High-Order Space–Time Finite Element Simulations of Fluid Mechanics Using MFEM

Ramin Pahnabi, Kendrick M. Shepherd

Brigham Young University

MFEM Community Workshop

September 2025

Motivation for High-Performance Compressible Navier-Stokes Simulation

✓ Multiphysics

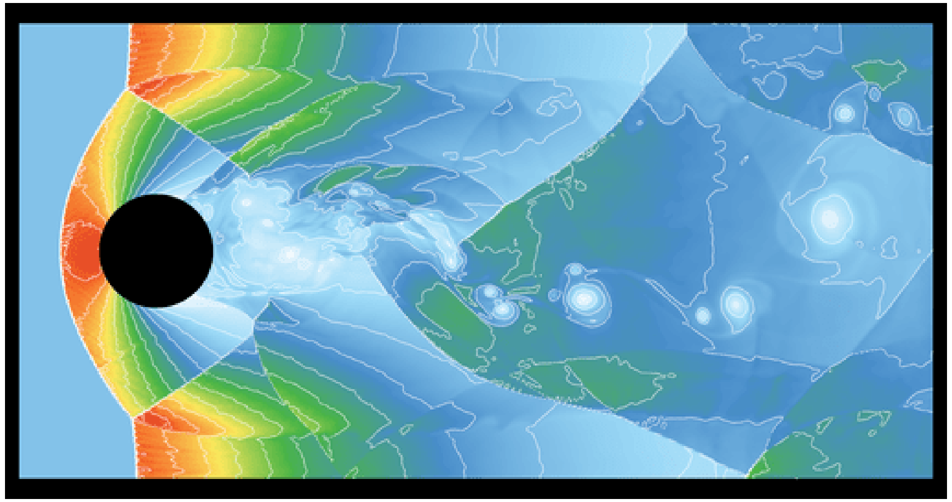
- Compressible Navier-Stokes equations form the core of many **Multiphysics problems**
    - Requires handling **tight coupling** of fluid, thermal and chemical processes
    - Enables simulation of **real-world engineering systems**

✓ Multiscale Dynamics

- Real-world applications exhibit a wide range of **spatial and temporal scales**
- Coexistence of fast (shocks, waves) and slow (thermal, structural) scales

✓ Advanced Applications

- **Hypersonic**: shock-shock interactions, high-temp gas physics
- **Aerothermoelasticity**: coupled fluid-thermal-structural response in high-speed vehicles



Compressible Euler equations, Mach 3 flow around a cylinder in 2D, stabilized DG-P1 spacial discretization. Image courtesy of Hennes Hajduk, as part of the 2021 MFEM Workshop Visualization Contest.

➢ ***Goal***: Enable robust, accurate simulation of complex, high-speed, Multiphysics systems using scalable space-time FEM

BYU

How to get High-Performance in Compressible Navier-Stokes Simulation

1. Scalability
How well the solver performs as we increase problem size and number of processors
- **Domain decomposition**
- **Parallel space-time methods**
- **Use of scalable libraries (e.g., MFEM, Hypre)**

Goal: Maintain performance as simulations grow to millions/billions of unknowns.

2. Efficiency
Maximizing computational speed and minimizing resource usage
- **Adaptive meshing** (spatial & temporal refinement only where needed)
- **High-order elements** for more accuracy per DOF
- **Preconditioning & solver optimization** (e.g., GMRES + block preconditioners)

Goal: Solve large problems quickly without wasting memory or compute time.

3. Stability
Ensuring the simulation doesn't blow up or produce non-physical results
- **Stabilization methods**:
SUPG, entropy viscosity, τ-stabilization
- **For advective stability Consistent of shock capturing methods** for compressible flow
- **Careful time integration**:

Goal: Handle shocks, stiffness, and Multiphysics coupling robustly.

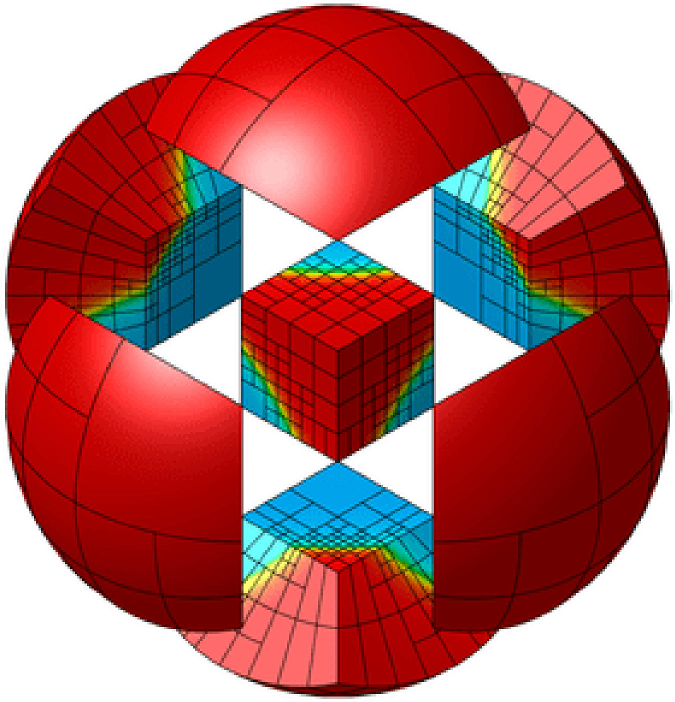
➢ Achieving high performance requires a careful balance of **parallel scalability**, **numerical efficiency**, and **algorithmic stability,** especially critical for large-scale, Multiphysics simulations.

BYU

What is MFEM?

- *__MFEM__ is an open-source C++ library for solving PDEs using finite element methods (FEM).*
- *Developed and maintained by __LLNL (Lawrence Livermore National Laboratory)__.*

- *Supports:*
  - High-performance computing (HPC)
  - High-order finite elements
  - Scalable, parallel simulations
  - *General static and dynamic PDEs*
  - Supports 1D, 2D, and 3D simulations
  - Handles structured & unstructured meshes
  - Compatible with AMR (adaptive mesh refinement)



BYU

Why Isogeometric Analysis (IGA): Accuracy with Fewer DOFs

- *IGA is powerful for getting high accuracy with fewer elements.*

Ideal for compressible flow, wave propagation, and Multiphysics systems where precision matters.

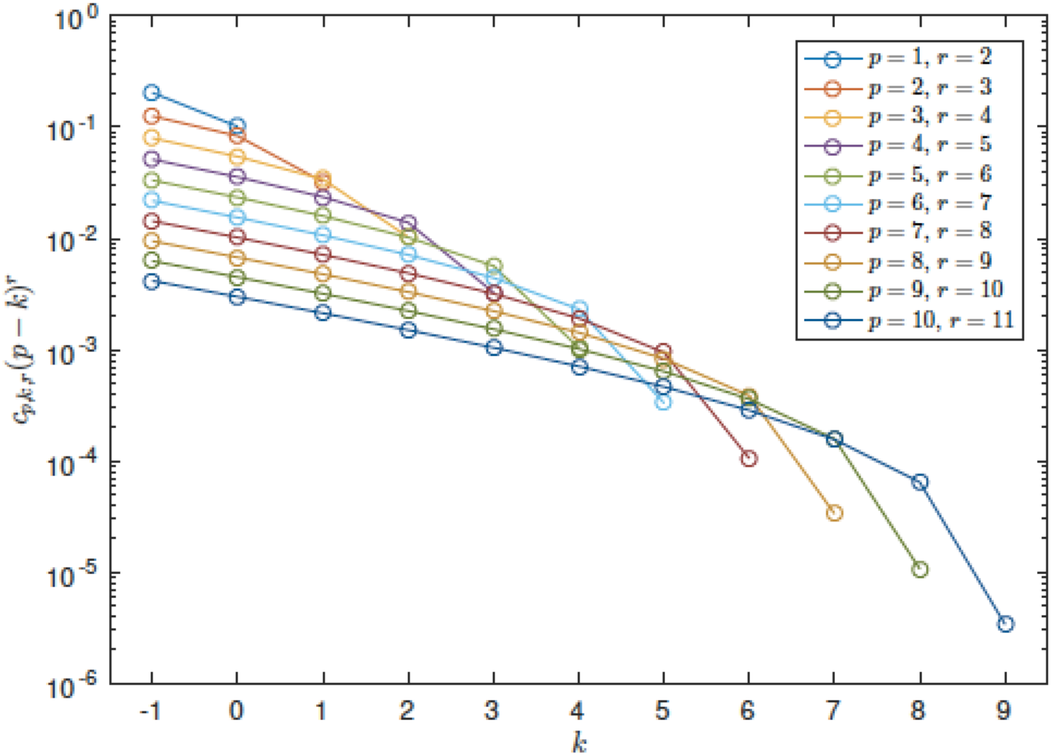
Why IGA?

✓ **Higher Accuracy per DOF**

- Spline bases are **smoother** than traditional FEM basis (e.g., C1, C2 continuity).
- Captures complex solution features **more accurately** with **fewer elements**.
- Ideal for problems involving **waves, sharp gradients**, or **geometry-sensitive physics**.

✓ **Built-in High-Order**

- Naturally supports **high-order polynomial degrees** without loss of stability.

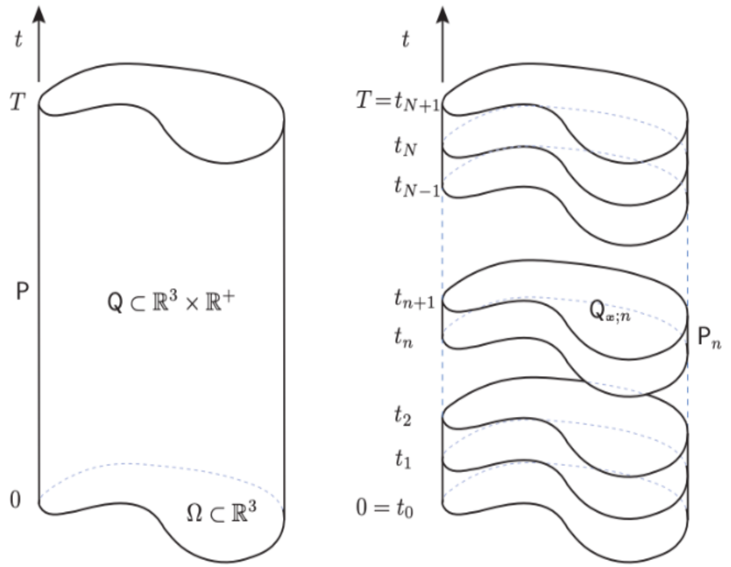


Espen Sande, Carla Manni and Hendrik Speleers, "Explicit error estimates for spline approximation of arbitrary smoothness in isogeometric analysis," *Numerische Mathematik,* vol. 144, 889-929, 2020.

BYU

Why Space-Time Finite Element Methods?

Feature	Finite Difference (FD) in Time	CG in Time	DG in Time
Time marching	✅ Yes	❌ No	✅ Yes
Accuracy (in time)	Typically, Low (1st or 2nd order)	Arbitrarily high depending on FEM choice (3rd order typical)	✅ Arbitrarily high depending on FEM choice (3rd order typical)
Parallelism in time	❌ Sequential	✅ Parallelizable	❌ Sequential



Hughes et al., 2010

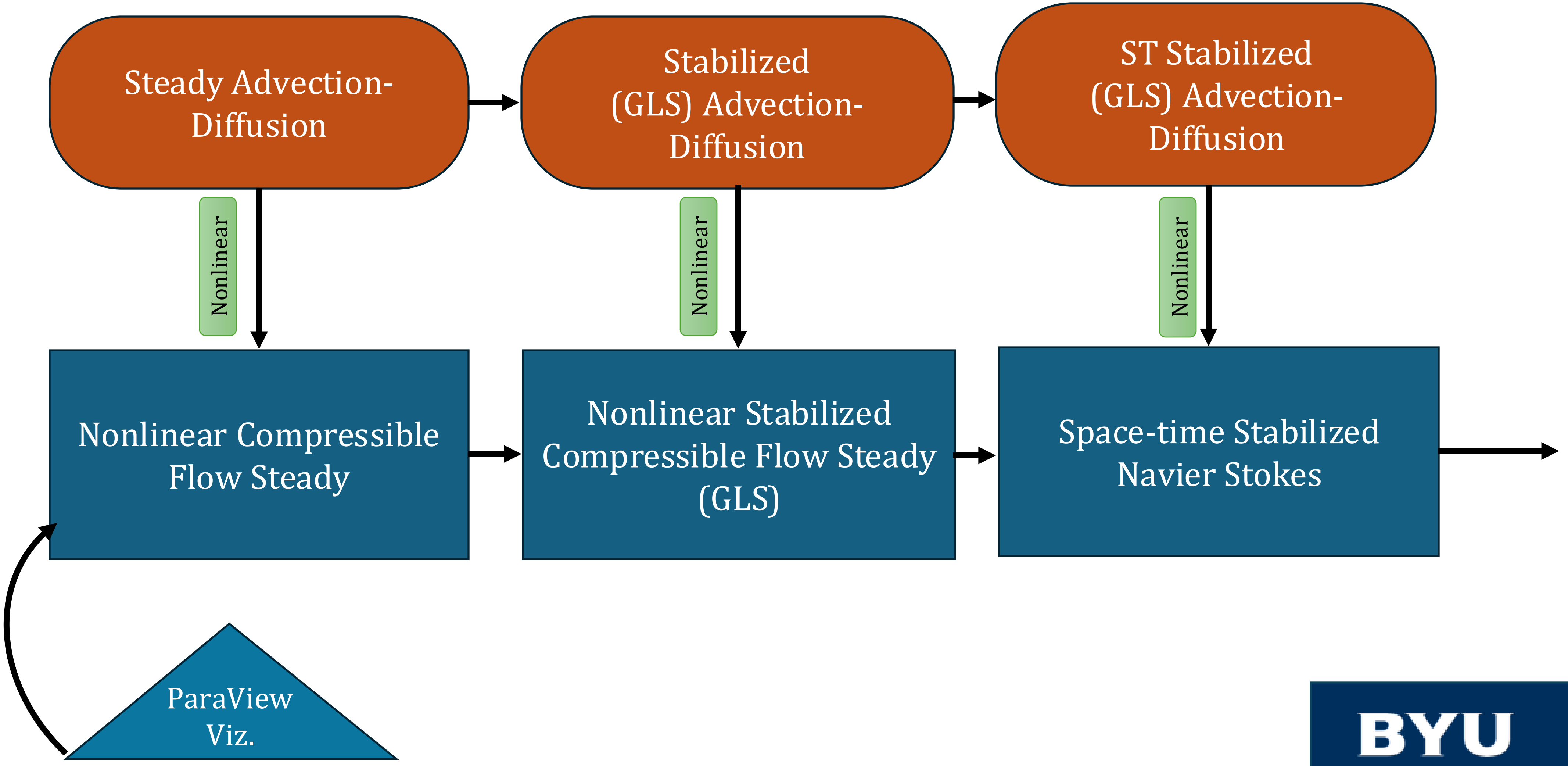
BYU

Integrated High-Performance Strategy with MFEM

By combining **MFEM + IGA + Space-Time FEM**, we create a powerful, scalable framework for solving advanced compressible flow and Multiphysics problems—**efficiently and accurately** on modern parallel computing architectures.

- ✓ *MFEM provides the engine and infrastructure.*
- ✓ IGA optimizes accuracy, stability, and resource use.
- ✓ Space-time FEM adds stability, accuracy, and flexibility in transient, nonlinear problems.

Roadmap



Steady Advection-Diffusion
Stabilized (GLS) Advection-Diffusion
ST Stabilized (GLS) Advection-Diffusion
Nonlinear
Nonlinear
Nonlinear
Nonlinear Compressible Flow Steady
Nonlinear Stabilized Compressible Flow Steady (GLS)
Space-time Stabilized Navier Stokes
ParaView Viz.

BYU

Compressible Navier-Stokes from Advection-Diffusion

Advection Diffusion

$$\frac{\partial u}{\partial t} + \boldsymbol{v} \cdot \nabla u - \nabla \cdot (\boldsymbol{\kappa} \nabla u) = f \text{ in } \Omega \times (0, T)$$

$$u = u_0 \text{ in } \Omega \times \{0\}$$

$$u = g \text{ on } \Gamma_g \times (0, T)$$

$$-v_n^- + (\kappa \nabla u) \cdot \boldsymbol{n} = \hbar \text{ on } \Gamma_\hbar \times (0, T),$$

Compressible Navier-Stokes

$$\frac{\partial \boldsymbol{U}}{\partial t} + \boldsymbol{\nabla} \cdot \boldsymbol{F} - \boldsymbol{\nabla} \cdot \boldsymbol{E} - \boldsymbol{R} = 0$$

$$\frac{\partial \boldsymbol{U}}{\partial t} + \boldsymbol{A}_i \frac{\partial \boldsymbol{U}}{\partial x_i} - \frac{\partial}{\partial x_i}\left(\boldsymbol{K}_{ij} \frac{\partial \boldsymbol{U}}{\partial x_j}\right) - \boldsymbol{R} = \boldsymbol{0},$$

$$\boldsymbol{U} = \begin{pmatrix} \rho \\ \rho \boldsymbol{v}_1 \\ \rho \boldsymbol{v}_2 \\ \rho \boldsymbol{v}_3 \\ \rho e \end{pmatrix} \quad \boldsymbol{F}_i = \begin{pmatrix} \boldsymbol{v}_i \rho \\ \boldsymbol{v}_i \rho \boldsymbol{v}_1 + \delta_{i1} p \\ \boldsymbol{v}_i \rho \boldsymbol{v}_2 + \delta_{i2} p \\ \boldsymbol{v}_i \rho \boldsymbol{v}_3 + \delta_{i3} p \\ \boldsymbol{v}_i (\rho e + p) \end{pmatrix} \quad \boldsymbol{E}_i = \begin{pmatrix} 0 \\ \boldsymbol{T}_{i1} \\ \boldsymbol{T}_{i2} \\ \boldsymbol{T}_{i3} \\ -\boldsymbol{q}_i + \boldsymbol{T}_{ik} \boldsymbol{v}_k \end{pmatrix} \quad \boldsymbol{R} = \rho \begin{pmatrix} 0 \\ b_1 \\ b_2 \\ b_3 \\ b_i \boldsymbol{v}_i + r \end{pmatrix},$$

Compressible Navier-Stokes (Entropy Variables):

$$\tilde{\boldsymbol{A}}_0 \frac{\partial \boldsymbol{V}}{\partial t} + \tilde{\boldsymbol{A}}_i \frac{\partial \boldsymbol{V}}{\partial x_i} - \frac{\partial}{\partial x_i}\left(\tilde{\boldsymbol{K}}_{ij} \frac{\partial \boldsymbol{V}}{\partial x_j}\right) - \boldsymbol{R} = \boldsymbol{0},$$

$$\boldsymbol{V} = \frac{\partial H}{\partial \boldsymbol{U}} = \frac{1}{\rho \varepsilon} \begin{pmatrix} -U_5 + \rho \varepsilon (\gamma + 1 - s + s_0) \\ U_2 \\ U_3 \\ U_4 \\ -U_1 \end{pmatrix},$$

BYU

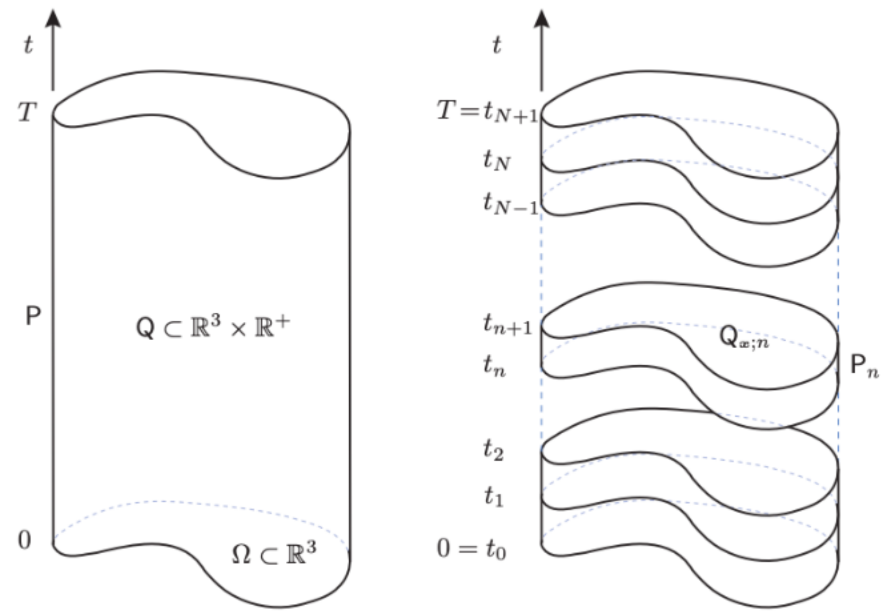
Space-Time Discretization

Advection Diffusion

$$B(w,u) = \int_Q -\frac{\partial w}{\partial t} u - \nabla w \cdot (\boldsymbol{v} u) + \nabla w \cdot (\boldsymbol{\kappa} \nabla u) dQ + \int_{P_h} w u v_n^+ dP_h + \int_\Omega w|_{t=T} u|_{t=T} d\Omega$$

$$L(w) = \int_Q w f dQ + \int_{P_h} w h dP_h + \int_\Omega w|_{t=0} u|_{t=0} d\Omega.$$

$$N_A(\boldsymbol{x}, t) = B_{\hat{A}}(\boldsymbol{x}) B_{\tilde{A}}(t),$$



Hughes et al., 2010

$$\int_Q -\nabla N_A \cdot (\boldsymbol{v} N_B) dQ = \int_0^T \int_\Omega \nabla (B_{\hat{A}} B_{\tilde{A}}) \cdot (\boldsymbol{v} B_{\hat{B}} B_{\tilde{B}}) d\Omega dt$$
$$= \int_0^T B_{\tilde{A}} B_{\tilde{B}} \int_\Omega \nabla B_{\hat{A}} \cdot (\boldsymbol{v} B_{\hat{B}}) d\Omega dt.$$

$$\begin{bmatrix} \boldsymbol{K}_{00} & \boldsymbol{K}_{01} & \dots & \boldsymbol{K}_{0q} \\ \boldsymbol{K}_{10} & \boldsymbol{K}_{11} & \dots & \boldsymbol{K}_{1q} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{K}_{q0} & \boldsymbol{K}_{q1} & \dots & \boldsymbol{K}_{qq} \end{bmatrix} \begin{bmatrix} \boldsymbol{d}_0 \\ \boldsymbol{d}_1 \\ \vdots \\ \boldsymbol{d}_q \end{bmatrix} = \begin{bmatrix} \boldsymbol{F}_0 \\ \boldsymbol{F}_1 \\ \vdots \\ \boldsymbol{F}_q \end{bmatrix}$$

$$\hat{K}_{\tilde{A}\tilde{B};\hat{A}\hat{B}}(t) = \frac{\partial B_{\tilde{A}}}{\partial t} B_{\tilde{B}} \int_\Omega B_{\hat{A}} B_{\hat{B}} d\Omega - B_{\tilde{A}} B_{\tilde{B}} \int_\Omega \nabla B_{\hat{A}} \cdot (\boldsymbol{v} B_{\hat{B}}) d\Omega + B_{\tilde{A}} B_{\tilde{B}} \int_\Omega \nabla B_{\hat{A}} \cdot (\boldsymbol{\kappa} \nabla B_{\hat{B}}) d\Omega$$
$$+ B_{\tilde{A}} B_{\tilde{B}} \int_{\Gamma_h} B_{\hat{A}} B_{\hat{B}} v_n^d \Gamma + B_{\tilde{A}}|_{t=T} B_{\tilde{B}}|_{t=T} \int_\Omega B_{\hat{A}} B_{\hat{B}} d\Omega \quad (138)$$

BYU

Steady Advection Diffusion Equations

The Steady Advection Diffusion that we are interested in solving is:

$V . \nabla u - \nabla.(k \nabla u) = f \text{ in } \Omega$

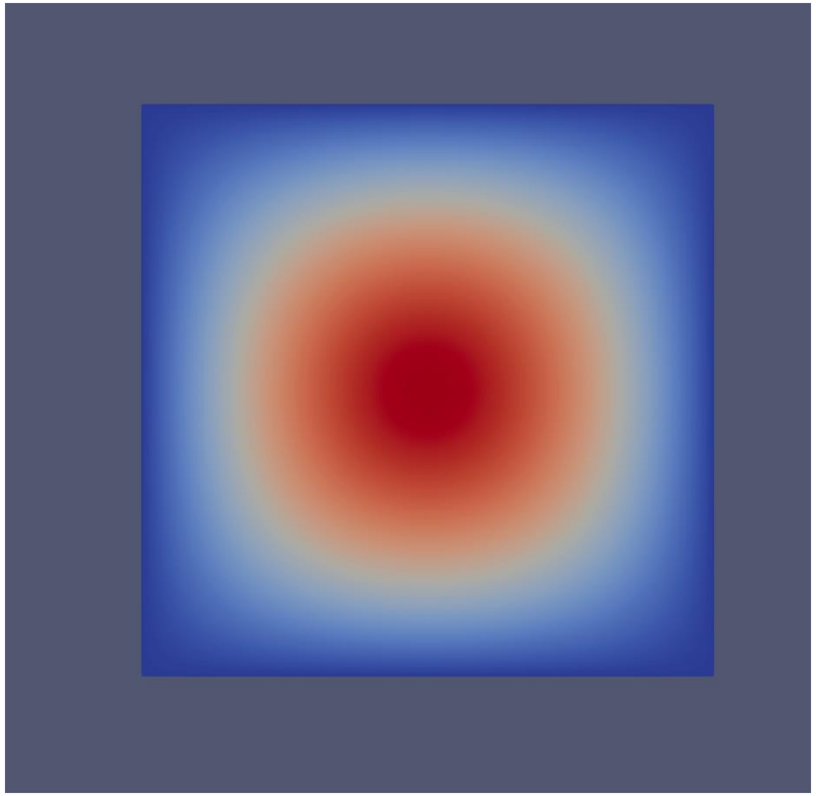
$u = g \text{ on } \Gamma_g$

Exact Solution: $U(x) = \sin(\beta x) * \sin(\beta y) + g$

$f = k * (2\,\beta * \sin(\beta x) * \sin(\beta y)) + v(x) * \beta * \cos(\beta x) * \sin(\beta y) + v(y) * \beta * \sin(\beta x) * \cos(\beta y)$

$V(x) = 100,\ V(y) = 0$

Mesh size	L2 Error	L2 Rate	H1 Error	H1 Rate
0.25	0.0187753	0	0.511981	0
0.125	0.00420775	2.15771	0.253011	1.01689
0.0625	0.00103452	2.02409	0.126061	1.00508
0.03125	0.000257731	2.00503	0.0629753	1.00126

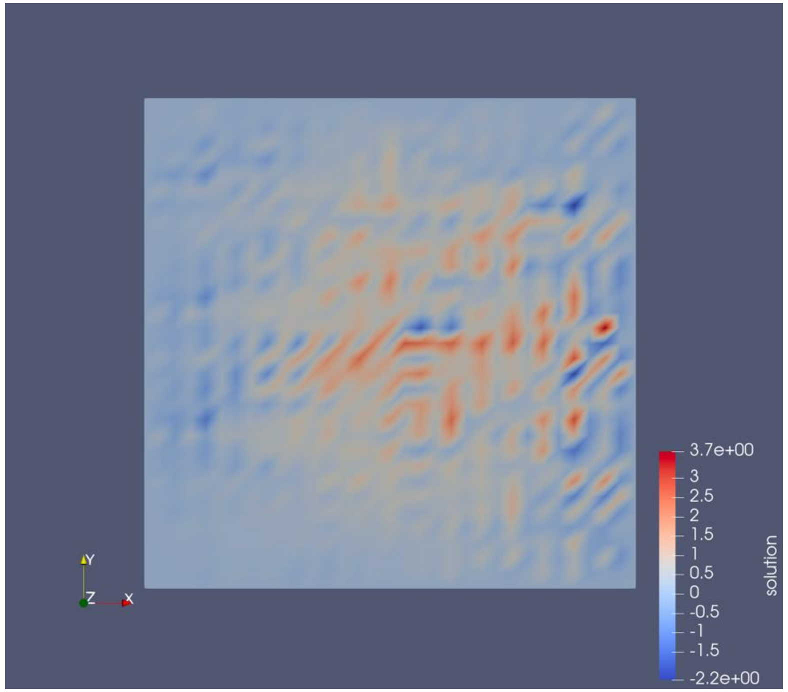


BYU

GLS Stabilization Impact

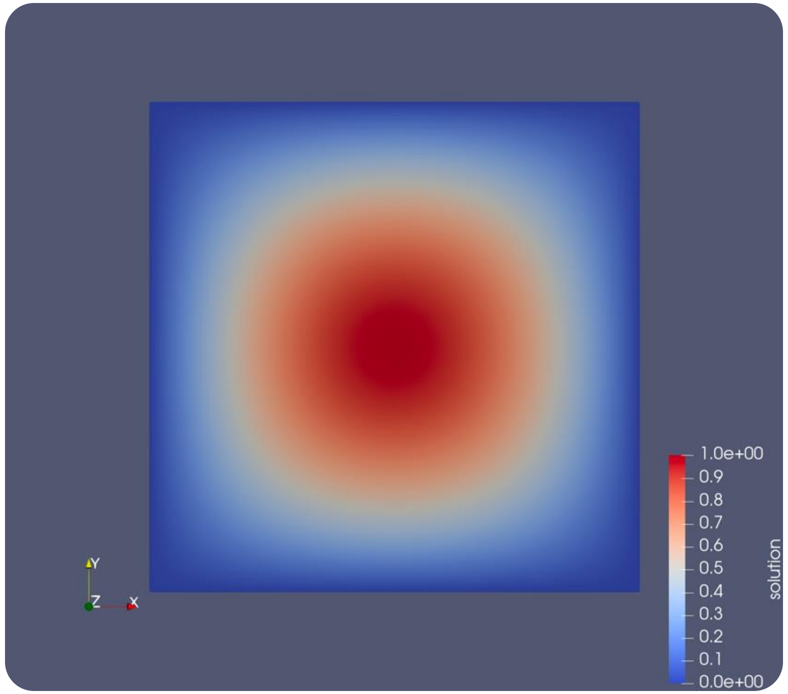
Without Stabilization

Total Refinement = 4
Serial Refinement = 4
Iteration Number = 900
No Convergence



With GLS Stabilization

Total Refinement = 4
Serial Refinement = 4
Iteration Number = 16



Mesh size	L2 Error	L2 Rate	H1 Error	H1 Rate
0.25	12516.2	0	112265	0
0.125	6.70046	10.8672	127.552	9.7816
0.0625	12.4654	-0.895598	485.702	-1.92898
0.03125	0.341456	5.19009	27.738	4.13013

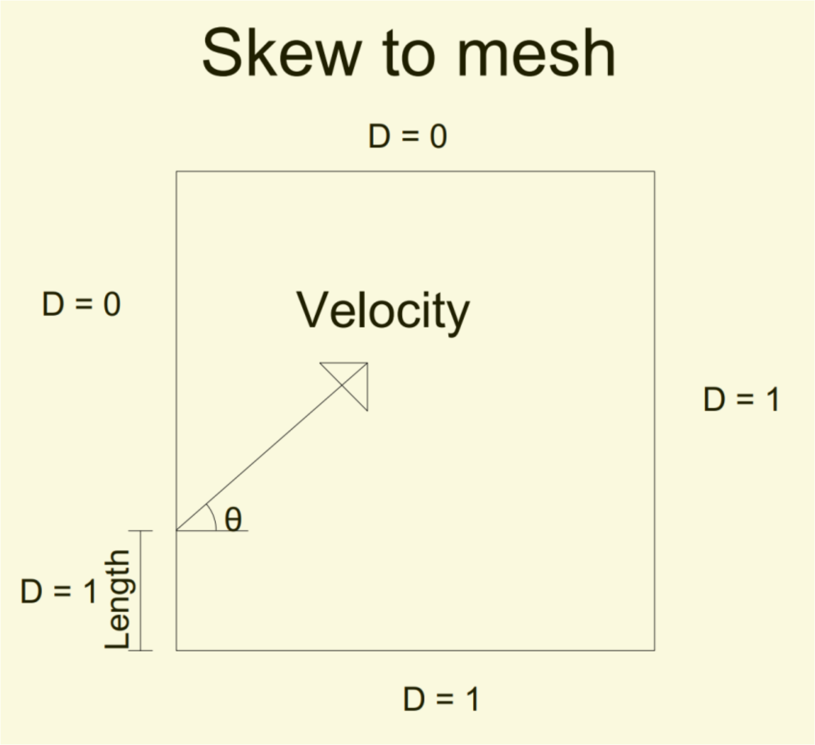
Mesh size	L2 Error	L2 Rate	H1 Error	H1 Rate
0.25	0.023214	0	0.505617	0
0.125	0.00478431	2.27861	0.252374	1.00248
0.0625	0.0011332	2.07791	0.125983	1.00233
0.03125	0.000301709	1.90917	0.0629614	1.00069

BYU

Skew To Mesh

Velocity = [500, 500], κ = 0.01

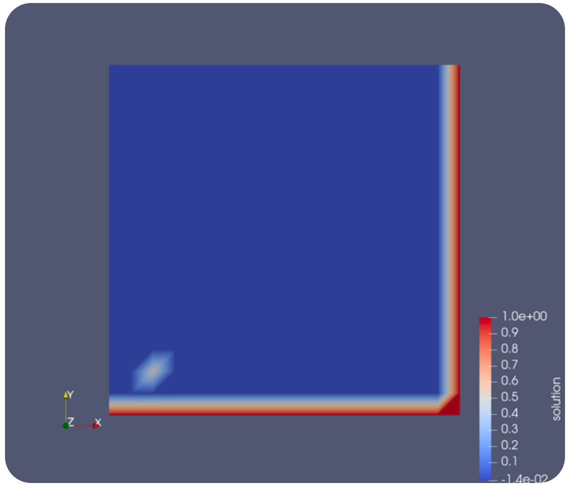
Length = 0, F = 0, *Ω = [0,L]*

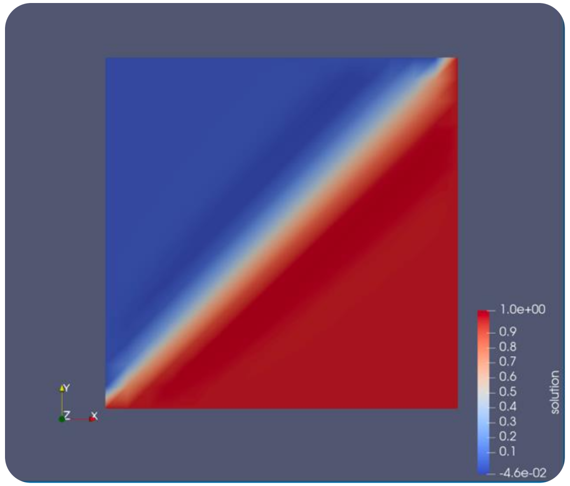


Without Stabilization

With GLS Stabilization

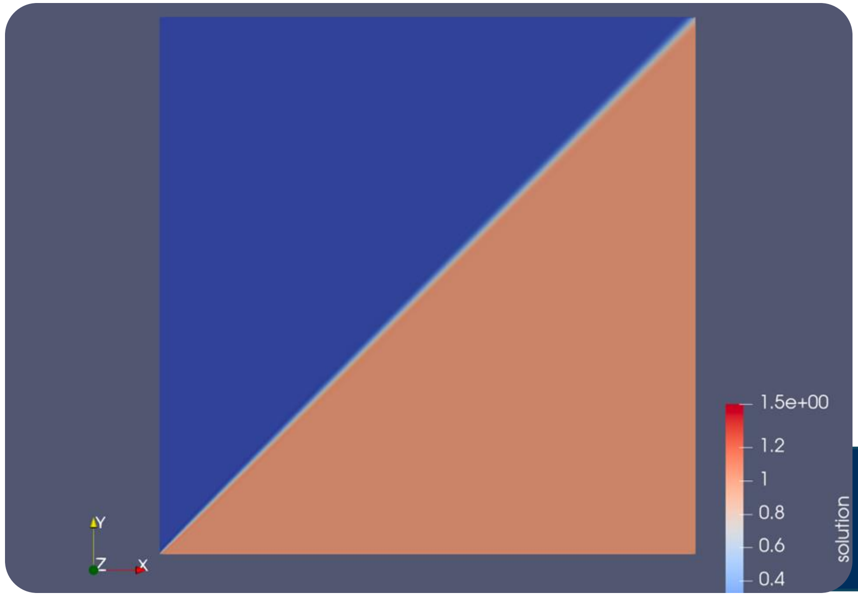
h = 6.25 e-2
Total Refinement = 2



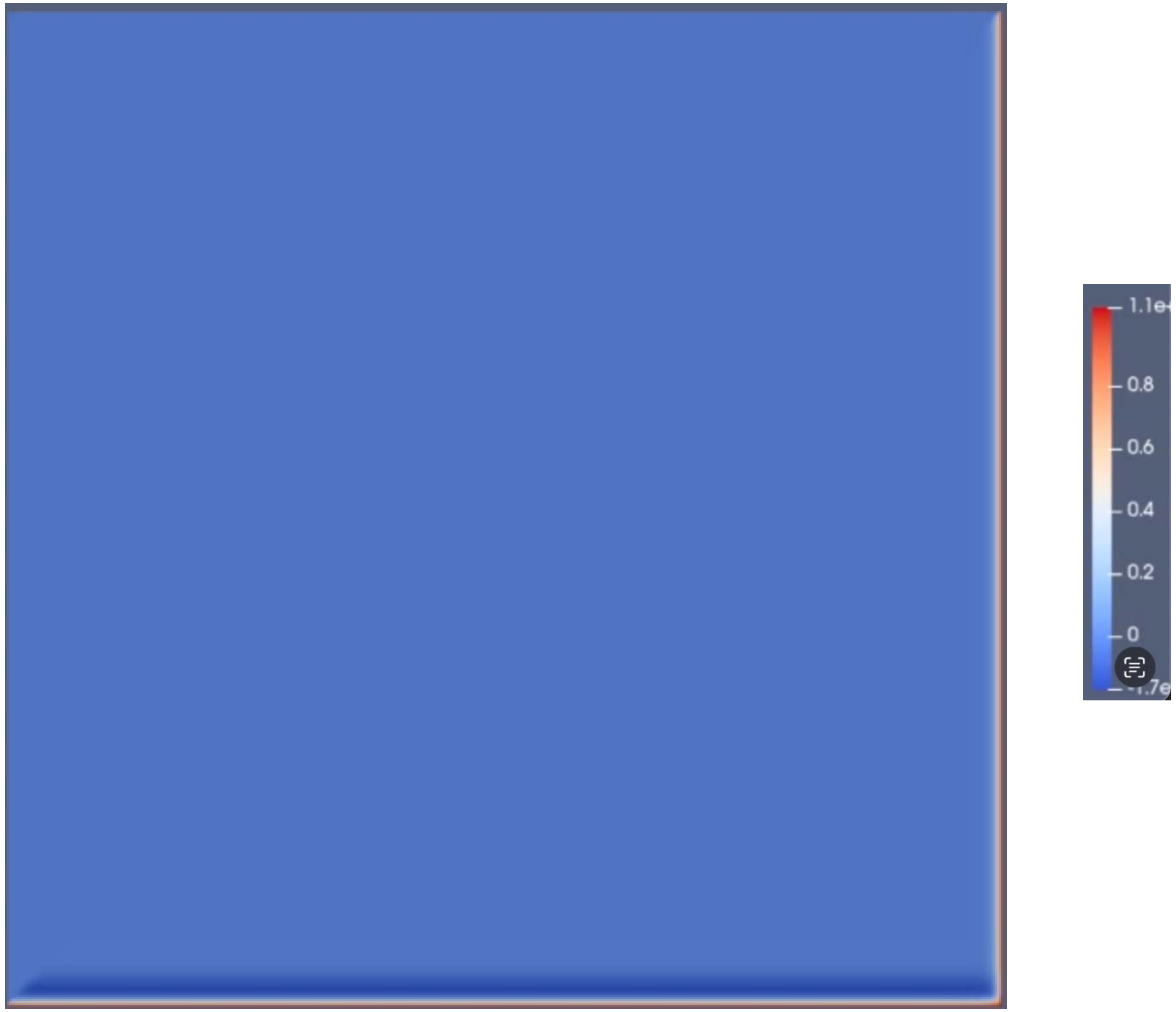


h = 6.25 e-2
Total Refinement = 8

With GLS Stabilization



Skew to Mesh for Spacetime Solver



BYU

Implementation Challenges: From Advection-Diffusion to Compressible Navier-Stokes

✓ Step 1: Success with Space-Time Advection-Diffusion

- Implemented space-time FEM for advection-diffusion
- Achieved **stable and accurate results**
- Validated numerical setup, mesh handling, and solver behavior

1. Creating Local Forms in MFEM

- MFEM has standard local integrators (e.g., $w \cdot \nabla u$)
- But compressible Navier-Stokes requires custom forms like: $(u \cdot \nabla w)$
- Advection-diffusion → 1 variable (e.g., scalar u)
- Compressible Navier-Stokes → **4+ variables** (think of density, velocity components, temperature)

✓ **Solution**: Implemented custom local integrators to define correct physics.

2. Manual Space-Time Integration in MFEM

- MFEM provides strong support for spatial finite element methods,
  but **no native functionality** for **fully-coupled space-time integration**.

✓ **Solution**: Manual Space-Time Formulation

1. Separated spatial and temporal components:
2. Custom Integration
3. Manual Assembly

✓ Enabled high-order space-time formulation despite MFEM limitations

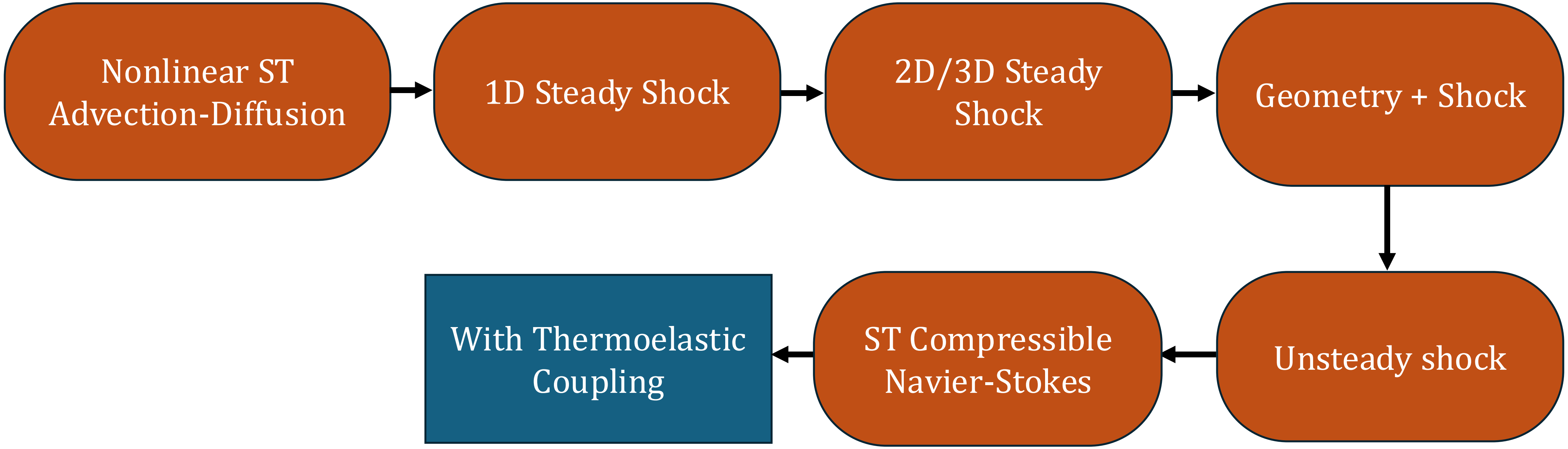
3. Complex Boundary Conditions

- Advection-diffusion: simpler Dirichlet/Neumann BCs
- Compressible flow: **nonlinear**, physically coupled BCs (e.g., inflow/outflow)

✓ **Solution**: Required more careful implementation and solver.

BYU

Path Forward



Additional Challenges we're facing …

- Step 2: Moving to Compressible Navier-Stokes
  - Introduced **new complexities and challenges**:

***1. Switching from Multi-Corrector (Shakib et al., 1989) to Newton Solver* Challenges Faced:**

Newton requires:

Consistent and smooth **residual** formulation

Compressible NS system is **strongly nonlinear**, with:

Coupled variables (ρ, u, E)

Nonlinear boundary terms

- *Even small mistakes in the residual blow up the solver. Still in progress.*

2. Applying Boundary Conditions in MFEM

In MFEM, boundary conditions are applied through FormLinearSystem(...),
which modifies the global stiffness matrix **K** and RHS vector **F** using constraints.

The Issue:

Repeated errors from EliminateRowsCols(...):
"Tried to eliminate rows/cols that should exist, but MFEM couldn't find them."
Debugging showed that all indices were present, but the process **still failed.**

✓ **Solution:**

Skipped FormLinearSystem, and instead applied boundary conditions **manually**:

Zeroed out relevant rows/columns

Set diagonal to 1

Adjusted the RHS accordingly

- *Manual application gave me more control and worked reliably.*

BYU

Conclusion:

- Integration of MFEM, IGA, and Space-Time FEM
  - *Combining MFEM + IGA + Space-Time FEM creates a powerful pathway toward high-performance simulation of complex PDEs—though it requires deep customization, especially when transitioning to fully nonlinear, compressible systems.*
- Validation on Benchmark Problems:
  - The developed framework has been successfully validated on linear and nonlinear advection-diffusion problems. These results confirm the accuracy, stability, and convergence of the method, providing confidence in its extension to more complex physics.
- Toward Space-Time Multiphysics Compressible Flow
  - Our objective is to develop Multiphysics fluid simulations based on space-time compressible flow. We have outlined the motivation and approach, and we hope to complete the work and publish the results soon.

BYU

Questions? / Feedback! / Advice!

E-mail: rpahnabi@byu.edu

BYU