High Order Discontinuous Galerkin Methods for Fluid and Solid Mechanics

Per-Olof Persson

Department of Mathematics
University of California, Berkeley

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

High Order Non-Oscillatory Methods for Wave Propagation,
Flow and Transport Problems 2009, Trento, Italy (HONOM 2009)

University of California Berkeley

March 30, 2009

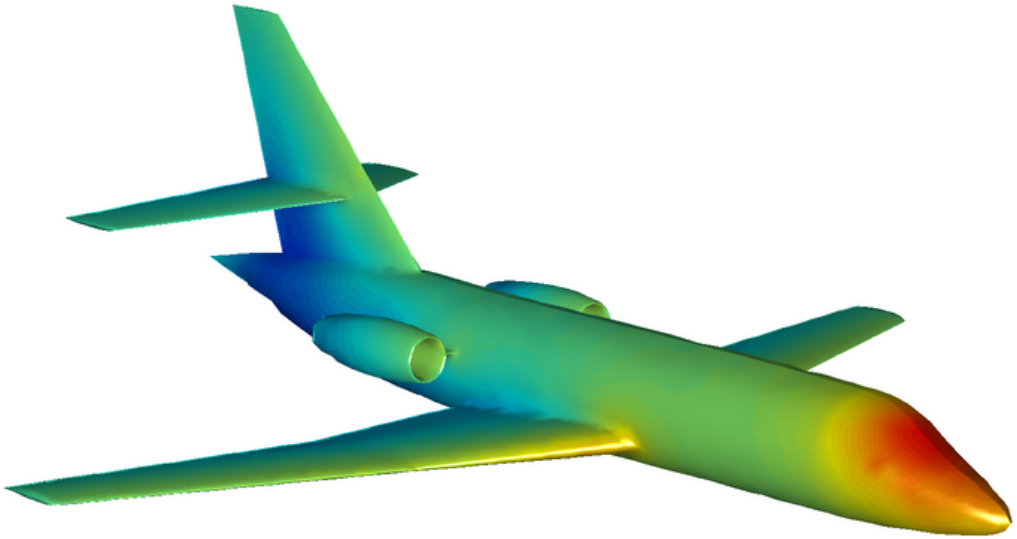
Berkeley Lab

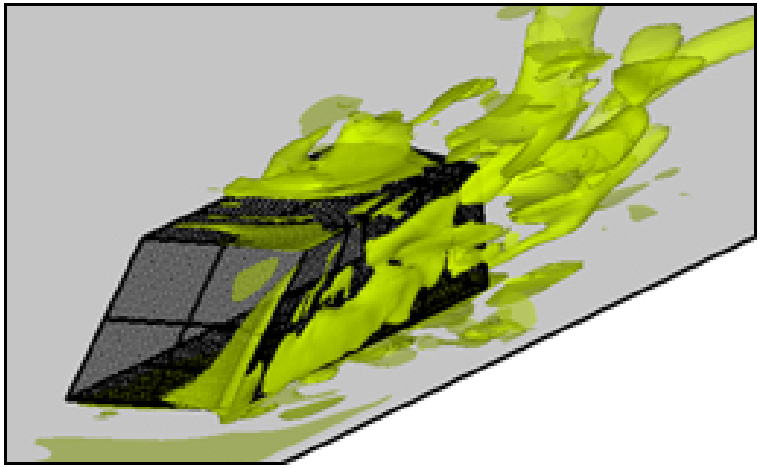
Outline

Outline

Motivation

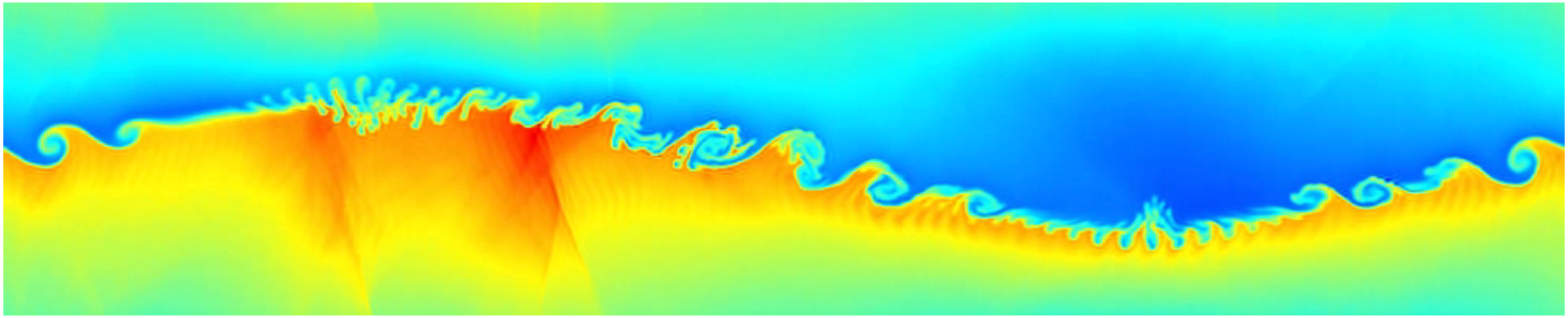
- Need for higher fidelity predictions in computational mechanics
  - DNS/LES/DES applications
  - Accurate RANS for engineering applications (drag prediction, rotor dynamics, fluid/structure interaction, flapping flight)
  - Computational aeroacoustics (direct solution of compressible flow, accurate computation of noise sources)
  - Other problems involving wave propagation, multiple scale phenomena, and non-linear interactions





Example: Aeroacoustics and K-H Instability

- Inspired by [Munz et al 03] (linearized)
- Nonlinear behavior: Large scale acoustic wave interacts with small scale flow features, leading to vorticity generation
- High-order accuracy *essential* to capture long-range wave propagations and nonlinear interactions
- $p = 7$ (8th order accuracy), 140-by-28 square elements



Motivation

- Fundamental properties of Discontinuous Galerkin (DG) methods:

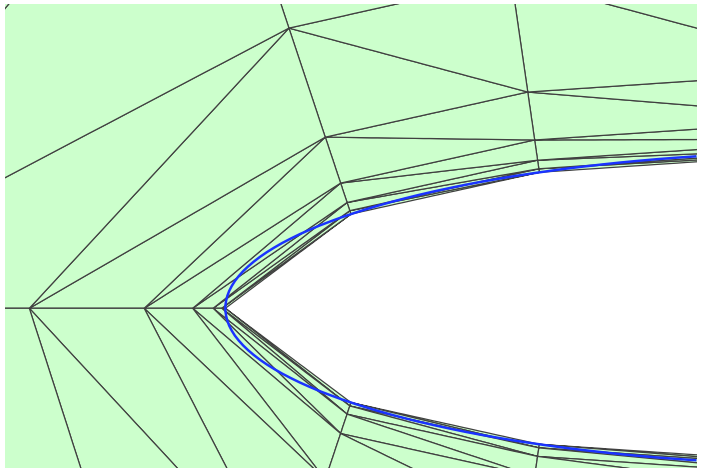
	FVM	FDM	FEM	DG
1) High-order/Low dispersion	✗	✓	✓	✓
2) Unstructured meshes	✓	✗	✓	✓
3) Stability for conservation laws	✓	✓	✗	✓

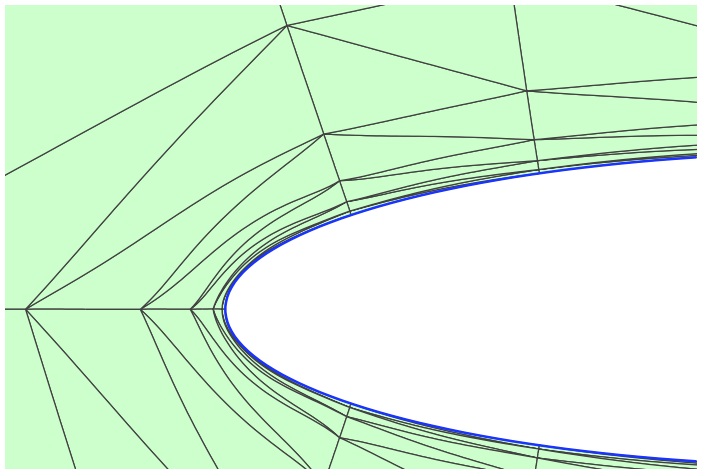
- However, several problems to resolve:
  - High CPU/memory requirements (compared to FVM or H-O FDM)
  - Low tolerance to under-resolved features
  - High-order geometry representation and mesh generation
- *The challenge is to make DG competitive for real-world problems*

Outline

Motivation

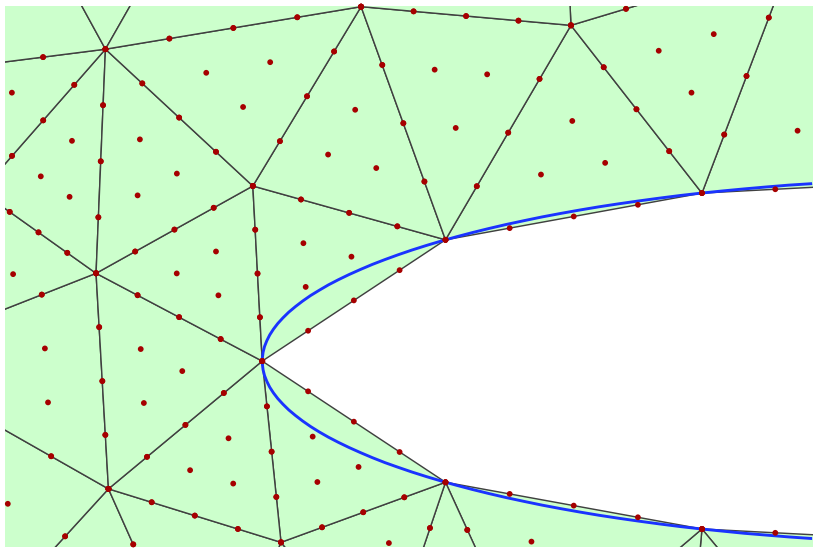
- Automatic generation of non-inverted curved elements largely an unresolved problem
- In general this is a global problem, affecting many elements except for simple isotropic 2-D meshes
- In [Persson/Peraire 09], we proposed a *non-linear solid mechanics* approach, where the mesh is considered an elastic deformable solid



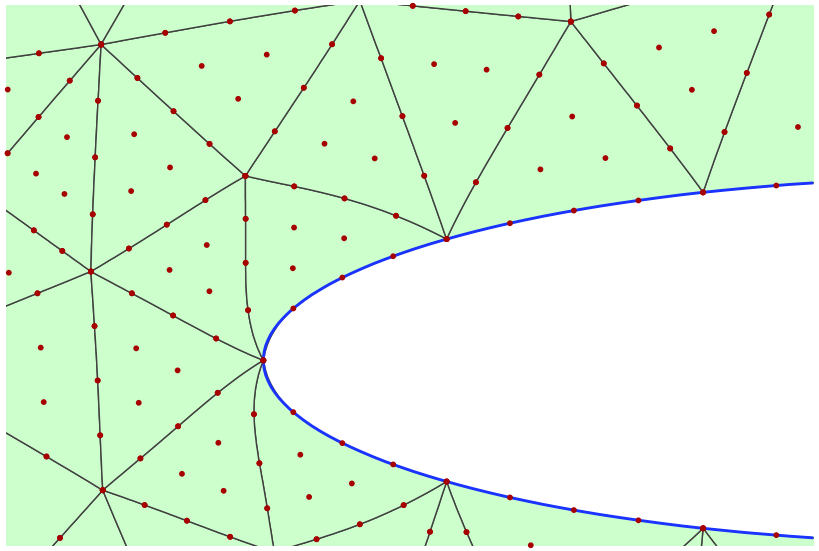


Curved Mesh Generation using Solid Mechanics

- The initial, straight-sided mesh corresponds to undeformed solid
- External forces come from the true boundary data
- Solving for a force equilibrium gives the deformed, curved, boundary conforming mesh
- Bottom-up approach can be used to obtain the boundary data



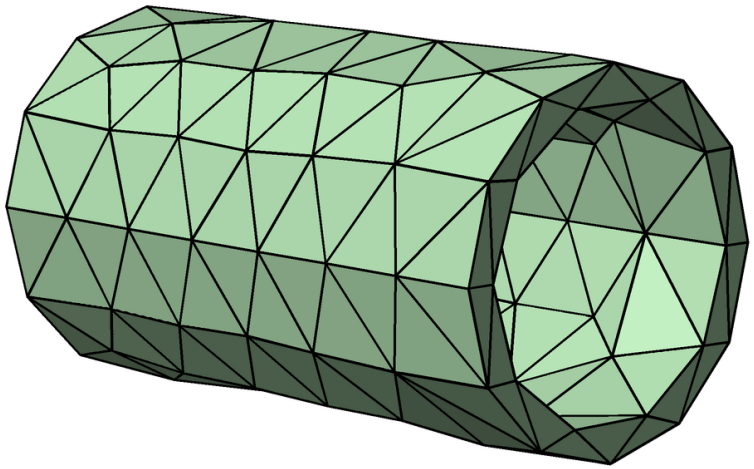
Reference domain, initial configuration

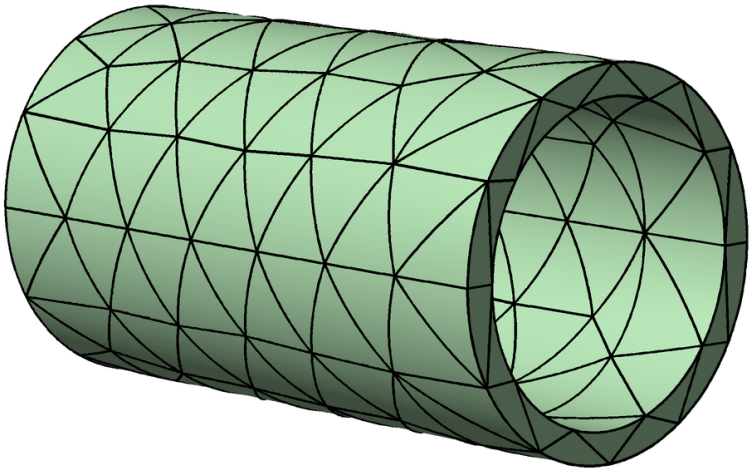


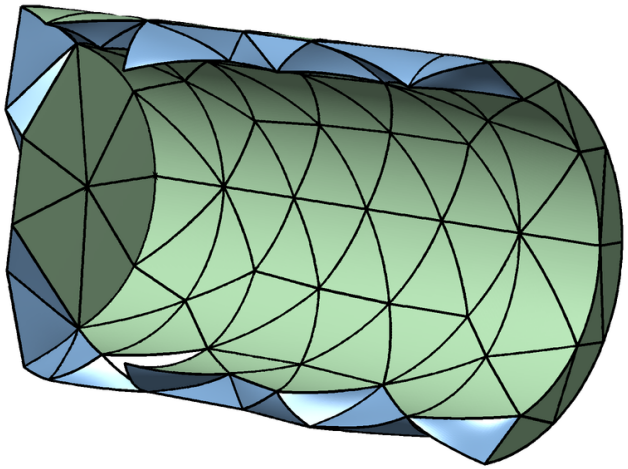
Equilibrium solution, final curved mesh

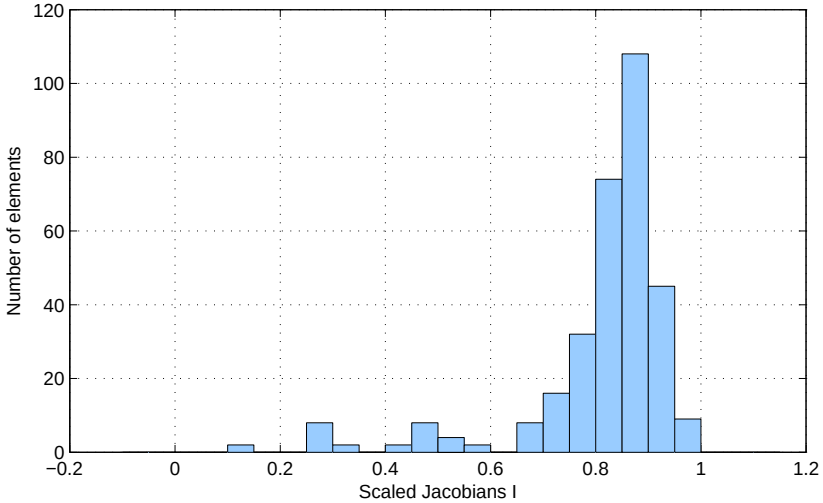
Tetrahedral Mesh of Cylindrical Component

- Large curved deformations handled without element inversion



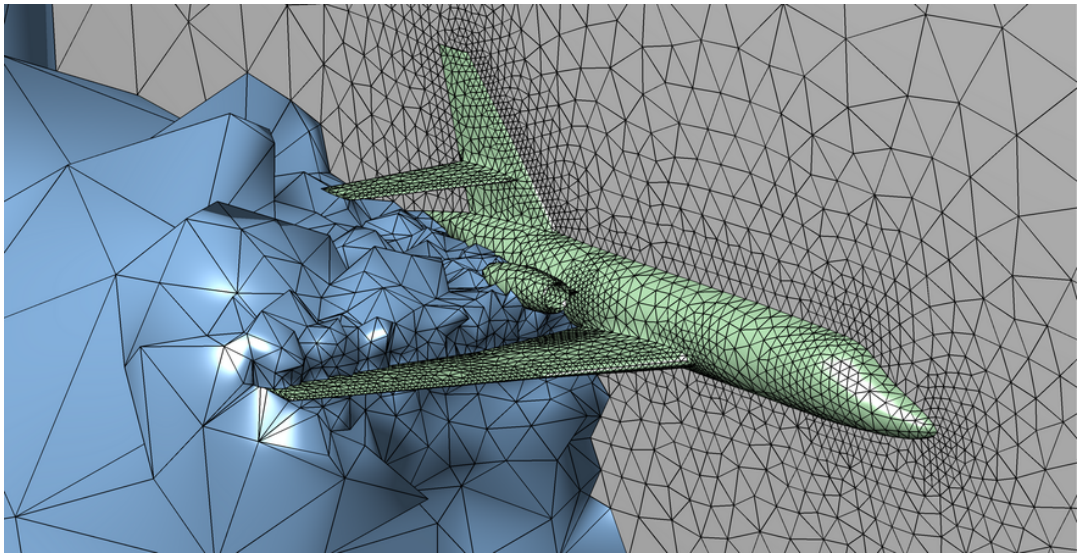




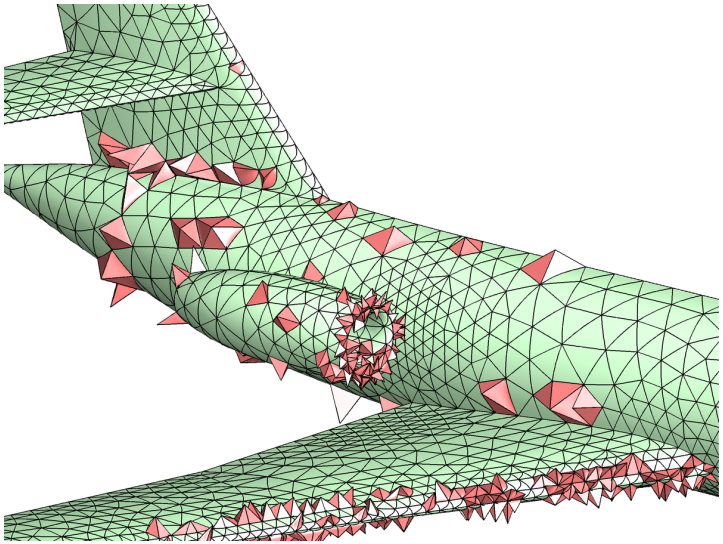


Tetrahedral Mesh of Falcon Aircraft

- Real-world mesh with coarse but realistic elements
- Unstructured Delaunay refinement mesh, with highly curved boundary segments
- Many elements would invert with a local element-wise approach



Tetrahedral mesh



Elements with $I < 0.5$

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:

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

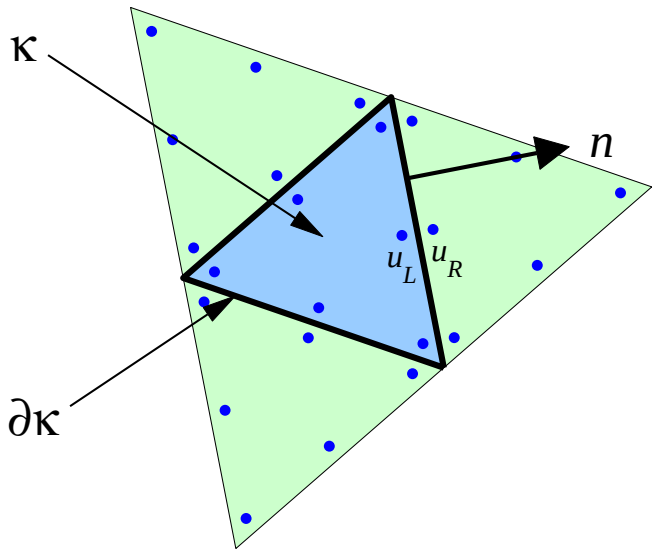
The Discontinuous Galerkin Method

- 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



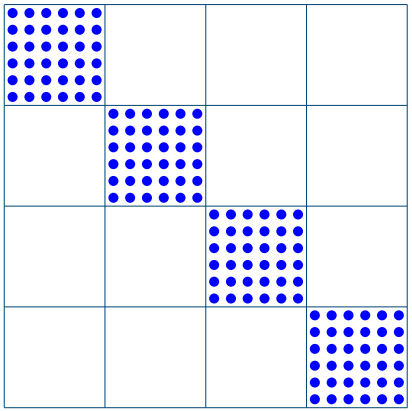
The DG Method – Observations

- Reduces to the finite volume method for $p = 0$:

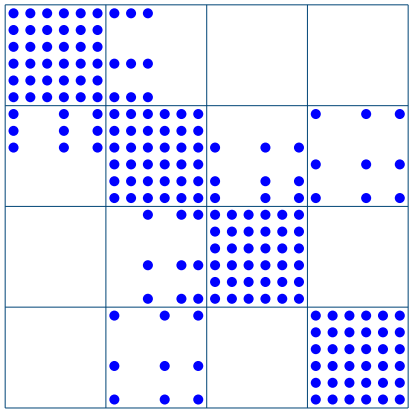
$$(\boldsymbol{u}_h)_t A_\kappa + \int_{\partial\kappa} \hat{\mathcal{F}}_i(\boldsymbol{u}_h^+, \boldsymbol{u}_h^-, \hat{\boldsymbol{n}})\, ds = 0$$

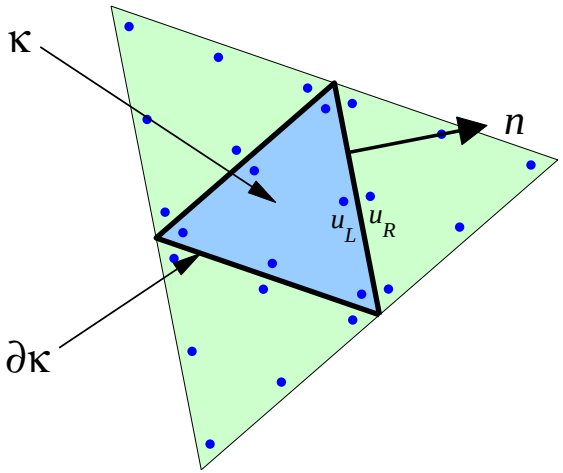
- Boundary conditions enforced naturally for any degree p
- Block-diagonal mass matrix (no overlap between basis functions)
- Block-wise compact stencil – neighboring elements connected

Mass Matrix



Jacobian





Viscous Discretization

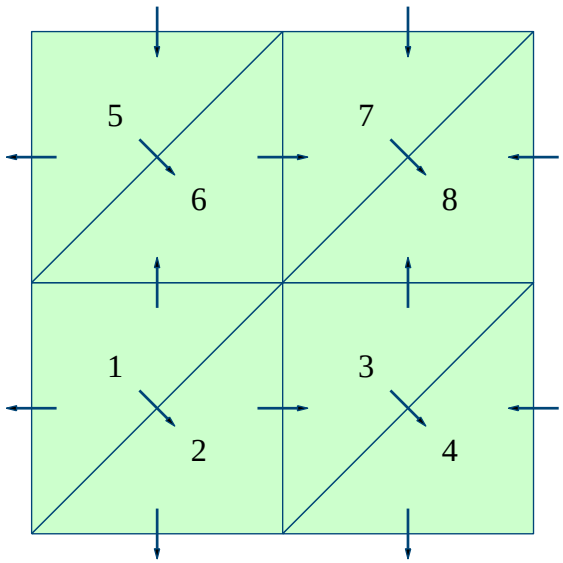
- General approach for second derivatives:
  - Write as system of first order equations [Arnold et al 02]:

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

  - Discretize using DG, choose appropriate numerical fluxes $\hat{\boldsymbol{\sigma}}$, $\hat{u}$
- Various schemes have been proposed:
  - *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 Local DG Method

- Consider Poisson problem $-\nabla \cdot (\kappa \nabla u) = f$
- Write as system of first order equations,



$$-\nabla \cdot \boldsymbol{\sigma} = f$$

$$\boldsymbol{\sigma} = \kappa \nabla u$$

- Use numerical inter-element fluxes

$$\hat{\boldsymbol{\sigma}} = \{\boldsymbol{\sigma}_h\} - C_{11}[u_h] + \boldsymbol{C}_{12}[\boldsymbol{\sigma}_h]$$

$$\hat{u} = \{u_h\} - \boldsymbol{C}_{12} \cdot [u_h]$$

where $\{\cdot\}, [\cdot]$ denote averaging and difference

- In particular, choosing $\boldsymbol{C}_{12} = 1$ or -1 depending on a *switch* for each edge, will upwind/downwind $\hat{\boldsymbol{\sigma}}, \hat{u}$

The Local DG Scheme

- Solving for the variables $\boldsymbol{\sigma}_h$ gives

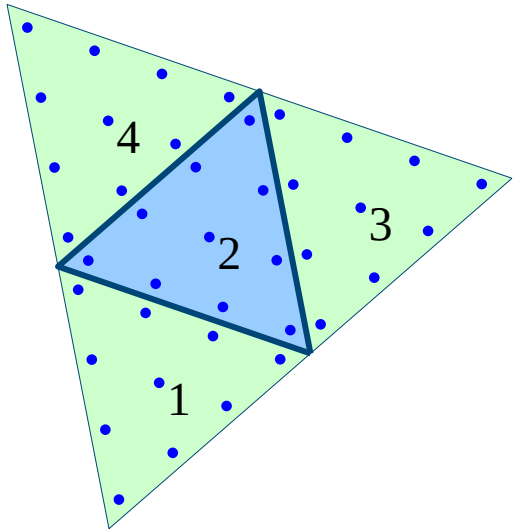
$$\boldsymbol{\sigma}_h = \kappa \nabla_h u_h + \bar{\boldsymbol{\sigma}}_h$$

where

$$\bar{\boldsymbol{\sigma}}_h = \kappa r([u_h]) + \kappa l(\boldsymbol{C}_{12} \cdot [u_h]) + \text{ boundary terms}$$

and $r(\phi)$ and $l(q)$ are *lifting operators* (essentially L_2-projections)

- In general, this introduces non-local couplings since the lifting operators involve all element edges



The Compact DG Scheme

- In the CDG scheme, we split the lifting operators into sums of edge-wise lifting operators $r^e(\phi)$, $l^e(q)$, and set

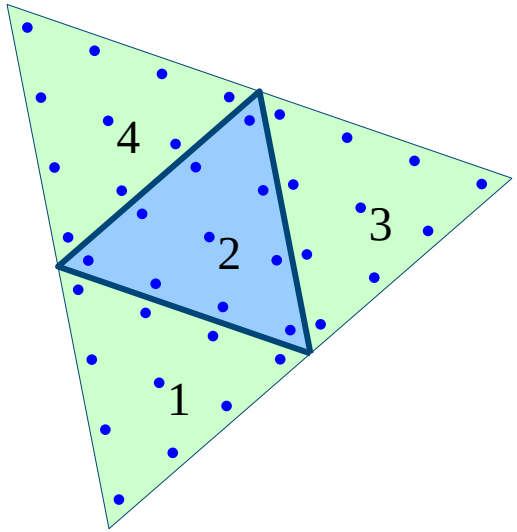
$$\hat{\boldsymbol{\sigma}} = \{\boldsymbol{\sigma}_h^e\} - C_{11}[u_h] + \boldsymbol{C}_{12}[\boldsymbol{\sigma}_h^e]$$

$$\hat{u} = \{u_h\} - \boldsymbol{C}_{12} \cdot [u_h]$$

where $\boldsymbol{\sigma}_h^e = \kappa \nabla_h u_h + \bar{\boldsymbol{\sigma}}_h^e$, with

$$\bar{\boldsymbol{\sigma}}_h^e = \kappa r^e([u_h]) + \kappa l^e(\boldsymbol{C}_{12} \cdot [u_h]) + \text{ boundary terms}$$

- Since only the lifting operator corresponding to the current edge is used, only neighboring elements are connected



Error Estimates

- In primal form, the LDG scheme becomes (ignoring bnd terms):

$$\int_\Omega \kappa(r([u]) + l(\boldsymbol{C}_{12} \cdot [u])) \cdot (r([v]) + l(\boldsymbol{C}_{12} \cdot [v]))\, dx =$$
$$\sum_{e \in \mathcal{E}_i} \sum_{f \in \mathcal{E}_i} \int_\Omega \kappa(r^e([u]) + l^e(\boldsymbol{C}_{12} \cdot [u])) \cdot (r^f([v]) + l^f(\boldsymbol{C}_{12} \cdot [v]))\, dx$$

- The CDG scheme excludes some terms that are indefinite:

$$\sum_{e \in \mathcal{E}_i} \int_\Omega \kappa(r^e([u]) + l^e(\boldsymbol{C}_{12} \cdot [u])) \cdot (r^e([v]) + l^e(\boldsymbol{C}_{12} \cdot [v]))\, dx =$$
$$\sum_{e \in \mathcal{E}_i} \sum_{f \in \mathcal{E}_i} \delta_{ef} \int_\Omega \kappa(r^e([u]) + l^e(\boldsymbol{C}_{12} \cdot [u])) \cdot (r^f([v]) + l^f(\boldsymbol{C}_{12} \cdot [v]))\, dx$$

- Non-compact terms are eliminated but the scheme remains stable

Error Estimates

- Coercivity and boundedness for the CDG scheme same as for LDG, leading to a-*priori* estimates:

$$|||u - u_h||| \leq Ch^p |u|_{p+1,\Omega}$$

and

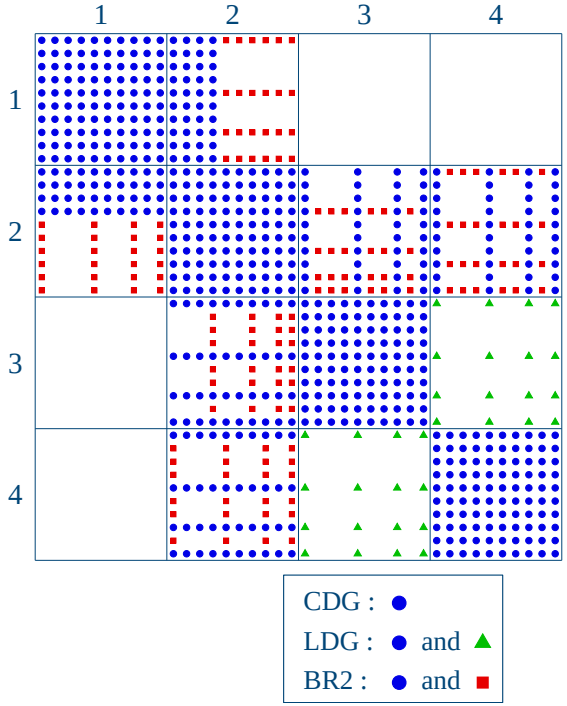
$$||u - u_h||_{0,\Omega} \leq Ch^{p+1} |u|_{p+1,\Omega}$$

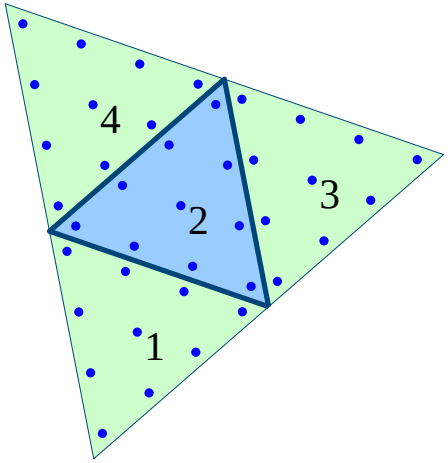
with the norm

$$|||v|||^2 = \sum_{K \in \mathcal{T}_h} |v|_{1,K}^2 + \sum_{e \in \mathcal{E}_i} ||r_e([v])||_{0,\Omega}^2 + \sum_{e \in \partial\Omega_D} ||r_D(v)||_{0,\Omega}^2$$

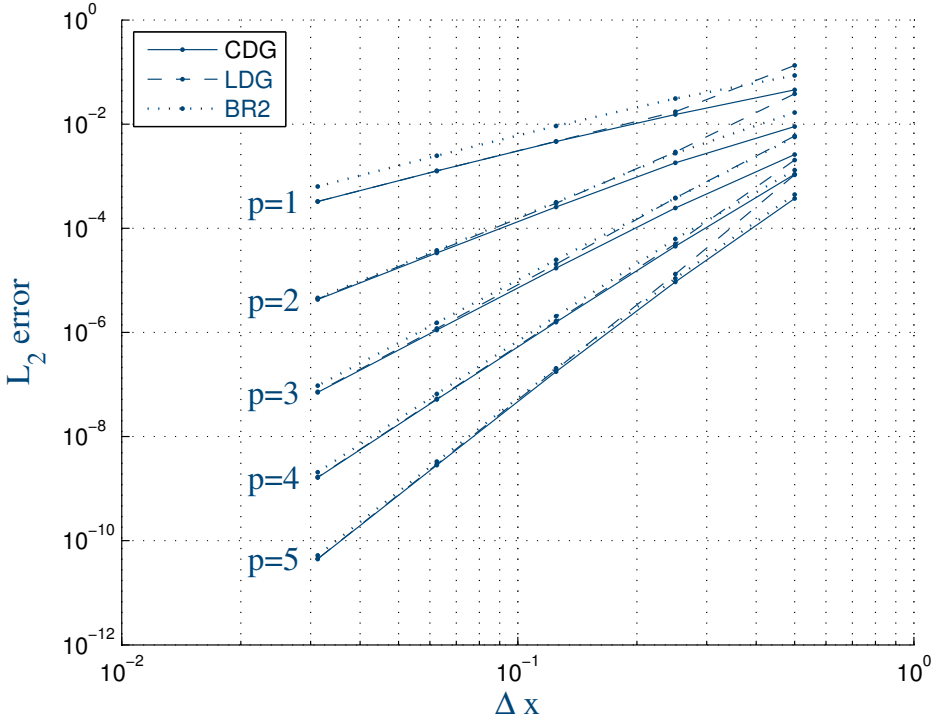
- Assumes $C_{11} = \mathcal{O}(h^{-1})$, but is observed numerically for $C_{11} = 0$

The CDG Method – Summary



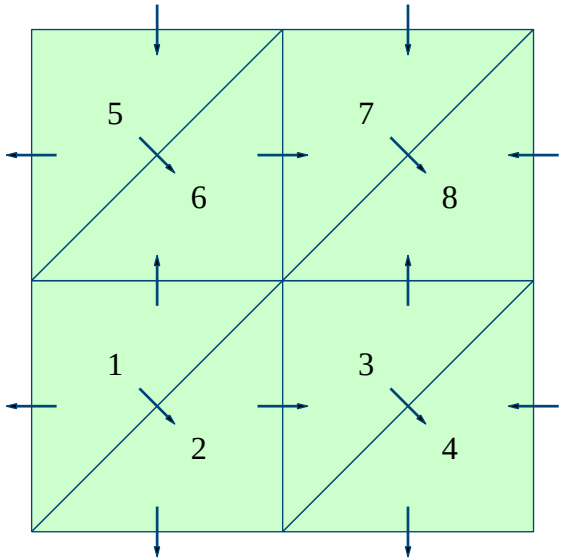


- Modification of numerical fluxes in LDG scheme
- Excludes non-compact and indefinite terms
- Provably optimal accuracy $\mathcal{O}(h^{p+1})$
- Higher stability/accuracy than LDG/BR2
- Sparser than LDG/BR2/IP



Switches and Null-space Dimensions

- Unlike the LDG scheme, the CDG scheme appears to be stable for $C_{11} = 0$ and an *inconsistent switch* such as highest element number
- Simple test [Sherwin et al 05]: Poisson problem, periodic boundary conditions, expected nullspace dimension = 1

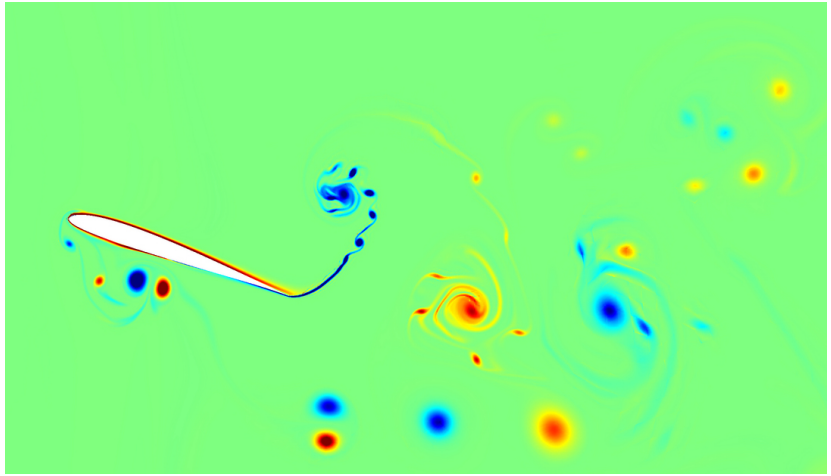


Nullspace dimension

Polynomial order p		1	2	3	4	5	6	7
Consistent switch	CDG	1	1	1	1	1	1	1
	LDG	1	1	1	1	1	1	1
Natural switch	CDG	1	1	1	1	1	1	1
	LDG	3	4	5	6	7	8	9

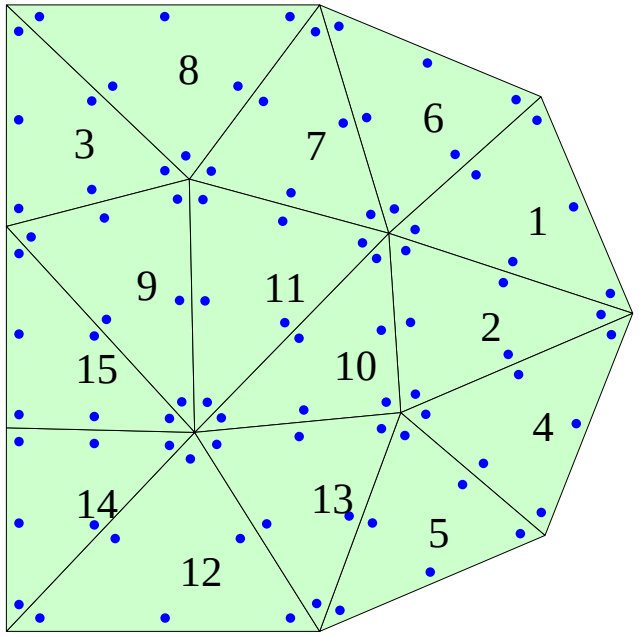
Implicit Time Integration

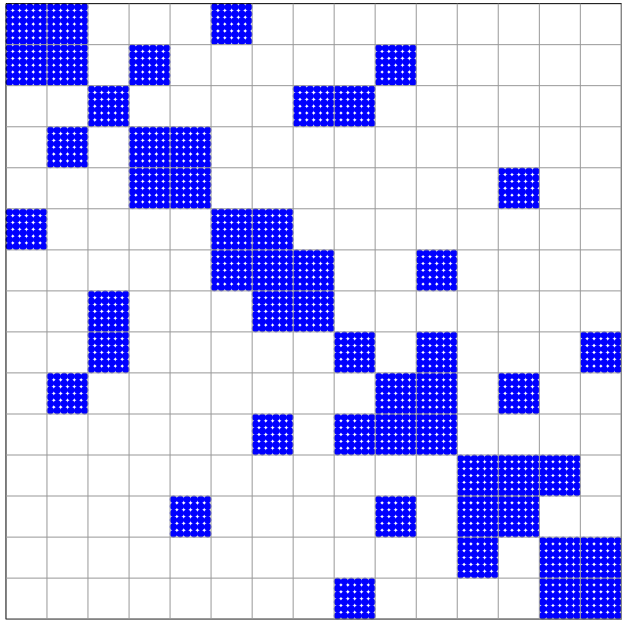
- Implicit solvers typically required because of CFL restrictions from viscous effects, low Mach numbers, and adaptive/anisotropic grids
- Jacobian matrices are large even at $p = 2$ or $p = 3$, however:
  - They are required for non-trivial preconditioners
  - They are very expensive to recompute
- Therefore, we consider matrix-based Newton-Krylov solvers
- In [Persson/Peraire, SISC'08], we proposed efficient preconditioners with block-ILU(0) and automatic element ordering
- Distributed parallel versions developed in [Persson '09]



Jacobian Matrix Sparsity Structure

- Dual mesh connectivity, with each entry a large complete graph $\boldsymbol{A}_{ij}$
- Off-diagonal blocks actually sparser with CDG, but assume dense for simplicity
- Size N of submatrices $\boldsymbol{A}_{ij}$ is often > 100
- Block-based storage format essential for high performance using BLAS routines
- For other structures (e.g. elimination matrices), use block-wise compressed column





Preconditioners for Krylov Methods

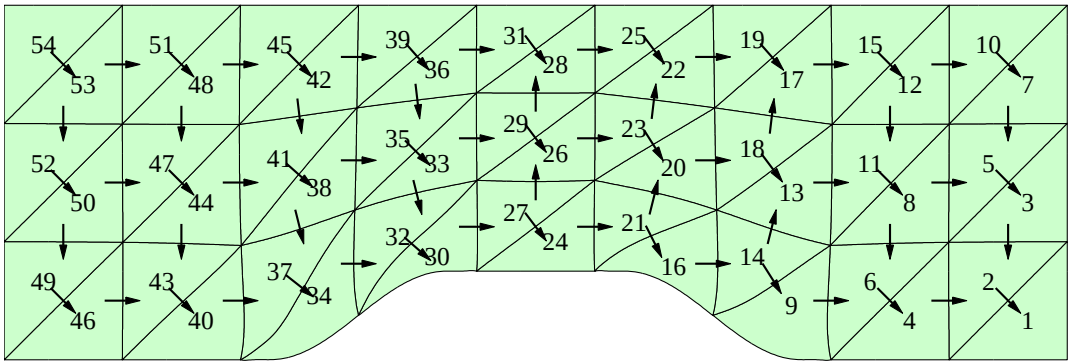
- Preconditioning required for fast convergence in Krylov methods
- Standard point-wise Jacobi, ILU, etc, ineffective for DG
- Block Jacobi and Gauss Seidel are generally poor:

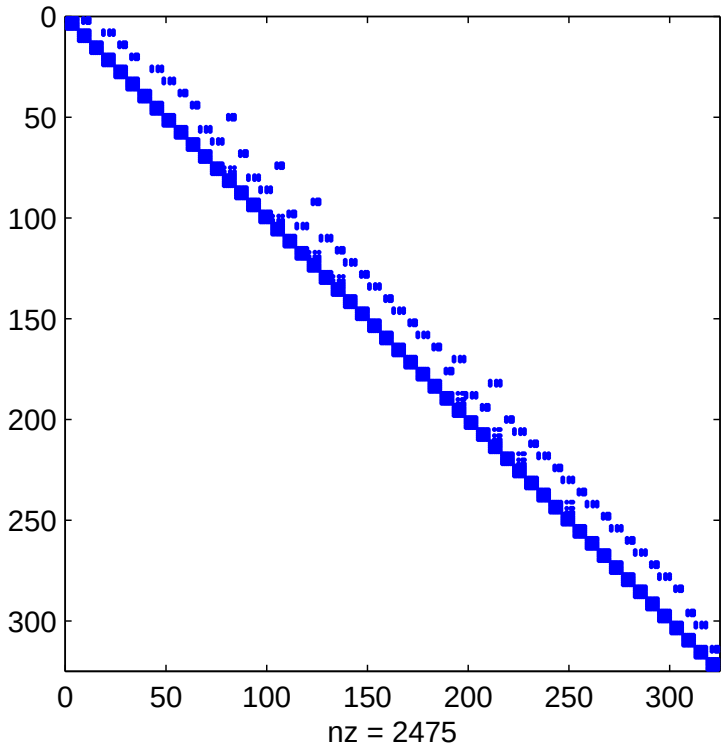
$$\tilde{\boldsymbol{A}}_{ij}^{\mathrm{J}} = \begin{cases} \boldsymbol{A}_{ij} & \text{if } i = j, \\ \boldsymbol{0} & \text{if } i \neq j, \end{cases} \qquad \text{and} \qquad \tilde{\boldsymbol{A}}_{ij}^{\mathrm{GS}} = \begin{cases} \boldsymbol{A}_{ij} & \text{if } i \leq j, \\ \boldsymbol{0} & \text{if } i > j. \end{cases}$$

- Block-ILU(0) algorithm $\tilde{\boldsymbol{A}}^{\mathrm{ILU}} = \tilde{\boldsymbol{L}}\tilde{\boldsymbol{U}}$ effective for good orderings
- Block-ILU(0) postsmoothing for coarse scale correction [Persson/Peraire '08], cheap, general purpose preconditioner

Element Order Dependency

- Properties of Gauss Seidel and ILU preconditioners highly dependent on the ordering of the elements
- For *upwinded scalar convection*, “ordering by lines” gives an optimal upper triangular matrix
- But for viscous or multivariate problems, best ordering not clear
- Matrix-approach: Minimize error in the approximations, rather than using physical observations





Minimum Discarded Fill Element Ordering

- Greedy algorithm for element ordering [Persson/Peraire '08]:
  At step j, if j' is chosen next, we would discard the fill

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

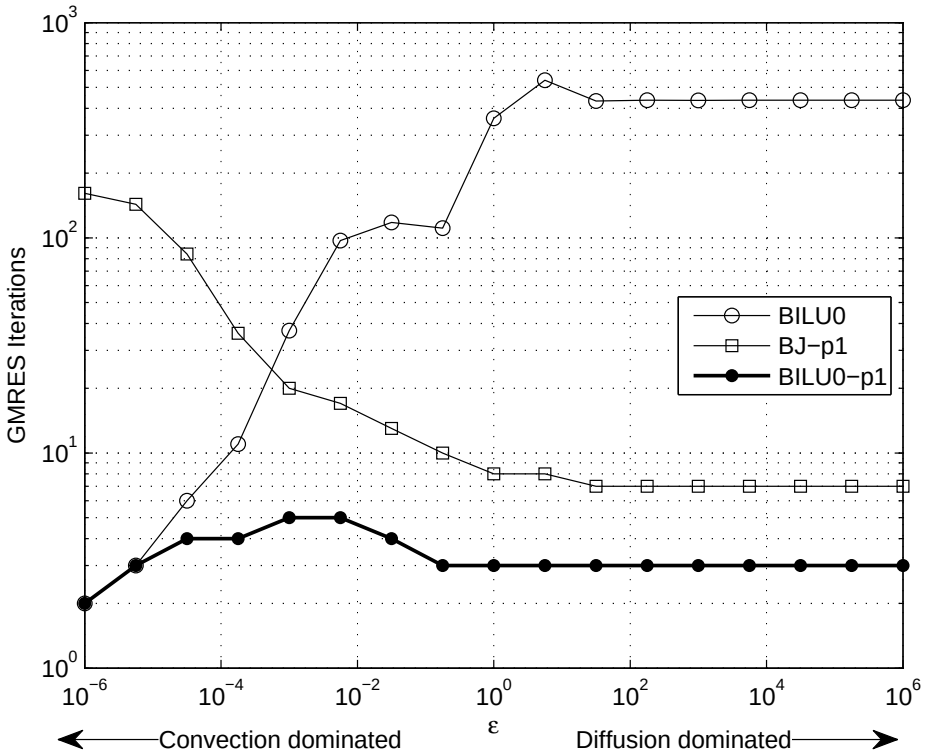
- Choose the j' that minimizes the norm of the discarded fill

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

- Some simplifications, min-heap data structure $\Longrightarrow \mathcal{O}(n \log n)$ cost
- Increased locality: Consider only neighbors for j'

Effect of Ordering on ILU

- MDF ordering makes block-ILU0 with coarse grid correction almost perfect for convection-diffusion
- Good element ordering critical for Navier-Stokes as well:



Problem	Element Ordering		
	Random	RCM	MDF
Inviscid	51	27	12
Laminar, Re=1,000	200	135	12
Laminar, Re=20,000	197	139	27
RANS, Re=10^6	98	99	18

Convergence – Model Navier-Stokes Problem

$\times =$ No convergence after 1,000 iterations, from [Persson/Peraire '08]

Problem	Parameters		Preconditioner/Iterations					
			Block Jacobi		Block G-S		Block ILU0	
	Δt	M	BJ	BJ-p1	BGS	BGS-p1	BILU0	BILU0-p1
Inviscid	10^{-3}	0.2	24	19	14	11	5	4
	10^{-1}	0.2	187	85	73	49	12	6
	∞	0.2	840	142	456	72	40	9
	10^{-3}	0.01	200	112	111	67	15	6
	10^{-1}	0.01	$\times$	$\times$	$\times$	532	94	10
	∞	0.01	$\times$	$\times$	$\times$	$\times$	374	16
Laminar Re=1,000	10^{-3}	0.2	50	34	25	19	4	4
	10^{-1}	0.2	$\times$	597	477	225	11	5
	∞	0.2	$\times$	$\times$	$\times$	$\times$	37	7
	10^{-3}	0.01	98	66	51	33	8	5
	10^{-1}	0.01	$\times$	619	$\times$	207	27	9
	∞	0.01	$\times$	$\times$	$\times$	748	135	12

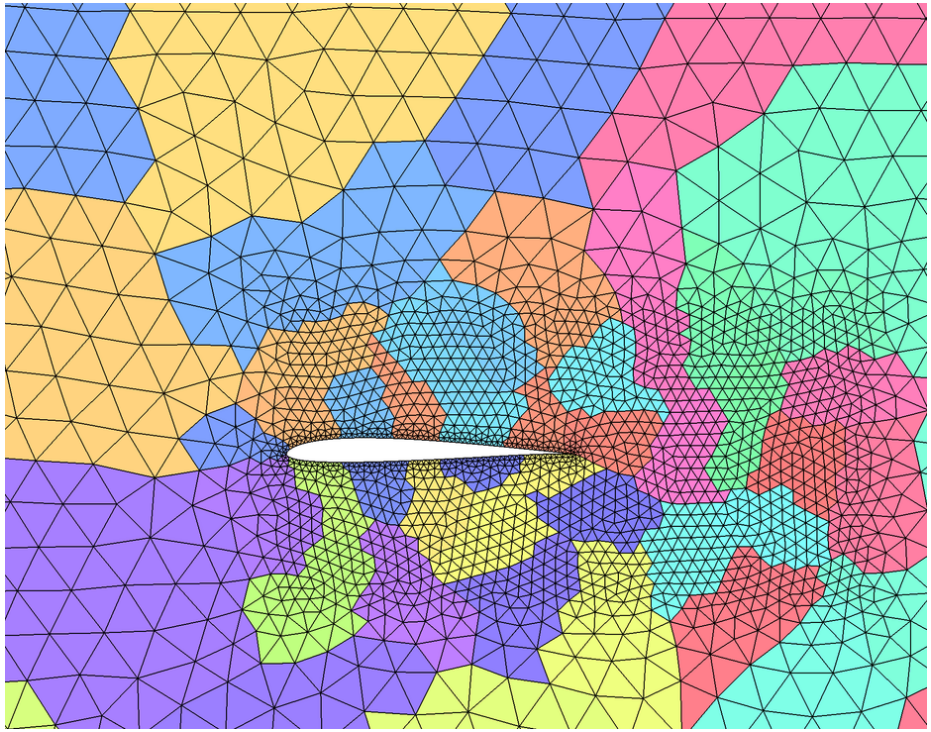
Convergence – Model Navier-Stokes Problem

$\times =$ No convergence after 1,000 iterations, from [Persson/Peraire '08]

Problem	Parameters		Preconditioner/Iterations					
			Block Jacobi		Block G-S		Block ILU0	
	Δt	M	BJ	BJ-p1	BGS	BGS-p1	BILU0	BILU0-p1
Laminar	10^{-3}	0.2	26	19	14	11	4	4
Re=20,000	10^{-1}	0.2	456	220	219	113	16	8
	∞	0.2	$\times$	$\times$	$\times$	$\times$	236	20
	10^{-3}	0.01	160	90	61	38	12	6
	10^{-1}	0.01	$\times$	$\times$	$\times$	735	80	16
	∞	0.01	$\times$	$\times$	$\times$	$\times$	$\times$	35
RANS	10^{-3}	0.2	76	56	33	28	8	7
Re=10^6	10^{-1}	0.2	$\times$	$\times$	$\times$	$\times$	35	25
	∞	0.2	$\times$	$\times$	$\times$	$\times$	70	18
	10^{-3}	0.01	411	231	174	110	14	9
	10^{-1}	0.01	$\times$	$\times$	$\times$	$\times$	46	16
	∞	0.01	$\times$	$\times$	$\times$	$\times$	132	28

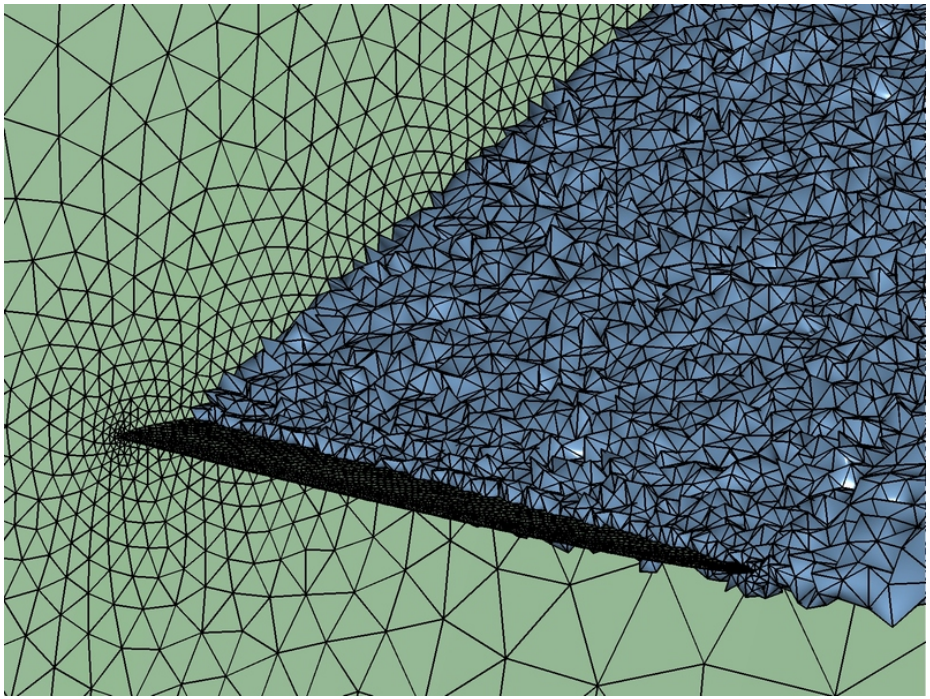
ILU Parallelization – Domain Decomposition

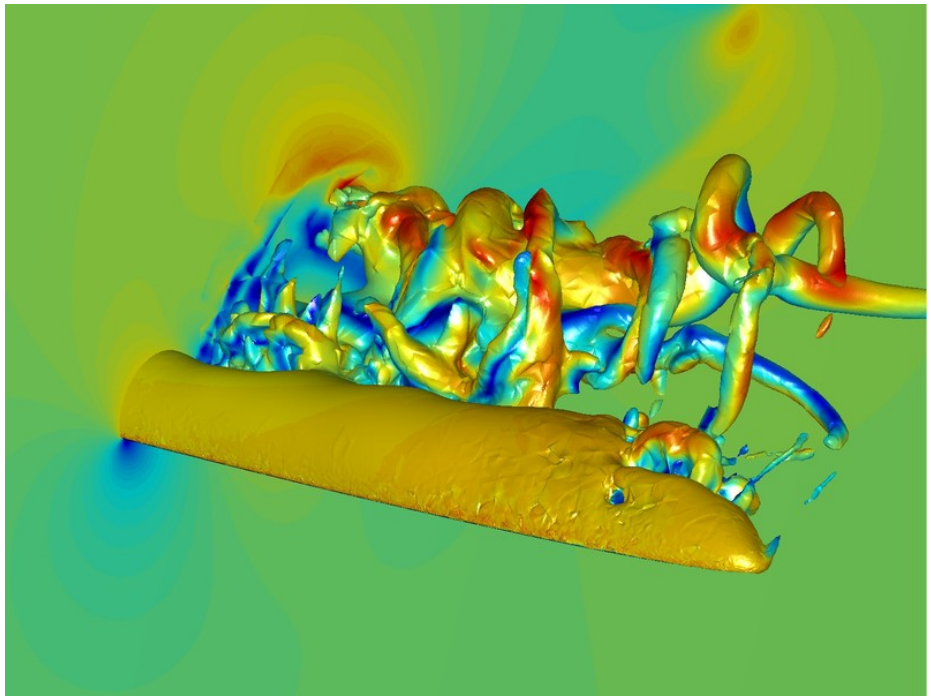
- In parallel, use partition-wise ILUs with MDF ordering
- Partition using the weights $C_{ij} = \|\boldsymbol{A}_{ii}^{-1}\boldsymbol{A}_{ij}\|_F$
- Essentially a "non-overlapping Schwartz preconditioner with incomplete solutions"
- Approaches Jacobi as # partitions $\rightarrow$ # elements
- Good option for many problems – GMRES iterations cheap compared to matrix creation



Elliptic Wing Problem

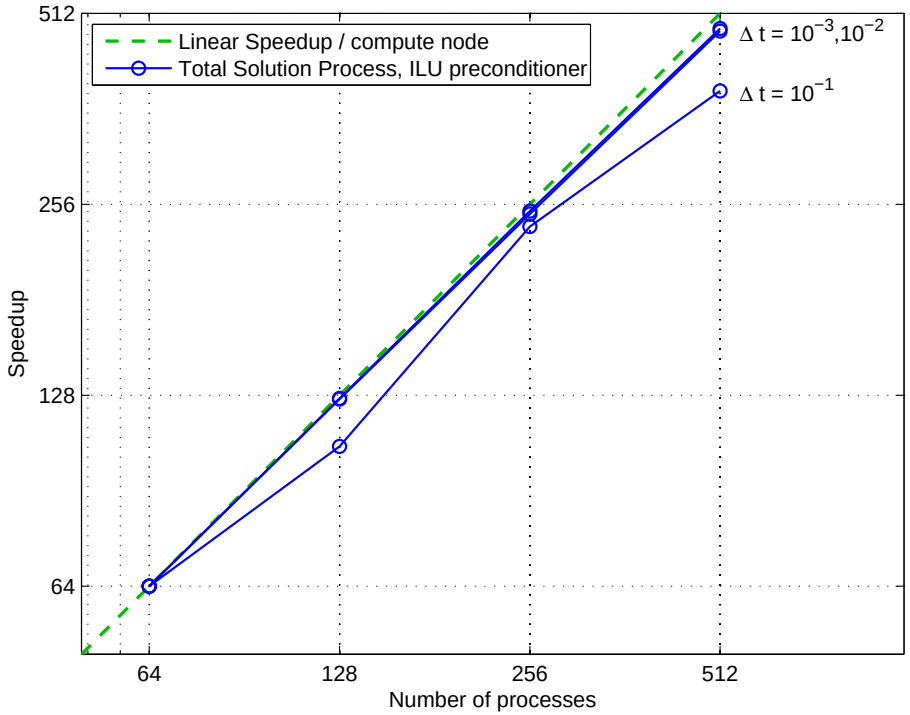
- Elliptic wing, $\mathrm{Re} = 2000$, $M = 0.3$, $\mathrm{AoA} = 30^\circ$
- 190,000 tetrahedral elements, $p = 3$, about 19 million DOFs
- Time integration by 2-stage, 3rd order accurate DIRK scheme
- Jacobian storage 36GB, no comparison with serial possible

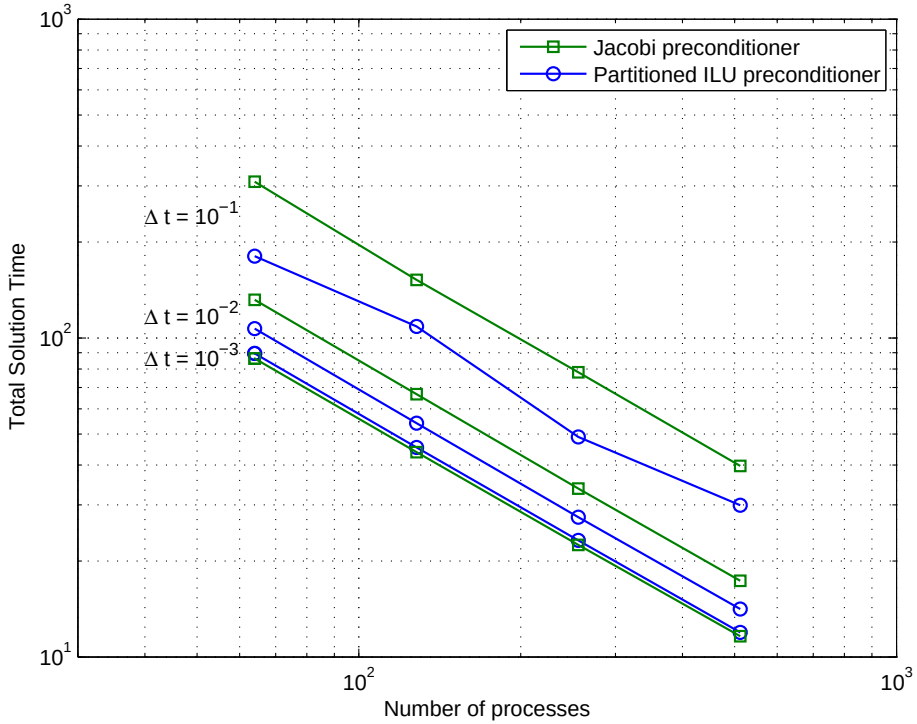




The Elliptic Wing Problem – Timings and Speedup

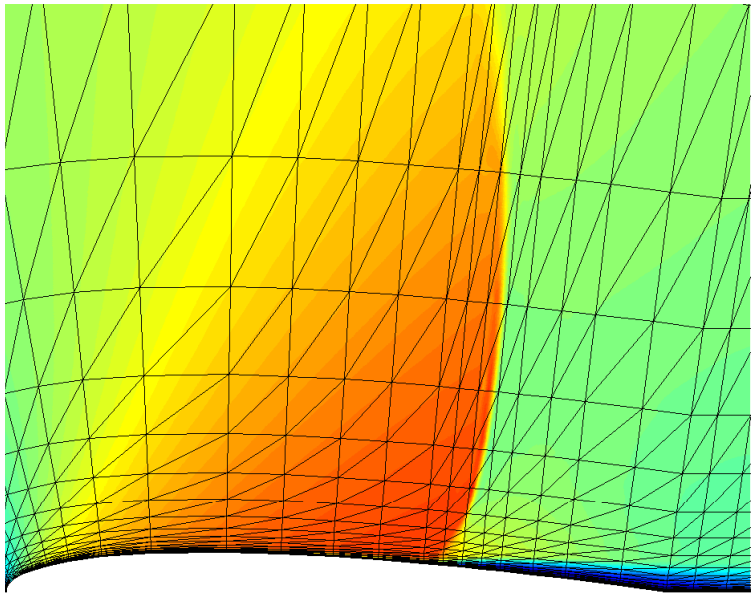
- Almost perfect speedup per compute node with 512 processes
- Solution time dominated by assembly for small Δt



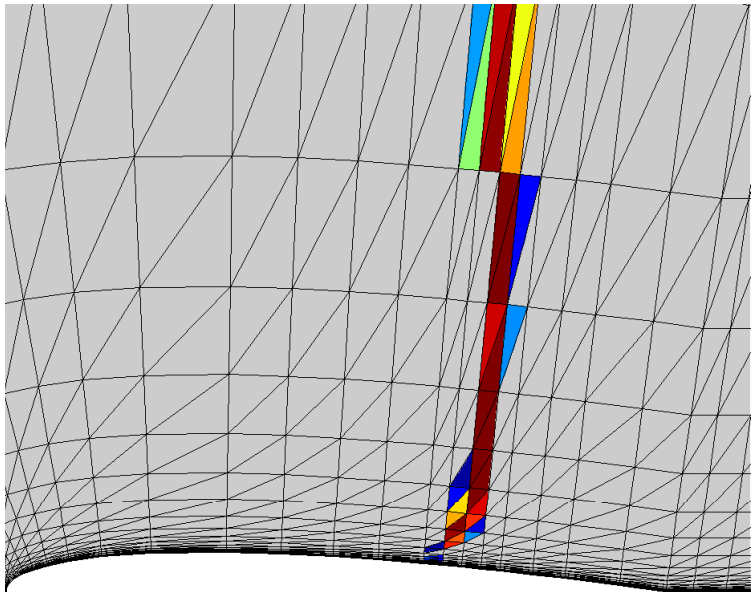


Artificial Viscosity for Underresolved Features

- Cannot resolve all solution features (shocks, RANS, 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 and robust solution of Spalart-Alamaras RANS model



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:

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

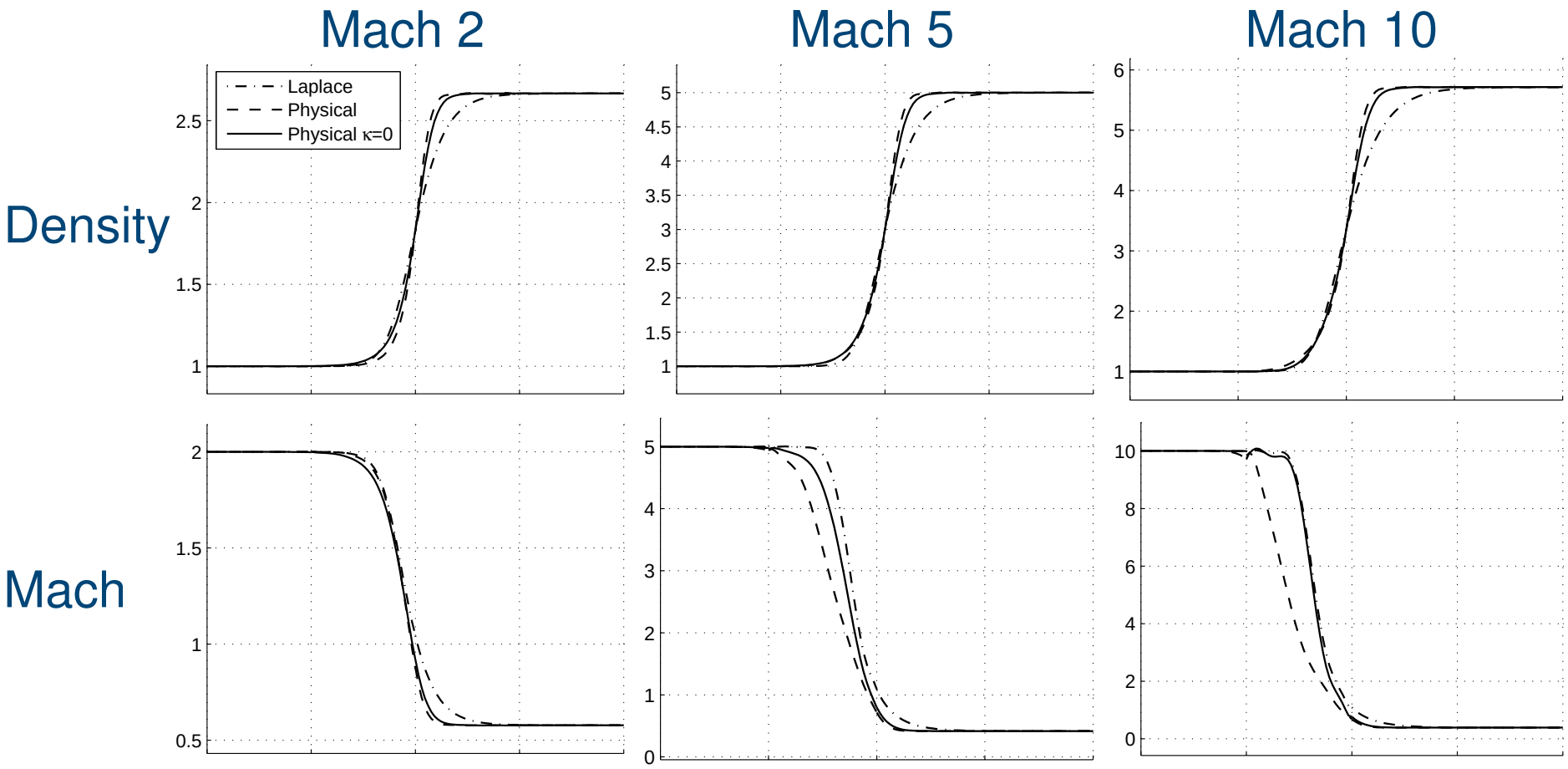
- Determine elemental piecewise constant ε_e

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

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

Euler Equations – Artificial Viscosity Models

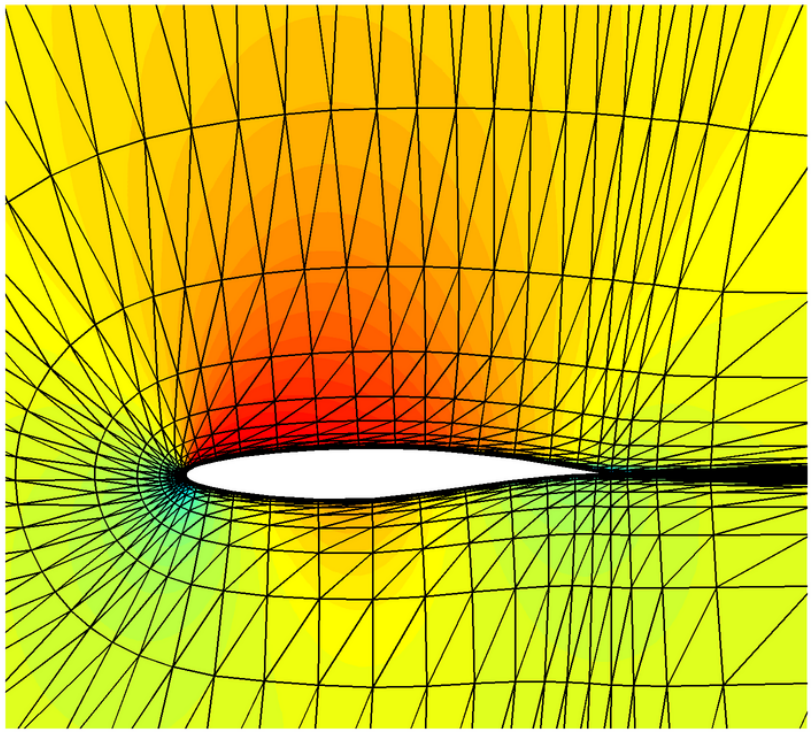
- Laplacian: $\frac{\partial \boldsymbol{u}}{\partial t} + \nabla \cdot \mathcal{F}(\boldsymbol{u}) = \nabla \cdot (\epsilon \nabla \boldsymbol{u})$
- Physical: $\frac{\partial \boldsymbol{u}}{\partial t} + \nabla \cdot \mathcal{F}(\boldsymbol{u}) = \nabla \cdot \mathcal{F}_v(\boldsymbol{u}; \mathrm{Re},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

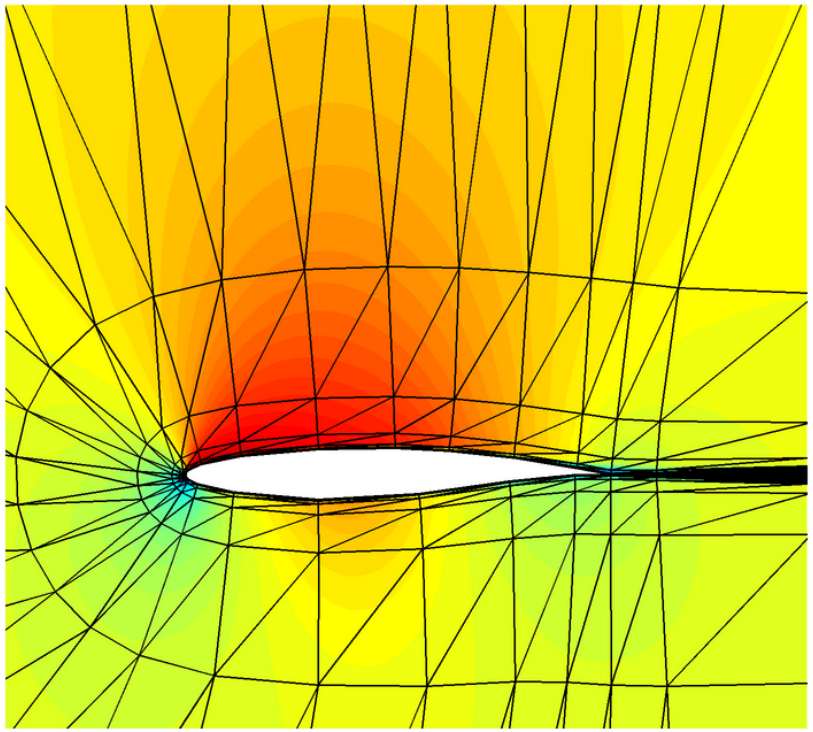


Example: RAE2822

- Turbulent RANS flow ($M = 0.675, \alpha = 2.31^o, \mathrm{Re} = 6.5 \cdot 10^6$)
- p-converged solution, fixed resolution h/p

$p = 2$ (constant h/p) $p = 4$





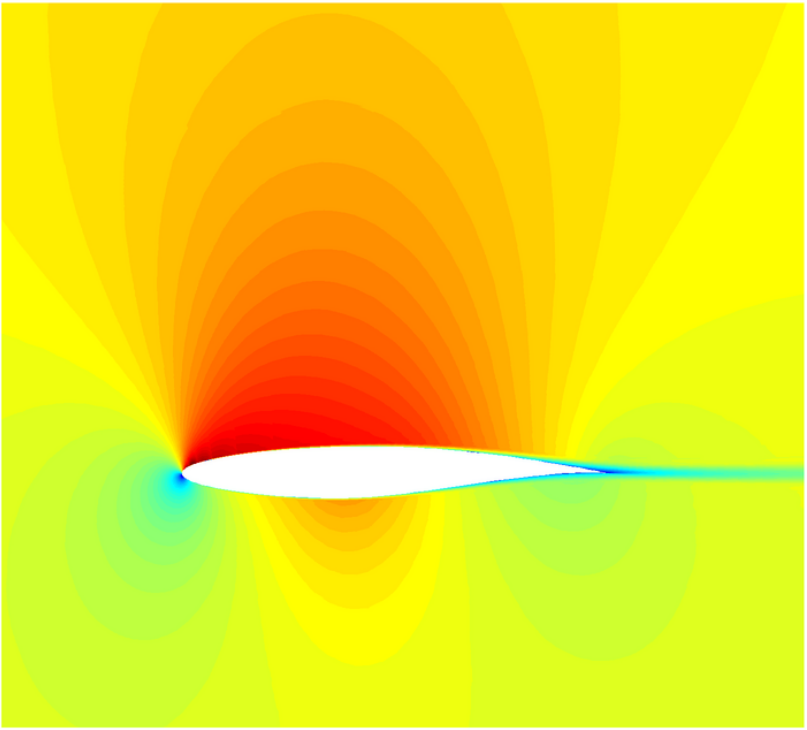
$C_L = 0.6144 \quad C_D = 0.0104$

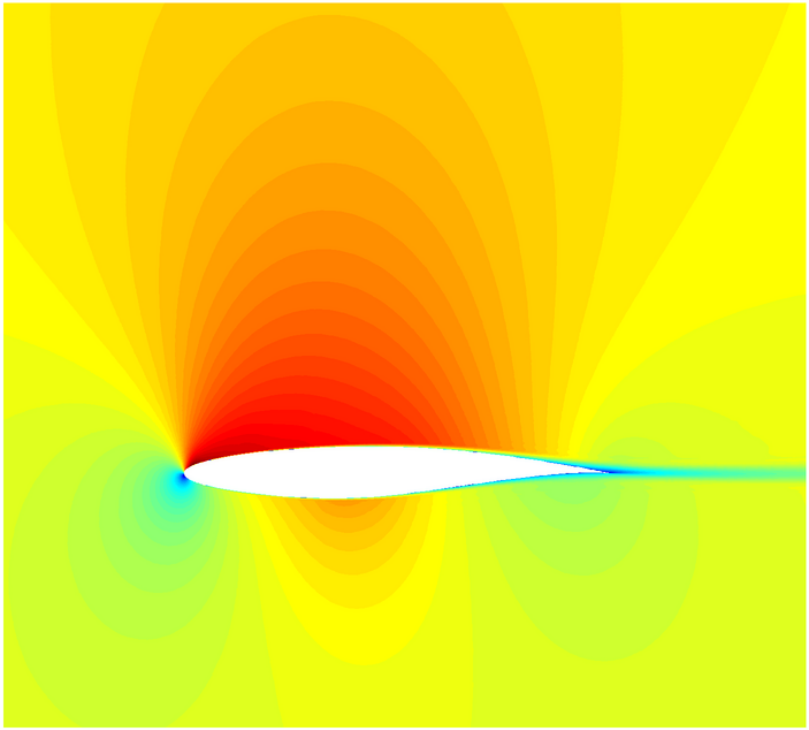
$C_L = 0.6131 \quad C_D = 0.0103$

Example: RAE2822

- Turbulent RANS flow ($M = 0.675, \alpha = 2.31^o, \mathrm{Re} = 6.5 \cdot 10^6$)
- p-converged solution, fixed resolution h/p

$p = 2$ (constant h/p) $p = 4$





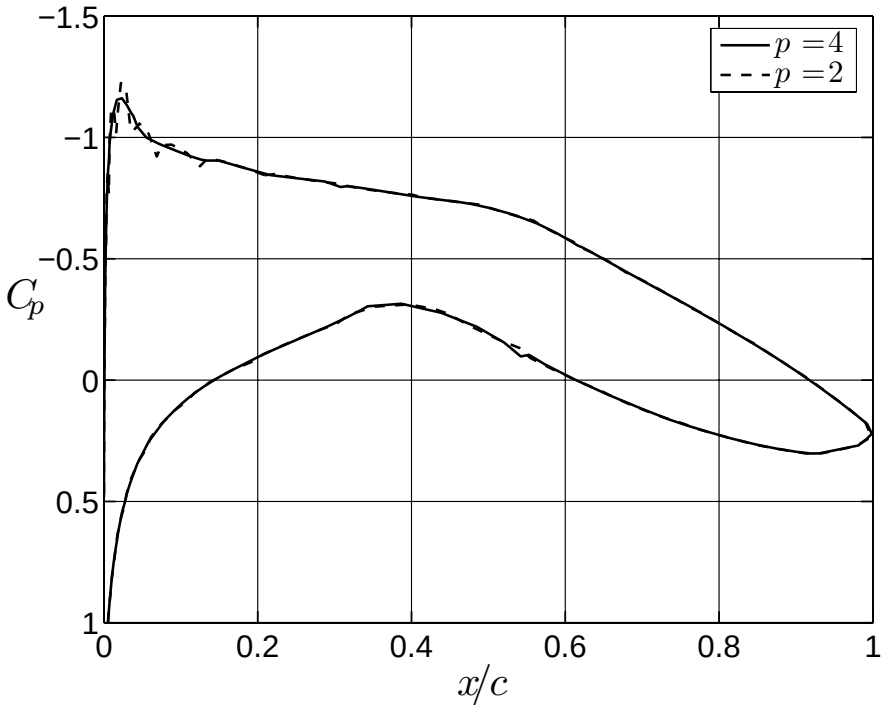
$C_L = 0.6144 \quad C_D = 0.0104$

$C_L = 0.6131 \quad C_D = 0.0103$

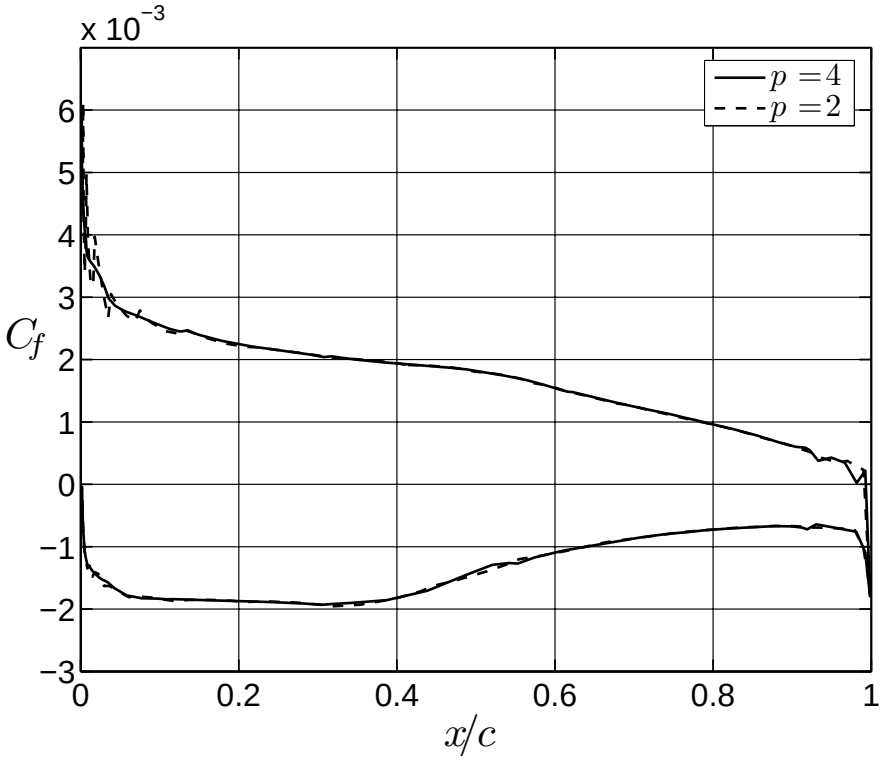
Example: RAE2822

- Highly accurate boundary forces even with coarse meshes

C_p



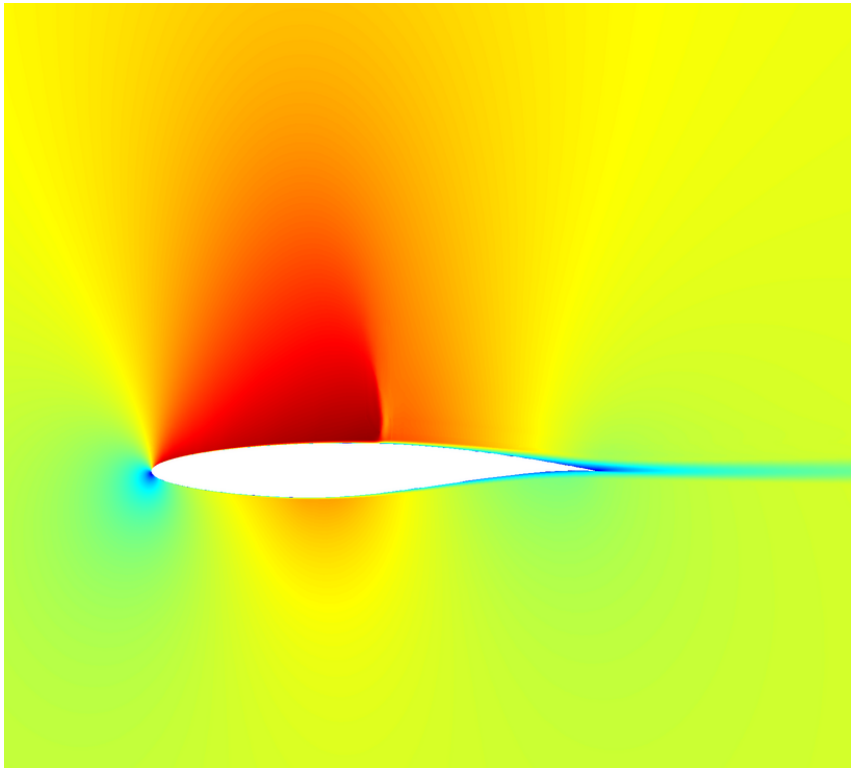
C_f

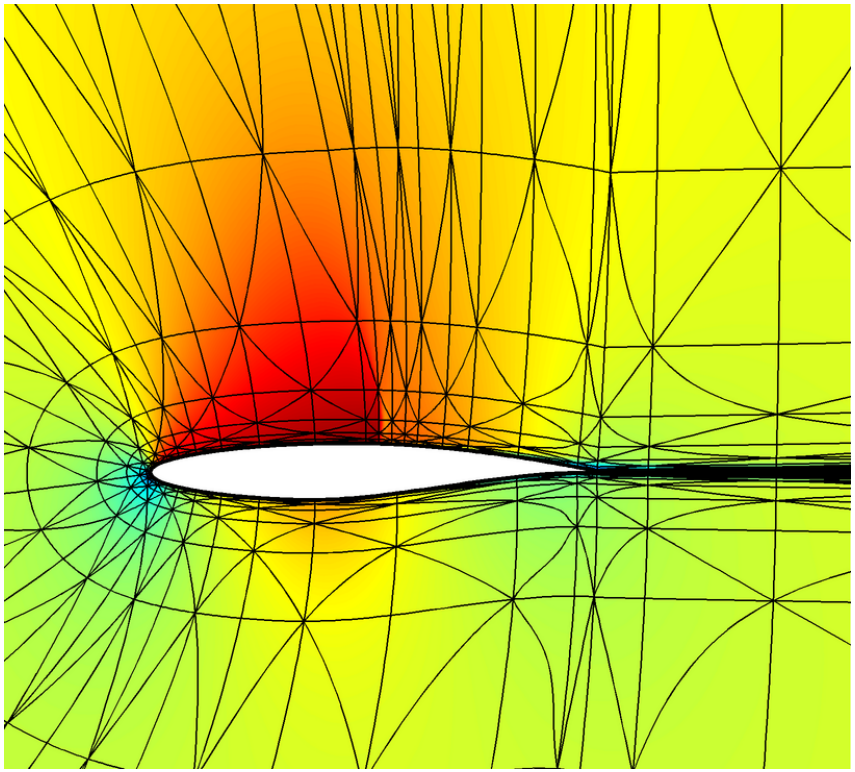


Example: RAE2822, Transonic

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

$p = 4$

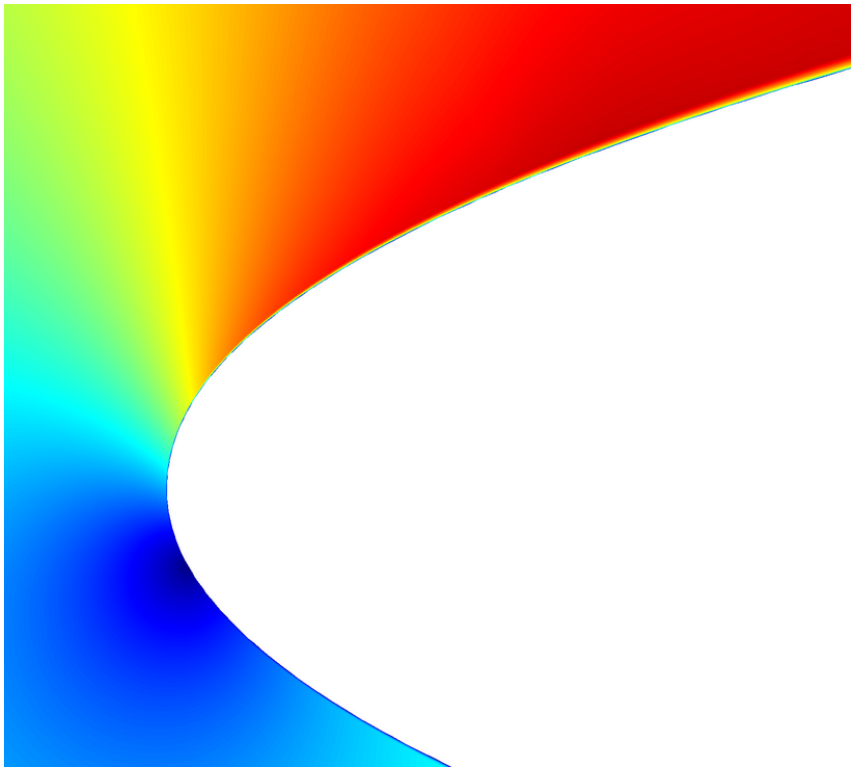


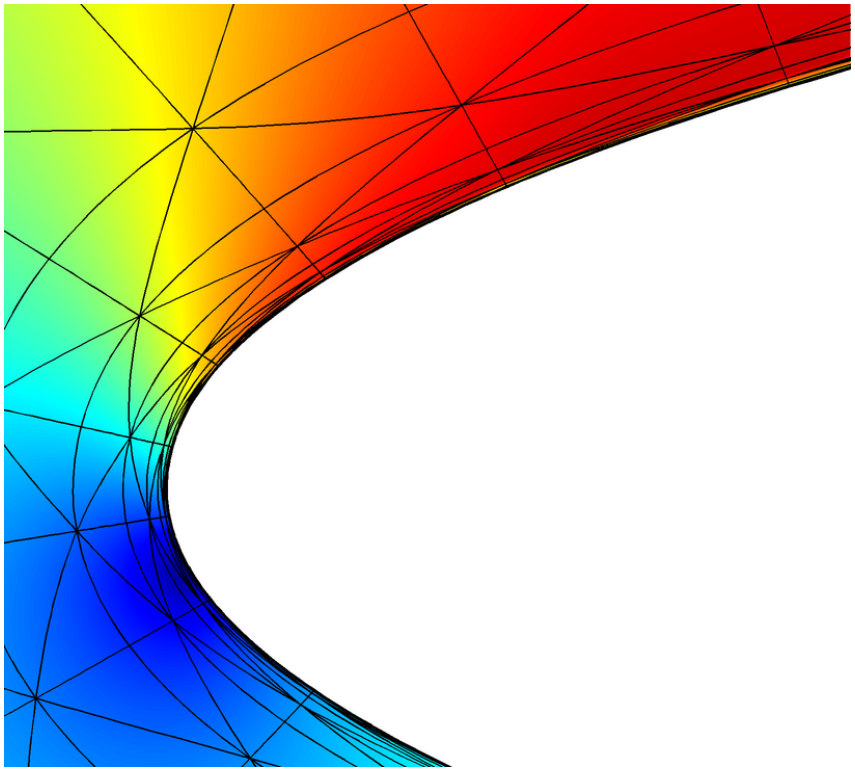


Example: RAE2822, Transonic

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

$p = 4$





Outline

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

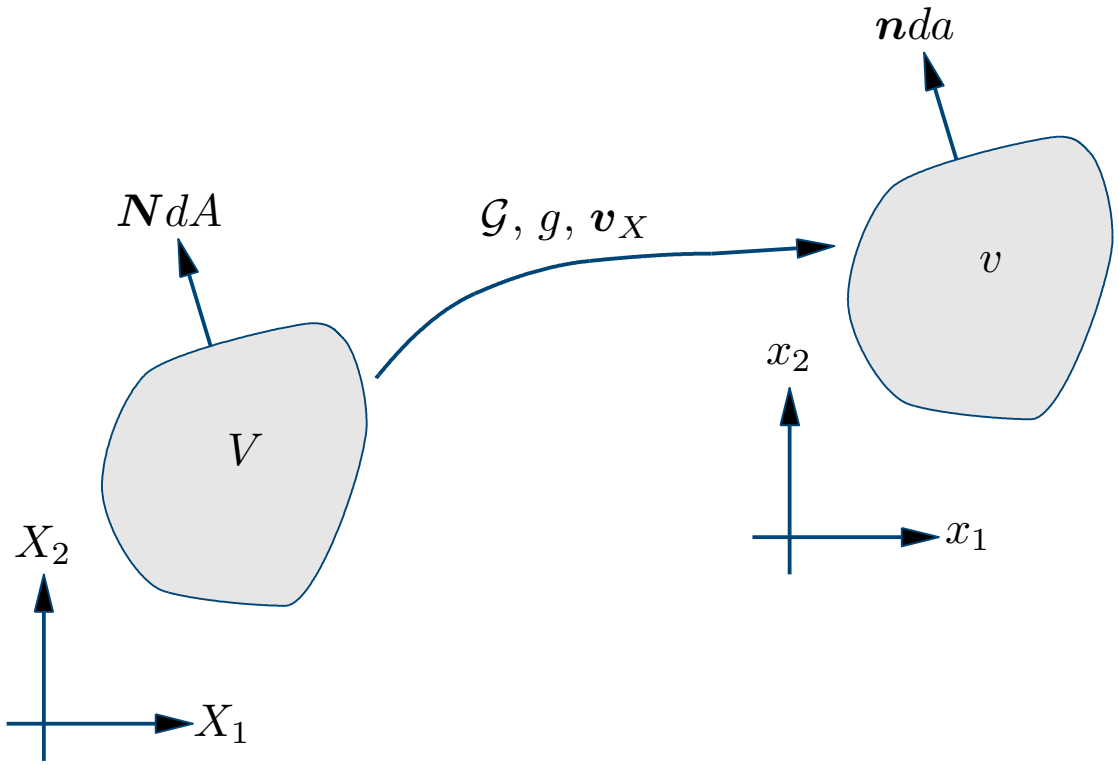
ALE Formulation

- 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

$$\boldsymbol{G} = \boldsymbol{\nabla}_X \boldsymbol{G}$$

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

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

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

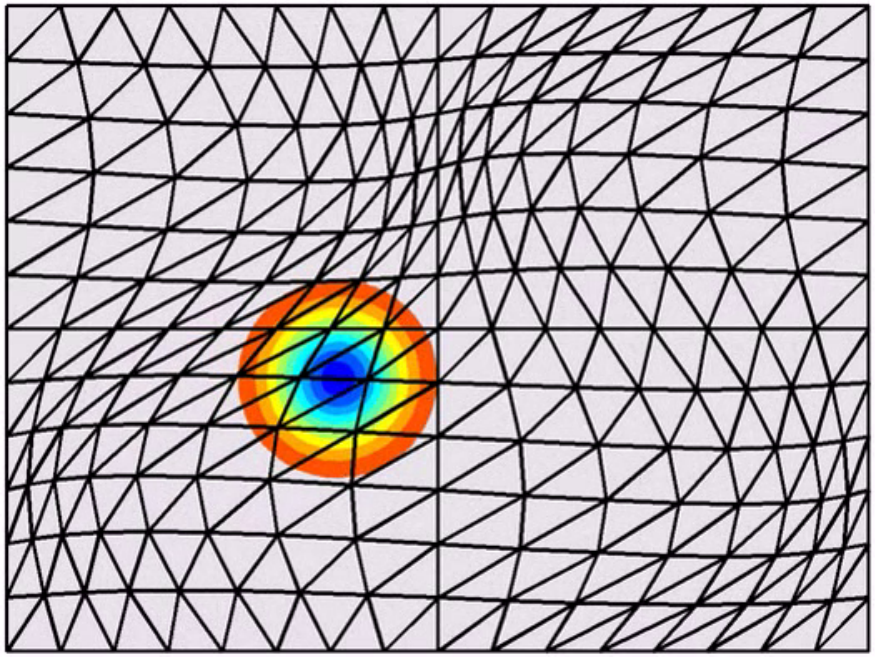
$$\left.\frac{\partial \bar{g}}{\partial t}\right|_X - \boldsymbol{\nabla}_X \cdot (g\boldsymbol{G}^{-1}\boldsymbol{v}_X) = 0$$

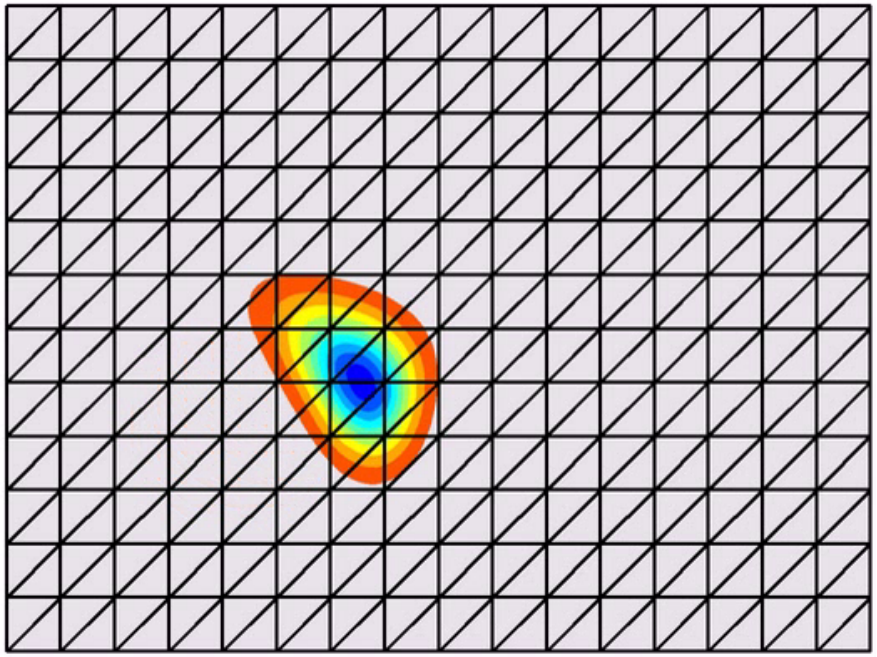
Example: Euler Vortex

- Propagate an Euler vortex on a variable domain with

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

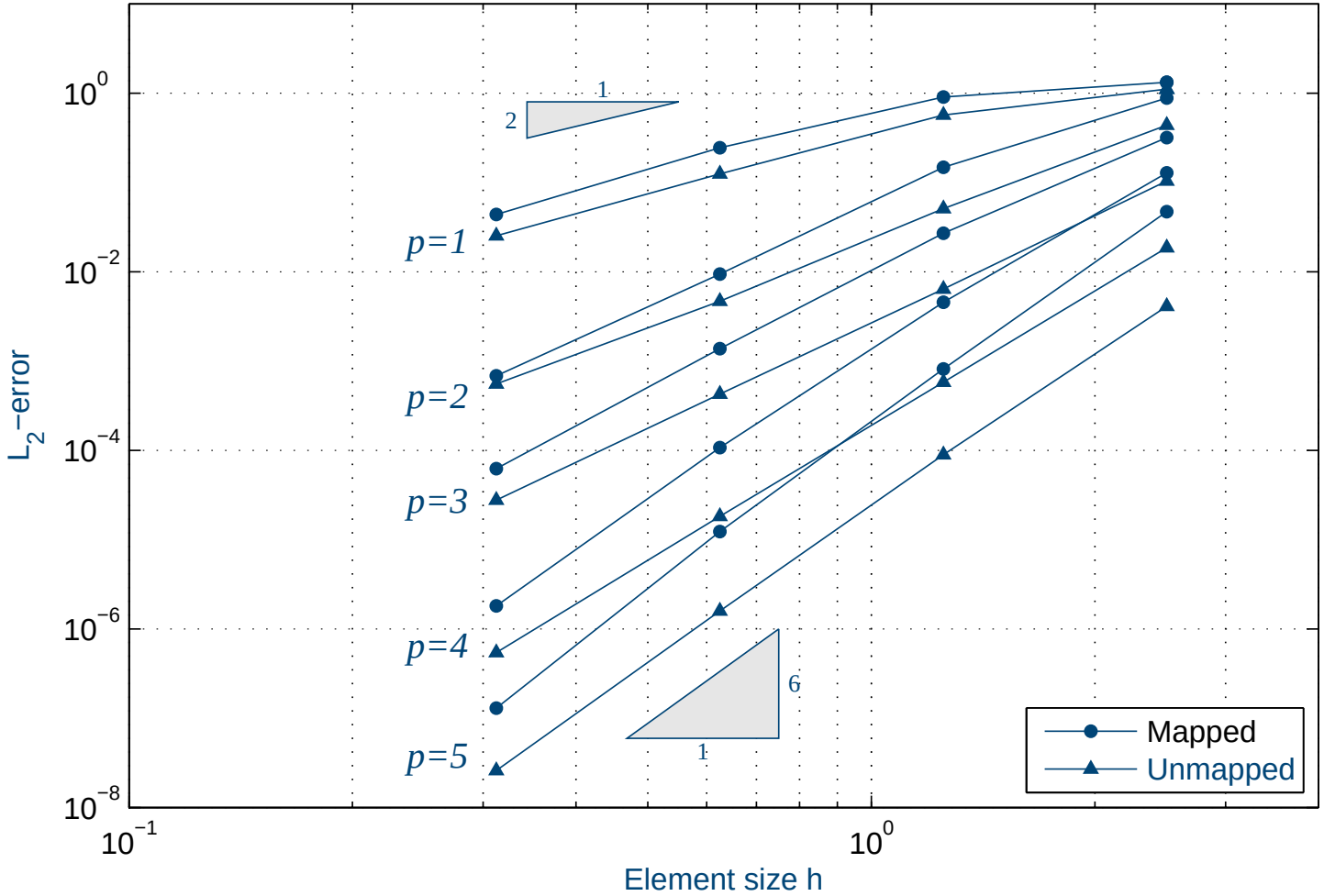
- Mapped scheme – Everything is computed on the reference mesh





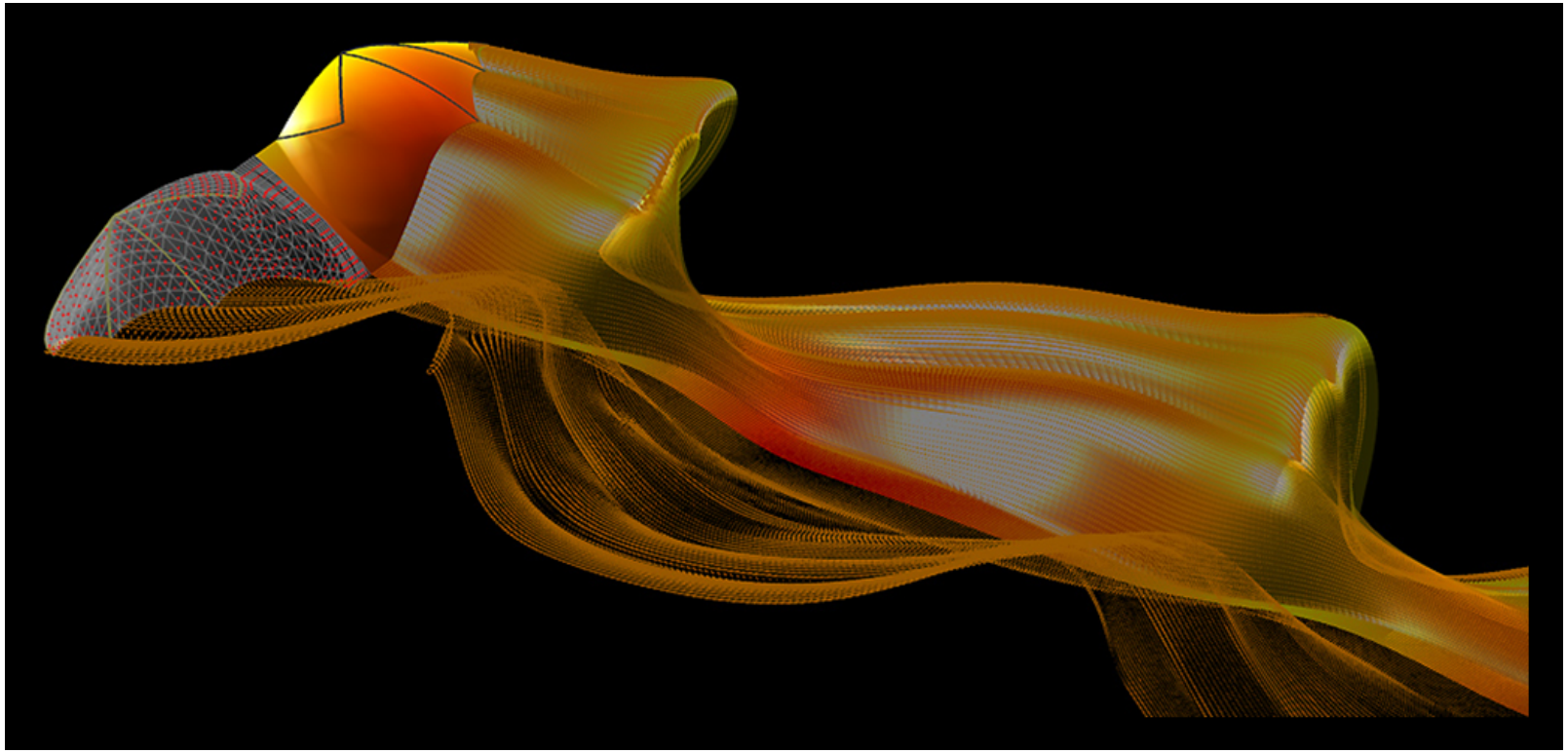
Example: Euler Vortex, Convergence

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



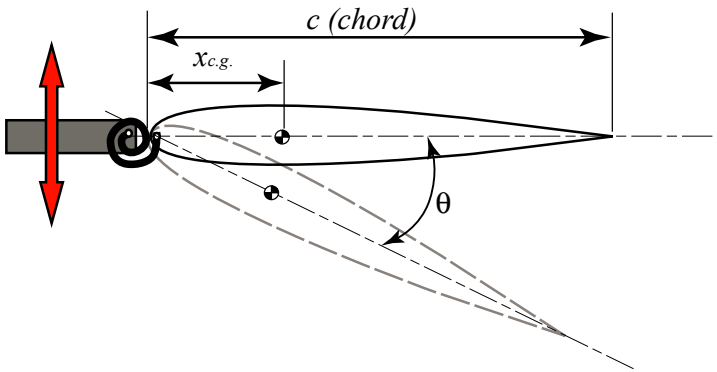
Biologically-Inspired Flapping Flight

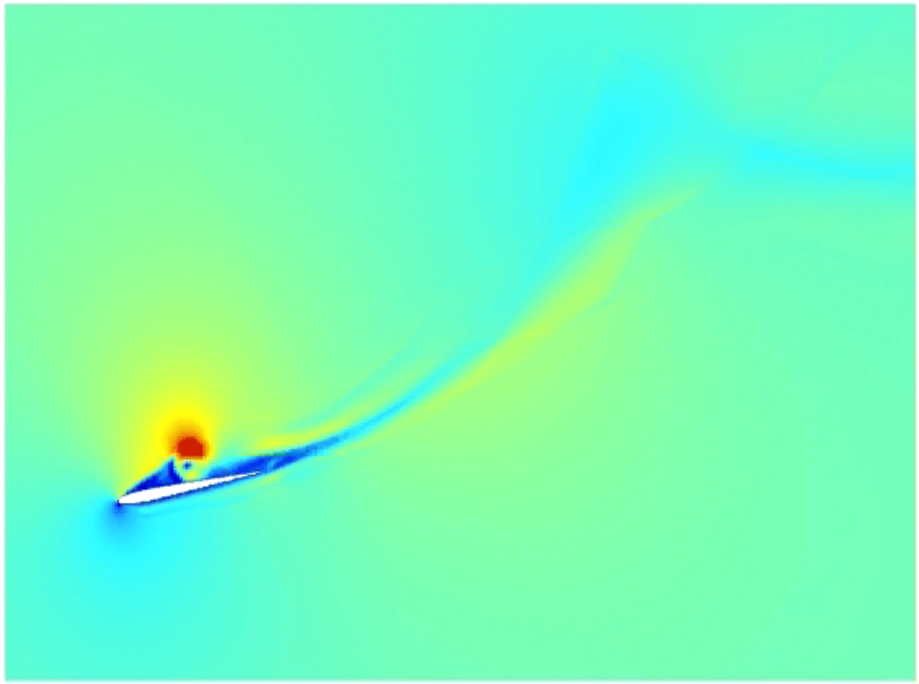
- Development of computational tools for studying flapping flight
- Challenging problems: Deforming domains, fluid-structure interaction, transitional flows, etc



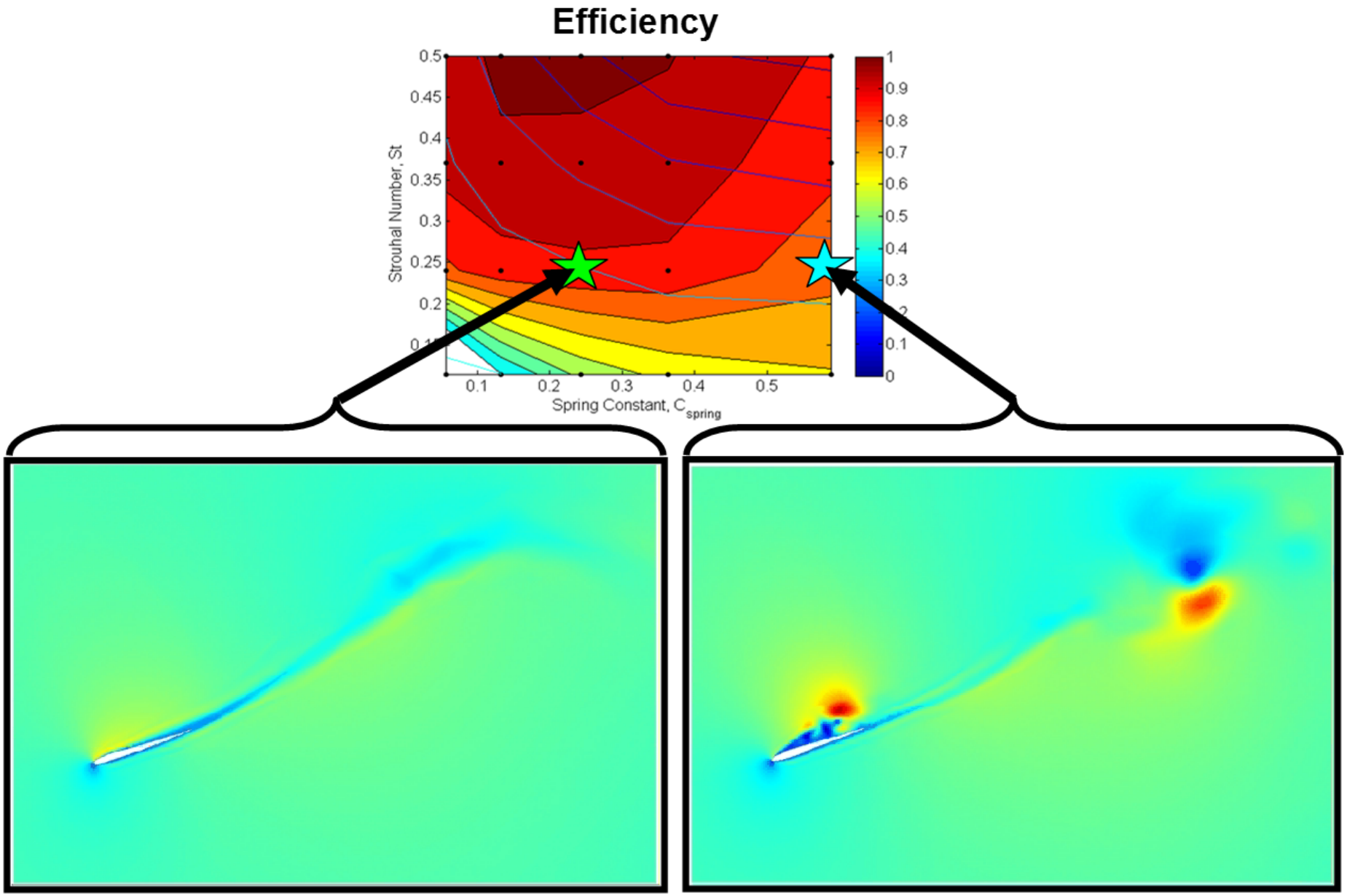
Example: Pitching Airfoil

- 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: $I\ddot{\theta} + C\theta - S\ddot{r}_z(t) + M_{\mathrm{aero}} = 0$
  - I moment of inertia, C spring stiffness, $S = mx_c$ static unbalance
  - M_{aero} moment from fluid



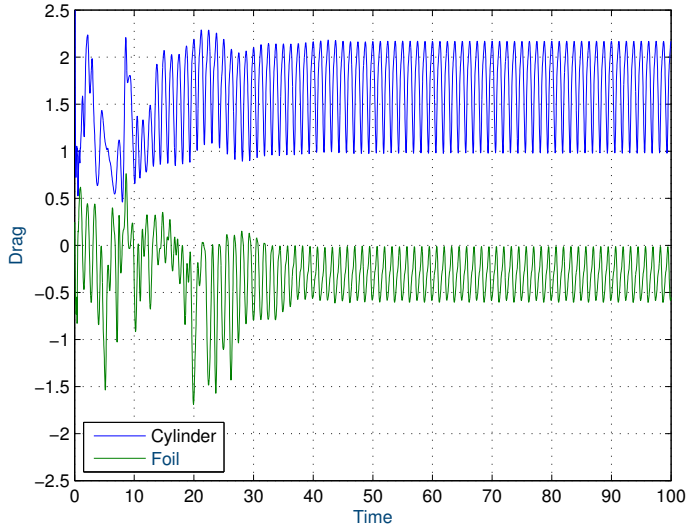


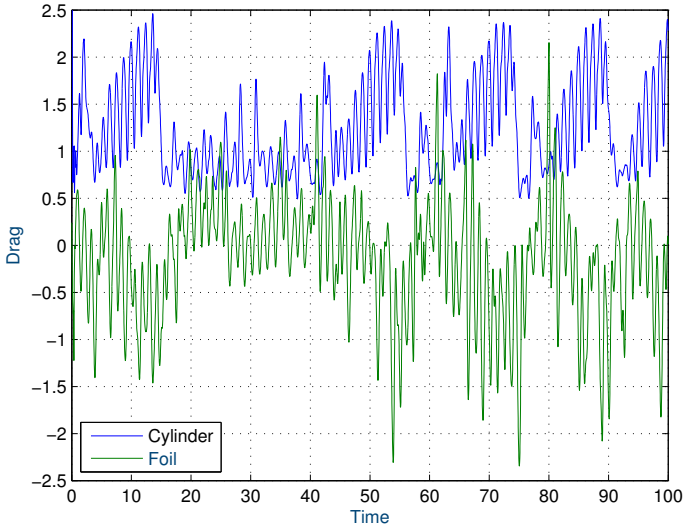
Example: Pitching Airfoil/Flapper Design

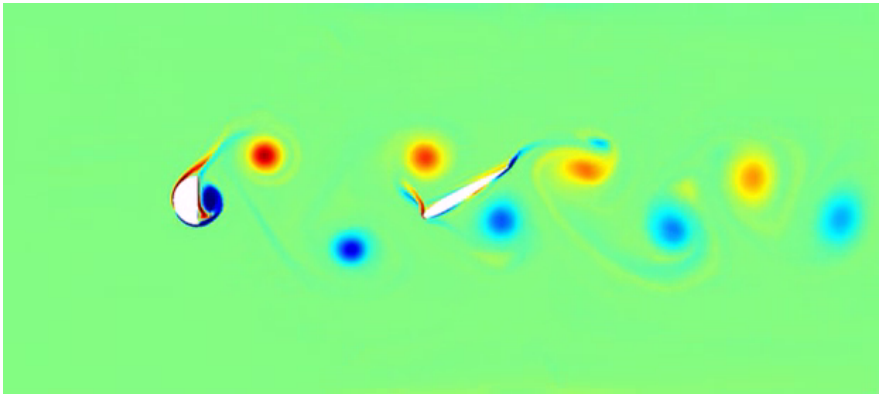


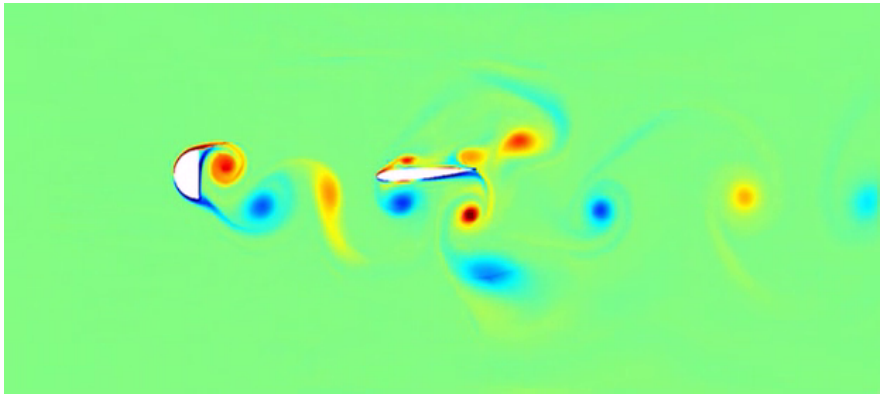
Example: Heaving and Pitching Foil in Wake

- NACA 0012 foil heaving and pitching in wake of D-section cylinder
- Both oscillate $y(t) = A\sin(2\pi ft)$, foil pitching $\theta = a\sin(2\pi ft + \pi/2)$
- Based on experimental study [Gopalkrishnan et al 94]



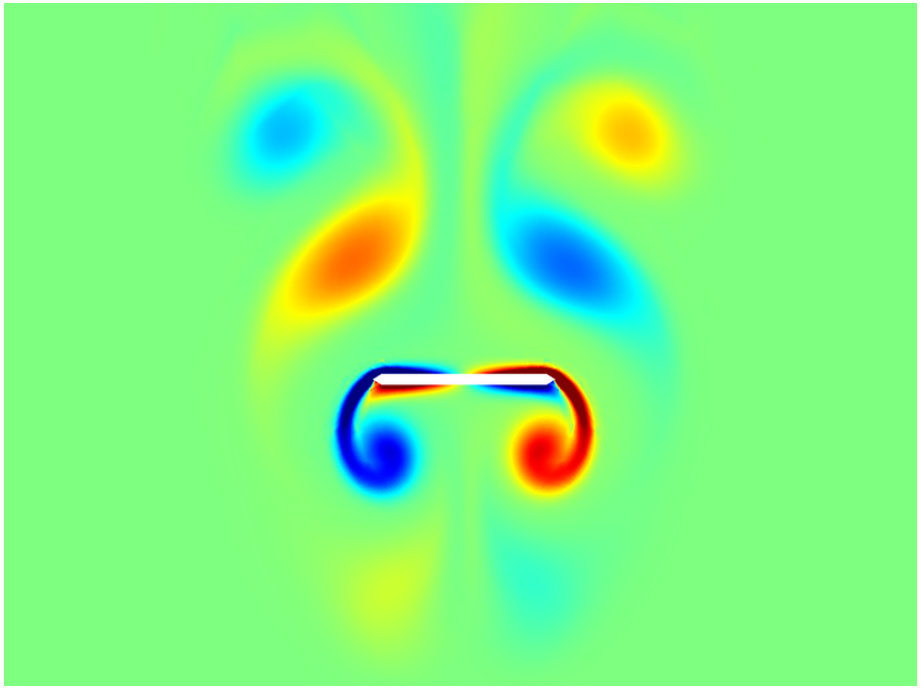


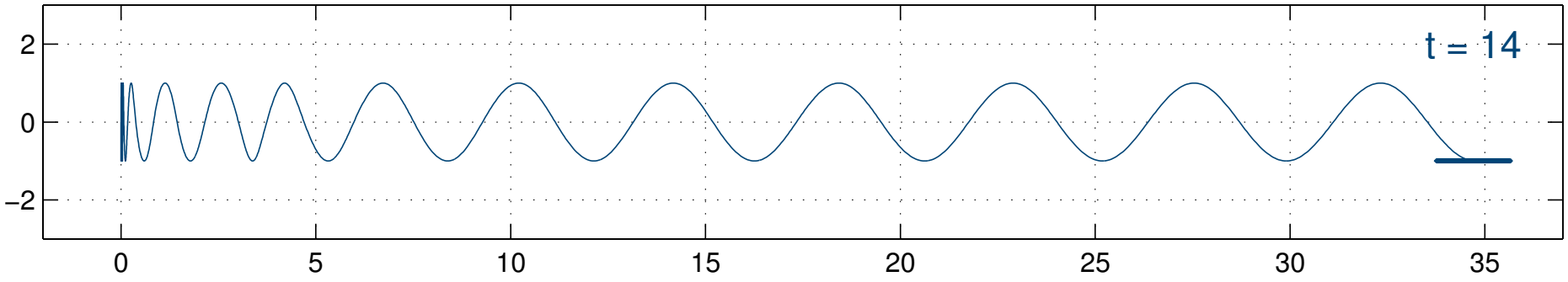




Example: Locomotion of Free Flapping Body

- Oscillating plate, unconstrained horizontally [Vandenberghe et al 04, Alben/Shelley 05]
- Instability breaks symmetry and forces plate into motion

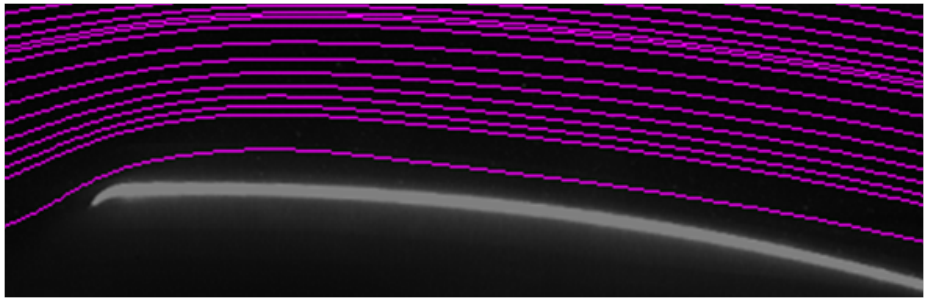




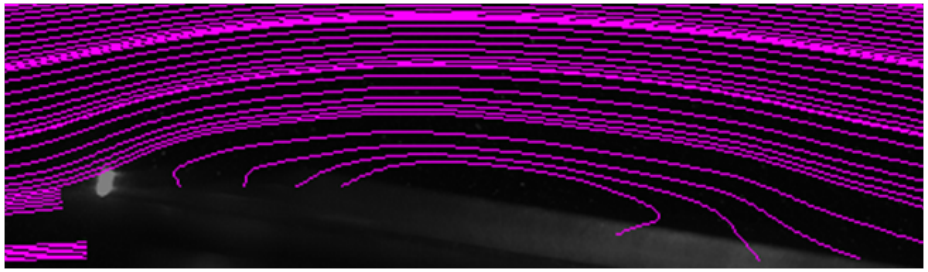
Example: Fluid-Structure Interaction

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

Experiment (A. Song, Brown U)

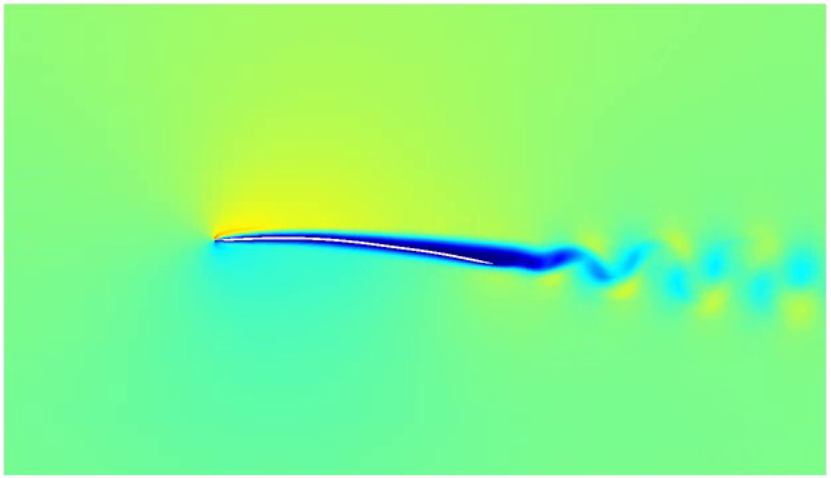


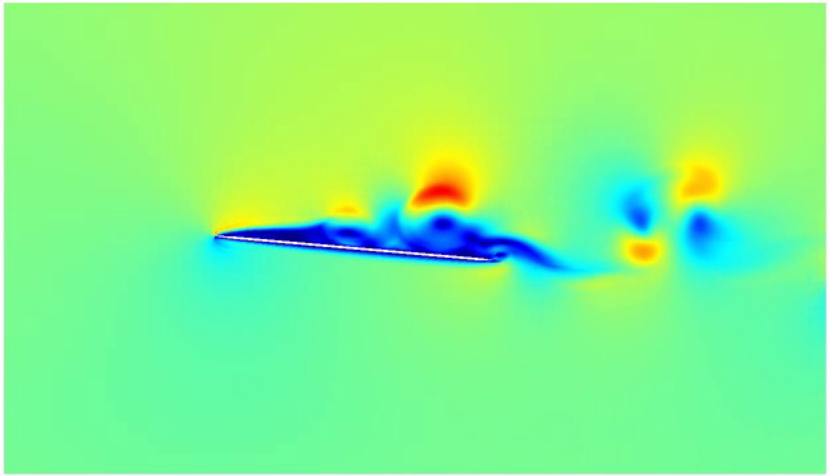
Compliant membrane



Rigid flat plate

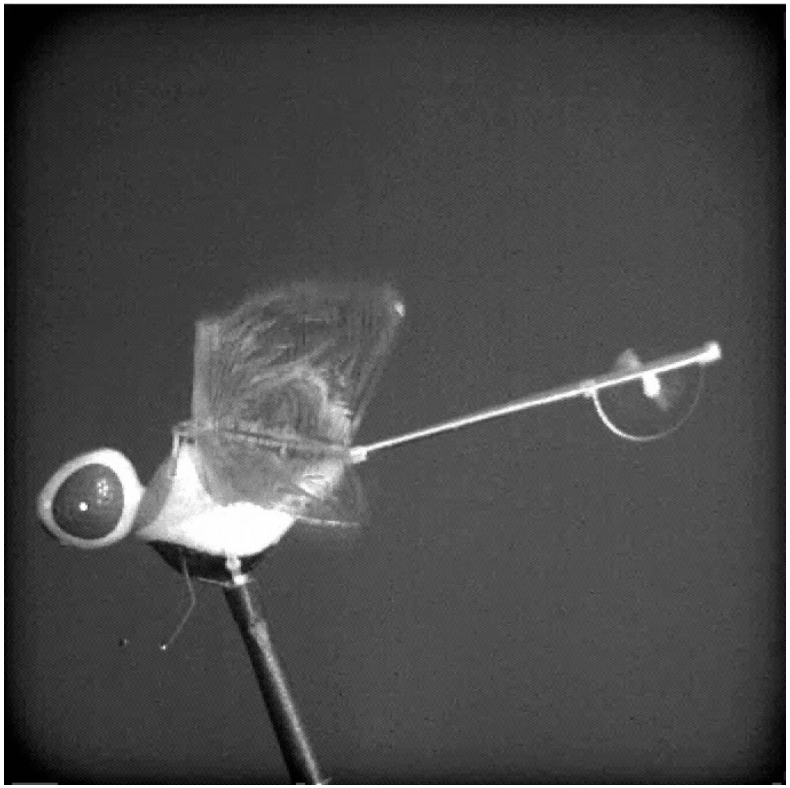
Fluid/membrane simulation



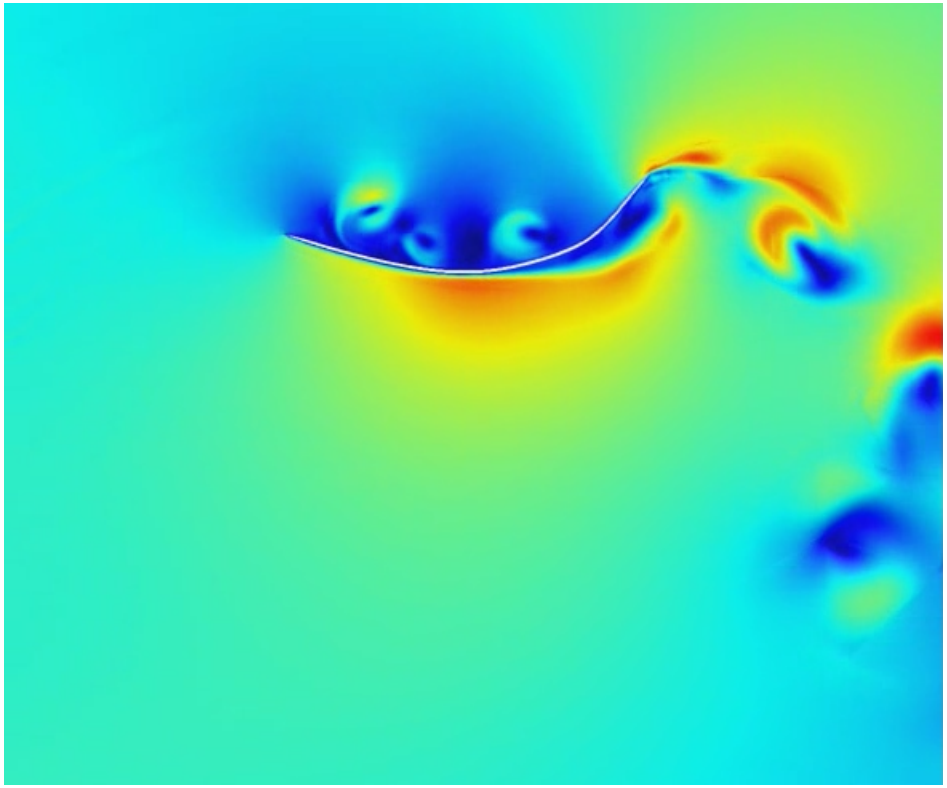


Example: Dragonfly, Compliant Wings

Experiment (A. Song, Brown U)



Fluid/membrane simulation



Outline

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

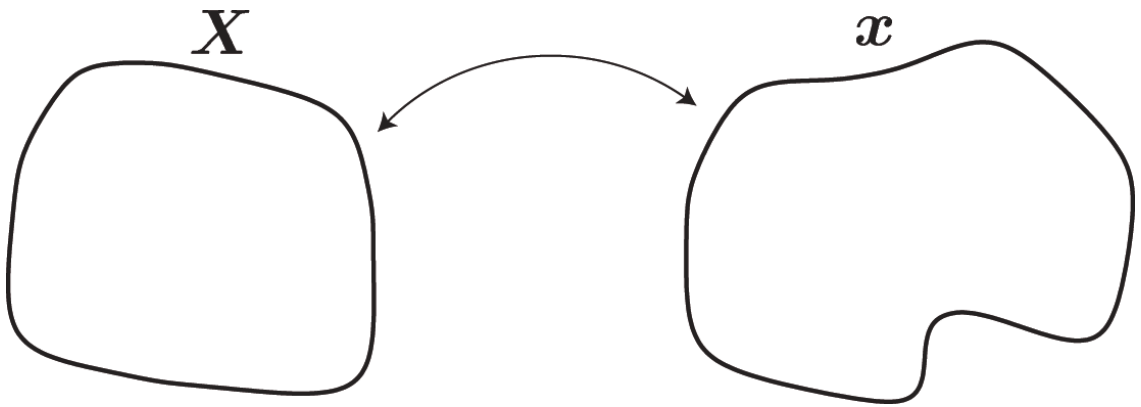
  $\Longrightarrow$ Thin structures (beams, shells) with highly stretched elements
- Stabilization for convection

  $\Longrightarrow$ 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:

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

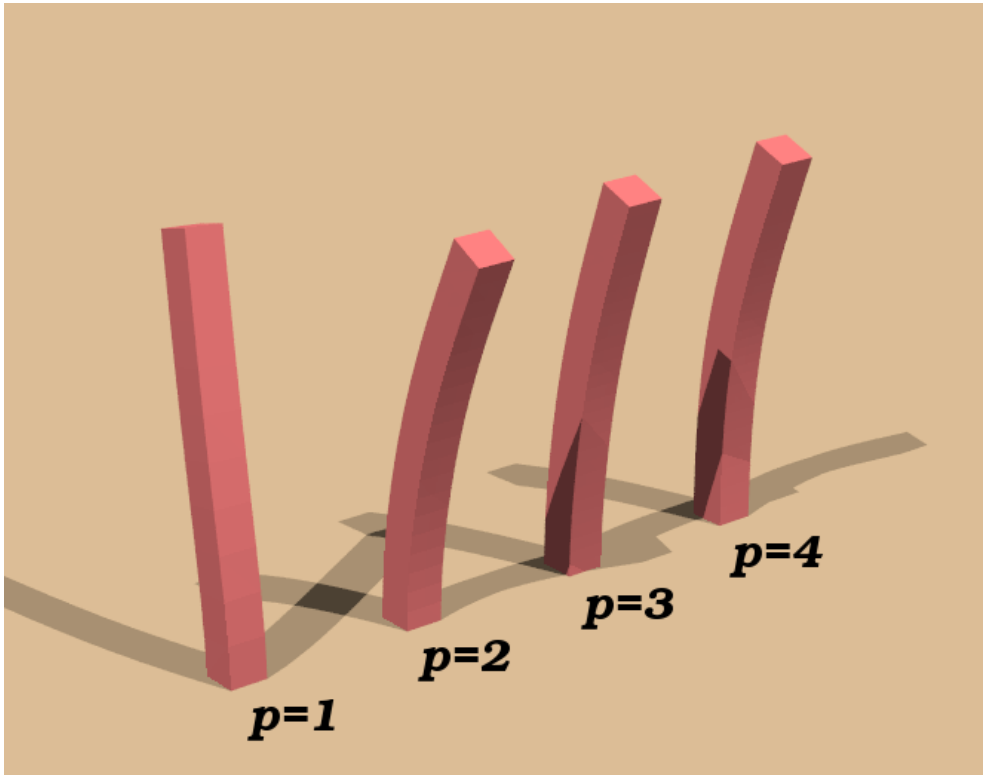
with

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

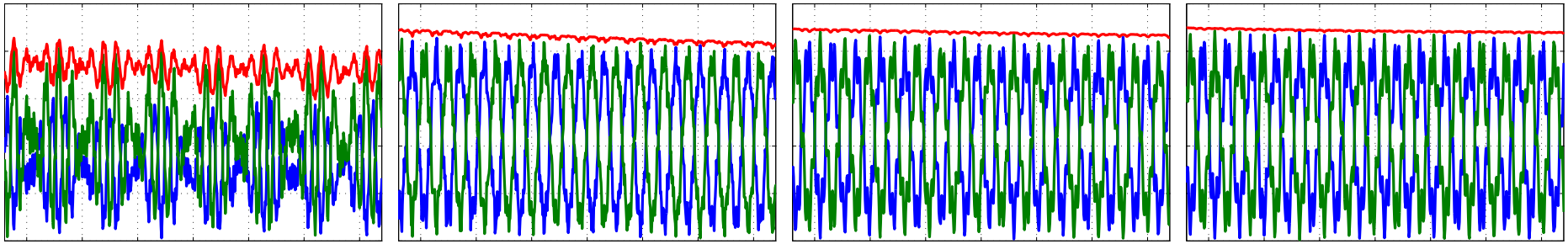
- Conservative formulation $\Longrightarrow$ borrow shock capturing from CFD, jump conditions physically meaningful
- However, computationally expensive, needs curl-free spaces for $\boldsymbol{F}$

Oscillating Beam, Energy Conservation

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



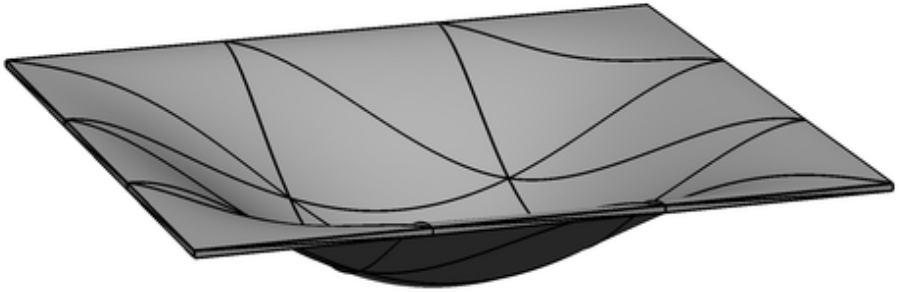
Kinetic, potential, and total energy

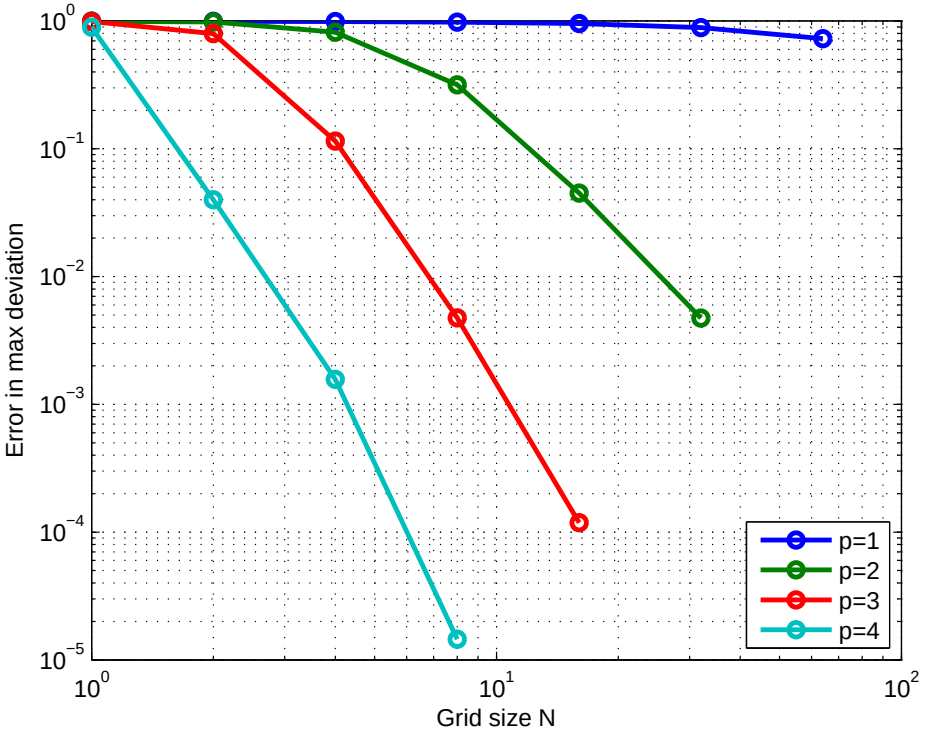


$p = 1$ $p = 2$ $p = 3$ $p = 4$

Volumetric Modeling of Thin Structures

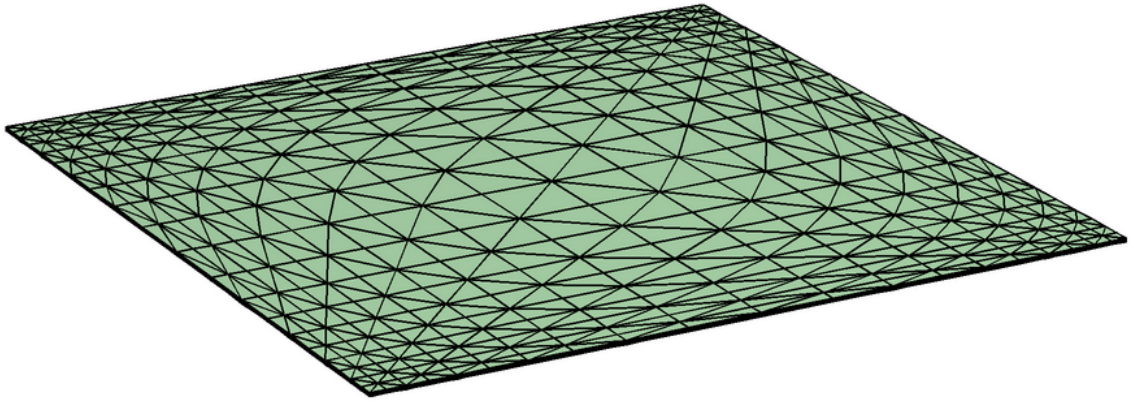
- The high order elements allow for high stretching
- Discontinuous basis functions highly insensitive to locking
- Model plate problem: Uniform loading, small deformations
- High order convergence rates for $p > 1$

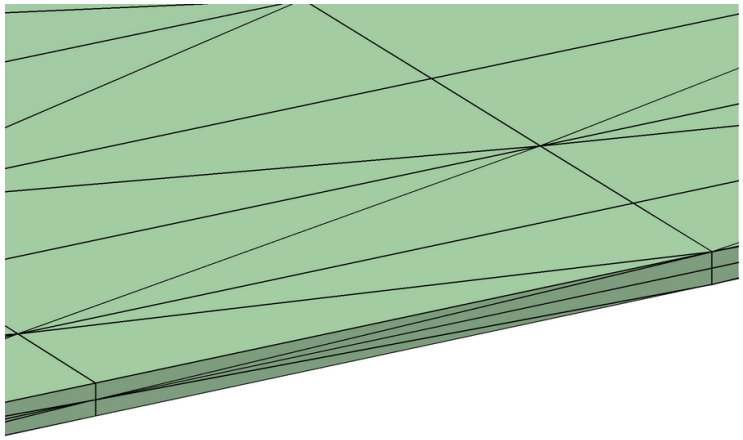




Example: Airbag Inflation

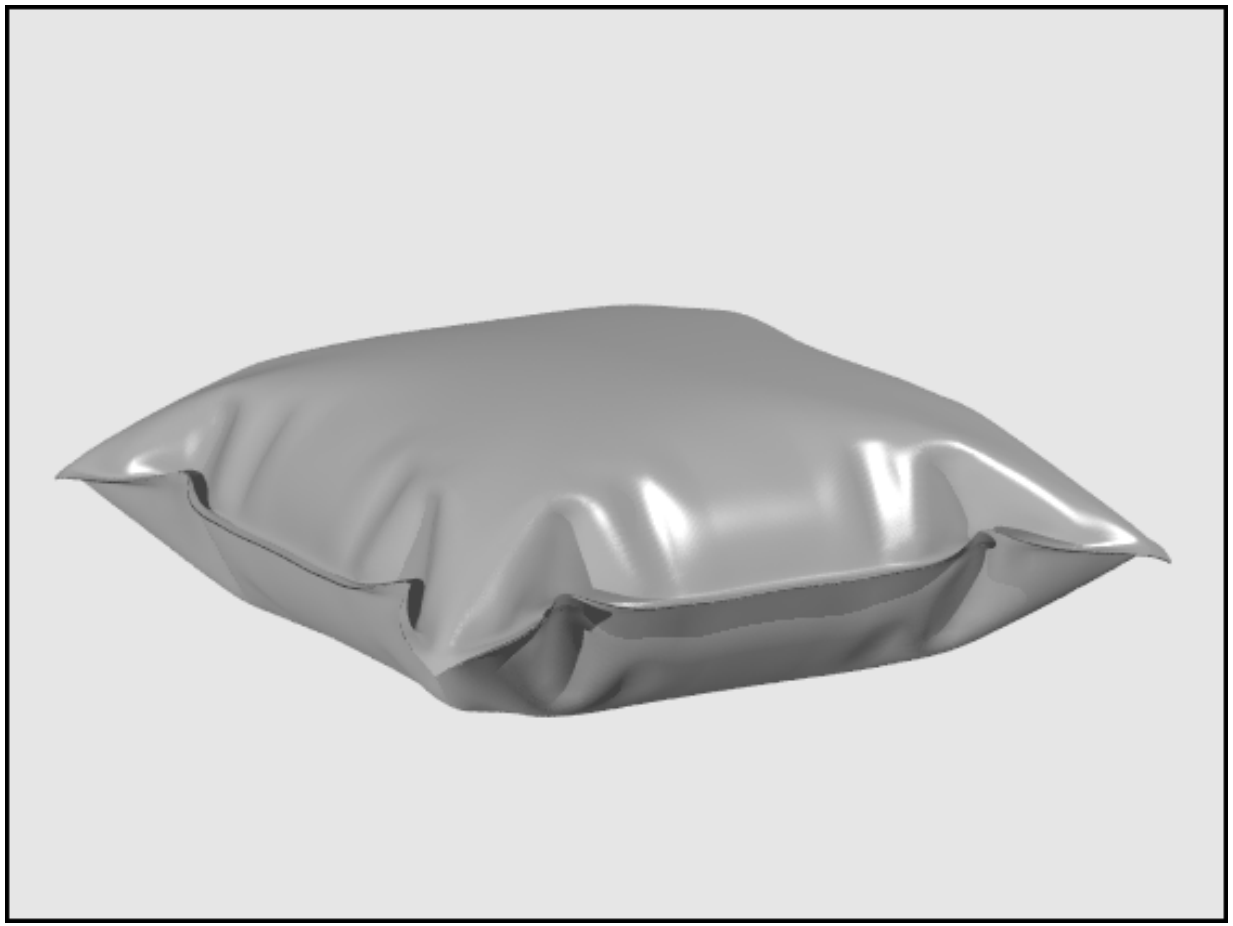
- Undeformed configuration: Two thin sheets, attached at the edges (thickness=10^{-3}, diagonal=1.2)
- Discretize with stretched tetrahedra and high-order DG elements
- Interior pressure p_0 applied at $t = 0$, quasi-static evolution





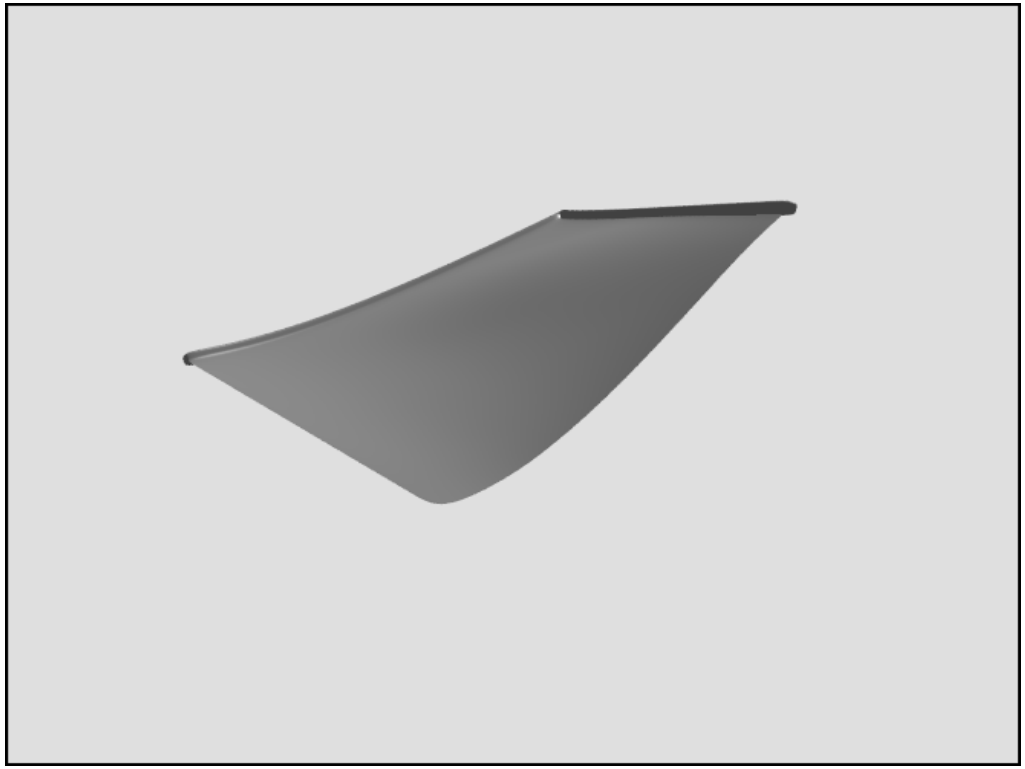
Example: Airbag Inflation

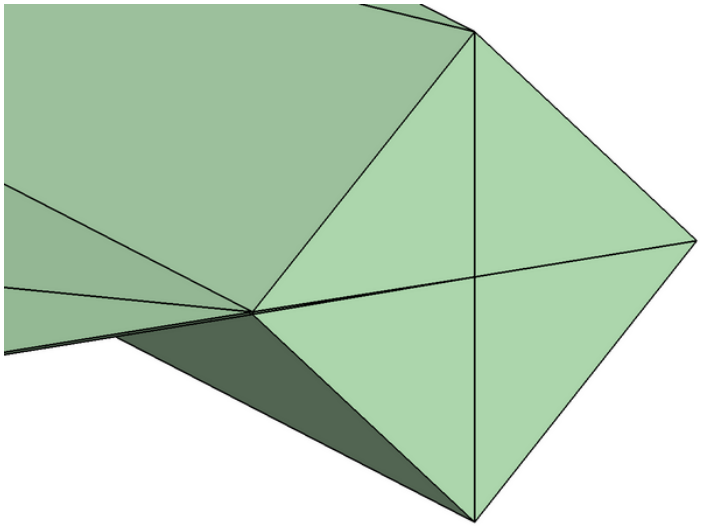
- 40-by-40 grid, 16000 tetrahedra, polynomial order $p = 3$
- No specialized membrane models required
- Hard to define convergence, but $\max(\Delta z) \rightarrow$ fixed value

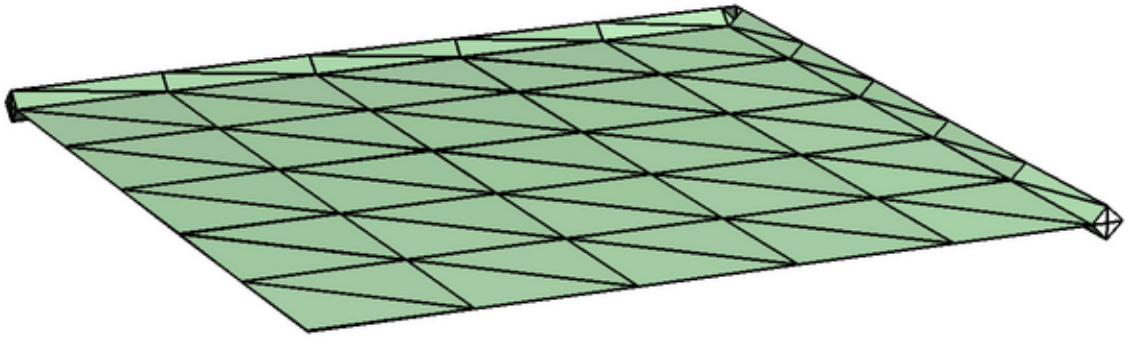


Volumetric Modeling of Coupled Beam/Membrane

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







Outline

Conclusions

- Important steps toward practical DG solver for realistic problems:
  - Non-linear elasticity for curved mesh generation
  - Efficient viscous discretization (the CDG method)
  - General purpose multigrid/block ILU preconditioner
  - Robustness by sensors and artificial viscosity, for RANS and shocks
  - Optimal accuracy for deformable domains by mapping approach
  - High order Lagrangian DG formulations for solid dynamics
- Current work: Full monolithic DG fluid-structure formulation, improved numerical schemes and algorithms, extensions for LES/DES, applications in flapping flight, aeroacoustics, instabilities in solids, and supersonic flows