Accurate and Efficient Simulation and Design Using High-Order CFD Methods

Dr. Li Wang and Dr. W. Kyle Anderson

SimCenter: National Center for Computational Engineering
University of Tennessee at Chattanooga
Chattanooga, Tennessee, USA

July 9, 2014

Modern Techniques for Aerodynamic Analysis and Design
2014 CFD Summer School, Beijing, China, July 7-11, 2014

A Brief Overview

1. High-Order Discontinuous Galerkin Discretizations and Implicit Schemes
2. Multigrid Solution Acceleration Strategies
3. Adjoint-Based Mesh Adaptation and Shape Optimization
4. Simulation of Turbulence Using High-Order Discontinuous Galerkin Methods

Outline (Lecture 3)

1. Background & Motivation
2. Model Problem and Discretizations
3. Adjoint-based Error Estimation and Mesh Adaptation
   - Spatial Error Estimation and Adaptive Meshing
   - Temporal Error Estimation and Time-Step Adaptation
   - Numerical Experiments
4. Sensitivity Analysis and Shape Optimization
   - Mesh Parameterization and Deformation
   - Adjoint-based Sensitivity Derivative Formulation
   - Numerical Experiments
5. Conclusions

Outline (Lecture 3)

Flow Analysis

- Computational methods
- Understand flow fields
- Predict critical situations
- High-order CFD methods

Sensitivity Analysis

- Determine the impacts of simulation inputs to outputs
- Provide search directions for minimizing or maximizing an objective functional

Coupling of Flow Analysis and Sensitivity Analysis

- Output-based error estimation and adaptive mesh refinement
- Shape/design optimization

Background & Motivation

- Sensitivity analysis techniques
  - Obtain sensitivity derivatives
  - Change in objective w.r.t. change in simulation inputs
  - Finite-difference and tangent methods
  - Adjoint method: Linearization of the analysis and transpose
- Adjoint-based adaptive discontinuous Galerkin (DG) methods
  - Hold great promise to guarantee and improve solution accuracy
  - Provide an efficient and robust computational process
  - Mesh adaptivity: Local mesh refinement
- Design optimization
  - Flexibility and efficiency
  - Minimization or maximization of a design target functional
  - Commonly applied for second-order finite volume methods
  - Needs for derivation of sensitivity formulation for high-order schemes

Outline (Lecture 3)

SimCenter
National Center for Computational Engineering

Model Problem

- Governing equations: Euler or Navier-Stokes (NS) equations
  - Conservation of mass (continuity):

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u}{\partial x} + \frac{\partial \rho v}{\partial y} + \frac{\partial \rho w}{\partial z} = 0$$

  - Conservation of momentum:

$$\frac{\partial \rho u}{\partial t} + \frac{\partial (\rho u^2 + p)}{\partial x} + \frac{\partial \rho uv}{\partial y} + \frac{\partial \rho uw}{\partial z} - \frac{\partial \tau_{xx}}{\partial x} - \frac{\partial \tau_{xy}}{\partial y} - \frac{\partial \tau_{xz}}{\partial z} = 0$$
$$\frac{\partial \rho v}{\partial t} + \frac{\partial \rho uv}{\partial x} + \frac{\partial (\rho v^2 + p)}{\partial y} + \frac{\partial \rho vw}{\partial z} - \frac{\partial \tau_{xy}}{\partial x} - \frac{\partial \tau_{yy}}{\partial y} - \frac{\partial \tau_{yz}}{\partial z} = 0$$
$$\frac{\partial \rho w}{\partial t} + \frac{\partial \rho uw}{\partial x} + \frac{\partial \rho vw}{\partial y} + \frac{\partial (\rho w^2 + p)}{\partial z} - \frac{\partial \tau_{xz}}{\partial x} - \frac{\partial \tau_{yz}}{\partial y} - \frac{\partial \tau_{zz}}{\partial z} = 0$$

  - Conservation of energy:

$$\frac{\partial \rho E}{\partial t} + \frac{\partial (\rho E + p) u}{\partial x} + \frac{\partial (\rho E + p) v}{\partial y} + \frac{\partial (\rho E + p) w}{\partial z} - \frac{\partial (u\tau_{xx} + v\tau_{xy} + w\tau_{xz} + \kappa \frac{\partial T}{\partial x})}{\partial x}$$
$$- \frac{\partial (u\tau_{xy} + v\tau_{yy} + w\tau_{yz} + \kappa \frac{\partial T}{\partial y})}{\partial y} - \frac{\partial (u\tau_{xz} + v\tau_{yz} + w\tau_{zz} + \kappa \frac{\partial T}{\partial z})}{\partial z} = 0$$

High-Order Discontinuous Galerkin Discretizations

- The weighted residual formulation

$$\sum_k \int_{\Omega_k} \phi_j \left[\frac{\partial \mathbf{U}_h(\mathbf{x},t)}{\partial t} + \nabla \cdot (\mathbf{F}_e(\mathbf{U}_h) - \mathbf{F}_v(\mathbf{U}_h, \nabla \mathbf{U}_h)) \right] d\Omega_k = 0$$

- Integrate by parts and Implement a symmetric interior penalty method

$$\begin{aligned}
&\int_{\Omega_k} \phi_j \frac{\partial \mathbf{U}_h}{\partial t} d\Omega_k - \int_{\Omega_k} \nabla \phi_j \cdot (\mathbf{F}_e(\mathbf{U}_h) - \mathbf{F}_v(\mathbf{U}_h, \nabla_h \mathbf{U}_h))\, d\Omega_k + \int_{\partial\Omega_k \setminus \partial\Omega} [[\underline{\phi_j}]] \mathbf{H}_c(\mathbf{U}_h^+, \mathbf{U}_h^-, \mathbf{n}) dS \\
&- \int_{\partial\Omega_k \setminus \partial\Omega} \{\mathbf{F}_v(\mathbf{U}_h, \nabla_h \mathbf{U}_h)\} \cdot [[\phi_j]] dS - \int_{\partial\Omega_k \setminus \partial\Omega} \{(\mathbf{G}_{i1} \frac{\partial \phi_j}{\partial \mathbf{x}_i}, \mathbf{G}_{i2} \frac{\partial \phi_j}{\partial \mathbf{x}_i}, \mathbf{G}_{i3} \frac{\partial \phi_j}{\partial \mathbf{x}_i})\} \cdot [[\mathbf{U}_h]] dS + \int_{\partial\Omega_k \setminus \partial\Omega} \eta \{\mathbf{G}\}[[\mathbf{U}_h]] \cdot [[\phi_j]] dS \\
&- \int_{\partial\Omega_k \cap \partial\Omega} \phi_j^+ \mathbf{F}_v^b(\mathbf{U}_b, \nabla_h \mathbf{U}_h^+) \cdot \mathbf{n} dS - \int_{\partial\Omega_k \cap \partial\Omega} (\mathbf{G}_{i1}(\mathbf{U}_b) \frac{\partial \phi_j^+}{\partial \mathbf{x}_i}, \mathbf{G}_{i2}(\mathbf{U}_b) \frac{\partial \phi_j^+}{\partial \mathbf{x}_i}, \mathbf{G}_{i3}(\mathbf{U}_b) \frac{\partial \phi_j^+}{\partial \mathbf{x}_i}) \cdot (\mathbf{U}_h^+ - \mathbf{U}_b) \mathbf{n} dS \\
&+ \int_{\partial\Omega_k \cap \partial\Omega} \eta \mathbf{G}(\mathbf{U}_b)(\mathbf{U}_h^+ - \mathbf{U}_b)\mathbf{n} \cdot \phi_j^+ \mathbf{n} dS + \int_{\partial\Omega_k \cap \partial\Omega} \phi_j \mathbf{F}_e(\mathbf{U}_b) \cdot \mathbf{n} dS = 0
\end{aligned}$$

where $\mathbf{G}_{1j} = \partial \mathbf{F}_v^x / \partial(\partial \mathbf{U}/\partial \mathbf{x}_j)$, $\mathbf{G}_{2j} = \partial \mathbf{F}_v^y / \partial(\partial \mathbf{U}/\partial \mathbf{x}_j)$ and $\mathbf{G}_{3j} = \partial \mathbf{F}_v^z / \partial(\partial \mathbf{U}/\partial \mathbf{x}_j)$

- Solution expansion and geometric mapping

$$\mathbf{U}_h = \sum_{i=1}^{M} \tilde{\mathbf{U}}_{h_i} \; \phi_i(\xi, \eta, \zeta) \qquad \mathbf{x}_k = \sum_{i=1}^{M} \tilde{\mathbf{x}}_{k_i} \phi_i(\xi, \eta, \zeta)$$

- Implicit time-integration schemes

Outline (Lecture 3)

1. Background & Motivation
2. Model Problem and Discretizations
3. Adjoint-based Error Estimation and Mesh Adaptation
   - Spatial Error Estimation and Adaptive Meshing
   - Temporal Error Estimation and Time-Step Adaptation
   - Numerical Experiments
4. Sensitivity Analysis and Shape Optimization
   - Mesh Parameterization and Deformation
   - Adjoint-based Sensitivity Derivative Formulation
   - Numerical Experiments
5. Conclusions

Spatial Error Estimation

- Some key functional outputs with engineering applications
  - Surface integrals of the flow-field variables
  - Lift, drag, integrated surface temperature, etc.
  - Single objective functional, L
- Coarse affordable mesh, H
  - Coarse level flow solution $\tilde{\mathbf{U}}_H$
  - Coarse level functional $L_H(\tilde{\mathbf{U}}_H)$
- Fine (Globally refined) mesh, h
  - Fine level flow solution $\tilde{\mathbf{U}}_h$
  - Fine level functional $L_h(\tilde{\mathbf{U}}_h)$
  - Not desired
- Goal: Find an approximation of $L_h(\tilde{\mathbf{U}}_h)$ without solving on the fine mesh.

Spatial Error Estimation

- Goal: Find an approximation of $L_h(\tilde{\mathbf{U}}_h)$ without solving on the fine mesh.
- Taylor series expansion for the fine level functional

$$L_h(\tilde{\mathbf{U}}_h) = L_h(\tilde{\mathbf{U}}_H^h) + \left(\frac{\partial L_h}{\partial \tilde{\mathbf{U}}_h}\right)_{\tilde{\mathbf{U}}_H^h} (\tilde{\mathbf{U}}_h - \tilde{\mathbf{U}}_H^h) + \cdots$$

- Taylor expansion for the fine level residual

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

- Approximation of the solution error

$$\tilde{\mathbf{U}}_h - \tilde{\mathbf{U}}_H^h \approx -\left[\frac{\partial \mathbf{R}_h}{\partial \tilde{\mathbf{U}}_h}\right]^{-1}_{\tilde{\mathbf{U}}_H^h} \mathbf{R}_h(\tilde{\mathbf{U}}_H^h)$$

- Expression to approximate the fine level functional

$$L_h(\tilde{\mathbf{U}}_h) \approx L_h(\tilde{\mathbf{U}}_H^h) - \left(\frac{\partial L_h}{\partial \tilde{\mathbf{U}}_h}\right)_{\tilde{\mathbf{U}}_H^h} \left[\frac{\partial \mathbf{R}_h}{\partial \tilde{\mathbf{U}}_h}\right]^{-1}_{\tilde{\mathbf{U}}_H^h} \mathbf{R}_h(\tilde{\mathbf{U}}_H^h)$$

Spatial Error Estimation

- Approximation of the fine level functional

$$L_h(\tilde{\mathbf{U}}_h) \approx L_h(\tilde{\mathbf{U}}_H^h) - \underbrace{\left(\frac{\partial L_h}{\partial \tilde{\mathbf{U}}_h}\right)_{\tilde{\mathbf{U}}_H^h} \left[\frac{\partial \mathbf{R}_h}{\partial \tilde{\mathbf{U}}_h}\right]^{-1}_{\tilde{\mathbf{U}}_H^h}}_{(\boldsymbol{\lambda}_h)^T_{\tilde{\mathbf{U}}_H^h}} \mathbf{R}_h(\tilde{\mathbf{U}}_H^h)$$

- Fine level adjoint problem

$$\left[\frac{\partial \mathbf{R}_h}{\partial \tilde{\mathbf{U}}_h}\right]^T_{\tilde{\mathbf{U}}_H^h} (\boldsymbol{\lambda}_h)_{\tilde{\mathbf{U}}_H^h} = \left(\frac{\partial L_h}{\partial \tilde{\mathbf{U}}_h}\right)^T_{\tilde{\mathbf{U}}_H^h}$$

- Instead, we formulate the coarse level adjoint problem

$$\left[\frac{\partial \mathbf{R}_H}{\partial \tilde{\mathbf{U}}_H}\right]^T \boldsymbol{\lambda}_H = \left(\frac{\partial L_H}{\partial \tilde{\mathbf{U}}_H}\right)^T$$

  - Linear system
  - Transpose of Jacobian matrix used in the implicit schemes
  - Delivers similar convergence rate as the flow solver

Spatial Error Estimation

- Reconstruction of coarse level adjoint through least-square methods

$$I\left((\boldsymbol{\lambda}_H^h)_{i_k}\right) = \sum_{l \in \mathcal{P}_k} \left\| \sum_{j=1}^{M^*} (\boldsymbol{\lambda}_H^h)_{j_k} \phi_j|_l - \sum_{j=1}^{M} (\boldsymbol{\lambda}_H)_{j_k} \phi_j|_l \right\|_{L_2}^2, \quad i = 1, \cdots, M^*$$

$$\frac{\partial I\left((\boldsymbol{\lambda}_H^h)_{i_k}\right)}{\partial (\boldsymbol{\lambda}_H^h)_{j_k}} = 0 \quad i, j = 1, \cdots, M^*$$

  - Reconstructed fine-adjoint approximation $(\boldsymbol{\lambda}_H^h)_{j_k}$
  - M^* and M denote the number of basis functions for the fine and current level spaces respectively.

- Fine-level functional approximation

$$L_h(\tilde{\mathbf{U}}_h) \approx L_h(\tilde{\mathbf{U}}_H^h) \underbrace{-(\boldsymbol{\lambda}_H^h)^T \mathbf{R}_h(\tilde{\mathbf{U}}_H^h)}_{\varepsilon_a} \underbrace{-((\boldsymbol{\lambda}_h)_{\tilde{\mathbf{U}}_H^h} - (\boldsymbol{\lambda}_H^h))^T \mathbf{R}_h(\tilde{\mathbf{U}}_H^h)}_{\varepsilon_r}$$

  - ε_a and ε_r denote the computable error correction and the remaining error, respectively.
  - ε_r is usually at least an order of magnitude smaller than ε_a.

Spatial Error Estimation

- Functional error approximation

$$L_h(\tilde{\mathbf{U}}_h) \approx L_h(\tilde{\mathbf{U}}_H^h) - (\boldsymbol{\lambda}_H^h)^T \mathbf{R}_h(\tilde{\mathbf{U}}_H^h)$$

- Functional error correction ε_c between fine and coarse levels

$$\underbrace{L_h(\mathbf{U}_h) - L_H(\mathbf{U}_H)}_{\varepsilon_c} \approx \underbrace{L_h(\mathbf{U}_H^h) - L_H(\mathbf{U}_H)}_{\varepsilon_d} \underbrace{-(\boldsymbol{\lambda}_H^h)^T \mathbf{R}_h(\tilde{\mathbf{U}}_H^h)}_{\varepsilon_a}$$

- Major computational cost for each adaptation cycle
  - The flow problem and the adjoint problem on the coarse (current) level

Spatial Error Estimation

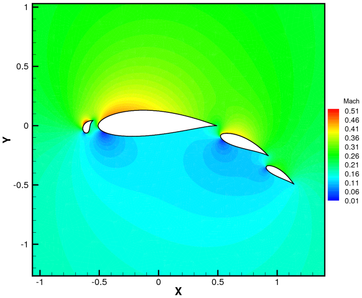
- Functional error approximation

$$L_h(\tilde{\mathbf{U}}_h) \approx L_h(\tilde{\mathbf{U}}_H^h) - (\boldsymbol{\lambda}_H^h)^T \mathbf{R}_h(\tilde{\mathbf{U}}_H^h)$$

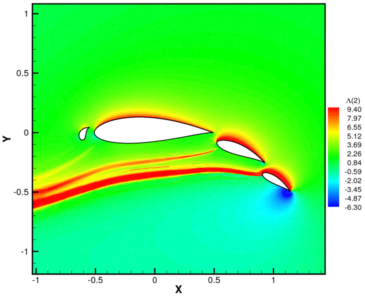
- Functional error correction ε_c between fine and coarse levels

$$\underbrace{L_h(\mathbf{U}_h) - L_H(\mathbf{U}_H)}_{\varepsilon_c} \approx \underbrace{L_h(\mathbf{U}_H^h) - L_H(\mathbf{U}_H)}_{\varepsilon_d} \underbrace{-(\boldsymbol{\lambda}_H^h)^T \mathbf{R}_h(\tilde{\mathbf{U}}_H^h)}_{\varepsilon_a}$$

- Major computational cost for each adaptation cycle
  - The flow problem and the adjoint problem on the coarse (current) level



Mach number solution



x-momentum component of adjoint

Refinement Criteria

$$\underbrace{L_h(\mathbf{U}_h) - L_H(\mathbf{U}_H)}_{\varepsilon_c} \approx \underbrace{L_h(\mathbf{U}_H^h) - L_H(\mathbf{U}_H)}_{\varepsilon_d} \underbrace{-(\boldsymbol{\lambda}_H^h)^T \mathbf{R}_h(\tilde{\mathbf{U}}_H^h)}_{\varepsilon_a}$$

- Spatial functional error estimator ε_a

$$\varepsilon_a = \sum_{k=1}^{n} \varepsilon_{a,k}$$

- Element-wise error indicator

$$\varepsilon_{a,k} = -(\boldsymbol{\lambda}_H^h)_k^T \mathbf{R}_{h,k}(\tilde{\mathbf{U}}_H^h)$$

- Refinement criteria

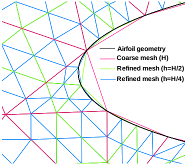
$$|\varepsilon_{a,k}| > \frac{E_{tol}}{N}$$

  - E_{tol} is a prescribed error tolerance.
  - N denotes the number of elements in $\mathcal{T}_H$.
  - No further refinement is performed if the error is equidistributed in $\mathcal{T}_H$.

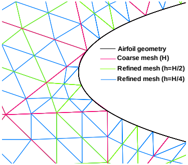
- Flag elements required for refinement

Mesh Refinement for High-Order DG Methods

- $h-$refinement
  - Refine the flagged element by adding nodes at the midpoint (keeping p fixed).
  - Newly added nodes must conform to the original geometry.

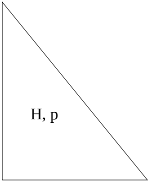


$h-$refinement ($p = 1$)

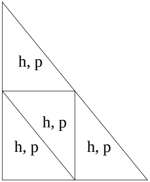


$h-$refinement ($p = 4$)

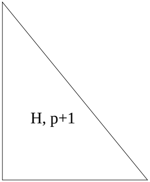
- $p-$refinement
  - Increase the discretization order $p \to p + 1$ (fixing the mesh).
- $hp-$refinement
  - Local implementation of the $h-$ or $p-$refinement individually



Coarse mesh



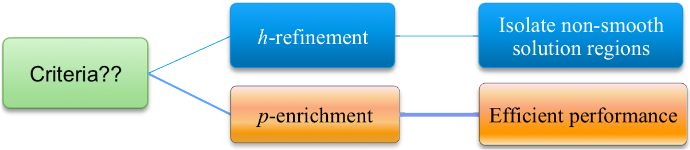
Subdivided mesh



Enriched mesh

Extra Criteria for *hp*−refinement

- For each flagged element: how to make a decision on $h-$ or $p-$refinement?



- Local smoothness indicator
  - Element-based Resolution indicator [Persson and Peraire]

$$S_k = \frac{\int_{\Omega_k} (q - \breve{q})^2 d\Omega_k}{\int_{\Omega_k} q^2 d\Omega_k} \qquad q = \sum_{i=1}^{M_p} \tilde{q}_i \ \phi_i \qquad \breve{q} = \sum_{i=1}^{M_{p-1}} \tilde{q}_i \ \phi_i$$

  - Inter-element Jump indicator [Krivodonova and Xin et al.]

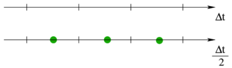
$$S_k = \frac{1}{|\partial\Omega_k|} \int_{\partial\Omega_k} \left| \frac{q^+ - q^-}{\frac{1}{2}(q^+ + q^-)} \right| dS$$

Temporal Error Estimation

- The same methodology for determining global functional error can be extended to unsteady flow problems.
  - Predict temporal error for a specified time-dependent objective functional
  - Identify temporal error distributions for discretizations in the time domain
  - Apply an adaptive time-step refinement procedure

- A time-dependent objective functional of interest, e.g. time-integrated lift or drag

$$L^f(\tilde{\mathbf{U}}) = \int_0^T L(\tilde{\mathbf{U}})dt$$



  - Coarse time-level functional (Δt_H) $\quad L_H^f = \int_0^T L_H(\tilde{\mathbf{U}}_H)dt$
  - Fine time-level functional ($\Delta t_h = \Delta t_H/2$) $\quad L_h^f = \int_0^T L_h(\tilde{\mathbf{U}}_h)dt$

- Taylor series expansion with respect to the projected $\tilde{\mathbf{U}}_H^h$

$$L_h^f(\tilde{\mathbf{U}}_h) = L_h^f(\tilde{\mathbf{U}}_H^h) \; + \; \left(\frac{\partial L_h^f}{\partial \tilde{\mathbf{U}}_h}\right)_{\tilde{\mathbf{U}}_H^h} (\tilde{\mathbf{U}}_h - \tilde{\mathbf{U}}_H^h) + \cdots$$

$$\mathbf{R}_{e\,h}(\tilde{\mathbf{U}}_h) = \mathbf{R}_{e\,h}(\tilde{\mathbf{U}}_H^h) \; + \; \left[\frac{\partial \mathbf{R}_{e\,h}}{\partial \tilde{\mathbf{U}}_h}\right]_{\tilde{\mathbf{U}}_H^h} (\tilde{\mathbf{U}}_h - \tilde{\mathbf{U}}_H^h) + \cdots \; = 0$$

Unsteady Adjoint Formulation

- Estimation of the temporal functional error

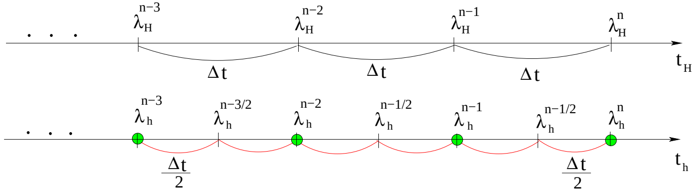
$$L_h^f(\tilde{\mathbf{U}}_h) - L_H^f(\tilde{\mathbf{U}}_H) \approx L_h^f(\tilde{\mathbf{U}}_H^h) - L_H^f(\tilde{\mathbf{U}}_H) \underbrace{-(\boldsymbol{\lambda}_H^h)^T \mathbf{R}_{e\,h}(\tilde{\mathbf{U}}_H^h)}_{\varepsilon_a^t}$$

- Unsteady adjoint problem
  - Coarse time resolution $\quad \left[\dfrac{\partial \mathbf{R}_{e\,H}}{\partial \tilde{\mathbf{U}}_H}\right]_{\tilde{\mathbf{U}}_H}^T \boldsymbol{\lambda}_H = \left(\dfrac{\partial L_H^f}{\partial \tilde{\mathbf{U}}_H}\right)_{\tilde{\mathbf{U}}_H}^T$
  - Matrix form of the unsteady adjoint for BDF1 scheme

$$\begin{bmatrix}
\left[\dfrac{\partial \mathbf{R_e}_H^1}{\partial \tilde{\mathbf{U}}_H^1}\right]^T & \left[\dfrac{\partial \mathbf{R_e}_H^2}{\partial \tilde{\mathbf{U}}_H^1}\right]^T & 0 & \cdots & 0 & 0 \\
0 & \left[\dfrac{\partial \mathbf{R_e}_H^2}{\partial \tilde{\mathbf{U}}_H^2}\right]^T & \left[\dfrac{\partial \mathbf{R_e}_H^3}{\partial \tilde{\mathbf{U}}_H^2}\right]^T & 0 & 0 & 0 \\
0 & 0 & \ddots & \ddots & 0 & 0 \\
\vdots & \vdots & & \ddots & \ddots & 0 \\
0 & 0 & \cdots & & \left[\dfrac{\partial \mathbf{R_e}_H^{n-1}}{\partial \tilde{\mathbf{U}}_H^{n-1}}\right]^T & \left[\dfrac{\partial \mathbf{R_e}_H^{n}}{\partial \tilde{\mathbf{U}}_H^{n-1}}\right]^T \\
0 & 0 & \cdots & \cdots & 0 & \left[\dfrac{\partial \mathbf{R_e}_H^{n}}{\partial \tilde{\mathbf{U}}_H^{n}}\right]^T
\end{bmatrix}_{\tilde{\mathbf{U}}_H}
\begin{bmatrix} \boldsymbol{\lambda}_H^1 \\ \boldsymbol{\lambda}_H^2 \\ \vdots \\ \boldsymbol{\lambda}_H^{n-1} \\ \boldsymbol{\lambda}_H^n \end{bmatrix}
=
\begin{bmatrix} \dfrac{\partial L_h^f}{\partial \tilde{\mathbf{U}}_H^1}^T \\ \dfrac{\partial L_H^f}{\partial \tilde{\mathbf{U}}_H^2}^T \\ \vdots \\ \dfrac{\partial L_H^f}{\partial \tilde{\mathbf{U}}_H^{n-1}}^T \\ \dfrac{\partial L_H^f}{\partial \tilde{\mathbf{U}}_H^n}^T \end{bmatrix}_{\tilde{\mathbf{U}}_H}$$

Distributed Functional Temporal Error

- Project coarse time-level adjoint solution to the fine time level: $\boldsymbol{\lambda}_H \rightarrow \boldsymbol{\lambda}_H^h$



$$L_h^f(\tilde{\mathbf{U}}_h) - L_H^f(\tilde{\mathbf{U}}_H) \approx L_h^f(\tilde{\mathbf{U}}_H^h) - L_H^f(\tilde{\mathbf{U}}_H) \underbrace{-(\boldsymbol{\lambda}_H^h)^T \mathbf{R}_{e\,h}(\tilde{\mathbf{U}}_H^h)}_{\varepsilon_a^t}$$

- Functional temporal error distribution in time step, i
  - Backward difference schemes

$$\varepsilon_{a,i}^t = -(\boldsymbol{\lambda}_H^h)^{i\,T} \mathbf{R}_{e\,h}^{\;i}(\tilde{\mathbf{U}}_H^h)$$

  - Multistage Runge-Kutta schemes

$$\varepsilon_{a,i}^t = -\sum_{s=2}^{\mathcal{S}} (\boldsymbol{\lambda}_H^h)^{(s),i\,T} \mathbf{R}_{e\,h}^{\;(s),i}(\tilde{\mathbf{U}}_H^h)$$

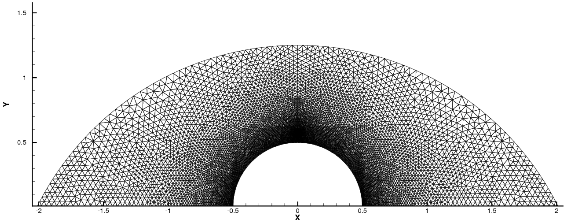
- Refinement criteria for a designed error tolerance, E_{tol}^t

$$|\varepsilon_{a,i}^t| > \frac{E_{tol}^t}{n}$$

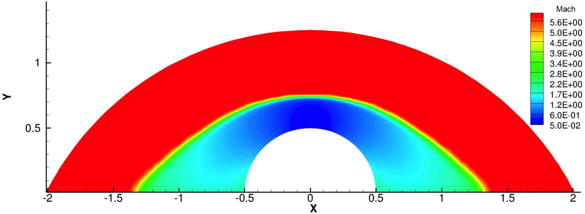
- *hp*-adaptation for hypersonic flow over a half-circular cylinder

hp-Adaptation for Hypersonic Flow over a Half-Circular Cylinder

- Free-stream Mach number of 6
- Objective functional: surface integrated temperature, $L = \int_{\partial\Omega_w} T ds$
- Initial discretization order $p = 0$ (first-order) and *hp*-adaptive mesh refinement



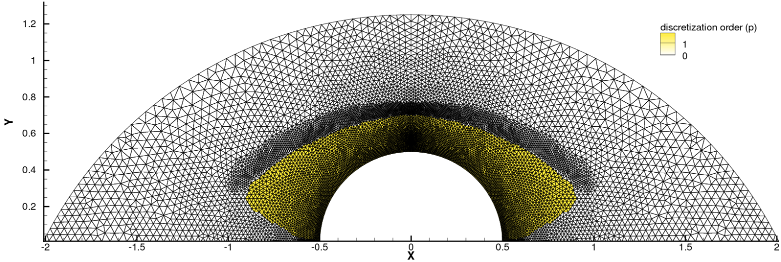
Initial mesh, 17072 elements



Mach number contours

hp-Adapted Meshes

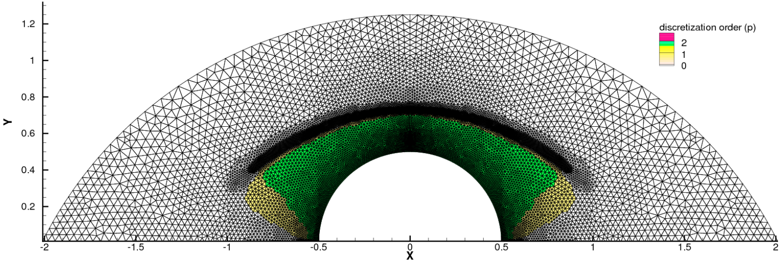
- Hypersonic flow over a half-circular cylinder ($M_\infty = 6$)



1^{st}-adapted mesh, 21372 elements, $p = 0 \sim 1$

hp-Adapted Meshes

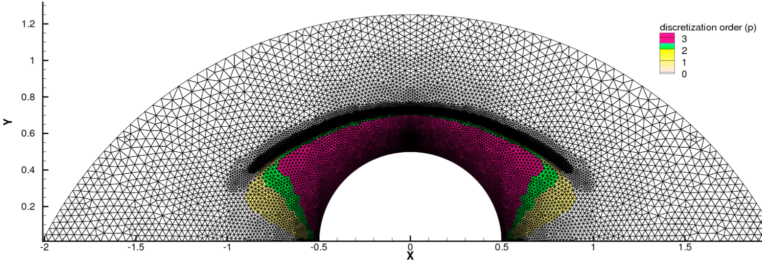
- Hypersonic flow over a half-circular cylinder ($M_\infty = 6$)



2^{nd}-adapted mesh, 29100 elements, $p = 0 \sim 2$

hp-Adapted Meshes

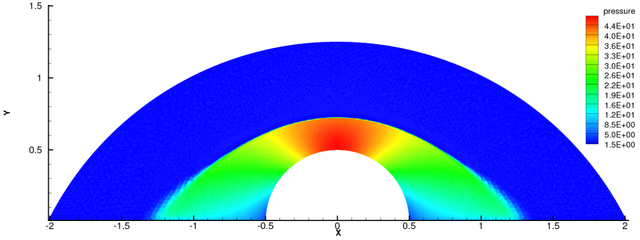
- Hypersonic flow over a half-circular cylinder ($M_\infty = 6$)



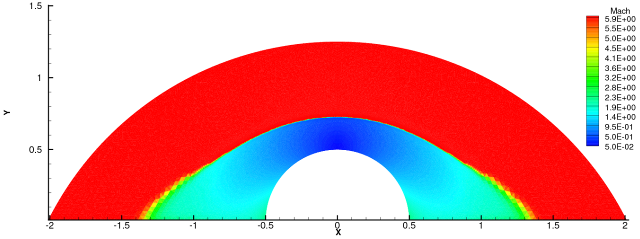
3^{rd}-adapted mesh, 42234 elements, $p = 0 \sim 3$

Hypersonic Flow over a Half-Circular Cylinder

- Solutions on the final adapted mesh



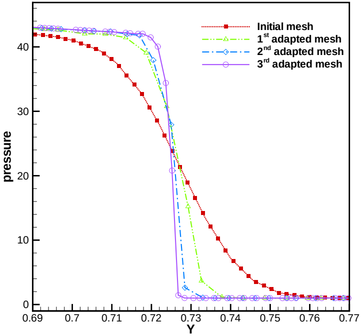
Pressure



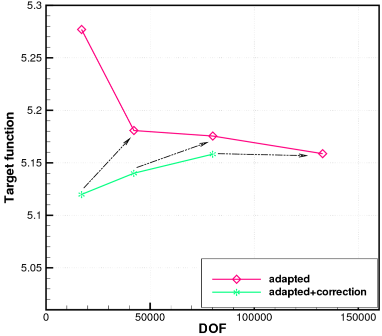
Mach number

Hypersonic Flow over a Half-Circular Cylinder

- Pressure profile across the shock



- Convergence of the objective functional



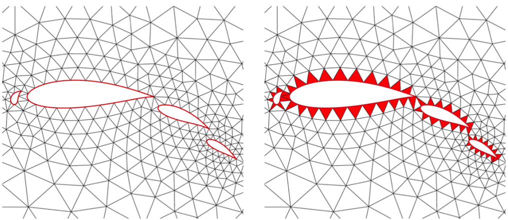
- Shock is resolved 10 times thinner on the final adapted mesh.
- Predicted fine-level functional is progressively more accurate.
- Sharp shock is captured without the use of solution filtering or artificial dissipation technique.

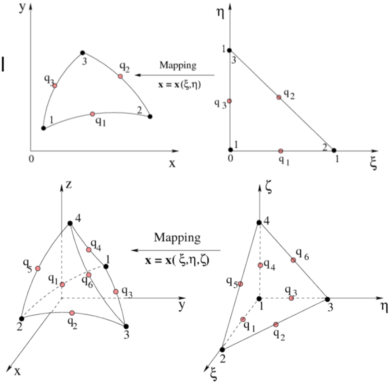
Outline (Lecture 3)

Mesh Parameterization and Deformation

- The use of high-order curved elements
  - Necessary for overall high-accuracy solution
  - Inviscid and viscous meshes
  - Geometric mapping from reference to physical

$$\mathbf{x}_k = \sum_{i=1}^{M} \tilde{\mathbf{x}}_{k_i} \phi_i(\xi, \eta, \zeta)$$





2D & 3D Mappings

- Triangular elements with straighted-sided edges: **Linear mapping**

$$\mathbf{x}_k = \tilde{\mathbf{x}}_{k_1}\phi_1(\xi,\eta) + \tilde{\mathbf{x}}_{k_2}\phi_2(\xi,\eta) + \tilde{\mathbf{x}}_{k_3}\phi_3(\xi,\eta)$$

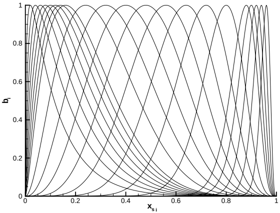
- Triangular elements with curved edges: **Nonlinear mapping**

$$\mathbf{x}_k = \tilde{\mathbf{x}}_{k_1}\phi_1(\xi,\eta)+\tilde{\mathbf{x}}_{k_2}\phi_2(\xi,\eta)+\tilde{\mathbf{x}}_{k_3}\phi_3(\xi,\eta)+\tilde{\mathbf{x}}_{k_4}\phi_4(\xi,\eta) + \tilde{\mathbf{x}}_{k_5}\phi_5(\xi,\eta) + \cdots + \tilde{\mathbf{x}}_{k_M}\phi_M(\xi,\eta)$$

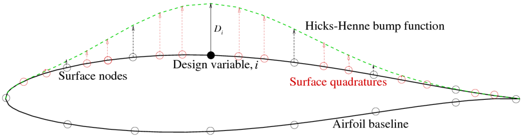
Mesh Parameterization and Deformation

- Design variables, $\mathbf{D} = \{D_i\}$
  - Set as the magnitudes of the Hicks-Henne bump function
  - Placed at a set of surface nodes
  - Deformation of surface geometry

$$b_k(x_k, D_i) = D_i sin^4(\pi {x_k}^{\ln 0.5/ln x_i}) \quad x_i, x_k \in [0, 1]$$



Hicks-Henn bump function



Surface perturbations

- Mesh Deformation for Interior Mesh
  - Tension spring analogy $[K]\Delta\mathbf{x} = \Delta\mathbf{x}_s$
- Discrete adjoint formulation
  - Linearization of the discretized system
  - Transpose operation to all matrices and vectors

Adjoint-based Sensitivity Derivative Formulation

- Objective functional

$$L = L(\tilde{\mathbf{x}}(\mathbf{D}), \tilde{\mathbf{U}}(\tilde{\mathbf{x}}(\mathbf{D})))$$

- Forward linearization via chain rule

$$\frac{dL}{d\mathbf{D}} = \left(\frac{\partial L}{\partial \tilde{\mathbf{x}}} + \frac{\partial L}{\partial \tilde{\mathbf{U}}}\frac{\partial \tilde{\mathbf{U}}}{\partial \tilde{\mathbf{x}}}\right)\left(\frac{\partial \tilde{\mathbf{x}}}{\partial \mathbf{x}}\frac{\partial \mathbf{x}}{\partial \mathbf{x}_s}\frac{\partial \mathbf{x}_s}{\partial \mathbf{D}} + \frac{\partial \tilde{\mathbf{x}}}{\partial \mathbf{x}_q}\frac{\partial \mathbf{x}_q}{\partial \mathbf{D}}\right)$$

  - Hicks-Henne bump function
  - Inverse of the projection mapping matrix
  - Mesh motion equations, $[K]^{-1}$
  - Objective functional definition
  - Implicitly defined by the discrete flow equations

- Discrete flow equations (steady or unsteady)

$$\mathbf{R}\left(\tilde{\mathbf{U}}(\tilde{\mathbf{x}}), \tilde{\mathbf{U}}^0(\tilde{\mathbf{x}}), \mathbf{U}^b(\tilde{\mathbf{x}}), \tilde{\mathbf{x}}\right) = 0$$

$$\left(\left[\frac{\partial \mathbf{R}}{\partial \tilde{\mathbf{U}}}\right]\frac{\partial \tilde{\mathbf{U}}}{\partial \tilde{\mathbf{x}}} + \left[\frac{\partial \mathbf{R}}{\partial \tilde{\mathbf{U}}^0}\right]\frac{\partial \tilde{\mathbf{U}}^0}{\partial \tilde{\mathbf{x}}} + \left[\frac{\partial \mathbf{R}}{\partial \mathbf{U}^b}\right]\frac{\partial \mathbf{U}^b}{\partial \tilde{\mathbf{x}}} + \frac{\partial \mathbf{R}}{\partial \tilde{\mathbf{x}}}\right)\frac{\partial \tilde{\mathbf{x}}}{\partial \mathbf{D}} = 0$$

$$\frac{\partial \tilde{\mathbf{U}}}{\partial \tilde{\mathbf{x}}} = -\left[\frac{\partial \mathbf{R}}{\partial \tilde{\mathbf{U}}}\right]^{-1}\frac{\partial \bar{\mathbf{R}}}{\partial \tilde{\mathbf{x}}}$$

Adjoint-based Sensitivity Derivative Formulation

- Objective functional

$$L = L(\tilde{\mathbf{x}}(\mathbf{D}), \tilde{\mathbf{U}}(\tilde{\mathbf{x}}(\mathbf{D})))$$

- Forward linearization via chain rule

$$\frac{dL}{d\mathbf{D}} = \left(\frac{\partial L}{\partial \tilde{\mathbf{x}}} + \frac{\partial L}{\partial \tilde{\mathbf{U}}}\frac{\partial \tilde{\mathbf{U}}}{\partial \tilde{\mathbf{x}}}\right)\left(\frac{\partial \tilde{\mathbf{x}}}{\partial \mathbf{x}}\frac{\partial \mathbf{x}}{\partial \mathbf{x}_s}\frac{\partial \mathbf{x}_s}{\partial \mathbf{D}} + \frac{\partial \tilde{\mathbf{x}}}{\partial \mathbf{x}_q}\frac{\partial \mathbf{x}_q}{\partial \mathbf{D}}\right)$$

  - Hicks-Henne bump function
  - Inverse of the projection mapping matrix
  - Mesh motion equations, $[K]^{-1}$
  - Objective functional definition
  - Implicitly defined by the discrete flow equations

- Discrete flow equations (steady or unsteady)

$$\mathbf{R}\left(\tilde{\mathbf{U}}(\tilde{\mathbf{x}}), \tilde{\mathbf{U}}^0(\tilde{\mathbf{x}}), \mathbf{U}^b(\tilde{\mathbf{x}}), \tilde{\mathbf{x}}\right) = 0$$

$$\left(\left[\frac{\partial \mathbf{R}}{\partial \tilde{\mathbf{U}}}\right]\frac{\partial \tilde{\mathbf{U}}}{\partial \tilde{\mathbf{x}}} + \left[\frac{\partial \mathbf{R}}{\partial \tilde{\mathbf{U}}^0}\right]\frac{\partial \tilde{\mathbf{U}}^0}{\partial \tilde{\mathbf{x}}} + \left[\frac{\partial \mathbf{R}}{\partial \mathbf{U}^b}\right]\frac{\partial \mathbf{U}^b}{\partial \tilde{\mathbf{x}}} + \frac{\partial \mathbf{R}}{\partial \tilde{\mathbf{x}}}\right)\frac{\partial \tilde{\mathbf{x}}}{\partial \mathbf{D}} = 0$$

$$\frac{\partial \tilde{\mathbf{U}}}{\partial \tilde{\mathbf{x}}} = -\left[\frac{\partial \mathbf{R}}{\partial \tilde{\mathbf{U}}}\right]^{-1}\frac{\partial \bar{\mathbf{R}}}{\partial \tilde{\mathbf{x}}}$$

Adjoint-based Sensitivity Derivative Formulation

- Forward linearization formulation becomes

$$\frac{dL}{d\mathbf{D}} = \left(\frac{\partial L}{\partial \tilde{\mathbf{x}}} - \frac{\partial L}{\partial \tilde{\mathbf{U}}} \left[\frac{\partial \mathbf{R}}{\partial \tilde{\mathbf{U}}}\right]^{-1} \frac{\partial \bar{\mathbf{R}}}{\partial \tilde{\mathbf{x}}}\right) \left(\frac{\partial \tilde{\mathbf{x}}}{\partial \mathbf{x}} [K]^{-1} \frac{\partial \mathbf{x}_s}{\partial \mathbf{D}} + \frac{\partial \tilde{\mathbf{x}}}{\partial \mathbf{x}_q} \frac{\partial \mathbf{x}_q}{\partial \mathbf{D}}\right)$$

- Discrete adjoint approach via transpose operation

$$\frac{dL}{d\mathbf{D}}^T = \left(\frac{\partial \mathbf{x}_s}{\partial \mathbf{D}}^T [K]^{-T} \frac{\partial \tilde{\mathbf{x}}}{\partial \mathbf{x}}^T + \frac{\partial \mathbf{x}_q}{\partial \mathbf{D}}^T \frac{\partial \tilde{\mathbf{x}}}{\partial \mathbf{x}_q}^T\right) \left(\frac{\partial L}{\partial \tilde{\mathbf{x}}}^T - \frac{\partial \bar{\mathbf{R}}}{\partial \tilde{\mathbf{x}}}^T \left[\frac{\partial \mathbf{R}}{\partial \tilde{\mathbf{U}}}\right]^{-T} \frac{\partial L}{\partial \tilde{\mathbf{U}}}^T\right)$$

- Flow-adjoint problem

$$\left[\frac{\partial \mathbf{R}}{\partial \tilde{\mathbf{U}}}\right]^{-T} \frac{\partial L}{\partial \tilde{\mathbf{U}}}^T = \boldsymbol{\lambda}_u \quad \text{or} \quad \left[\frac{\partial \mathbf{R}}{\partial \tilde{\mathbf{U}}}\right]^T \boldsymbol{\lambda}_u = \frac{\partial L}{\partial \tilde{\mathbf{U}}}^T$$

- Objective sensitivities

$$\frac{\bar{\partial} L}{\partial \tilde{\mathbf{x}}}^T = \frac{\partial L}{\partial \tilde{\mathbf{x}}}^T - \frac{\partial \bar{\mathbf{R}}}{\partial \tilde{\mathbf{x}}}^T \boldsymbol{\lambda}_u \quad \text{and} \quad \frac{\bar{\partial} L}{\partial \mathbf{x}}^T = \frac{\partial \tilde{\mathbf{x}}}{\partial \mathbf{x}}^T \frac{\bar{\partial} L}{\partial \tilde{\mathbf{x}}}^T \quad \frac{\bar{\partial} L}{\partial \mathbf{x}_q}^T = \frac{\partial \tilde{\mathbf{x}}}{\partial \mathbf{x}_q}^T \frac{\bar{\partial} L}{\partial \tilde{\mathbf{x}}}^T$$

Adjoint-based Sensitivity Derivative Formulation

- Mesh adjoint problem

$$\frac{dL}{d\mathbf{D}}^T = \frac{\partial \mathbf{x}_s}{\partial \mathbf{D}}^T \underbrace{[K]^{-T} \frac{\bar{\partial} L}{\partial \mathbf{x}}^T}_{\boldsymbol{\lambda}_x} + \frac{\partial \mathbf{x}_q}{\partial \mathbf{D}}^T \frac{\bar{\partial} L}{\partial \mathbf{x}_q}^T$$

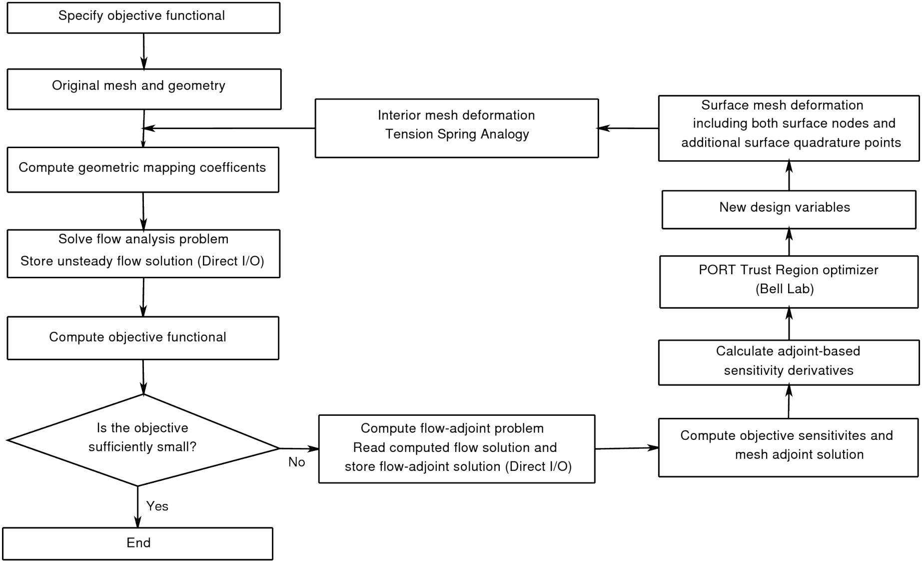
$$[K]^{-T} \frac{\bar{\partial} L}{\partial \mathbf{x}}^T = \boldsymbol{\lambda}_x \quad \text{or} \quad [K]^T \boldsymbol{\lambda}_x = \frac{\bar{\partial} L}{\partial \mathbf{x}}^T$$

- Final formulation for adjont-based sensitivity derivatives

$$\frac{dL}{d\mathbf{D}}^T = \frac{\partial \mathbf{x}_s}{\partial \mathbf{D}}^T \boldsymbol{\lambda}_x + \frac{\partial \mathbf{x}_q}{\partial \mathbf{D}}^T \frac{\bar{\partial} L}{\partial \mathbf{x}_q}^T$$

  - Evaluation of the adjoint-based sensitivity analysis is independent of the number of design variables.
  - Primary computational cost is relevant to solving the flow and flow-adjoint solution.
  - The flow-adjoint problem is also used for the functional error estimation discussed previously.

Shape Optimization Procedure



- Unsteady shape optimization for single airfoil gust response
- Steady-state design optimization for target lift

Shape Optimization for Single Airfoil Gust Response

- Unsteady shape optimization
  - Two-dimensional vorticity gust ($k_1 = k_2 = 5$)

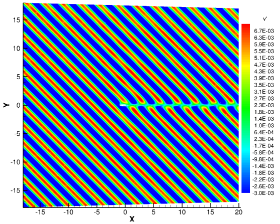
$$u_g = -(\epsilon k_2 M_\infty a_\infty / \sqrt{k_1^2 + k_2^2}) \cos(k_1 x + k_2 y - \omega t)$$

$$v_g = \ (\epsilon k_1 M_\infty a_\infty / \sqrt{k_1^2 + k_2^2}) \cos(k_1 x + k_2 y - \omega t)$$

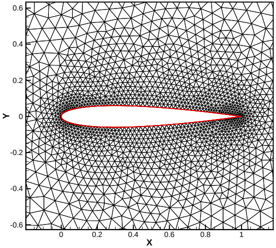
  - Original airfoil: NACA0012 and target airfoil: RAE-2822
  - Mathing unsteady surface pressure

$$L = \sqrt{\frac{\sum_{n=n_s}^{N} \sum_{j=1}^{N_s} \sum_{q=1}^{N_q} \left(p_{q,j}^n - (p_{q,j}^n)^*\right)^2}{N_T \cdot N_s \cdot N_q}}$$

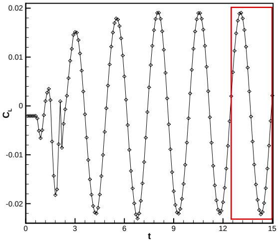
  - DG $p3$ discretization and IRK4 scheme
  - A time-step size of 0.1 and a total of 150 time steps



Y-component velocity disturbance



Original mesh and airfoil geometry (4657 elements)



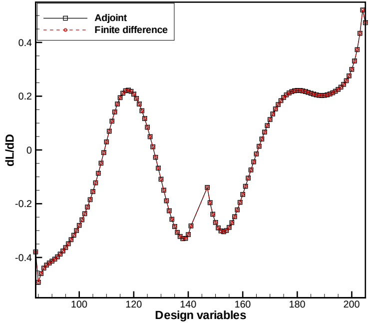
Time history of lift coefficients

Shape Optimization for Single Airfoil Gust Response

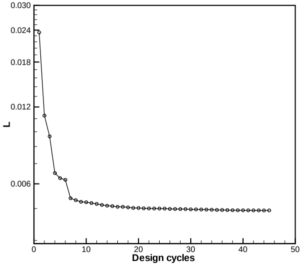
- Comparison of sensitivity derivatives

Design Variable ID	Adjoint	Finite-Difference
91	-**0.4062**8512714012	-**0.4062**3851553511
122	**0.17095**240484446	**0.17095**377499886
136	-**0.3066**6089213105	-**0.3066**1269811299
164	-**0.04423**150309117	-**0.04423**559342156
179	**0.22018**104622794	**0.22018**367947491
204	**0.52079**634064327	**0.52079**549792106

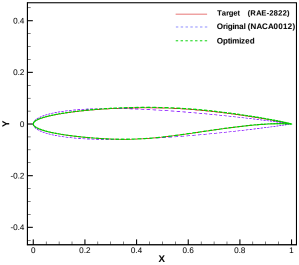
- Full sensitivity vector at the initial design step



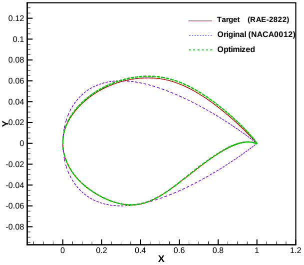
- Shape optimization performance
  - Functional convergence versus the number of design cycles
  - Comparison of the original, target and optimized airfoil geometries



Functional convergence



1:1 scale

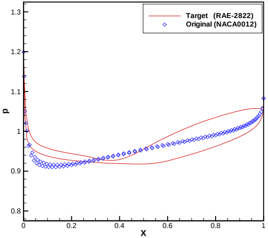


Exaggerated vertical scale

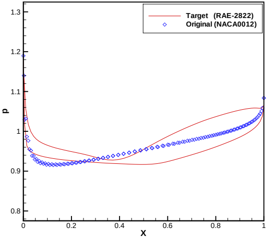
Shape Optimization for Single Airfoil Gust Response

SimCenter National Center for Computational Engineering

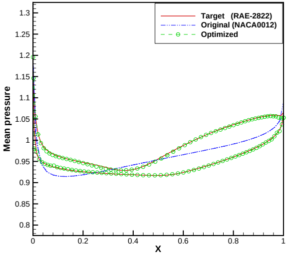
- Comparison of unsteady pressure distributions



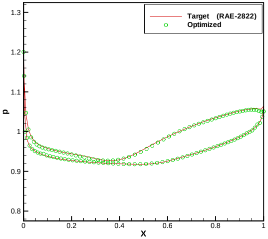
Design step 0, $n = 142$



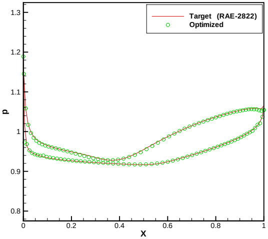
Design step 0, $n = 150$



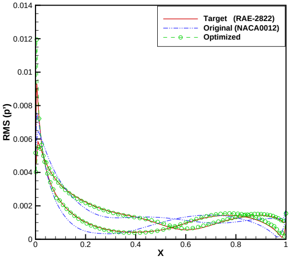
Mean pressure



Design step 45, $n = 142$



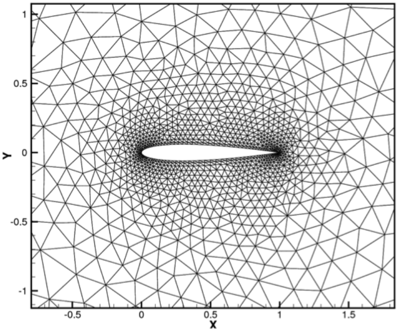
Design step 45, $n = 150$



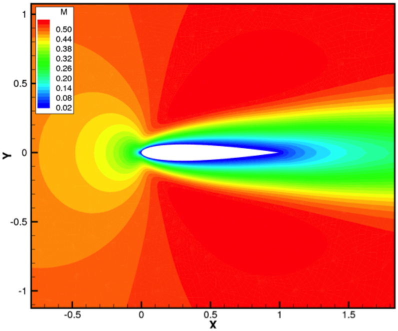
RMS pressure disturbance

Steady-State Design Optimization

- Airfoil shape design
  - Baseline NACA-0012 airfoil
  - Flow conditions: $M_\infty = 0.1$, AOA$= 0^\circ$ and $R_e = 100$
  - Design goal: to obtain a target lift coefficient, $L = (C_L - C_{L,target})^2$
  - Employ a fifth-order DG ($p = 4$) scheme on coarse and fine meshes
  - 57 and 86 design variables (90% chord length) on the airfoil surface



Baseline coarse mesh (2741 triangular elements)



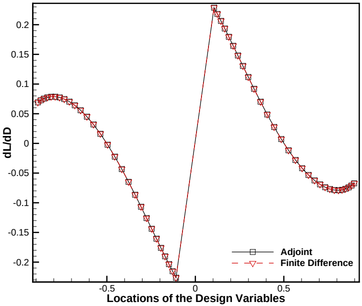
Contours of Mach number

Steady-State Design Optimization

- Comparison of sensitivity derivatives

Design Variables i	Adjoint	Finite Difference	Relative Difference (%)
1	$6.914030138 \times 10^{-2}$	$6.913736532 \times 10^{-2}$	0.004
2	$7.267210215 \times 10^{-2}$	$7.266973392 \times 10^{-2}$	0.003
3	$7.552060283 \times 10^{-2}$	$7.551885877 \times 10^{-2}$	0.002
4	$7.756409954 \times 10^{-2}$	$7.756302806 \times 10^{-2}$	0.001
5	$7.866736070 \times 10^{-2}$	$7.866701625 \times 10^{-2}$	0.0004

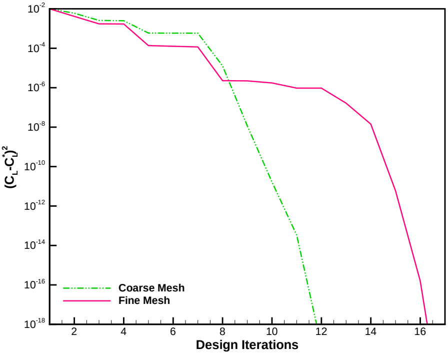
- Full sensitivity vector at the initial design step



Steady-State Design Optimization

SimCenter National Center for Computational Engineering

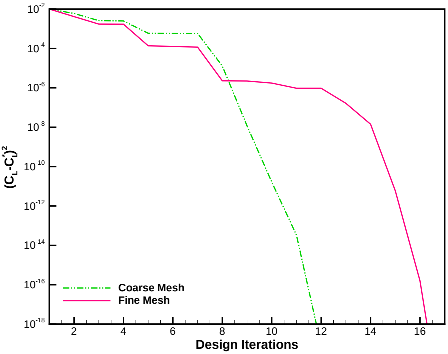
- Airfoil shape design
  - New design to achieve the design goal



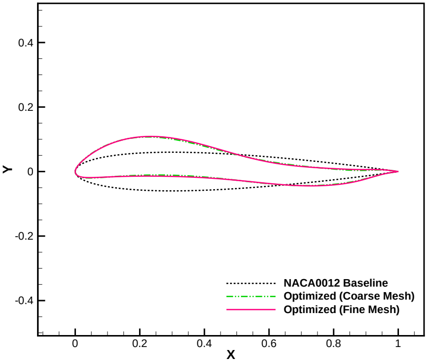
Objective functional convergence

Steady-State Design Optimization

- Airfoil shape design
  - New design to achieve the design goal



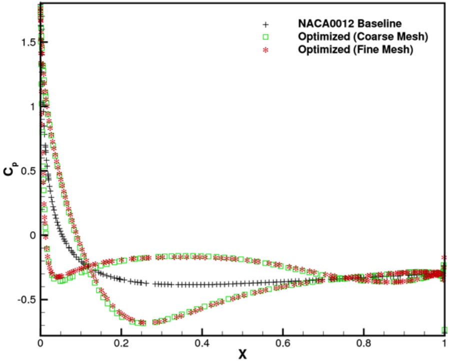
Objective functional convergence



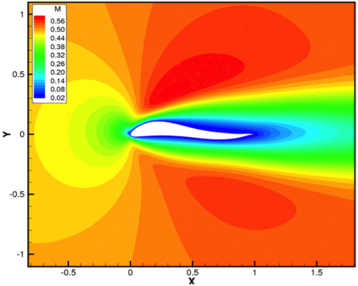
Comparison on surface shapes

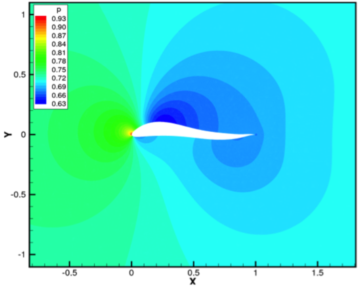
Steady-State Design Optimization

- Airfoil shape design
  - Design results



Distributions of surface pressures





Contours of Mach number and pressure ($p = 4$)

Outline (Lecture 3)

SimCenter National Center for Computational Engineering

1. Background & Motivation
2. Model Problem and Discretizations
3. Adjoint-based Error Estimation and Mesh Adaptation
   - Spatial Error Estimation and Adaptive Meshing
   - Temporal Error Estimation and Time-Step Adaptation
   - Numerical Experiments
4. Sensitivity Analysis and Shape Optimization
   - Mesh Parameterization and Deformation
   - Adjoint-based Sensitivity Derivative Formulation
   - Numerical Experiments
5. Conclusions

Conclusions

- A discrete adjoint method is developed for high-order discontinuous Galerkin methods.
  - Viable approach for adaptive mesh refinement and adaptive time-step refinement.
  - Calculate sensitivity derivatives and drive gradient-based shape optimization.
- The hp-refinement approach exhibits great shock-capturing properties.
- The correction provided by the adjoint-based error estimation is able to predict fine-level functional outputs accurately without solving on a globally refined mesh.
- The evaluation of sensitivity derivatives must account for the mesh sensitivities arising from mesh points and extra quadrature points.
- A similar deformation strategy is performed for surface nodes and additional surface quadrature points.
- The cost of the adjoint-based sensitivity-derivative calculation is independent of the number of design variables, thus being suitable for aerodynamic shape optimization.

References

L. Wang and D.J. Mavriplis, Adjoint-based h-p Adaptive Discontinuous Galerkin Methods for the 2D Compressible Euler Equations, Journal of Computational Physics, 228 (20), pp. 7643-7661, 2009.

P.-O. Persson, J. Peraire, Sub-cell Shock Capturing for Discontinuous Galerkin Methods, AIAA Paper 2006-112, January 2006.

L. Krivodonova, J. Xin, J.-F. Remacle, N. Chevaugeon, J. Flaherty, Shock Detection and Limiting with Discontinuous Galerkin Methods for Hyperbolic Conservation Laws, Appl. Numer. Math. 48 (3) (2004) 323-338.

L. Wang, D.J. Mavriplis and W.K. Anderson, Adjoint Sensitivity Formulation for Discontinuous Galerkin Discretizations in Unsteady Inviscid Flow Problems, AIAA Journal, 48 (12), pp. 2867-2883, 2010.

L. Wang and W.K. Anderson, Shape Sensitivity Analysis for the Compressible Navier-Stokes Equations Via Discontinuous Galerkin Methods, Computers Fluids, 69, pp. 93-107, 2012.

D.J. Mavriplis, A Discrete Adjoint-Based Approach for Optimization Problems on Three-Dimensional Unstructured Meshes, AIAA Paper 2006-0050, Jan. 2006.

L. Wang, D.J. Mavriplis, Implicit Solution of the Unsteady Euler Equations for High-order Accurate Discontinuous Galerkin Discretizations, J. Comput. Phys. 225 (2) (2007) 1994-2015.