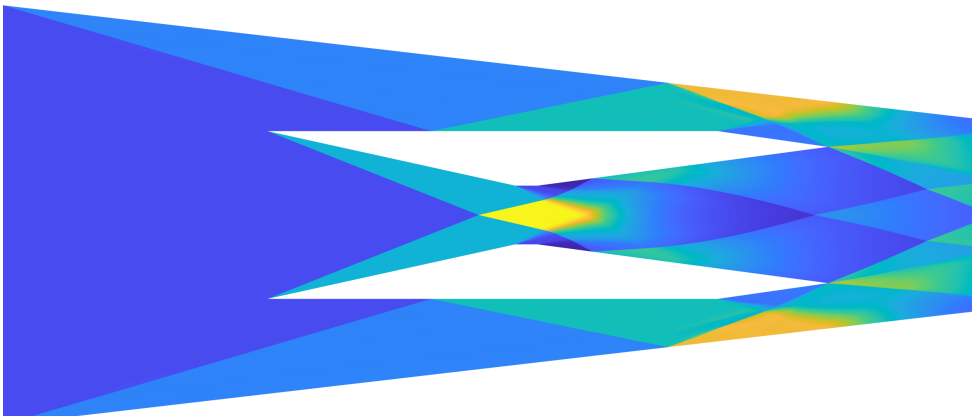
Key observation: Accuracy improvement of tracking approach relative to (specialized) adaptive mesh refinement is more exaggerated for high-order approximations: $\mathcal{O}(10^1)$ for $p = 1$ and $\mathcal{O}(10^6)$ for $p = 3$.

Why high-order tracking: Accurate solutions on coarse meshes

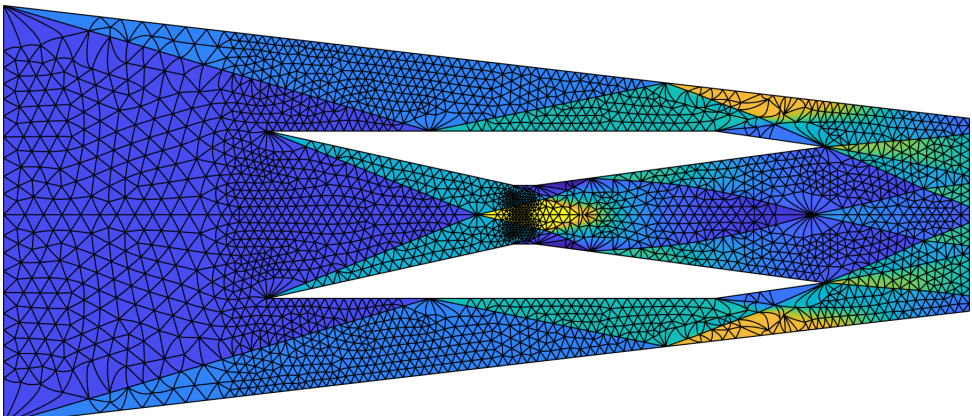


Key observation: High-order tracking enables accurate resolution of high-speed flows on traditionally coarse grids (high accuracy per degree of freedom)

Why not tracking: Difficult for complex discontinuity surfaces



Why not tracking: Difficult for complex discontinuity surfaces

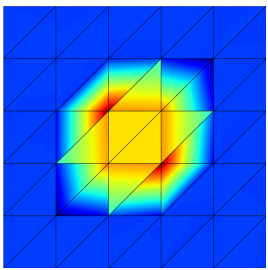


Implicit shock tracking

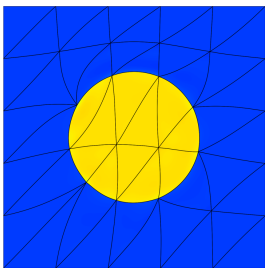
Aims to overcome the difficulty of explicitly meshing the unknown shock surface, e.g., HOIST [Zahr, Persson; 2018], MDG-ICE [Corrigan, Kercher, Kessler; 2019]

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], [Huang, Zahr; 2022]

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:

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

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)

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

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

$$\begin{bmatrix} \boldsymbol{B_{uu}}(\boldsymbol{z}_k, \hat{\boldsymbol{\lambda}}(\boldsymbol{z}_k)) & \boldsymbol{B_{ux}}(\boldsymbol{z}_k, \hat{\boldsymbol{\lambda}}(\boldsymbol{z}_k)) & \boldsymbol{J_u}(\boldsymbol{z}_k)^T \\ \boldsymbol{B_{ux}}(\boldsymbol{z}_k, \hat{\boldsymbol{\lambda}}(\boldsymbol{z}_k))^T & \boldsymbol{B_{xx}}(\boldsymbol{z}_k, \hat{\boldsymbol{\lambda}}(\boldsymbol{z}_k)) & \boldsymbol{J_x}(\boldsymbol{z}_k)^T \\ \boldsymbol{J_u}(\boldsymbol{z}_k) & \boldsymbol{J_x}(\boldsymbol{z}_k) & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \Delta \boldsymbol{u}_k \\ \Delta \boldsymbol{x}_k \\ \boldsymbol{\eta}_k \end{bmatrix} = - \begin{bmatrix} \boldsymbol{g_u}(\boldsymbol{z}_k) \\ \boldsymbol{g_x}(\boldsymbol{z}_k) \\ \boldsymbol{r}(\boldsymbol{z}_k) \end{bmatrix},$$

where

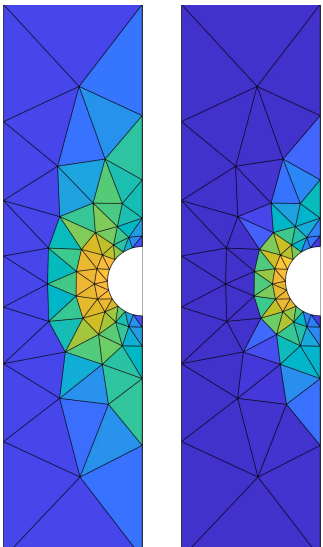
$$\boldsymbol{g_u}(\boldsymbol{z}) = \frac{\partial f}{\partial \boldsymbol{u}}(\boldsymbol{z})^T, \quad \boldsymbol{J_u}(\boldsymbol{z}) = \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{z}), \quad \boldsymbol{g_x}(\boldsymbol{z}) = \frac{\partial f}{\partial \boldsymbol{x}}(\boldsymbol{z})^T, \quad \boldsymbol{J_x}(\boldsymbol{z}) = \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{x}}(\boldsymbol{z}),$$

the approximate Hessian of the Lagrangian is taken as

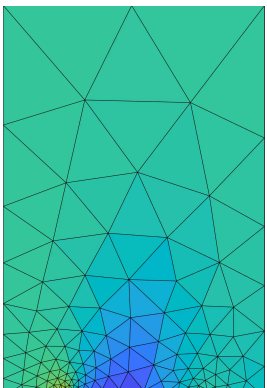
$$\begin{aligned} \boldsymbol{B_{uu}}(\boldsymbol{z}, \boldsymbol{\lambda}) &= \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{u}}(\boldsymbol{z})^T \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{u}}(\boldsymbol{z}), \quad \boldsymbol{B_{ux}}(\boldsymbol{z}, \boldsymbol{\lambda}) = \frac{\partial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

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

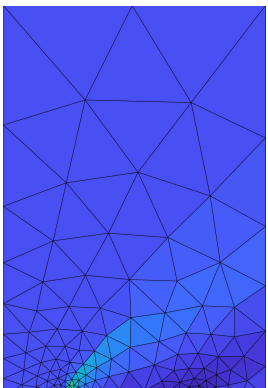
Implicit shock tracking for simple 2D compressible flows



$M = 2$ $M = 3$



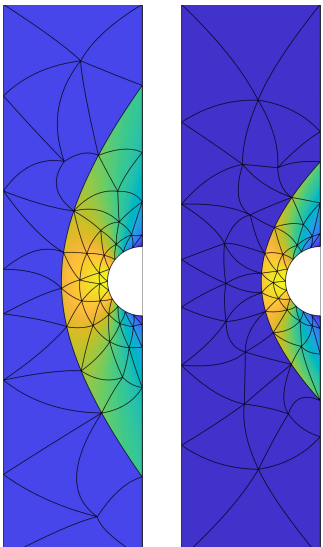
$M = 0.8$



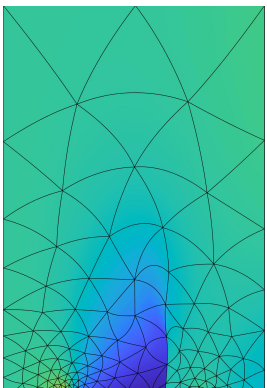
$M = 2$

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).

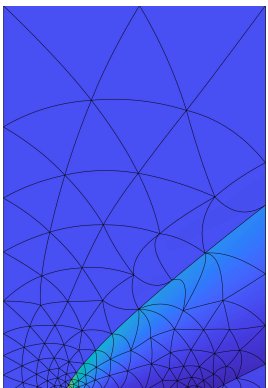
Implicit shock tracking for simple 2D compressible flows



$M = 2$ $M = 3$



$M = 0.8$



$M = 2$

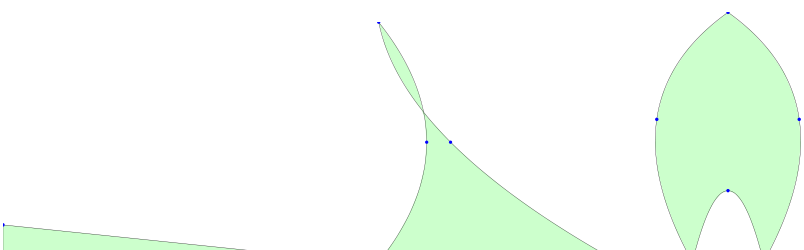
Observation: Quickly tracks bow shocks, shocks attached to curved boundaries, and secondary shocks; high-order elements curve to approximate curvature in shock surface; high-quality solutions on coarse high-order meshes ($\mathcal{O}(100)$ elements).

Practical solver for IST

Practical considerations: element collapse

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

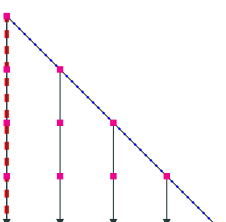
- Tag elements for removal based on volume, quality, edge length

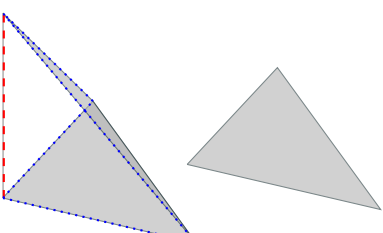


Practical considerations: element collapse

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

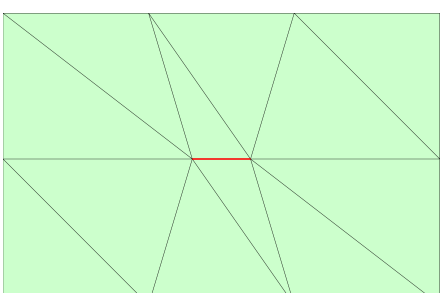


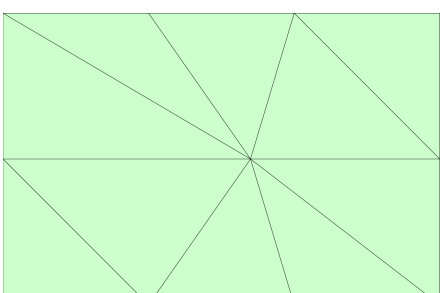


Practical considerations: element collapse

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

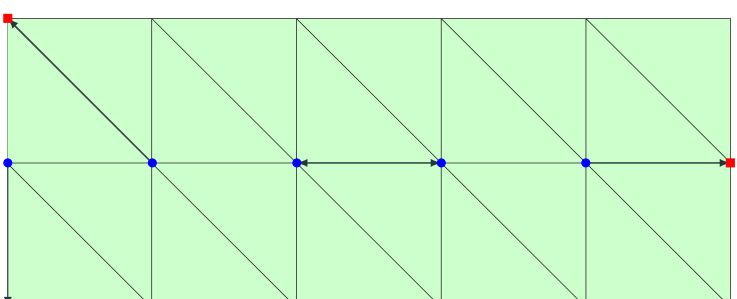




Practical considerations: element collapse

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

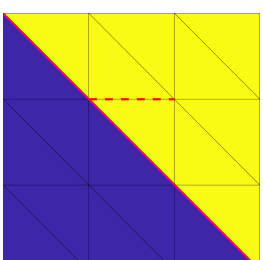
- Tag elements for removal based on volume, quality, edge length
- Collapse shortest edge: well-defined for simplices of any order in any dimension
- Remove all zero-volume elements
- Must preserve boundaries



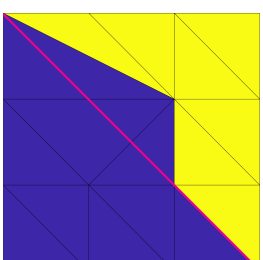
Practical considerations: element collapse

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

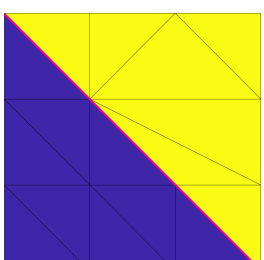
- Tag elements for removal based on volume, quality, edge length
- Collapse shortest edge: well-defined for simplices of any order in any dimension
- Remove all zero-volume elements
- Must preserve boundaries and shocks



before collapse



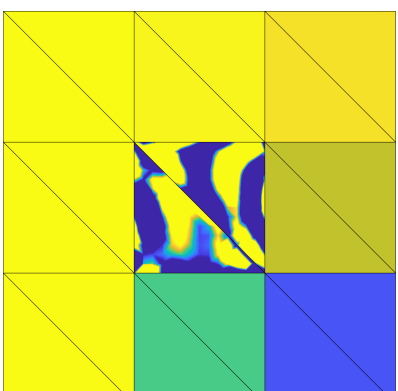
ignore shock



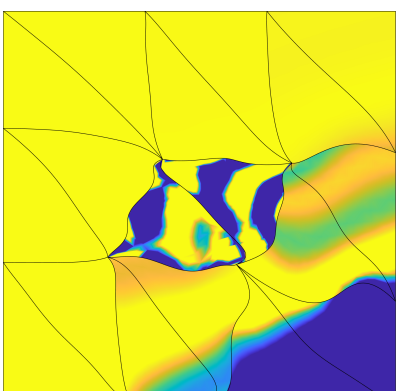
shock-aware

Practical considerations: solution re-initialization

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



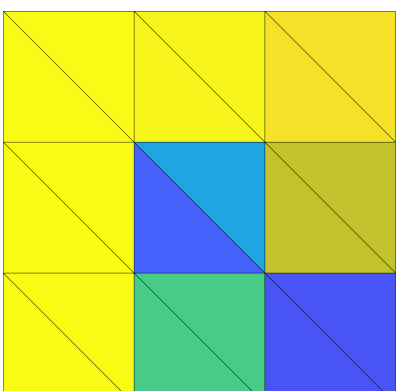
before SQP step (without re-init)



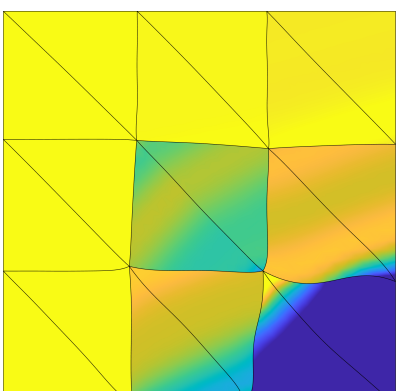
after SQP step (without re-init)

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)



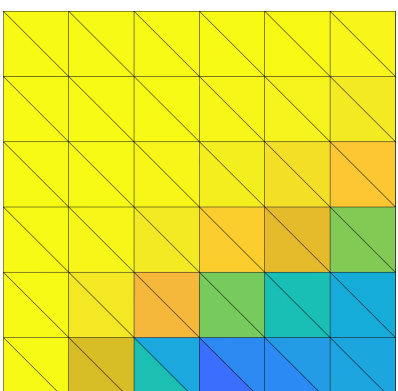
before SQP step (with re-init)



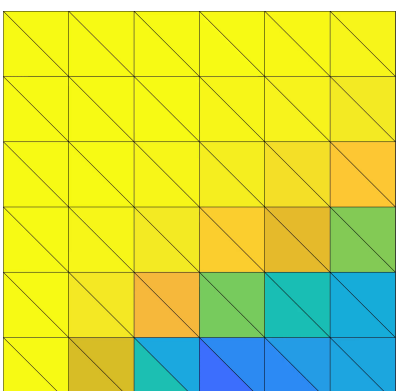
after SQP step (with re-init)

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



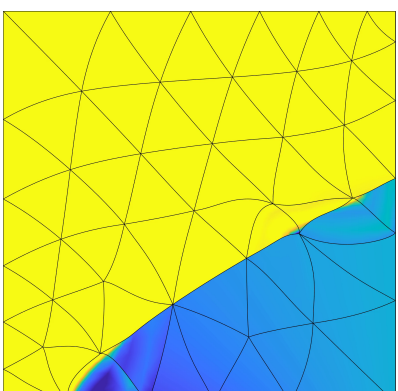
without re-initialization



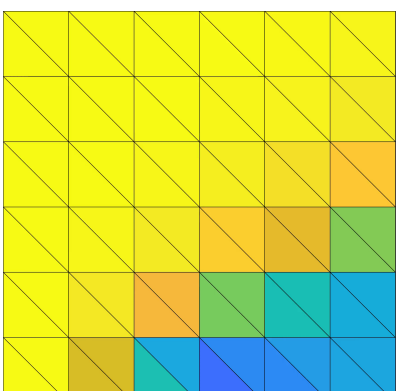
with re-initialization

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



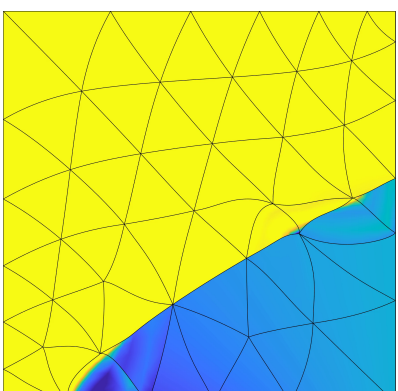
without re-initialization



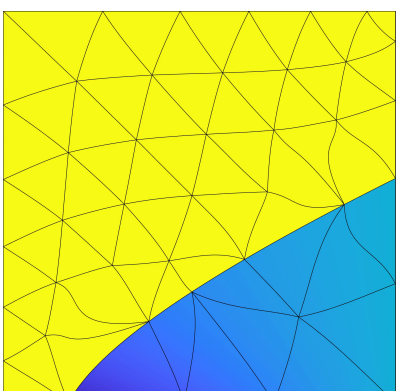
with re-initialization

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

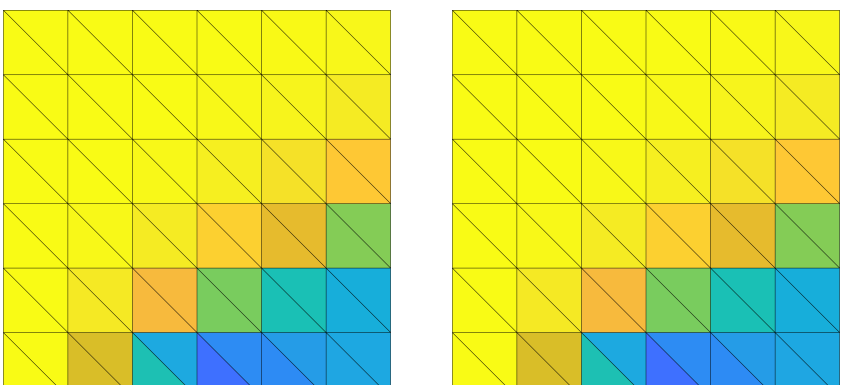


with re-initialization

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 not required

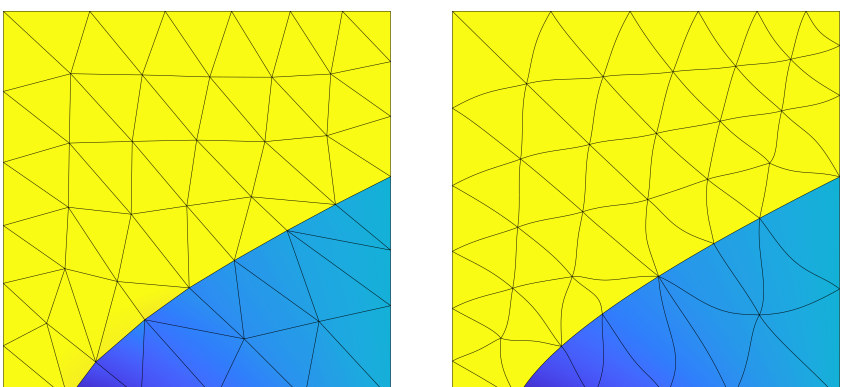


Reference mesh, $p = 0$ DG solution

Practical considerations: initialization

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

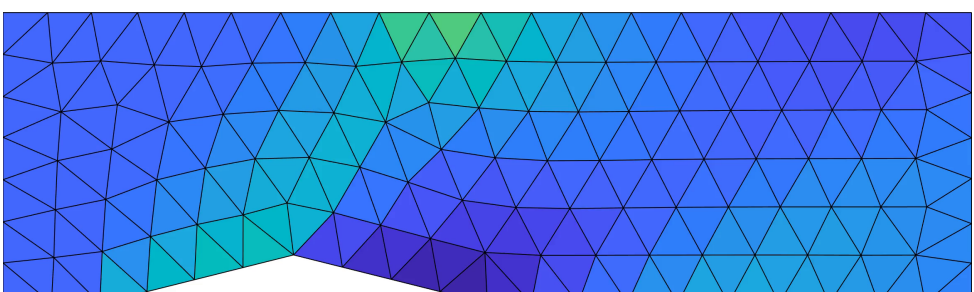
- $\boldsymbol{x}_0$: directly from mesh generation
- $\boldsymbol{u}_0$: DG($p = 0$) solution on mesh $\boldsymbol{x}_0$
- homotopy in p not required



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

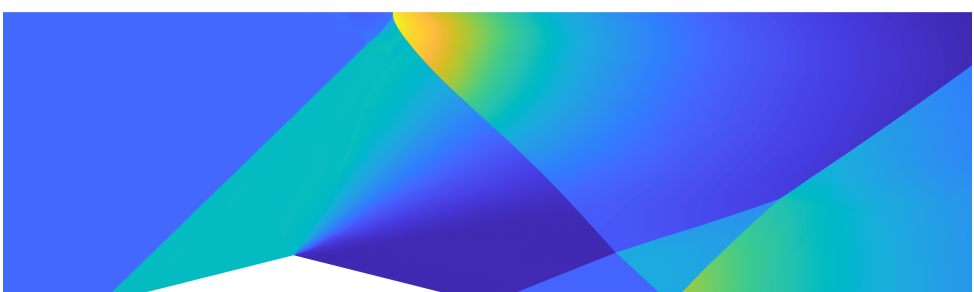
2D Supersonic flow: $M = 2$ flow over diamond

- 2D steady Euler equations (ideal gas law)
- Local space: $p = q = 2$ triangle elements
- Starting point: 220 elements (no knowledge of shocks), 1st order FV solution
- HOIST solution (250 iter): 201 elements, shock-aligned



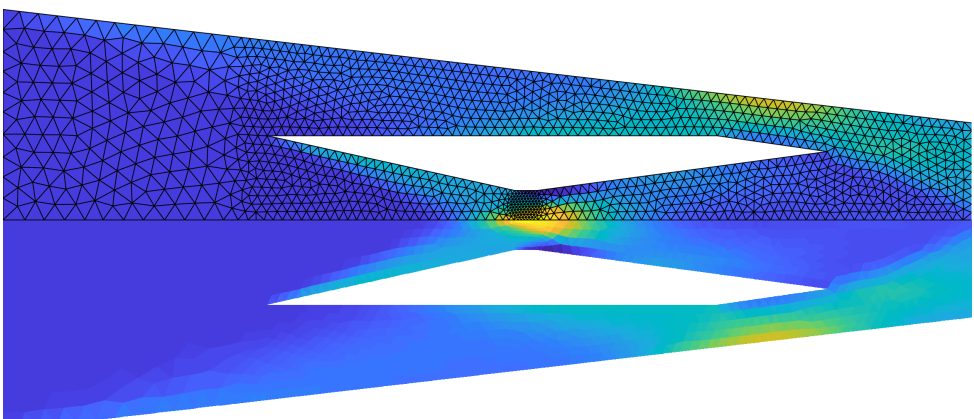
2D Supersonic flow: $M = 2$ flow over diamond

- 2D steady Euler equations (ideal gas law)
- Local space: $p = q = 2$ triangle elements
- Starting point: 220 elements (no knowledge of shocks), 1st order FV solution
- HOIST solution (250 iter): 201 elements, shock-aligned



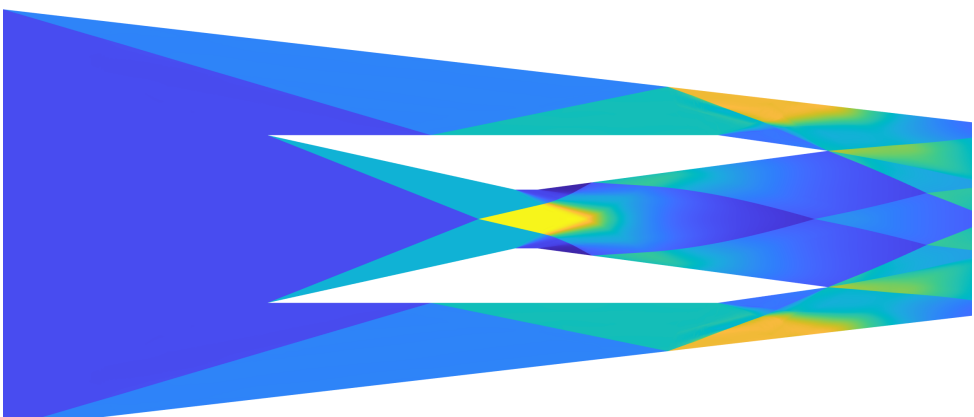
2D Hypersonic flow: $M = 5$ flow through scramjet

- 2D steady Euler equations (ideal gas law)
- Local space: $p = q = 2$ triangle elements
- Starting point: 2679 elements (no knowledge of shocks), 1st order FV solution
- HOIST solution (250 iter): 2523 elements, shock-aligned



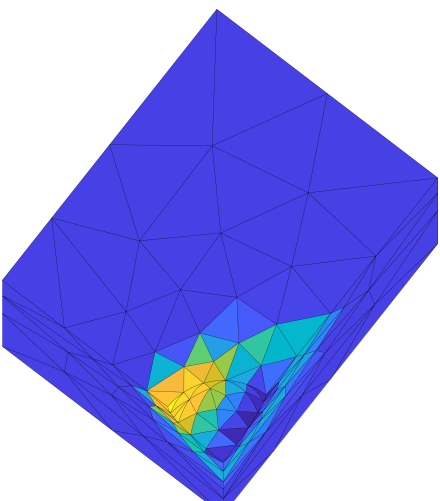
2D Hypersonic flow: $M = 5$ flow through scramjet

- 2D steady Euler equations (ideal gas law)
- Local space: $p = q = 2$ triangle elements
- Starting point: 2679 elements (no knowledge of shocks), 1st order FV solution
- HOIST solution (250 iter): 2523 elements, shock-aligned



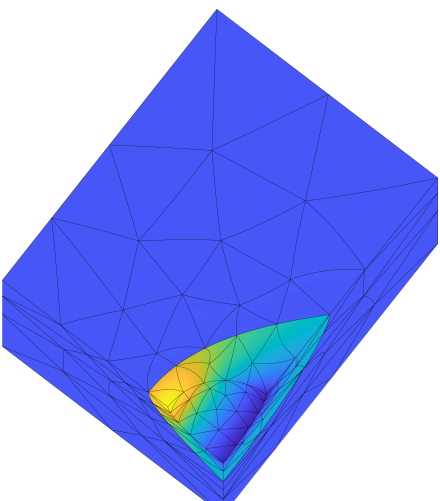
3D Supersonic flow: $M = 2$ flow over sphere

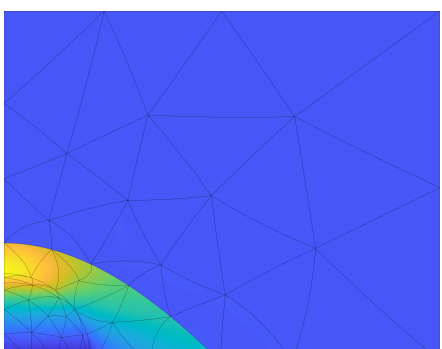
- 3D steady Euler equations (ideal gas law)
- Local space: $p = q = 2$ tetrahedral elements
- Starting point: 511 elements (no knowledge of shocks), 1st order FV solution
- HOIST solution (30 iter): 491 elements, shock-aligned



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

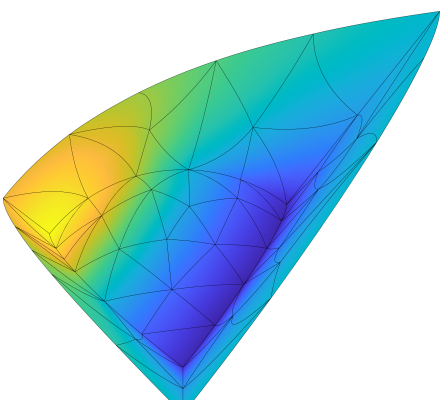
- 3D steady Euler equations (ideal gas law)
- Local space: $p = q = 2$ tetrahedral elements
- Starting point: 511 elements (no knowledge of shocks), 1st order FV solution
- HOIST solution (30 iter): 491 elements, shock-aligned

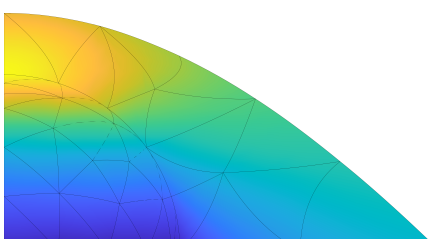




3D Supersonic flow: $M = 2$ flow over sphere

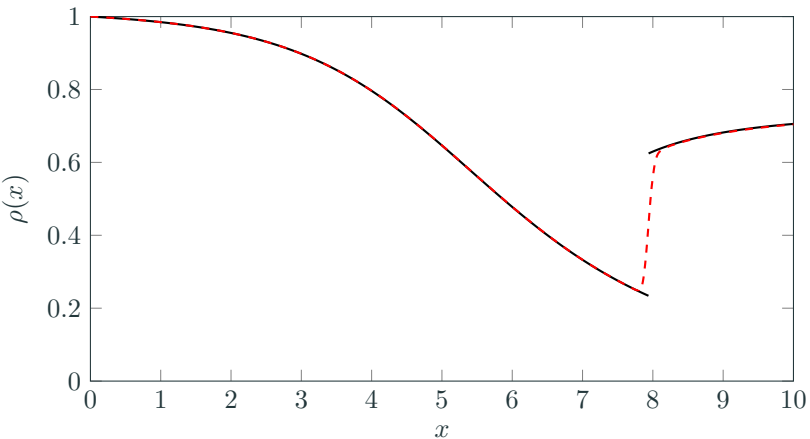
- 3D steady Euler equations (ideal gas law)
- Local space: $p = q = 2$ tetrahedral elements
- Starting point: 511 elements (no knowledge of shocks), 1st order FV solution
- HOIST solution (30 iter): 491 elements, shock-aligned





IST for viscous problems

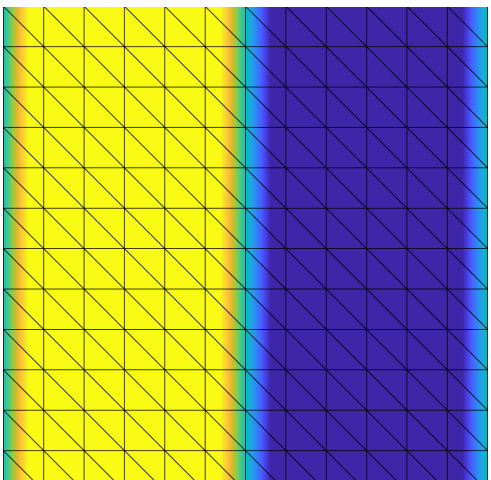
Shocks in viscous fluids have non-zero thickness



Inviscid (——) vs. viscous (- - -) flow through nozzle

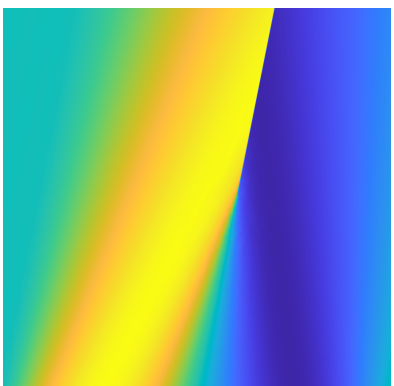
Approach: Instead of "tracking" the shock, we aim to aggressively compress elements into the shock thickness to resolve it accurately.

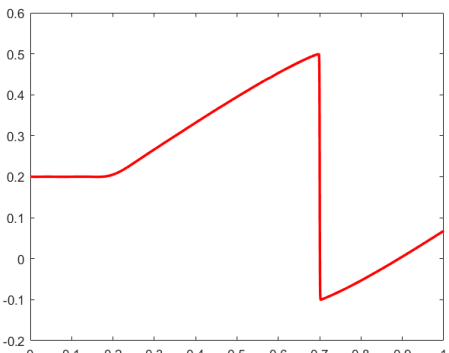
Viscous Burgers' shock formation, $\nu = 10^{-4}$



$p = 2$, $q = 1$; 3 compressed columns

Viscous Burgers' shock formation, $\nu = 10^{-4}$





$p = 2$, $q = 1$; 3 compressed columns

Extension of implicit shock tracking to viscous problems

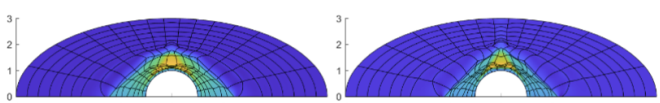
Our extension of HOIST to viscous problem leverages the same optimization formulation and SQP solver with a few important changes:

- Viscous DG discretization (e.g., IP, LDG, BR2, CDG, etc)
- No element collapses
- Viscosity continuation (to avoid carbuncles, obtain sufficient grid compression)
- Initialization from shock capturing solution (instead of $p = 0$ solve)
- p-adaptivity to avoid excessive compression in shock thickness

Extension of implicit shock tracking to viscous problems

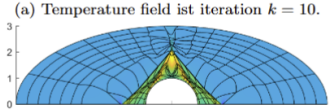
Our extension of HOIST to viscous problem leverages the same optimization formulation and SQP solver with a few important changes:

- Viscous DG discretization (e.g., IP, LDG, BR2, CDG, etc)
- No element collapses
- Viscosity continuation (to avoid carbuncles, obtain sufficient grid compression)
- Initialization from shock capturing solution (instead of $p = 0$ solve)
- p-adaptivity to avoid excessive compression in shock thickness

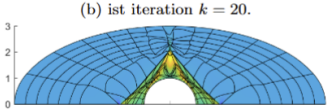


(a) Temperature field ist iteration $k = 10$.

(b) ist iteration $k = 20$.

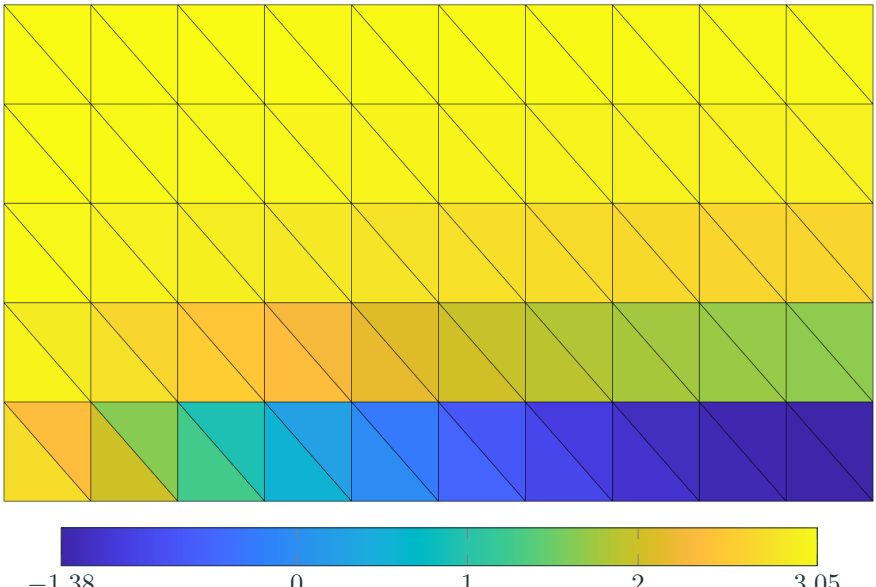


(c) ist iteration $k = 50$.



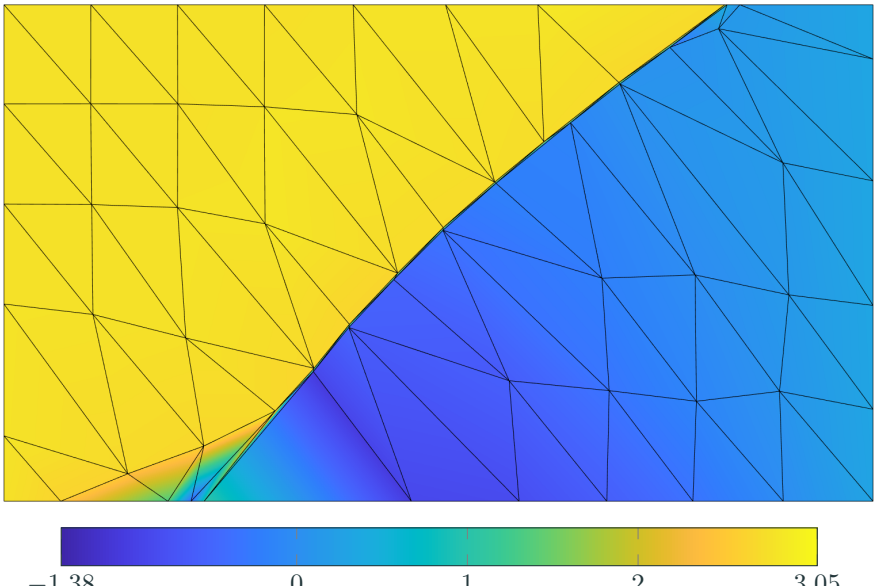
(d) ist iteration $k = 100$.

Burgers' equation ($\nu = 10^{-3}$), shock formation/acceleration (curved)



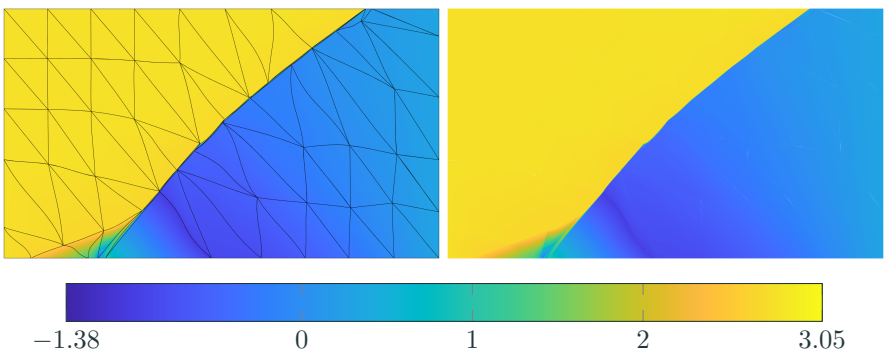
Initialization ($p = 0$ solve)

Burgers' equation ($\nu = 10^{-3}$), shock formation/acceleration (curved)



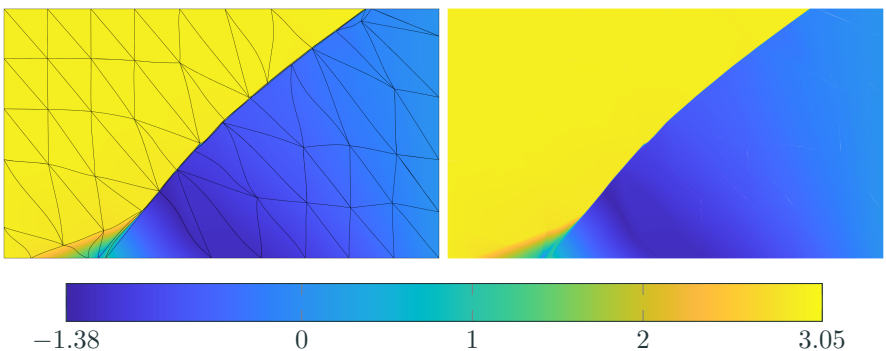
IST solution, before p-adaptivity

Burgers' equation ($\nu = 10^{-3}$), shock formation/acceleration (curved)



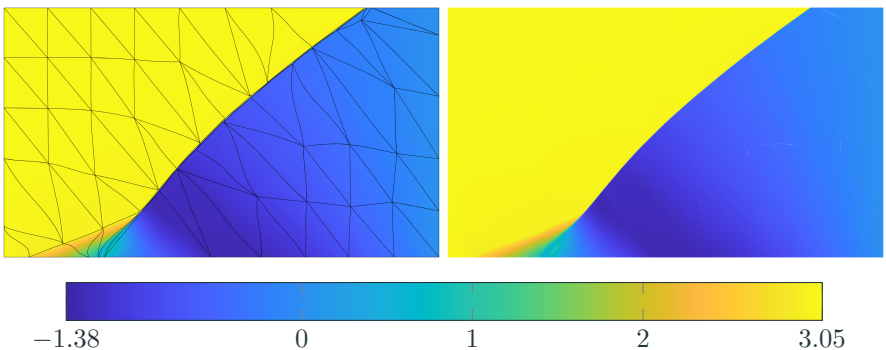
IST solution, before p-adaptivity, after q-refinement

Burgers' equation ($\nu = 10^{-3}$), shock formation/acceleration (curved)



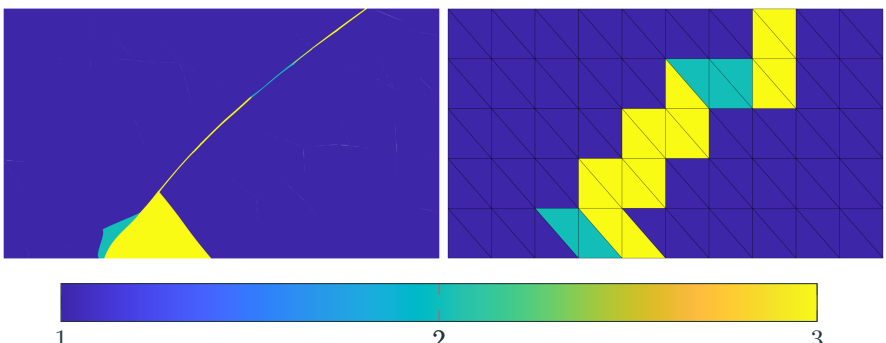
IST solution, after p-adaptivity (state gradient), first round

Burgers' equation ($\nu = 10^{-3}$), shock formation/acceleration (curved)



IST solution, after p-adaptivity (state gradient), two (final) round

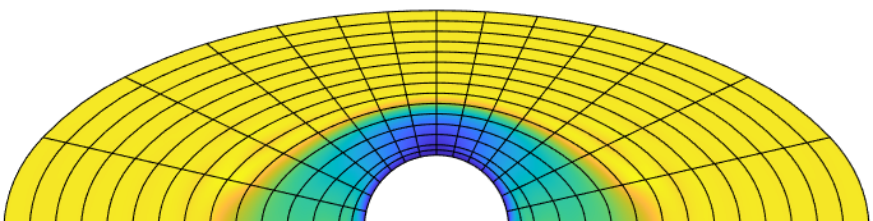
Burgers' equation ($\nu = 10^{-3}$), shock formation/acceleration (curved)



Polynomial degree, after p-adaptivity (state gradient), two (final) round

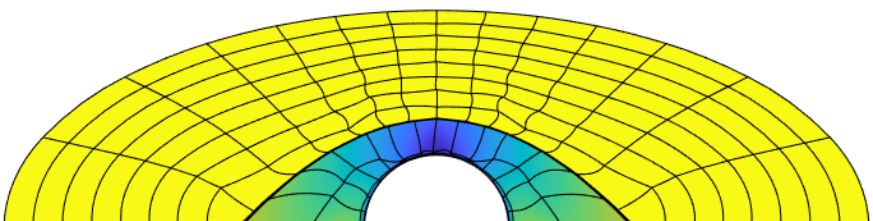
Reference domain (*left*) vs. physical domain (*right*)

$M_\infty = 5$, $Re = 10^3$ flow over an half cylinder



Observation: elements compress bow shock and boundary layer.

$M_\infty = 5$, $Re = 10^3$ flow over an half cylinder

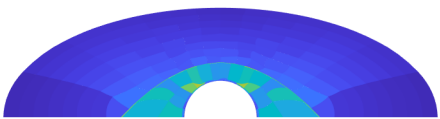


Observation: elements compress bow shock and boundary layer.

$M_\infty = 5$, $Re = 10^3$ flow over an half cylinder

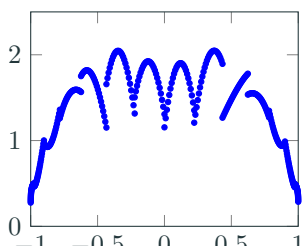
0th adaptation

p distribution



Enriched residual error indicator

Mach number



Heat flux profile

Observation: refinement around shock and downstream smooth region; error reduction; flow solution; point-wise heat flux improvement.

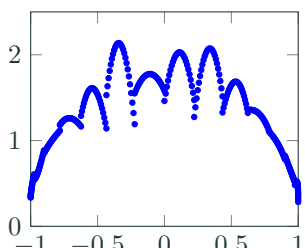
$M_\infty = 5$, $Re = 10^3$ flow over an half cylinder

1st adaptation

p distribution

Enriched residual error indicator

Mach number



Heat flux profile

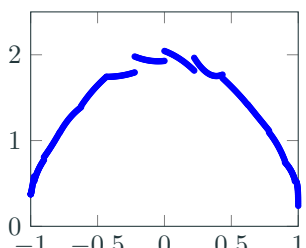
Observation: refinement around shock and downstream smooth region; error reduction; flow solution; point-wise heat flux improvement.

2nd adaptation

p distribution

Enriched residual error indicator

Mach number



Heat flux profile

Observation: refinement around shock and downstream smooth region; error reduction; flow solution; point-wise heat flux improvement.

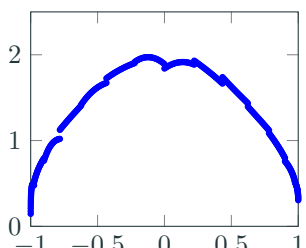
$M_\infty = 5$, $Re = 10^3$ flow over an half cylinder

3rd adaptation

p distribution

Enriched residual error indicator

Mach number



Heat flux profile

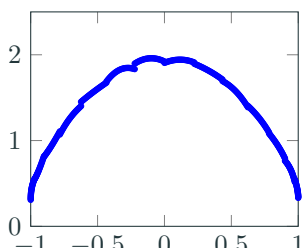
Observation: refinement around shock and downstream smooth region; error reduction; flow solution; point-wise heat flux improvement.

4th adaptation

p distribution

Enriched residual error indicator

Mach number

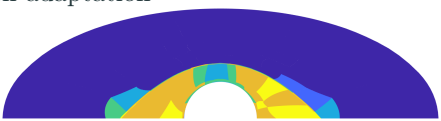


Heat flux profile

Observation: refinement around shock and downstream smooth region; error reduction; flow solution; point-wise heat flux improvement.

$M_\infty = 5$, $Re = 10^3$ flow over an half cylinder

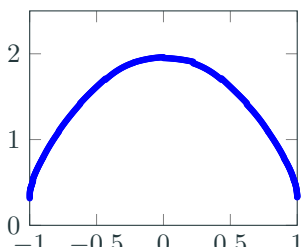
5th adaptation



p distribution

Enriched residual error indicator

Mach number



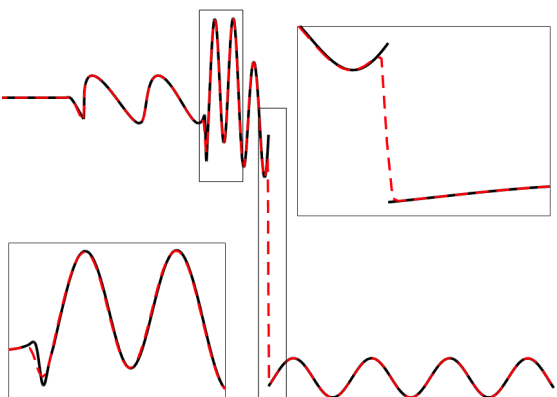
Heat flux profile

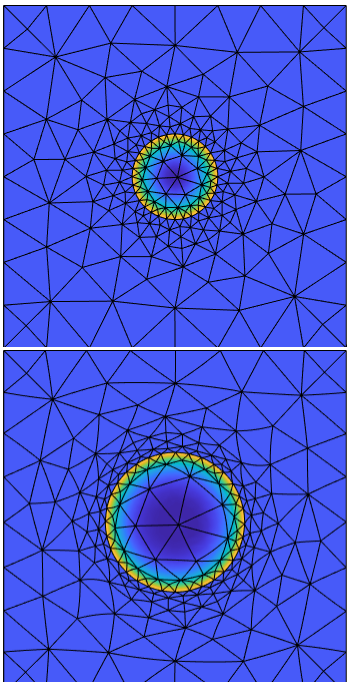
Observation: refinement around shock and downstream smooth region; error reduction; flow solution; point-wise heat flux improvement.

IST for time-dependent problems

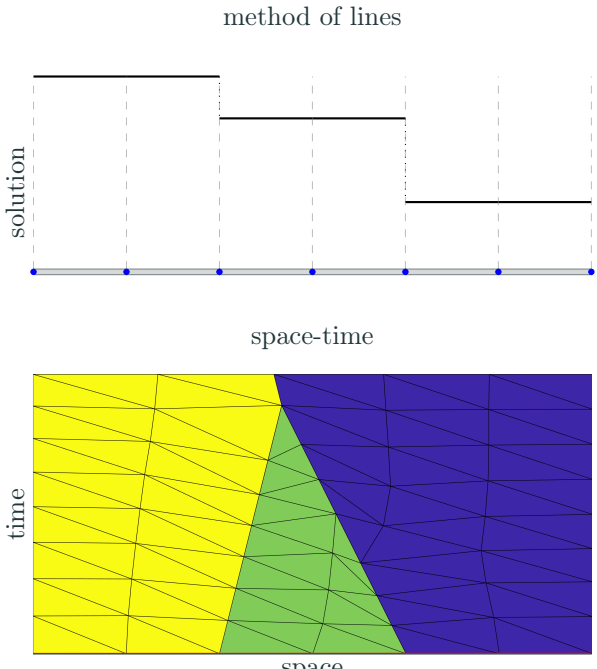
MOL/HOIST is preferred approach for a limited class of problems

Suitable for problems with arbitrarily complex shocks, but *limited shock motion*

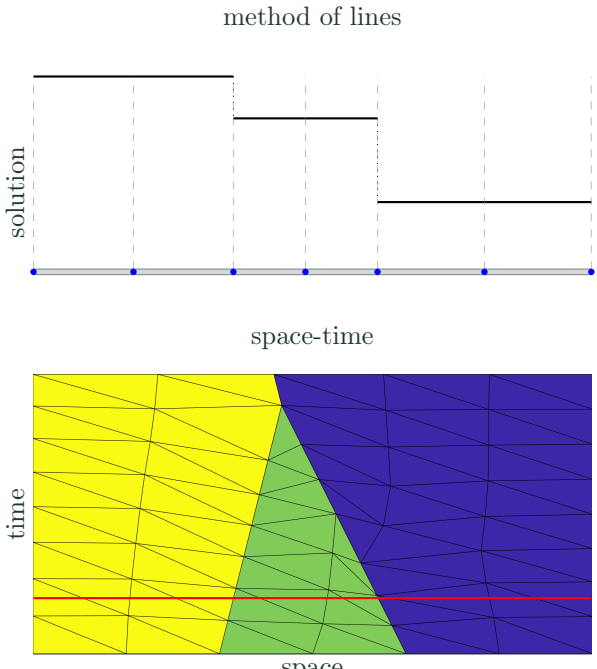




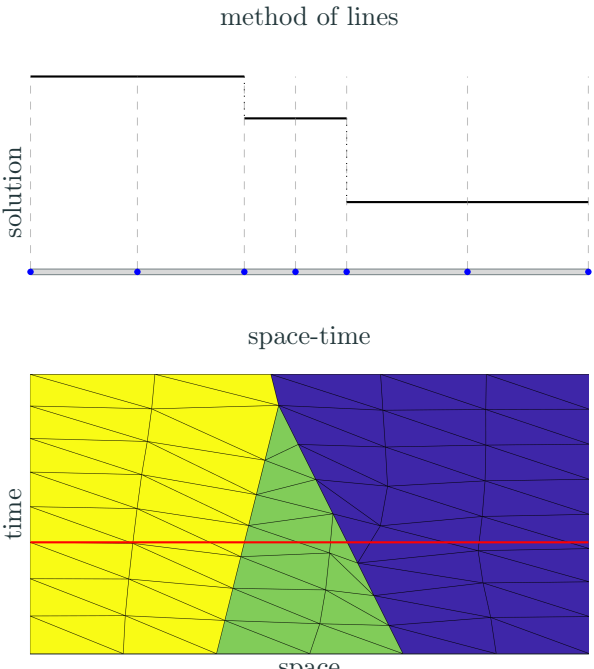
Space-time > MOL for colliding shocks



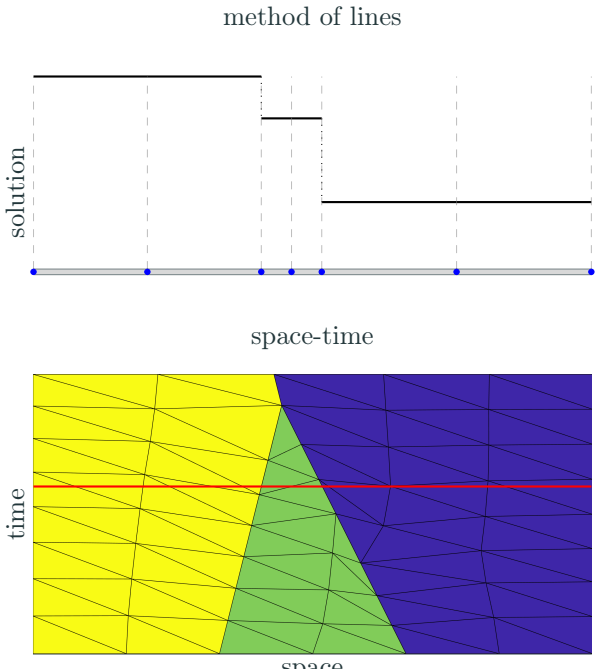
Space-time > MOL for colliding shocks



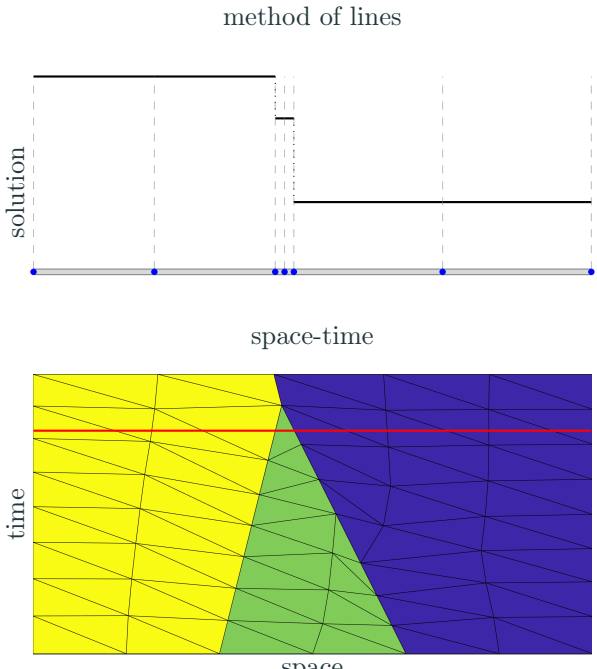
Space-time > MOL for colliding shocks



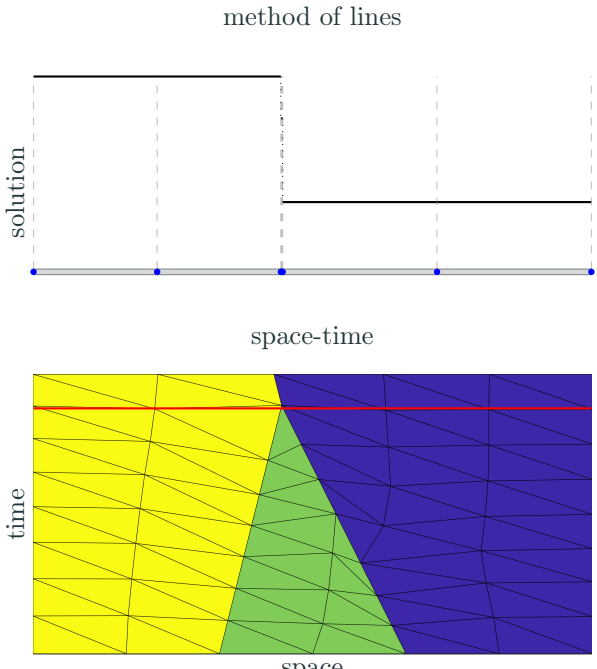
Space-time > MOL for colliding shocks



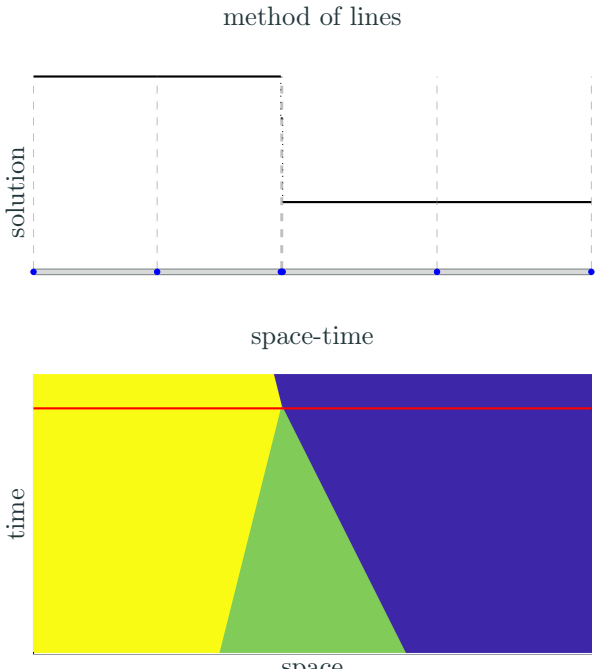
Space-time > MOL for colliding shocks



Space-time > MOL for colliding shocks

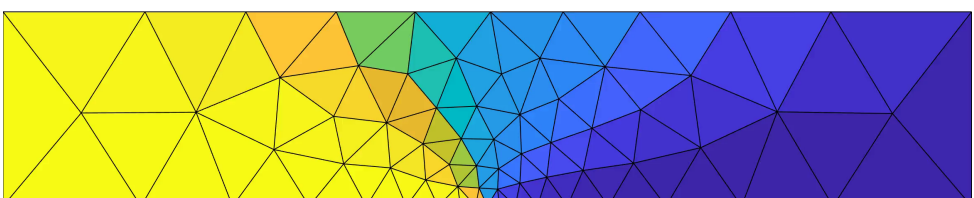


Space-time > MOL for colliding shocks



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

- 1D unsteady Euler equations (space-time formulation)
- Local space: $p = 2$, $q = 1$ triangle elements
- Starting point: 108 elements (no knowledge of shocks), 1st order FV solution
- HOIST solution (120 iter): 84 elements, shock-aligned



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

Observation: For complex problems, not practical to couple entire time domain.

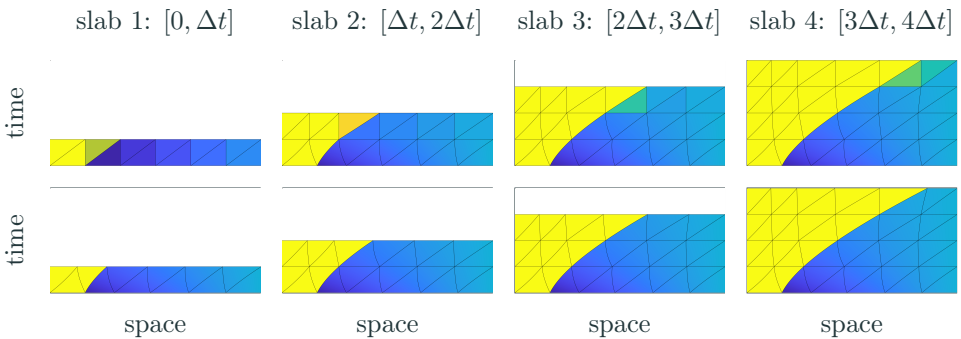
Unsteady, inviscid flow, space-time: Sod shock tube

- 1D unsteady Euler equations (space-time formulation)
- Local space: $p = 2$, $q = 1$ triangle elements
- Starting point: 108 elements (no knowledge of shocks), 1st order FV solution
- HOIST solution (120 iter): 84 elements, shock-aligned

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

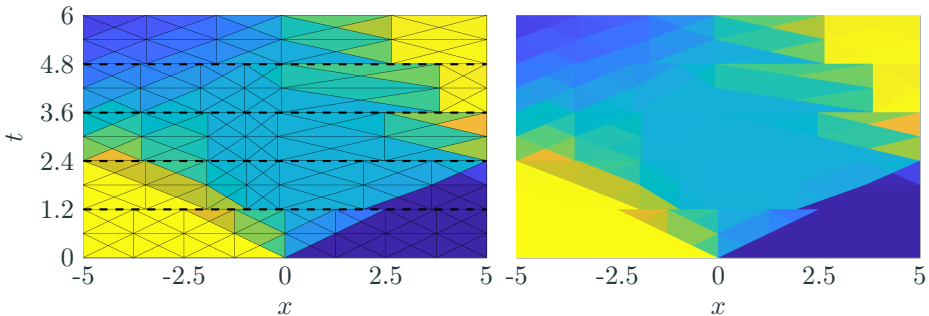
Observation: For complex problems, not practical to couple entire time domain.

Slab-based, space-time implicit shock tracking: 1D+time simulation



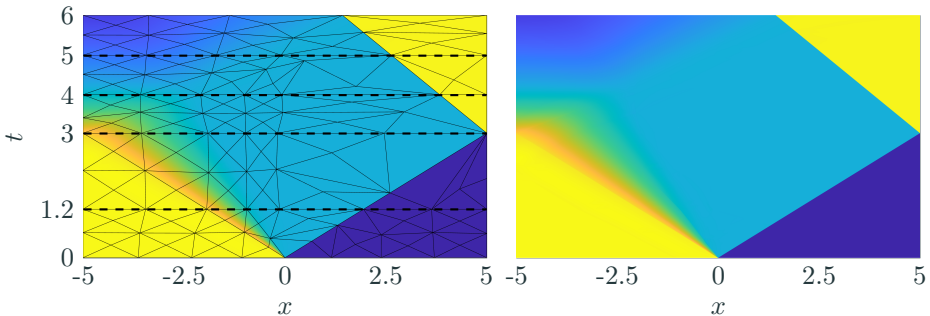
1) The spatial mesh at the end of the nth slab is extruded in the temporal dimension
2) The space-time elements are split into simplices (triangles)
3) A starting point for the flow solution over the slab is obtained using a first-order finite volume method (*top row*)
4) From this starting point, HOIST is applied over the slab to track the discontinuity in space-time (*bottom row*)

Shallow water equations (1d+time): Dam break



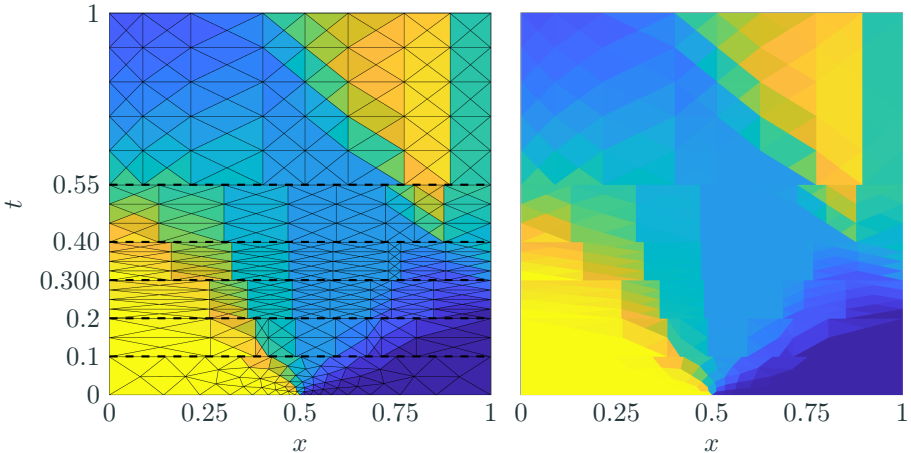
$p = 0$ solve for each slab with slab interface (- - -)

Shallow water equations (1d+time): Dam break



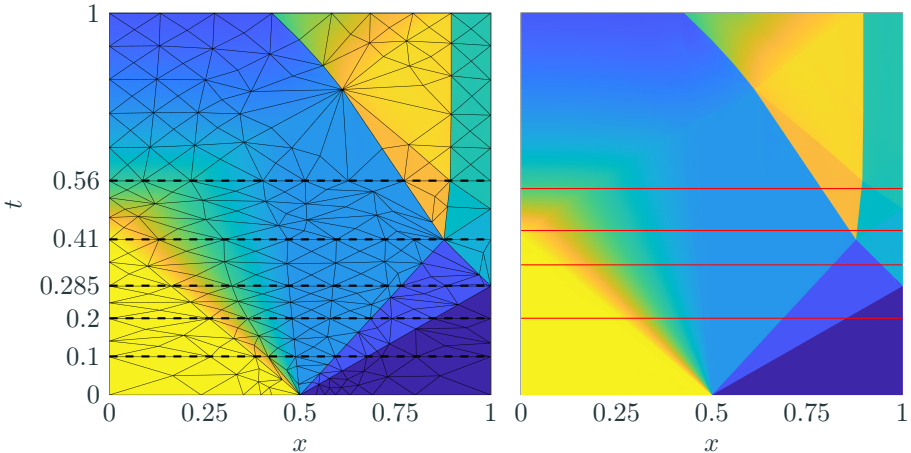
Converged IST solution for each slab with slab interface (- - -)

Euler equations, ideal gas (1d+time): Sod shock tube



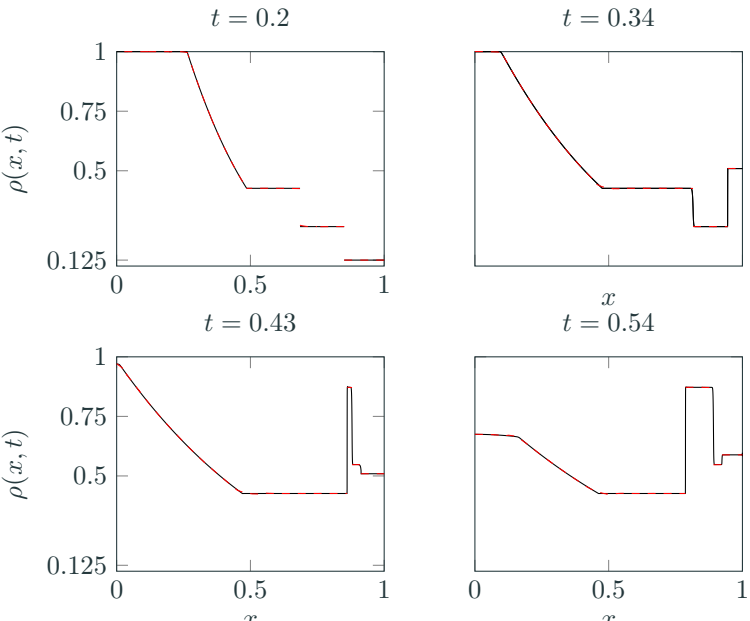
$p = 0$ solve for each slab with slab interface (- - -)

Euler equations, ideal gas (1d+time): Sod shock tube



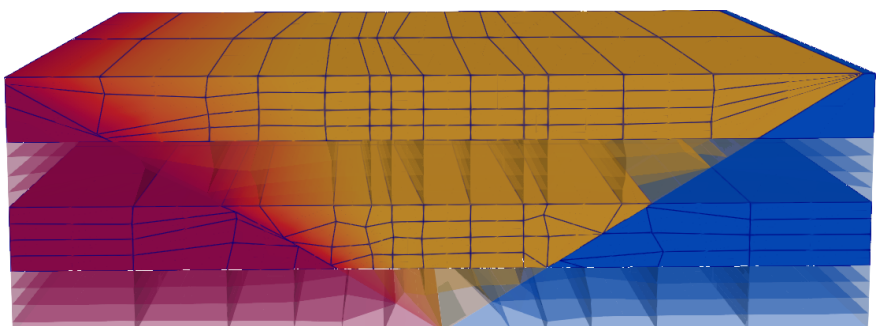
Converged IST solution for each slab with slab interface (- - -)

Euler equations, ideal gas (1d+time): Sod shock tube



Analytical solution (density) (——) vs. IST solution (- - -)

Shallow water equations (2d+time): Dam break



Two-dimensional dam break solution using hexahedral elements (4 slabs)

Nonlinear Finite Element Manifolds

Practicality of IST?

Implicit shock tracking can achieve unprecedented accuracy per degree of freedom, even for problems with complex shock structures. However, it has two major drawbacks that have made it very difficult to scale to large, real-world problems:

- IST inherently mixes L^2 and H^1 spaces, which leads to large and complex linearized KKT systems, where efficient preconditioning is challenging

$$\begin{bmatrix} \boldsymbol{B_{uu}}(\boldsymbol{z}_k, \hat{\boldsymbol{\lambda}}(\boldsymbol{z}_k)) & \boldsymbol{B_{ux}}(\boldsymbol{z}_k, \hat{\boldsymbol{\lambda}}(\boldsymbol{z}_k)) & \boldsymbol{J_u}(\boldsymbol{z}_k)^T \\ \boldsymbol{B_{ux}}(\boldsymbol{z}_k, \hat{\boldsymbol{\lambda}}(\boldsymbol{z}_k))^T & \boldsymbol{B_{xx}}(\boldsymbol{z}_k, \hat{\boldsymbol{\lambda}}(\boldsymbol{z}_k)) & \boldsymbol{J_x}(\boldsymbol{z}_k)^T \\ \boldsymbol{J_u}(\boldsymbol{z}_k) & \boldsymbol{J_x}(\boldsymbol{z}_k) & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \Delta \boldsymbol{u}_k \\ \Delta \boldsymbol{x}_k \\ \boldsymbol{\eta}_k \end{bmatrix} = - \begin{bmatrix} \boldsymbol{g_u}(\boldsymbol{z}_k) \\ \boldsymbol{g_x}(\boldsymbol{z}_k) \\ \boldsymbol{r}(\boldsymbol{z}_k) \end{bmatrix}$$

- Substantial effort required to retain high-quality mesh (Hessian regularization, mesh distortion penalization, solution reinitialization, element collapses, etc)

Potential solution: Nonlinear finite element manifolds, a discontinuous finite element method on a fixed grid that leverages *nonlinear trial manifolds*

IST as a particular case of a nonlinear trial manifold

Consider the solution $U : \Omega \subset \mathbb{R}^d \to \mathbb{R}^m$ of a conservation law and a z-parametrized mapping $\mathcal{G}_z : \bar{\Omega} \to \Omega$, where $\bar{\Omega} \subset \mathbb{R}^d$ is a fixed (z-independent) computational domain. We define the solution $\bar{U} : \bar{\Omega} \to \mathbb{R}^m$ as

$$\bar{U}(X) \coloneqq U(\mathcal{G}_z(X))$$

and approximate it using standard finite element techniques

$$\bar{U}_h(X) \coloneqq \alpha_i \phi_i(X).$$

This induces the following approximation in the physical domain ($x = \mathcal{G}_z(X)$)

$$U_h(x) \coloneqq \bar{U}_h(\mathcal{G}_z^{-1}(x)) = \alpha_i \phi_i(\mathcal{G}_z^{-1}(x)) = \alpha_i \psi_i(x; z)$$

where $\psi_i(x; z) = \phi_i(\mathcal{G}_z^{-1}(x))$.

Interpretation: IST employs a nonlinear approximation where we simultaneously seek the shape of the trial space basis function, ψ_i, through the nonlinear parameter z, and the linear coefficients of the approximation (α_i).

Nonlinear trial manifold: general construction via enrichment

On each element K, enrich the standard polynomial space with basis functions modulated by a nonlinear parameter $z \in \mathbb{R}^n$

$$V \coloneqq \operatorname{span}\{\phi_0, \dots, \phi_k\}, \qquad \hat{V}_z \coloneqq \operatorname{span}\{\psi_0(\cdot; z), \dots, \psi_{\bar{k}}(\cdot; z)\}.$$

The union of $\hat{V}_z$ over all $z \in \mathbb{R}^n$ forms a *nonlinear trial manifold*

$$\mathcal{M} \coloneqq V \oplus \bigcup_{z \in \mathbb{R}^n} \hat{V}_z \subset L^2(K),$$

and the element-wise solution takes the form

$$U_h(x; z)|_K = \sum_{l=0}^{k} \alpha_l \, \phi_l(\mathcal{G}_K^{-1}(x)) + \sum_{l=0}^{\bar{k}} \beta_l \, \psi_l(\mathcal{G}_K^{-1}(x); z),$$

where $\mathcal{G}_K$ is the isoparametric mapping for element K.

Key property: linear coefficients (α_l, β_l) and nonlinear parameters z are *jointly* determined to adapt the trial space to the solution.

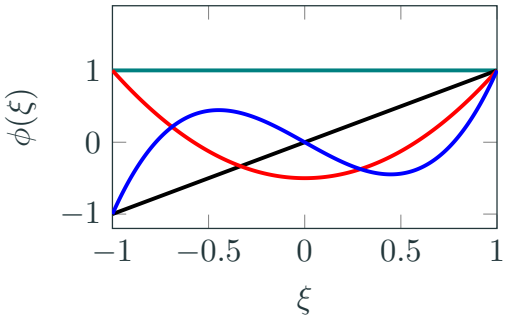
Nonlinear parameterization: composition with polynomial basis

We construct ψ_j by composing the standard polynomial basis with a parameterized coordinate map $\mathsf{g} : [-1,1]^d \times \mathbb{R}^d \to [-1,1]^d$

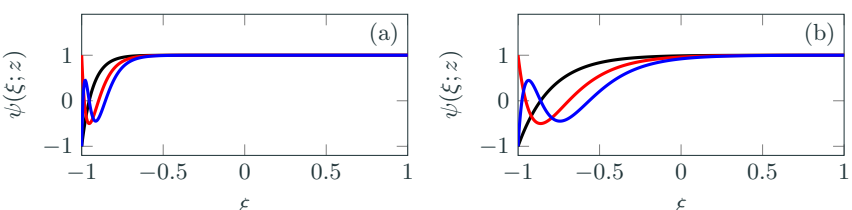
$$\psi_j(\xi;\tau) = \phi_j\big(\mathsf{g}(\xi,\tau)\big), \qquad \mathsf{g}(\xi,\tau) \coloneqq 2\,\frac{\exp\big(\tau(\xi+1)/2\big)-1}{\exp(\tau)-1} - 1.$$

- $\tau \to 0$: g is the identity $\Rightarrow$ recovers standard polynomial basis
- $|\tau|$ large: basis functions concentrate toward element boundary $\Rightarrow$ resolves thin layers
- No a priori knowledge of layer scale or wall distance required

Nonlinear parameterization: standard vs. composed basis



standard polynomial basis ($\tau = 0$)



composed basis: (a) $\tau = -30$, (b) $\tau = -10$

Optimization formulation: minimize DG residual over $\mathcal{M}$

Given an IPDG discretization with trial space $\mathcal{V}_{h,Z}$ (parameterized by element-wise $Z = \{z_K\}$) and a *Petrov–Galerkin* test space $\mathcal{B}_{h,\delta,Z}$ of elevated degree $p + \delta$, the algebraic residual

$$\boldsymbol{R} : \mathbb{R}^{N_{\boldsymbol{\gamma}}} \times \mathbb{R}^{N_{\boldsymbol{z}}} \to \mathbb{R}^{N_{\boldsymbol{r}}}, \qquad N_{\boldsymbol{r}} > N_{\boldsymbol{\gamma}} + N_{\boldsymbol{z}}$$

is overdetermined. We determine $(\boldsymbol{\gamma}, \boldsymbol{z})$ by minimizing the DG residual norm over the nonlinear manifold

$$(\boldsymbol{\gamma}_*, \boldsymbol{z}_*) \coloneqq \underset{\boldsymbol{\gamma} \in \mathbb{R}^{N_{\boldsymbol{\gamma}}},\, \boldsymbol{z} \in \mathbb{R}^{N_{\boldsymbol{z}}}}{\arg\min} \frac{1}{2} \|\boldsymbol{R}(\boldsymbol{\gamma}, \boldsymbol{z})\|_2^2.$$

Heterogeneous enrichment: partition mesh into enriched and standard elements, $\mathcal{E}_h = \mathcal{E}_h^{\text{enr}} \cup \mathcal{E}_h^{\text{std}}$, so nonlinear parameters are introduced only where needed.

vs. IST: fixed mesh, purely algebraic unknowns, no mesh-quality management, no L^2/H^1 coupling.

Levenberg–Marquardt solver for nonlinear least squares

Iterates $(\boldsymbol{\gamma}^{(n)}, \boldsymbol{z}^{(n)})$ updated as

$$\boldsymbol{\gamma}^{(n+1)} = \boldsymbol{\gamma}^{(n)} + \alpha_{n+1}\Delta\boldsymbol{\gamma}^{(n+1)}, \qquad \boldsymbol{z}^{(n+1)} = \boldsymbol{z}^{(n)} + \alpha_{n+1}\Delta\boldsymbol{z}^{(n+1)},$$

with step $(\Delta\boldsymbol{\gamma}, \Delta\boldsymbol{z})$ from the damped linear least-squares problem

$$\underset{(\Delta\boldsymbol{\gamma},\Delta\boldsymbol{z})}{\arg\min} \left\| \begin{bmatrix} \boldsymbol{R}^{(n)} \\ \boldsymbol{0} \end{bmatrix} + \begin{bmatrix} \boldsymbol{J}_{\boldsymbol{\gamma}}^{(n)} & \boldsymbol{J}_{\boldsymbol{z}}^{(n)} \\ \sqrt{\lambda}\boldsymbol{I} & \sqrt{\lambda}\boldsymbol{I} \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{\gamma} \\ \Delta\boldsymbol{z} \end{bmatrix} \right\|_2,$$

where $\lambda > 0$ is adapted via the gain ratio $\rho^{(n+1)}$ and Armijo line search selects α_{n+1}.

- **Gauss–Newton-like** step $\Rightarrow$ fast local convergence
- λ damping $\Rightarrow$ robustness far from the solution
- First-order optimality: $\boldsymbol{J}_{\boldsymbol{\gamma}}^T\boldsymbol{R} = \boldsymbol{0}$, $\boldsymbol{J}_{\boldsymbol{z}}^T\boldsymbol{R} = \boldsymbol{0}$

Three DG Formulations

CoDG Conventional DG: standard polynomial trial space V of degree p with Galerkin test space of equal dimension.

FNLDG Frozen Nonlinear DG: enriched trial space with *prescribed* nonlinear parameter

$$z^* = \frac{Pe}{pN},$$

relating mesh size, polynomial degree, and boundary-layer thickness $\delta_{\mathrm{BL}} = L/Pe$ (z^* is the ratio between the effective grid spacing h/p and the BL thickness).

NLDG Nonlinear DG: enriched trial space where nonlinear parameters are jointly optimized within a Petrov–Galerkin framework.

Problem

Domain $\Omega = [0, L]$, $L = 1$, homogeneous Dirichlet BCs.

$$\frac{\partial u}{\partial t} + \nabla \cdot (\beta\, u) = \nu \Delta u - 2\sin(\pi x/L).$$

- Advection velocity $\beta = 1$, diffusion $\nu = 10^{-3}$.
- Péclet number: $Pe = \beta L/\nu = 1000$.
- Thin boundary layer with steep gradients near $x = 1$.

Convergence Study

p	h	CoDG		FNLDG		NLDG	
		Error	Rate	Error	Rate	Error	Rate
1	1.250e-01	3.732e+00	-	1.113e-02	-	**2.529e-03**	-
	6.250e-02	1.212e+00	1.62	3.136e-03	1.83	**4.641e-04**	2.45
	3.125e-02	3.618e-01	1.74	7.797e-04	2.01	**8.444e-05**	2.46
	1.562e-02	8.976e-02	2.01	1.667e-04	2.23	**1.803e-05**	2.23
2	1.250e-01	5.239e-01	-	1.166e-04	-	**1.500e-05**	-
	6.250e-02	1.198e-01	2.13	1.175e-05	3.31	**5.420e-07**	4.79
	3.125e-02	2.140e-02	2.48	1.011e-06	3.54	**2.240e-08**	4.60
	1.562e-02	2.401e-02	-0.17	6.331e-08	4.00	**5.999e-09**	1.90
3	1.250e-01	1.047e-01	-	7.602e-06	-	**1.225e-06**	-
	6.250e-02	2.670e-02	1.97	**3.420e-07**	4.47	4.114e-07	1.57
	3.125e-02	3.327e-02	-0.32	**7.947e-09**	5.43	2.587e-08	3.99
	1.562e-02	1.815e-02	0.87	**1.130e-10**	6.14	8.627e-10	4.91

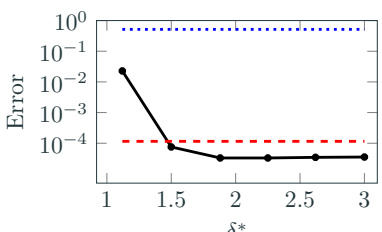
Table 1: L^2-errors and observed convergence rates for the one-dimensional advection–diffusion equation under uniform mesh refinement.

Test Space Enrichment Study

- 4-element mesh, $p = 3$, $Pe = 10^3$.
- Enrichment factor:

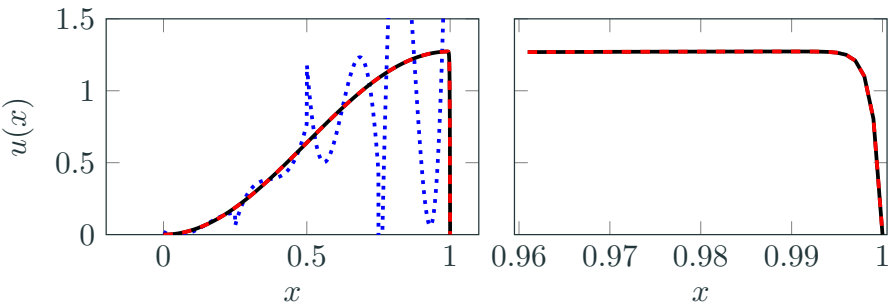
$$\delta^* = \frac{\dim(\text{Test Space})}{\dim(\text{Trial Space})}$$

- NLDG error decreases with δ^*, plateaus around $\delta^* \approx 1.9$.
- At plateau: $\sim 3.5\times$ reduction in L^2 error vs FNLDG.



L^2 error as a function of test space enrichment factor δ^* for the one-dimensional advection–diffusion problem on a 4-element mesh with $p = 3$. Legend: CoDG (·····), FNLDG (- - -), NLDG (—●—).

1-D Solution Comparison



Solution profiles $u(x)$ for the one-dimensional advection–diffusion problem at $Pe = 10^3$ on a coarse 4-element mesh with $p = 3$. **Left**: full domain comparison among CoDG/coarse (·····), CoDG/fine (——), NLDG/coarse (- - -). **Right**: zoomed view of the boundary layer near $x = 1$.

2-D Advection–Diffusion

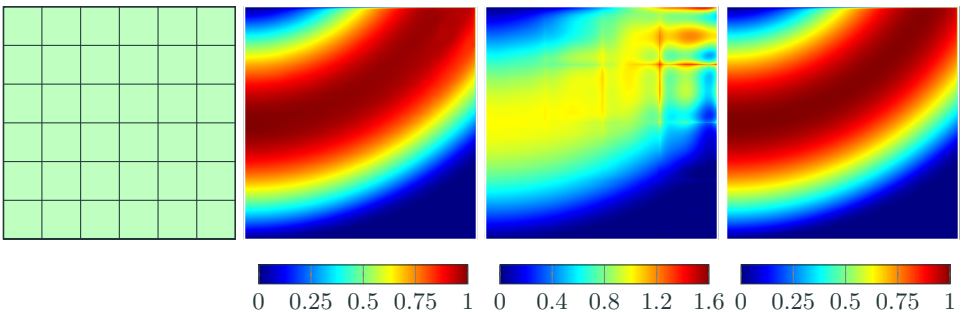
Problem

Domain $\Omega = [0,1]^2$.

$$\frac{\partial u}{\partial t} + \nabla \cdot (\boldsymbol{\beta}\, u) = \nu\, \Delta u, \qquad \boldsymbol{\beta}(x_1, x_2) = \left(-x_2 + \tfrac{3}{2},\ x_1\right).$$

- Diffusion: $\nu = 10^{-3} \Rightarrow Pe = 10^3$.
- Dirichlet BCs: $u(0, x_2) = \sin(\pi x_2)$; $u = 0$ on remaining boundaries.
- Solution exhibits a pronounced corner boundary layer.

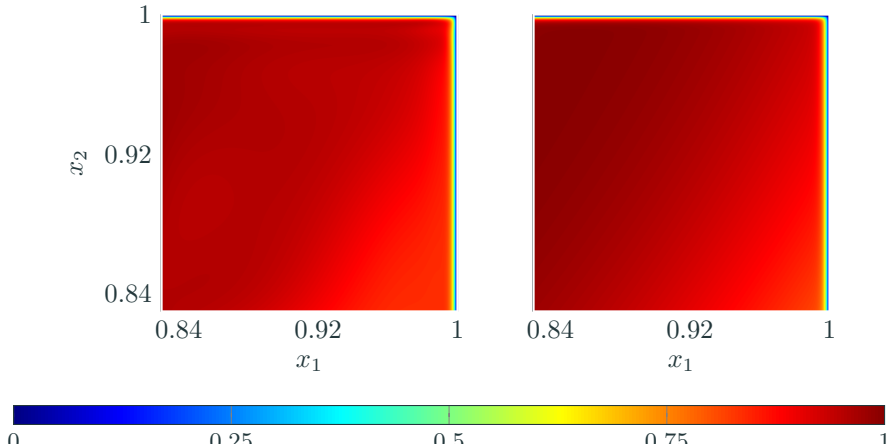
2-D Solution Contours



(a) Mesh (coarse) **(b)** NLDG (coarse) **(c)** CoDG (coarse) **(d)** CoDG (fine)

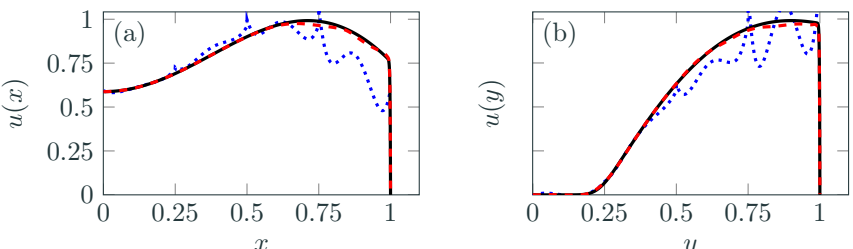
Solution contours for the two-dimensional corner boundary layer at $\mathrm{Pe} = 10^3$ obtained using NLDG and CoDG on a 6×6 coarse mesh and the corresponding reference.

2-D Zoomed Corner Region



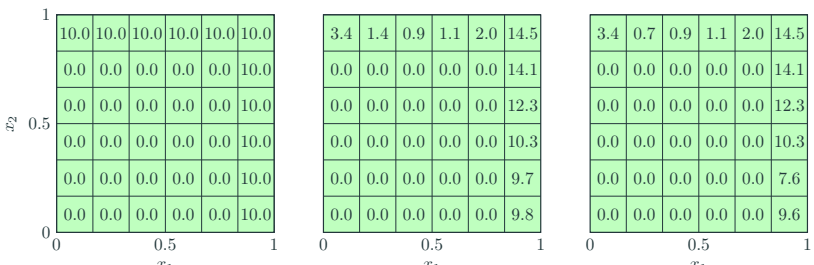
Zoomed view of the corner region $[0.83, 1]^2$. *Left*: NLDG (coarse). *Right*: CoDG (fine).

2-D Line Comparisons



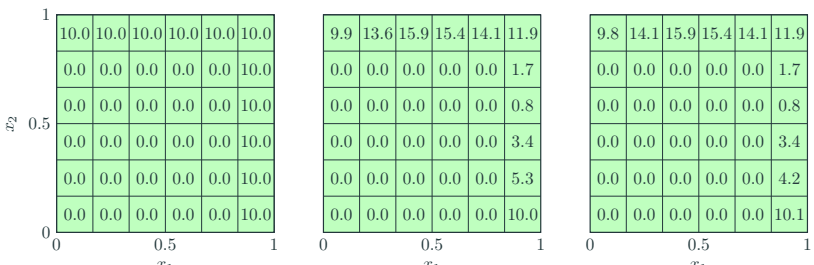
Comparison among solution profiles at $x_2 = 0.8$ (left) and $x_1 = 0.8$ (right) for the two-dimensional corner boundary layer at $\mathrm{Pe} = 10^3$ obtained using CoDG/coarse (·····), CoDG/fine (——), and NLDG/coarse (- - -).

Optimized Nonlinear Parameters — ξ_1-direction



Element-wise $z_1^{(e)}$ at initialization (left), iteration 150 (center), and iteration 300 (right).

Optimized Nonlinear Parameters — ξ_2-direction



Element-wise $z_2^{(e)}$ at initialization (left), iteration 150 (center), and iteration 300 (right).

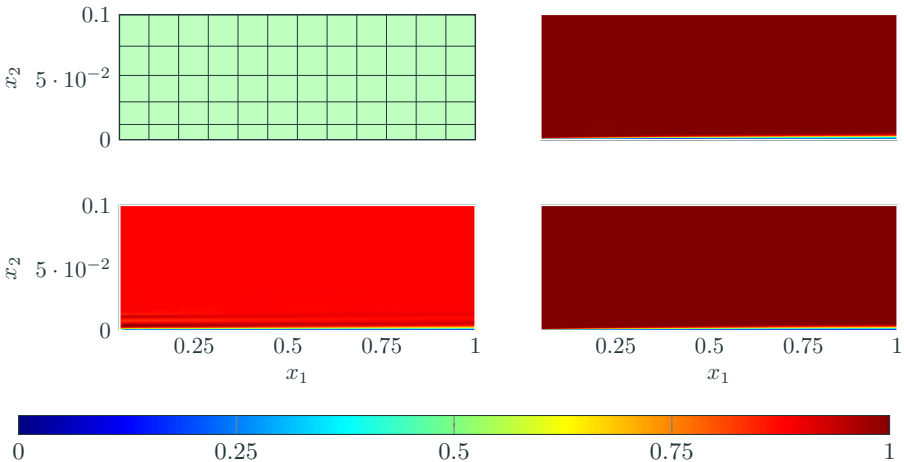
Governing Equations — Incompressible Navier–Stokes

$$\nabla \cdot \mathbf{u} = 0, \qquad \frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u} \otimes \mathbf{u}) + \nabla q = \nu \, \Delta \mathbf{u},$$

where $\mathbf{u}$ is the velocity field and q is the kinematic pressure.

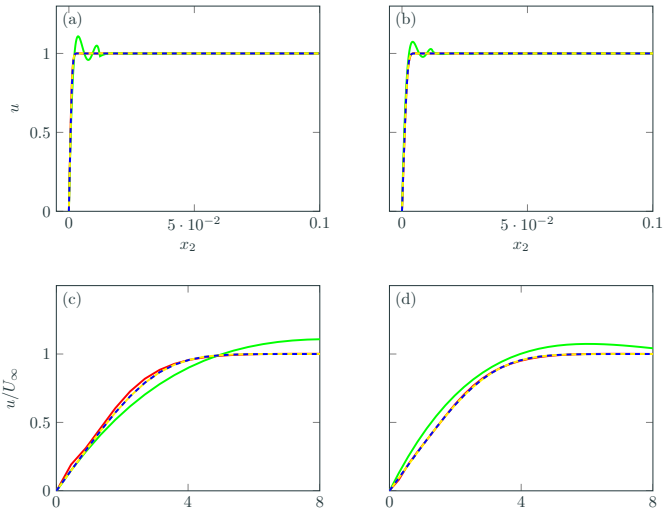
- Flow over a flat plate (no pre-inlet buffer region).
- Reynolds number: $Re = 10^6$ — thin laminar boundary layer develops along the plate surface.
- Polynomial degree $p = 4$.

Velocity Magnitude Contours



Velocity magnitude contours for the flat-plate boundary layer at $\mathrm{Re} = 10^6$ obtained using CoDG/coarse (*bottom left*), CoDG/fine (*top right*), NLDG/coarse (*bottom right*). The coarse mesh (*top left*) uses 12×5 elements slightly skewed towards boundary layer.

Velocity Profiles



Velocity profiles u_1 at $x_1 = 0.2$ (a) and $x_1 = 0.5$ (b) x_2, and the corresponding profiles normalized by U_∞ plotted against the similarity variable η (c, d). Legend: NLDG/coarse (——), CoDG/coarse (——), CoDG/fine (——), and Blasius (- - -).

Conclusion

Optimization-based simulation for high-speed flows

Optimization-based simulation via nonlinear trial manifolds: on each element, enrich a polynomial trial space with basis functions modulated by element-wise nonlinear parameters; jointly solve for linear coefficients and nonlinear parameters by minimizing the DG residual with a Petrov–Galerkin test space.

IST is a special case: take the nonlinear parameter z to be a mesh deformation $\mathcal{G}_z$; the trial manifold is $\alpha_i \, \phi_i(\mathcal{G}_z^{-1}(x))$ — basis functions deformed in physical space to align with shocks.

Demonstrated benefits on advection–diffusion and INS:

- Composed-parameterization enrichment recovers thin boundary layers on coarse, isotropic meshes with no layer scale or wall distance specified.
- NLDG achieves much lower L^2 errors than CoDG/FNLDG on coarse meshes, tracking fine-mesh references for 1D and 2D corner-layer problems.
- Element-wise parameters concentrate resolution in the layer automatically; interior elements recover the standard polynomial space.

Future directions: compressible Navier–Stokes (coupled velocity/thermal layers); efficient preconditioners and initialization for Levenberg–Marquardt; broader classes of nonlinear manifolds beyond IST and boundary-layer enrichment.