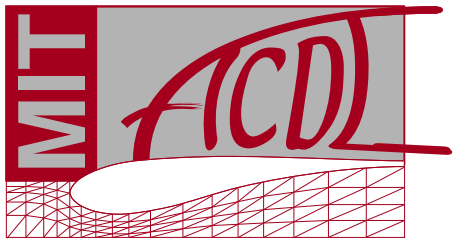
High Order Discontinuous Galerkin Methods for Computational Mechanics

Per-Olof Persson

Department of Mathematics
Massachusetts Institute of Technology

Collaborators: J. Peraire, J. Bonet, M. Drela, D. J. Willis,
N. C. Nguyen, E. Israeli, A. Uranga, A. Mohnot, J. P. Whitney



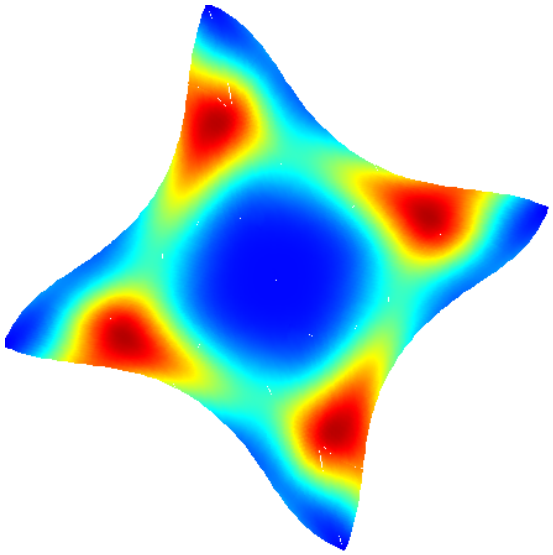
Department of Mechanical Engineering
Massachusetts Institute of Technology
March 18, 2008

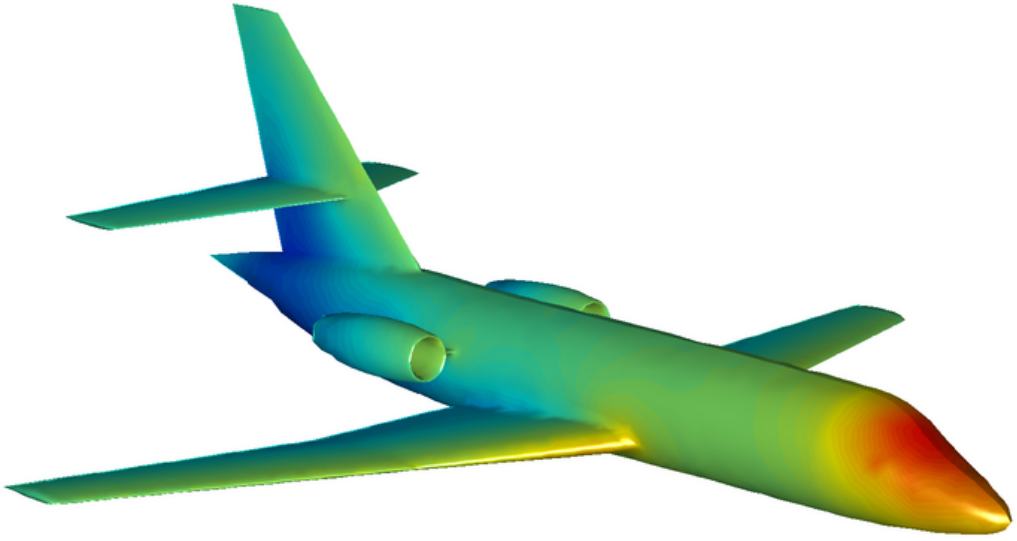
Outline

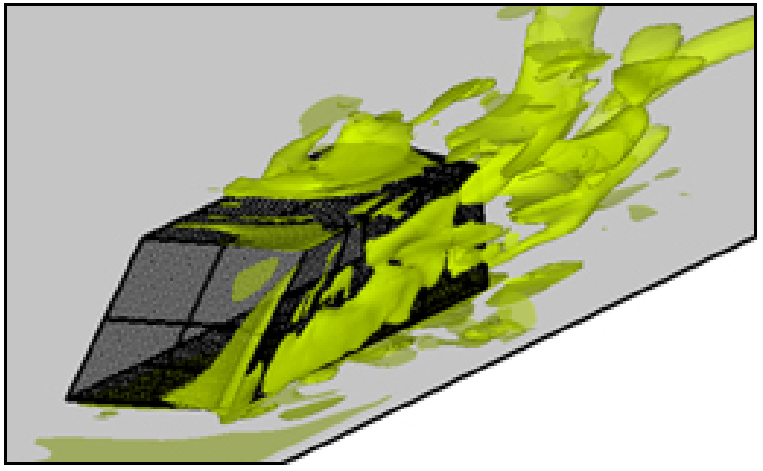
1. Introduction
2. Discretization and Solvers
   - Mesh Generation
   - The Discontinuous Galerkin Method
   - The CDG Method, Preconditioning, and Stabilization
3. Large Deformation Solid Dynamics
   - High-Order Lagrangian DG Formulations
   - Volumetric Modeling of Beams, Shells, and Membranes
   - J2 Viscoplasticity
4. Deformable Domains
   - Mapping-based ALE Formulation
   - Flapping Flight Applications
5. Conclusions

Motivation

- *Need for higher fidelity predictions in computational mechanics*
  - Wave propagation
  - Multiple scale phenomena
  - Non-linear interactions
  - Multiphysics couplings
- Applications: Acoustics, turbulence, shocks, instabilities, fluid-structure interaction, ...

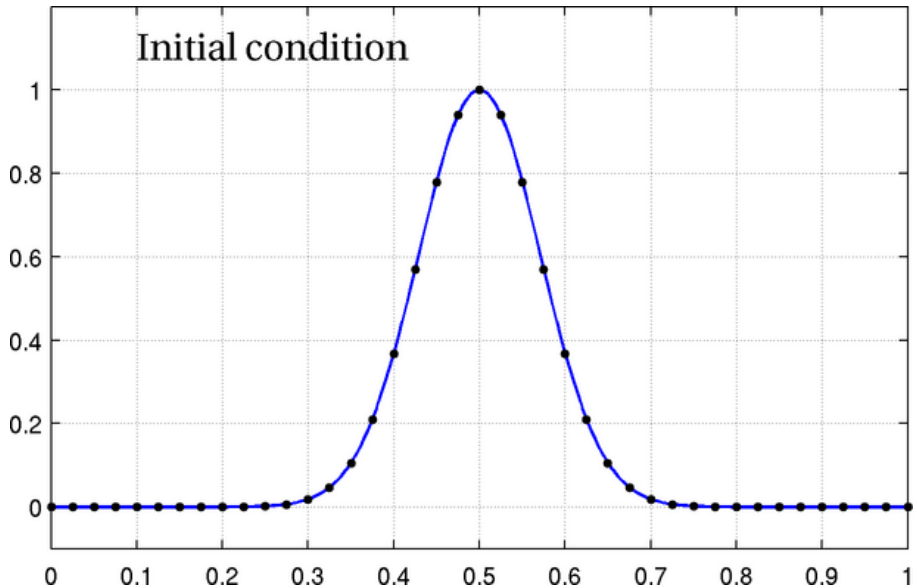


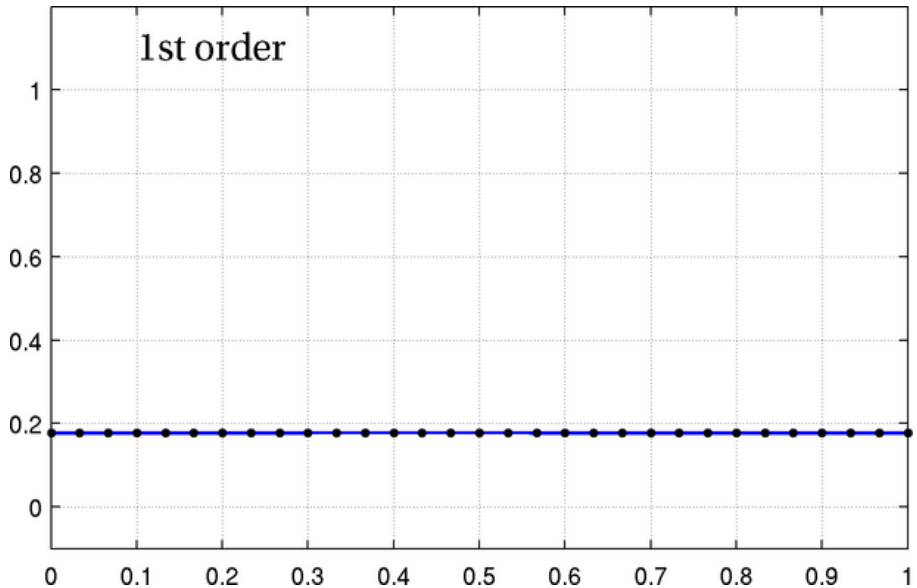


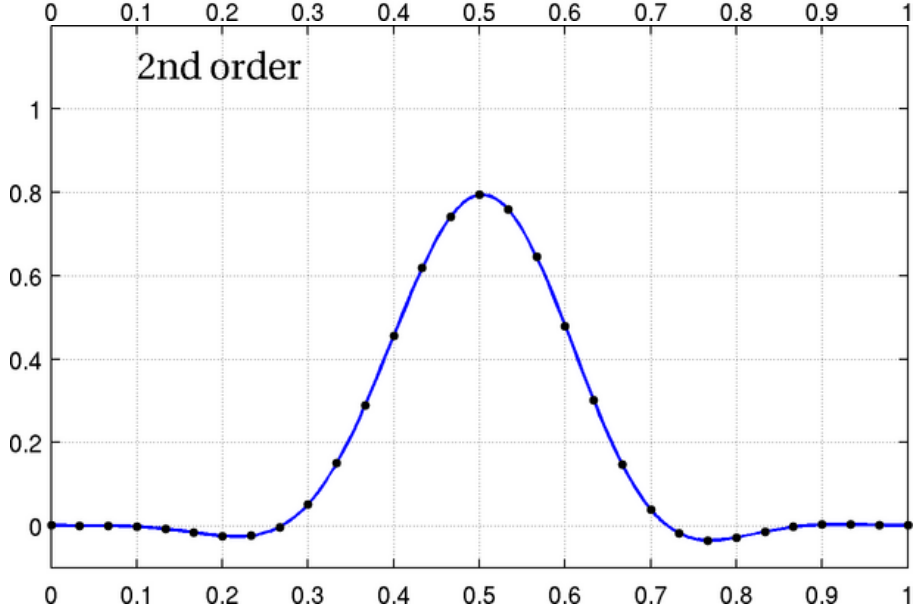


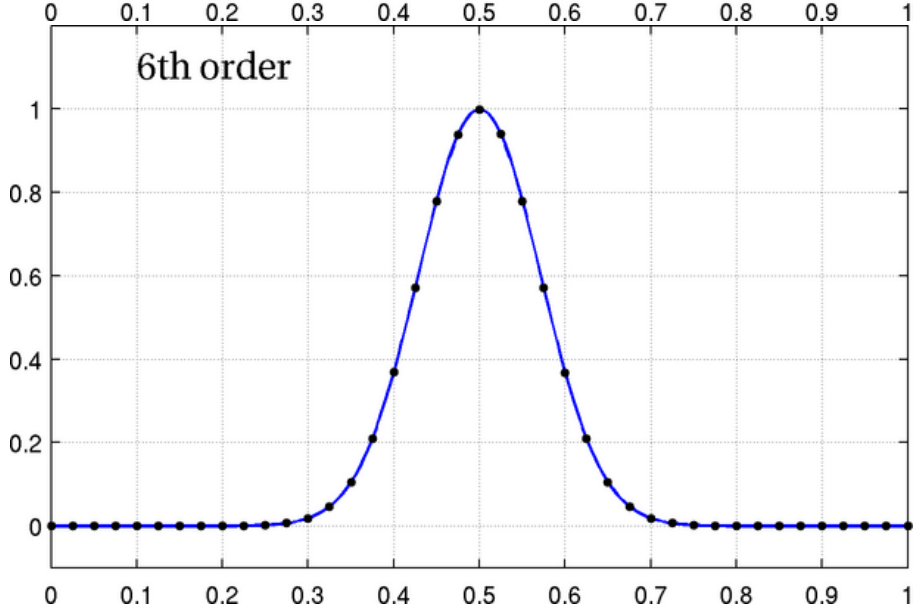
Example: Wave propagation

- Scalar convection equation $u_t + u_x = 0$, fixed resolution





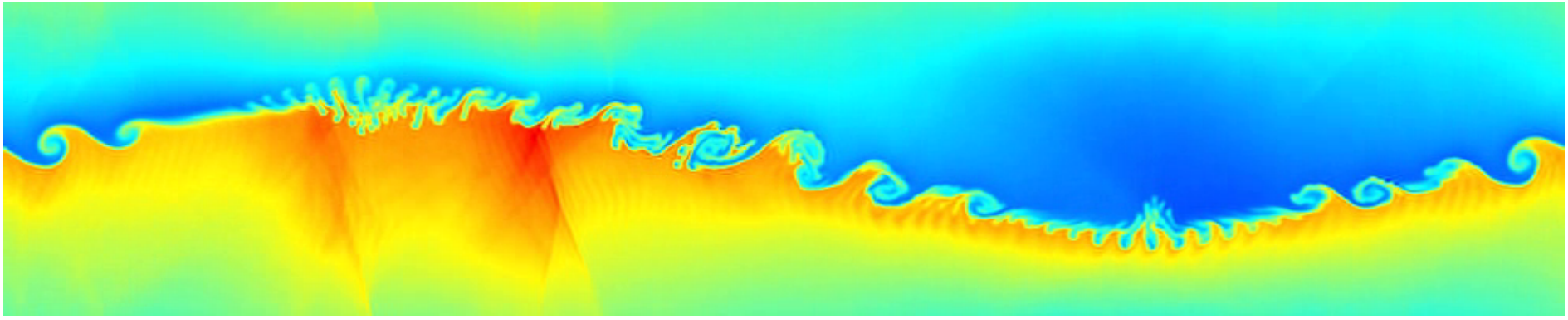




Example: Aeroacoustics and K-H Instability

- Inspired by calculations of Munz et al [03] (linearized)
- Nonlinear behavior: Large scale acoustic wave interacts with small scale flow features, leading to vorticity generation
- $p = 7$ (8th order accuracy), 140-by-28 square elements

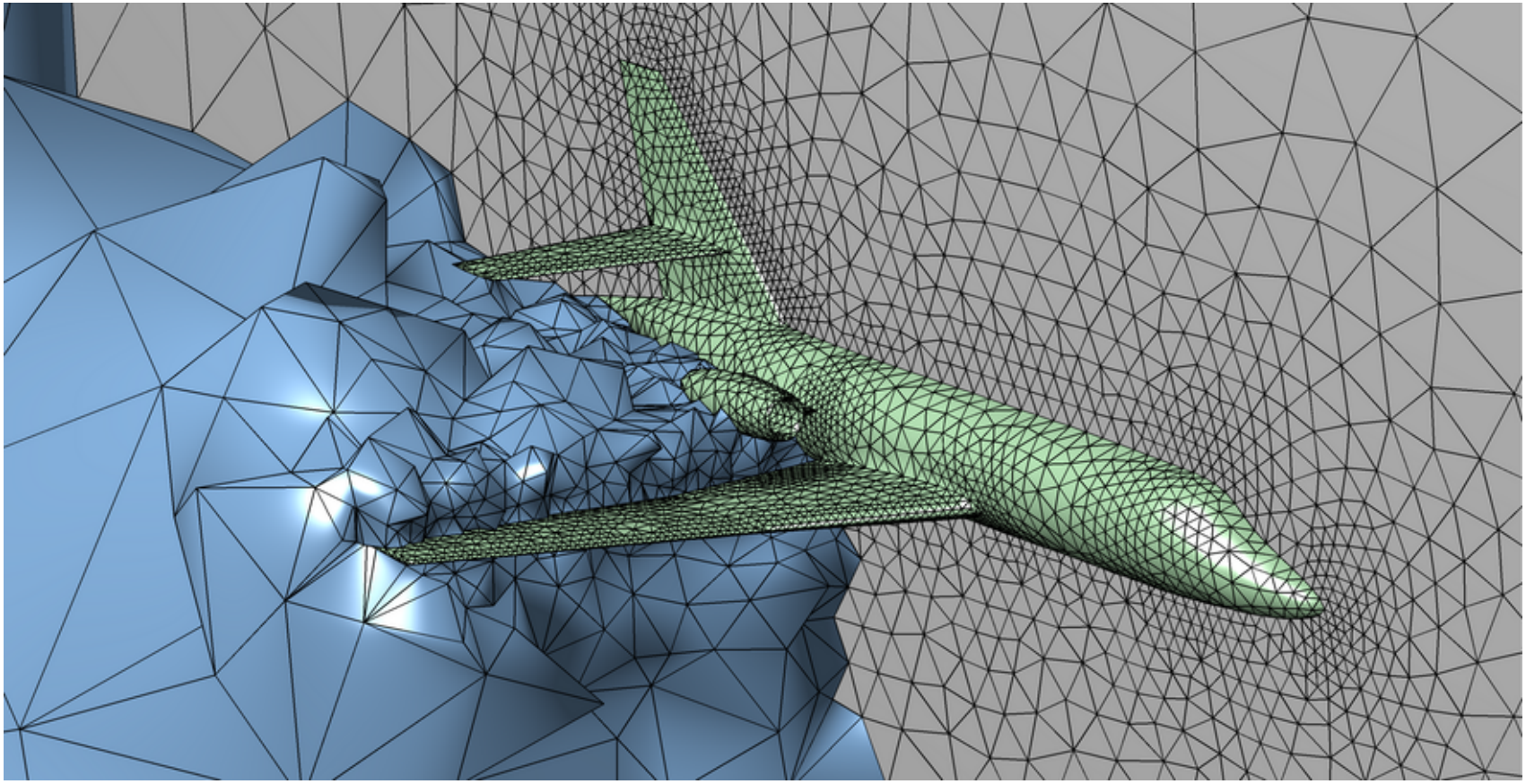
Density (blue=low, red=high)



Mesh Generation for Complex Geometries

Unstructured meshes *required for realistic problems*

- Geometric and adaptive flexibility, accurate boundary treatment
- Tetrahedral meshes give robustness and automation



Motivation

- Fundamental properties of Discontinuous Galerkin (DG) methods:

	FVM	FDM	FEM	DG
1) High-order/Low dispersion	✗	✓	✓	✓
2) Unstructured meshes	✓	✗	✓	✓
3) Stability for conservation laws	✓	✓	✗	✓

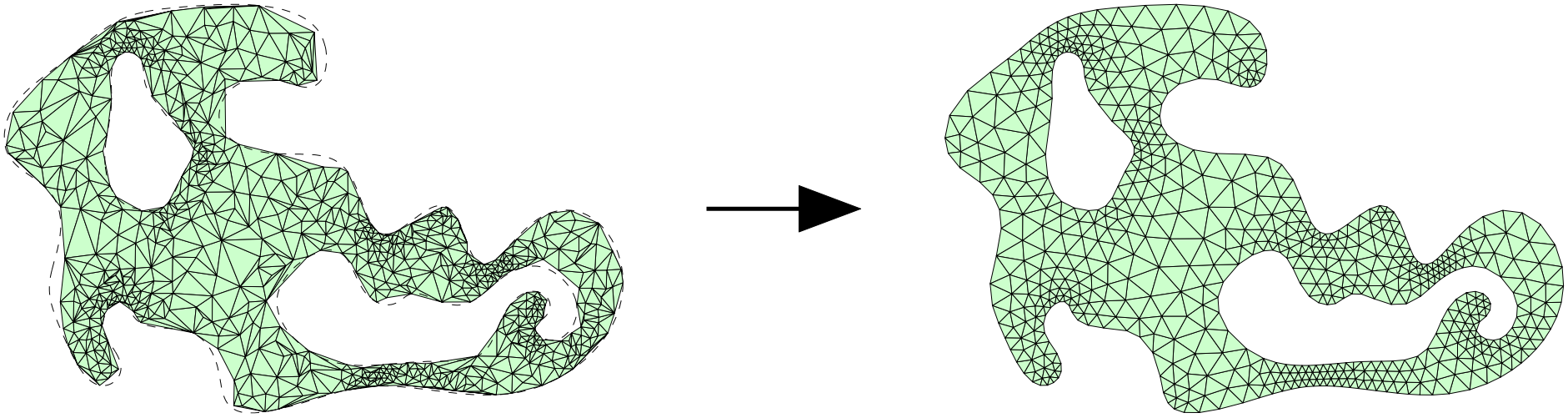
- However, several problems to resolve:
  - High CPU/memory requirements (compared to FVM or H-O FDM)
  - Low tolerance to under-resolved features
  - High-order geometry representation and mesh generation
- *The challenge is to make DG competitive for real-world problems*

Outline

1. Introduction
2. Discretization and Solvers
   - Mesh Generation
   - The Discontinuous Galerkin Method
   - The CDG Method, Preconditioning, and Stabilization
3. Large Deformation Solid Dynamics
   - High-Order Lagrangian DG Formulations
   - Volumetric Modeling of Beams, Shells, and Membranes
   - J2 Viscoplasticity
4. Deformable Domains
   - Mapping-based ALE Formulation
   - Flapping Flight Applications
5. Conclusions

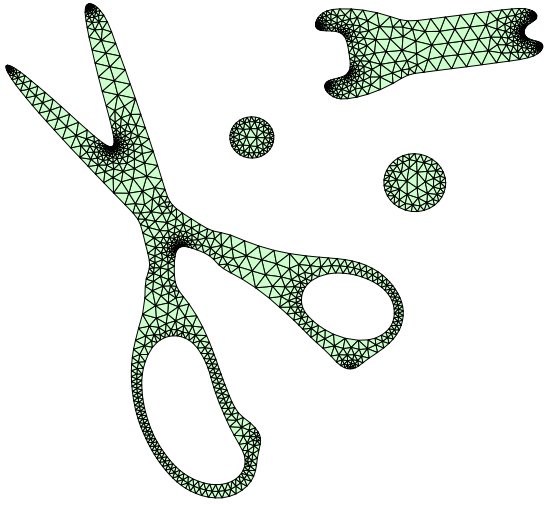
The DistMesh Mesh Generator

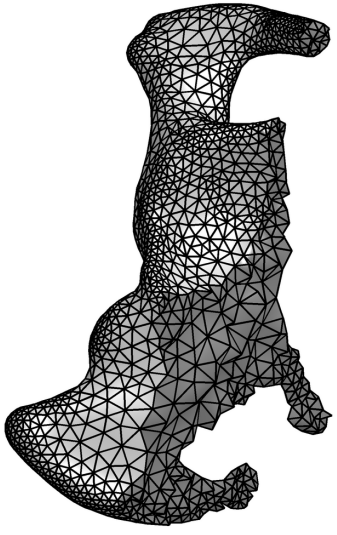
- High quality meshes obtained using the *DistMesh* algorithm [Persson Ph.D. thesis, 2005]
  1. Start with *any* topologically correct initial mesh, for example random node distribution and Delaunay triangulation
  2. Move nodes to find force equilibrium in edges
     - Project boundary nodes using *implicit geometry* $\phi(\boldsymbol{x})$
     - Update element connectivities with Delaunay

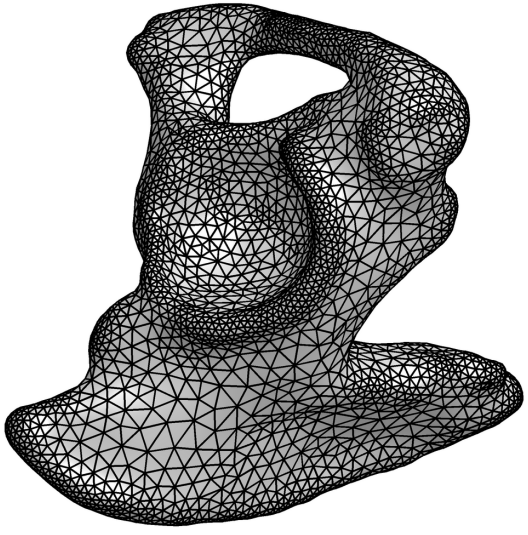


DistMesh Applications

- High quality meshes from images and MRI/CT scans



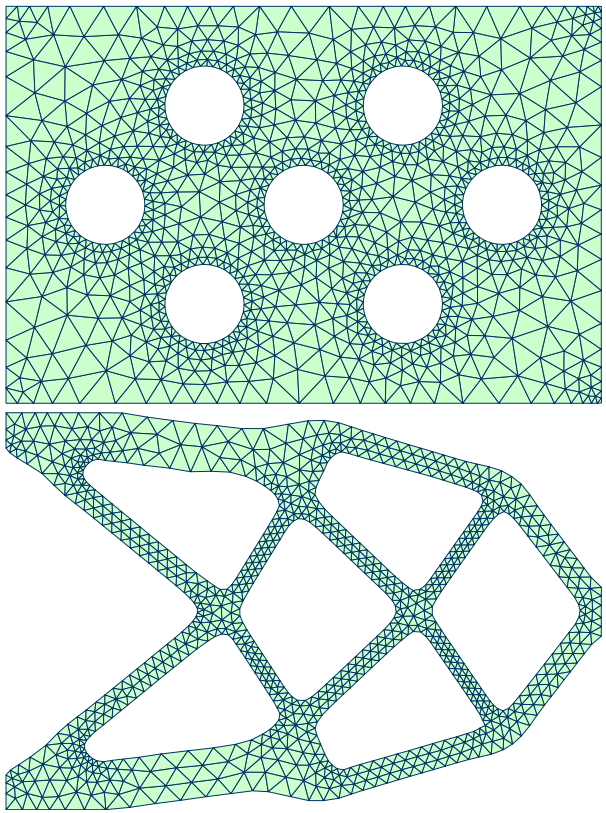




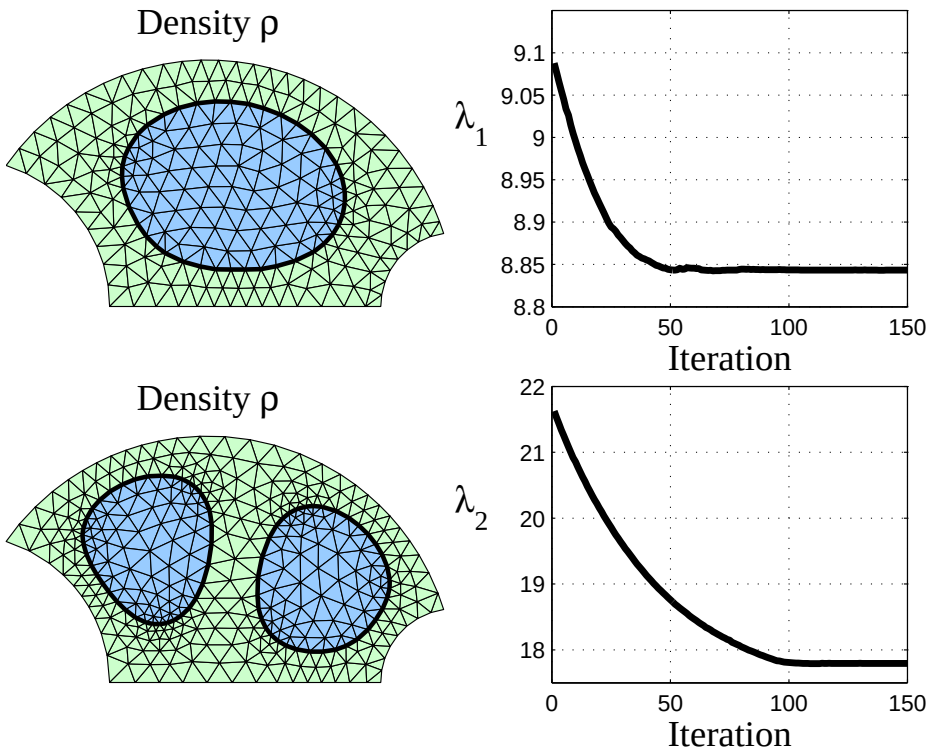
DistMesh Applications

- Shape optimization by combined levelset/finite element method

Structural design (compliance minimization)



Vibration control (eigenvalue minimization)



Outline

1. Introduction
2. Discretization and Solvers
   - Mesh Generation
   - The Discontinuous Galerkin Method
   - The CDG Method, Preconditioning, and Stabilization
3. Large Deformation Solid Dynamics
   - High-Order Lagrangian DG Formulations
   - Volumetric Modeling of Beams, Shells, and Membranes
   - J2 Viscoplasticity
4. Deformable Domains
   - Mapping-based ALE Formulation
   - Flapping Flight Applications
5. Conclusions

The Discontinuous Galerkin Method

- (Reed/Hill 1973, Lesaint/Raviart 1974, Cockburn/Shu 1989-, etc)
- Consider non-linear hyperbolic system in conservative form:

$$\boldsymbol{u}_t + \nabla \cdot \mathcal{F}_i(\boldsymbol{u}) = 0$$

- Triangulate domain Ω into elements $\kappa \in T_h$
- Seek approximate solution $\boldsymbol{u}_h$ in space of element-wise polynomials:

$$\mathcal{V}_h^p = \{\boldsymbol{v} \in L^2(\Omega) : \boldsymbol{v}|_\kappa \in P^p(\kappa)\ \forall \kappa \in T_h\}$$

- Multiply by test function $\boldsymbol{v}_h \in \mathcal{V}_h^p$ and integrate over element κ:

$$\int_\kappa \left[(\boldsymbol{u}_h)_t + \nabla \cdot \mathcal{F}_i(\boldsymbol{u}_h)\right] \boldsymbol{v}_h \, d\boldsymbol{x} = 0$$

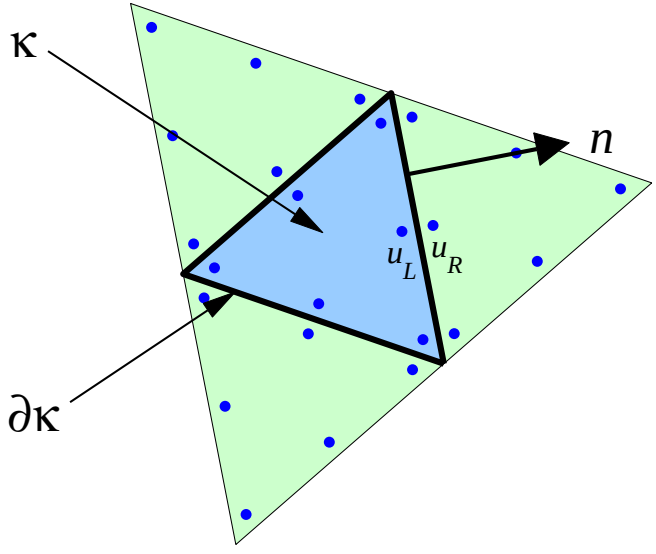
The Discontinuous Galerkin Method

- Integrate by parts:

$$\int_\kappa [(\boldsymbol{u}_h)_t]\, \boldsymbol{v}_h \, d\boldsymbol{x} - \int_\kappa \mathcal{F}_i(\boldsymbol{u}_h) \nabla \boldsymbol{v}_h \, d\boldsymbol{x} + \int_{\partial\kappa} \hat{\mathcal{F}}_i(\boldsymbol{u}_h^+, \boldsymbol{u}_h^-, \hat{\boldsymbol{n}}) \boldsymbol{v}_h^+ \, ds = 0$$

  with numerical flux function $\hat{\mathcal{F}}_i(\boldsymbol{u}_L, \boldsymbol{u}_R, \hat{\boldsymbol{n}})$ for left/right states $\boldsymbol{u}_L$,$\boldsymbol{u}_R$ in direction $\hat{\boldsymbol{n}}$ (Godunov, Roe, Osher, Van Leer, Lax-Friedrichs, etc)

- Global problem: Find $\boldsymbol{u}_h \in \mathcal{V}_h^p$ such that this weighted residual is zero for all $\boldsymbol{v}_h \in \mathcal{V}_h^p$
- Error $= \mathcal{O}(h^{p+1})$ for smooth solutions



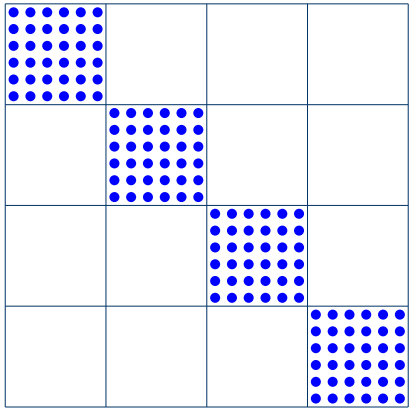
The DG Method – Observations

- Reduces to the standard FEM with continuous spaces
- Reduces to the standard FVM for $p = 0$:

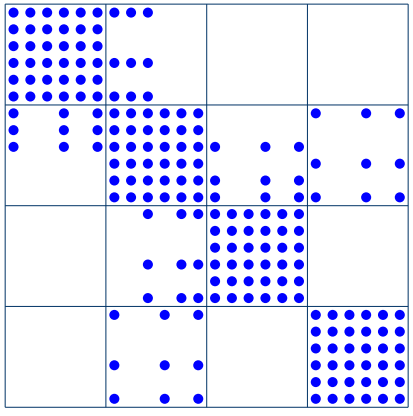
$$(\boldsymbol{u}_h)_t A_\kappa + \int_{\partial\kappa} \hat{\mathcal{F}}_i(\boldsymbol{u}_h^+, \boldsymbol{u}_h^-, \hat{\boldsymbol{n}})\, ds = 0$$

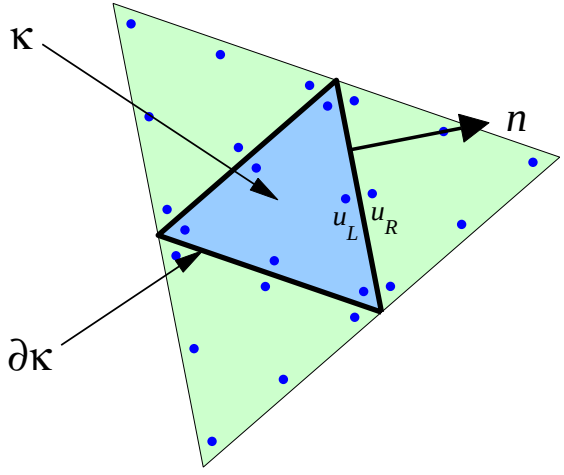
- Boundary conditions enforced naturally for any degree p
- Block-diagonal mass matrix (no overlap between basis functions)
- Block-wise compact stencil – neighboring elements connected

Mass Matrix



Jacobian





Outline

1. Introduction
2. Discretization and Solvers
   - Mesh Generation
   - The Discontinuous Galerkin Method
   - The CDG Method, Preconditioning, and Stabilization
3. Large Deformation Solid Dynamics
   - High-Order Lagrangian DG Formulations
   - Volumetric Modeling of Beams, Shells, and Membranes
   - J2 Viscoplasticity
4. Deformable Domains
   - Mapping-based ALE Formulation
   - Flapping Flight Applications
5. Conclusions

Viscous Discretization

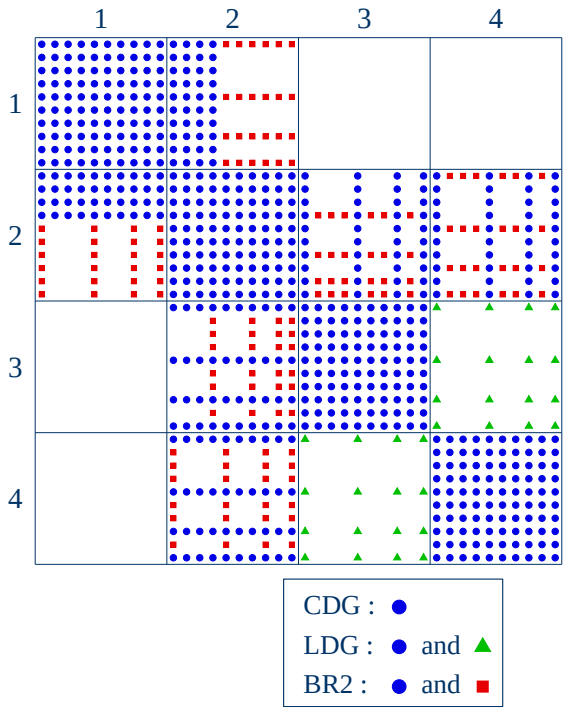
- General approach for second derivatives:
  - Write as system of first order equations:

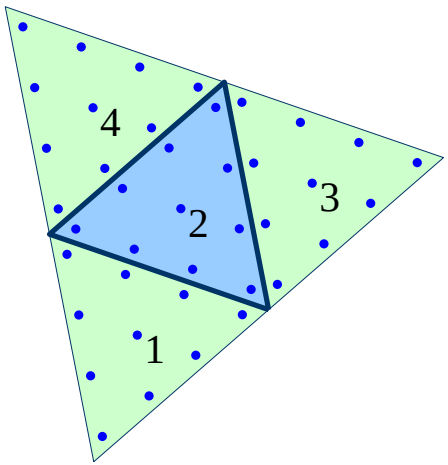
$$\boldsymbol{u}_t + \nabla \cdot \mathcal{F}_i(\boldsymbol{u}) - \nabla \cdot \mathcal{F}_v(\boldsymbol{u}, \boldsymbol{\sigma}) = 0$$

$$\boldsymbol{\sigma} - \nabla \boldsymbol{u} = 0$$

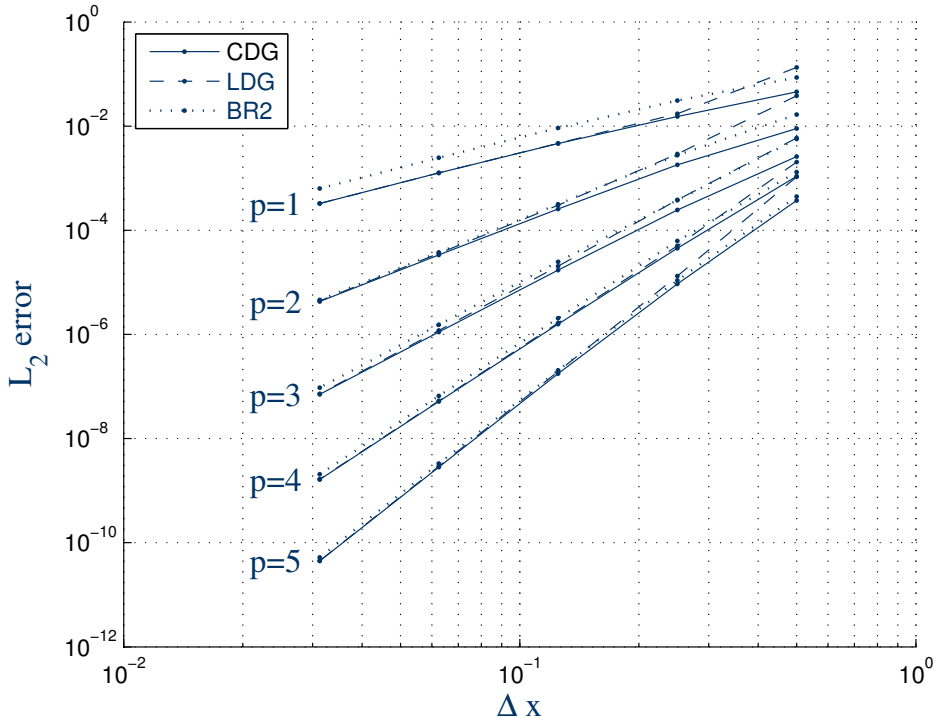
  - Discretize using DG, choose appropriate numerical fluxes $\hat{\sigma}$, $\hat{u}$
- Various schemes have been proposed:
  - *BR1* [Bassi/Rebay 97]: Averaging, unstable and non-compact
  - *BR2* [Bassi/Rebay 98]: Different lifting operator for each edge, compact connectivities, similar to Interior Penalty (IP)
  - *LDG* [Cockburn/Shu 98]: Upwind/Downwind, non-compact
  - *CDG* [Peraire/Persson 07]:
    Modification of LDG for local dependence – sparse and compact

The Compact Discontinuous Galerkin Method



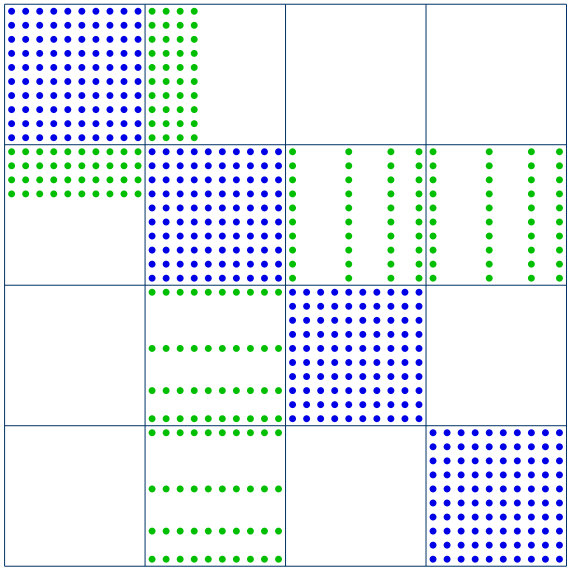


- Modification of numerical fluxes in LDG scheme
- Excludes non-compact and indefinite terms
- Provably optimal accuracy $\mathcal{O}(h^{p+1})$
- Higher stability/accuracy than LDG/BR2
- Sparser than LDG/BR2/IP

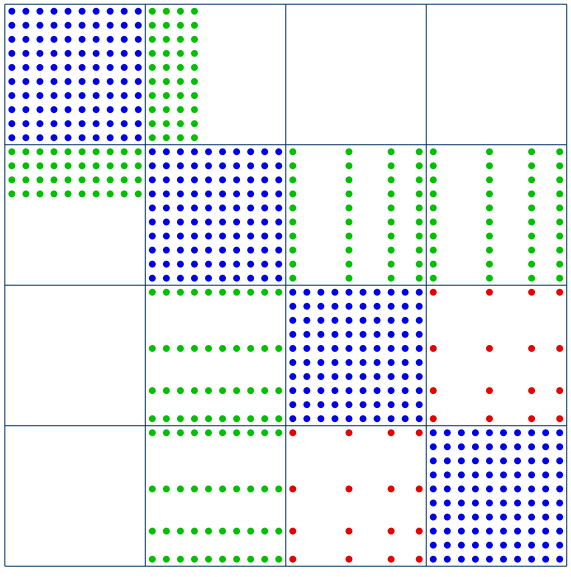


Matrix Representation

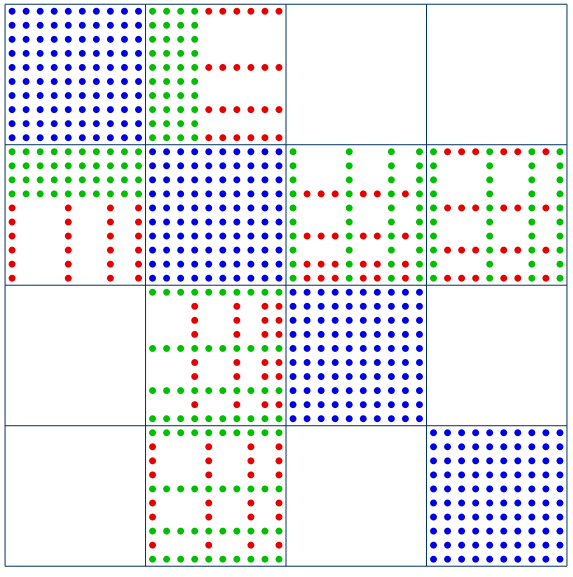
- Block matrix representation *fundamental for high performance*
  - Solver algorithms based on blocks
  - Up to 10 times higher performance with optimized BLAS
- Compact stencil $\Longrightarrow$ Matrix structure given by mesh connectivities
- Hard to store LDG/BR2/IP efficiently



CDG – 2 arrays



LDG – 3 arrays + struct



BR2 / IP – 3 arrays

Preconditioners for Krylov Methods

- Preconditioning required for GMRES convergence
- Standard point-wise Jacobi, ILU, etc, ineffective for DG
- Some success with standard block-wise methods:
  - Block diagonal: $\boldsymbol{A} \approx \mathrm{blockdiag}(\boldsymbol{A})$
    - Poor in general
  - Block Incomplete LU: $\boldsymbol{A} \approx \tilde{\boldsymbol{L}}\tilde{\boldsymbol{U}}$
    - Good for convection (with the right ordering, more later)
  - p-multigrid (low-degree) preconditioner
    - Good for diffusion
- Highly efficient combination [Persson/Peraire 06]
  - Use block ILU(0) as post-smoother for coarse scale correction
  - Combines advantages of ILU and low-degree preconditioners
  - Cheap general purpose preconditioner

Minimum Discarded Fill Element Ordering

- Performance of ILU(0) highly dependent of ordering
- Greedy algorithm for element ordering [Persson/Peraire 07]:
  At step j, if j' is chosen next, we would discard the fill

$$\Delta\tilde{\boldsymbol{U}}_{ik}^{(j,j')} = -\tilde{\boldsymbol{U}}_{ij'}\tilde{\boldsymbol{U}}_{j'j'}^{-1}\tilde{\boldsymbol{U}}_{j'k}, \quad \text{for neighbors } i \geq j, k \geq j \text{ of element } j'$$

- Choose the j' that minimizes the norm of the discarded fill

$$w^{(j,j')} = \|\Delta\tilde{\boldsymbol{U}}^{(j,j')}\|_{\mathrm{F}}$$

- Some simplifications, min-heap data structure $\Longrightarrow \mathcal{O}(n \log n)$ cost
- Similar to the *Minimum Degree* algorithm, but considering the magnitude of the fill instead of just the size
- Related to the *Minimum Discarded Fill* algorithm [D'Azevedo, Forsyth, Tang 92]

The ILU(0)-p1 Preconditioner

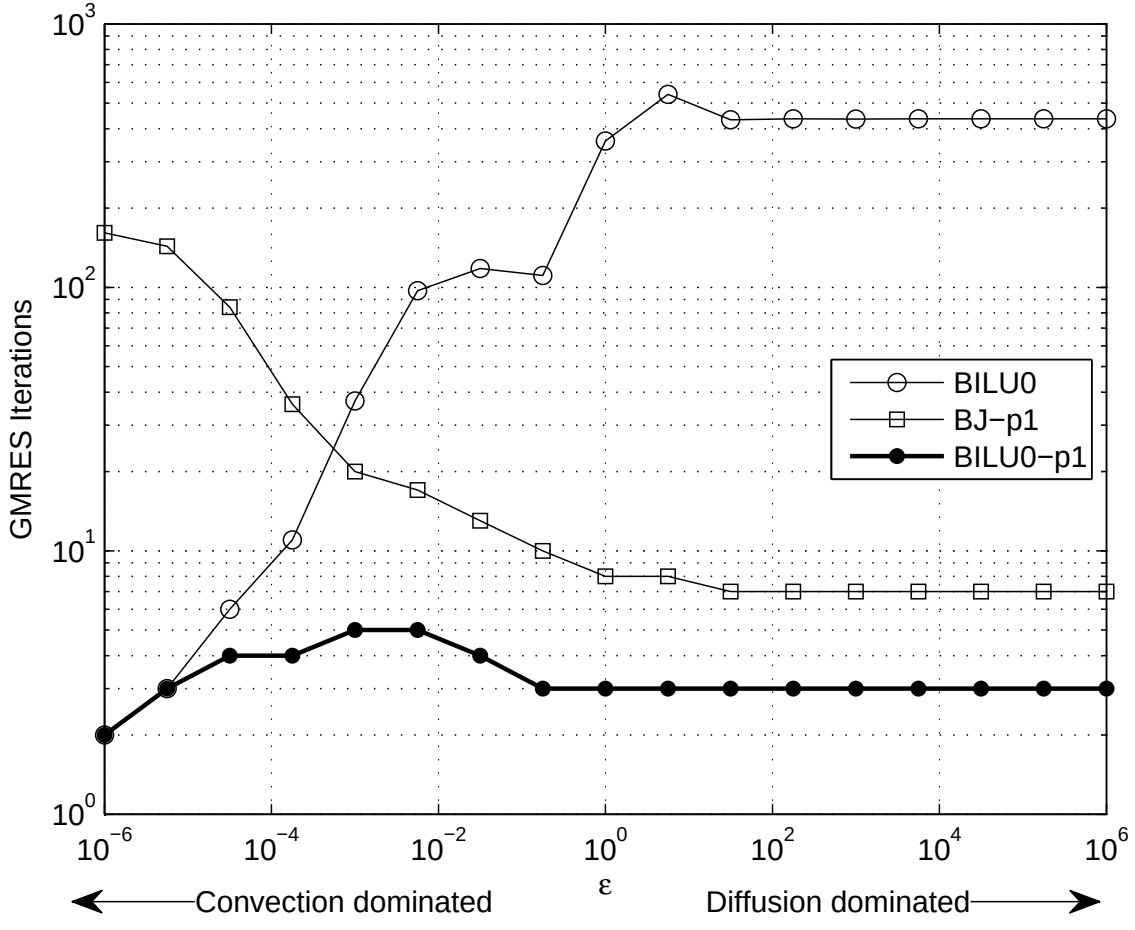
- Combination of block ILU and multigrid [Persson/Peraire 06]
- Coarse scale correction + post-smoothing by $\tilde{\boldsymbol{A}} = \tilde{\boldsymbol{L}}\tilde{\boldsymbol{U}}$:

0. $\boldsymbol{A}^{(0)} = \boldsymbol{P}^T\boldsymbol{A}\boldsymbol{P}$	*Precompute coarse operator, block wise*
1. $\boldsymbol{b}^{(0)} = \boldsymbol{P}^T\boldsymbol{b}$	*Restrict residual element/component wise*
2. $\boldsymbol{A}^{(0)}\boldsymbol{u}^{(0)} = \boldsymbol{b}^{(0)}$	*Solve coarse scale problem*
3. $\boldsymbol{u} = \boldsymbol{P}\boldsymbol{u}^{(0)}$	*Prolongate solution element/component wise*
4. $\boldsymbol{u} = \boldsymbol{u} + \alpha\tilde{\boldsymbol{A}}^{-1}(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{u})$	*Apply smoother $\tilde{\boldsymbol{A}}$ with damping α*

- Restriction/prolongation operator $\boldsymbol{P}$ block diagonal, based on orthogonal Koornwinder polynomials
- Coarse scale problem solved directly (2-D problems, serial) or iteratively by GMRES or h-multigrid

The ILU-p1 Preconditioner - Convection-Diffusion

- Block ILU perfect for convection, multigrid perfect for diffusion
- Block ILU-smoothed multigrid (BILU0-p1) almost perfect for any ε



The ILU-p1 Preconditioner – Navier-Stokes

- Convergence for various Reynolds numbers, Δt, and M
- $\times$= No convergence after 1,000 iterations

Problem	Parameters		Preconditioner/Iterations		
	Δt	M	Block Jacobi-p1	Block GS-p1	Block ILU0-p1
Inviscid	10^{-3}	0.2	19	11	4
	10^{-1}	0.2	85	49	6
	∞	0.2	142	72	9
	10^{-3}	0.01	112	67	6
	10^{-1}	0.01	$\times$	532	10
	∞	0.01	$\times$	$\times$	16
Laminar	10^{-3}	0.2	34	19	4
Re=1,000	10^{-1}	0.2	597	225	5
	∞	0.2	$\times$	$\times$	7
	10^{-3}	0.01	66	33	5
	10^{-1}	0.01	619	207	9
	∞	0.01	$\times$	748	12

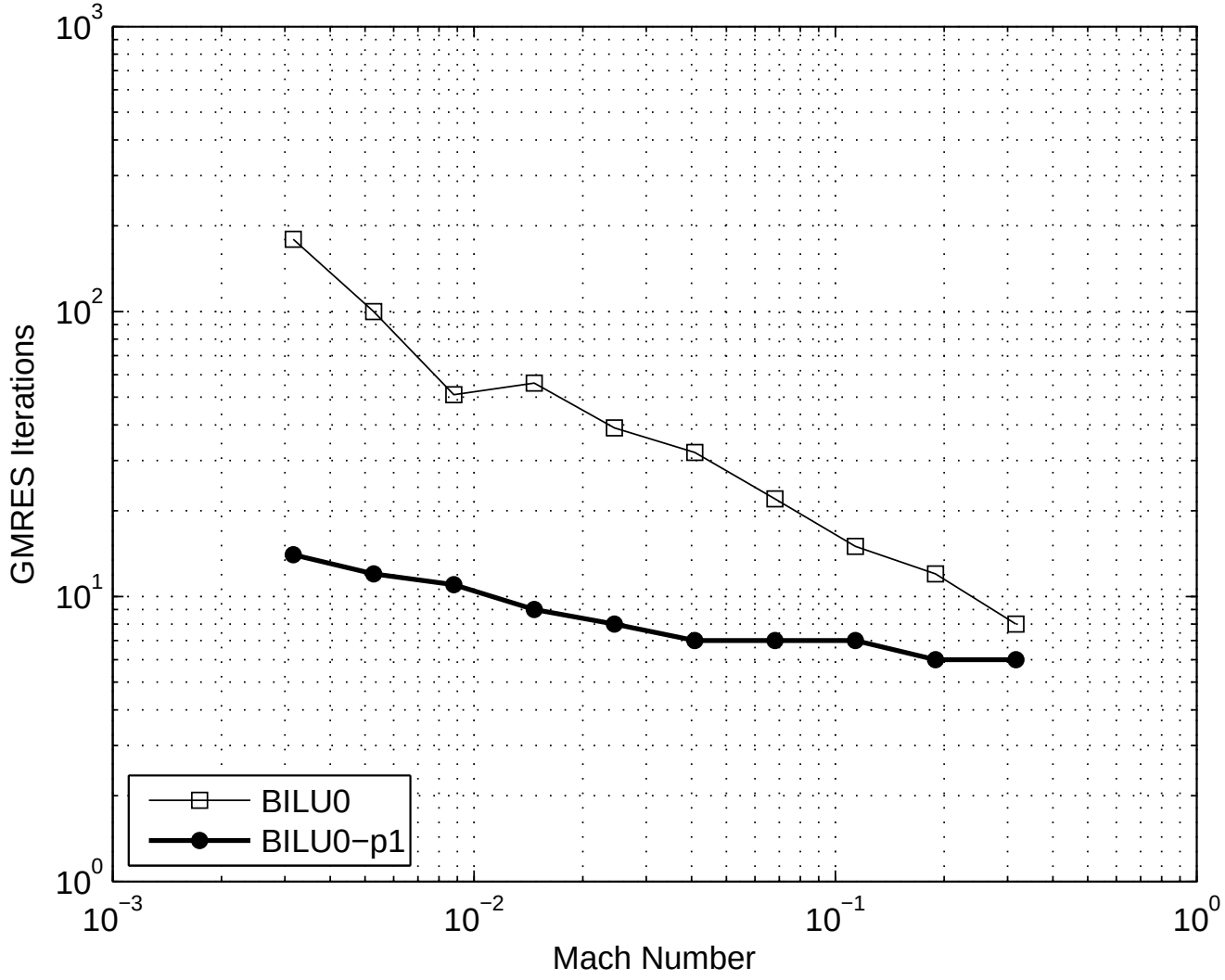
The ILU-p1 Preconditioner – Navier-Stokes

- Convergence for various Reynolds numbers, Δt, and M
- $\times$ = No convergence after 1,000 iterations

Problem	Parameters		Preconditioner/Iterations		
	Δt	M	Block Jacobi-p1	Block GS-p1	Block ILU0-p1
Laminar	10^{-3}	0.2	19	11	4
Re=20,000	10^{-1}	0.2	220	113	8
	∞	0.2	$\times$	$\times$	20
	10^{-3}	0.01	90	38	6
	10^{-1}	0.01	$\times$	735	16
	∞	0.01	$\times$	$\times$	35
RANS	10^{-3}	0.2	56	28	7
Re=10^6	10^{-1}	0.2	$\times$	$\times$	25
	∞	0.2	$\times$	$\times$	18
	10^{-3}	0.01	231	110	9
	10^{-1}	0.01	$\times$	$\times$	16
	∞	0.01	$\times$	$\times$	28

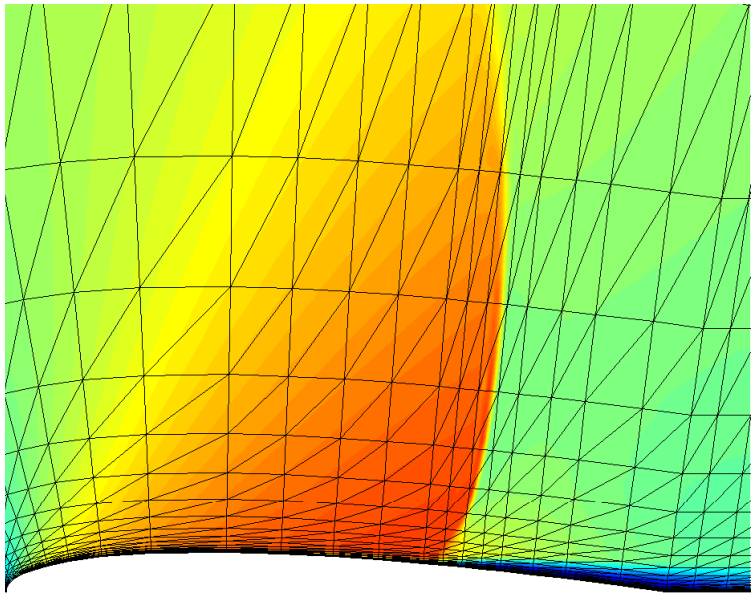
Mach Number Dependence

- Excellent convergence for Mach number $M << 1$ with BILU0-p1

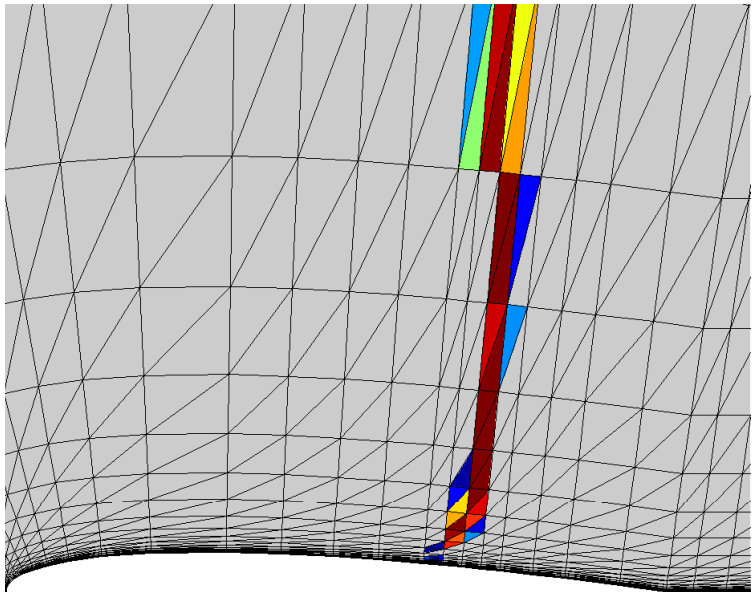


Artificial Viscosity for Underresolved Features

- Cannot resolve all solution features (shocks, singularities)
- Low dissipation makes DG sensitive to underresolution
- Detect by sensors and add viscosity [Persson/Peraire 06,07]
- Enables shock capturing with sub-cell resolution



Mach



Sensor

Shock Sensor

- Regularity of solution determined from the decay rate of expansion coefficients in orthogonal basis
- Example: Periodic Fourier case: $f(x) = \sum_{k=-\infty}^{\infty} g_k e^{ikx}$
  If $f(x)$ has m continuous derivatives $\rightarrow$ $|g_k| \sim k^{-(m+1)}$
- For simplices: Expand solution in orthonormal Koornwinder basis:

$$u = \sum_{i=1}^{N(p)} u_i \psi_i, \quad \hat{u} = \sum_{i=1}^{N(p-1)} u_i \psi_i, \quad s_e = \log_{10}\left(\frac{(u-\hat{u}, u-\hat{u})_e}{(u,u)_e}\right)$$

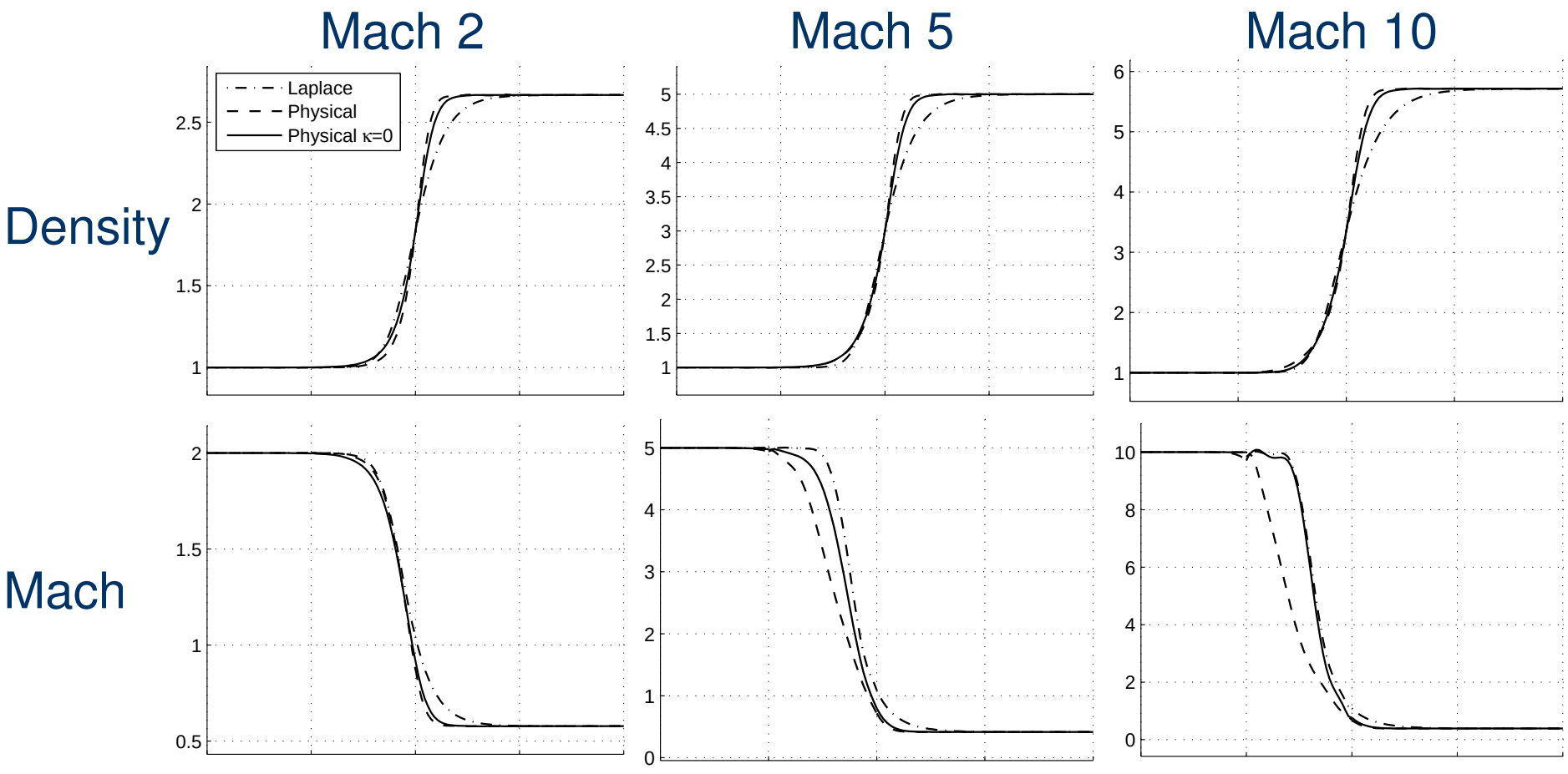
- Determine elemental piecewise constant ε_e

$$\varepsilon_e = \begin{cases} 0 & \text{if} \quad s_e < s_0 - \kappa \\ \frac{\varepsilon_0}{2}\left(1 + \sin\frac{\pi(s_e - s_0)}{2\kappa}\right) & \text{if} \quad s_0 - \kappa \le s_e \le s_0 + \kappa \\ \varepsilon_0 & \text{if} \quad s_e > s_0 + \kappa \end{cases} .$$

where $\varepsilon_0 \sim h/p$, $s_0 \sim 1/p^4$ and κ empirical

Euler Equations – Artificial Viscosity Models

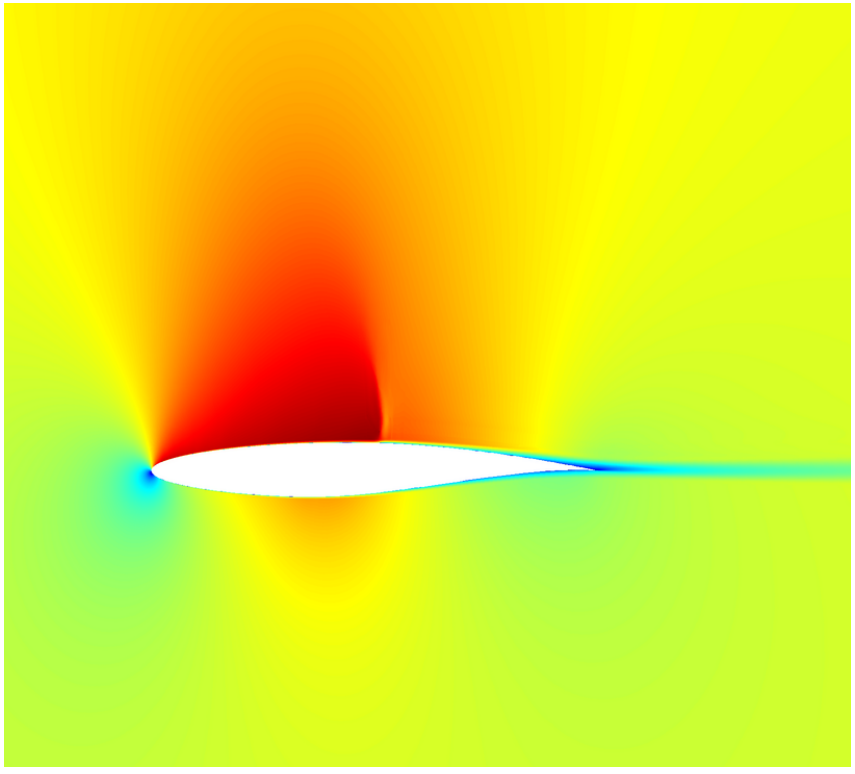
- Laplacian: $\frac{\partial \boldsymbol{u}}{\partial t} + \nabla \cdot \mathcal{F}(\boldsymbol{u}) = \nabla \cdot (\epsilon \nabla \boldsymbol{u})$
- Physical: $\frac{\partial \boldsymbol{u}}{\partial t} + \nabla \cdot \mathcal{F}(\boldsymbol{u}) = \nabla \cdot \mathcal{F}_v(\boldsymbol{u}; \mathrm{Re}, \mathrm{Pr})$

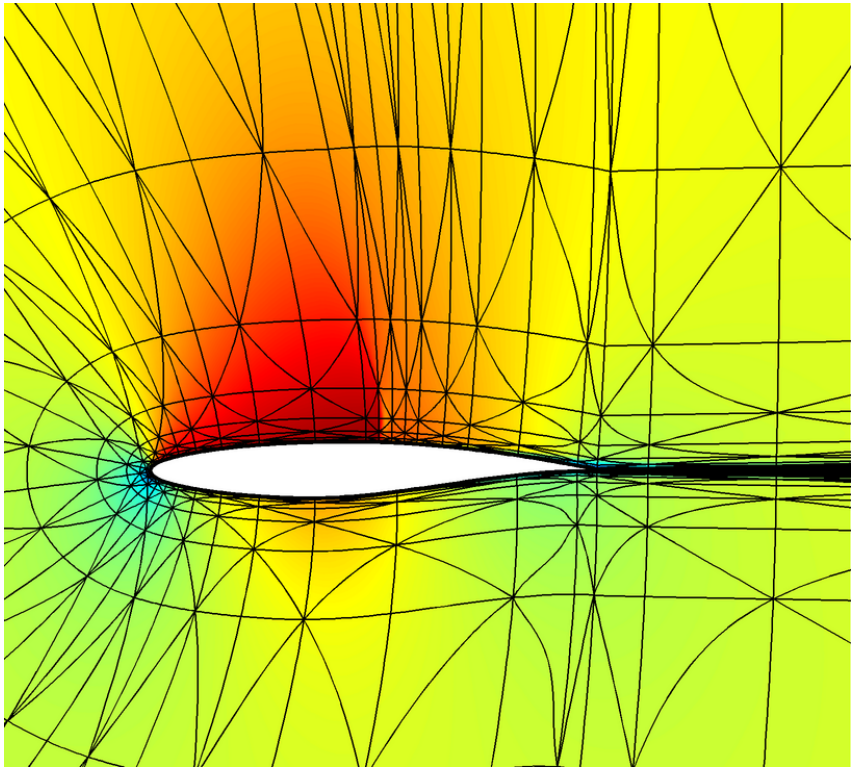


Example: RAE2822, Viscous and Transonic

- Transonic flow ($M = 0.729, \mathrm{Re} = 6.5 \cdot 10^6$)
- Sub-cell resolution of shocks
- p-converged solutions

$p = 4$

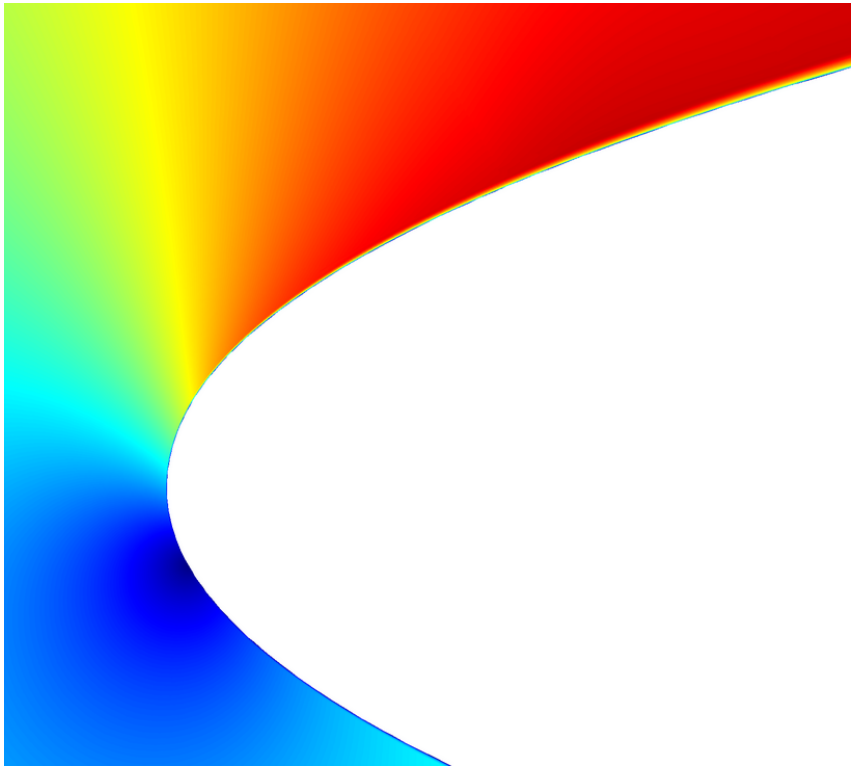


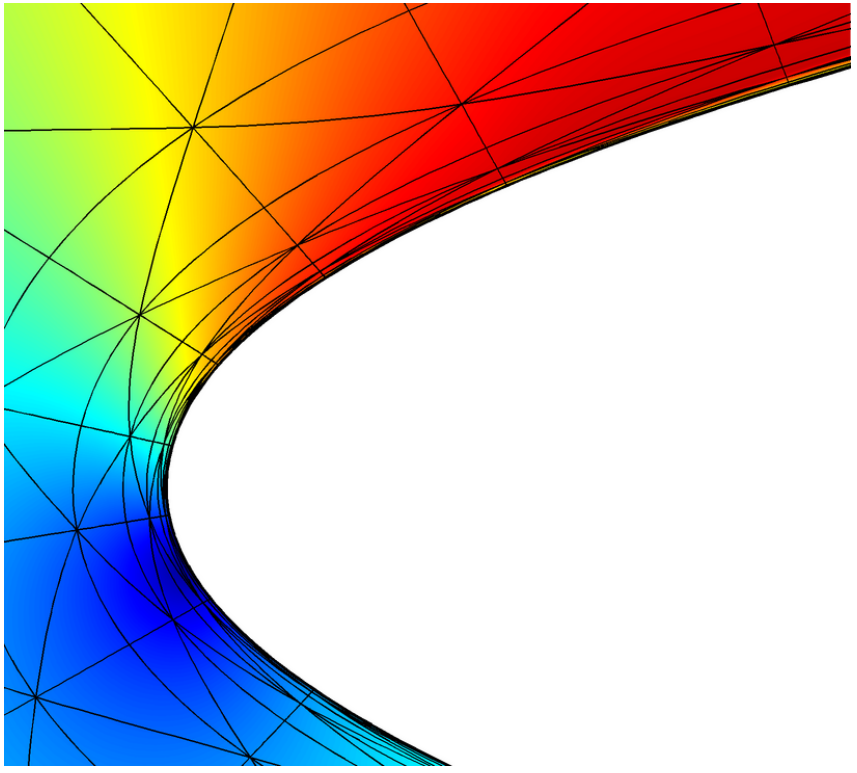


Example: RAE2822, Viscous and Transonic

- Transonic flow ($M = 0.729, \mathrm{Re} = 6.5 \cdot 10^6$)
- Sub-cell resolution of shocks
- p-converged solutions

$p = 4$





Outline

1. Introduction
2. Discretization and Solvers
   - Mesh Generation
   - The Discontinuous Galerkin Method
   - The CDG Method, Preconditioning, and Stabilization
3. Large Deformation Solid Dynamics
   - High-Order Lagrangian DG Formulations
   - Volumetric Modeling of Beams, Shells, and Membranes
   - J2 Viscoplasticity
4. Deformable Domains
   - Mapping-based ALE Formulation
   - Flapping Flight Applications
5. Conclusions

Lagrangian DG Formulation for Solid Dynamics

- Many potential benefits with DG for solid dynamics
- Arbitrary orders of accuracy with tetrahedral elements
- Discontinuous approach highly insensitive to locking

  $\Longrightarrow$ Almost incompressible materials

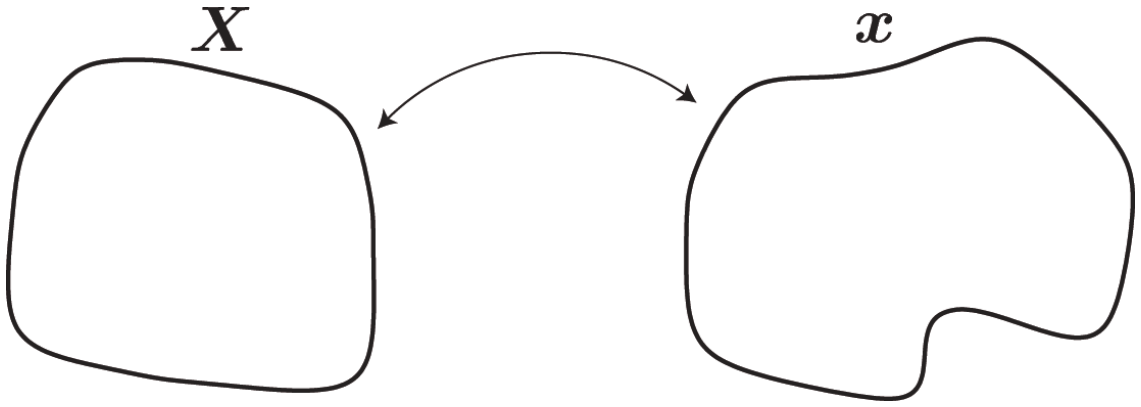
  $\Longrightarrow$ Thin structures (beams, shells) with highly stretched elements
- Stabilization for convection

  $\Longrightarrow$ High-order accurate ALE
- Block-diagonal mass matrix

  $\Longrightarrow$ Efficient explicit methods

Lagrangian Formulation

- Map from reference domain V to physical domain $v(t)$



$$\boldsymbol{F} = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{X}} \,, \qquad J = \det \boldsymbol{F} \,, \qquad \boldsymbol{v}(\boldsymbol{X}, t) = \frac{\partial \boldsymbol{x}}{\partial t} \,, \qquad \boldsymbol{p} = \rho_0 \boldsymbol{v}$$

- Conservation of linear momentum:

$$\frac{d}{dt} \int_V \boldsymbol{p} \, dV = \int_V \rho_0 \boldsymbol{b} \, dV + \int_{\partial V} \boldsymbol{t} \, dA \,, \qquad \boldsymbol{t} = \boldsymbol{P}\boldsymbol{N}$$

$$\frac{\partial \boldsymbol{p}}{\partial t} = \boldsymbol{\nabla} \cdot \boldsymbol{P} + \rho_0 \boldsymbol{b}$$

with *first Piola-Kirchhoff stress tensor* $\boldsymbol{P}(\boldsymbol{F})$

Hyperelastic Neo-Hookean Constitutive Model

- Strain Energy Potential

$$\psi(\boldsymbol{F}) = \psi_{iso}(J^{-1/3}\boldsymbol{F}) + \psi_{vol}(J)$$

$$\psi_{iso} = \frac{\mu}{2}[tr(J^{-2/3}\boldsymbol{F} : \boldsymbol{F}) - 3] \ , \qquad \psi_{vol} = \frac{1}{2}\kappa(J-1)^2$$

- First Piola-Kirchhoff Stress Tensor

$$\boldsymbol{P} = \boldsymbol{P}_{dev} + \boldsymbol{P}_{vol} \ ; \ \boldsymbol{P}_{dev} = \frac{\partial \psi_{iso}}{\partial \boldsymbol{F}} \ , \ \boldsymbol{P}_{vol} = \frac{\partial \psi_{vol}}{\partial \boldsymbol{F}}$$

$$\boldsymbol{P}_{vol} = \frac{d\psi_{vol}(J)}{dJ}\frac{\partial J}{\partial \boldsymbol{F}} = pJ\boldsymbol{F}^{-T} \ ; \ p = \frac{d\psi_{vol}(J)}{dJ} = \kappa(J-1)$$

$$\boldsymbol{P}_{dev} = \mu J^{-2/3}[\boldsymbol{F} - \frac{1}{3}(\boldsymbol{F} : \boldsymbol{F})\boldsymbol{F}^{-T}]$$

Second-Order Formulation

- Straight-forward second-order formulation in terms of material points $\boldsymbol{x}$ and momentum $\boldsymbol{p}$:

$$\frac{\partial \boldsymbol{x}}{\partial t} = \boldsymbol{p}/\rho_0$$
$$\frac{\partial \boldsymbol{p}}{\partial t} - \nabla \cdot \boldsymbol{P}(\boldsymbol{F}) = \rho_0 \boldsymbol{b}$$

- Computationally inexpensive, first equation essentially an ODE
- Full pth order polynomial spaces for $\boldsymbol{x}$ and $\boldsymbol{p}$
- Use CDG scheme for stable discretization of second derivatives
- However, conservative treatment of shocks unclear

First-Order Conservation Law Formulation

- Solve for momentum $\boldsymbol{p}$ and deformation gradient $\boldsymbol{F}$
- Time evolution of the deformation gradient:

$$\frac{\partial \boldsymbol{F}}{\partial t} = \boldsymbol{\nabla}(\boldsymbol{p}/\rho_0) \qquad \text{or} \qquad \frac{\partial \boldsymbol{F}}{\partial t} = \boldsymbol{\nabla} \cdot \left(\frac{1}{\rho_0} \boldsymbol{p} \otimes \boldsymbol{I} \right)$$

- Write as system of conservation laws:

$$\mathcal{U}_t + \nabla \cdot \mathcal{F}(\mathcal{U}) = \mathcal{S}(\mathcal{U})$$

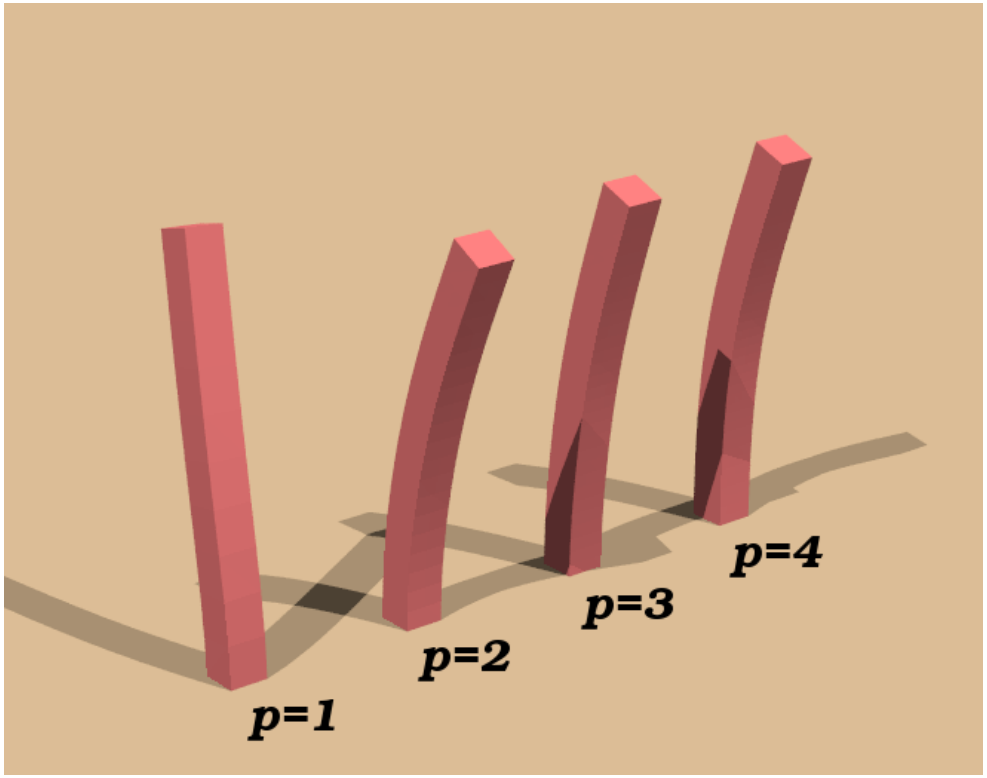
with

$$\mathcal{U} = \begin{pmatrix} \boldsymbol{p} \\ \boldsymbol{F} \end{pmatrix}, \quad \mathcal{F} = \begin{pmatrix} \boldsymbol{P} \\ \frac{1}{\rho_0} \boldsymbol{p} \otimes \boldsymbol{I} \end{pmatrix}, \quad \mathcal{S} = \begin{pmatrix} \rho_0 \boldsymbol{b} \\ \boldsymbol{0} \end{pmatrix}$$

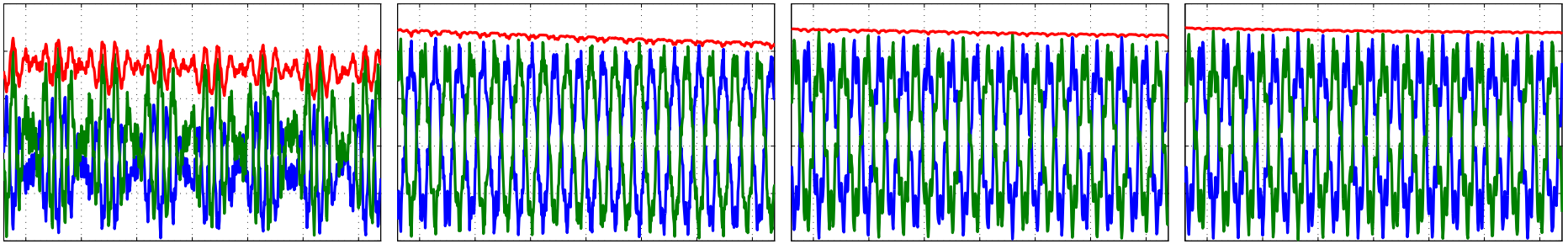
- Conservative formulation $\Longrightarrow$ borrow shock capturing from CFD, jump conditions physically meaningful
- However, computationally expensive, needs curl-free spaces for $\boldsymbol{F}$

Oscillating Beam, Energy Conservation

- Energy conservation not explicitly enforced by scheme
- Therefore, it is a good indicator of accuracy



Kinetic, potential, and total energy



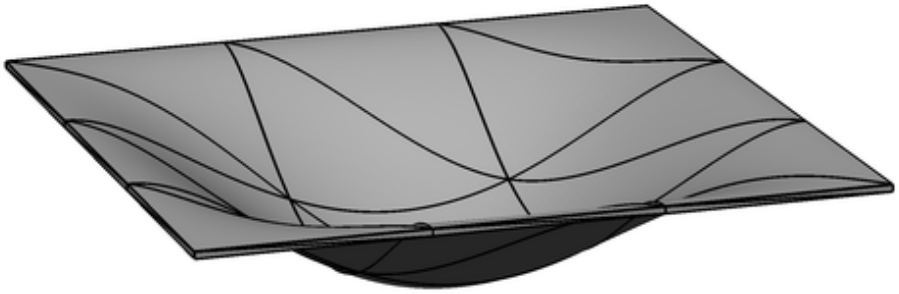
$p = 1$ $p = 2$ $p = 3$ $p = 4$

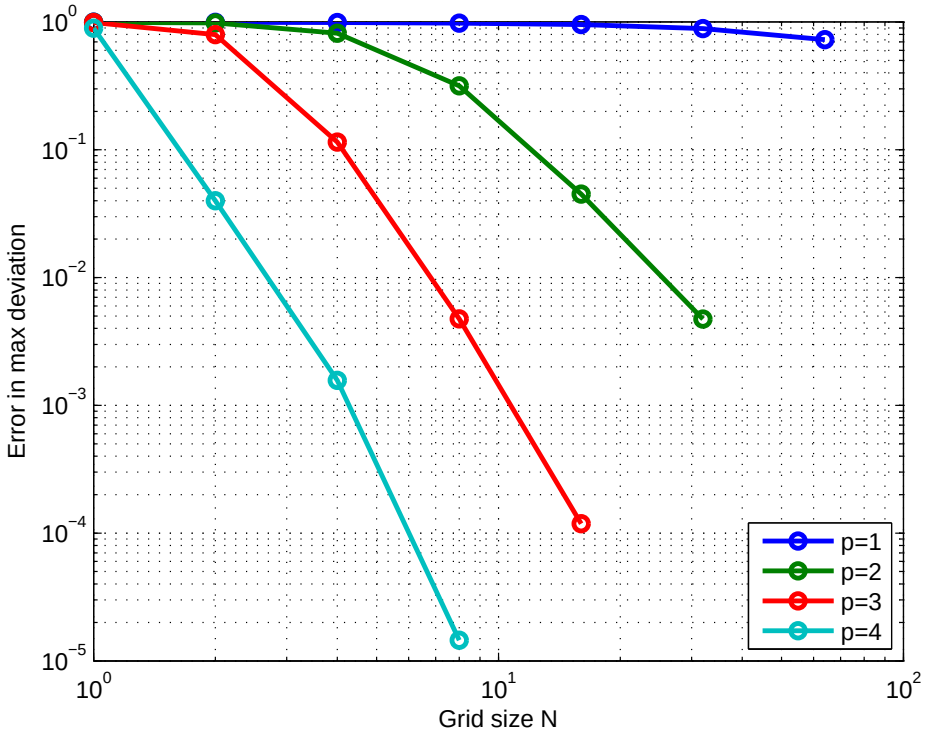
Outline

1. Introduction
2. Discretization and Solvers
   - Mesh Generation
   - The Discontinuous Galerkin Method
   - The CDG Method, Preconditioning, and Stabilization
3. Large Deformation Solid Dynamics
   - High-Order Lagrangian DG Formulations
   - Volumetric Modeling of Beams, Shells, and Membranes
   - J2 Viscoplasticity
4. Deformable Domains
   - Mapping-based ALE Formulation
   - Flapping Flight Applications
5. Conclusions

Volumetric Modeling of Thin Structures

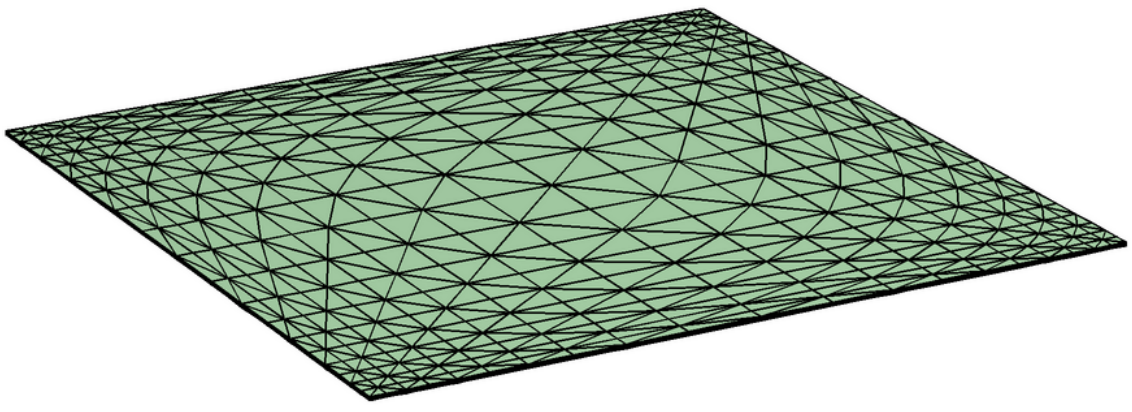
- The high order elements allow for high stretching
- Discontinuous basis functions highly insensitive to locking
- Model plate problem: Uniform loading, small deformations
- High order convergence rates for $p > 1$

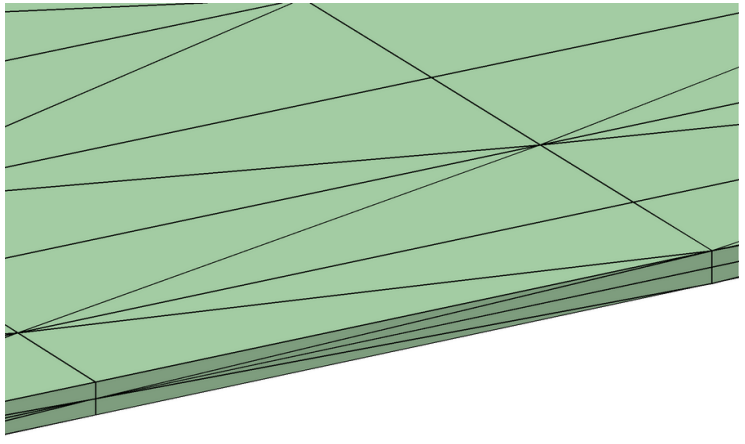




Example: Airbag Inflation

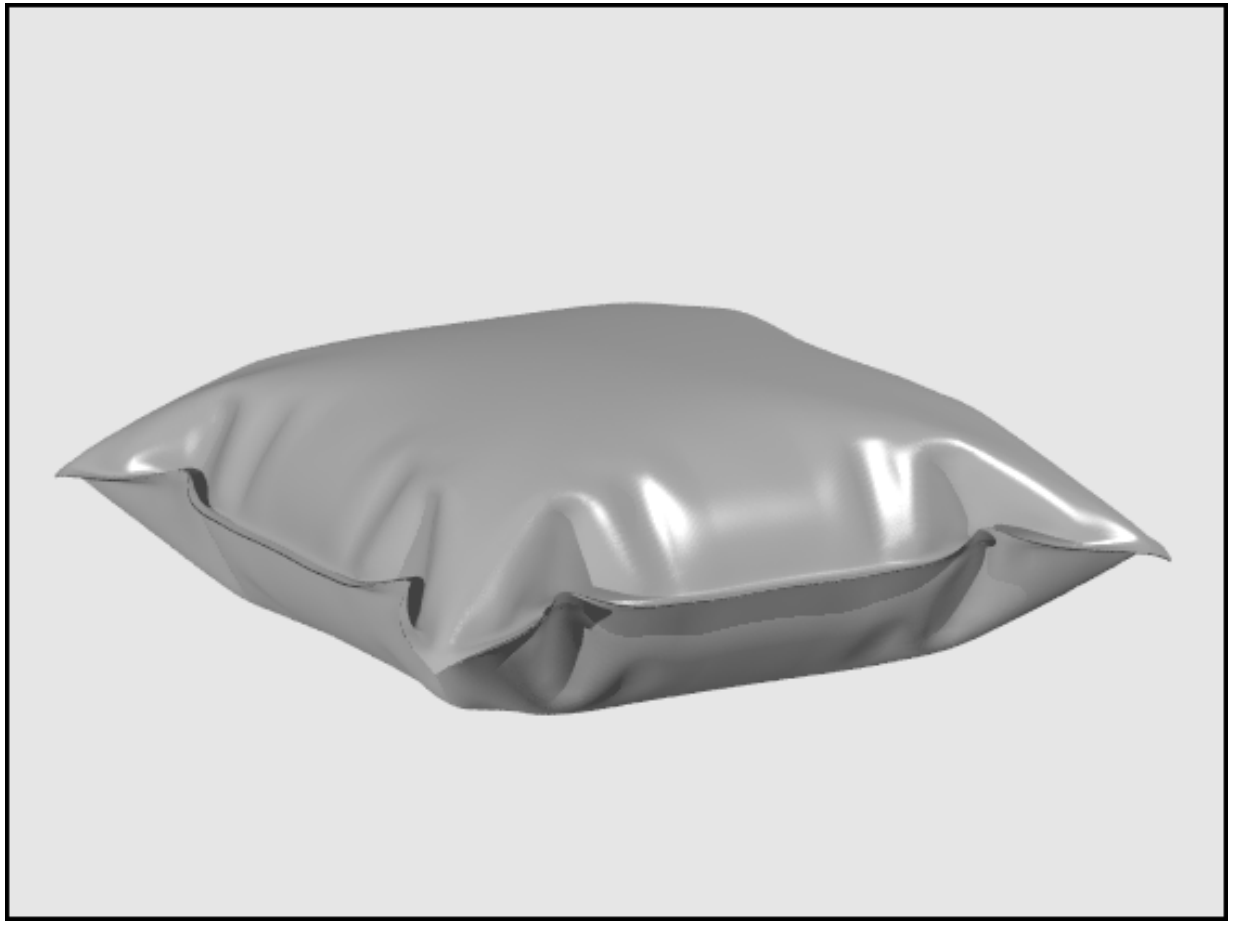
- Undeformed configuration: Two thin sheets, attached at the edges (thickness=10^{-3}, diagonal=1.2)
- Discretize with stretched tetrahedra and high-order DG elements
- Interior pressure p_0 applied at $t = 0$, quasi-static evolution





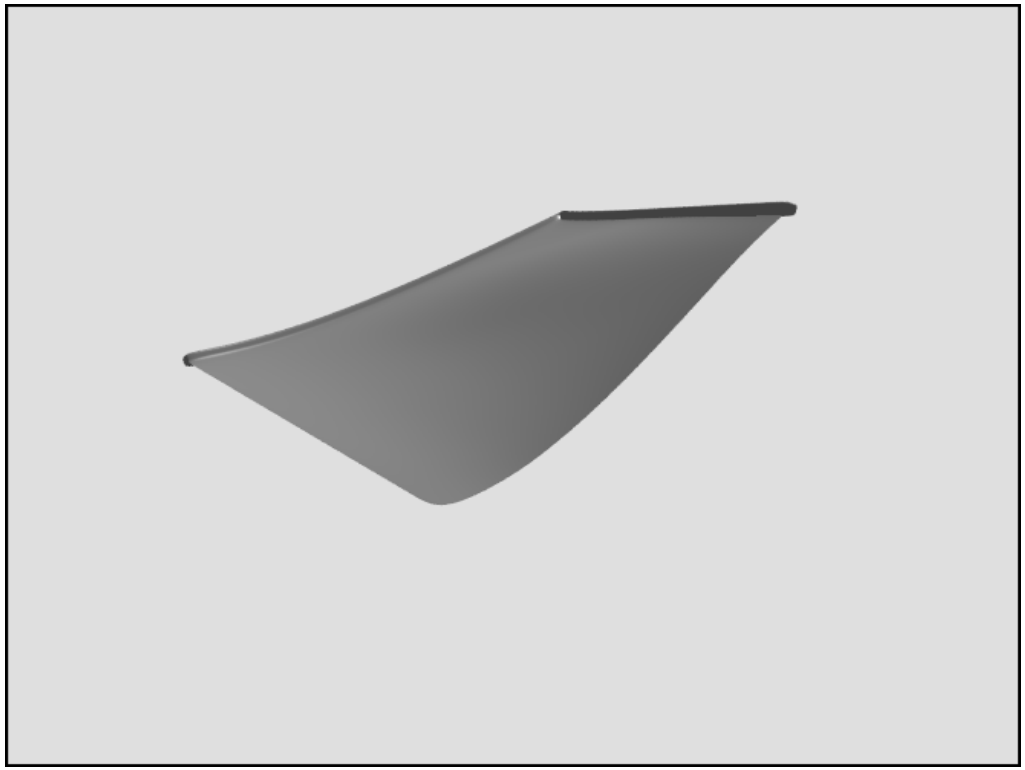
Example: Airbag Inflation

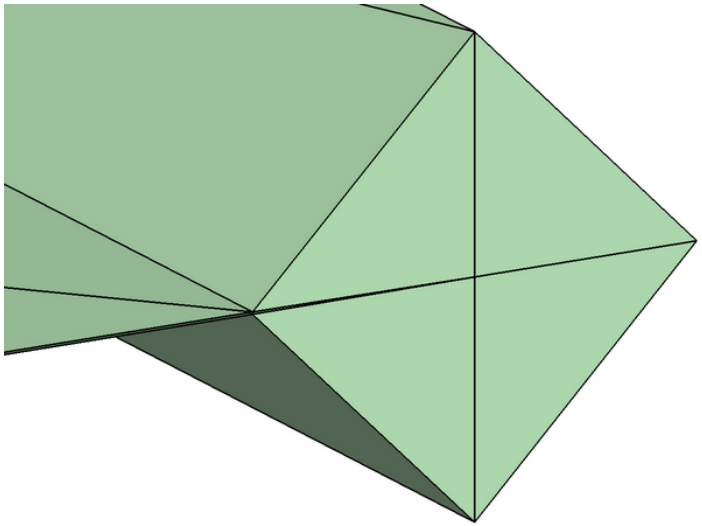
- 40-by-40 grid, 16000 tetrahedra, polynomial order $p = 3$
- No specialized membrane models required
- Hard to define convergence, but $\max(\Delta z) \rightarrow$ fixed value

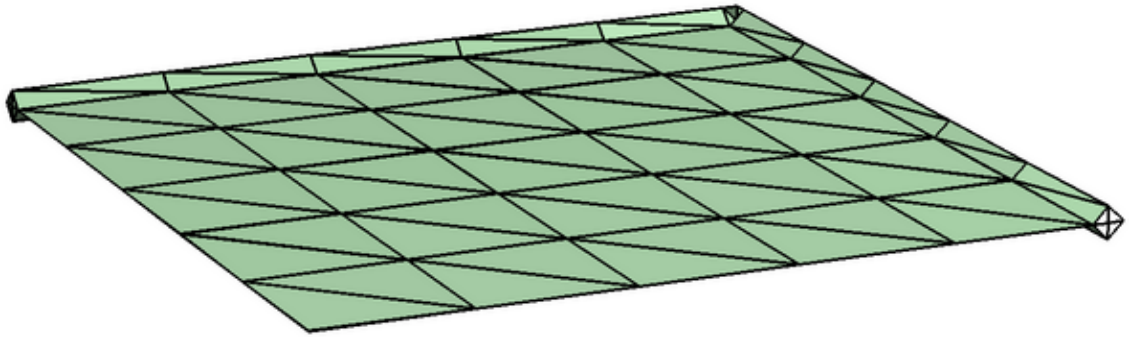


Volumetric Modeling of Coupled Beam/Membrane

- No modeling required to couple beams/membranes
- Only a question of generating the stretched tetrahedral meshes





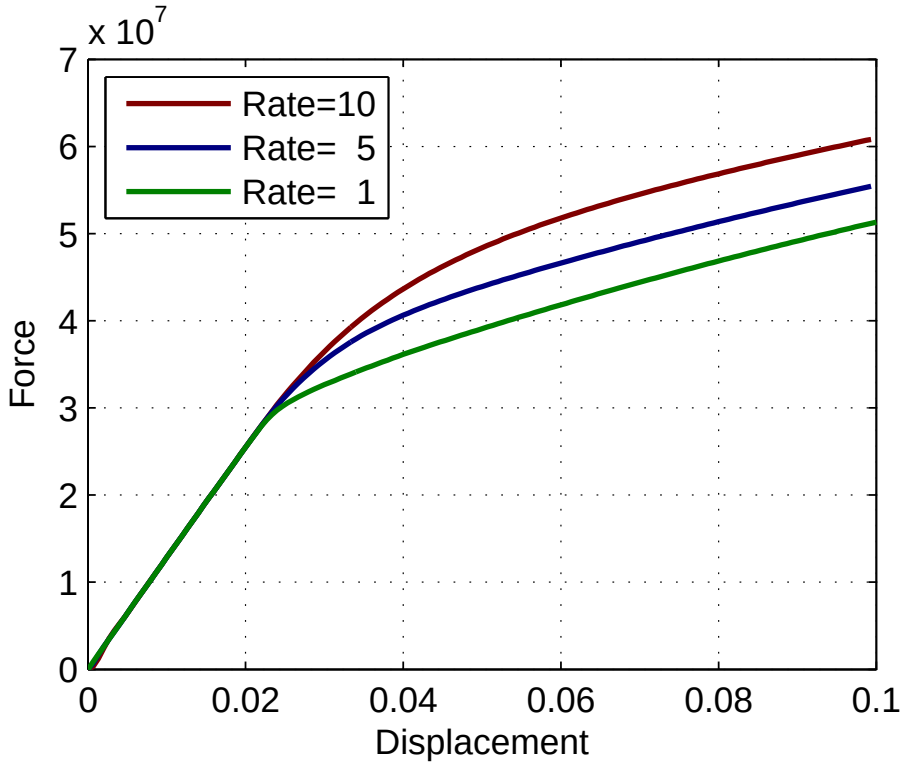


Outline

1. Introduction
2. Discretization and Solvers
   - Mesh Generation
   - The Discontinuous Galerkin Method
   - The CDG Method, Preconditioning, and Stabilization
3. Large Deformation Solid Dynamics
   - High-Order Lagrangian DG Formulations
   - Volumetric Modeling of Beams, Shells, and Membranes
   - J2 Viscoplasticity
4. Deformable Domains
   - Mapping-based ALE Formulation
   - Flapping Flight Applications
5. Conclusions

J2 Viscoplasticity

- Formulate J2 Viscoplasticity model as system of equations
- *Systematic discretizations* which converge under h/p refinement
- *Arbitrary orders of accuracy* in space and time



Rate = 1

Rate = 10

J2 Viscoplasticity as System of Equations

- Introduce the *scalar plastic deformation* ϵ_p and the *right plastic Cauchy-Green tensor* $\boldsymbol{C}_p$ as unknowns

$$\mathcal{U} = \begin{pmatrix} \boldsymbol{x} & \boldsymbol{p} & \epsilon_p & \boldsymbol{C}_p \end{pmatrix}$$

- Evolution equations for $\boldsymbol{x}, \boldsymbol{p}$ similar to before:

$$\dot{\boldsymbol{x}} = \boldsymbol{p}/\rho_0$$
$$\dot{\boldsymbol{p}} - \nabla \cdot \boldsymbol{P} = \rho_0 \boldsymbol{b}$$

where

$$\boldsymbol{P} = \mu(\boldsymbol{b}_e \boldsymbol{F}^{-T} - \boldsymbol{F}^{-T}) + \lambda(\ln J_e)\boldsymbol{F}^{-T},$$
$$J_e = \sqrt{|\boldsymbol{b}_e|}, \quad \boldsymbol{b}_e = \boldsymbol{F}\boldsymbol{C}_p^{-1}\boldsymbol{F}^T$$

J2 Viscoplasticity as System of PDEs/ODEs

- Evolution equations for ϵ_p and $\boldsymbol{C}_p$:

$$\dot{\epsilon}_p = \frac{1}{\eta}\langle \bar{\tau} - \tau_y(\epsilon_p)\rangle$$

$$\dot{\boldsymbol{C}}_p = \frac{3}{\eta/\mu}\frac{\langle \bar{\tau} - \tau_y(\epsilon_p)\rangle}{\bar{\tau}}\left(\boldsymbol{F}^T\boldsymbol{F} - \frac{1}{3}(tr\boldsymbol{b}_e)\boldsymbol{C}_p\right)$$

where η is a viscosity parameter, $\langle\cdot\rangle$ is the ramp function, and

$$\boldsymbol{\tau} = \boldsymbol{P}\boldsymbol{F}^T = \mu(\boldsymbol{b}_e - \boldsymbol{I}) + \lambda(\ln J_e)\boldsymbol{I}$$

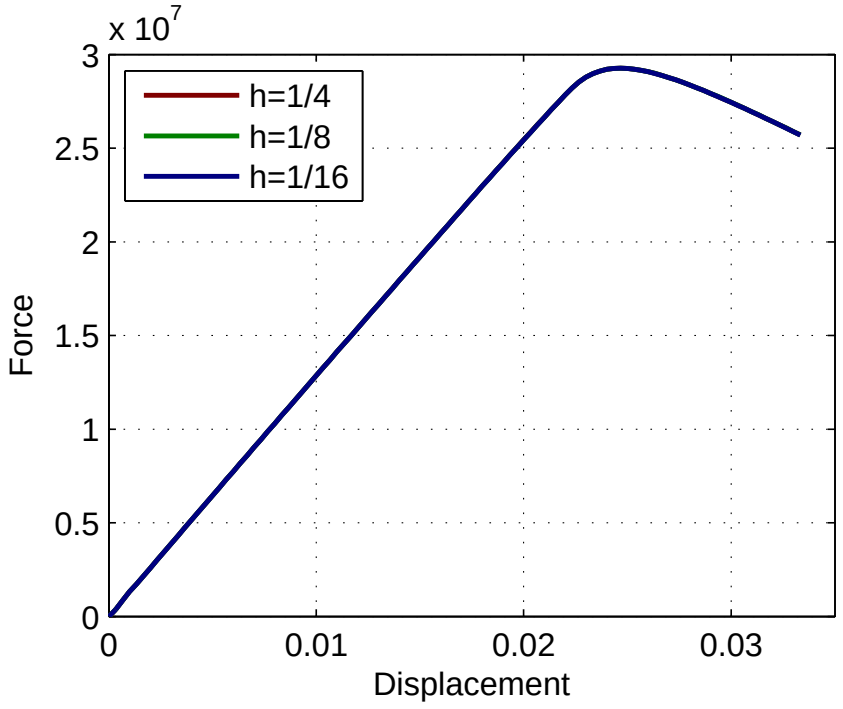
$$\boldsymbol{\tau}' = \boldsymbol{\tau} - \frac{1}{3}(tr\boldsymbol{\tau})\boldsymbol{I} = \mu(\boldsymbol{b}_e - \frac{1}{3}(tr\boldsymbol{b}_e)\boldsymbol{I})$$

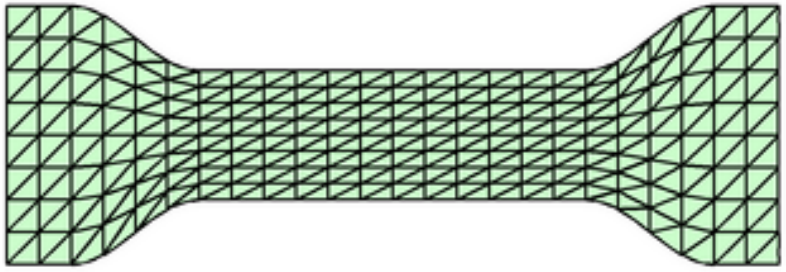
$$\bar{\tau} = \sqrt{\frac{3}{2}\boldsymbol{\tau}' : \boldsymbol{\tau}'}$$

$$\tau_y = \tau_y^0 + H\epsilon_p$$

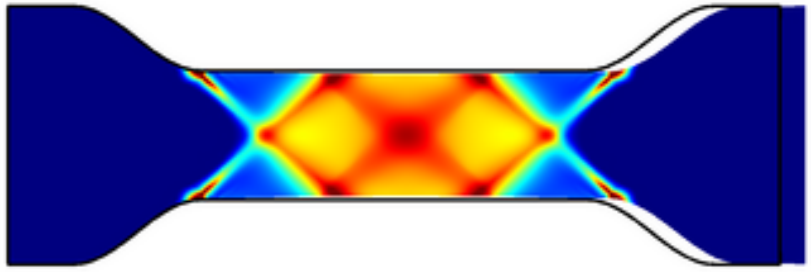
Example: Strain Localization

- With *strain softening* ($H < 0$), deformations localize in narrow bands of intense straining
- Sharp jumps in displacements, location unknown a priori
- Insensitive to mesh with $p > 2$





Mesh, $p = 4$



Plastic deformation ϵ_p

Outline

1. Introduction
2. Discretization and Solvers
   - Mesh Generation
   - The Discontinuous Galerkin Method
   - The CDG Method, Preconditioning, and Stabilization
3. Large Deformation Solid Dynamics
   - High-Order Lagrangian DG Formulations
   - Volumetric Modeling of Beams, Shells, and Membranes
   - J2 Viscoplasticity
4. Deformable Domains
   - Mapping-based ALE Formulation
   - Flapping Flight Applications
5. Conclusions

Methods for Deforming Domains

- Many ALE formulations for unstructured meshes
  - Equations discretized on a deforming grid, time-dependent metric
  - At most third order accuracy in space and time demonstrated
- Alternative approach for finite differences [Visbal/Gaitonde 02]
  - Map from fixed reference domain to time-varying physical domain
  - Correction for the Geometric Conservation Law (GCL) with sources
- Our method: Mapping approach in a DG setting, with a conservative formulation for satisfying the GCL
  - Guaranteed stability
  - Arbitrary orders of accuracy in space and time

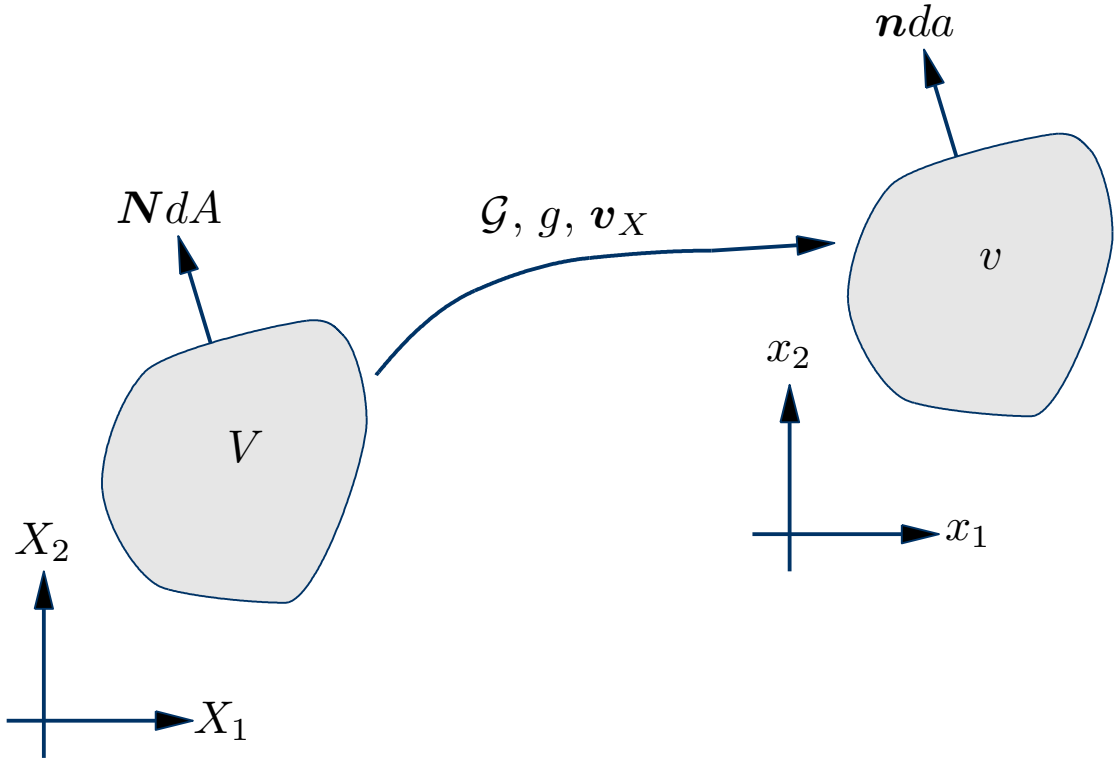
ALE Formulation

- Map from reference domain V to physical deformable domain $v(t)$
- A point $\boldsymbol{X}$ in V is mapped to a point $\boldsymbol{x}(t) = \boldsymbol{G}(\boldsymbol{X}, t)$ in $v(t)$
- Introduce the *mapping deformation gradient* $\boldsymbol{G}$ and the *mapping velocity* $\boldsymbol{v}_X$ as

$$\boldsymbol{G} = \nabla_X \boldsymbol{G}$$

$$\boldsymbol{v}_X = \left.\frac{\partial \boldsymbol{G}}{\partial t}\right|_X$$

and set $g = \det(\boldsymbol{G})$



Transformed Equations

- The system of conservation laws in the physical domain $v(t)$

$$\left.\frac{\partial \boldsymbol{U}_x}{\partial t}\right|_x + \boldsymbol{\nabla}_x \cdot \boldsymbol{F}_x(\boldsymbol{U}_x, \boldsymbol{\nabla}_x \boldsymbol{U}_x) = 0$$

  can be written in the reference configuration V as

$$\left.\frac{\partial \boldsymbol{U}_X}{\partial t}\right|_X + \boldsymbol{\nabla}_X \cdot \boldsymbol{F}_X(\boldsymbol{U}_X, \boldsymbol{\nabla}_X \boldsymbol{U}_X) = 0$$

  where

$$\boldsymbol{U}_X = g\boldsymbol{U}_x \,, \qquad \boldsymbol{F}_X = g\boldsymbol{G}^{-1}\boldsymbol{F}_x - \boldsymbol{U}_X\boldsymbol{G}^{-1}\boldsymbol{v}_X$$

  and

$$\boldsymbol{\nabla}_x \boldsymbol{U}_x = \boldsymbol{\nabla}_X(g^{-1}\boldsymbol{U}_X)\boldsymbol{G}^{-T} = (g^{-1}\boldsymbol{\nabla}_X \boldsymbol{U}_X - \boldsymbol{U}_X \boldsymbol{\nabla}_X(g^{-1}))\boldsymbol{G}^{-T}$$

- *Proof.* See [Persson/Peraire/Bonet 07]

Geometric Conservation Law

- A constant solution in $v(t)$ is not necessarily a solution in V, due to inexact integration of the Jacobian g
  - The time evolution of g is

$$\left.\frac{\partial g}{\partial t}\right|_X - \boldsymbol{\nabla}_X \cdot (g\boldsymbol{G}^{-1}\boldsymbol{v}_X) = 0 \ ,$$

    which in general is non-zero
- Visbal and Gaitonde added source terms to cancel the errors
- Our approach solves instead the conservative system

$$\left.\frac{\partial(\bar{g}g^{-1}\boldsymbol{U}_X)}{\partial t}\right|_X - \boldsymbol{\nabla}_X \cdot \boldsymbol{F}_X = \boldsymbol{0}$$

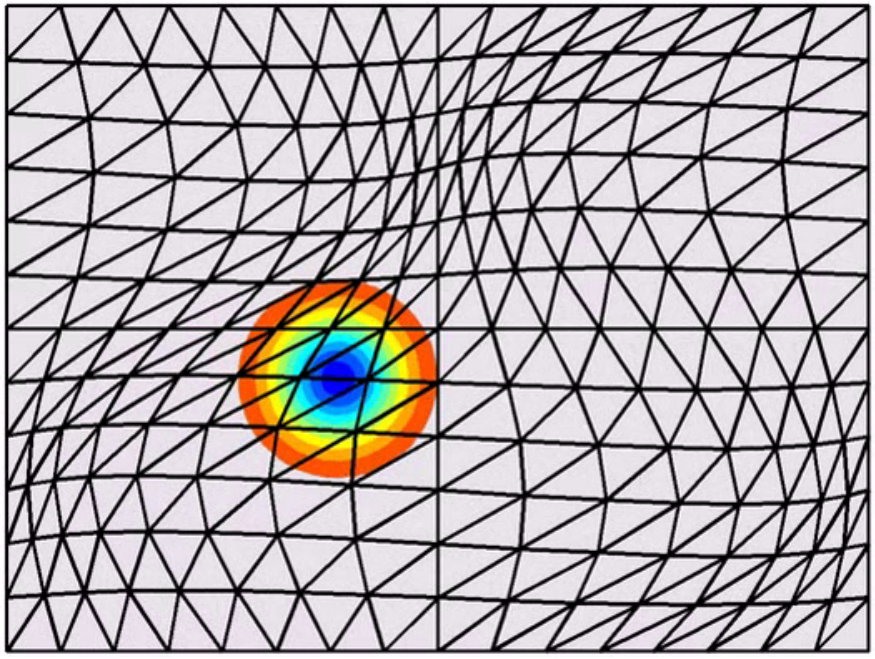
$$\left.\frac{\partial \bar{g}}{\partial t}\right|_X - \boldsymbol{\nabla}_X \cdot (g\boldsymbol{G}^{-1}\boldsymbol{v}_X) = 0$$

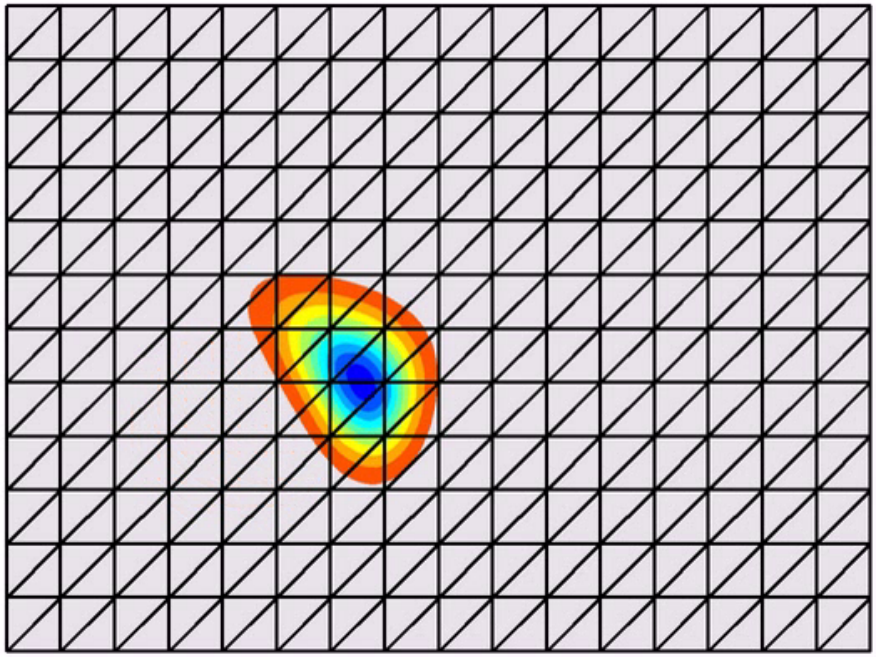
Example: Euler Vortex

- Propagate an Euler vortex on a variable domain with

$$x(\xi,\eta,t) = \xi + 2.0\sin(2\pi\xi/20)\sin(\pi\eta/7.5)\sin(1.0\cdot 2\pi t/t_0)$$
$$y(\xi,\eta,t) = \eta + 1.5\sin(2\pi\xi/20)\sin(\pi\eta/7.5)\sin(2.0\cdot 2\pi t/t_0)$$

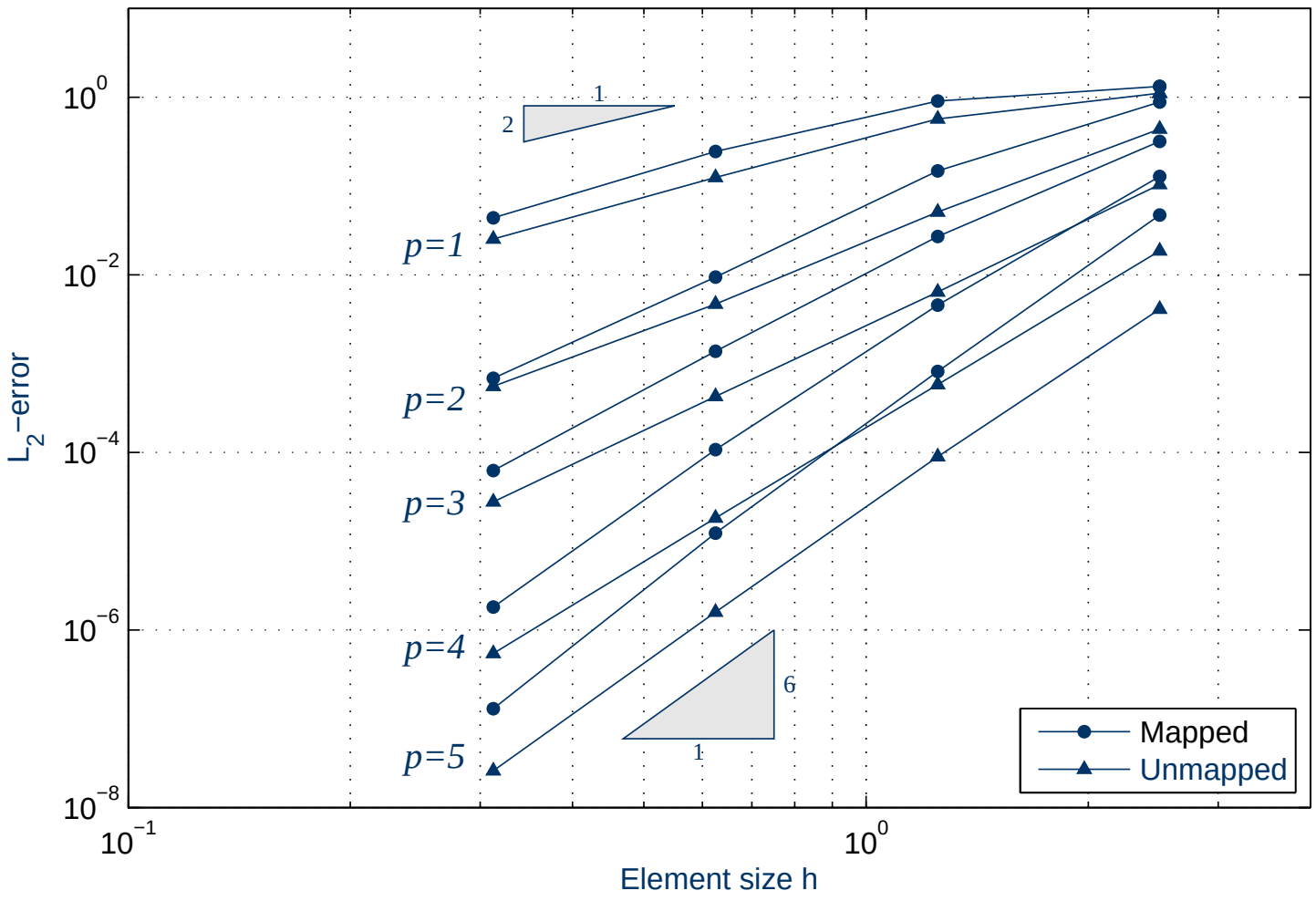
- Mapped scheme – Everything is computed on the reference mesh





Example: Euler Vortex, Convergence

- Optimal order of convergence $O(h^{p+1})$ for mapped scheme

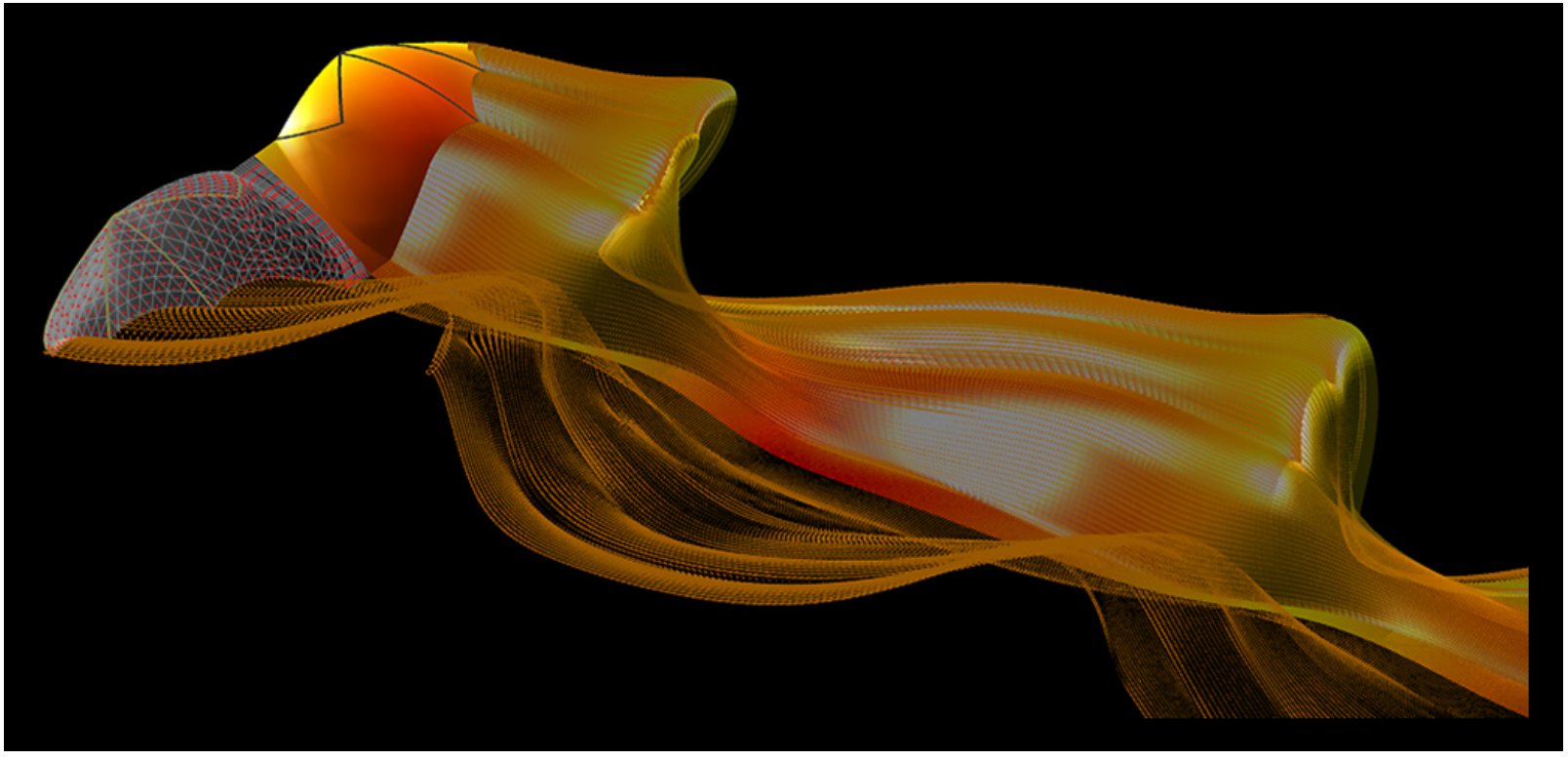


Outline

1. Introduction
2. Discretization and Solvers
   - Mesh Generation
   - The Discontinuous Galerkin Method
   - The CDG Method, Preconditioning, and Stabilization
3. Large Deformation Solid Dynamics
   - High-Order Lagrangian DG Formulations
   - Volumetric Modeling of Beams, Shells, and Membranes
   - J2 Viscoplasticity
4. Deformable Domains
   - Mapping-based ALE Formulation
   - Flapping Flight Applications
5. Conclusions

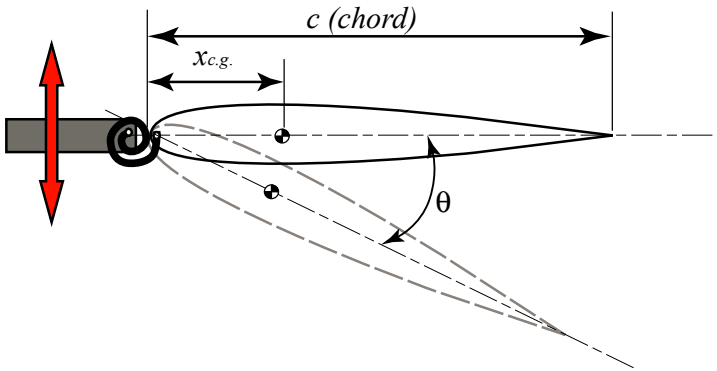
Biologically-Inspired Flapping Flight

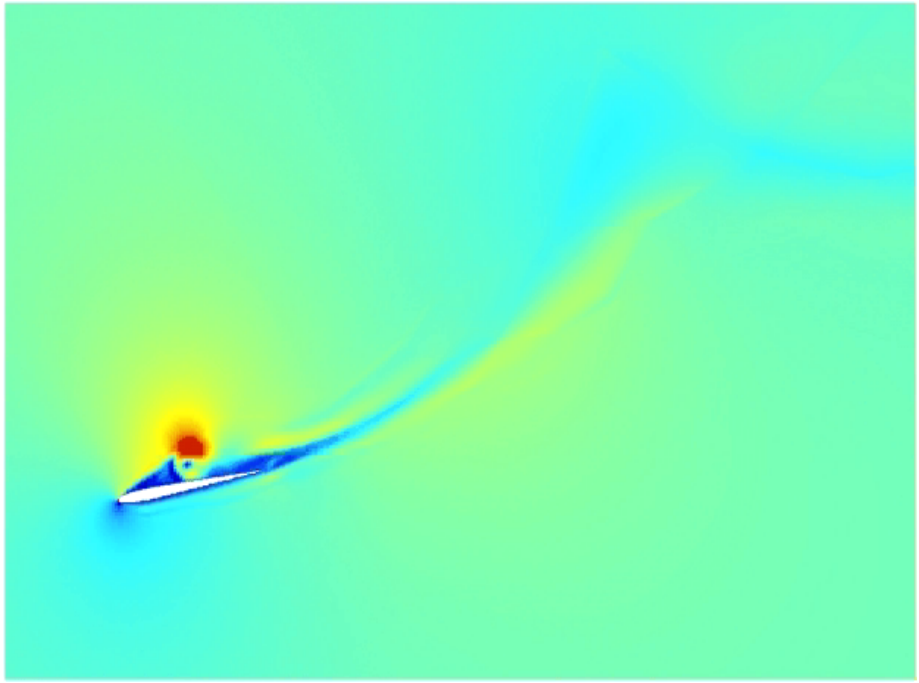
- Development of computational tools for studying flapping flight
- Challenging problems: Deforming domains, fluid-structure interaction, transitional flows, etc



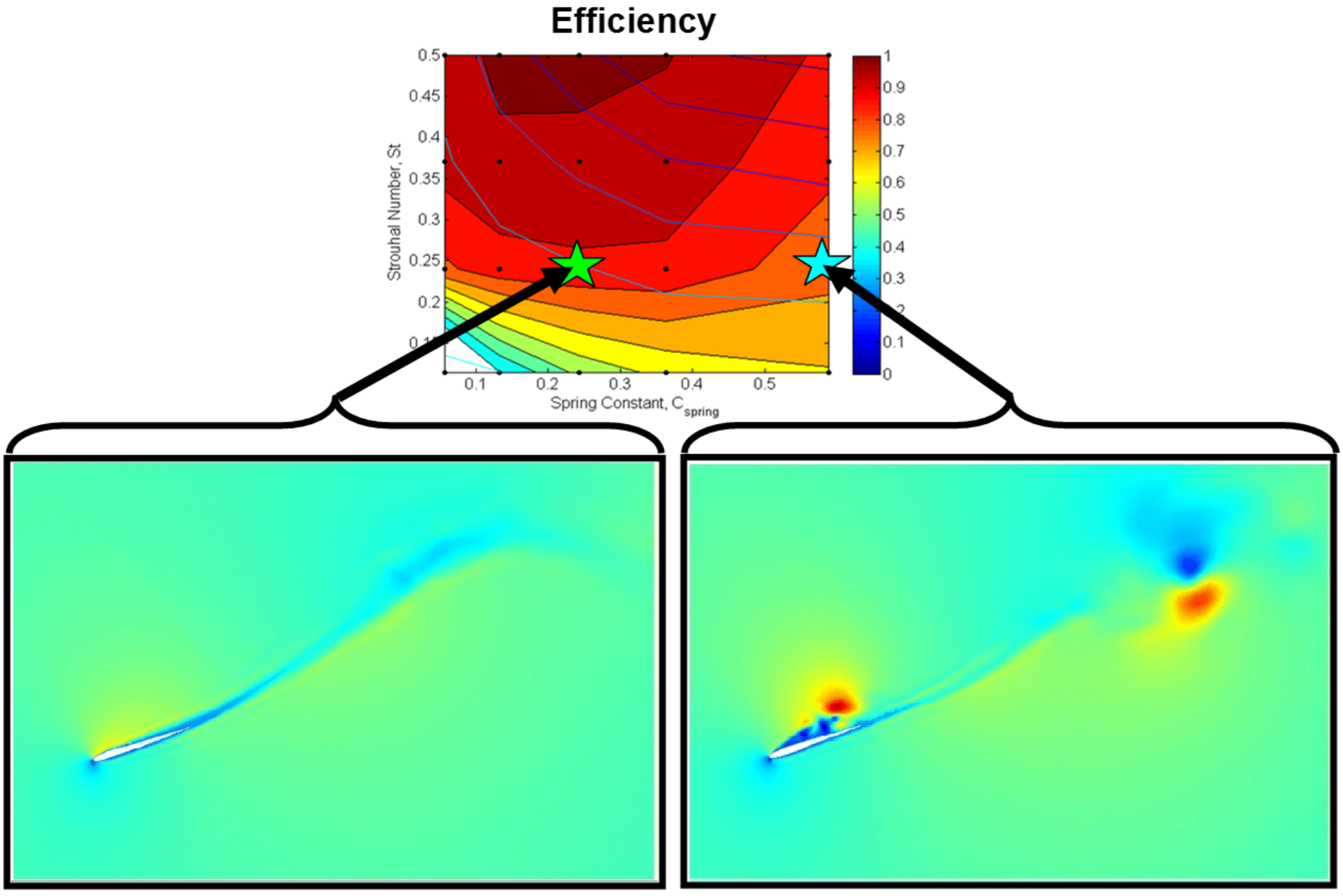
Example: Pitching Airfoil

- Airfoil attached to translating and heaving point by torsional spring
- Fluid properties: $\mathrm{Re} = 5000$, $M = 0.2$
- Forced vertical motion $r_z(t) = r_0 \sin \omega t$ (at leading edge)
- Moment equation: $I\ddot{\theta} + C\theta - S\ddot{r}_z(t) + M_{\text{aero}} = 0$
  - I moment of inertia, C spring stiffness, $S = mx_c$ static unbalance
  - M_{aero} moment from fluid



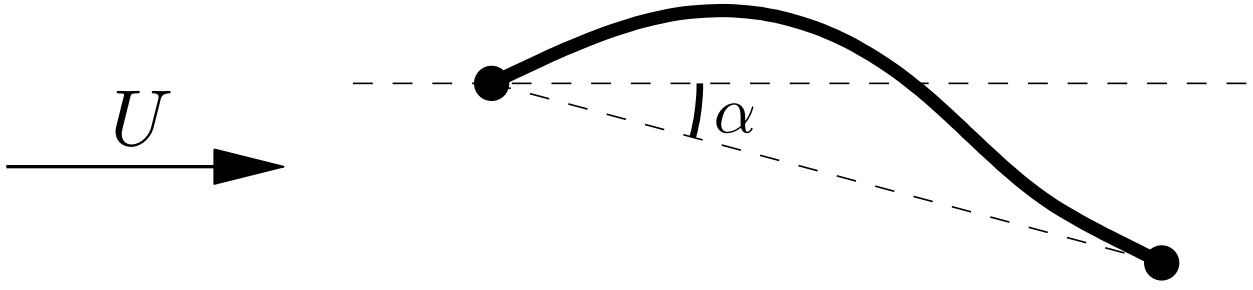


Example: Pitching Airfoil/Flapper Design



Compliant membrane, Fluid-Structure Interaction

- Consider interaction between fluid and a membrane
- Structural model:
  - Hyperelastic Neo-Hookean membrane formulation
  - Viscous, fluid-like dissipation
  - Continuous Galerkin (FEM) discretization
- Fully time-accurate coupling, since at any time t:
  - The membrane shape $x_c(s)$, $y_c(s)$ induces a smooth mapping $\boldsymbol{x}(\boldsymbol{X}, t)$ for the entire domain
  - The DG solution $\boldsymbol{u}(t)$ gives forces $f_x(s)$, $f_y(s)$ on membrane

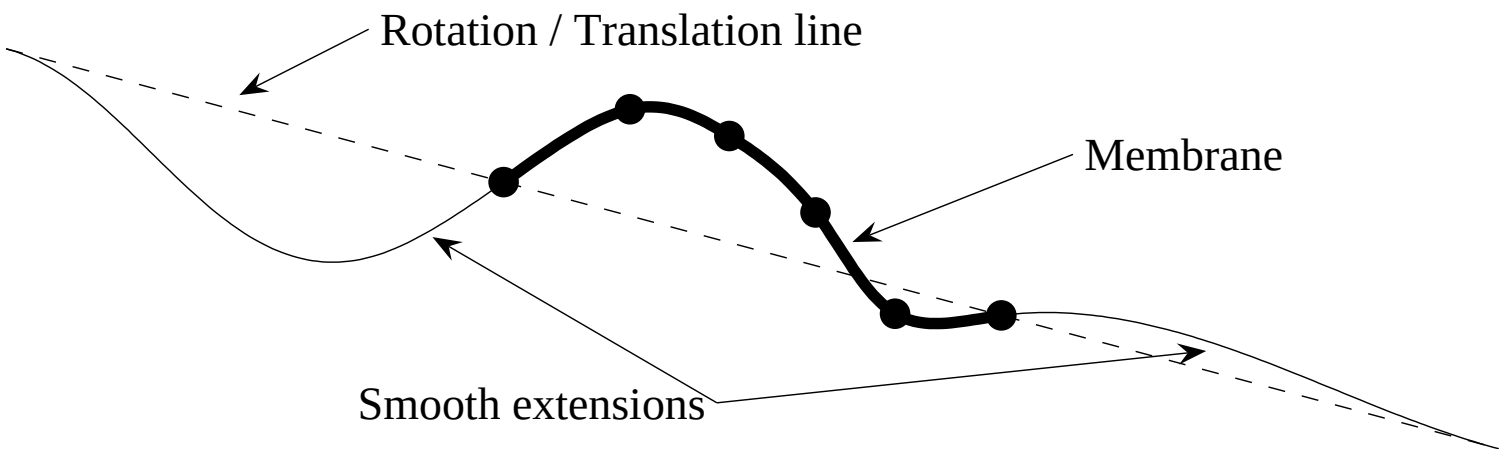


Blending for Deforming Membrane

- Need explicit smooth mapping $\boldsymbol{x} = \boldsymbol{G}(\boldsymbol{X}, t)$
- Define nth order polynomial *blending functions* r_n with $r(0) = 0$, $r(1) = 1$, and $(n-1)/2$ zero derivatives at $x = 0, 1$
- Rotate/translate membrane curve, fit spline with smooth extension
- The blending $b(Y) = (1 - r_n(|Y|/Y_0))$ gives the mapping

$$x = X$$

$$y = Y + b(Y)y_c(X)$$

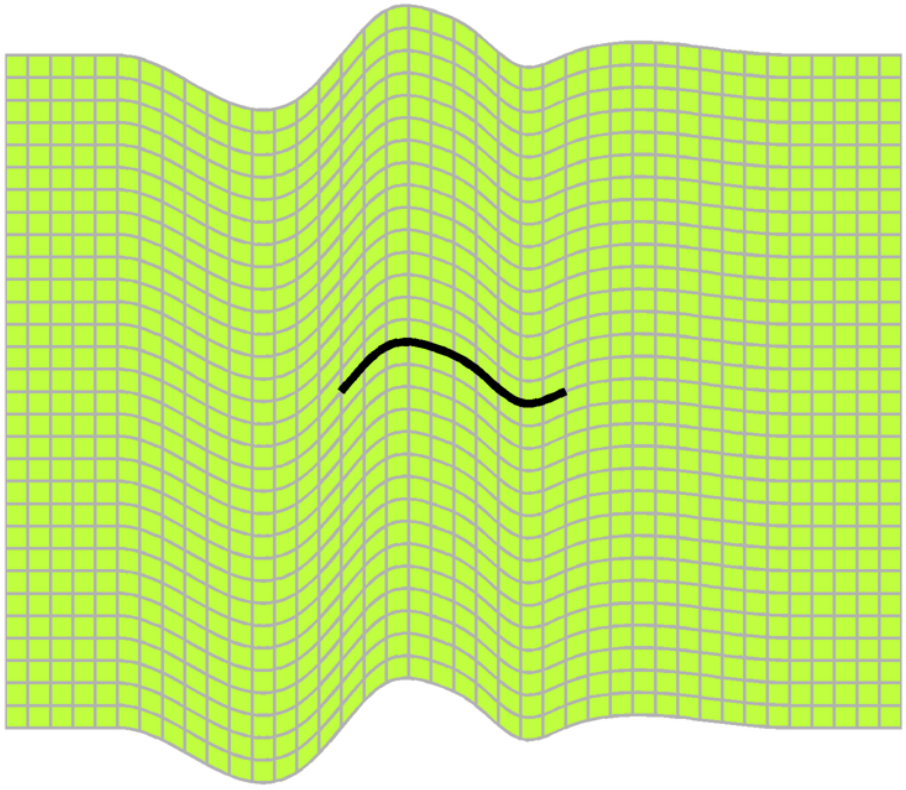


Blending for Deforming Membrane

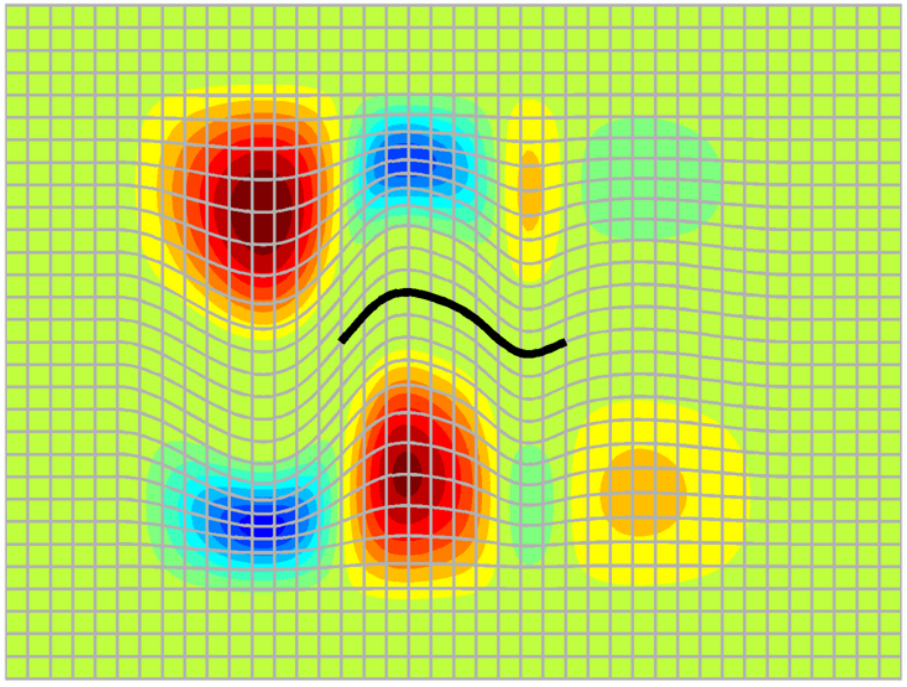
1) Map from spline: $x = X$, $y = Y + y_c(X)$

2) Blend displacements for $|Y| > 0$: $x = X$, $y = Y + b(Y)y_c(X)$

1) Spline mapping



2) Blending in $|Y|$

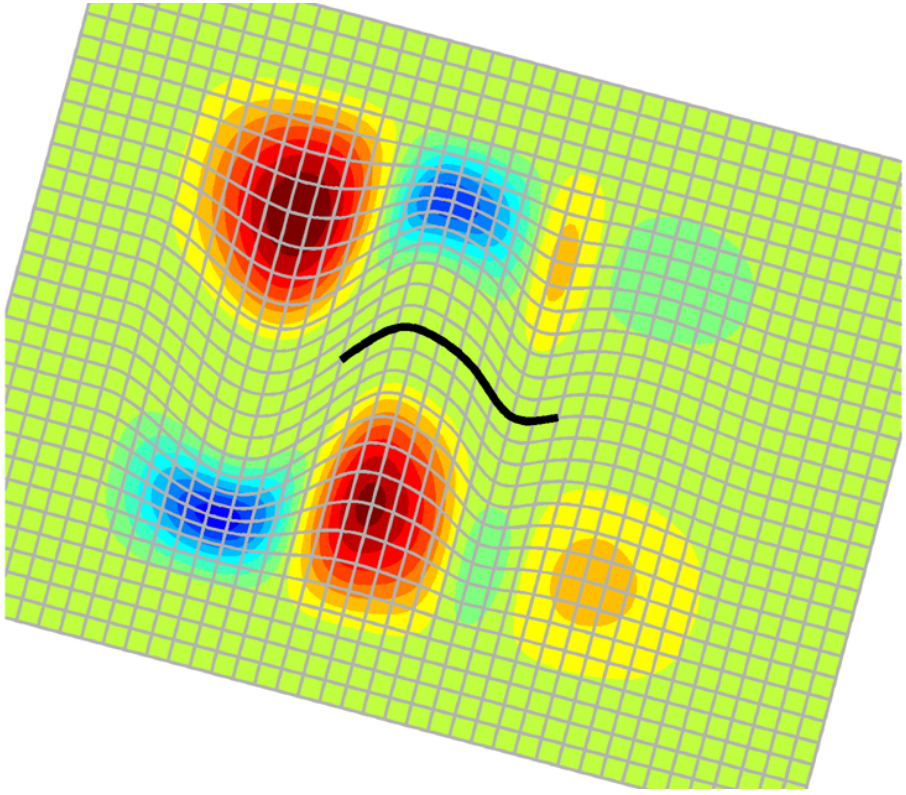


Blending for Deforming Membrane

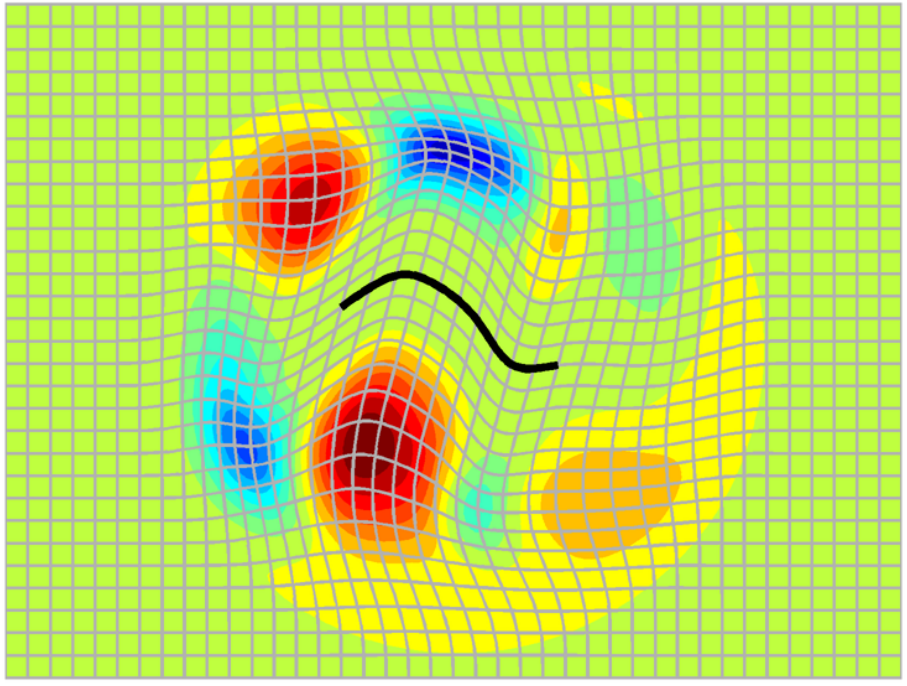
3) Rotate and translate

4) Blend with identity mapping as function of radius

3) Rotation/Translation



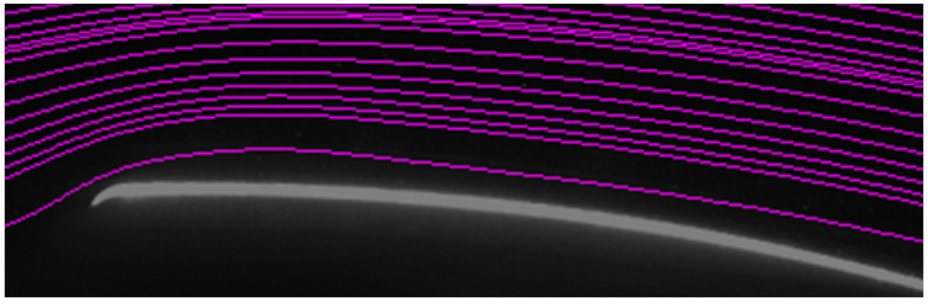
4) Blending in $R = \sqrt{X^2 + Y^2}$



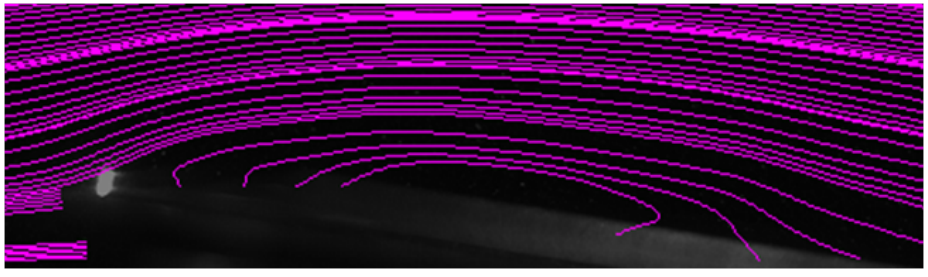
Example: Fluid-Structure Interaction

- Interaction between fluid and a hyperelastic membrane
- Compliancy can alleviate leading edge separation

Experiment (A. Song, Brown U)

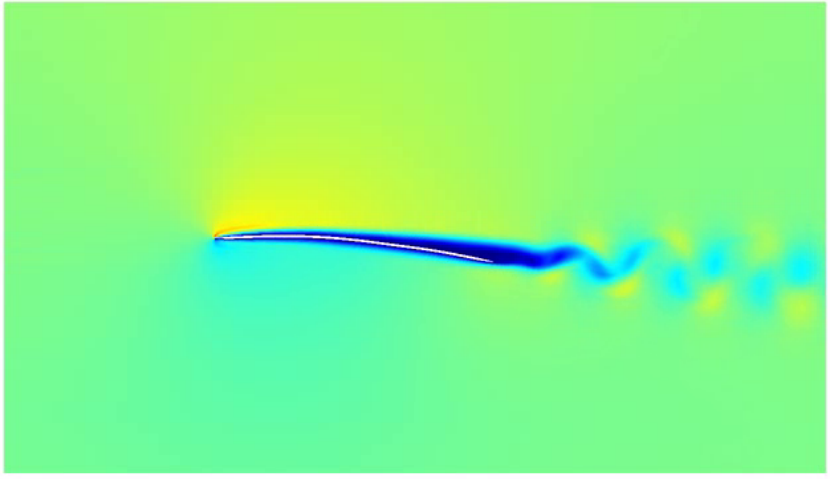


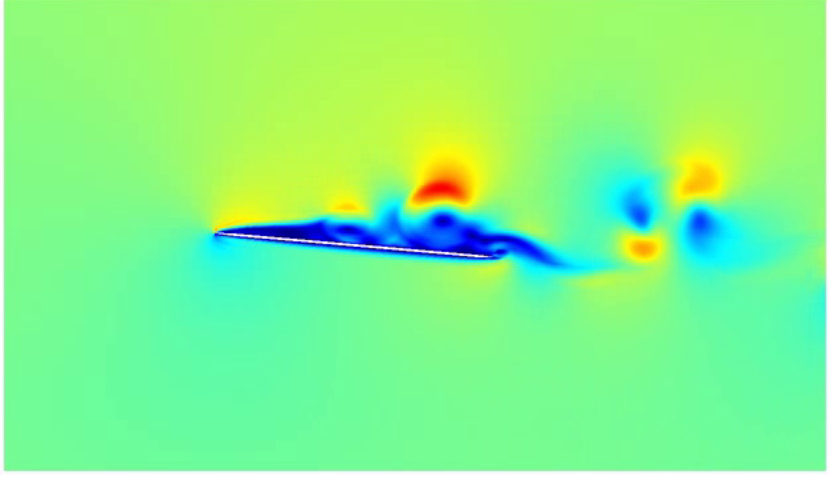
Compliant membrane



Rigid flat plate

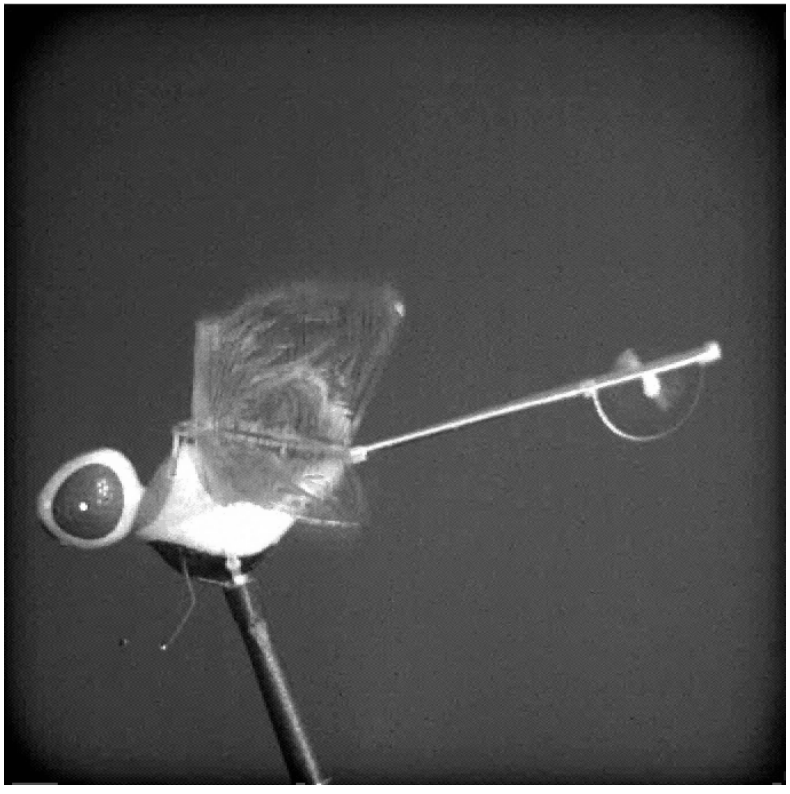
Fluid/membrane simulation



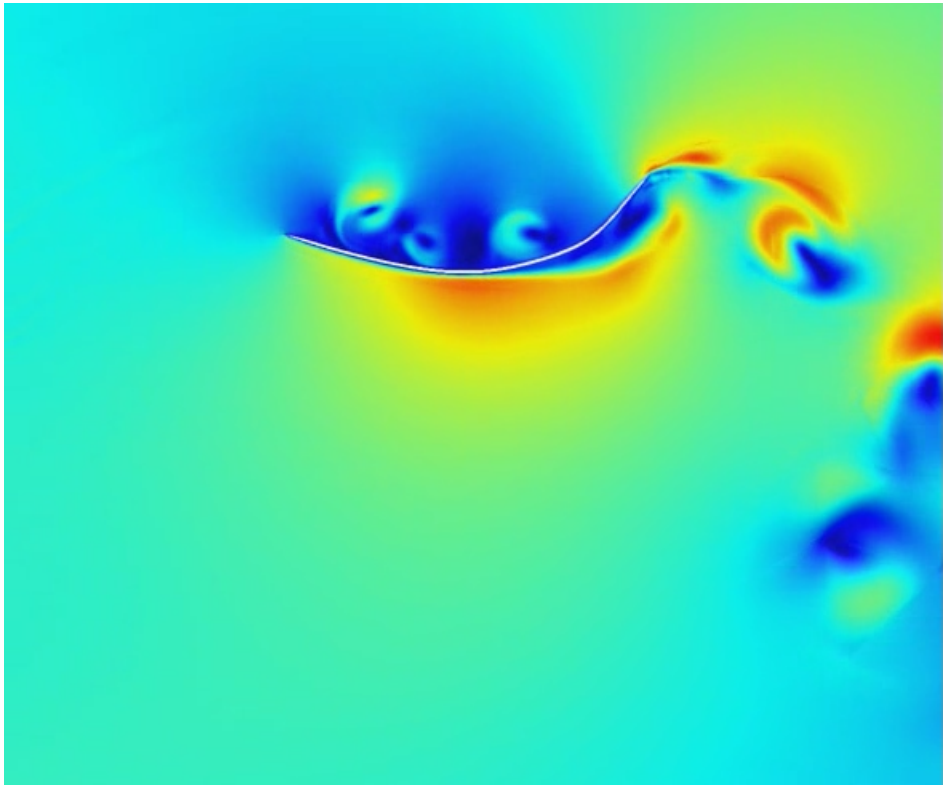


Example: Dragonfly, Compliant Wings

Experiment (A. Song, Brown U)



Fluid/membrane simulation



Conclusions

- Important steps toward practical DG solver for realistic problems in computational mechanics:
  - Efficient discretizations, solvers, and stabilizations
  - Two high-order accurate formulations for solid dynamics
  - Thin structures with tetrahedral elements, fully coupled J2 plasticity
  - Optimal accuracy for deformable domains by mapping approach
- Current work: Full monolithic DG fluid-structure formulation, subcell shock capturing, improved numerical schemes and algorithms, 3-D curved mesh generation, applications in flapping flight, acoustics, instabilities in solids, and extreme loading conditions