Adjoint-Based Sensitivity Analysis for Computational Fluid Dynamics

Dimitri J. Mavriplis
Department of Mechanical Engineering
University of Wyoming
Laramie, WY

Motivation

- Computational fluid dynamics analysis capabilities commonplace today
- In addition to analysis capability, sensitivity capability is highly desirable
  - Design optimization
  - Error estimation
  - Parameter sensitivity
- Sensitivities may be obtained by:
  - Perturb input, rerun analysis code (Finite difference)
  - Linearizing analysis code (tangent method)
    - Good for 1 input, many outputs
  - Adjoint method
    - Good for many inputs, one output

Motivation

- Continuous vs. Discrete Adjoint Approaches
  - Continuous: Linearize then discretize
  - Discrete: Discretize then Linearize

- Continuous Approach:
  - More flexible adjoint discretizations
  - Framework for non-differentiable tasks (limiters)
  - Often invoked using flow solution as constraint using Lagrange multipliers

- Discrete Approach:
  - Reproduces exact sensitivities of code
    - Verifiable through finite differences
  - Relatively simple implementation
    - Chain rule differentiation of analysis code
    - Transpose these derivates
      - (transpose and reverse order)
    - Includes boundary conditions

Generalized Discrete Sensitivities

- Consider a multi-phase analysis code:

$$\mathbf{L}(\mathbf{D}) = \mathbf{L}(F_{n-1}(F_{n-2}(....F_2(F_1(\mathbf{D}))....))))$$

  - L = Objective(s)
  - D = Design variable(s)

- Sensitivity Analysis

$$\delta\mathbf{L} = \frac{d\mathbf{L}}{d\mathbf{D}}\delta\mathbf{D}$$

  - Using chain rule:

$$\frac{d\mathbf{L}}{d\mathbf{D}} = \frac{\partial\mathbf{L}}{\partial F_{n-1}} \cdot \frac{\partial F_{n-1}}{\partial F_{n-2}} \cdot \cdots \cdot \frac{\partial F_2}{\partial F_1} \cdot \frac{\partial F_1}{\partial\mathbf{D}}$$

Tangent Model

- Special Case:
  - 1 Design variable D, many objectives L
- Precompute all stuff depending on single D
- Construct dL/dD elements as:

$$\frac{d\mathbf{L}}{d\mathbf{D}} = \frac{\partial \mathbf{L}}{\partial F_{n-1}} \cdot \left[\frac{\partial F_{n-1}}{\partial F_{n-2}} \left[\ldots\ldots \left[\frac{\partial F_2}{\partial F_1} \left[\frac{\partial F_1}{\partial \mathbf{D}} \right] \right] \right] \right]$$

Adjoint Model

- Special Case:
  - 1 Objective L, Many Design Variables D
  - Would like to precompute all left terms

$$\frac{d\mathbf{L}}{d\mathbf{D}} = \frac{\partial \mathbf{L}}{\partial F_{n-1}} \cdot \frac{\partial F_{n-1}}{\partial F_{n-2}} \cdot \cdots \cdot \frac{\partial F_2}{\partial F_1} \cdot \frac{\partial F_1}{\partial \mathbf{D}}$$

  - Transpose entire equation:

$$\frac{d\mathbf{L}}{d\mathbf{D}}^T = \frac{\partial F_1}{\partial \mathbf{D}}^T \cdot \frac{\partial F_2}{\partial F_1}^T \cdot \frac{\partial F_{n-1}}{\partial F_{n-2}}^T \cdot \cdots \cdot \frac{\partial \mathbf{L}}{\partial F_{n-1}}^T$$

Adjoint Model

- Special Case:
  - 1 Objective L, Many Design Variables D
  - Would like to precompute all left terms

$$\frac{d\mathbf{L}}{d\mathbf{D}} = \frac{\partial \mathbf{L}}{\partial F_{n-1}} \cdot \frac{\partial F_{n-1}}{\partial F_{n-2}} \cdot \cdots \cdot \frac{\partial F_2}{\partial F_1} \cdot \frac{\partial F_1}{\partial \mathbf{D}}$$

  - Transpose entire equation: precompute as:

$$\frac{d\mathbf{L}}{d\mathbf{D}}^T = \frac{\partial F_1}{\partial \mathbf{D}}^T \cdot \left[\frac{\partial F_2}{\partial F_1}^T \cdot \left[\frac{\partial F_{n-1}}{\partial F_{n-2}}^T \cdot \left[\cdots \left[\frac{\partial \mathbf{L}}{\partial F_{n-1}}^T \right] \right] \right] \right]$$

Shape Optimization Problem

- Multi-phase process:

$$\mathbf{L}(\mathbf{D}) = \mathbf{L}(F_3(F_2(F_1(\mathbf{D}))))$$

$$\mathbf{x}_{surf} = F_1(\mathbf{D})$$

$$\mathbf{x}_{int} = F_2(\mathbf{x}_{surf})$$

$$\mathbf{w} = F_3(\mathbf{x}_{int})$$

$$L = L(\mathbf{w}, \mathbf{x}_{int})$$

Tangent Problem (forward linearization)

$$\frac{d\mathbf{L}}{d\mathbf{D}} = \frac{\partial \mathbf{L}}{\partial \mathbf{w}} \cdot \frac{\partial \mathbf{w}}{\partial \mathbf{x}_{int}} \cdot \frac{\partial \mathbf{x}_{int}}{\partial \mathbf{x}_{surf}} \cdot \frac{\partial \mathbf{x}_{surf}}{\partial \mathbf{D}} + \frac{\partial \mathbf{L}}{\partial \mathbf{x}_{int}} \cdot \frac{\partial \mathbf{x}_{int}}{\partial \mathbf{x}_{surf}} \cdot \frac{\partial \mathbf{x}_{surf}}{\partial \mathbf{D}}$$

- Examine Individual Terms:
  - $\frac{\partial \mathbf{x}_{surf}}{\partial \mathbf{D}}$: Design variable definition (CAD)
  - $\frac{\partial \mathbf{L}}{\partial \mathbf{w}}$: Objective function definition

Tangent Problem (forward linearization)

$$\frac{d\mathbf{L}}{d\mathbf{D}} = \frac{\partial \mathbf{L}}{\partial \mathbf{w}} \cdot \frac{\partial \mathbf{w}}{\partial \mathbf{x}_{int}} \cdot \frac{\partial \mathbf{x}_{int}}{\partial \mathbf{x}_{surf}} \cdot \frac{\partial \mathbf{x}_{surf}}{\partial \mathbf{D}} + \frac{\partial \mathbf{L}}{\partial \mathbf{x}_{int}} \cdot \frac{\partial \mathbf{x}_{int}}{\partial \mathbf{x}_{surf}} \cdot \frac{\partial \mathbf{x}_{surf}}{\partial \mathbf{D}}$$

- Examine Individual Terms:
  - $[K]\,\delta x_{int} = \delta x_{surf}$ $\qquad \dfrac{\partial \mathbf{x}_{int}}{\partial \mathbf{x}_{surf}} = [K]^{-1}$
  - $\mathbf{R}(\mathbf{w}(\mathbf{x}_{int}), \mathbf{x}_{int}) = 0$ $\qquad \dfrac{\partial \mathbf{w}}{\partial \mathbf{x}_{int}} = -\left[\dfrac{\partial \mathbf{R}}{\partial \mathbf{w}}\right]^{-1} \dfrac{\partial \mathbf{R}}{\partial \mathbf{x}_{int}}$

Sensitivity Analysis

- Tangent Problem:

$$\frac{d\mathbf{L}}{d\mathbf{D}} = -\frac{\partial \mathbf{L}}{\partial \mathbf{w}} \cdot \left[\frac{\partial \mathbf{R}}{\partial \mathbf{w}}\right]^{-1} \frac{\partial \mathbf{R}}{\partial \mathbf{x}_{int}} [K]^{-1} \frac{\partial \mathbf{x}_{surf}}{\partial \mathbf{D}} + \frac{\partial \mathbf{L}}{\partial \mathbf{x}_{int}} \cdot [K]^{-1} \frac{\partial \mathbf{x}_{surf}}{\partial \mathbf{D}}$$

- Adjoint Problem

$$\frac{d\mathbf{L}}{d\mathbf{D}}^{T} = \frac{\partial \mathbf{x}_{surf}}{\partial \mathbf{D}}^{T} [K]^{-T} \left[\frac{\partial \mathbf{L}}{\partial \mathbf{x}_{int}}^{T} - \frac{\partial \mathbf{R}}{\partial \mathbf{x}_{int}}^{T} \left[\frac{\partial \mathbf{R}}{\partial \mathbf{w}}\right]^{-T} \frac{\partial \mathbf{L}}{\partial \mathbf{w}}^{T}\right]$$

Tangent Problem

- 1: Surface mesh sensitivity: $\frac{\partial \mathbf{x}_{surf}}{\partial \mathbf{D}}$
- 2: Interior mesh sensitivity: $[K]\frac{\partial \mathbf{x}_{int}}{\partial \mathbf{D}} = \frac{\partial \mathbf{x}_{surf}}{\partial \mathbf{D}}$
- 3: Residual sensitivity: $\frac{\partial \mathbf{R}}{\partial \mathbf{D}} = \frac{\partial \mathbf{R}}{\partial \mathbf{x}_{int}} \cdot \frac{\partial \mathbf{x}_{int}}{\partial \mathbf{D}}$
- 4: Flow variable sensitivity: $\left[\frac{\partial \mathbf{R}}{\partial \mathbf{w}}\right]\frac{\partial \mathbf{w}}{\partial \mathbf{D}} = -\frac{\partial \mathbf{R}}{\partial \mathbf{D}}$
- 5: Final sensitivity $\frac{d\mathbf{L}}{d\mathbf{D}} = \frac{\partial \mathbf{L}}{\partial \mathbf{w}} \cdot \frac{\partial \mathbf{w}}{\partial \mathbf{D}} + \frac{\partial \mathbf{L}}{\partial \mathbf{x}_{int}} \cdot \frac{\partial \mathbf{x}_{int}}{\partial \mathbf{D}}$

Adjoint Problem

- 1: Objective flow sensitivity: $\frac{\partial \mathbf{L}}{\partial \mathbf{w}}$
- 2: Flow adjoint: $\left[\frac{\partial \mathbf{R}}{\partial \mathbf{w}}\right]^T \mathbf{\Lambda}_w = \frac{\partial \mathbf{L}}{\partial \mathbf{w}}^T$
- 3:Objective sens. wrt mesh: $\frac{d\mathbf{L}}{d\mathbf{x}^*_{\mathbf{int}}}^T = \frac{\partial \mathbf{L}}{\partial \mathbf{x}_{int}}^T - \frac{\partial \mathbf{R}}{\partial \mathbf{x}_{int}}^T \mathbf{\Lambda}_w$
- 4: Mesh adjoint: $[K]^T \mathbf{\Lambda}_x = \frac{d\mathbf{L}}{d\mathbf{x}^*_{int}}^T$
- 5: Final sensitivity: $\frac{d\mathbf{L}}{d\mathbf{D}}^T = \frac{\partial \mathbf{x}_{surf}}{\partial \mathbf{D}}^T \mathbf{\Lambda}_x$

General Approach

- Linearize each subroutine/process individually in analysis code
  - Check linearization by finite difference
  - Transpose, and check duality relation
- Build up larger components
  - Check linearization, duality relation
- Check entire process for FD and duality
- Use single modular AMG solver for all phases

General Duality Relation

•Analysis Routine: $f = f(x)$

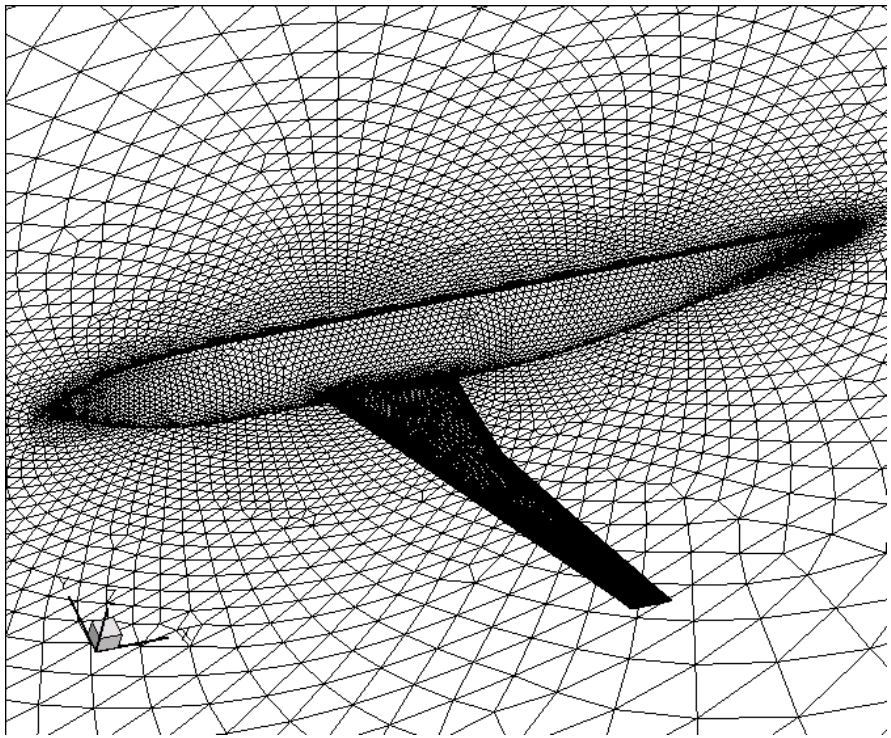
•Tangent Model: $\delta f_1 = \frac{\partial f}{\partial x} \delta x_1$

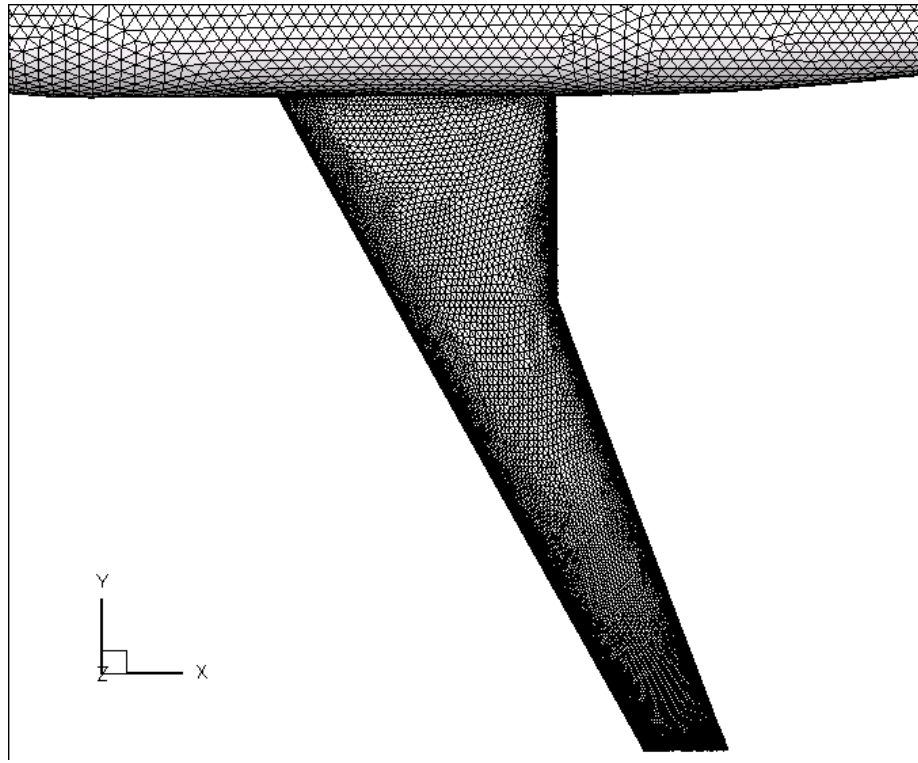
•Adjoint Model: $\delta x_2 = \frac{\partial f}{\partial x}^T \partial f_2$

•Duality Relation: $\partial f_2^T . \partial f_1 = \partial x_2^T . \partial x_1$

- Necessary but not sufficient test
  - Check using series of arbitrary input vectors $\partial x_1 \partial f_2$

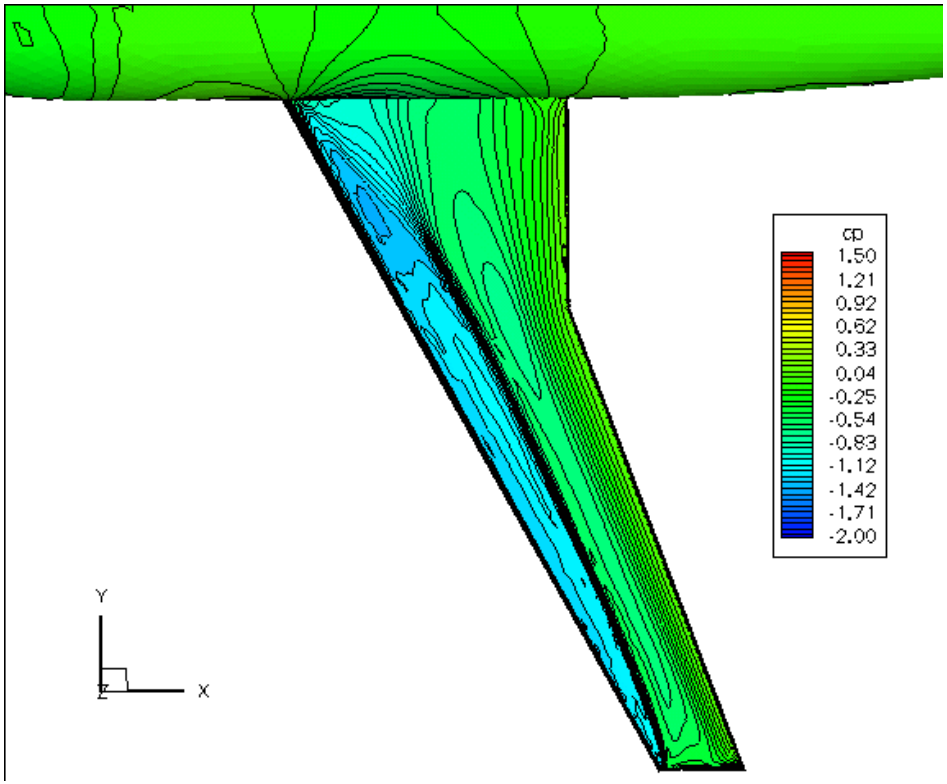
Drag Minimization Problem

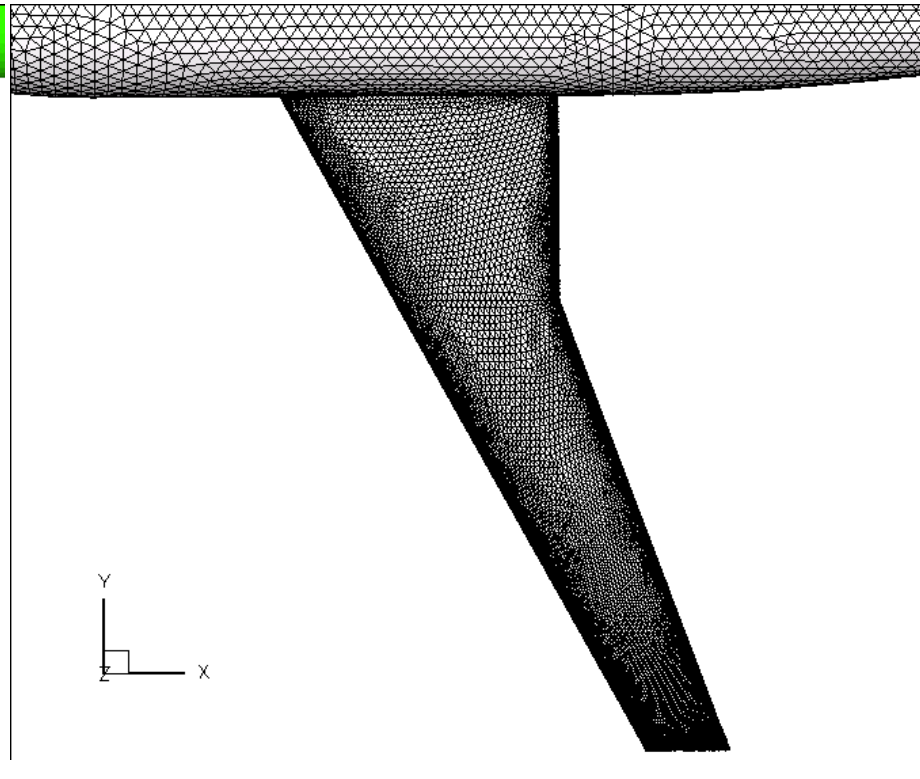




- DLR-F6 Wing body configuration
- 1.12M vertices, 4.2M cells

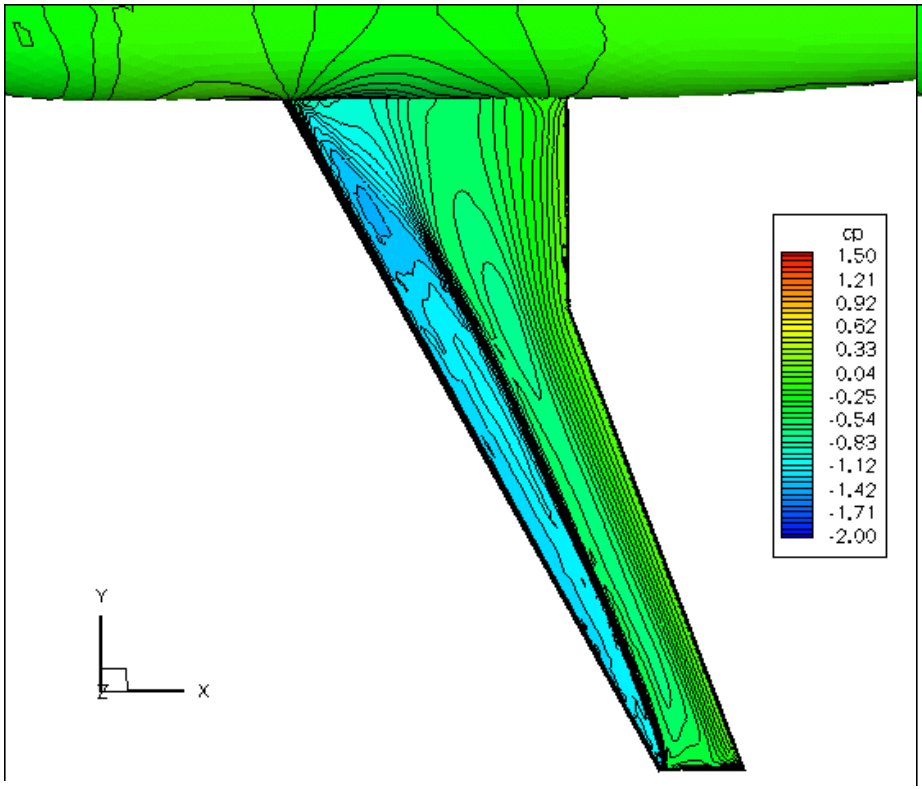
Drag Minimization Problem

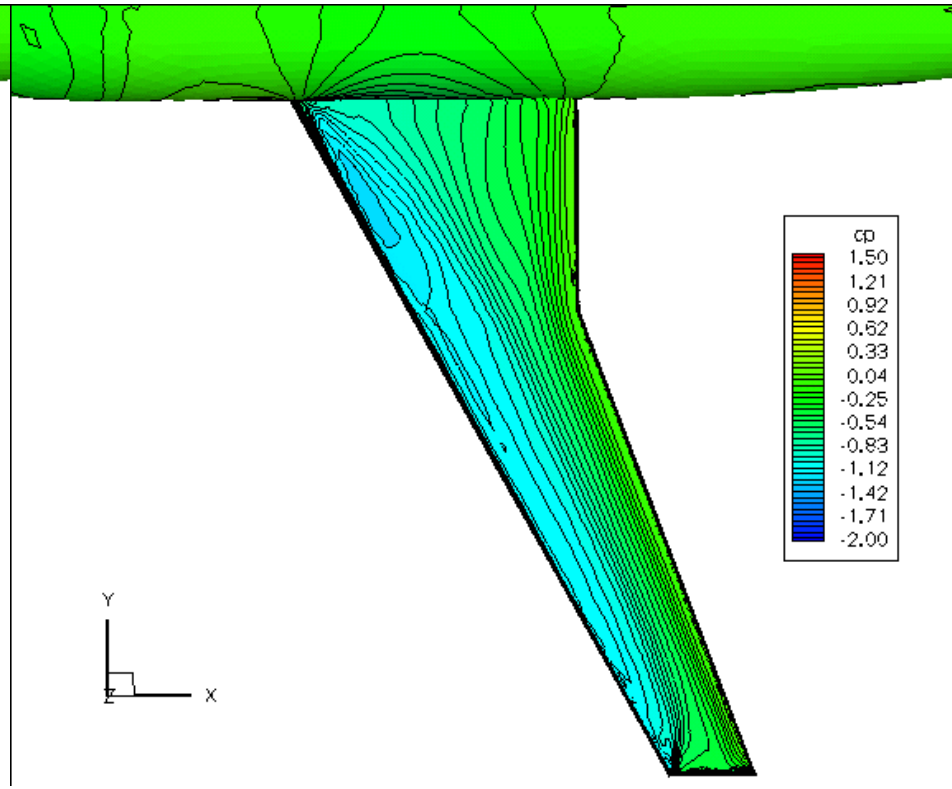




- DLR-F6 Wing body configuration
- Mach=0.75, Incidence=1^{o} , Re=3M

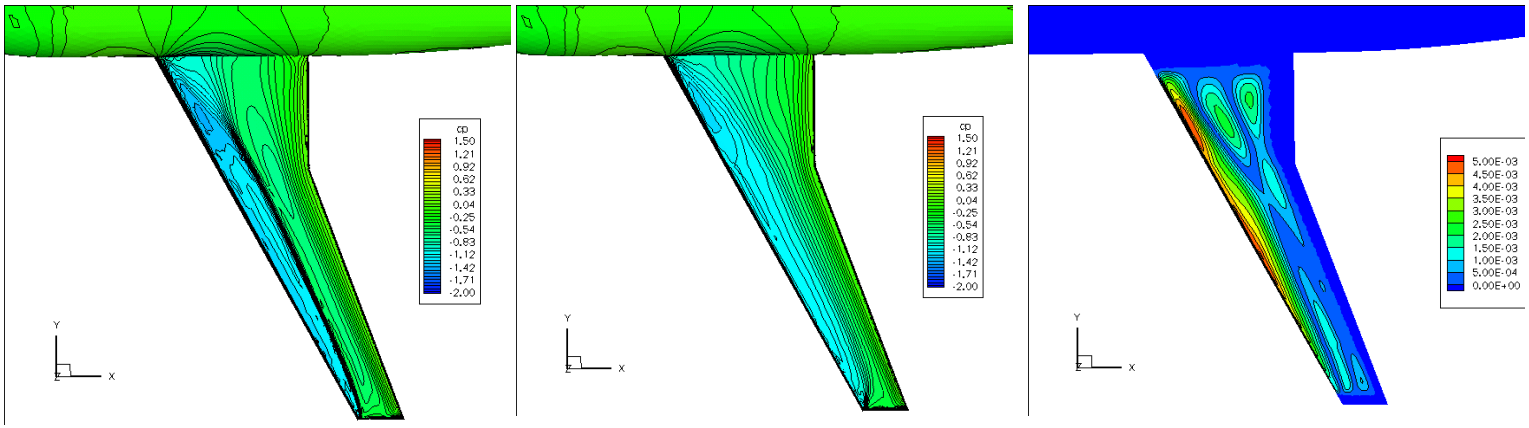
Drag Minimization Problem

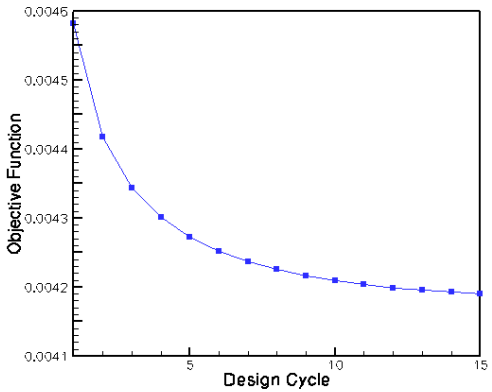




- Substantial reduction in shock strength after 15 design cycles
- CD: 302 counts → 288 counts : -14 counts
  - Wave drag

Drag Minimization Problem





Total Optimization Time for 15 Design Cycles:
6 hours on 16 cpus of PC cluster

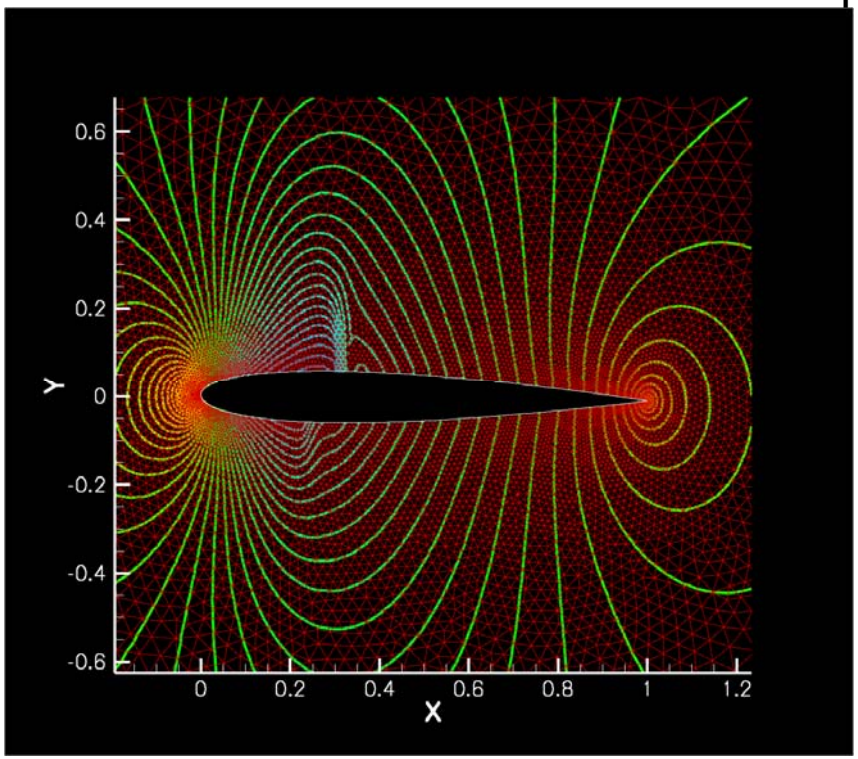
- Flow Solver: 150 MG cycles
- Flow Adjoint: 50 Defect-Correction cycles (x 4 MG)
- Mesh Adjoint: 25 MG cycles
- Mesh Motion: 25 MG cycles

Extension to Unsteady Problems

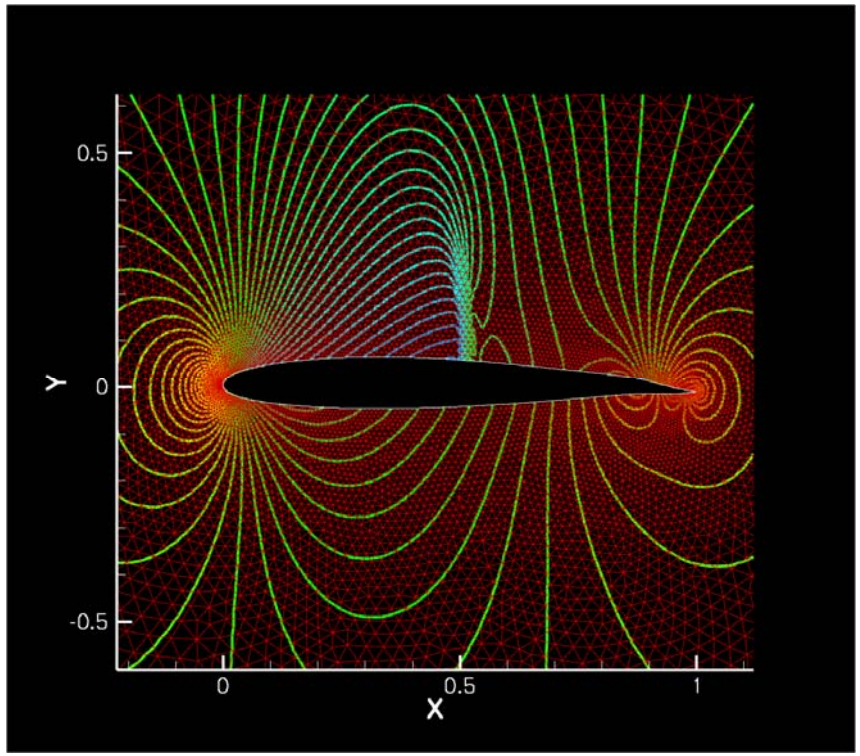
Pressure Contours for Pitching Airfoils

M_{inf} = 0.755, α_0 = 0.016º, α_{max} = 2.51º, ω = 0.1628, t=0 to 54

27 time-steps with dt=2.0



NACA0012 Baseline Airfoil



Optimized Airfoil

Governing Equations (ALE)

$$\frac{\partial \mathbf{w}(\mathbf{x},t)}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{w}) = 0$$

$$\frac{\partial V\mathbf{w}}{\partial t} + \int_{dB(t)} [\mathbf{F}(\mathbf{w}) - \dot{\mathbf{x}}\mathbf{w}] \cdot \mathbf{n} dB = 0$$

- In ALE Form:
  - V(t) = control volume
  - $\int \dot{x}.ndB =$ Face integrated mesh velocity

    Formulated to obey GCL = f(x^n,x^{n-1},x^{n-2},…)

    Mavriplis and Yang (AIAA 2005-5114) for high order IRK

    Mavriplis and Nastase (AIAA/JCP to appear) for DG methods

Unsteady Residual Form

- BDF1:

$$\mathbf{R}^n(\mathbf{w}^n, \mathbf{w}^{n-1}, \mathbf{x}^n, \mathbf{x}^{n-1}) =$$
$$\frac{1}{\Delta t} V(x^n)\mathbf{w}^n - \frac{1}{\Delta t} V(x^{n-1})\mathbf{w}^{n-1} + \mathbf{S}^n(\mathbf{w}^n, \mathbf{x}^n, \mathbf{x}^{n-1})$$

  Due uniquely to ALE grid speed terms

- BDF2: Similar expression depending on w^n, w^{n-1}, w^{n-2}, x^n, x^{n-1}, x^{n-2}

Shape Optimization

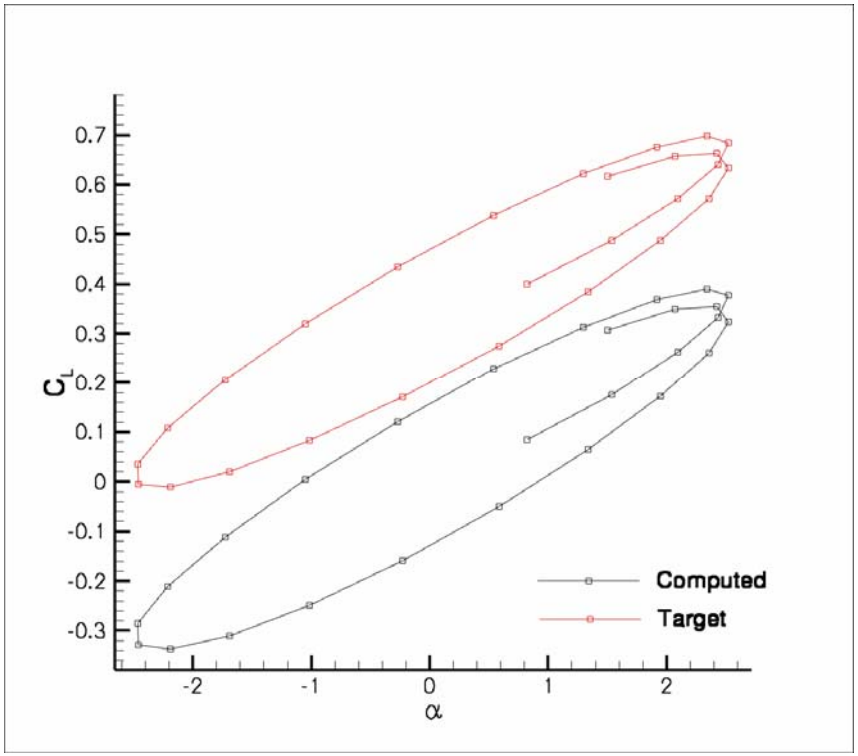
- Per design cycle (Steady Case)
  - One mesh deformation problem
  - One flow analysis
  - One flow adjoint solution
  - One mesh adjoint solution
- Unsteady Shape Optimization
  - General functional dependence involves previous time step values
  - Chain rule results in forward time recurrence relation
  - When transposed (adjoint) results in backwards “integration” in time
  - Time history of solution must be stored for use by adjoint in reverse time integration
    - Write out to local cluster disks, read back in during adjoint phase

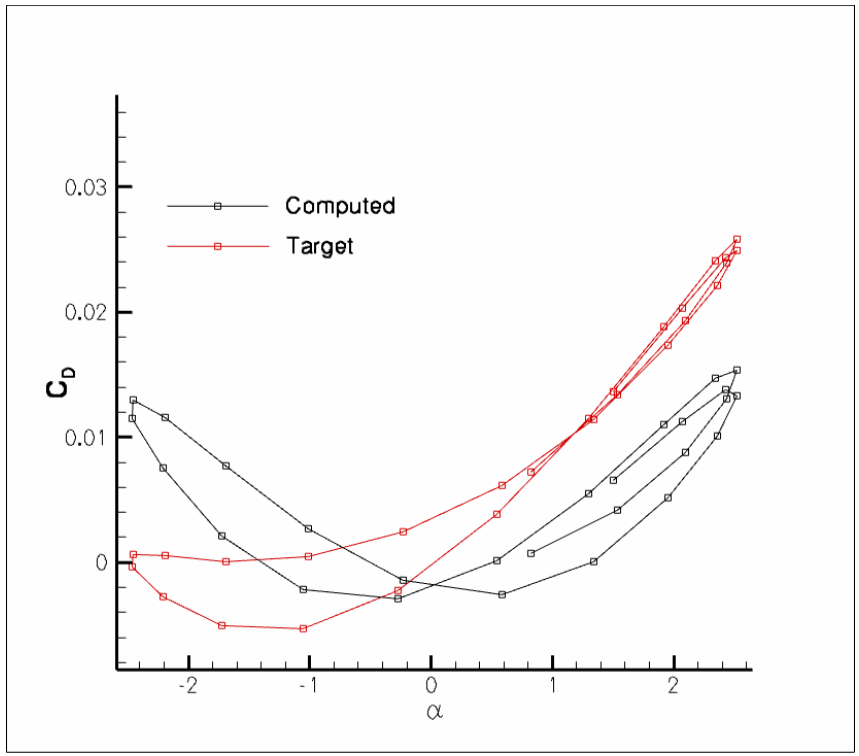
Validation of Sensitivities

	Steady State		Time = 1		Time = 10	
	C_L	C_D	C_L	C_D	C_L	C_D
FD: $\epsilon = 10^{-08}$	-0.3520915892	0.2643615382	-0.3768952150	0.2413373861	-	-
FD: $\epsilon = 10^{-09}$	-0.3520720937	0.2642965013	-0.3768726442	0.2412741950	-0.3504422307	0.2730632664
FD: $\epsilon = 10^{-10}$	-0.3521249958	0.2642880359	-0.3768896306	0.2412681165	-	-
Tangent	-0.3516050678	0.2652594347	-0.3759034567	0.2425679122	-0.3494761235	0.2746345922
Adjoint	-0.3516050677	0.2652594347	-0.3759034566	0.2425679122	-0.3494761233	0.2746345923

- Steady/Unsteady sensitivities compare well with finite difference values
- Tangent/Adjoint values equivalent to machine precision
  - Duality principle

Time-Dependent Load Convergence/Comparison



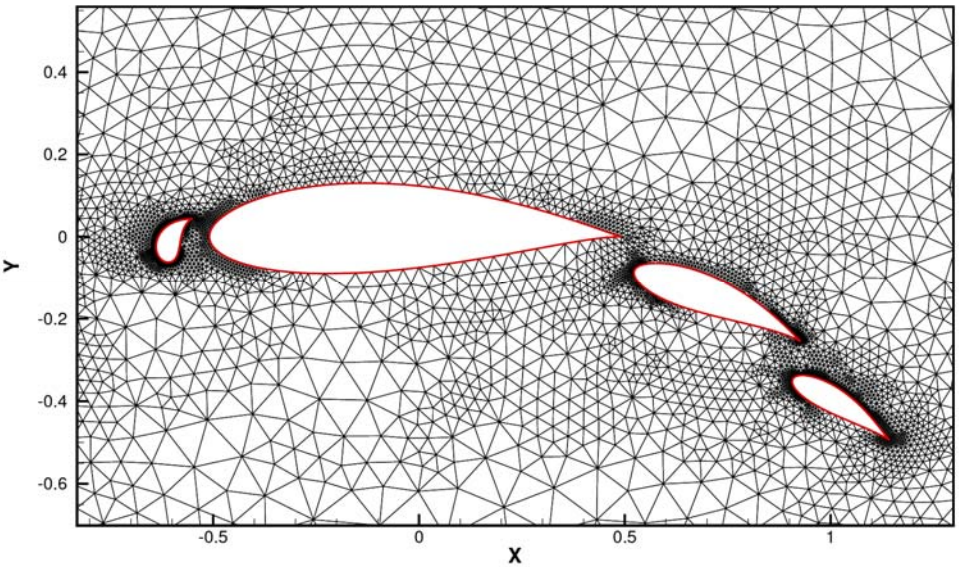


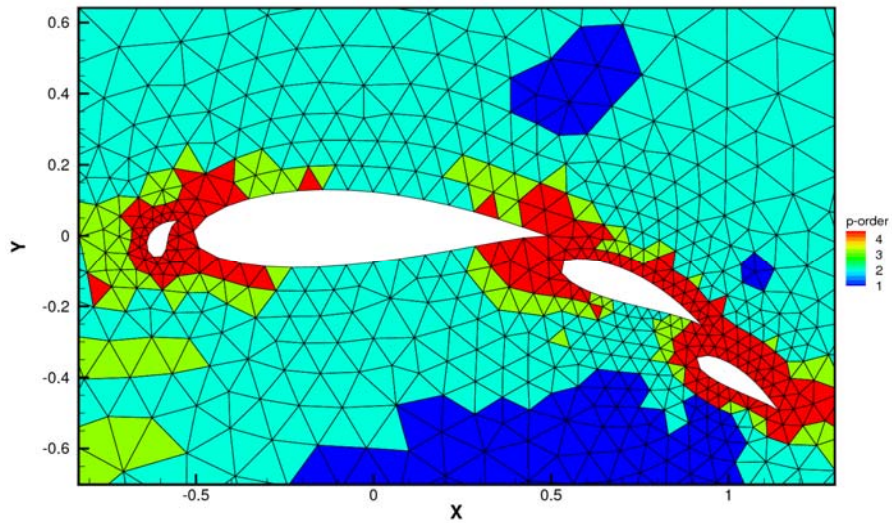
Mani and Mavriplis AIAA 2007-0060

Adjoint-Based Error Estimation

- Complex simulations have multiple error sources
- Engineering simulations concerned with specific output objectives
- Adjoint methods / Goal Oriented Approach
  - Methodical approach for constructing discrete adjoint
  - Use for a posteriori error estimation
    - Spatial error
    - Temporal error
    - Other error sources
  - Use to drive adaptive process

Goal-Oriented Spatial Adaptivity (steady-state)





DG Discretization

h-adaptivity (Drag)

p-adaptivity (Drag)

ADJOINT-BASED ERROR ESTIMATION

- Formulation
  - Taylor series

$$J_h(\tilde{u}_h) = J_h(\tilde{u}_H^h) + \left.\frac{\partial J_h}{\partial \tilde{u}_h}\right|_{\tilde{u}_H^h} \left(\tilde{u}_h - \tilde{u}_H^h\right) + \cdots \qquad (\tilde{u}_H^h = I_H^h \tilde{u}_H)$$

$$R_h(\tilde{u}_h) = R_h(\tilde{u}_H^h) + \left[\frac{\partial R_h}{\partial \tilde{u}_h}\right]_{\tilde{u}_H^h} \left(\tilde{u}_h - \tilde{u}_H^h\right) + \cdots = 0$$

$$\left(\tilde{u}_h - \tilde{u}_H^h\right) \approx -\left[\left.\frac{\partial R_h}{\partial \tilde{u}_h}\right|_{\tilde{u}_H^h}\right]^{-1} R_h(\tilde{u}_H^h)$$

$$J_h(\tilde{u}_h) \approx J_h(\tilde{u}_H^h) - \left.\frac{\partial J_h}{\partial \tilde{u}_h}\right|_{\tilde{u}_H^h} \left[\left.\frac{\partial R_h}{\partial \tilde{u}_h}\right|_{\tilde{u}_H^h}\right]^{-1} R_h(\tilde{u}_H^h)$$

$\left.\Lambda_h^T\right|_{\tilde{u}_H^h}$, adjoint solution

ADJOINT-BASED ERROR ESTIMATION

- Formulation
  - Discrete adjoint problem

$$\Lambda_h^T\Big|_{\tilde{\mathbf{u}}_H^h} = \frac{\partial J_h}{\partial \tilde{\mathbf{u}}_h}\Big|_{\tilde{\mathbf{u}}_H^h}\left[\frac{\partial \mathbf{R}_h}{\partial \tilde{\mathbf{u}}_h}\Big|_{\tilde{\mathbf{u}}_H^h}\right]^{-1} \quad \text{or} \quad \left[\frac{\partial \mathbf{R}_h}{\partial \tilde{\mathbf{u}}_h}\Big|_{\tilde{\mathbf{u}}_H^h}\right]^{T}\Lambda_h\Big|_{\tilde{\mathbf{u}}_H^h} = \left(\frac{\partial J_h}{\partial \tilde{\mathbf{u}}_h}\right)^T\Big|_{\tilde{\mathbf{u}}_H^h}$$

Transpose of Jacobian matrix

    - Linear system of equations
    - Delivers similar convergence rate as the flow solver

- Approximated objective becomes

$$J_h(\tilde{\mathbf{u}}_h) \approx J_h(\tilde{\mathbf{u}}_H^h) - \Lambda_h^T\Big|_{\tilde{\mathbf{u}}_H^h}\mathbf{R}_h(\tilde{\mathbf{u}}_H^h)$$

ADJOINT-BASED ERROR ESTIMATION

- Formulation
  - Avoid solving the adjoint variables on the fine mesh, instead,
  - Solve Λ_H on the coarse mesh

$$\left[\left.\frac{\partial \mathbf{R}_H}{\partial \tilde{\mathbf{u}}_H}\right|_{\tilde{\mathbf{u}}_H}\right]^{\mathsf{T}} \Lambda_H = \left(\left.\frac{\partial J_H}{\partial \tilde{\mathbf{u}}_H}\right|_{\tilde{\mathbf{u}}_H}\right)^T$$

  - Reconstruct Λ_H onto the fine mesh by using least squares method

$$\Lambda_H^h = \mathbf{J}_H^h \Lambda_H$$

  - Approximated objective becomes

$$J_h(\tilde{\mathbf{u}}_h) \approx J_h(\tilde{\mathbf{u}}_H^h) - (\Lambda_h^H)^T \mathbf{R}_h(\tilde{\mathbf{u}}_H^h)$$

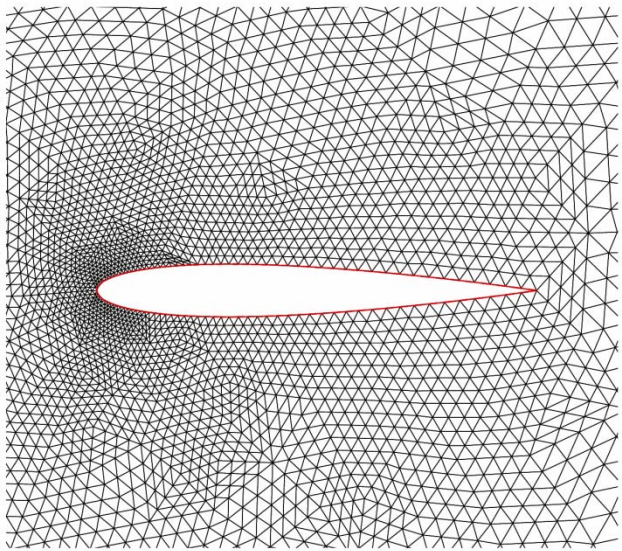
or

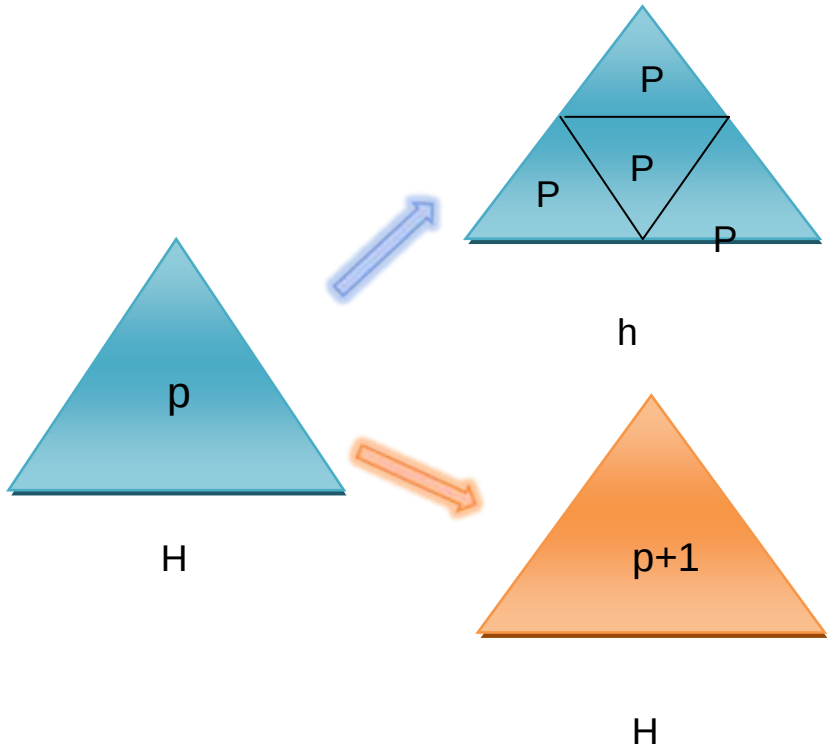
$$J_h(\tilde{\mathbf{u}}_h) - J_H(\tilde{\mathbf{u}}_H) \approx J_h(\tilde{\mathbf{u}}_H^h) - J_H(\tilde{\mathbf{u}}_H) - (\Lambda_h^H)^T \mathbf{R}_h(\tilde{\mathbf{u}}_H^h)$$

correction $\quad \varepsilon_1 \quad \varepsilon_2$

ADAPTIVE MESH STRATEGIES

- *h*-refinement
  - Local mesh subdivision





- *p-enrichment*
  - Local variation of discretization orders

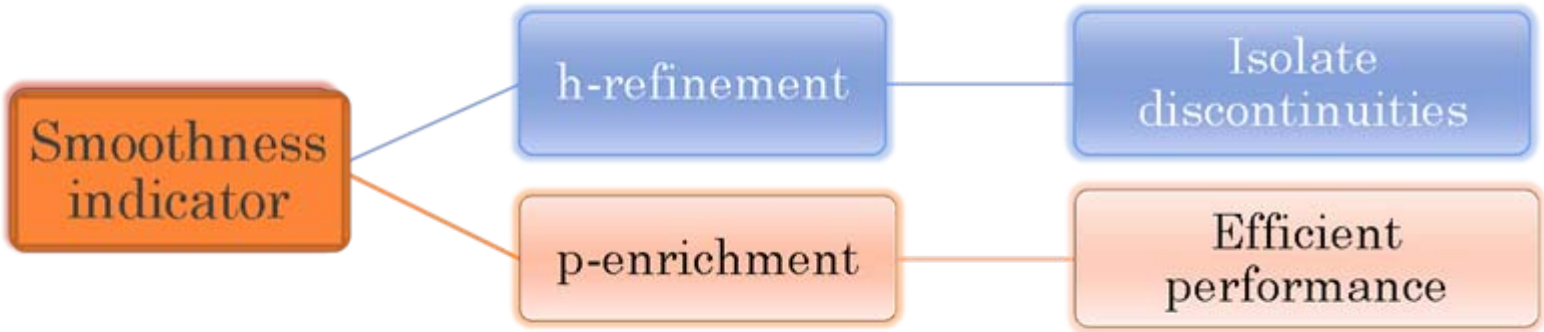
- *hp-refinement*

Local implementation of the *h*- **or** *p*-refinement individually

ADAPTIVE MESH STRATEGIES

- *hp*-refinement (contd.)
  - For **EACH** flagged element,

How to make a decision between *h*- and *p*-refinement??



- Local smoothness / shock indicator

ADAPTIVE MESH STRATEGIES

- Local smoothness / shock indicator
  - Element-wise smoothness [*Persson, Peraire*],

$$S_{ek} = \frac{(u_p - u_{p-1}, u_p - u_{p-1})_k}{(u_p, u_p)_k}$$

where $u_p = \sum_{i=1}^{N(p)} U_i \phi_i$ $u_{p-1} = \sum_{i=1}^{N(p-1)} U_i \phi_i$

$$s_{ek} = \log_{10}(S_{ek})$$

$s_{ek} > 1/p^4 - \kappa$ h-refinement; otherwise, p-enrichment

  - Inter-element jumps[*Krivodonova,Xin,Chevaugeon,Flaherty*],

$$S_{ek} = \frac{1}{|\partial l|} \int_{\partial l} \left| \frac{q^+ - q^-}{\frac{1}{2}(q^+ + q^-)} \right| dS$$

$s_{ek} > \frac{1}{K}$ h-refinement; otherwise, p-enrichment

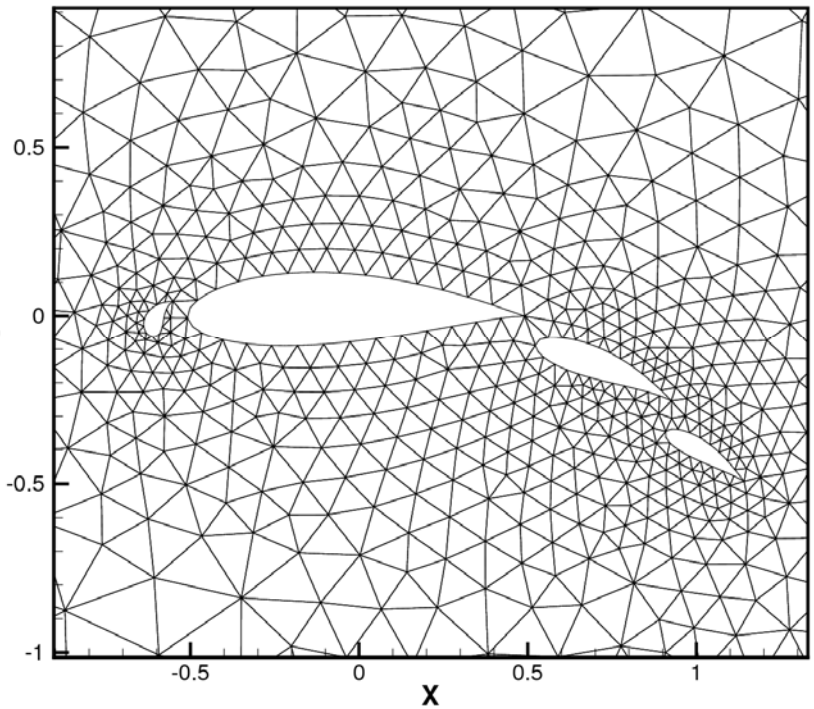
NUMERICAL RESULTS

- Subsonic flow over a four-element airfoil (M_∞=0.2)

- Zero angle of attack, α=0
- Target function: lift or drag

$$J = \int_{\partial\Omega_w} p n_x dS \qquad J = \int_{\partial\Omega_w} p n_y dS$$

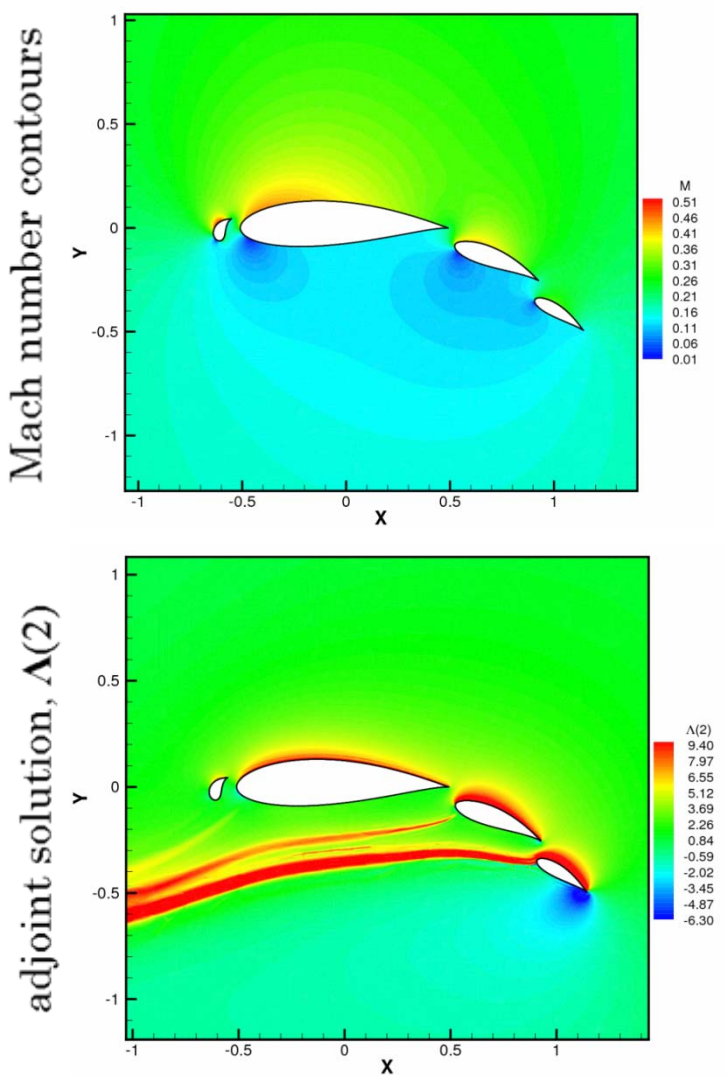
- Starting interpolation order $p = 1$
- *HLLC* Riemann solver & special wall boundary treatment
- ***hp*-Multigrid accelerator**
- Various adaptation algorithms
  - ✓ *h*-refinement
  - ✓ *p*-refinement
  - ✓ *hp*-refinement



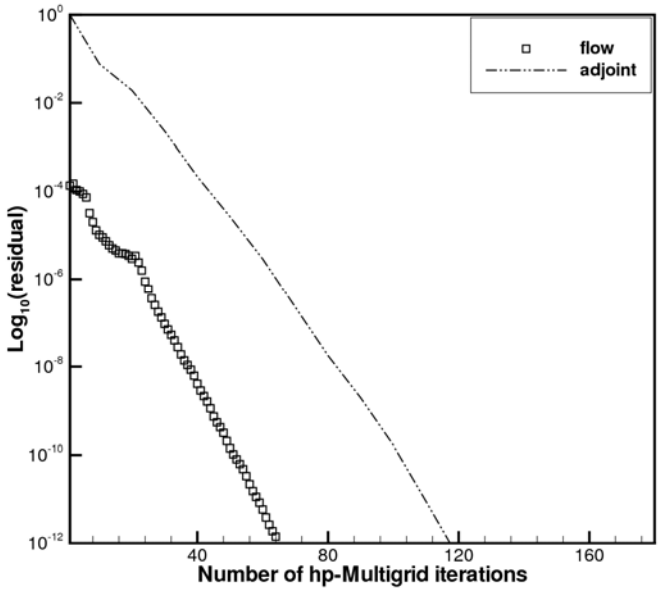
initial mesh (1508 elements)

NUMERICAL RESULTS

- Subsonic flow over a four-element airfoil (M_∞=0.2)



Primal and dual problems
target function : lift

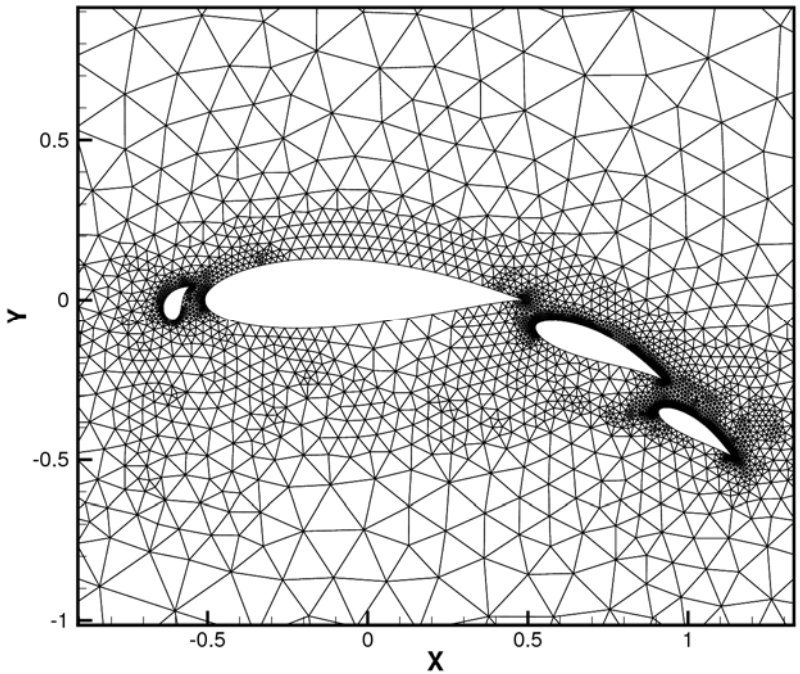


Comparisons on *hp*-Multigrid convergence for the flow and adjoint solutions

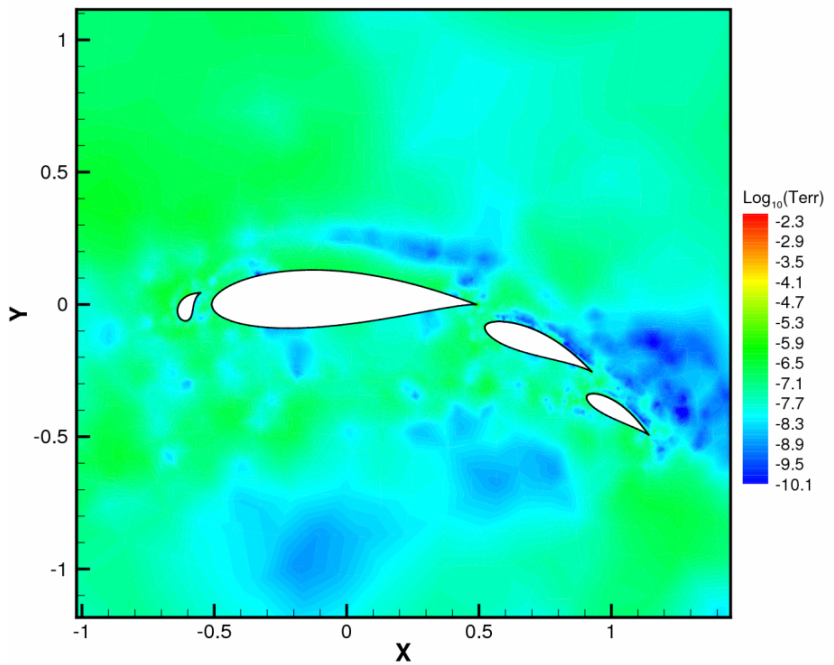
NUMERICAL RESULTS

- Subsonic flow over a four-element airfoil (M_∞=0.2)

Target function of lift ($p = 1$)



adapted mesh (8387 elements)

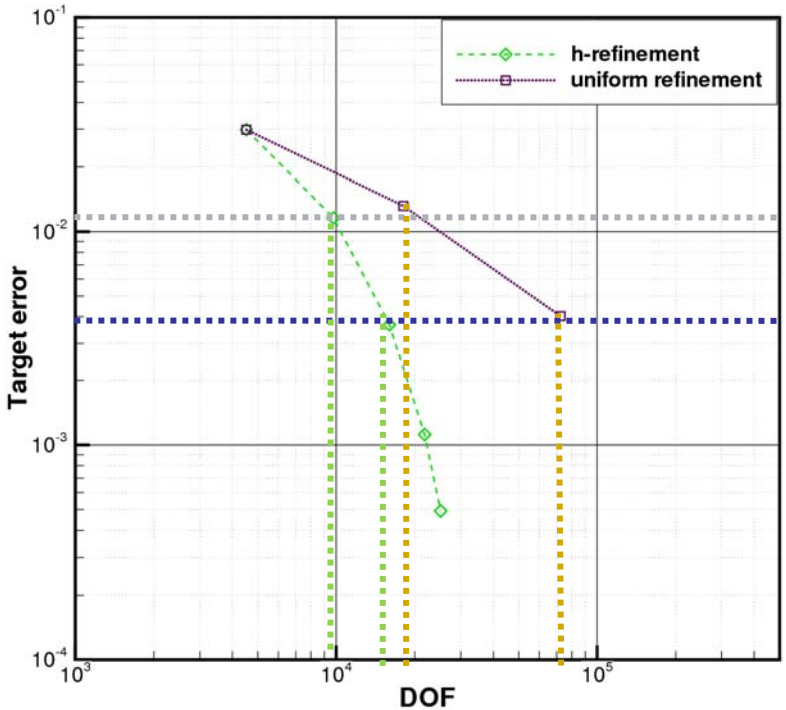


spatial error distribution in the objective functional of lift

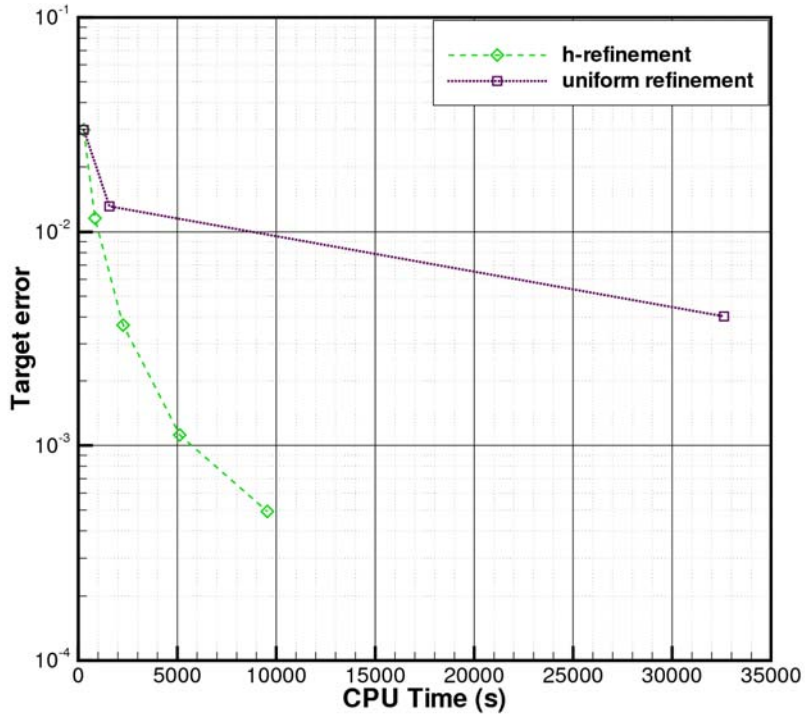
NUMERICAL RESULTS

- Subsonic flow over a four-element airfoil (M_∞=0.2)

Target function of lift ($p = 1$)



error convergence history vs. degrees of freedom

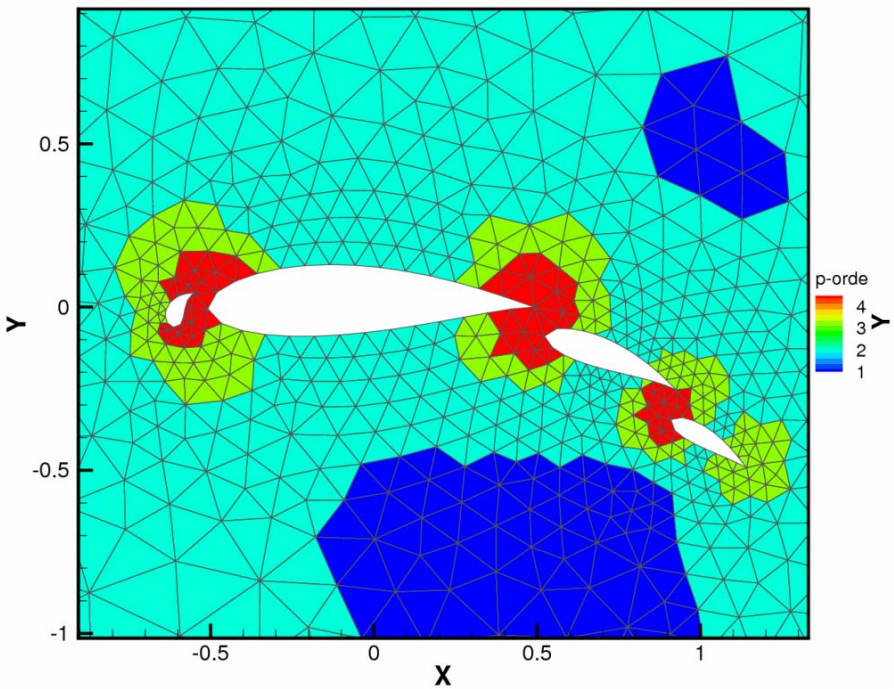


error convergence history vs. CPU time (sec)

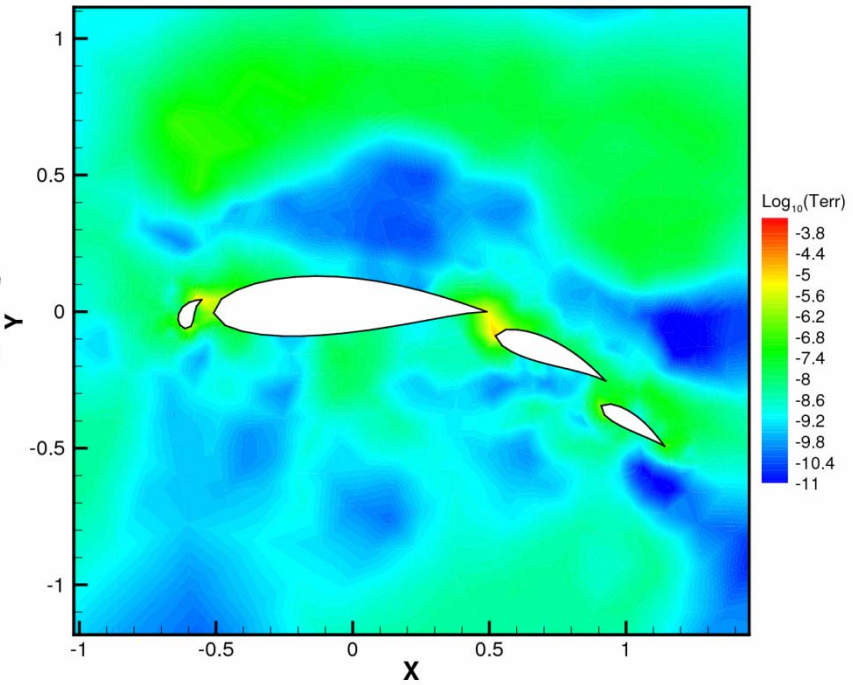
Li Wang, Dimitri Mavriplis (UW)

NUMERICAL RESULTS

- Subsonic flow over a four-element airfoil (M_∞=0.2)
  - *p*-enrichment, Target function of drag



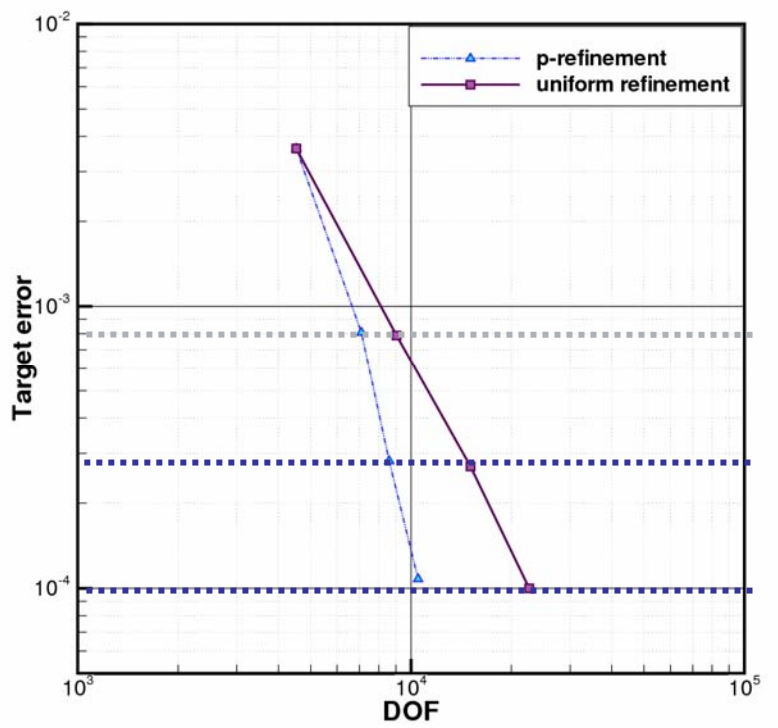
adapted mesh (1508 elements)
discretization orders: p=1~4



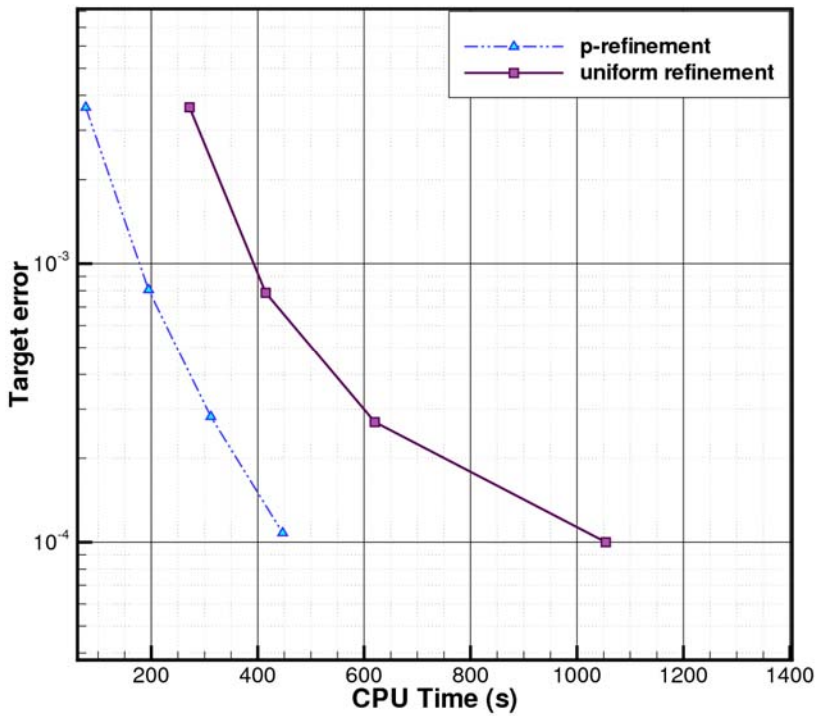
spatial error distribution for
the objective functional of drag

NUMERICAL RESULTS

- Subsonic flow over a four-element airfoil (M_∞=0.2)
  - *p*-enrichment, Target function of drag



error convergence history vs. degrees of freedom

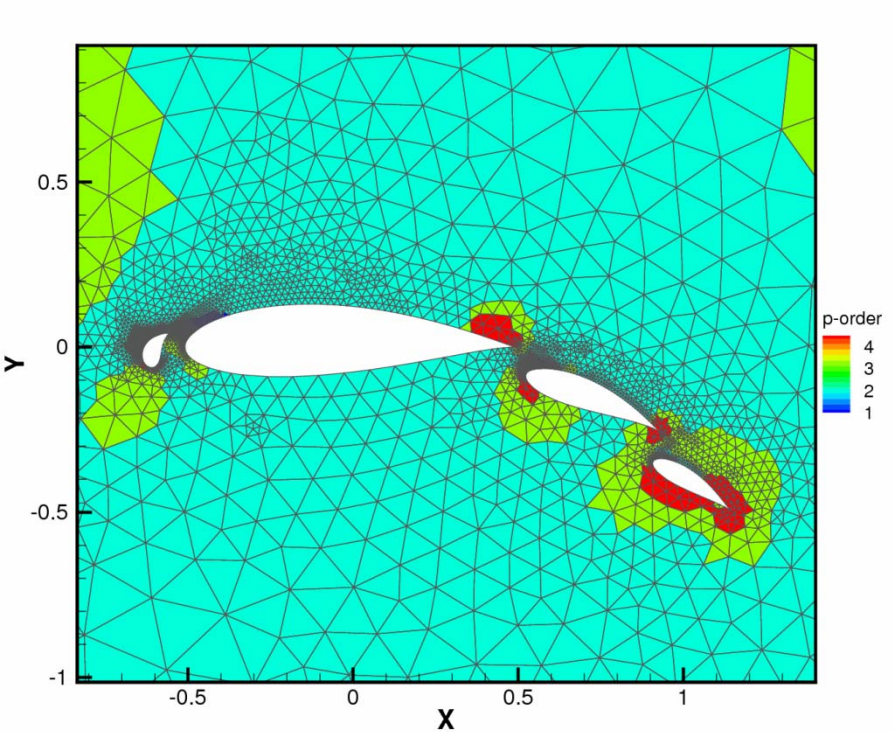


error convergence history vs. CPU time (sec)

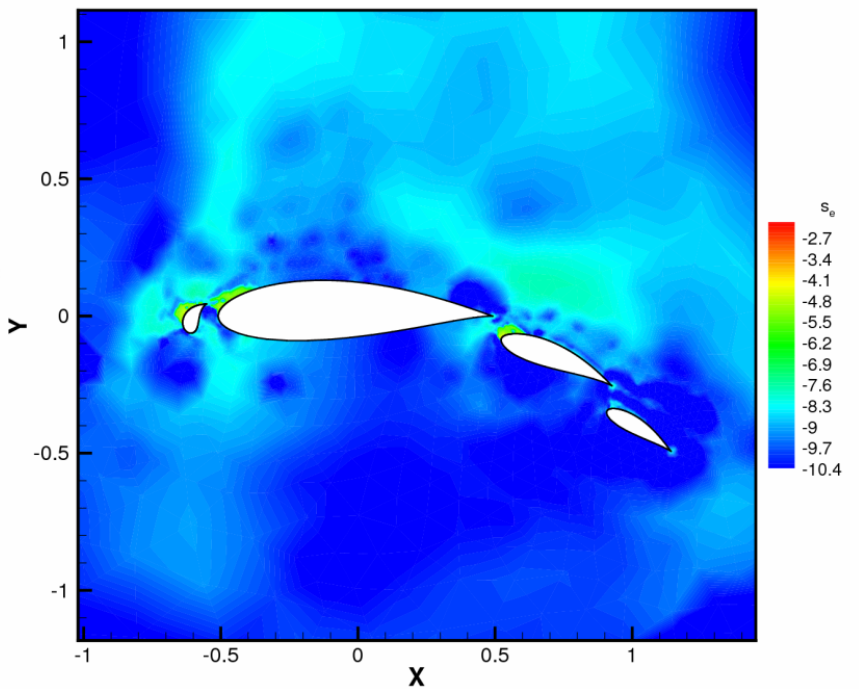
Li Wang, Dimitri Mavriplis (UW)

NUMERICAL RESULTS

- Subsonic flow over a four-element airfoil (M_∞=0.2)
  - *hp*-refinement, Target function of drag



adapted mesh (7,105 elements)
discretization orders: p=1~4

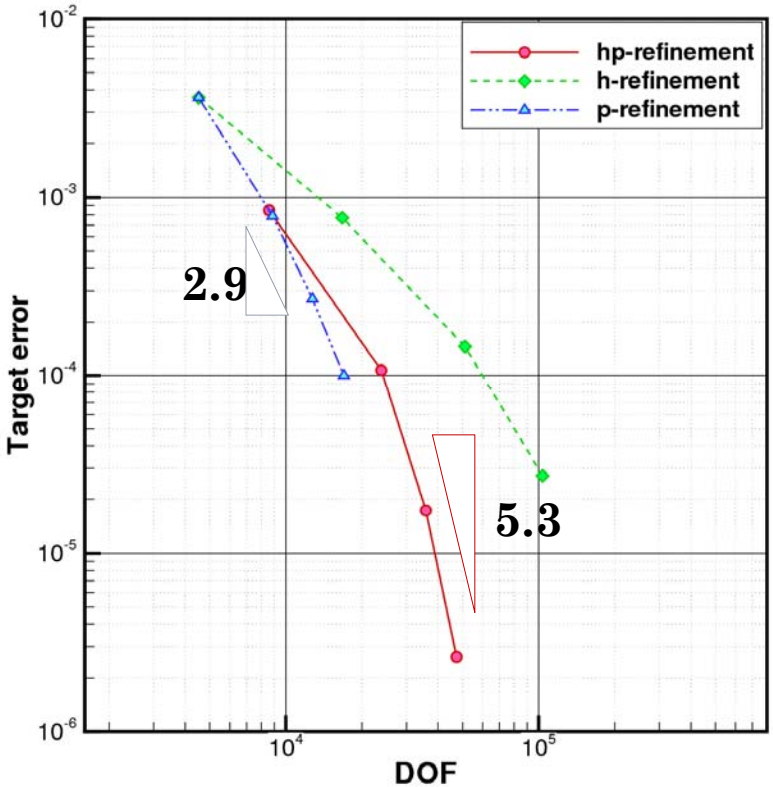


smoothness indicator

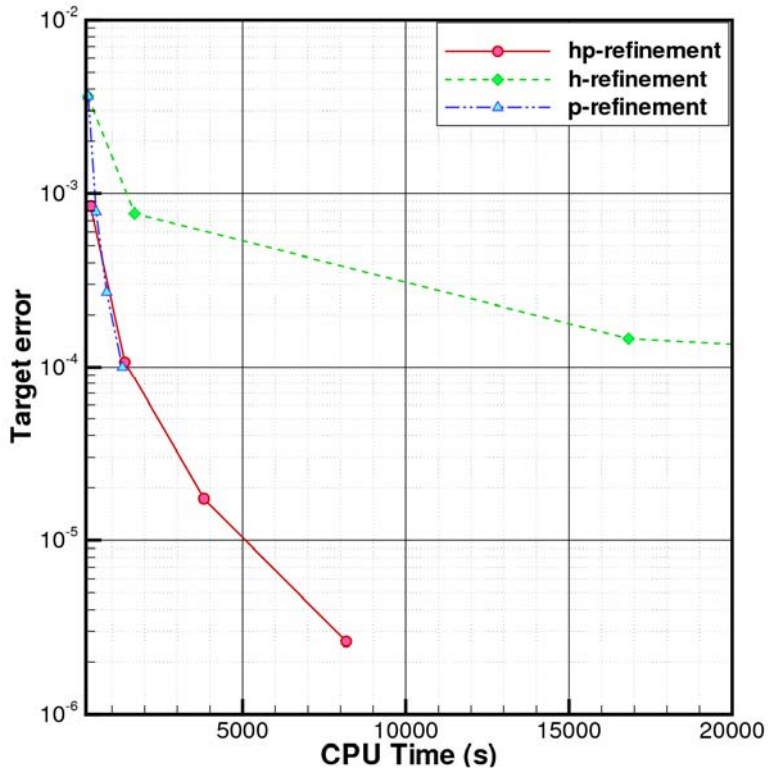
NUMERICAL RESULTS

- Subsonic flow over a four-element airfoil (M_∞=0.2)
  - *hp*-refinement

Target function of drag



error convergence history vs. degrees of freedom

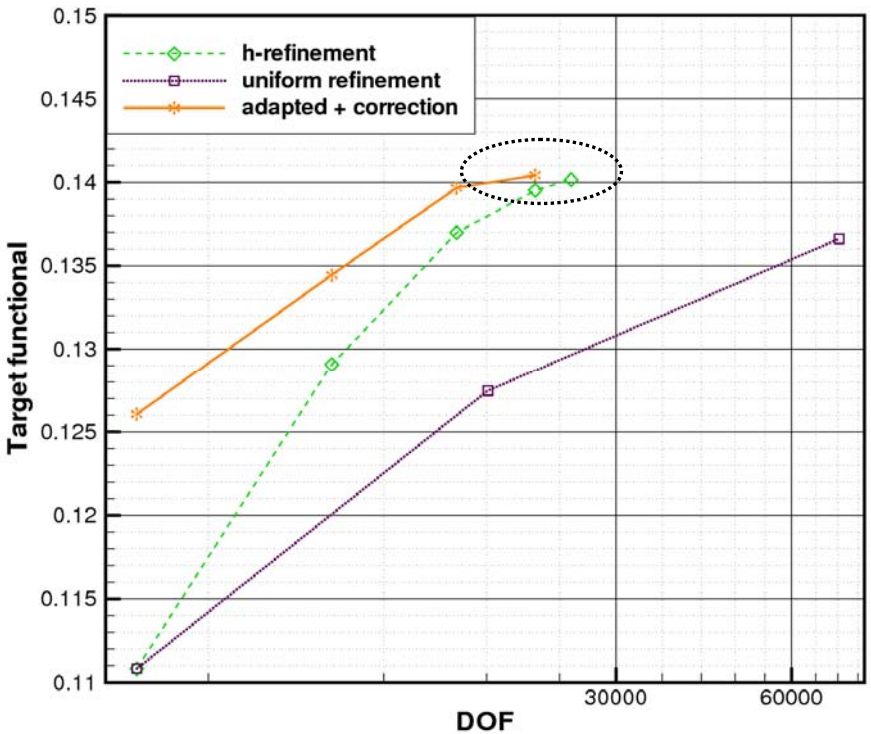


error convergence history vs. CPU time (sec)

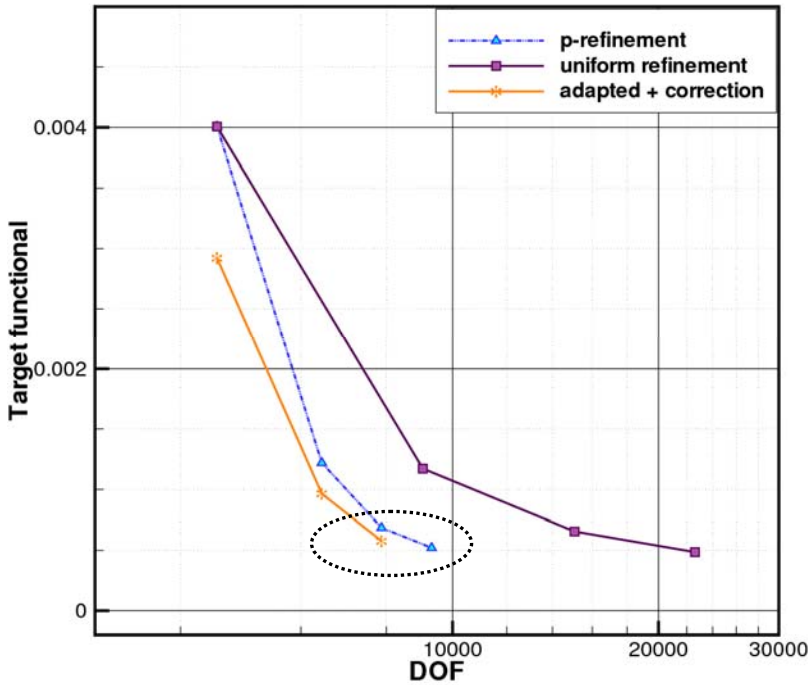
Li Wang, Dimitri Mavriplis (UW)

NUMERICAL RESULTS

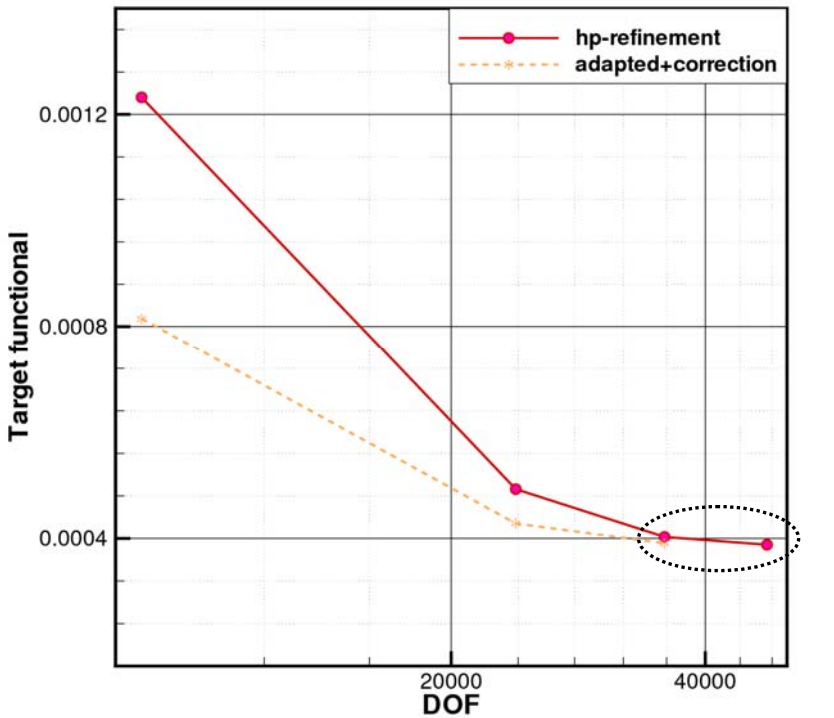
– Comparisons on convergence of objective functionals



h-refinement



p-enrichment



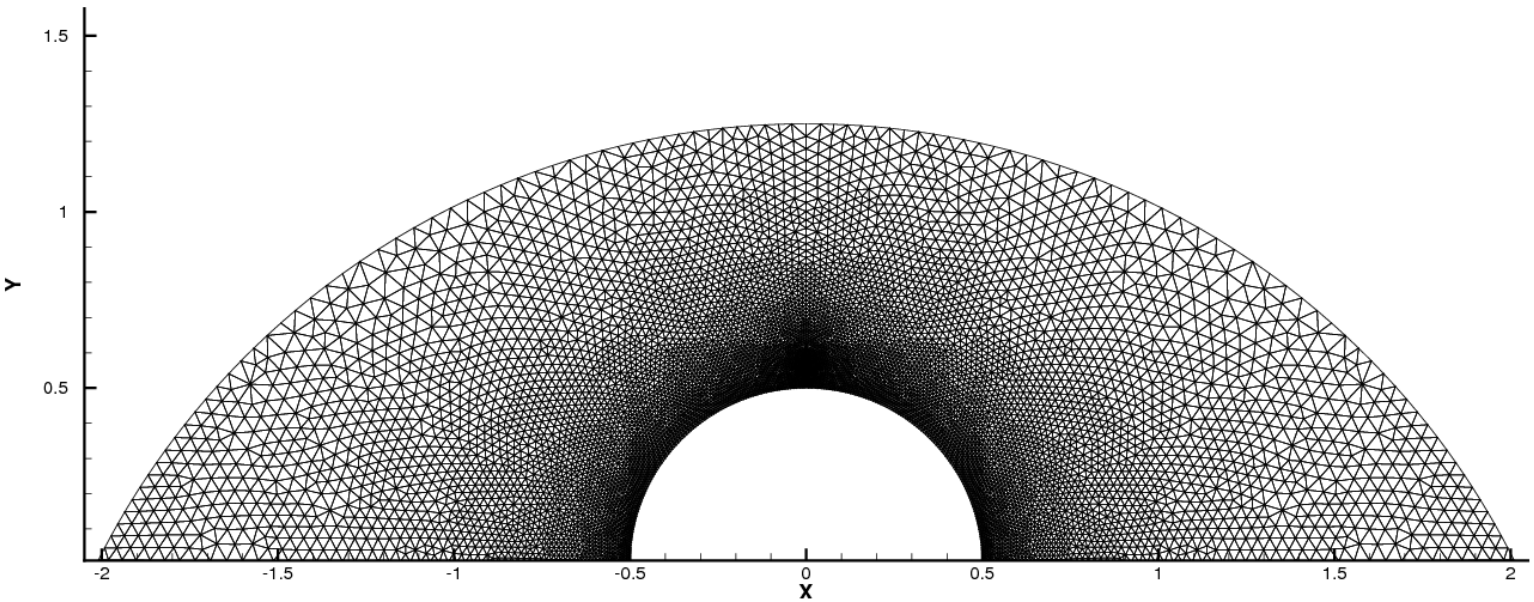
hp-refinement

NUMERICAL RESULTS

- High-speed flow over a half circular-cylinder

  Target function of integrated temperature $J = \int_{\partial\Omega_w} T dS$

  - *hp*-refinement
  - starting discretization order $p = 0$ (first-order accurate)

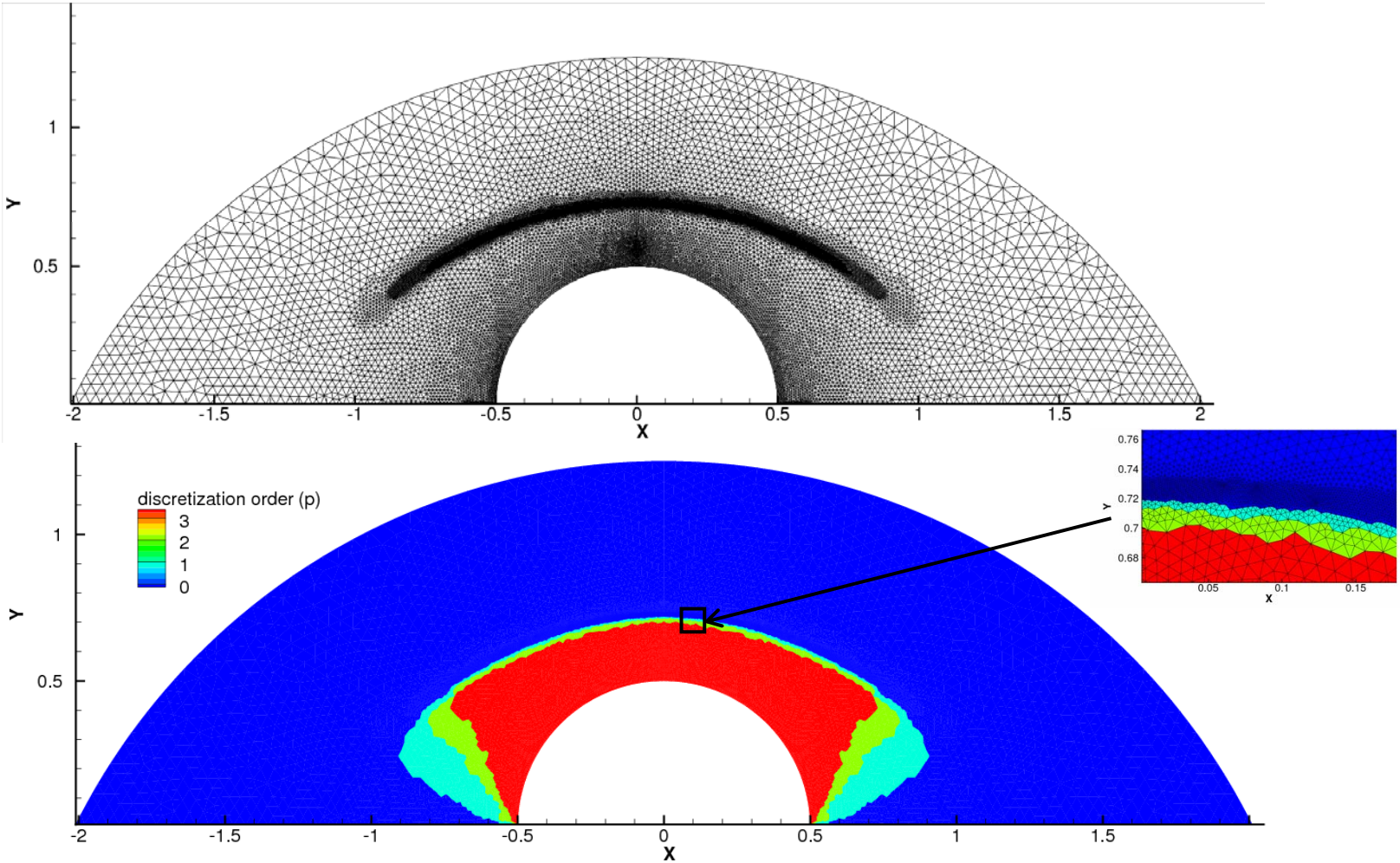


initial mesh: 17,072 elements

Li Wang, Dimitri Mavriplis (UW)

NUMERICAL RESULTS

- High-speed flow over a half circular-cylinder (M_∞=6)



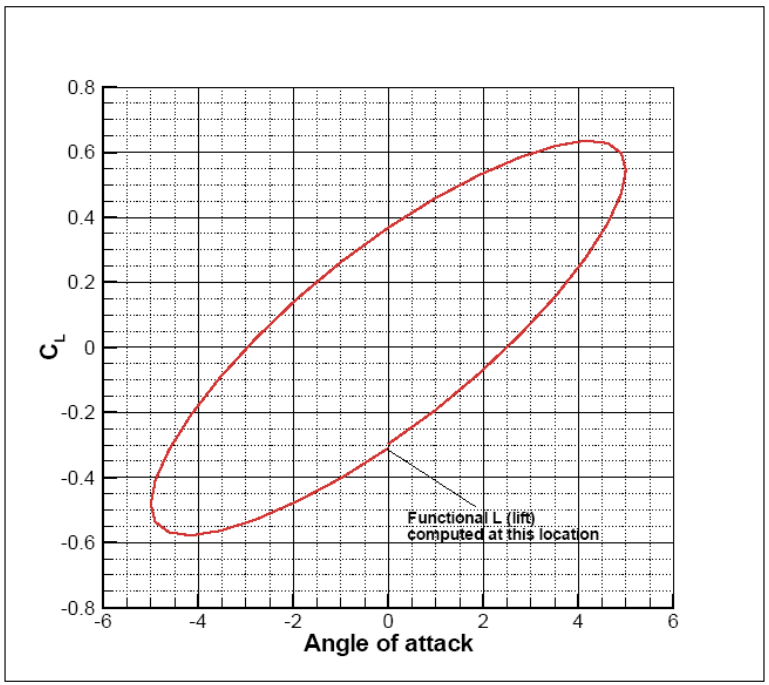
adapted mesh: 42,234 elements,
discretization orders p=0~3

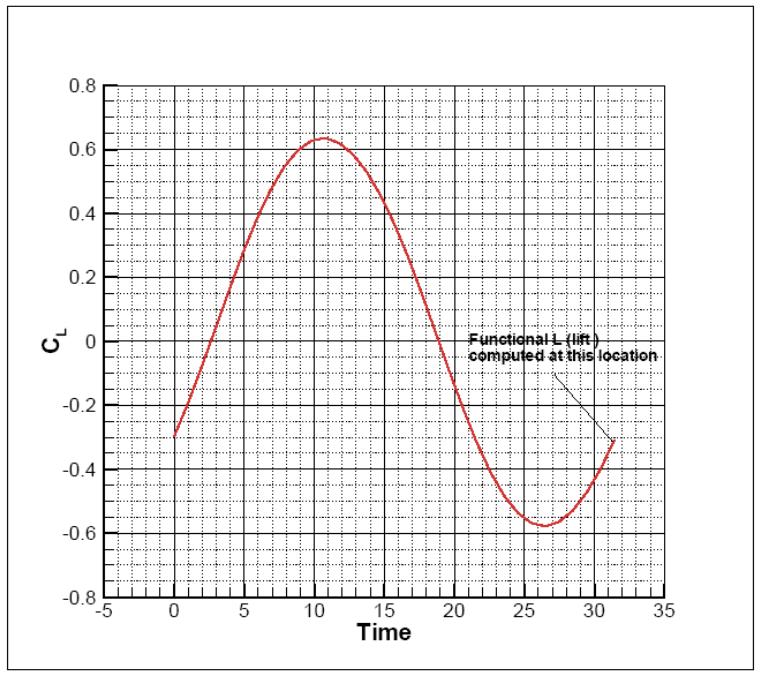
Li Wang, Dimitri Mavriplis (UW)

Error Estimation for Time Dependent Problems

Test Case Description

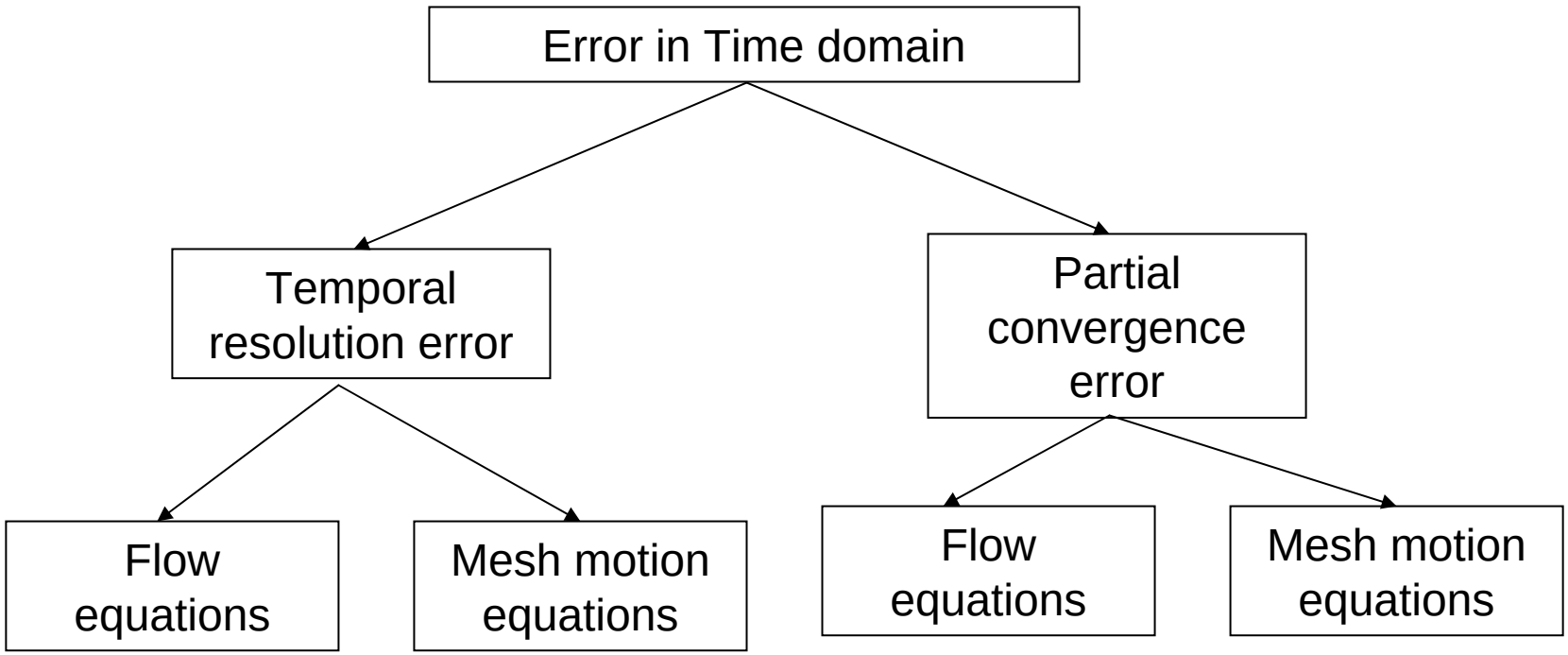
Sinusoidally pitching airfoil
Functional scalar is Lift after 1 period





Easily extended to estimate error in time-integrated Lift history

Sources of Error



Flow equations

Conservative form of Euler equations:

$$\frac{\partial U}{\partial t} + \nabla . F(U) = 0$$

Integrate over moving control volume to get Arbitrary-Lagrangian-Eulerian (ALE) finite-volume form:

$$\frac{\partial AU}{\partial t} + \oint_{dB(t)} [F(U) - \dot{x}U] \cdot \vec{n} dB = 0$$

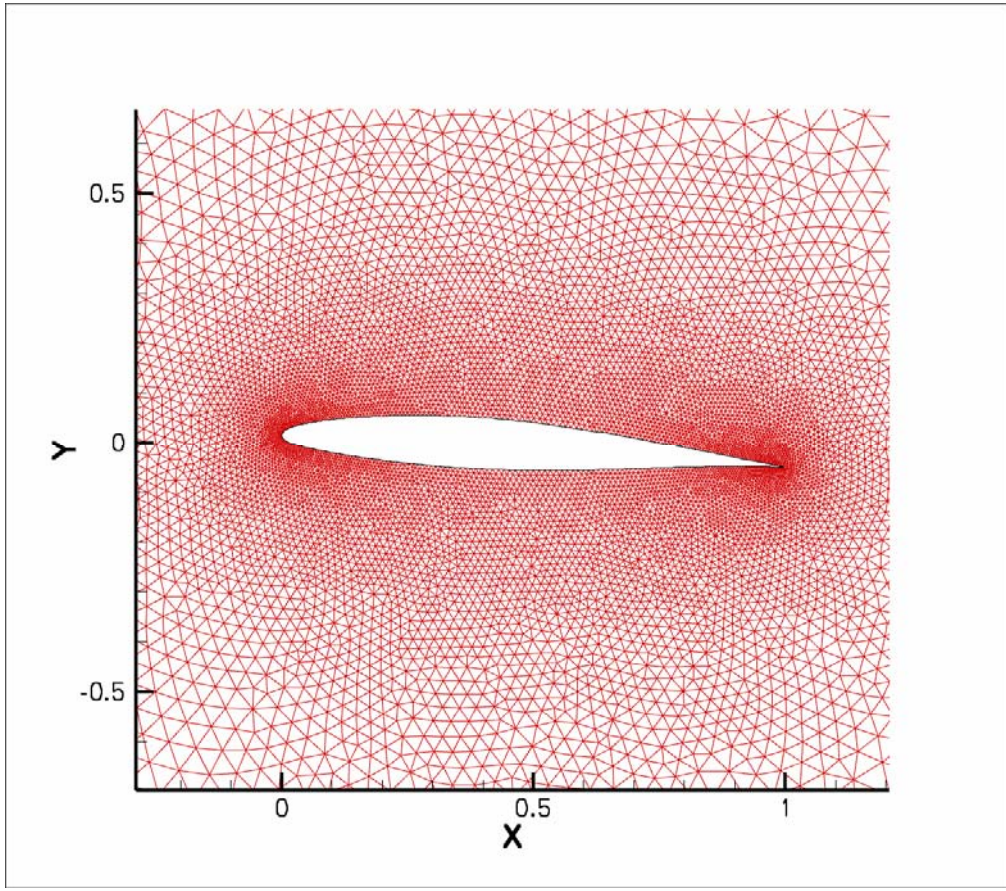
Mesh Deformation

Linear Tension Spring Analogy:

Mesh is a series of interconnected springs

$$[K]\delta x_{int}=\delta x_{surf}$$

2 independent force balance equations at each node



Temporal Resolution Error

Fine level Taylor expansion of functional objective L:

$$L_h(U_h, x_h) =$$

$$L_h(U_h^H, x_h^H) + \left[\frac{\partial L}{\partial U}\right]_{U_h^H x_h^H} (U_h - U_h^H) + \left[\frac{\partial L}{\partial x}\right]_{x_h^H U_h^H} (x_h - x_h^H)$$

h = fine time domain : $\Delta t/2$
H = coarse time domain : Δt

Evaluation of Flow Contribution to Temporal Resolution Error

Fine level flow residual Taylor expansion

$$R_h(U_h, x_h) =$$

$$R_h(U_h^H, x_h^H) + \left[\frac{\partial R}{\partial U}\right]_{U_h^H x_h^H} (U_h - U_h^H) + \left[\frac{\partial R}{\partial x}\right]_{x_h^H U_h^H} (x_h - x_h^H) = 0$$

Continued…

$$(U_h - U_h^H) = -\left[\frac{\partial R}{\partial U}\right]^{-1}_{U_h^H x_h^H} \left\{ R_h(U_h^H, x_h^H) + \left[\frac{\partial R}{\partial x}\right]_{x_h^H U_h^H} (x_h - x_h^H) \right\}$$

$$\left[\frac{\partial L}{\partial U}\right]_{U_h^H x_h^H} (U_h - U_h^H) =$$

$$-\left[\frac{\partial L}{\partial U}\right]_{U_h^H x_h^H} \left[\frac{\partial R}{\partial U}\right]^{-1}_{U_h^H x_h^H} \left\{ R_h(U_h^H, x_h^H) + \left[\frac{\partial R}{\partial x}\right]_{x_h^H U_h^H} (x_h - x_h^H) \right\}$$

$$= -\Lambda^T_{U_h}$$

Continued…

Adjoint equation over entire fine time domain:

$$\left[\frac{\partial R}{\partial U}\right]^{T}_{U_h^H x_h^H} \Lambda_{Uh} = \left[\frac{\partial L}{\partial U}\right]^{T}_{U_h^H x_h^H}$$

Recast on coarse time domain:

$$\left[\frac{\partial R}{\partial U}\right]^{T}_{U_H x_H} \Lambda_{U_H} = \left[\frac{\partial L}{\partial U}\right]^{T}_{U_H x_H}$$

$$\Lambda_{U_h} = I_h^H \Lambda_{U_H}$$

Continued…

$$\Lambda_{Uh}^{T} R_h(U_h^H, x_h^H) + \left\{ \Lambda_{Uh}^{T} \left[\frac{\partial R}{\partial x} \right]_{x_h^H U_h^H} + \left[\frac{\partial L}{\partial x} \right]_{x_h^H U_h^H} \right\} (x_h - x_h^H)$$

Contribution to temporal resolution error from flow equations

$$E_h(x_h - x_h^H)$$

Remaining temporal resolution error
(due to mesh motion equations, but also feeds into flow state)

$$\Lambda_{Uh}^{T} R_h(U_h^H, x_h^H) = \Lambda_1^T R_1 + \Lambda_2^T R_2 + \Lambda_3^T R_3 + \ldots$$

Intepret as sum of dot products of adjoint and residual at each time step

Evaluation of Mesh contribution to Temporal Resolution Error

$$G(x) = [K]\delta x - \delta x_{surf} = 0$$

$$G(x_h) = G(x_h^H) + \left[\frac{\partial G}{\partial x}\right]_{x_h^H} (x_h - x_h^H) = 0$$

$$(x_h - x_h^H) = -[K]^{-1} G(x_h^H)$$

$$E_h(x_h - x_h^H) = -E_h[K]^{-1} G(x_h^H)$$

$$= \Lambda_{x_h}^T$$

Continued…

$$[K]^T \Lambda_{x_h} = -E_h^T$$

$$[K]^T \Lambda_{xH} = -E_H^T$$

$$\Lambda_{xh} = -I_h^H \Lambda_{xH}$$

$$\Lambda_{x_h}^T G_h(x_h^H)$$

Contribution temporal resolution error from mesh motion equations

$$\Lambda_{U_h}^T R_h(U_h^H, x_h^H)$$

Contribution temporal resolution error from flow equations

Summary of Temporal Resolution Error Evaluation

- Compute unsteady flow solution on coarse time domain
- Compute adjoint variables on coarse time domain
  - Integrating backward in time
- Project adjoint variables, flow solution and mesh solution onto fine time domain
- Temporal resolution error is then inner product of adjoint with corresponding non-zero residual on fine time domain
  - Distribution in time is used to drive adaptation

Validation

Description	Functional value	% Error vs. Target	% Predicted Error
Target functional - exact at 32 steps (fully converged)	-0.309957065250867	-	-
Fully converged flow and mesh at 16 steps	-0.285768366164898	+7.804	+7.463
Corrected for resolution from 16 to 32 steps	-0.3089006774509025	+0.341	-

•Adjoint is linearization about current state (16 time steps) to predict objective value on modified state (32 time steps)

Partial Convergence Error

Coarse level Taylor expansion about partial solution functional:

$$L_H(U_H, x_H) =$$

$$L_H(\overline{U}_H, \overline{x}_H) + \left[\frac{\partial L}{\partial U}\right]_{\overline{U}_h^H \overline{x}_h^H} (U_H - \overline{U}_H) + \left[\frac{\partial L}{\partial x}\right]_{\overline{x}_h^H \overline{U}_h^H} (x_H - \overline{x}_H)$$

$\overline{U}_H, \overline{x}_H =$ Partially converged flow and mesh solution

$U_H, x_H =$ Fully converged flow and mesh solution

Partial Convergence Error

Fine level flow residual Taylor expansion

$$R_H(U_H, x_H) =$$

$$R_H(\overline{U}_H, \overline{x}_H) + \left[\frac{\partial R}{\partial U}\right]_{\overline{U}_h^H \overline{x}_h^H} (U_H - \overline{U}_H) + \left[\frac{\partial R}{\partial x}\right]_{\overline{x}_h^H \overline{U}_h^H} (x_H - \overline{x}_H)$$

non-zero due to partial convergence

Continued…

$$\left[\frac{\partial R}{\partial U}\right]^T_{\bar{U}_H \bar{x}_H} \Lambda_{U_H} = \left[\frac{\partial L}{\partial U}\right]^T_{\bar{U}_H \bar{x}_H}$$

$$\left[K\right]^T_{\bar{x}_H \bar{U}_H} \Lambda_{xH} = -E^T_{\bar{x}_H \bar{U}_H}$$

Same adjoint equations as those for temporal resolution error

Partial convergence error due to flow equations:

$$\Lambda^T_{U_H} R_H(\bar{U}_H, \bar{x}_H)$$

Partial convergence error due to mesh equations:

$$\Lambda^T_{xH} G_H(\bar{x}_H)$$

Non-zero residuals due to partial convergence

Summary of Total Error Evaluation and Decomposition

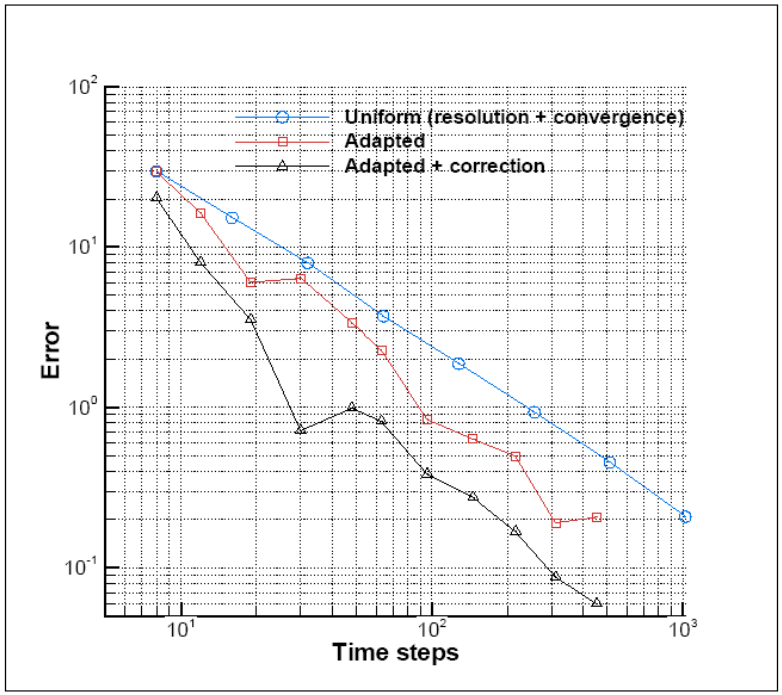
- Compute partially converged flow and mesh solution on coarse time domain
- Compute adjoint variables on coarse domain using partially converged solution
- Compute partial convergence error on coarse level time domain
  - Inner product of adjoint with partially converged (non-zero) residual
- Project partially converged solution and adjoint variables onto fine time domain
- Evaluate fine level error estimate as previously
  - Combined temporal resolution and partial convergence error
- Determine temporal resolution error by subtracting partial convergence error from total error estimate on fine time domain

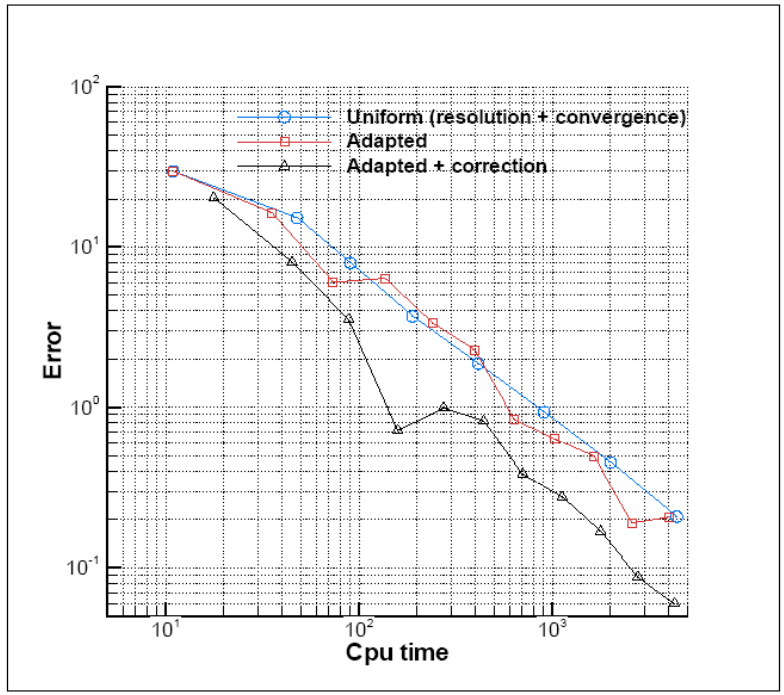
Adaptation

- Compute time-integrated averages of error component distributions

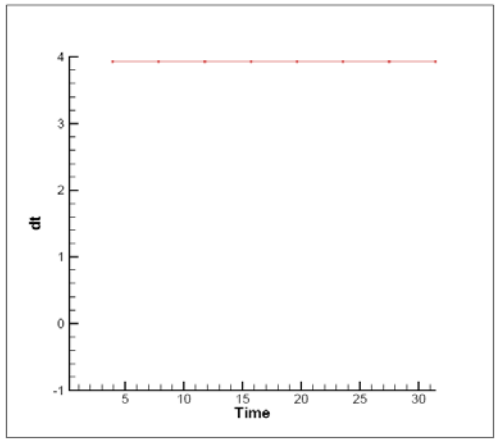
- Adapt where error is greater than time-integrated average
  - Time resolution error: divide time step by 2
  - Convergence error: tighten tolerance by predetermined factor

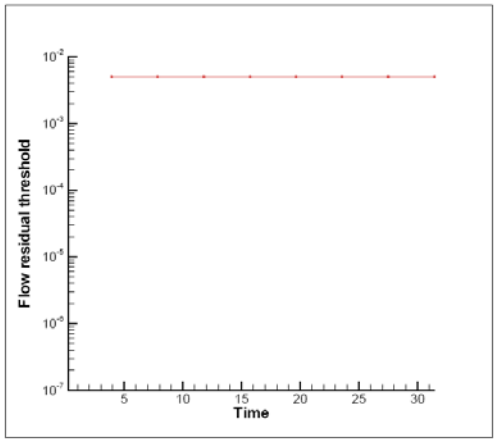
Adaptive Time Step and Convergence Criteria Example

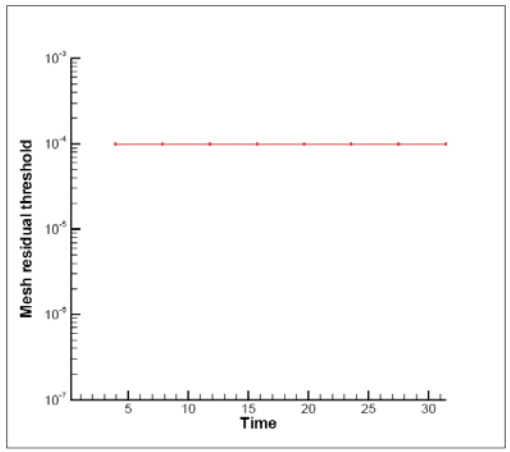


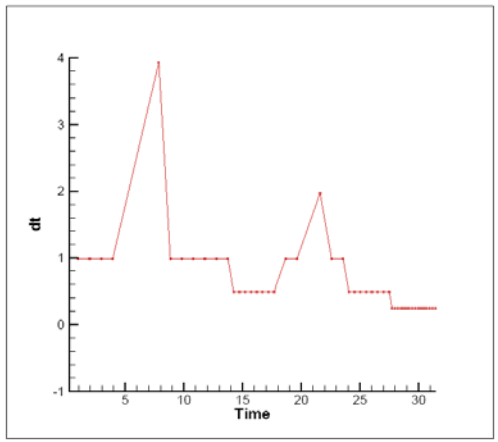


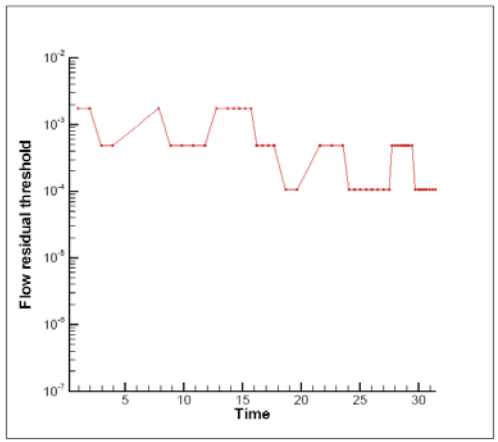
Distribution of time steps and convergence limits in the time domain after 6 adaptation cycles

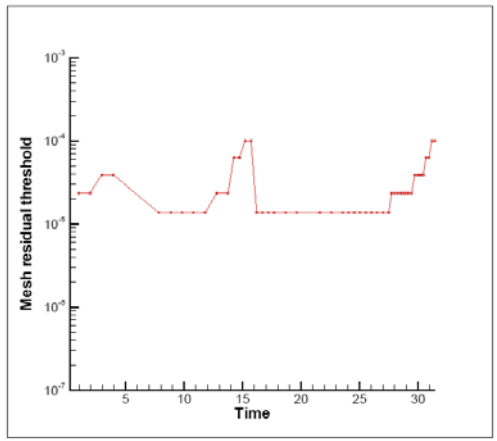












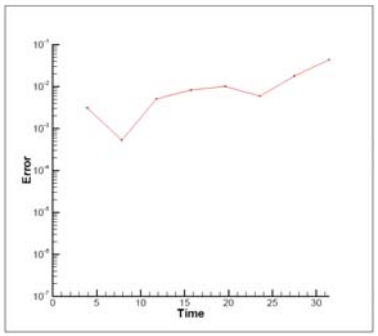
Time steps

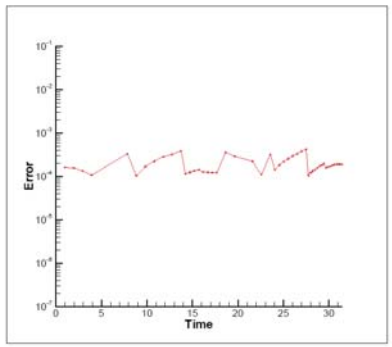
Flow limits

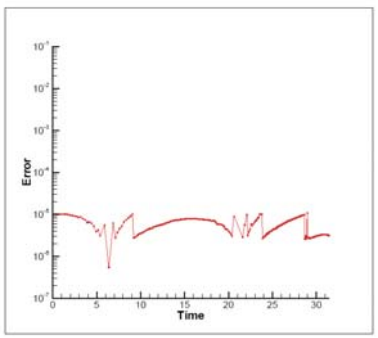
Mesh limits

Distribution of Error Components

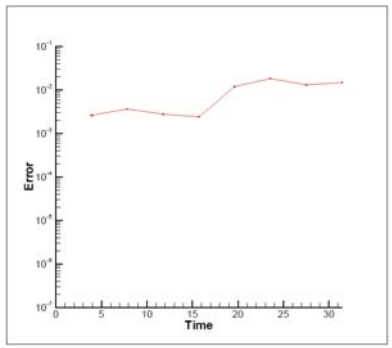
Resolution error

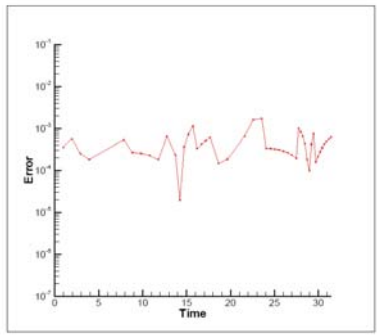


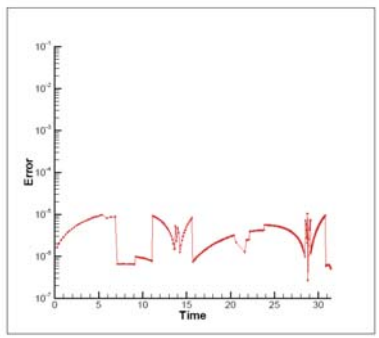




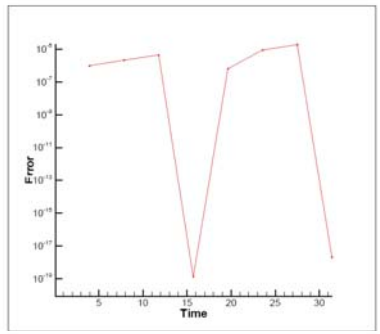
Flow convergence error

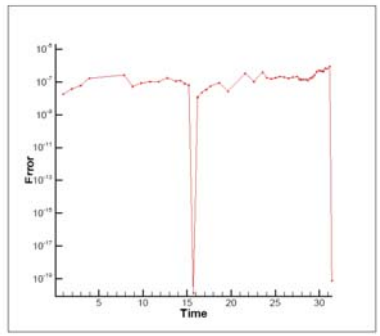


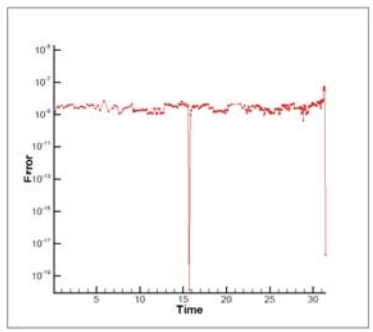




Mesh convergence error







Conclusions

- Rigorous procedure for determining global error
- Identifies individual error contributions from each set of governing equations
- Distribution of each component available and can be used for adaptation
- Can be extended to problems involving multiple sets of governing equations such as conjugate heat transfers, structural equations etc.
- Future Work
  - Temporal adaptivity work extended to BDF2
  - Temporal adaptivity for DG methods
  - Combined Spatial – Temporal Adaptivity
  - Multi-physics and coupling error sensitivity
  - Parameter sensitivities