Discontinuous Galerkin Methods for Fluid Flows and Implicit Mesh Generation

Per-Olof Persson

Massachusetts Institute of Technology

Collaborators: J. Peraire, N. C. Nguyen, G. Strang, A. Edelman

Stanford FPC Fluid Mechanics Seminar

April 25, 2006

Background

- Objective: Solve large scale problems in fluid mechanics efficiently and accurately on complex geometries
- Requirements:
  - Low dispersion/high accuracy discretization
  - Methods based on unstructured simplex meshes
  - Robust and fully automated solution process
  - Efficient solvers/simple parallelization

Application Areas

- Computational aeroacoustics
  - Direct solution of the compressible Navier-Stokes equations
  - No modeling of acoustic waves (Lighthill, linearized Euler, etc)
  - High accuracy essential to capture turbulent noise sources, propagate acoustic waves, and to ensure stability at low Mach numbers
  - Complex geometries, e.g. in wind noise prediction for vehicles
- Other DNS/LES/DES problems
  - High order methods a clear benefit in DNS/LES region
  - Need flexible meshes e.g. in the boundary layers
- Drag prediction
  - Solve high Reynolds number transonic flow for various geometries
  - Determine drag coefficient accurately

Numerical Schemes

- Various numerical methods:
  - FD: Only structured grids
  - FVM: Only second order accurate
  - HO FVM: Hard to generalize, boundary conditions difficult
  - FEM/SUPG: Not clear how to stabilize for high orders
  - DG: Unstructured, general, but rarely used for practical problems
- Our goal: Make DG methods competitive for real-world problems
  - New low-memory compact viscous discretization
  - Multigrid/ILU preconditioned Krylov solvers
  - Shock capturing with subcell resolution
  - RANS turbulence modeling by Spalart-Allmaras and k-ω models

The Discontinuous Galerkin Method

- (Lesaint/Raviart 1974, Cockburn/Shu 1989,1990,1998,2001, 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

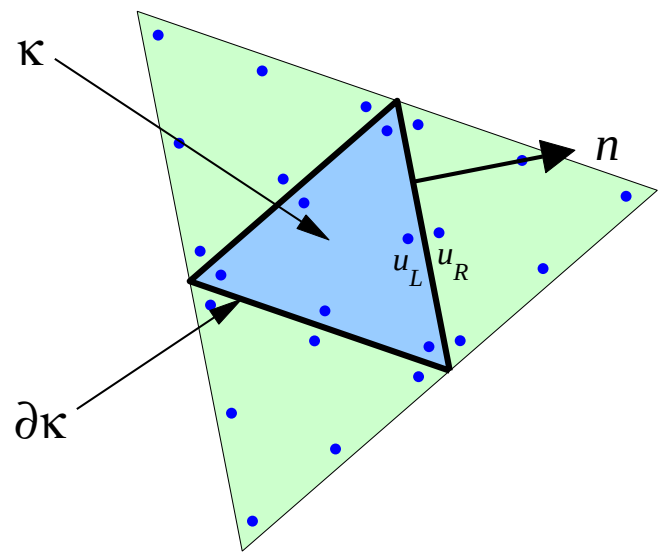
- Integrate by parts:

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

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

- Global view: 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 problems



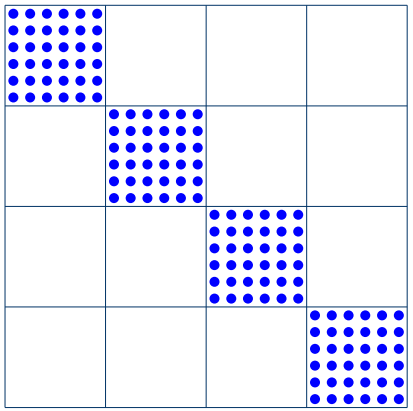
The DG Method – Observations

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

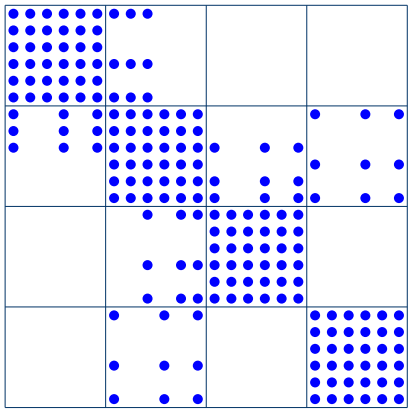
$$(\boldsymbol{u}_h)_t A_\kappa + \int_{\partial\kappa} \mathcal{H}_i(\boldsymbol{u}_h^+, \boldsymbol{u}_h^-, \hat{\boldsymbol{n}})\, ds = 0$$

- Boundary conditions enforced naturally for any polynomial degree p
- Mass matrix block-diagonal (no overlap between element basis functions)
- Edge integrals connect neighboring elements – block-wise compact stencil

Mass Matrix



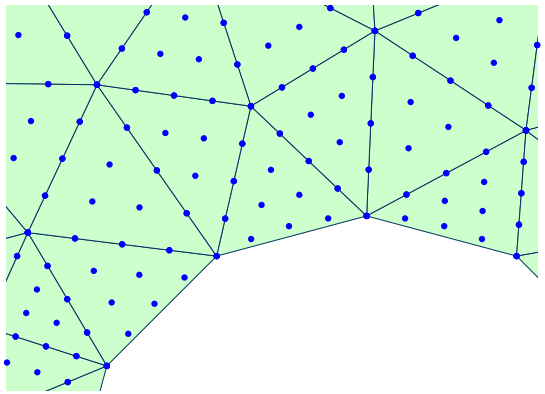
Jacobian



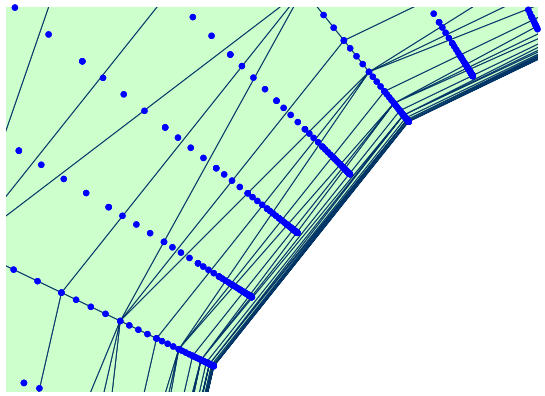
Nodal Basis and Curved Boundaries

- Nodal basis within each element $\longrightarrow$ sparse element connectivities
- Isoparametric treatment of curved elements (not only at boundaries)

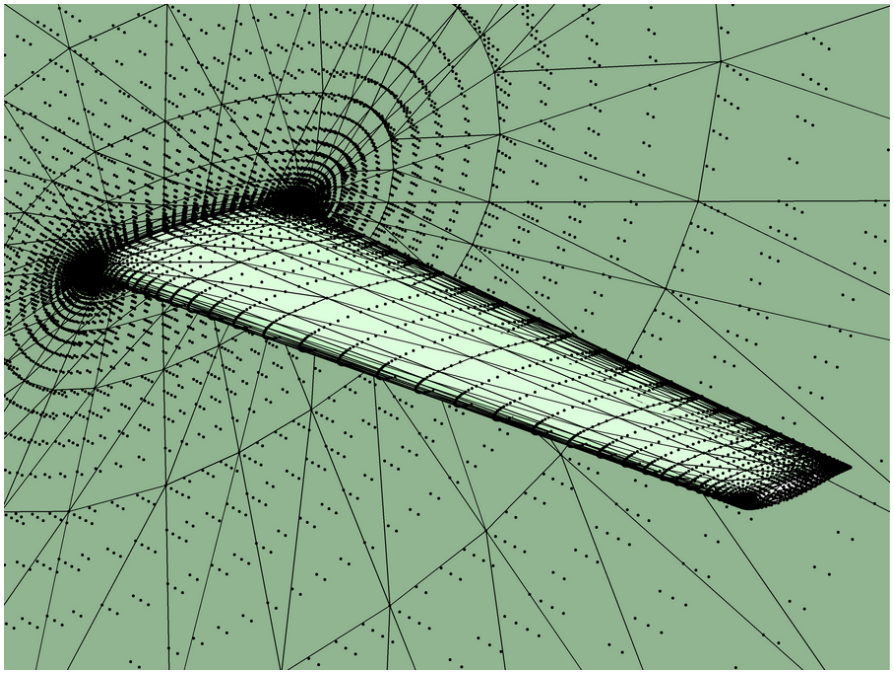
Isotropic mesh



Anisotropic mesh



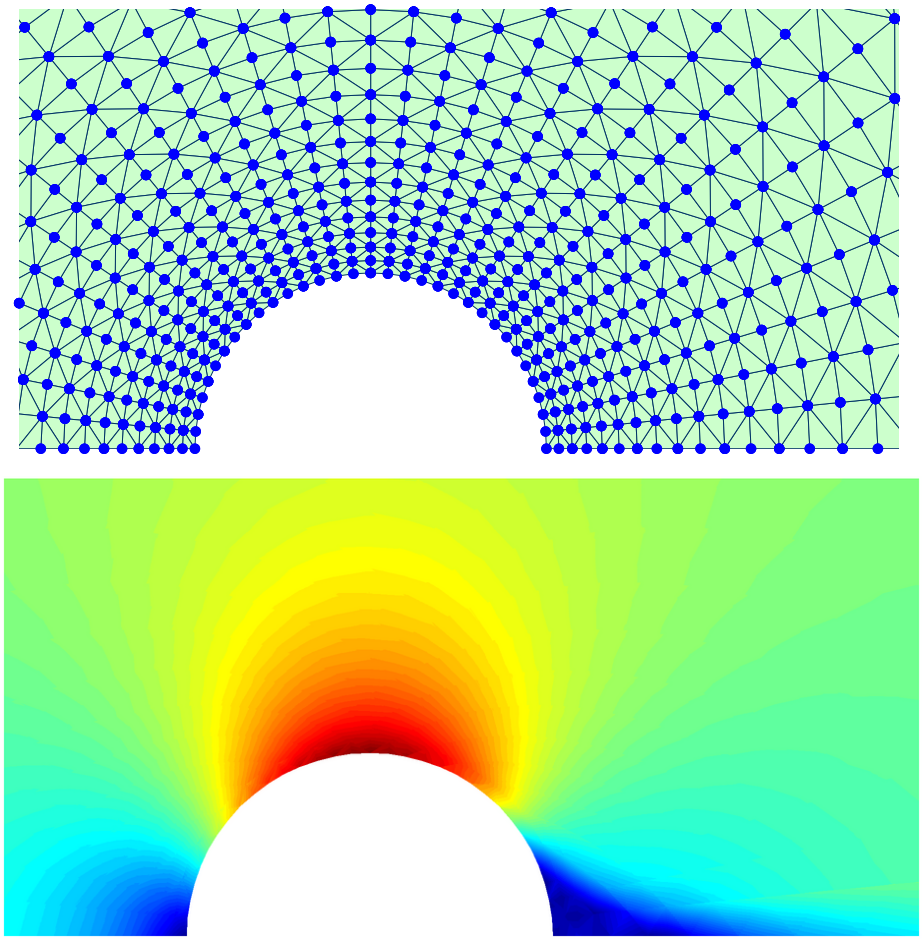
W1 wing (Drag Prediction Workshop)



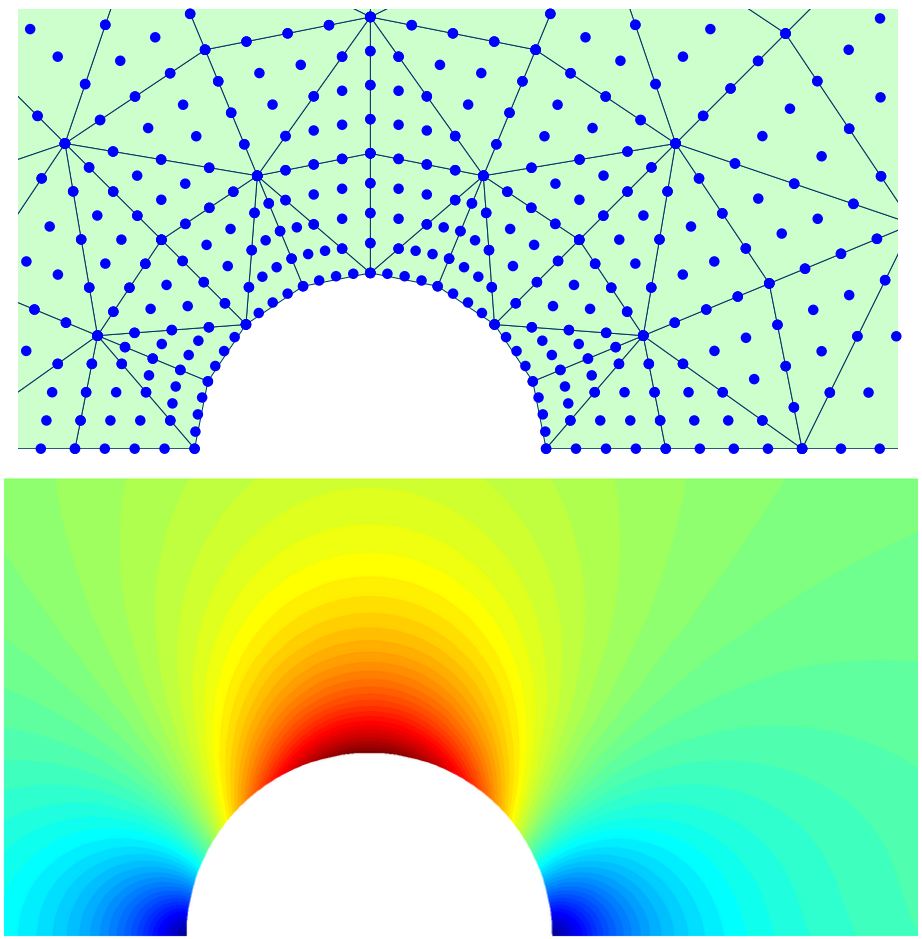
Example: Inviscid 2-D Flow

- Lower dissipation with high order schemes $\longrightarrow$ Higher accuracy/DOF

24576 DOFs, $p = 1$



7680 DOFs, $p = 4$



Discretization of Viscous Terms

- Write as system of first order equations:

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

- Discretize using DG, various formulations and numerical fluxes:
  - The BR1 scheme (Bassi/Rebay 1997):

    Averaging of fluxes, connections between non-neighboring elements
  - The BR2 scheme (Bassi/Rebay 1998):

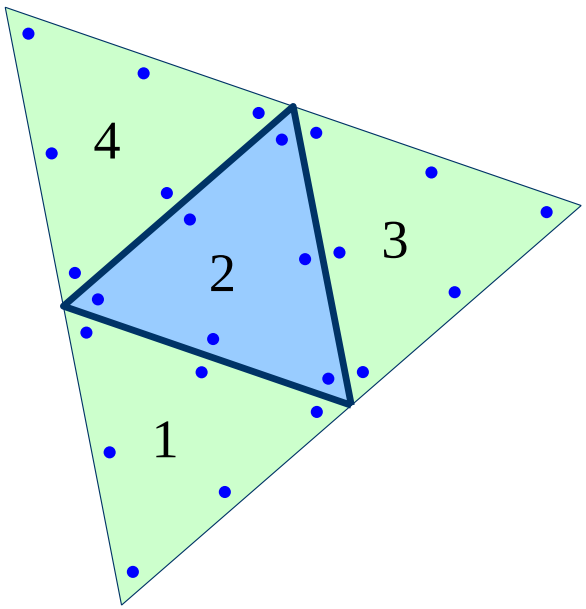
    Uses lifting operator, connections only between neighbors
  - The LDG scheme (Cockburn/Shu 1998):

    Upwind/Downwind, sparse connectivities, some wide connections
  - The CDG scheme (Peraire/Persson 2006):

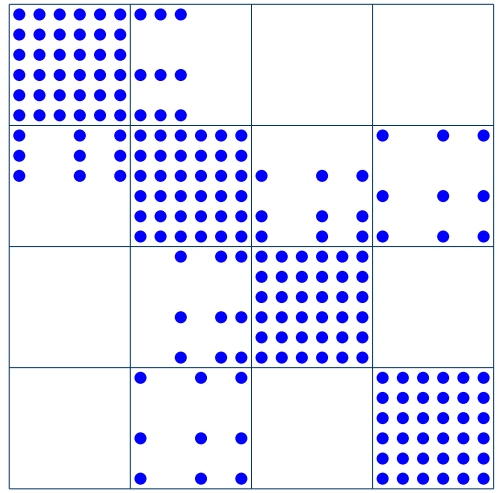
    Modification of LDG for local dependence – sparse and compact

Viscous Discretizations – Stencils

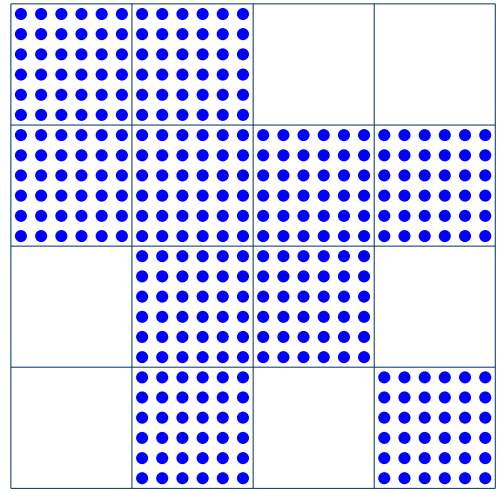
- About 50% fewer DOFs with CDG than BR2 (3-D tetrahedra, $p = 4$)



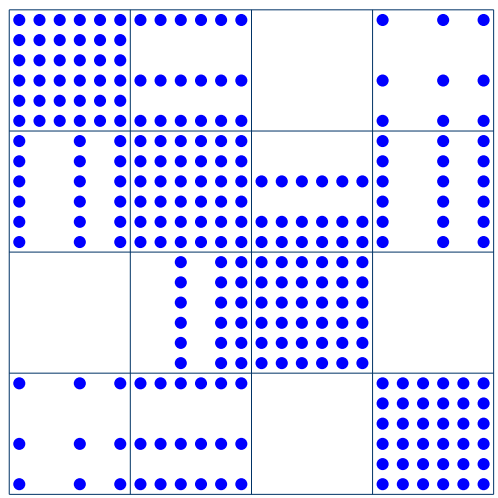
Inviscid



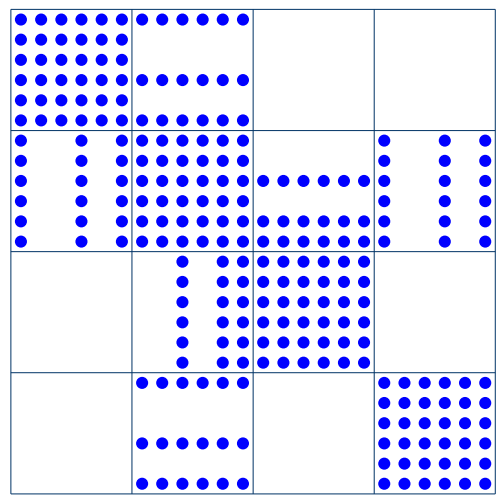
BR2



LDG



CDG

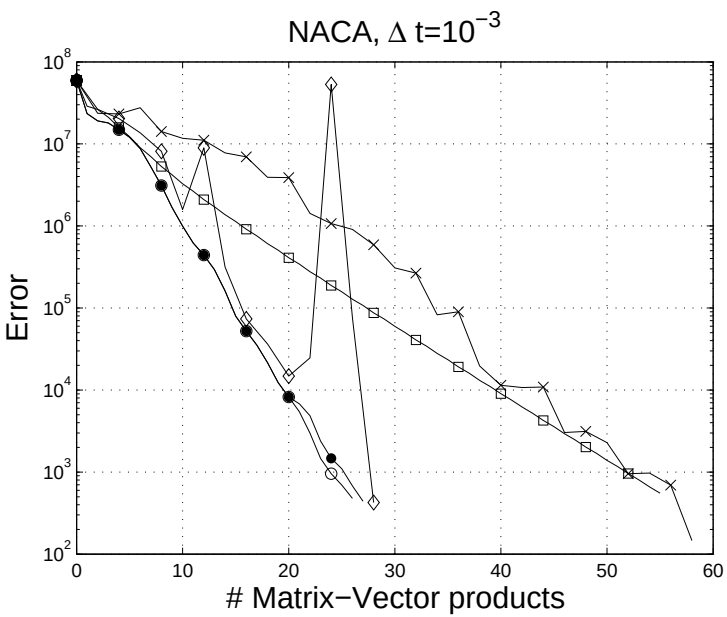


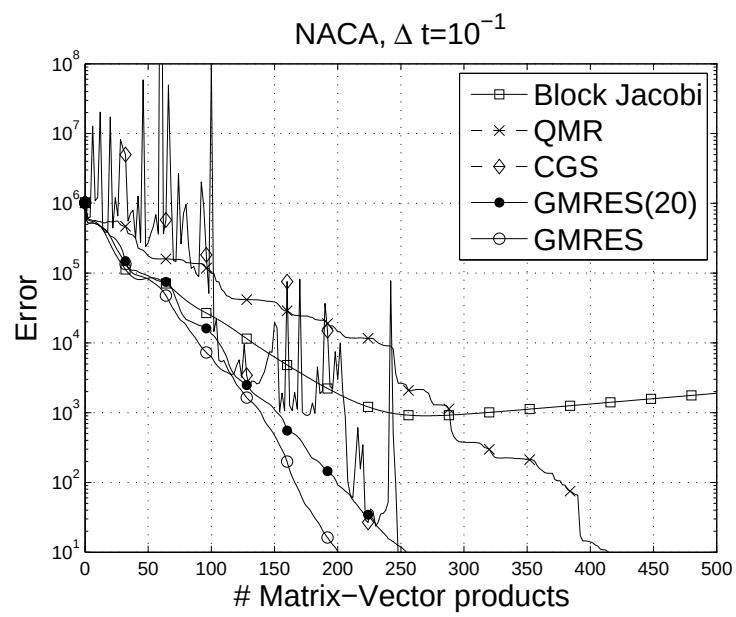
Time Integration and Iterative Solvers

- Discretization in space using DG gives system of ODEs

$$\boldsymbol{M}\dot{\boldsymbol{U}} = \boldsymbol{R}(\boldsymbol{U}) \text{ with Jacobian } \boldsymbol{K} = d\boldsymbol{R}/d\boldsymbol{U}$$

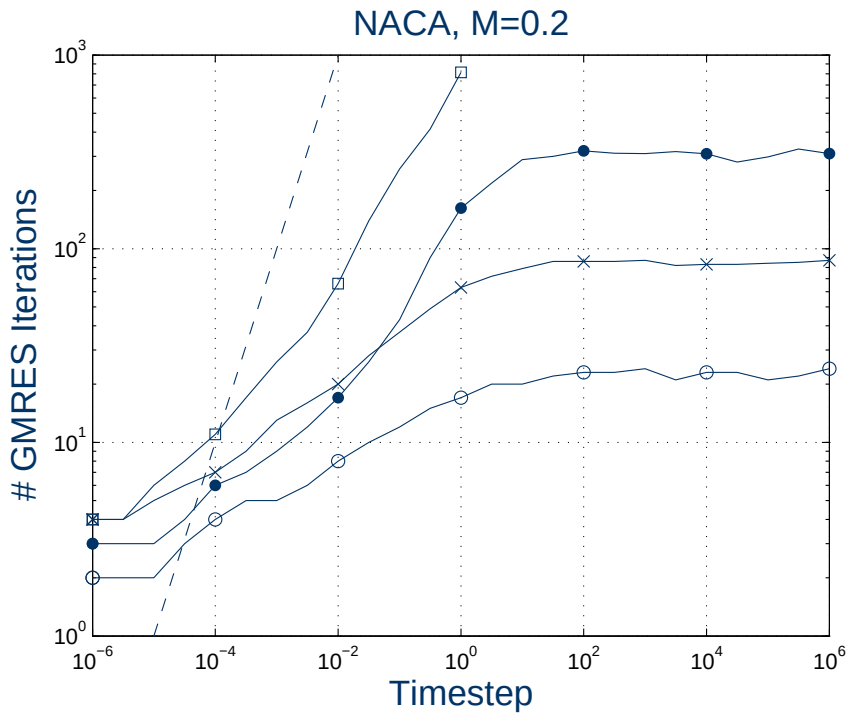
- Implicit time integration requires solving $\boldsymbol{A} = \alpha_0 \boldsymbol{M} - \Delta t \boldsymbol{K}$
- Block-Jacobi efficient solver for simple problems (inviscid flow, isotropic)
- In general we need Krylov solvers with sophisticated preconditioning

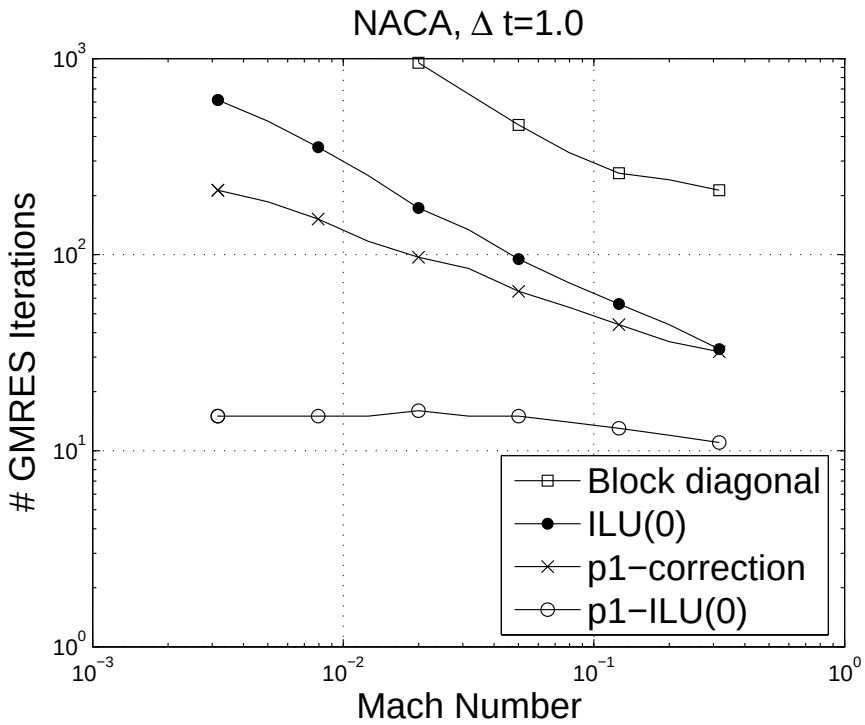




Multiscale Preconditioner with Block-ILU(0)

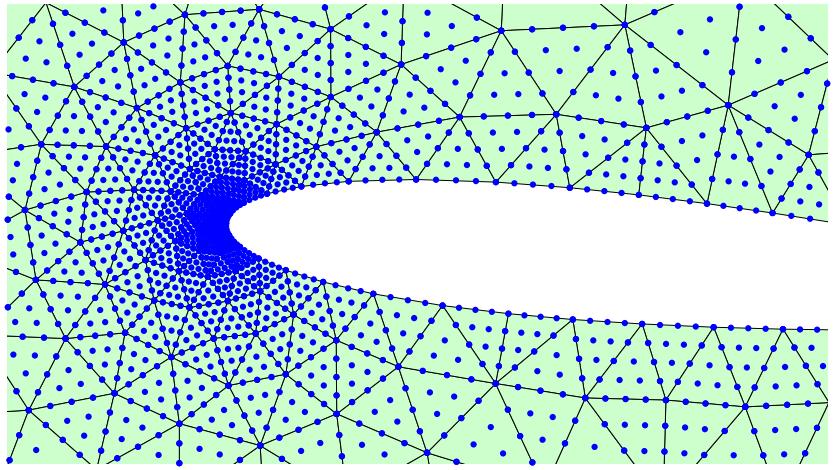
- ILU(0): Gaussian elimination, ignoring all new matrix elements (fill-in)
- A remarkably efficient preconditioner: Apply a coarse grid correction followed by post-smoothing with Block-ILU(0) (Persson/Peraire 2006)
- In particular it makes convergence almost independent of Mach number



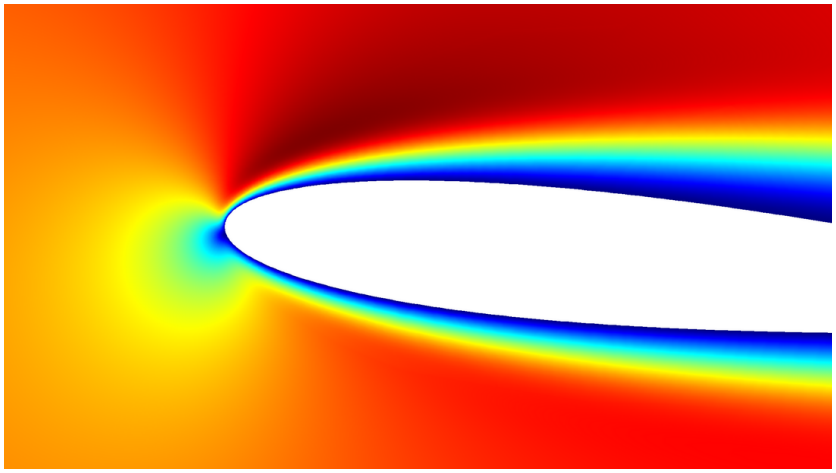


Example: Viscous Flow

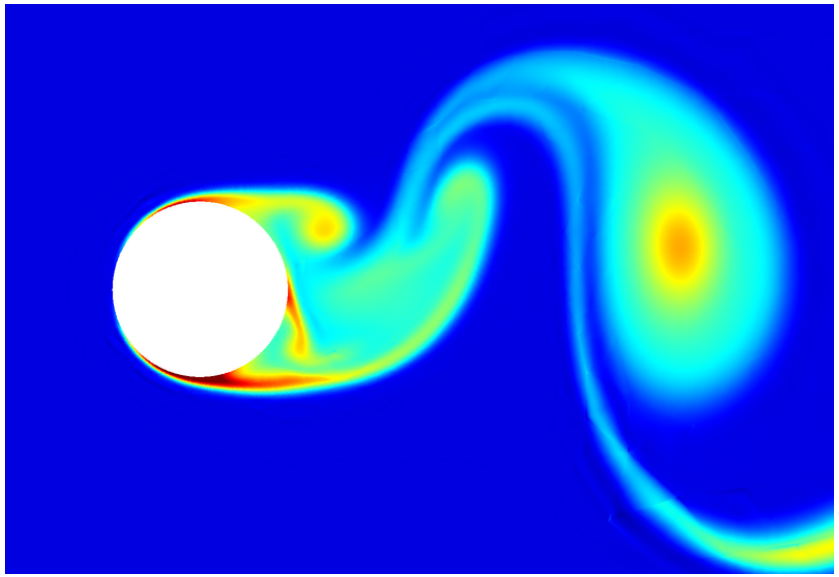
NACA 0012, Re=1000, (Mesh)



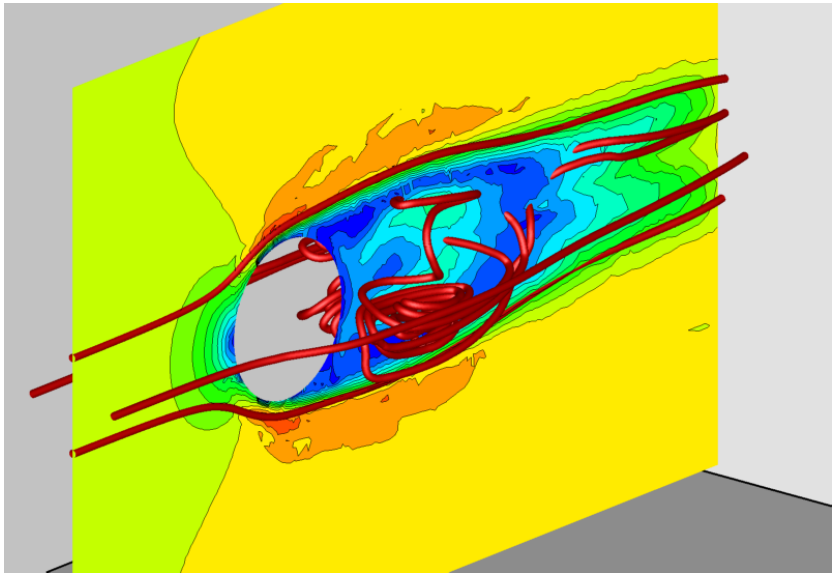
NACA 0012, Re=1000 (Mach)



Cylinder, Re=2000 (Entropy)

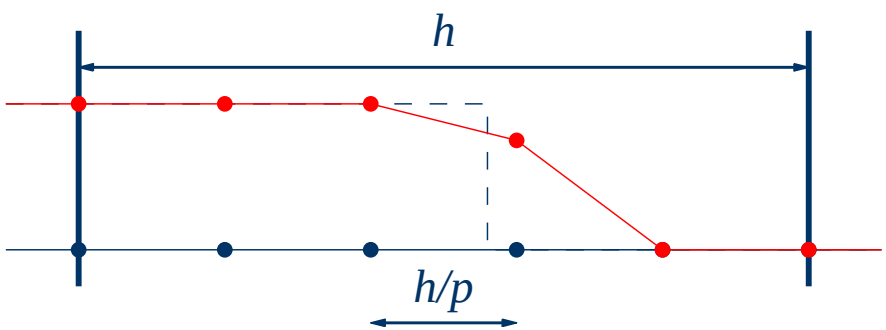


Sphere, Re=1000 (Mach & Velocity)



Shock Capturing with Artificial Viscosity

- Need stabilization for shock problems with $p > 0$
- Common approaches: Slope limiting, filtering – shock width $\delta_s \sim 2h$
- Use artificial viscosity with a non-linear sensor: (Persson/Peraire 2006)
  - Select parameters based on available resolution
  - Consistent treatment of high order terms (CDG)
  - Do not exploit nature of discontinuous approximation
  - Shock width $\delta_s \sim 4h/p$ (subgrid resolution, p-convergence)
  - Continuous sensor and viscosity – Newton solver



$4h/p \ll 2h$ for high p

Shock Sensor

- Determine regularity of the solution from rate of decay of frequencies
- Periodic Fourier case: $f(x) = \sum_{k=-\infty}^{\infty} g_k e^{ikx}$
  If $f(x)$ has m continuous derivatives $\longrightarrow$ $|g_k| \sim k^{-(m+1)}$
- Simplex element e: Expand solution in orthonormal Koornwinder basis ψ_i:

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

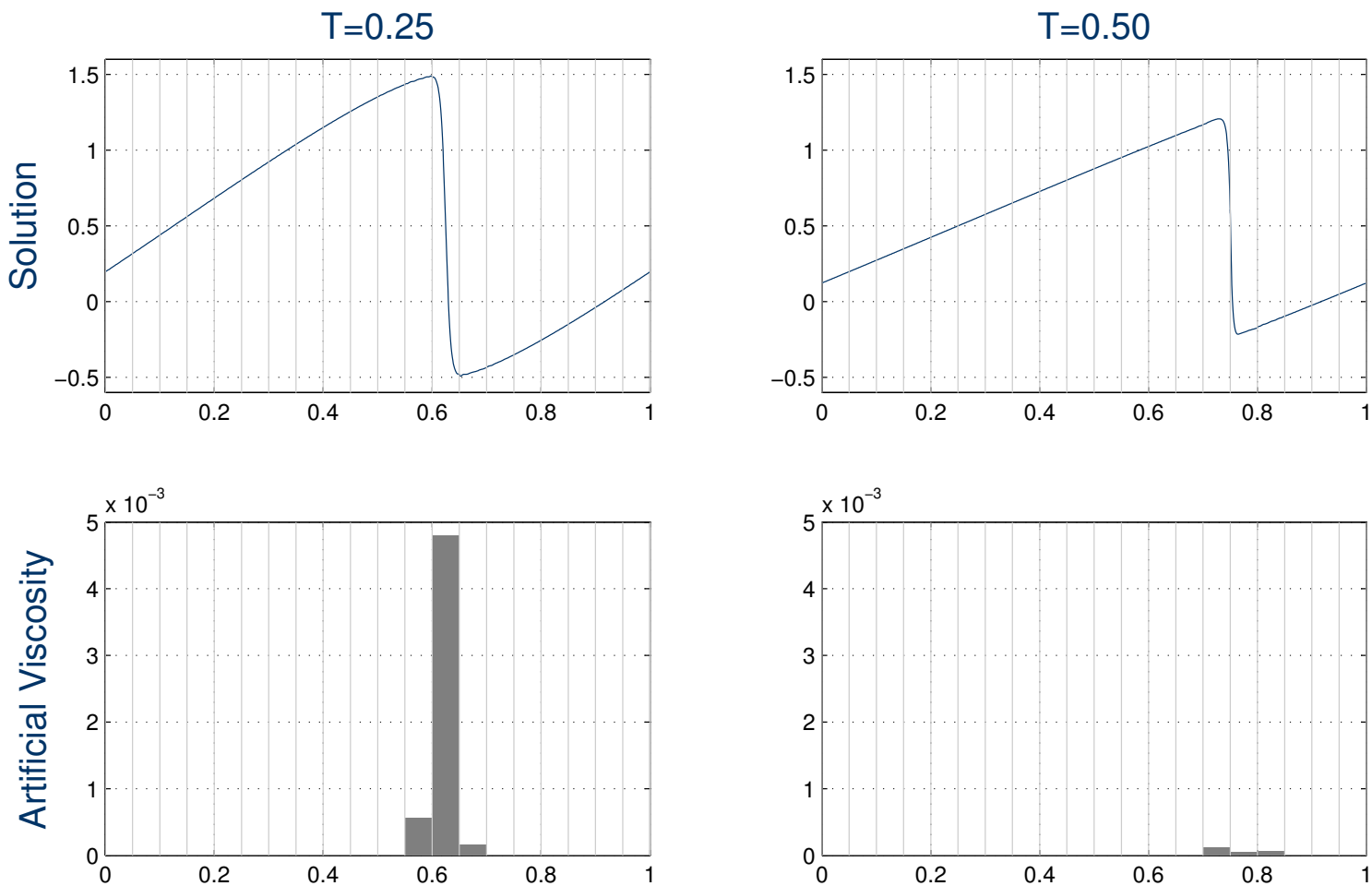
- Determine elemental piecewise constant ε_e

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

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

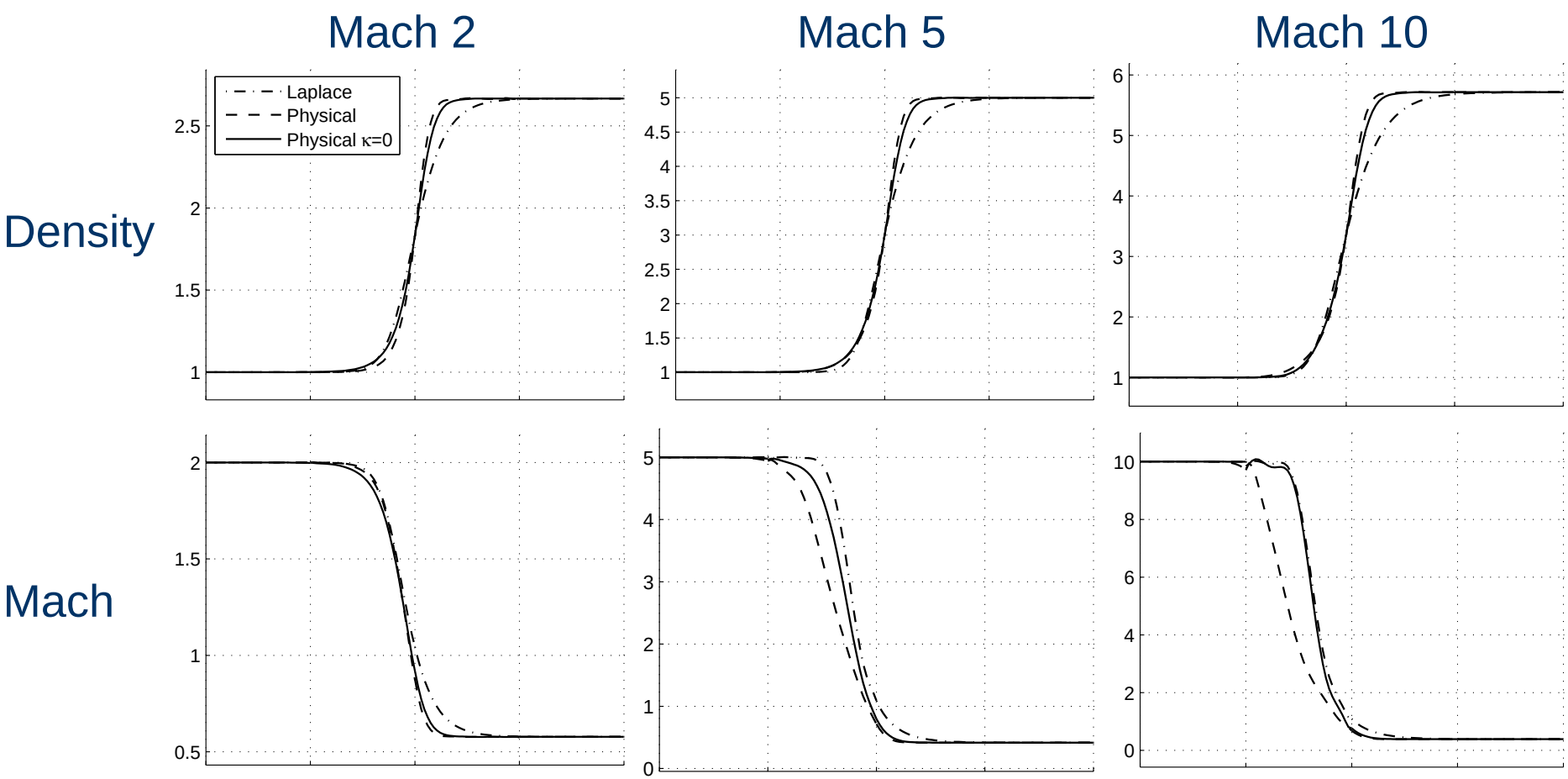
Scalar Results – Burgers' Equation ($p = 10$)

Initial condition $u(x, 0) = 1/2 + \sin 2\pi x$



Euler Equations – Artificial Viscosity Models

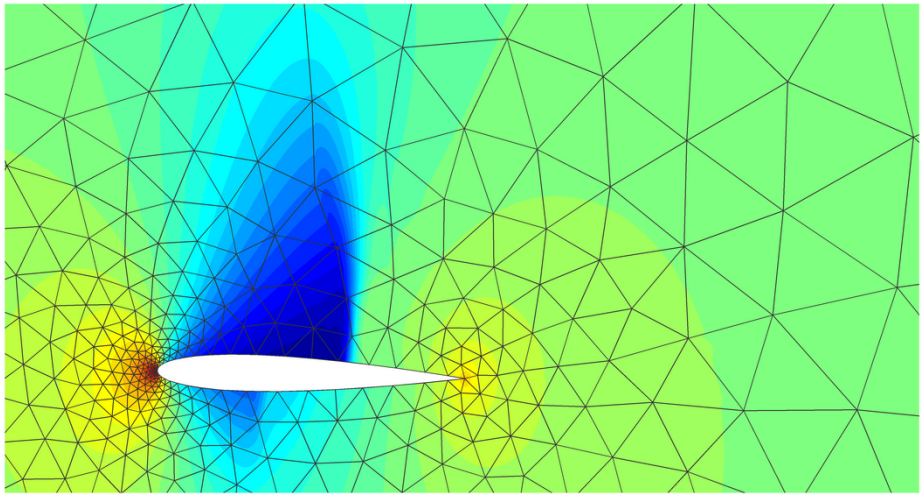
- Laplacian: $\frac{\partial \boldsymbol{u}}{\partial t} + \nabla \cdot \mathcal{F}(\boldsymbol{u}) = \nabla \cdot (\epsilon \nabla \boldsymbol{u})$
- Physical: $\frac{\partial \boldsymbol{u}}{\partial t} + \nabla \cdot \mathcal{F}(\boldsymbol{u}) = \nabla \cdot \mathcal{F}_v(\boldsymbol{u}; \mathrm{Re},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



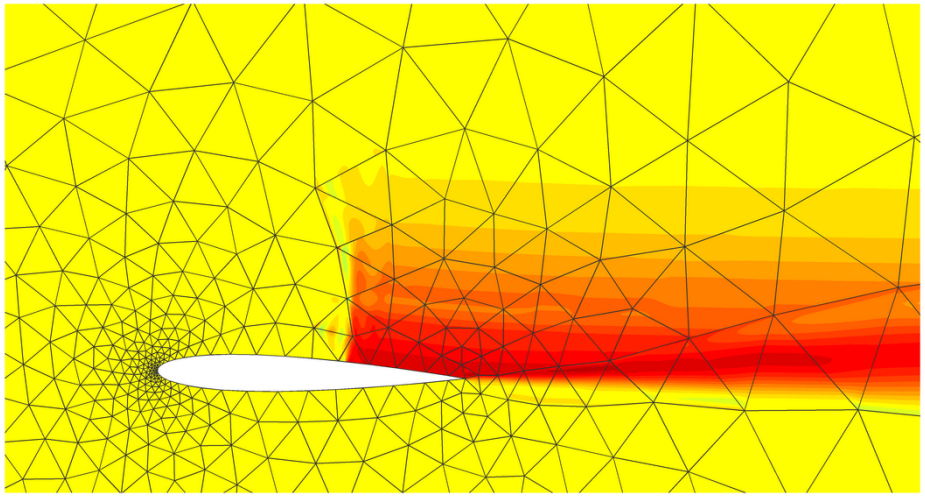
Transonic – Laplacian model/Density sensor

- NACA 0012, 1.5° attack angle
- $M = 0.8$ and $p = 4$
- Implicit steady-state solution with Newton solver

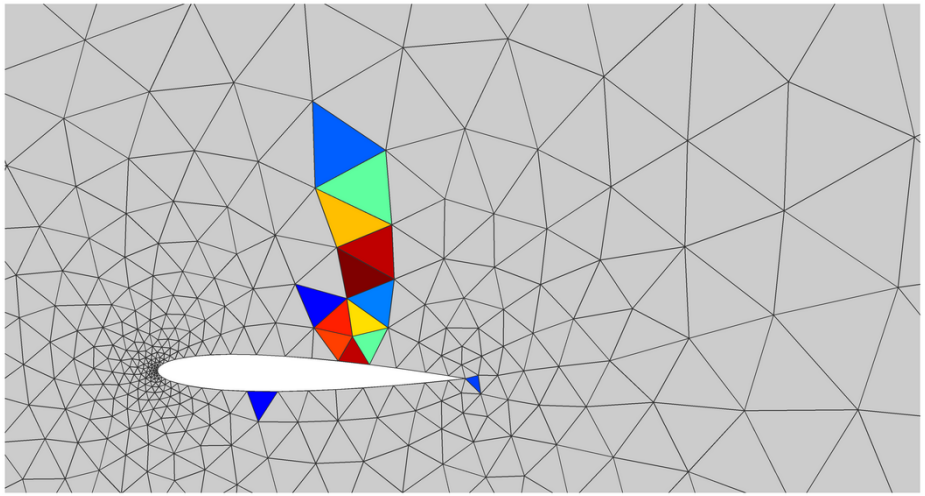
Pressure



Entropy



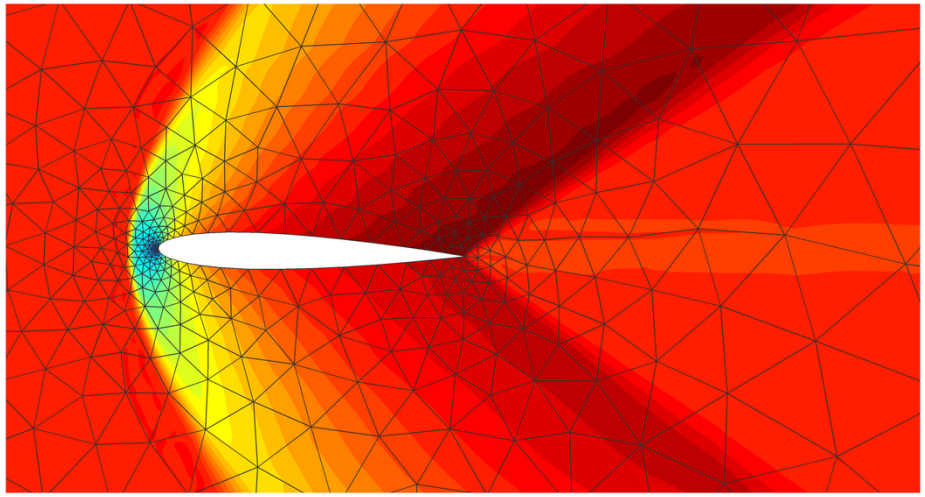
Sensor



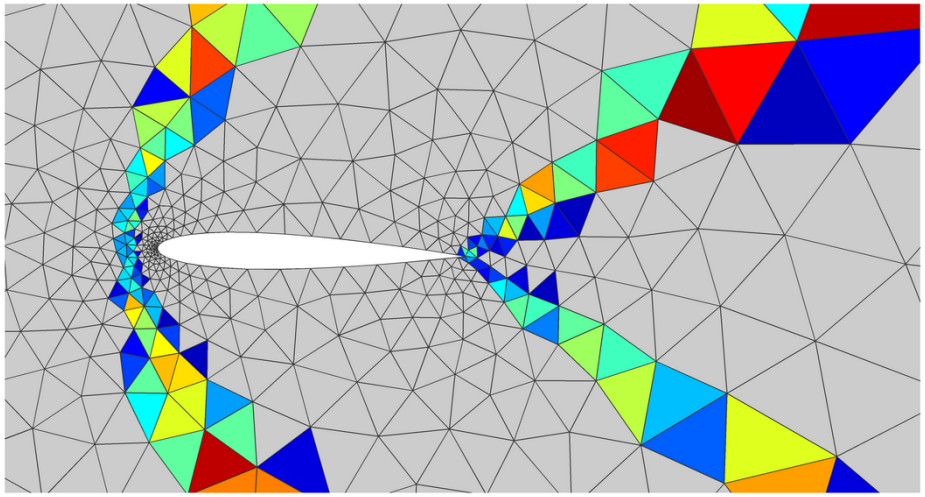
Supersonic – Laplacian model/Density sensor

- NACA 0012, 1.5° attack angle
- $M = 1.5$ and $p = 4$
- Implicit steady-state solution with Newton solver

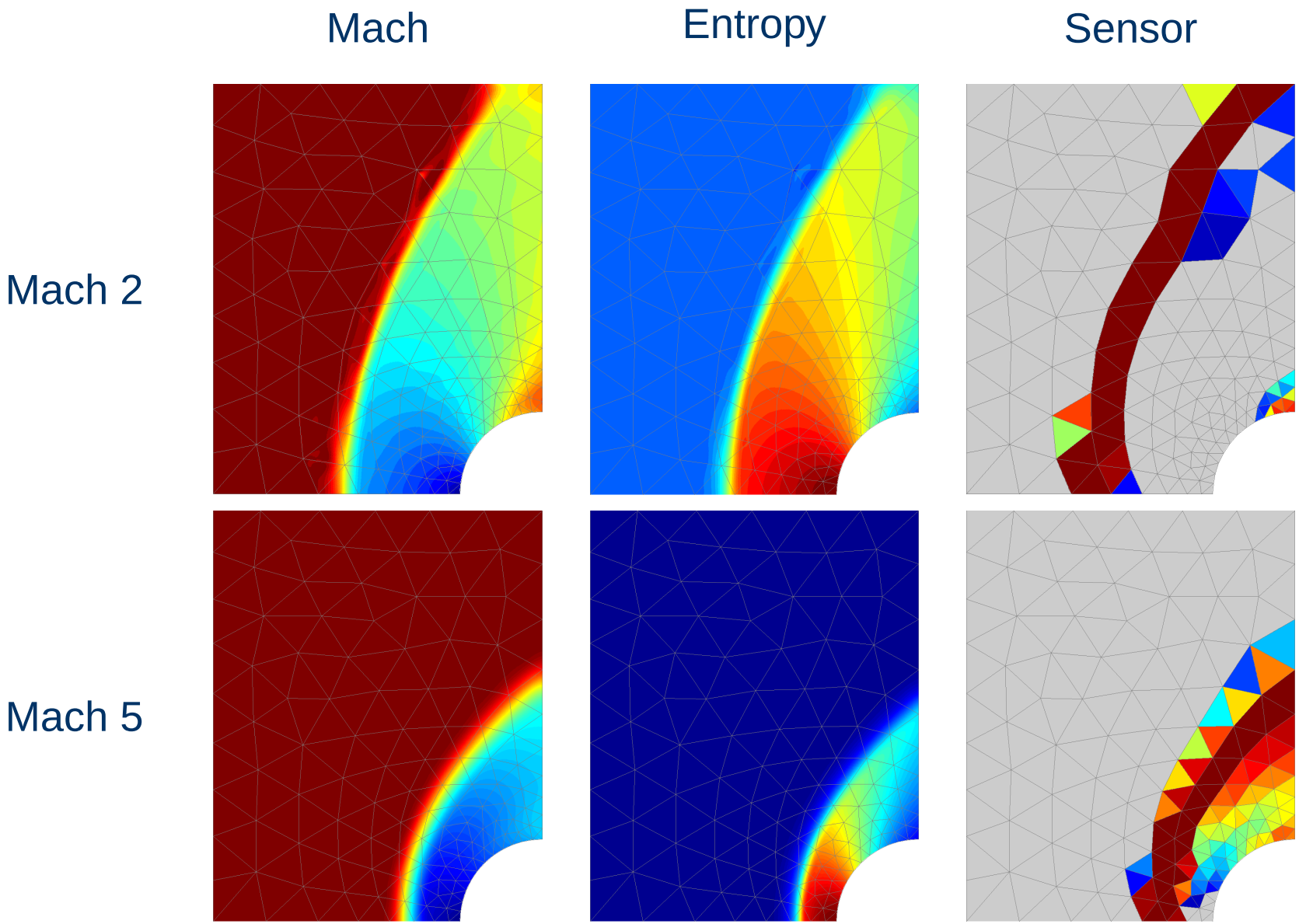
Mach



Sensor



Supersonic – Physical model/Entropy sensor



RANS Turbulence Modeling

- Recast the Spalart-Allmaras one-equation turbulence model into conservative form: (Persson/Nguyen/Peraire 2006)

$$\frac{\partial}{\partial t}(\rho\tilde{\nu}) + \frac{\partial}{\partial x_j}(\rho u_j \tilde{\nu}) - \nabla \cdot (f_d \nabla (\rho\tilde{\nu})) = S_R$$

$$S_R = c_{b1}\rho\tilde{S}\tilde{\nu} + \frac{c_{b2}}{\sigma}\rho(\nabla\tilde{\nu})^2 - c_{w1}\rho f_w \left[\frac{\tilde{\nu}}{d}\right]^2 - \nabla\rho \cdot \left(\frac{(\nu+\tilde{\nu})}{\sigma}\nabla\tilde{\nu}\right)$$

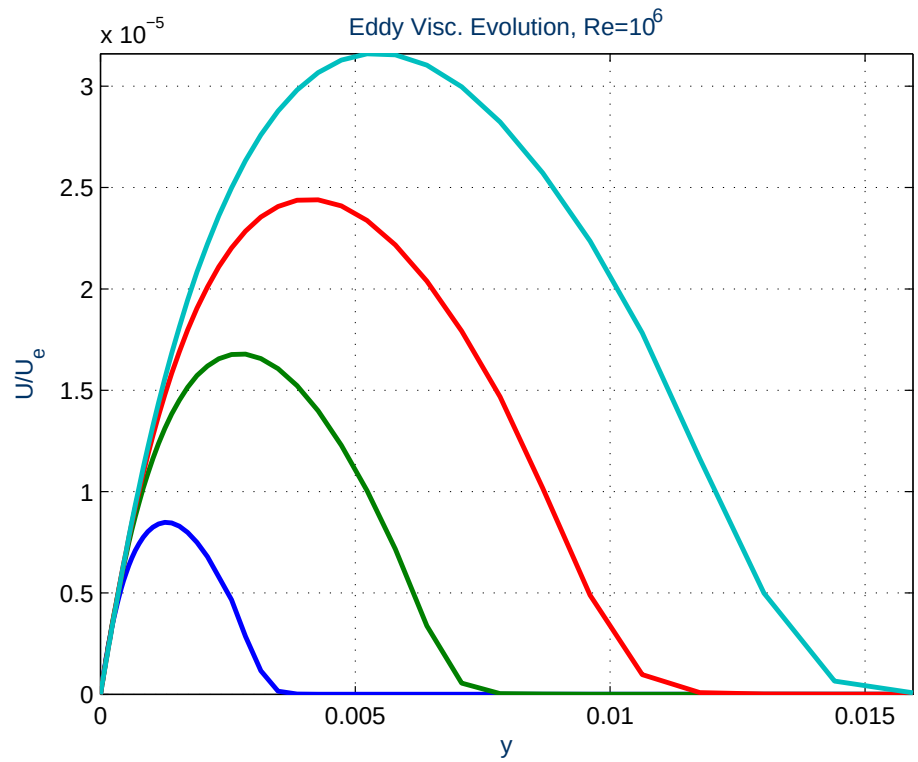
with modified coefficient

$$f_d = \left[(f_{d1})^6 + (f_{d2})^6\right]^{1/6}$$

$$f_{d1} = \frac{\nu+\tilde{\nu}}{\sigma}$$

$$f_{d2} = c_d \nu \left(\frac{\Delta_{\min}/p}{0.16 Re^{-1/7} + d}\right)^2,$$

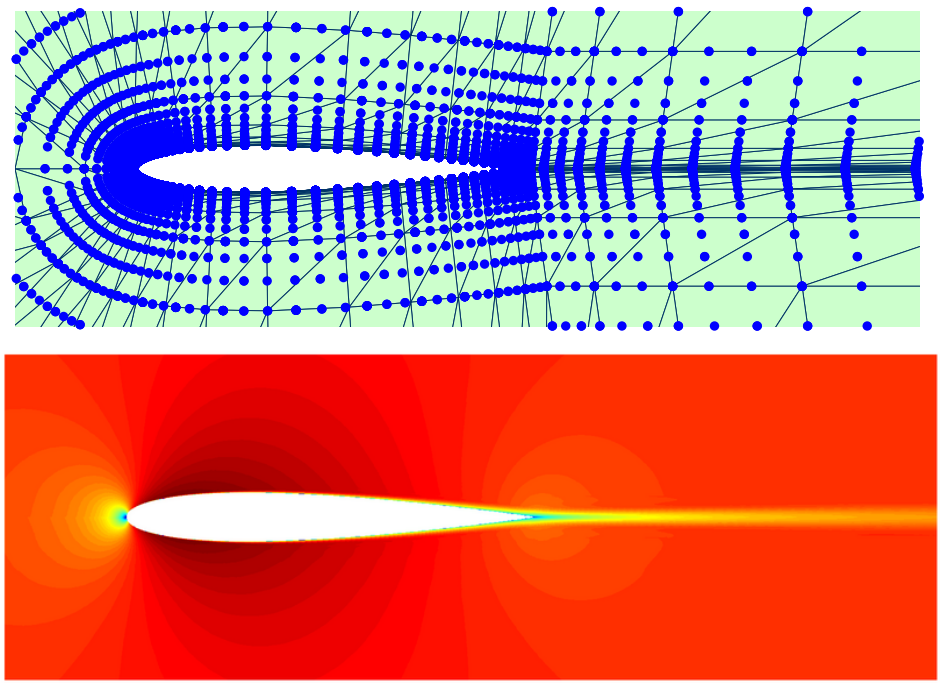
- Modification introduced for $p \geq 1$ to avoid negative eddy viscosity at end of boundary layer



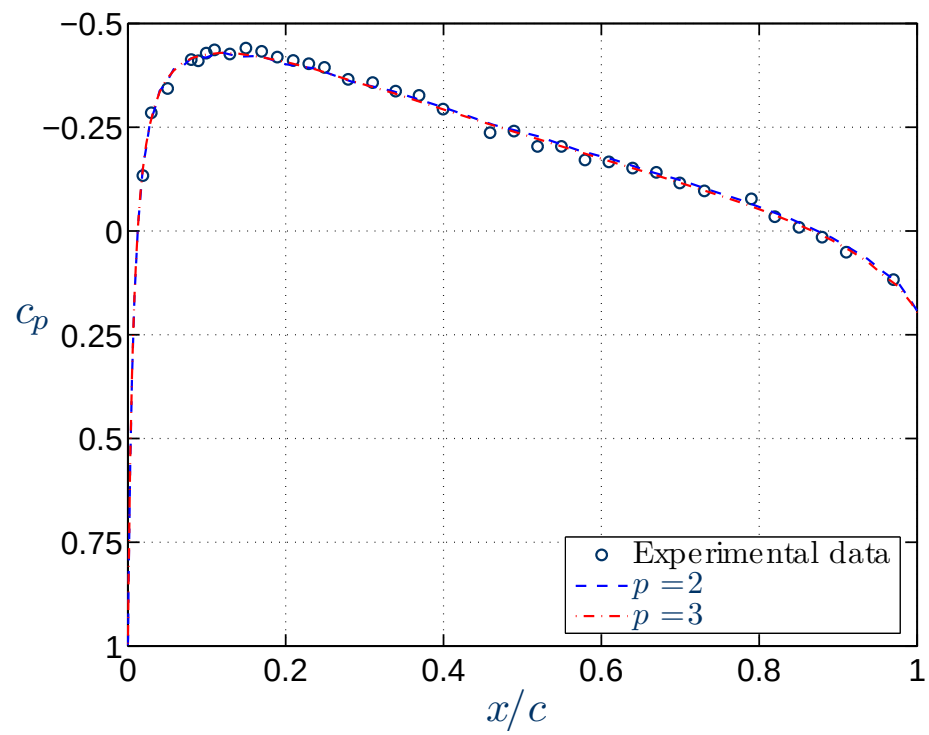
Turbulent Flow over NACA 0012 Airfoil

- Spalart-Allmaras turbulence modeling, $M = 0.3$, $\mathrm{Re} = 1.86 \cdot 10^6$
- p-convergence and good agreement with experimental data

Mesh and Mach



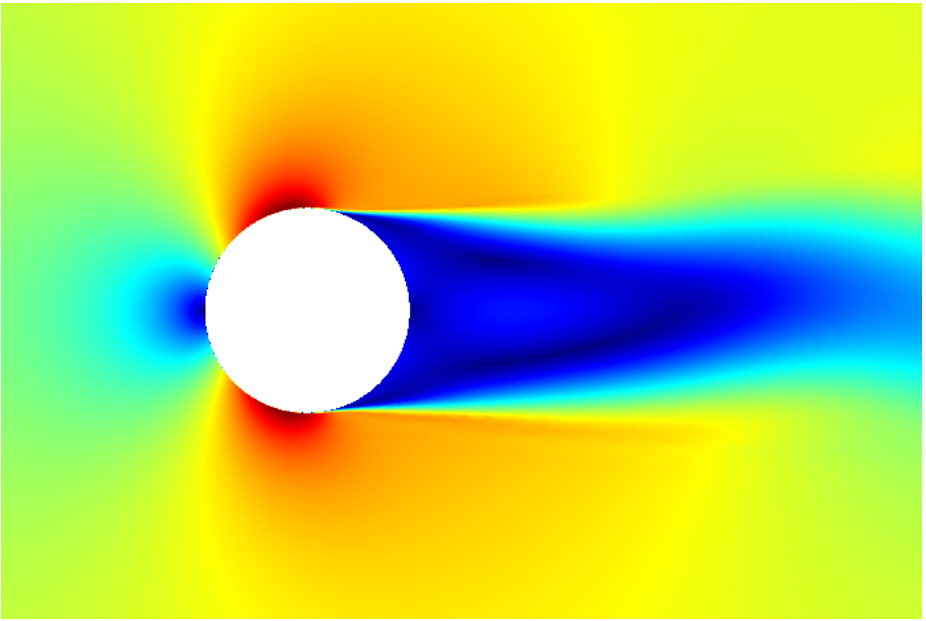
Pressure Coefficient



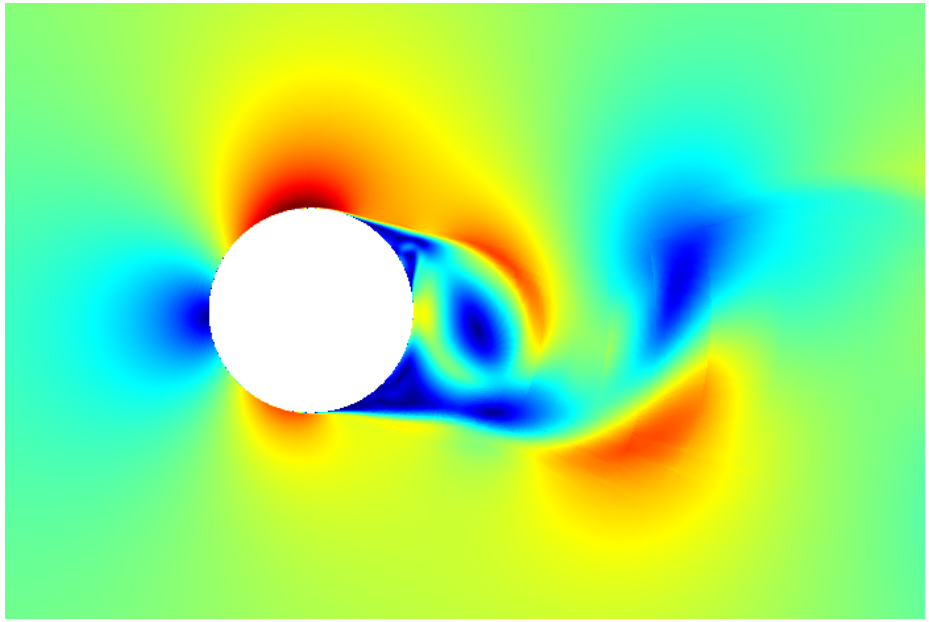
Unsteady Turbulent Flow over Cylinder

- Spalart-Allmaras turbulence modeling, $M = 0.2, \mathrm{Re} = 5 \cdot 10^5, p = 2$
- LES in 2-D not physically meaningful, but comparison shows how URANS smears out complex flow features

Mach, URANS



Mach, DES

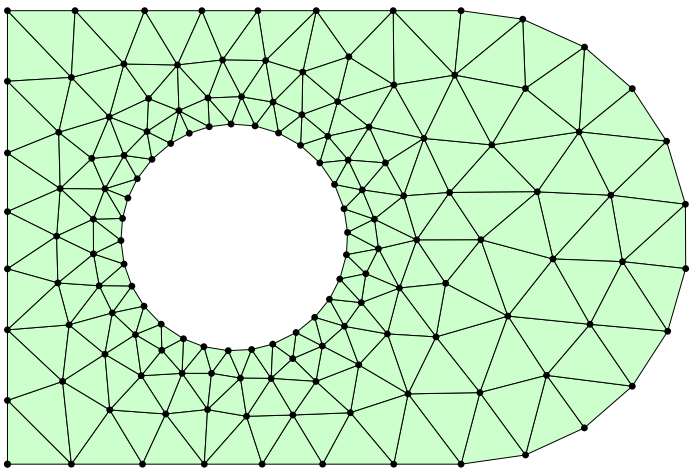


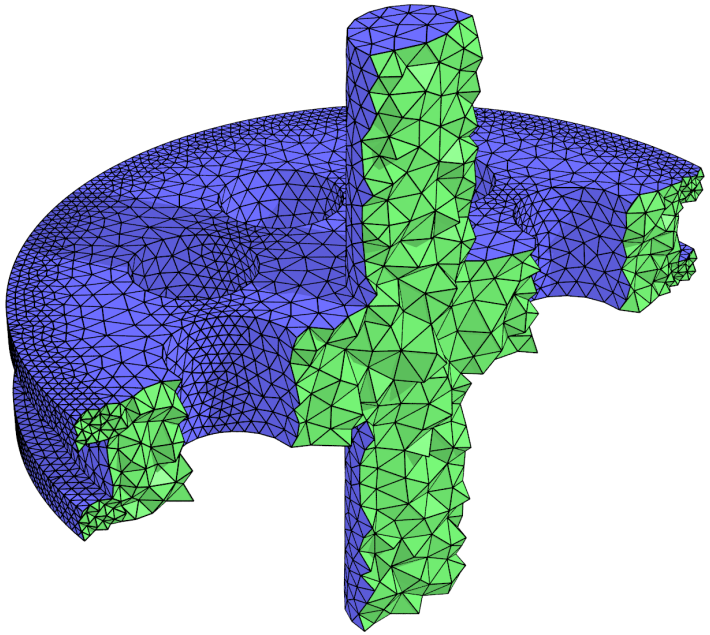
Summary DG Methods

- Contributions:
  - Low-cost compact viscous discretization
  - Highly efficient multigrid/ILU preconditioner
  - Shock capturing with subcell resolution using artificial viscosity
  - DG discretization of modified Spalart-Allmaras turbulence model
- Current status:
  - Flexible, lightweight, and highly efficient n-D code fully developed
  - Implicit and explicit solvers, various Krylov solvers and precondtitioners
  - Handles large 2-D problems and moderate size 3-D problems in serial
- Current work:
  - Parallelization of code, implementation of LES/DES capabilities, solution of large scale 3-D problems

Mesh Generation

- Given a geometry, determine node points and element connectivity
- Resolve the geometry and high element qualities, but few elements
- Applications: Numerical solution of PDEs (FEM, FVM, BEM), interpolation, computer graphics, visualization
- Popular algorithms: Delaunay refinement, Advancing front, Octree

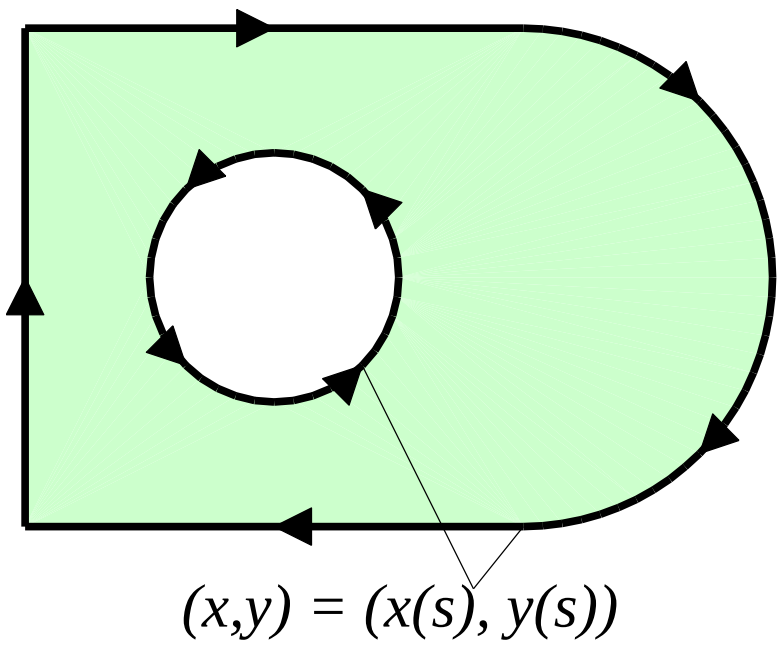




Geometry Representations

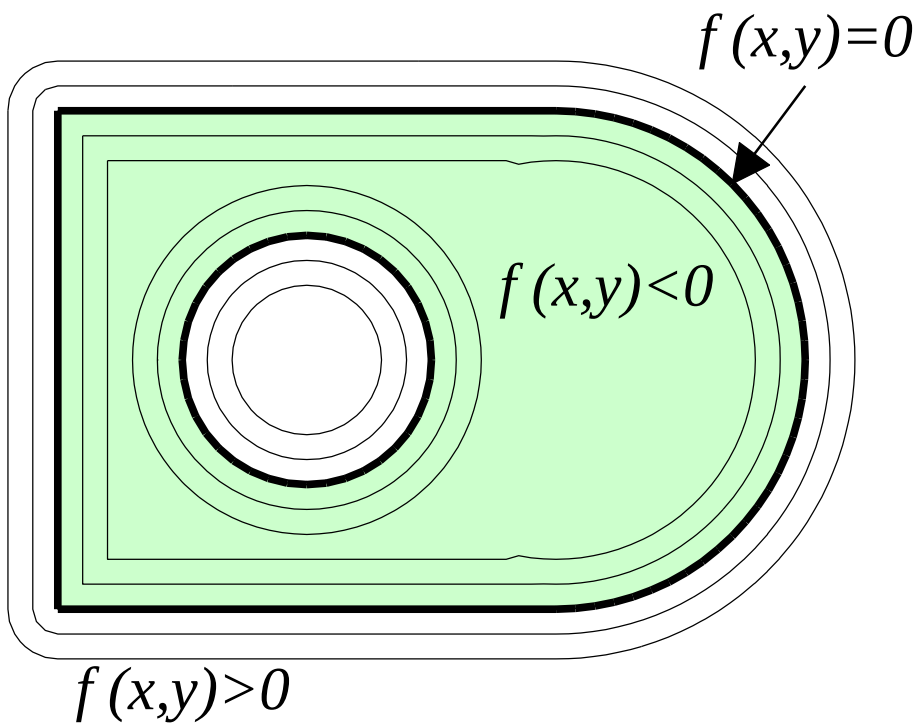
Explicit Geometry

- Parameterized boundaries



Implicit Geometry

- Boundaries given by zero level set



Explicit vs Implicit

- Most CAD programs represent geometries explicitly (NURBS and B-rep)
- Most mesh generators require explicit expressions for boundaries (Delaunay refinement, Advancing front)
- Increasing interest for implicit geometries:
  - Easy to implement: Set operations, offsets, blendings, etc
  - Automatic “geometry repair” from resolution of implicit function
  - Extend naturally to higher dimensions
  - Level set methods for evolving interfaces
  - Image based geometry representations (MRI/CT scans)

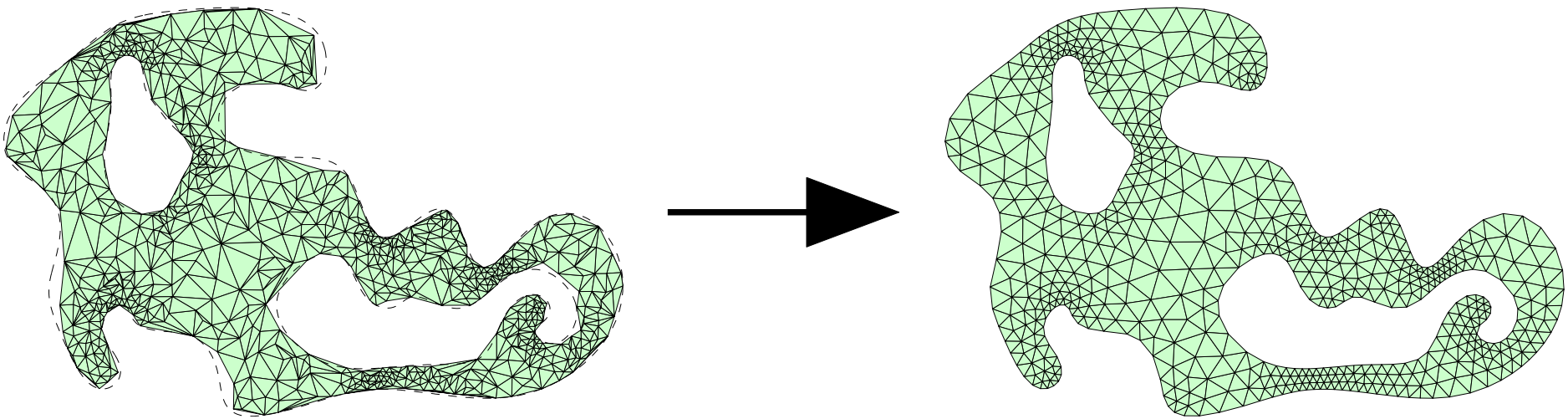
The New Mesh Generator

(Persson/Strang, SIAM Review 46(2), 2004)

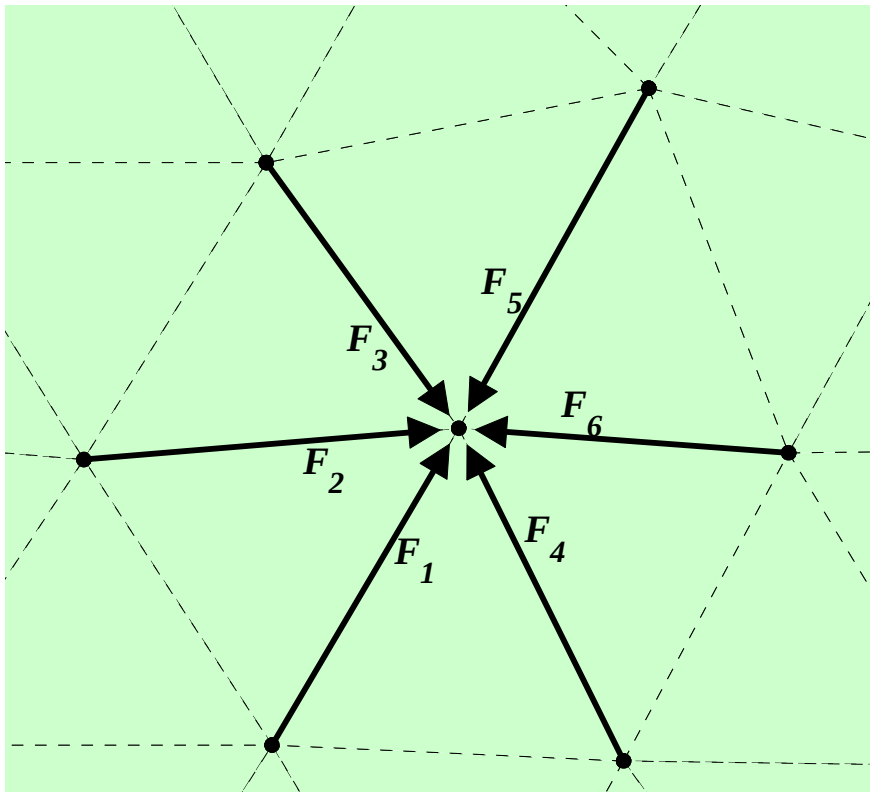
1. Start with *any* topologically correct initial mesh, for example random node distribution and Delaunay triangulation

2. Move nodes to find force equilibrium in edges

   - Project boundary nodes using ϕ
   - Update element connectivities



Internal Forces



For each *interior* node:

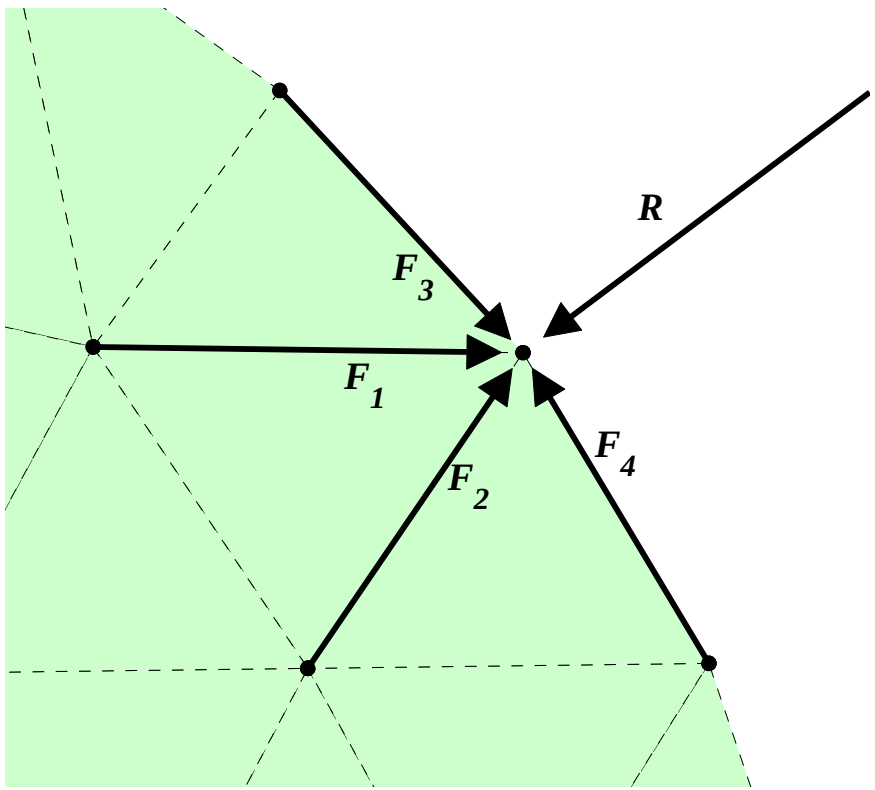
$$\sum_i \boldsymbol{F}_i = 0$$

Repulsive forces depending on edge length ℓ and equilibrium length ℓ_0:

$$|\boldsymbol{F}| = \begin{cases} k(\ell_0 - \ell) & \text{if } \ell < \ell_0, \\ 0 & \text{if } \ell \geq \ell_0. \end{cases}$$

Get expanding mesh by choosing ℓ_0 larger than desired length h

Reactions at Boundaries



For each *boundary* node:

$$\sum_i \boldsymbol{F}_i + \boldsymbol{R} = 0$$

Reaction force $\boldsymbol{R}$:

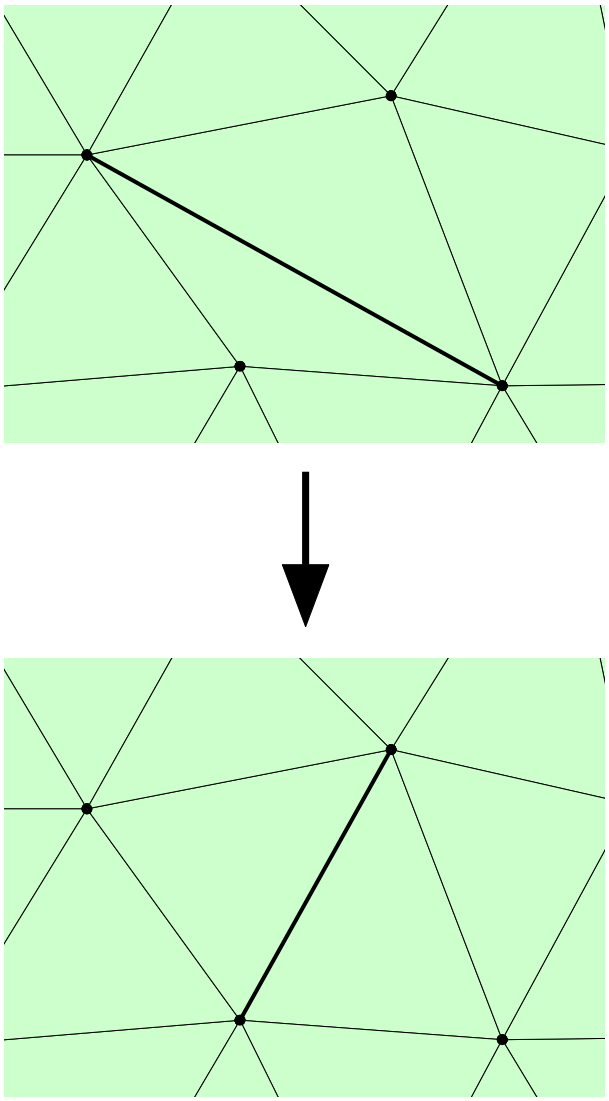
- Orthogonal to boundary
- Keeps node along boundary

Node Movement and Connectivity Updates

- Move nodes $\boldsymbol{p}$ to find force equilibrium:

$$\boldsymbol{p}_{n+1} = \boldsymbol{p}_n + \Delta t \boldsymbol{F}(\boldsymbol{p}_n)$$

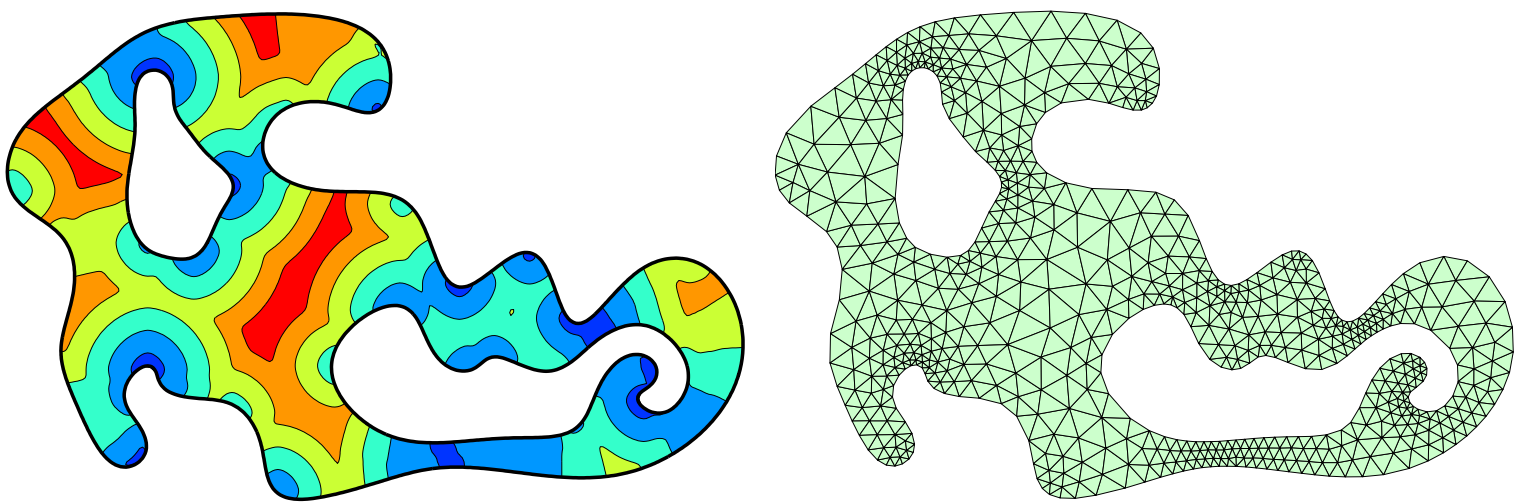
- Project boundary nodes to $\phi(\boldsymbol{p}) = 0$
- Elements deform, change connectivity based on element quality or in-circle condition (Delaunay)



Mesh Size Functions

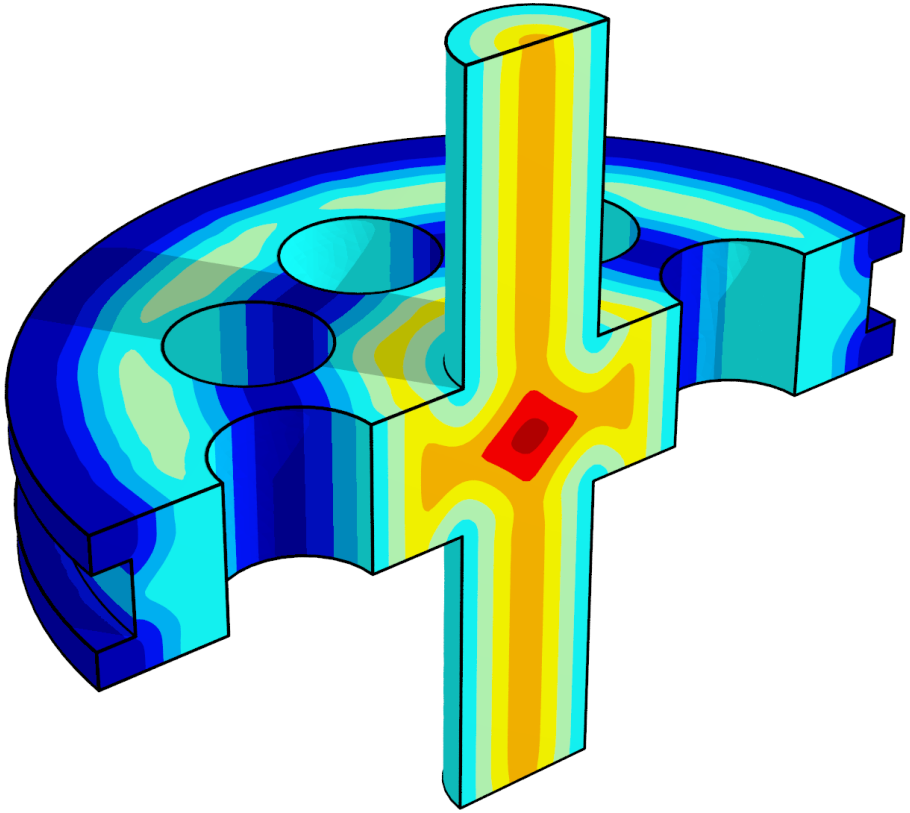
- Need good distribution of mesh element sizes $h(\boldsymbol{x})$ based on curvatures, feature sizes, and external size constraints
- Also: $|\nabla h(\boldsymbol{x})| \leq g$ to limit ratio $G = g + 1$ of neighboring element sizes
- Optimal *gradient limited* $h(\boldsymbol{x})$ steady-state solution to: (Persson 2005)

$$\frac{\partial h}{\partial t} + |\nabla h| = \min(|\nabla h|, g),$$
$$h(t = 0, \boldsymbol{x}) = h_0(\boldsymbol{x}).$$

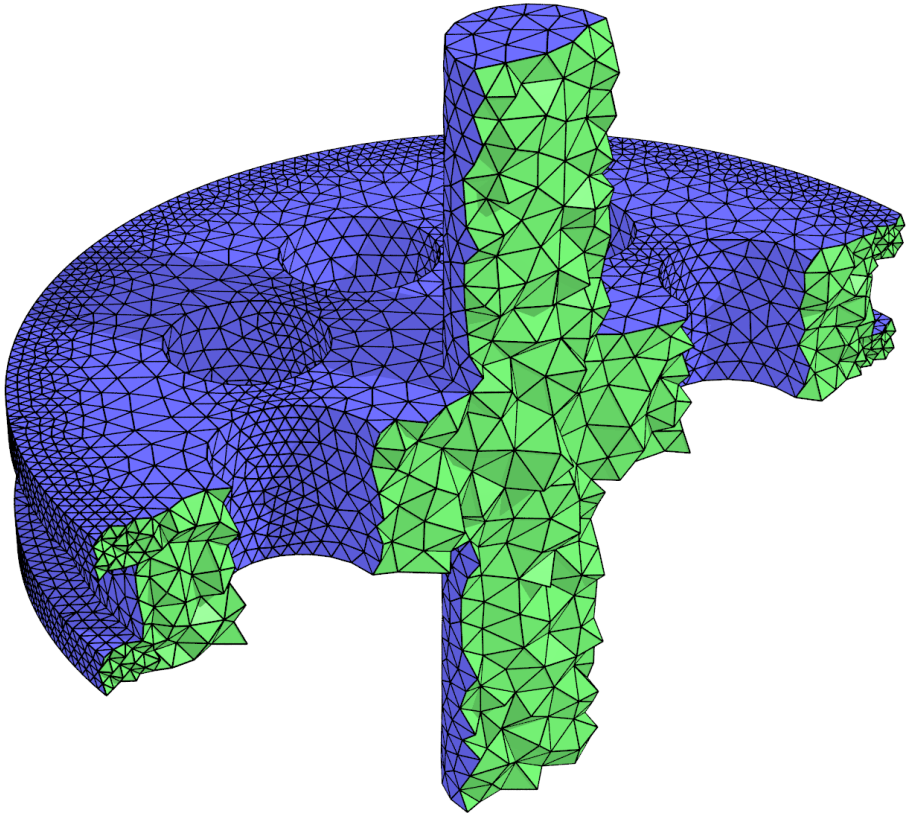


Mesh Size Functions – 3-D Examples

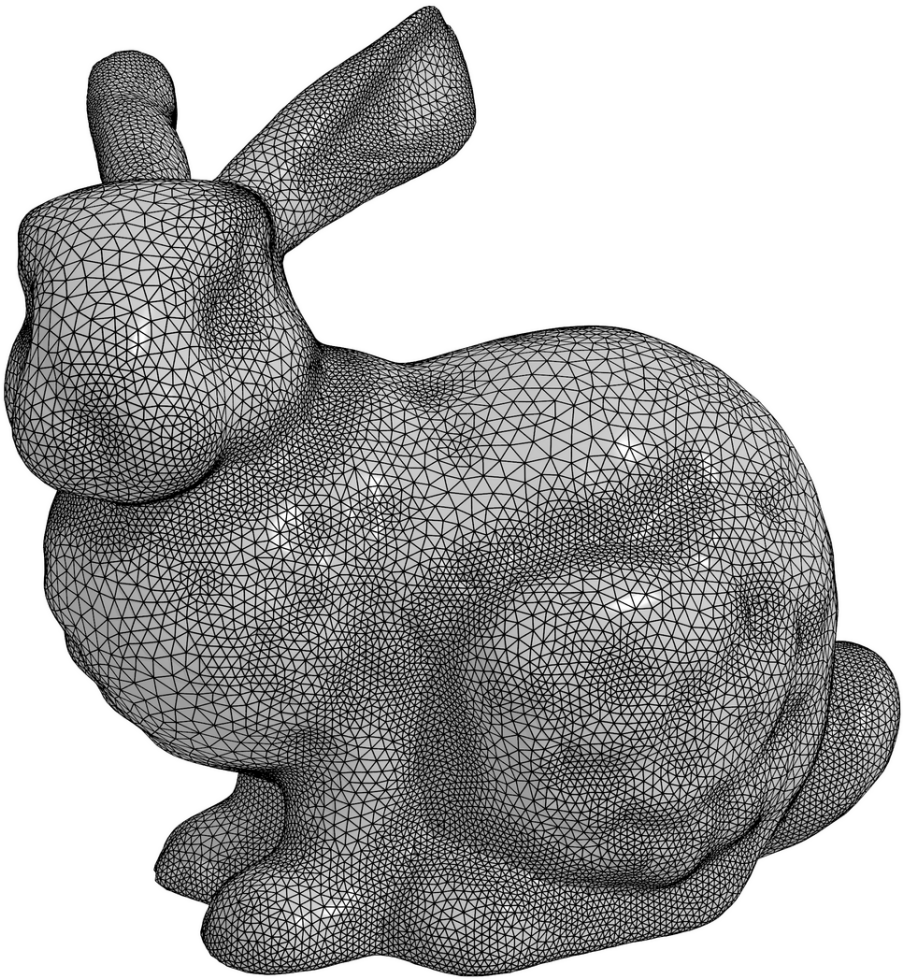
Mesh Size Function $h(\boldsymbol{x})$

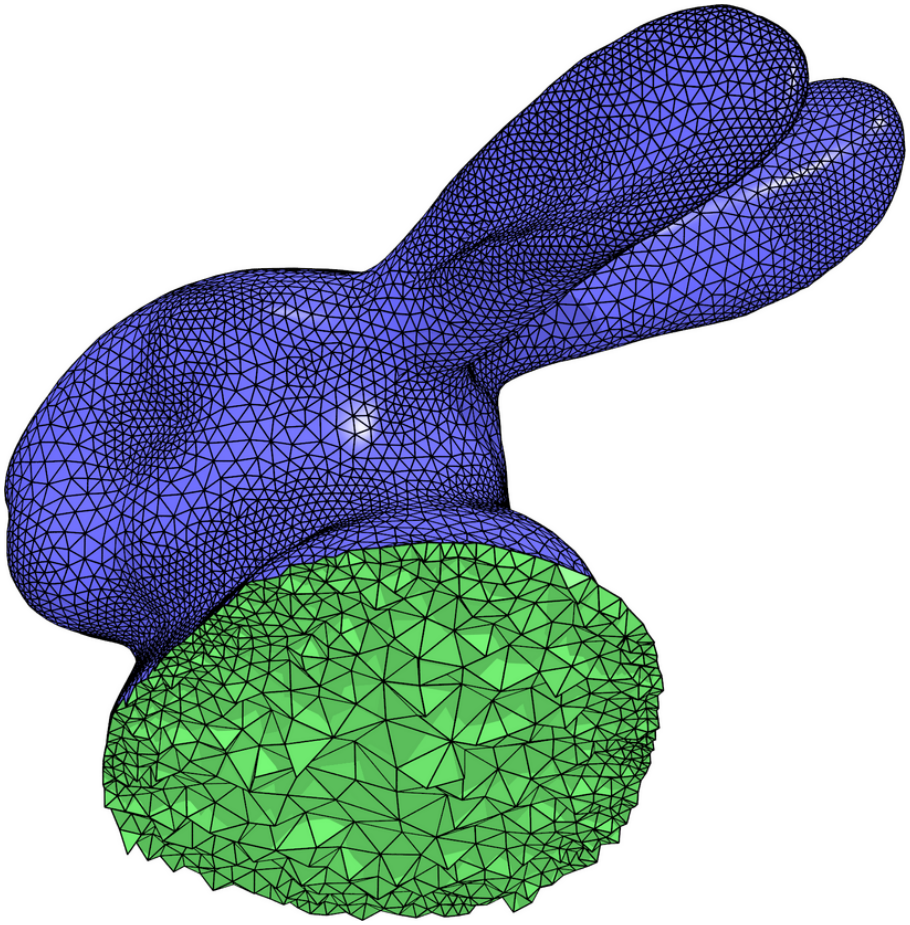


Mesh Based on $h(\boldsymbol{x})$



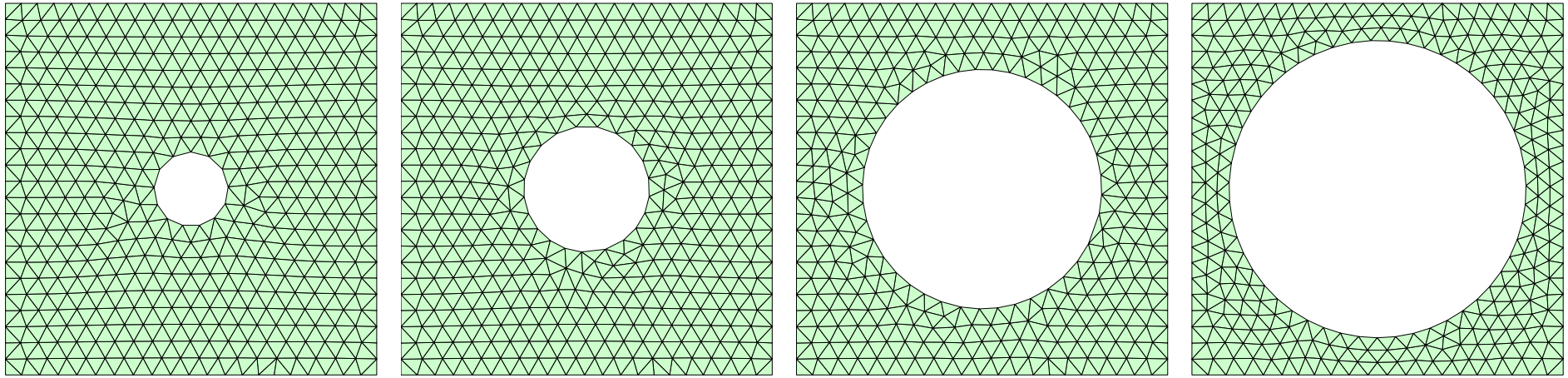
Mesh Size Functions – 3-D Examples





Moving Meshes

- Iterative formulation well-suited for geometries with moving boundaries
- Mesh from previous time step good initial condition, a new mesh obtained by a few additional iterations
- Even better if smooth embedding of boundary velocity – move all nodes
- Density control if area changes



Level Sets and Finite Elements

- Level Set Methods superior for interface propagation:
  - Numerical stability and entropy satisfying solutions
  - Topology changes easily handled, in particular in three dimensions
- Unstructured meshes better for the physical problems:
  - Better handling of boundary conditions
  - Natural mesh grading and adaptation
- Proposal: Combine Level Sets and Finite Elements (Persson 2005)
  - Use level sets on background grid for interface
  - Mesh using our iterations, and solve physical problem using FEM
  - Interpolate boundary velocity to background grid

Structural Vibration Control

- Consider eigenvalue problem

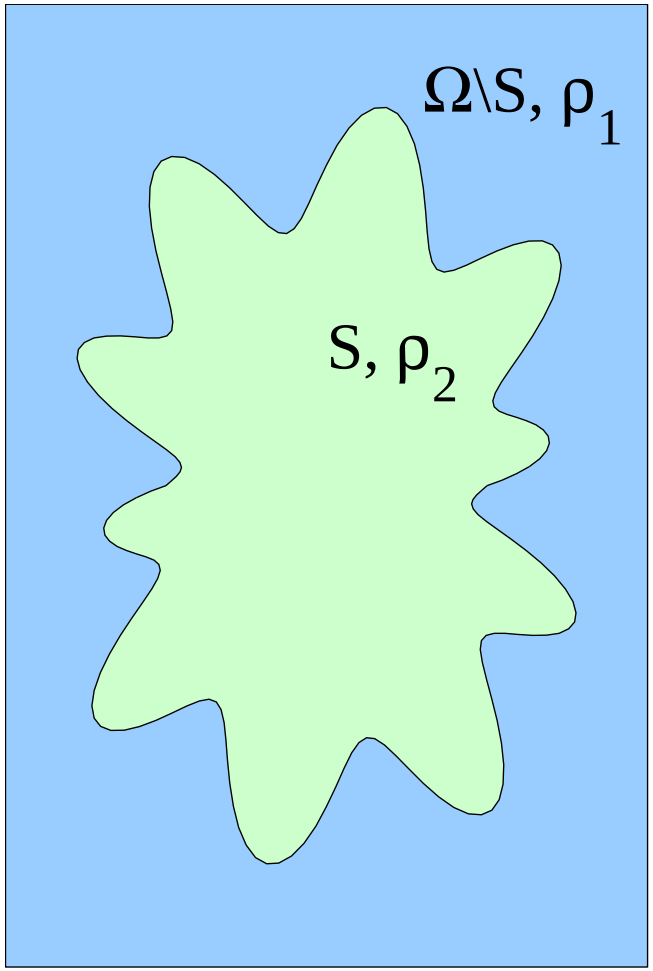
$$
\begin{aligned}
-\Delta u &= \lambda \rho(\boldsymbol{x}) u, && x \in \Omega \\
u &= 0, && x \in \partial\Omega.
\end{aligned}
$$

with

$$
\rho(\boldsymbol{x}) = \begin{cases} \rho_1 & \text{for } x \notin S \\ \rho_2 & \text{for } x \in S. \end{cases}
$$

- Solve the optimization

$$
\min_S \lambda_1 \text{ or } \lambda_2 \text{ subject to } \|S\| = K.
$$



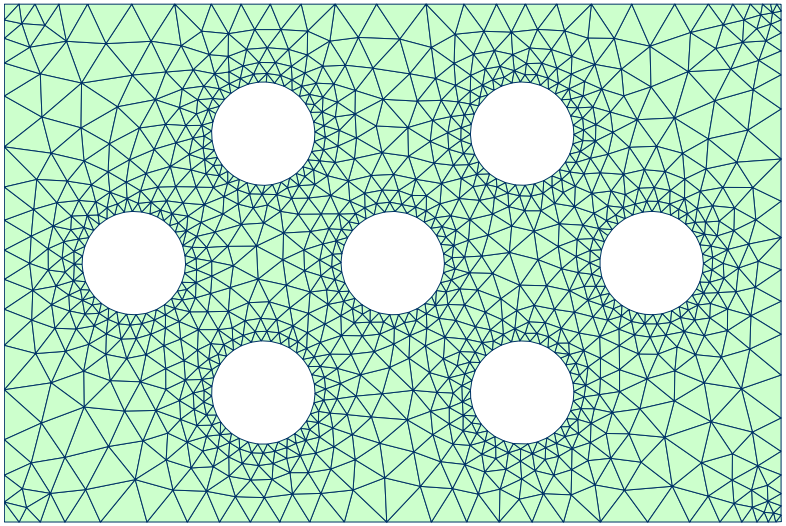
Structural Design Improvement

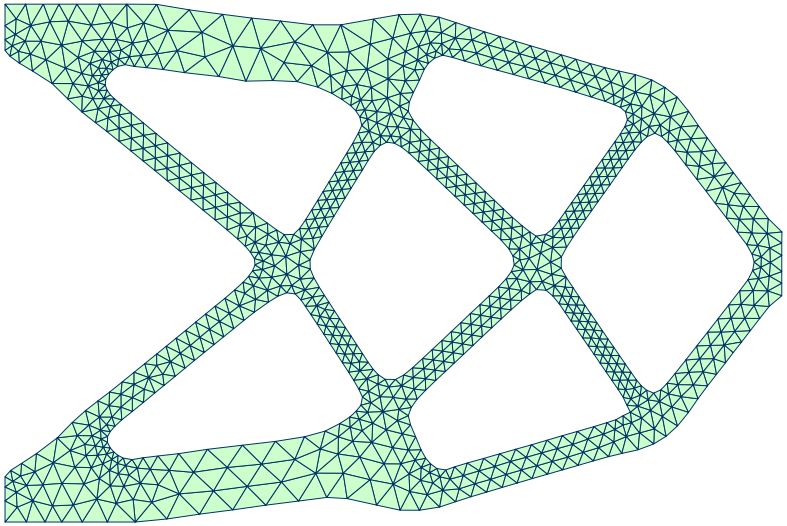
- Linear elastostatics, minimize compliance

$$\int_{\partial\Omega} g \cdot u \, ds = \int_{\Omega} A\epsilon(u) \cdot \epsilon(u) \, dx \text{ subject to } \|\Omega\| = K.$$

- Level set approaches by Sethian and Wiegmann (Immersed interface method) and Allaire et al (*Ersatz material*, Young's modulus ε)

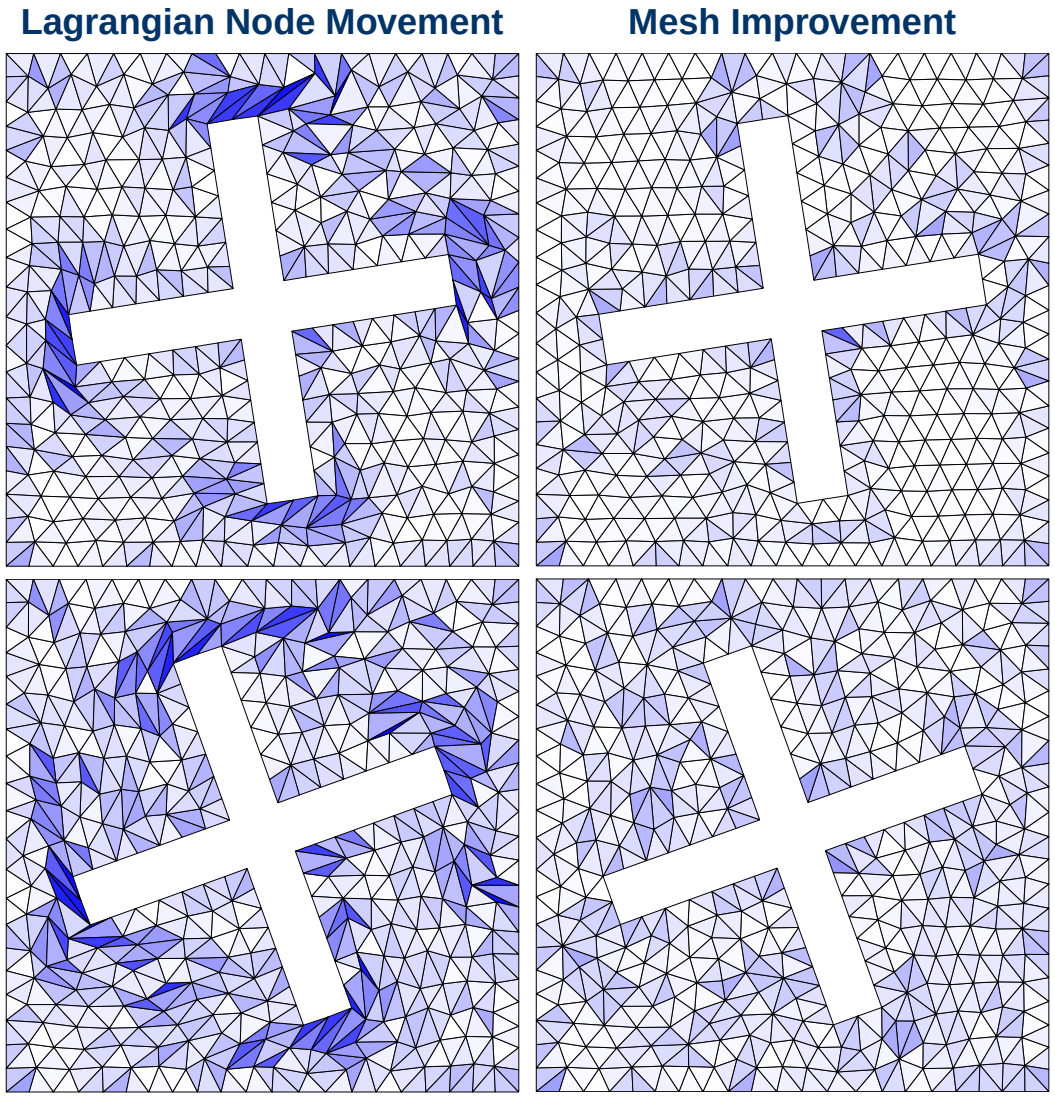
- Level sets for geometry, unstructured FEM for elasticity (Persson 2005):





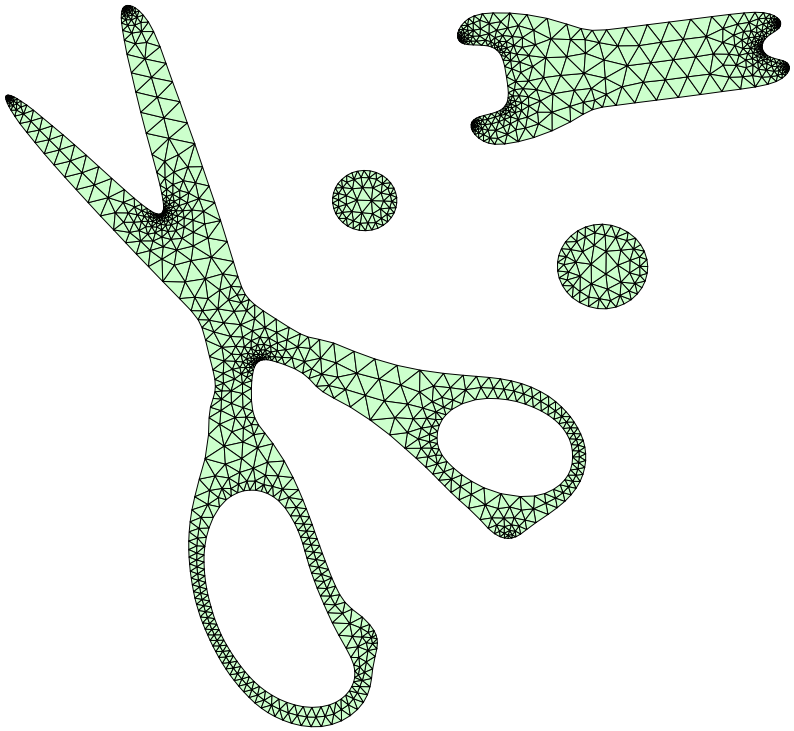
Fluid Dynamics with Moving Boundaries

- Incompressible N-S with moving boundaries
- Lagrangian node movement for convective term
- Improve poor mesh elements using new force equilibrium
- Interpolate new velocity field (or L_2-projections, ALE, Space-Time, etc)
- Project for incompressibility



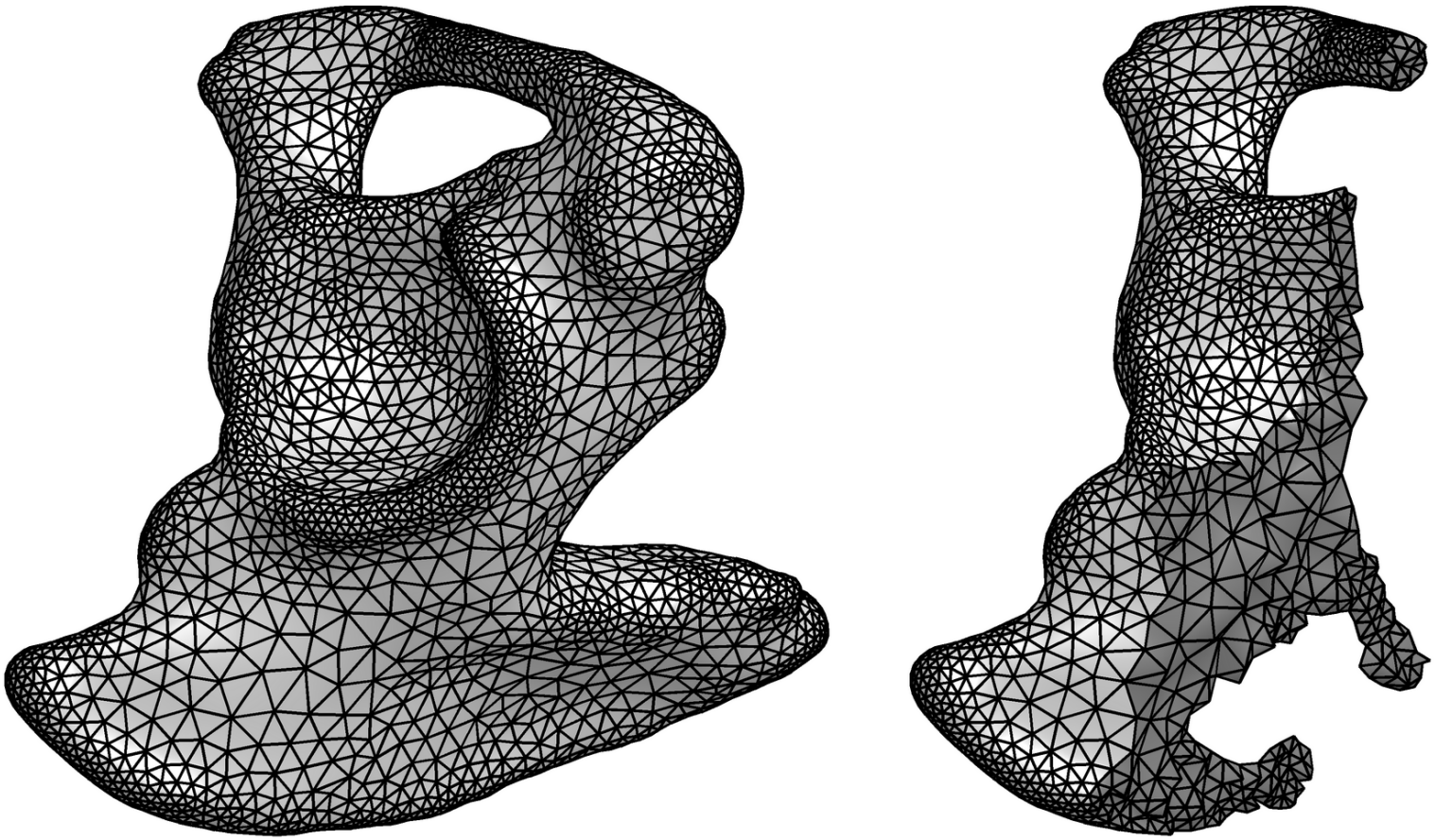
Meshing Images

- Images are implicit representations of objects
- Segment image, find distance function/size function, and create mesh



Meshing Images - Medical Applications

- Method extends to three dimensions (MRI/CT scans)
- Tetrahedral volume mesh of Iliac bone (from CT scan)



Summary Implicit Mesh Generation

- New iterative mesh generator for implicit geometries
- Simple algorithm, high element qualities, works in any dimension
- Automatic computation of size functions, PDE-based gradient limiting
- Combine FEM and level set methods for moving interface problems in shape optimization, structural instabilities, fluid dynamics, etc
- Mesh generation for images (in any dimension)
- Future work: Space-time meshes, sliver removal, performance and robustness improvements, quadrilateral/hexahedral meshes, anisotropic gradient limiting, more applications, . . .