Mesh adaptation (*r* & *p*) for compressible flows in Nektar++

Joaquim Peiro

with Dave Moxey and Dirk Ekelschot

Nektar++ Workshop 2016

What is mesh adaptation?

Continuous PDE: $\mathcal{L}(u) = 0$

Discrete PDE: $\mathcal{L}_h(u_h) = 0$

$$u = u_h + e$$

Polynomial order

Solution smoothness

Error estimate

$$e \approx \boxed{e_h} \propto \boxed{h^p}\,\boxed{\frac{d^p u_h}{dx^p}}$$

Numerical resolution

Mesh size

Adaptation strategies

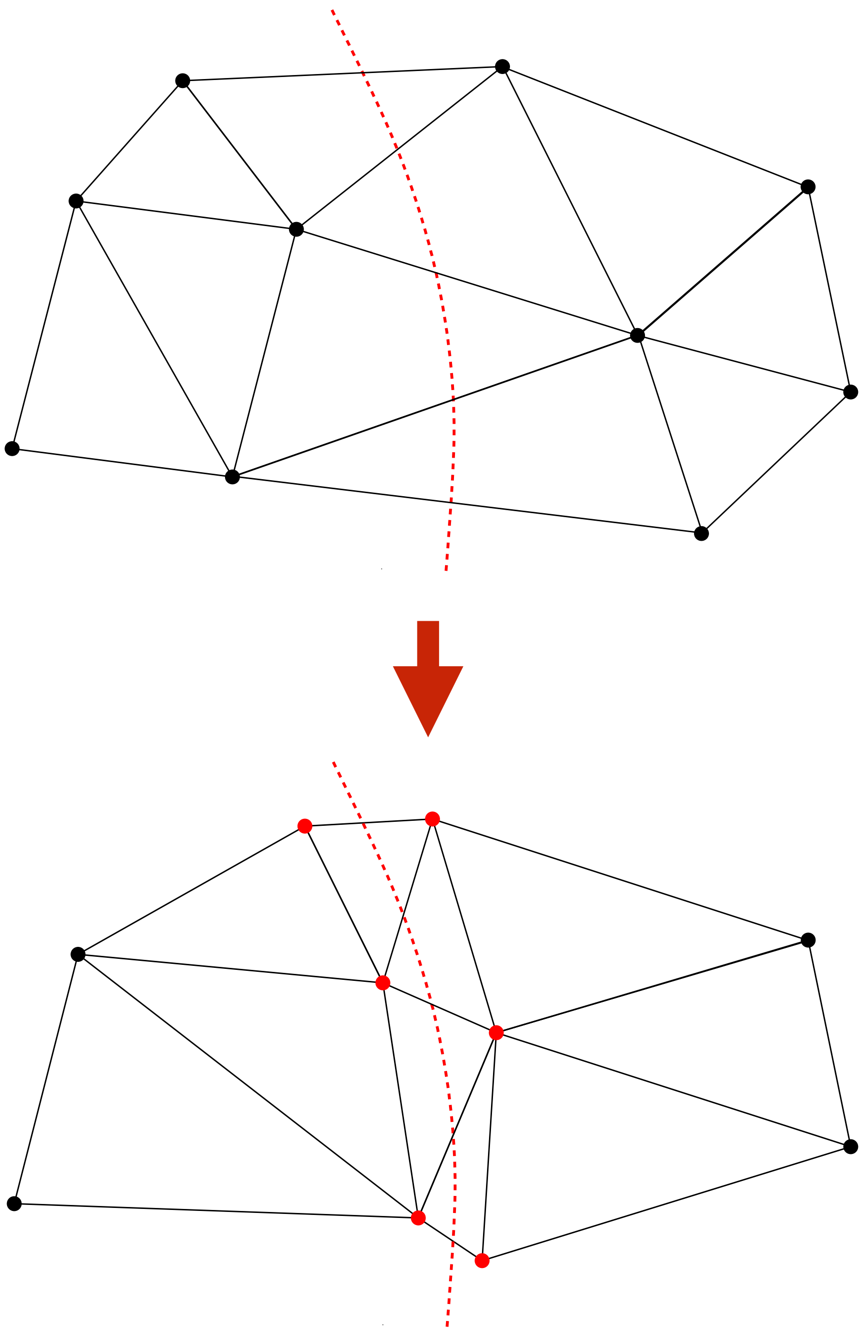
Aim: to reduce the error through numerical resolution

What do we need?

1. A computed solution u_h
2. An *a posteriori* error estimator $e_h(u_h, h, p)$
    a. Interpolation theory
    b. Adjoint
3. A method for increasing resolution (h and/or p)

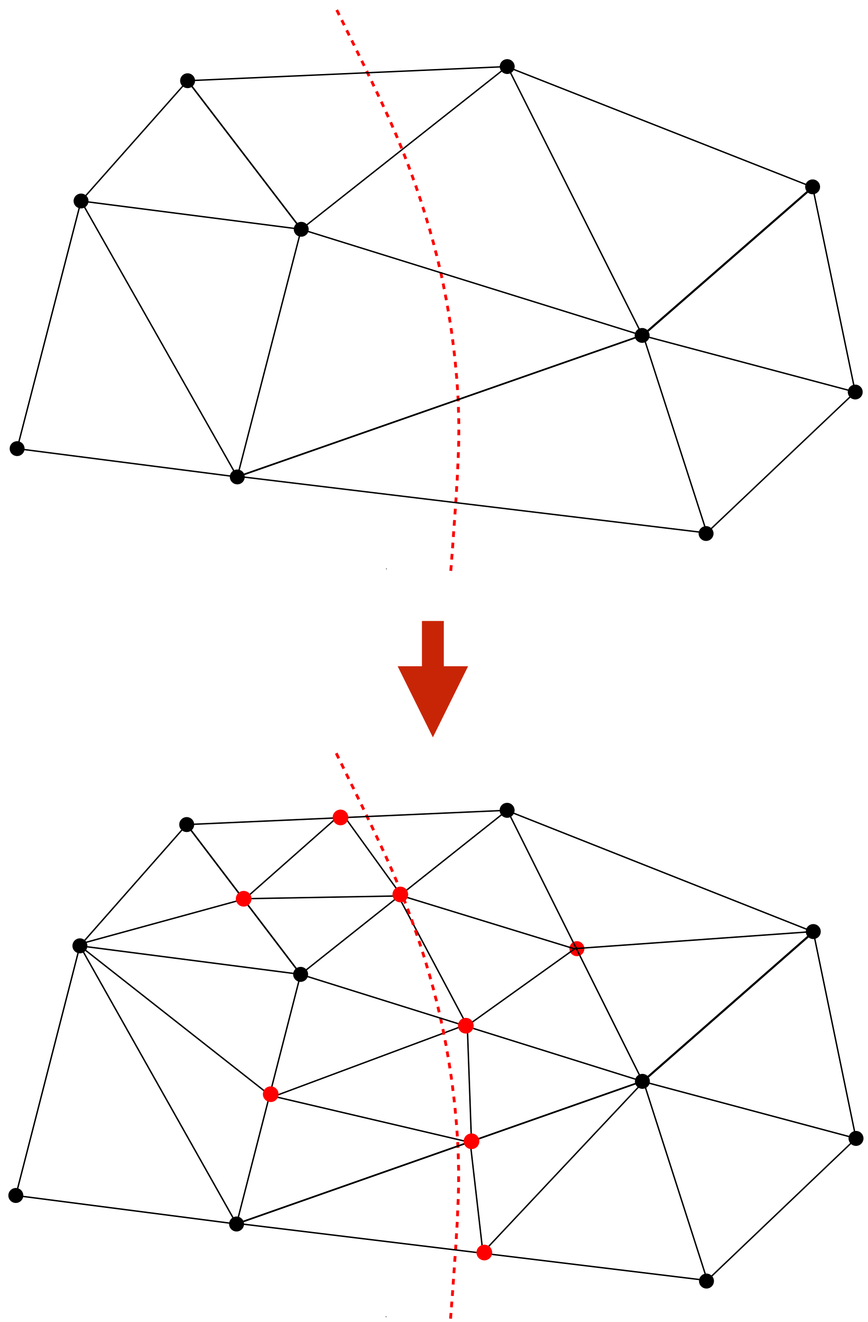
How to increase mesh resolution?

- Smaller h:
  - $\boxed{r\text{-adaptation}}$
  - h-adaptation
- Larger p:
  - p-adaptation



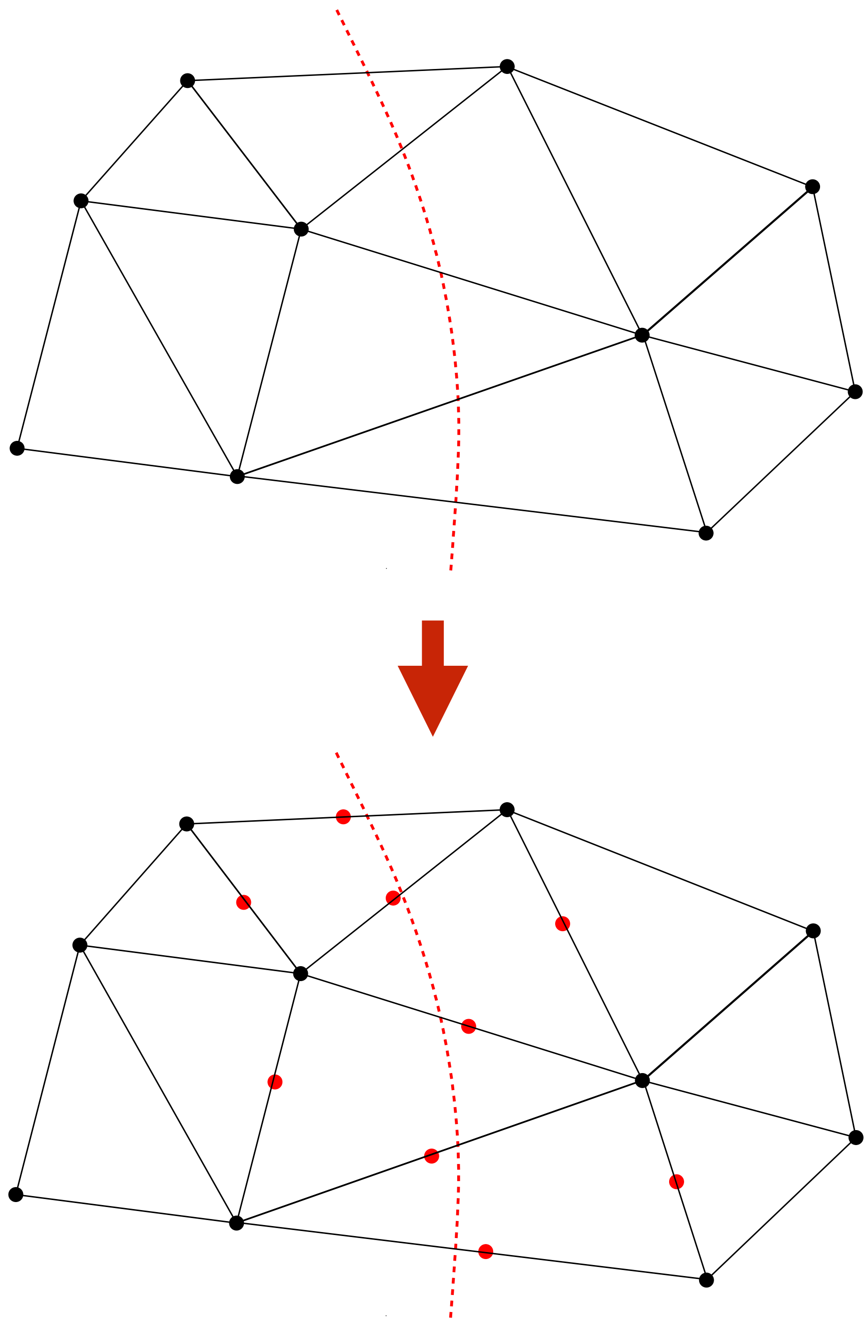
How to increase mesh resolution?

- Smaller h:
  - r-adaptation
  - h-adaptation
- Larger p:
  - p-adaptation

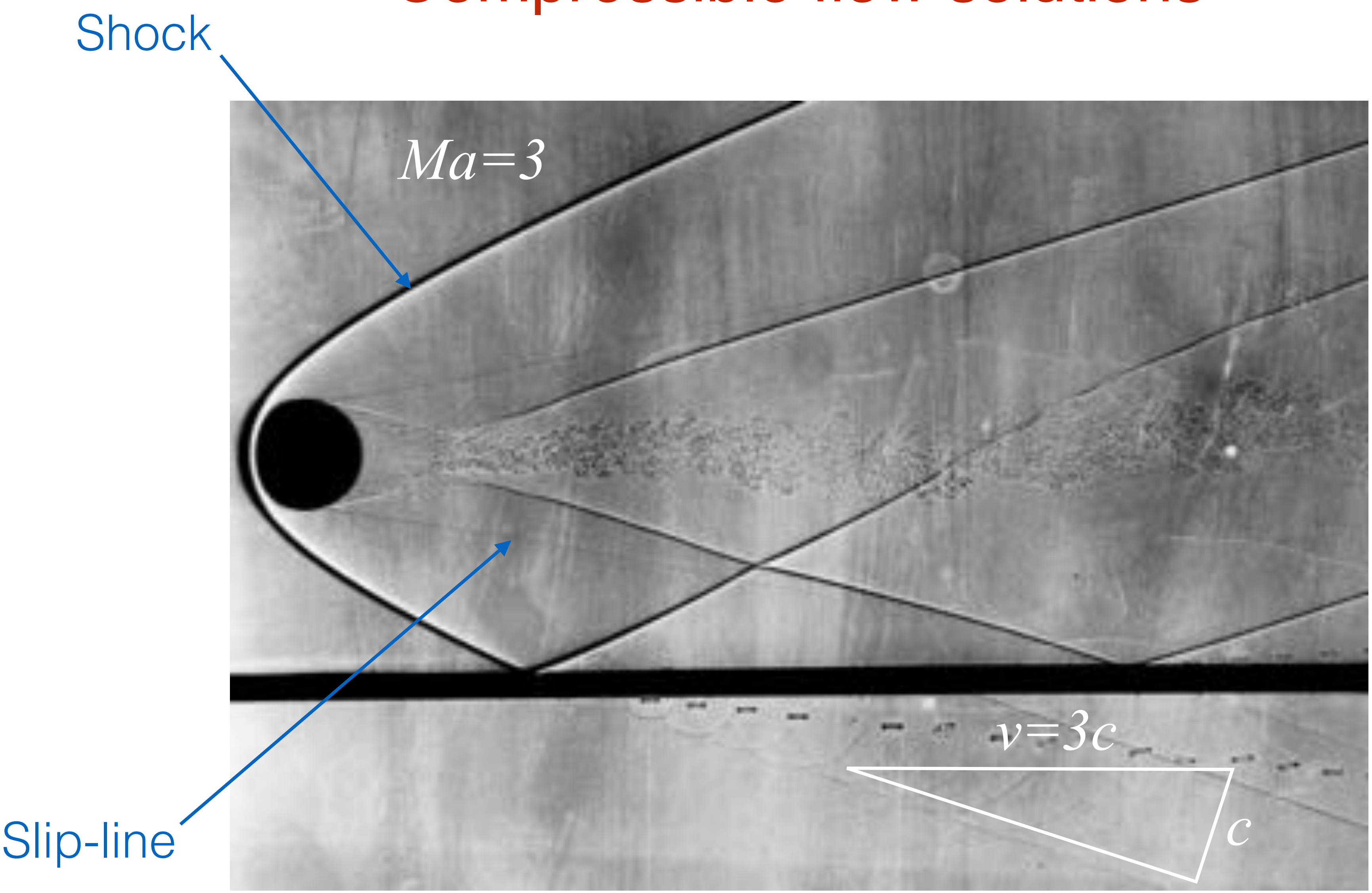


How to increase mesh resolution?

- Smaller h:
  - r-adaptation
  - h-adaptation
- Larger p:
  - p-adaptation



Compressible flow solutions



Regions of smooth flow separated by discontinuities

Compressible flow solver

Governing equations

$$\boldsymbol{R}(\boldsymbol{u}, \nabla \boldsymbol{u}) = \sum_{i=1}^{d} \frac{\partial}{\partial x_i} \left\{ \boldsymbol{f}_i^c(\boldsymbol{u}) - \boldsymbol{f}_i^v(\boldsymbol{u}, \nabla \boldsymbol{u}) \right\} = \boldsymbol{0}; \quad \boldsymbol{u} \in \Omega$$

$d = 2$ or 3 dimensions

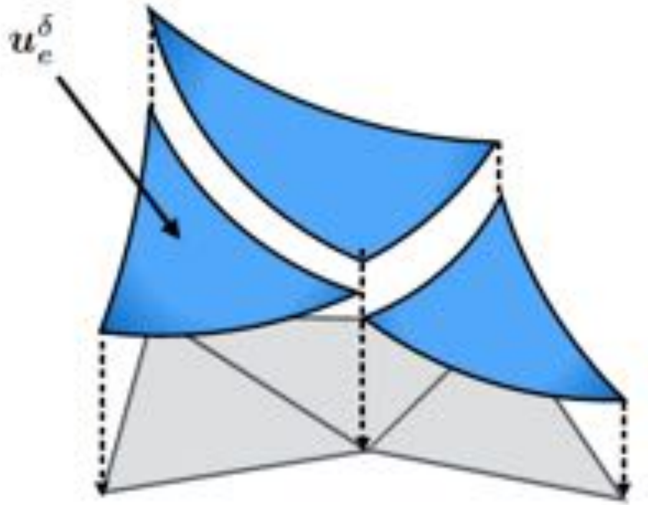
Mixed formulation

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

$$\sum_{i=1}^{d} \frac{\partial}{\partial x_i} \left\{ \boldsymbol{f}_i^c(\boldsymbol{u}) - \boldsymbol{f}_i^v(\boldsymbol{u}, \boldsymbol{g}) \right\} = \boldsymbol{0}$$

Discontinuous Galerkin discretization

$$\boldsymbol{u}_e^\delta = \phi_e \bar{\boldsymbol{u}}_e = \sum_{p=0}^{P} \sum_{q=0}^{Q} \phi_{pq}(\xi_1, \xi_2) \bar{\boldsymbol{u}}_{pq}$$



Discrete equations

$$\sum_{e=1}^{N_{el}} \left\{ \int_{\Omega_e} \phi^e \boldsymbol{g}_e^\delta \, d\Omega - \int_{\Gamma_e} \phi^e \boldsymbol{u}_e^\delta \vec{n} \, d\Gamma + \int_{\Omega_e} \nabla \phi^e \boldsymbol{u}_e^\delta \, d\Omega \right\} = \mathbf{0}$$

$$- \sum_{e=1}^{N_{el}} \int_{\Omega_e} \sum_{i=1}^{d} \frac{\partial \phi^e}{\partial x_i} \left\{ \boldsymbol{f}_i^c(\boldsymbol{u}_e^\delta) - \boldsymbol{f}_i^v(\boldsymbol{u}_e^\delta, \boldsymbol{g}_e^\delta) \right\} d\Omega$$

$$+ \sum_{e=1}^{N_{el}} \int_{\Gamma_e} \phi^e \boxed{\left[\boldsymbol{f}_n^c(\boldsymbol{u}_e^\delta) - \boldsymbol{f}_n^v(\boldsymbol{u}_e^\delta, \boldsymbol{g}_e^\delta) \right]} d\Gamma = \mathbf{0}$$

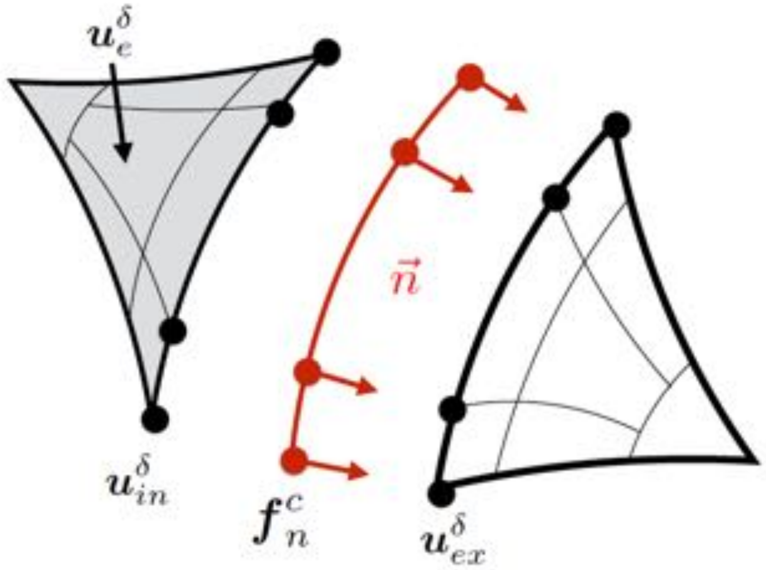
Interface fluxes

Treatment of interface fluxes

Convective flux

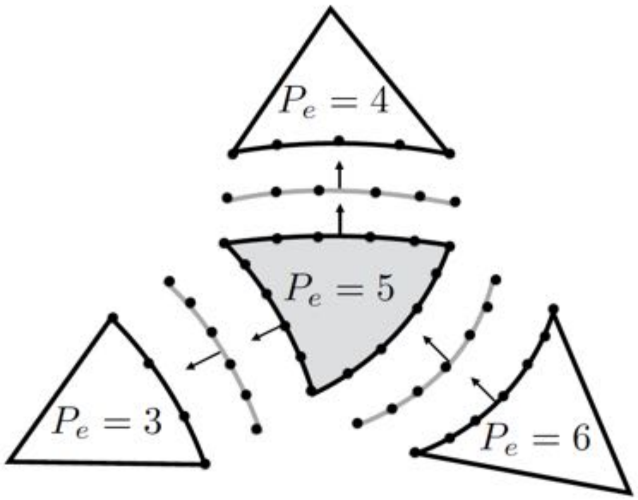
$$\boldsymbol{f}_n^c(\boldsymbol{u}_e^\delta) \approx \mathcal{H}^c(\boldsymbol{u}_{ex}^\delta, \boldsymbol{u}_{in}^\delta; \vec{n})$$

$\mathcal{H}^c$: exact or approximate Riemann solver



Viscous flux (LDG)

$$\boldsymbol{f}_n^v(\boldsymbol{u}_e^\delta, \boldsymbol{g}_e^\delta)\big|_{\Gamma_e} = \sum_{i=1}^{d} \boldsymbol{g}_{i,in} n_i$$



Variable p

Ensure flux continuity

$$\int_{\Gamma_{f_r}} \boldsymbol{f}(\boldsymbol{u}_{ex}^\delta) d\Gamma = \int_{\Gamma_{f_l}} \boldsymbol{f}(\boldsymbol{u}_{in}^\delta) d\Gamma$$

Shock capturing terms

$$\boldsymbol{f} \rightarrow \boldsymbol{f} \boxed{- \mu_a(s_e)\nabla \boldsymbol{u}}$$

Artificial dissipation

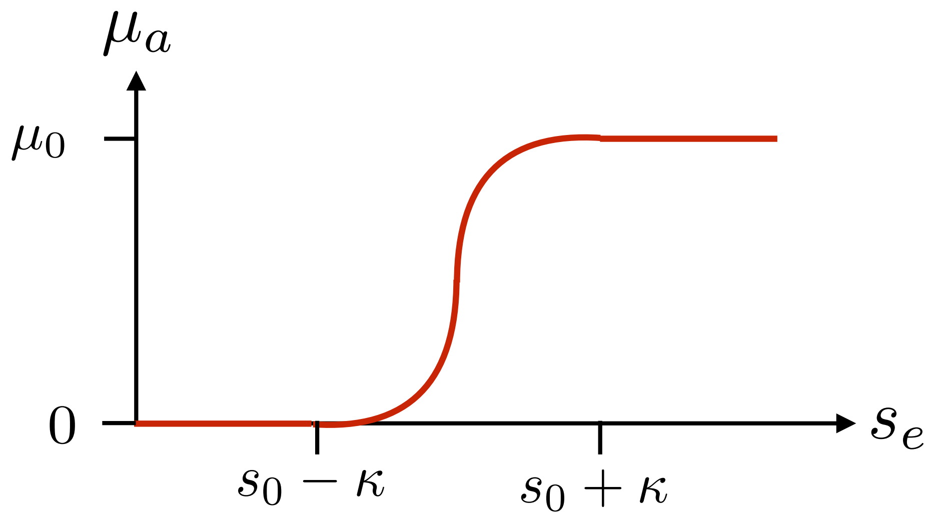
$$s_e = \log_{10}\left(\frac{||\beta(\mathbf{u})_e^p - \beta(\mathbf{u})_e^{p-1}||_{L_2}}{||\beta(\mathbf{u})_e^p||_{L_2}}\right)$$

Shock sensor

Viscosity blending

$$\mu_a = \begin{cases} 0 & \text{if} \quad s_e < s_0 - \kappa \\ \frac{\mu_0}{2}\left(1 + \sin\left(\frac{\pi(s_e - s_0)}{2\kappa}\right)\right) & \text{if} \quad s_0 - \kappa < s_e < s_0 + \kappa \\ \mu_0 & \text{if} \quad s_e > s_0 + \kappa \end{cases}$$

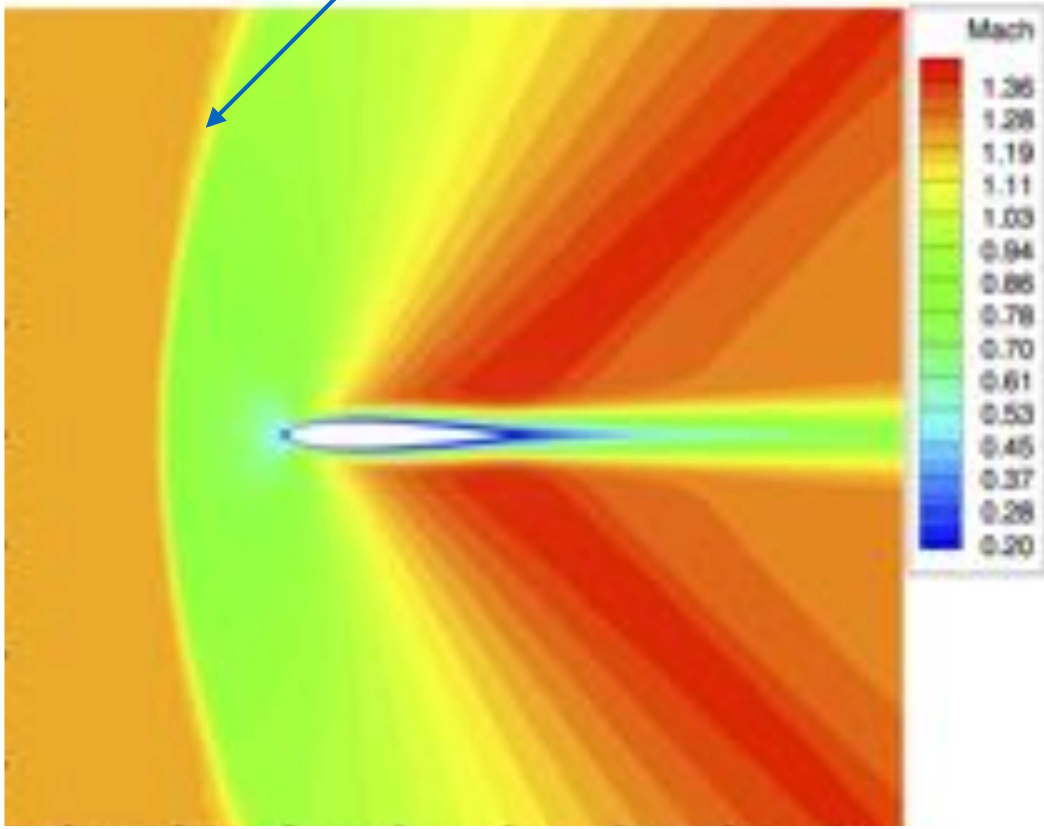
$\mu_0 \sim h/p$ and $s_0 \sim 1/p^4$ and κ define the range of elements that require artificial diffusion.



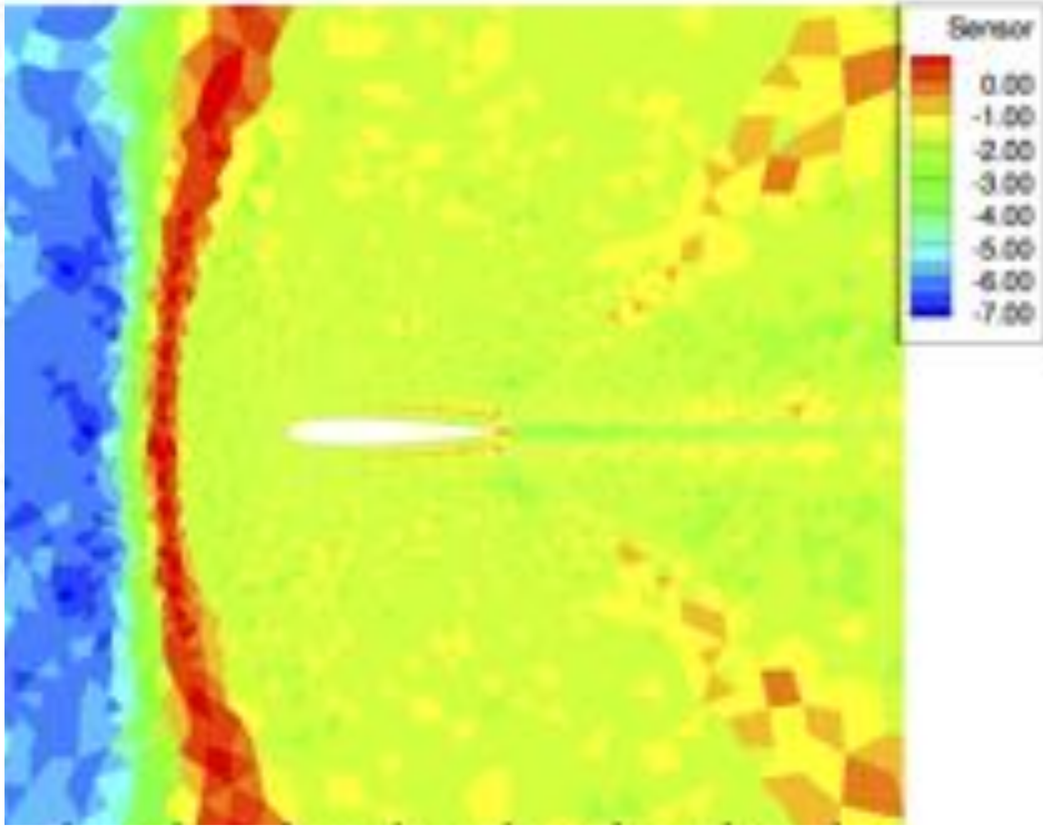
Laminar flow past a NACA0012

$$(\mathrm{Ma} = 1.2;\ \mathrm{Re} = 1\,000;\ \alpha = 0^\circ)$$

Shock



Flow simulation: Mach number

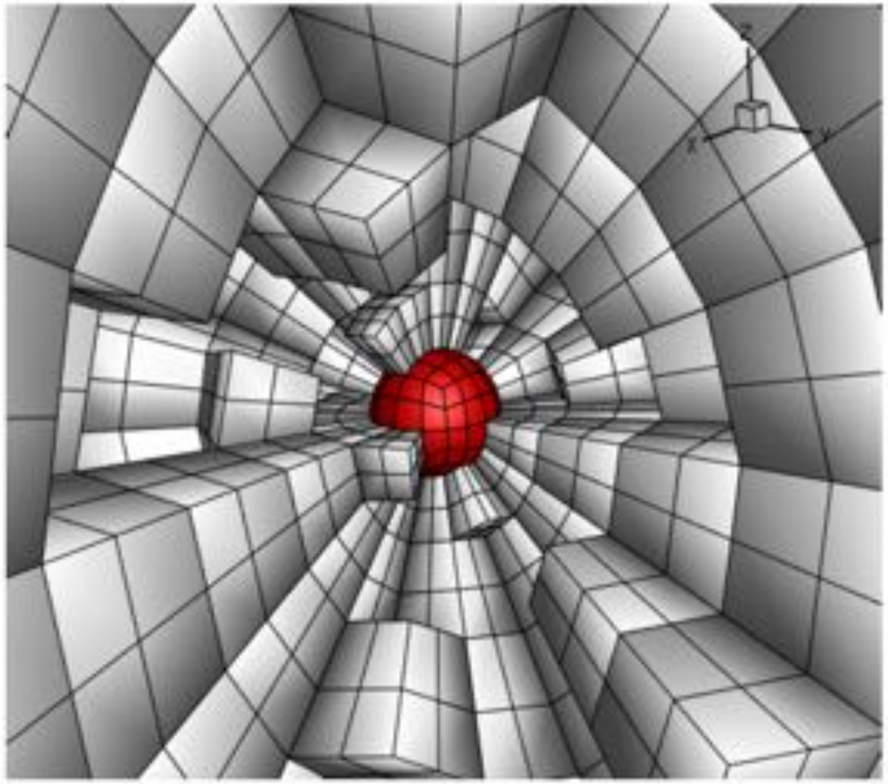


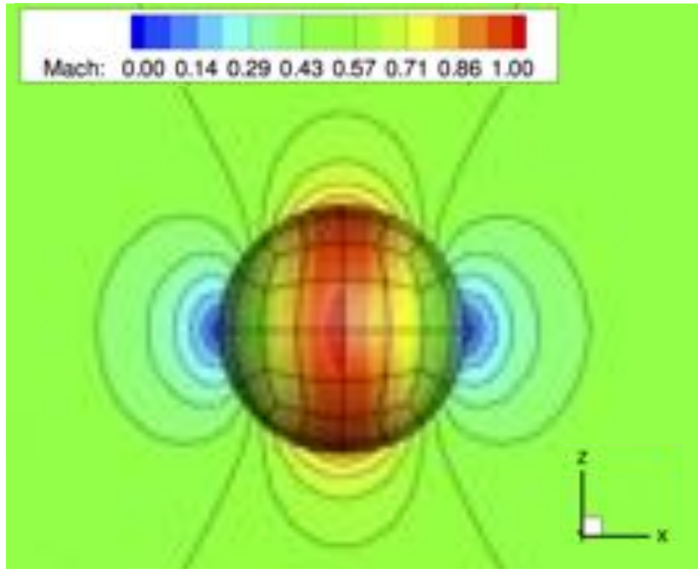
Mach number based sensor

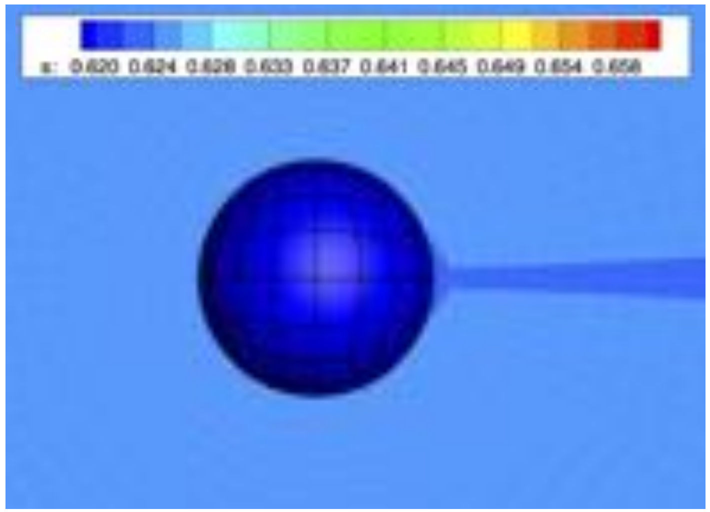
Interpolation-based error estimation

The shock sensor is used as a “smoothness” sensor

$$p_e \longrightarrow \begin{cases} p_e + 3 & \text{if} \quad s_e > s_3 \\ p_e + 2 & \text{if} \quad s_2 < s_e < s_3 \\ p_e + 1 & \text{if} \quad s_1 < s_e < s_2 \\ p_e & \text{if} \quad s_e < s_1 \end{cases}$$

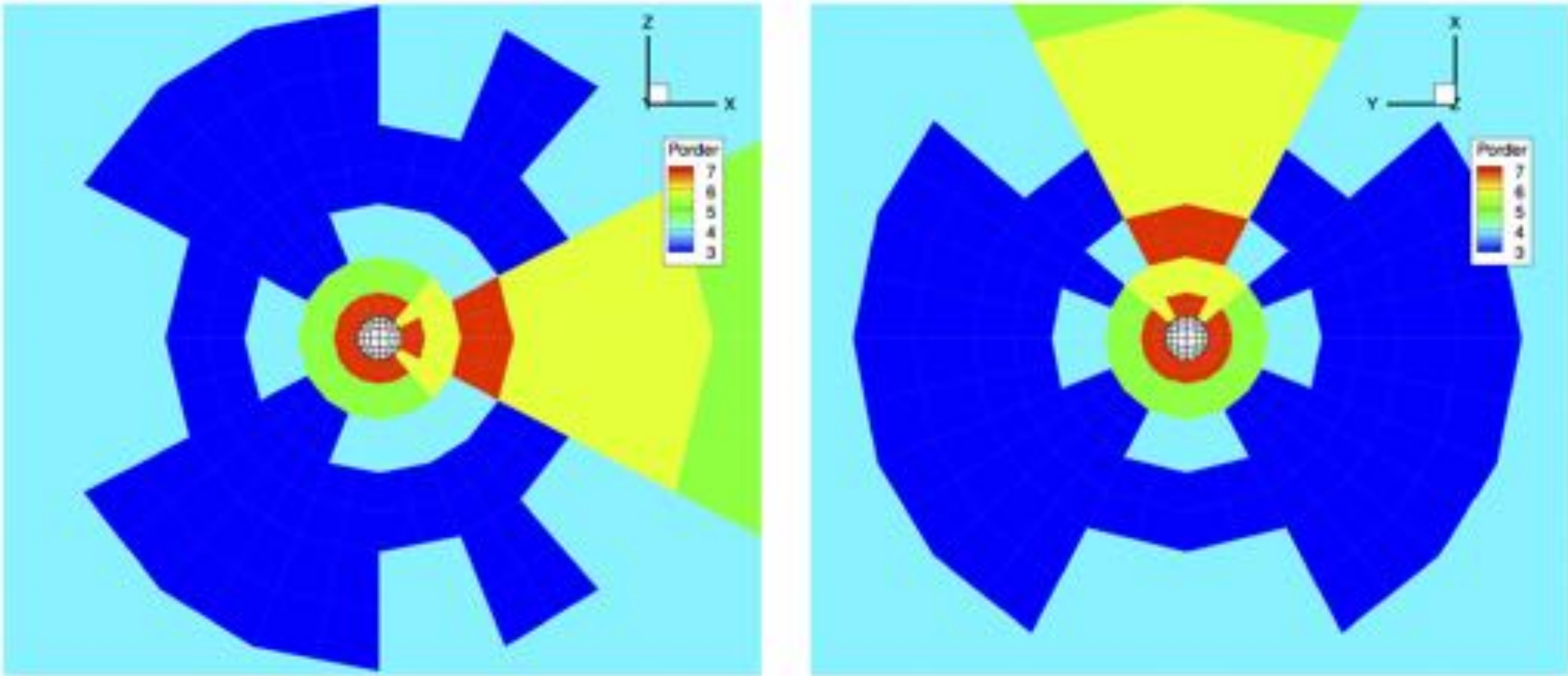






Flow past a sphere Ma=0.5

p-adaptation



Uniform *P*	$P = 2$	$P = 6$	$P = 7$
N_Q	55296	442368	629856
ϵ_{L_2}	2.1×10^{-3}	3.17×10^{-7}	1.10×10^{-7}
Variable *P*	$2 \leq P \leq 6$	$3 \leq P \leq 7$	
N_Q	160588	255700	
ϵ_{L_2}	1.6×10^{-5}	1.15×10^{-6}	

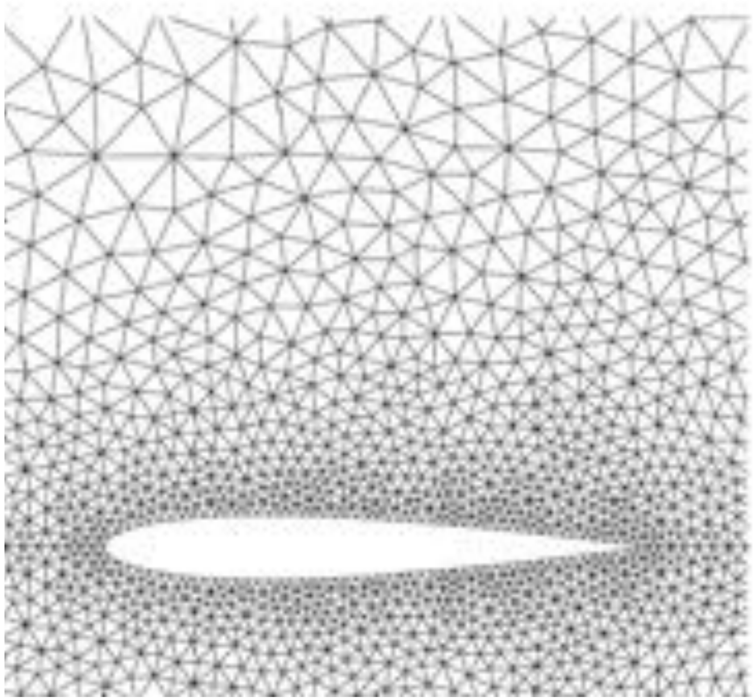
$$\epsilon_{L_2} = \sqrt{\frac{\int_\Omega \frac{p/\rho^\gamma - p_\infty/\rho_\infty^\gamma}{p_\infty/\rho_\infty^\gamma} d\Omega}{\int_\Omega d\Omega}}$$

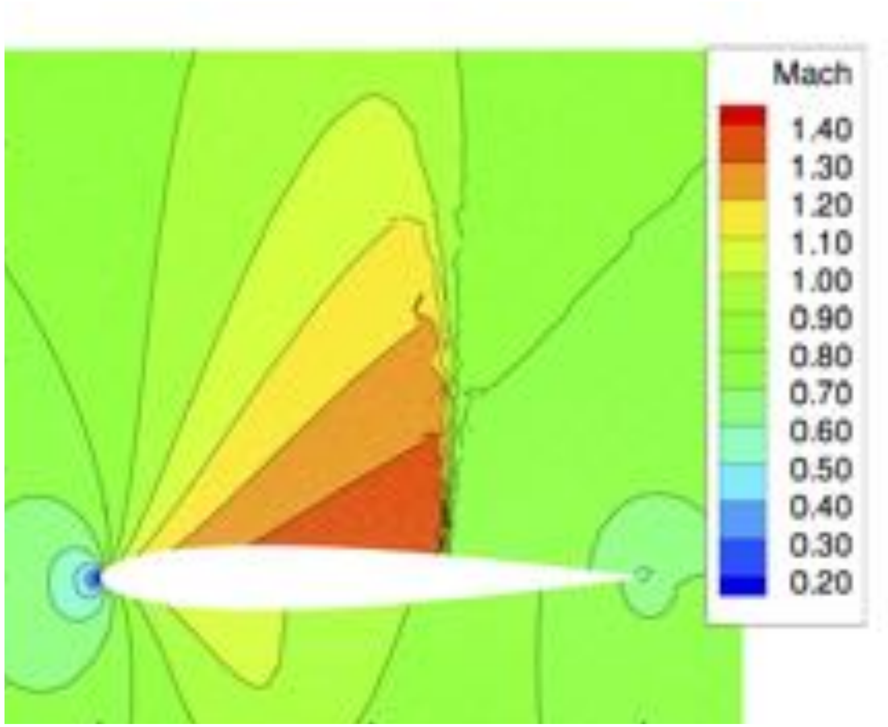
L_2 entropy error

r-adaptation for shocks

High-order degrees of freedom are wasted at shocks

Increase resolution by clustering nodes near shocks





Mesh deformation: thermo-elasticity

$$\nabla \cdot (\underbrace{\mathbf{S}_e}_{} + \underbrace{\mathbf{S}_t}_{}) + \mathbf{f} = 0$$

$\mathbf{q} \rightarrow$ displacement

Thermal stresses

$$\boxed{\mathbf{S}_t = \beta \, \Delta T \, \mathbf{I}}$$

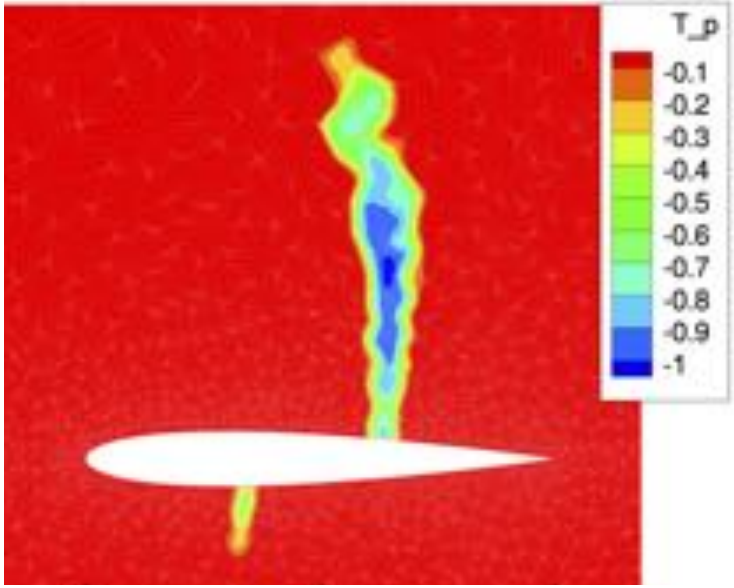
Elastic stresses

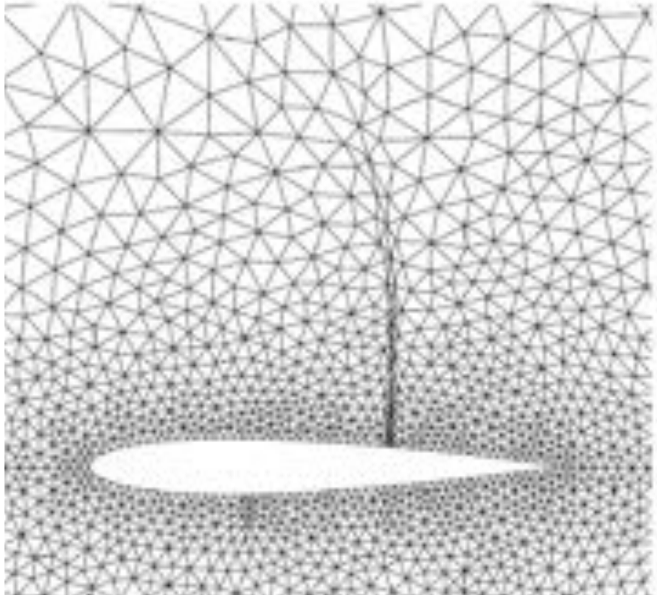
$$\mathbf{S}_e = \lambda \, \mathrm{tr}(\mathbf{E})\mathbf{I} + \mu\mathbf{E}$$

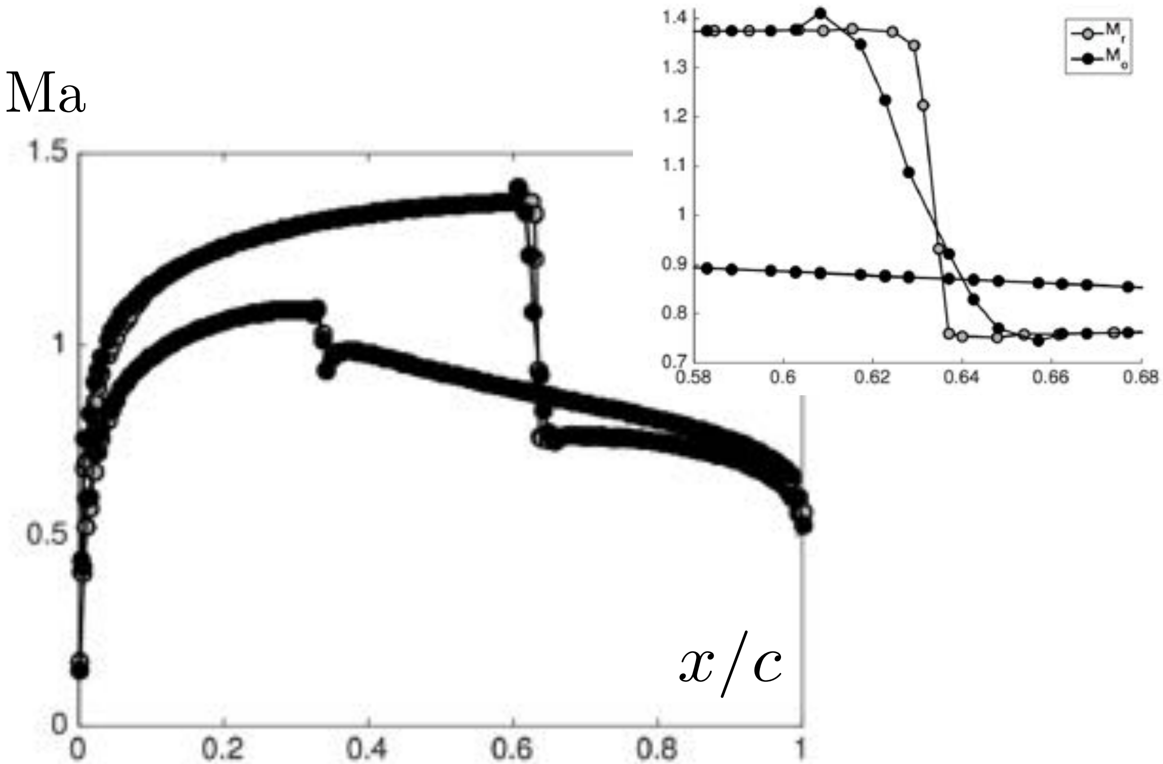
$$\mathbf{E} = \frac{1}{2}\left(\nabla\mathbf{q} + (\nabla\mathbf{q})^t\right)$$

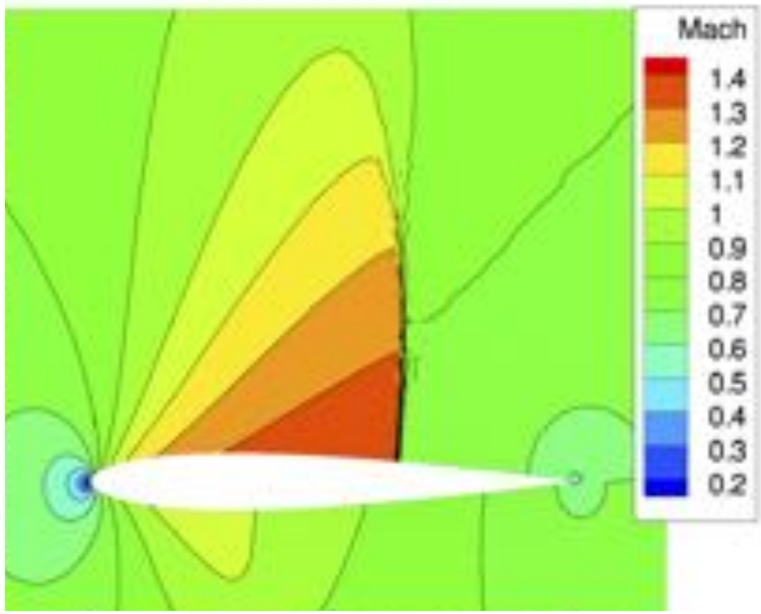
Relate temperature changes to shock sensor to “cool” elements near the shock

Transonic NACA0012

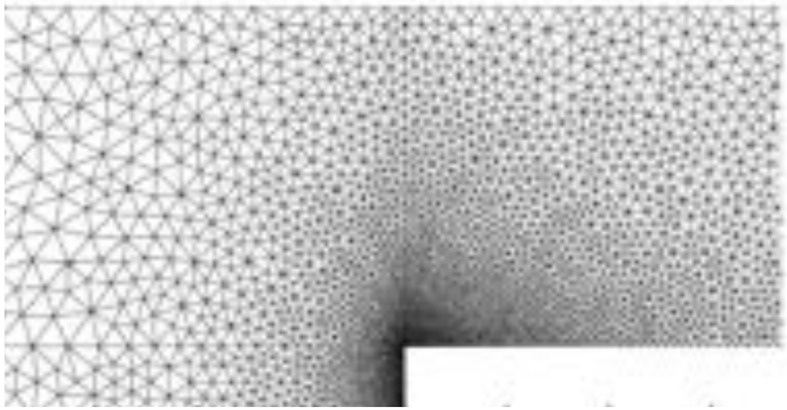


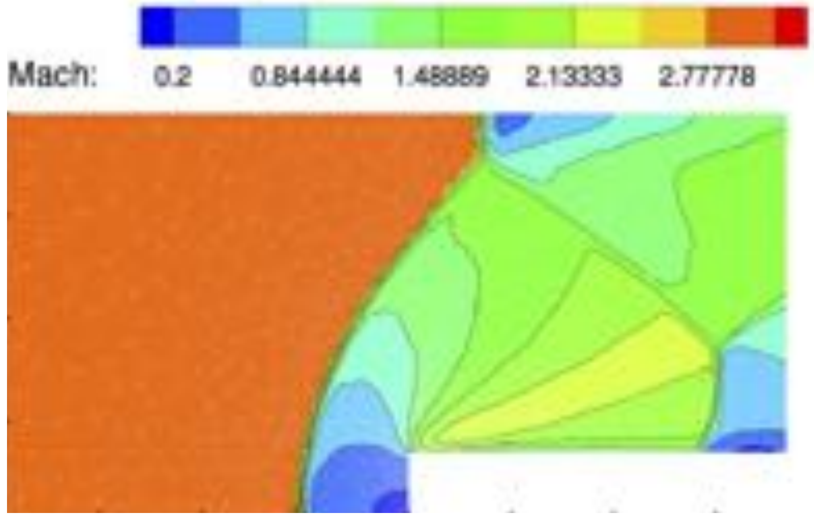


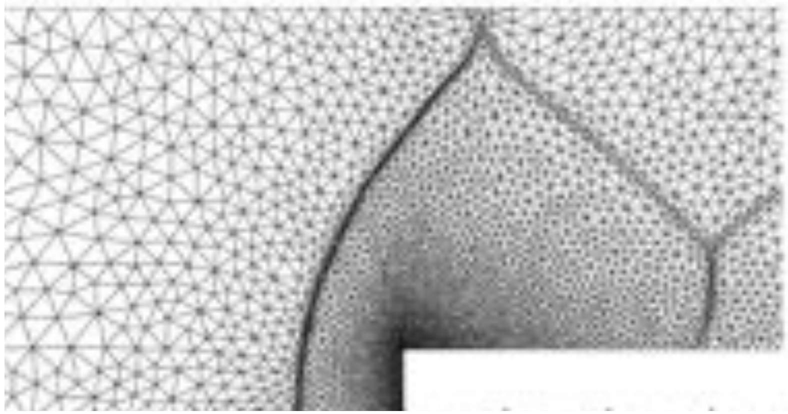


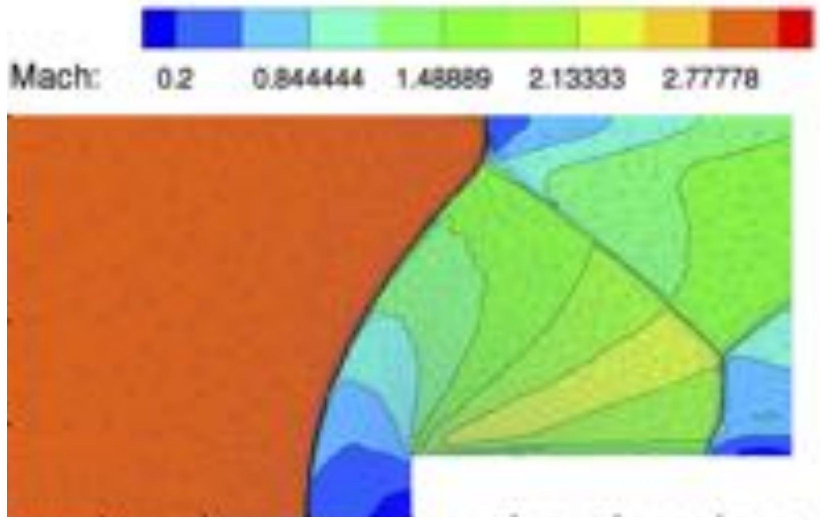


r-adaptation: inviscid flow past a step (Ma=3.0)



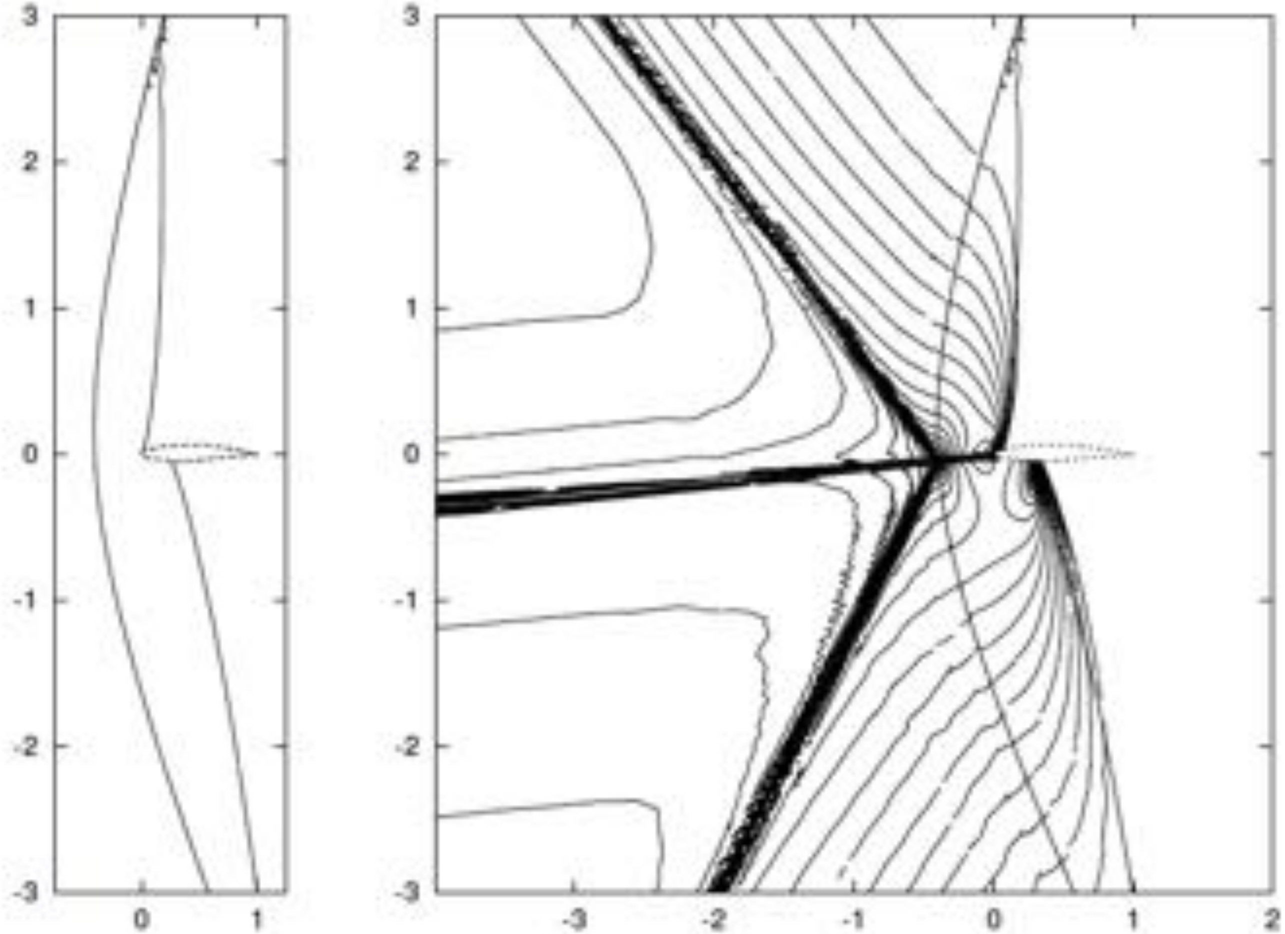






Why adjoint-based error estimation?

$\mathrm{Ma} = 1.2,\ \alpha = 5^\circ$



R. Hartmann and P. Houston. *Adaptive Discontinuous Galerkin Finite Element Methods for the Compressible Euler Equations*. J. Comput. Phys. 183(2):508-532, 2002.

What is an adjoint method?

Consider a function $\mathcal{J}(\boldsymbol{u}, \boldsymbol{p})$ where $\boldsymbol{u}$ is the solution of a non-linear equation $\boxed{\mathcal{R}(\boldsymbol{u}, \boldsymbol{p}) = 0}$ and $\boldsymbol{p}$ is a set of control parameters.

Calculate sensitivity to parameters: $\dfrac{d\mathcal{J}}{d\boldsymbol{p}}(\boxed{\boldsymbol{u}(\boldsymbol{p})}, \boldsymbol{p})$

$$\frac{d\mathcal{J}}{d\boldsymbol{p}} = \frac{\partial \mathcal{J}}{\partial \boldsymbol{u}} \boxed{\frac{d\boldsymbol{u}}{d\boldsymbol{p}}} + \frac{\partial \mathcal{J}}{\partial \boldsymbol{p}}$$

How to evaluate this?

$$\hat{\mathcal{J}} = \mathcal{J} + \psi^t \mathcal{R} \rightarrow \frac{d\hat{\mathcal{J}}}{d\boldsymbol{p}} = \left(\underbrace{\frac{\partial \mathcal{J}}{\partial \boldsymbol{u}} + \psi^t \frac{\partial \mathcal{R}}{\partial \boldsymbol{u}}}_{=0} \right) \frac{d\boldsymbol{u}}{d\boldsymbol{p}} + \frac{\partial \mathcal{J}}{\partial \boldsymbol{p}} + \psi^t \frac{\partial \mathcal{R}}{\partial \boldsymbol{p}}$$

We do not!

Adjoint: $\dfrac{\partial \mathcal{J}}{\partial \boldsymbol{u}} + \psi^t \dfrac{\partial \mathcal{R}}{\partial \boldsymbol{u}} = 0$ Sensitivity: $\dfrac{d\hat{\mathcal{J}}}{d\boldsymbol{p}} = \dfrac{\partial \mathcal{J}}{\partial \boldsymbol{p}} + \psi^t \dfrac{\partial \mathcal{R}}{\partial \boldsymbol{p}}$

Goal-based adaptation

Error in a target functional $J(\boldsymbol{u})$ where $\boldsymbol{R}(\boldsymbol{u}) = \boldsymbol{0}$

$$\epsilon_J = \{J_\delta(\boldsymbol{u}+\delta\boldsymbol{u}) - J(\boldsymbol{u})\}$$

$$= \underbrace{\boxed{\{J_\delta(\boldsymbol{u}+\delta\boldsymbol{u}) - J_\delta(\boldsymbol{u}(\boldsymbol{x}_\delta))\}}}_{\text{Error in discrete solution}} + \underbrace{\boxed{\{J_\delta(\boldsymbol{u}(\boldsymbol{x}_\delta)) - J(\boldsymbol{u})\}}}_{\text{Truncation error}}$$

$$\epsilon_J \approx J_\delta(\boldsymbol{u}+\delta\boldsymbol{u}) - J_\delta(\boldsymbol{u}(\boldsymbol{x}_\delta))$$

$$\approx \left\{\frac{\partial J_\delta}{\partial \boldsymbol{u}}\right\}^t \delta\boldsymbol{u} \approx \left\{\frac{\partial J_\delta}{\partial \boldsymbol{u}}\right\}^t \left[\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{u}}\right]^{-1} \delta\boldsymbol{R} = \underline{\psi^t \delta\boldsymbol{R}}$$

$$\delta\boldsymbol{R} \approx \underbrace{\boxed{\boldsymbol{R}_\delta(\boldsymbol{u}_H^L)}}_{\substack{\text{Low order}\\ \text{interpolated to}\\ \text{high order}}} - \underbrace{\boxed{\boldsymbol{R}_\delta(\boldsymbol{u}_H)}}_{\substack{\text{High order}\\ = 0\ !}}$$

$$= \boldsymbol{R}_\delta(\boldsymbol{u}_H^L)$$

$$\boxed{\epsilon_J = \left\| \sum_{e=1}^{N_{el}} \left[\boldsymbol{\psi}^t \boldsymbol{R}_\delta(\boldsymbol{u}_H^L)\right]_e \right\|}$$

Compressible flow

Governing equations (non-linear)

$$\boldsymbol{R}(\boldsymbol{u}, \nabla \boldsymbol{u}) = \sum_{i=1}^{d} \frac{\partial}{\partial x_i} \left\{ \boldsymbol{f}_i^c(\boldsymbol{u}) - \boldsymbol{f}_i^v(\boldsymbol{u}, \nabla \boldsymbol{u}) \right\} = \boldsymbol{0}$$

Adjoint equations (linear)

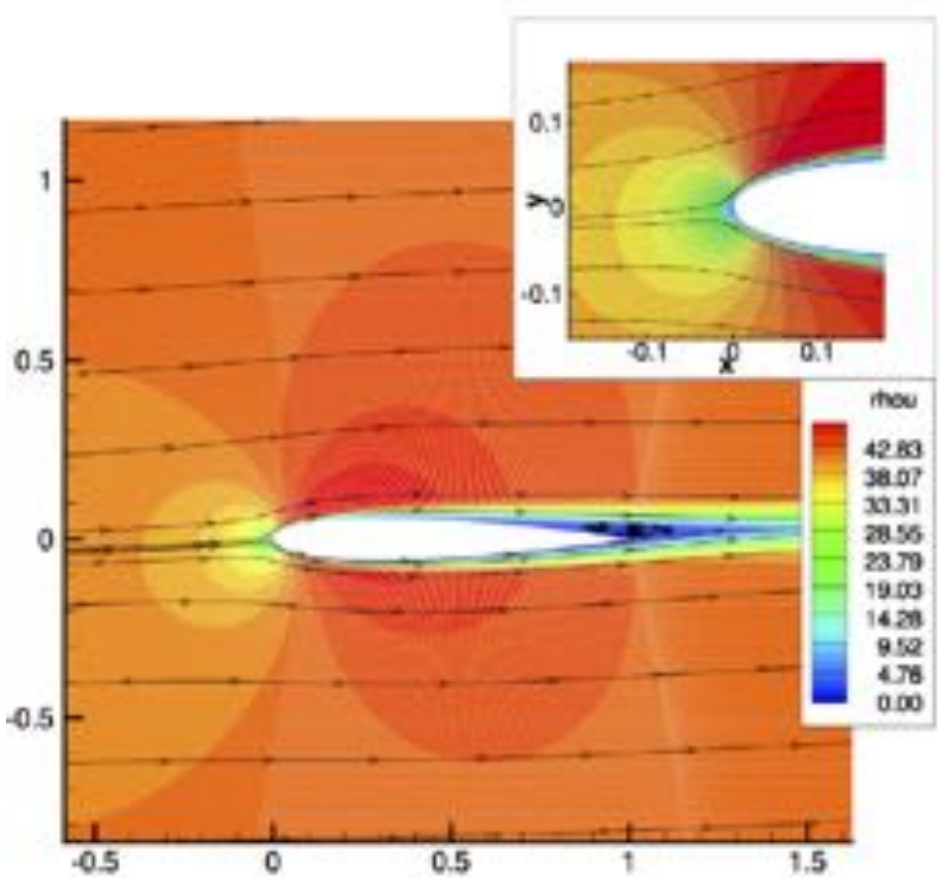
$$\hat{\boldsymbol{R}}(\boldsymbol{\psi}) = -\sum_{i=1}^{d} \left[\frac{\partial \boldsymbol{f}_i^c}{\partial \boldsymbol{u}} - \frac{\partial \boldsymbol{f}_i^v}{\partial \boldsymbol{u}} \right]^t \frac{\partial \boldsymbol{\psi}}{\partial x_i} - \sum_{i,j=1}^{d} \left[\frac{\partial \boldsymbol{f}_j^v}{\partial \boldsymbol{u}_{x_i}} \right]^t \frac{\partial^2 \boldsymbol{\psi}}{\partial x_i^2} = \boldsymbol{0}$$

Goal functional appears in the BCs.

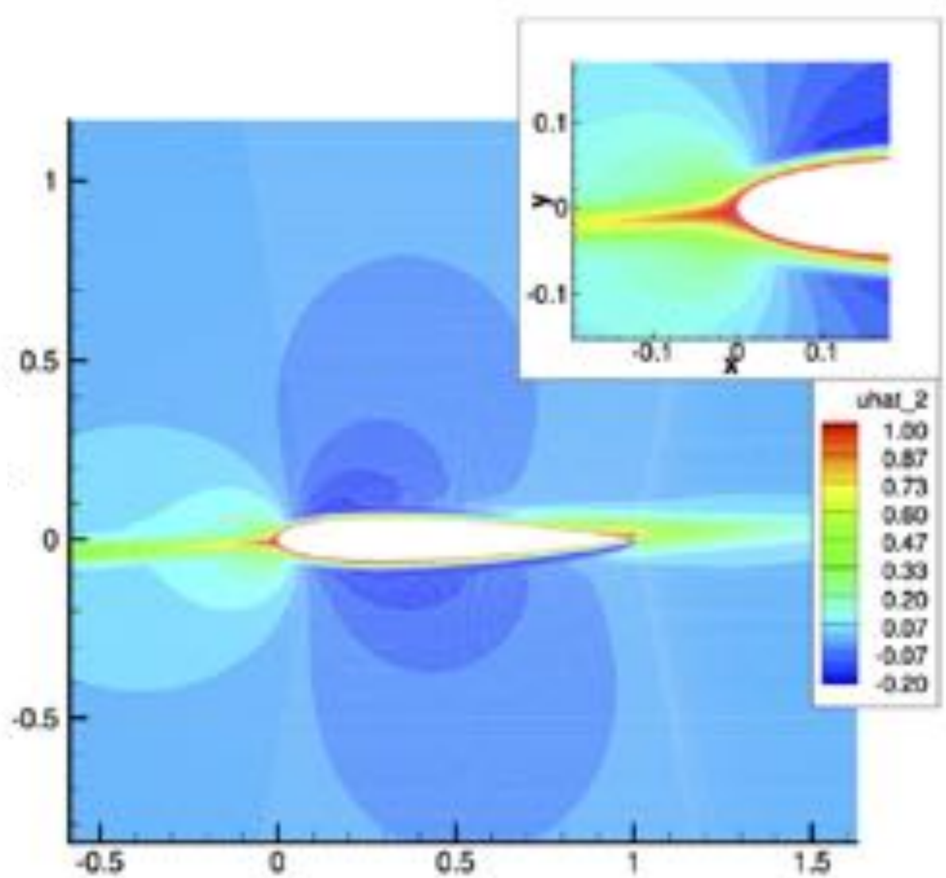
Subsonic laminar flow past a NACA0012

$(\mathrm{Ma} = 0.1, \mathrm{Re} = 5000, \alpha = 2^\circ)$

Goal: minimize error in drag coefficient



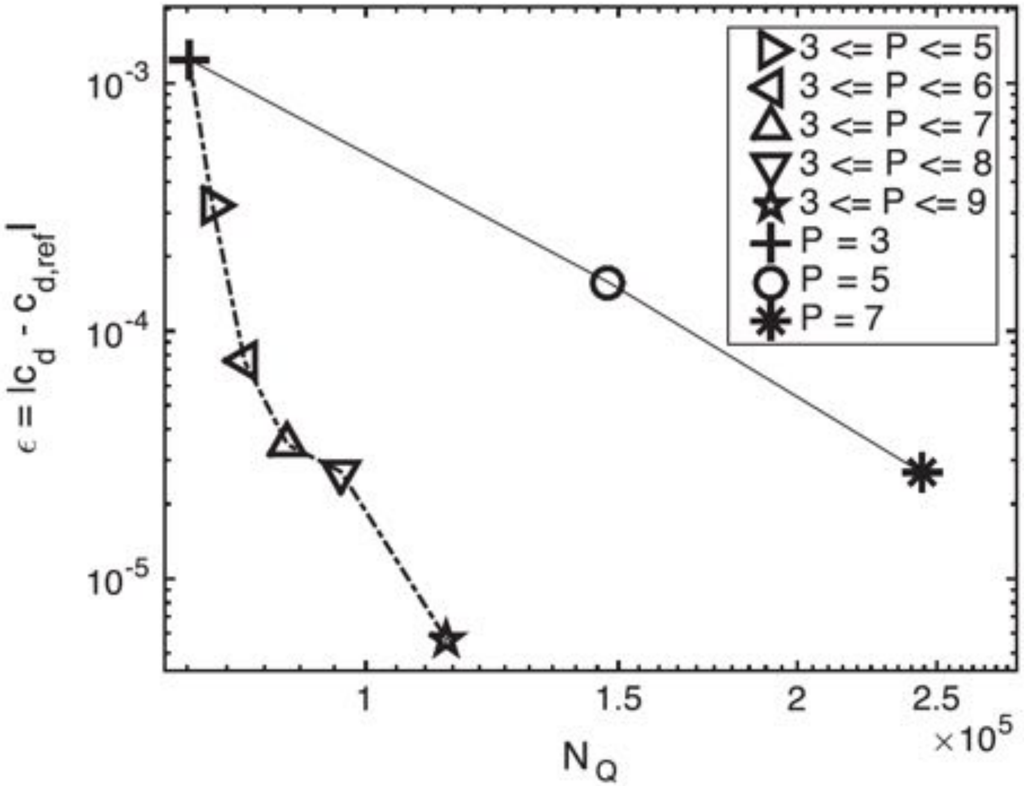
(a) The x-momentum solution (ρu).



(b) The corresponding adjoint variable ψ_2.

Subsonic laminar flow past a NACA0012

$$(\mathrm{Ma} = 0.1, \mathrm{Re} = 5000, \alpha = 2^\circ)$$



Reference solution

Constant **P**	$P=3$	$P=5$	$P=7$	$P=9$	
N_Q	75300	147588	243972	364452	
t_{CPU}	0.28	0.29	0.64	1	
c_d	0.01507	0.01616	0.01629	0.01632	
ϵ_{c_d}	1.20×10^{-3}	1.57×10^{-4}	2.69×10^{-5}	–	
Variable **P**	$3 \leq P \leq 5$	$3 \leq P \leq 6$	$3 \leq P \leq 7$	$3 \leq P \leq 8$	$3 \leq P \leq 9$
N_Q	78183	82286	88165	96013	113909
t_{CPU}	0.31	0.32	0.34	0.36	0.45
c_d	0.01599	0.01624	0.01628	0.01629	0.01631
ϵ_{c_d}	3.19×10^{-4}	7.55×10^{-5}	3.47×10^{-5}	2.71×10^{-5}	5.63×10^{-6}

Where in Nektar++?

Current release of Nektar++ (4.3.2)

- Compressible flow solver
- Variable polynomial order
- Shock capturing

Next release

- Adjoint solver
- Development versions in the git repository:

  branch …. feature/cfs-adjoint