Is My CFD Mesh Adequate? A Quantitative Answer

Krzysztof J. Fidkowski

Gas Dynamics Research Colloqium
Aerospace Engineering Department
University of Michigan

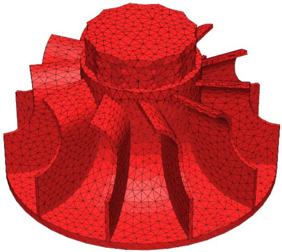
January 26, 2011

Outline

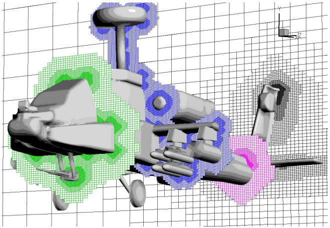
Outline

Meshes for Computational Fluid Dynamics

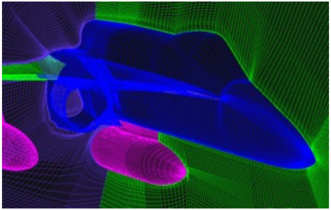
- Various types supporting different discretizations.
- Resolution (mesh size, order) affects accuracy of flowfield approximation.
- In unsteady simulations, time step size is part of the "mesh."



Unstructured surface mesh

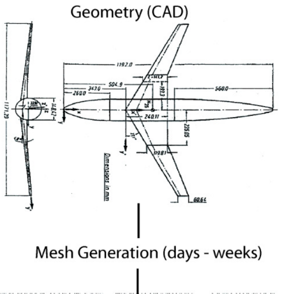


Cartesian cut-cells

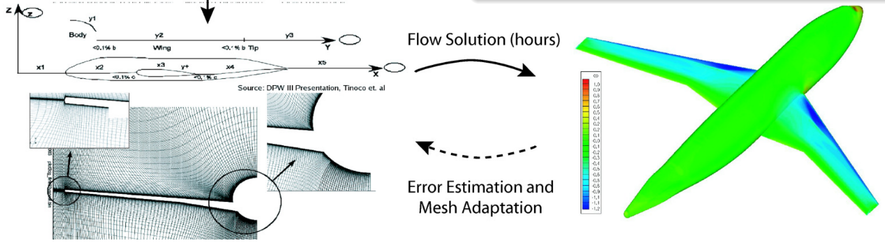


Multiblock volume mesh

Current Practices in Mesh Generation



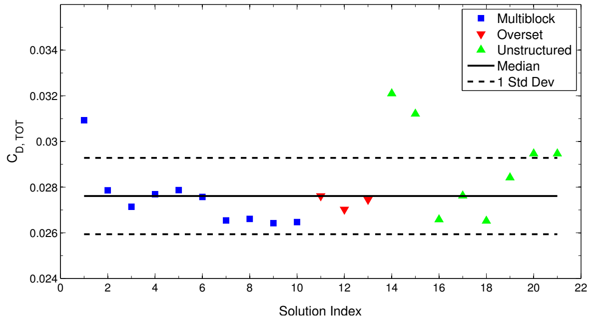
- Unstructured meshes can be generated with less user intervention (still not fully automated for complex geometries).
- Multi-block meshes are of highest quality for high *Re* viscous calculations.



Resulting Errors

AIAA Drag Prediction Workshop III (2006)

- Wing-body geometry, $M = 0.75$, $C_L = 0.5$, $Re = 5 \times 10^6$.
- Drag computed with various state of the art CFD codes.

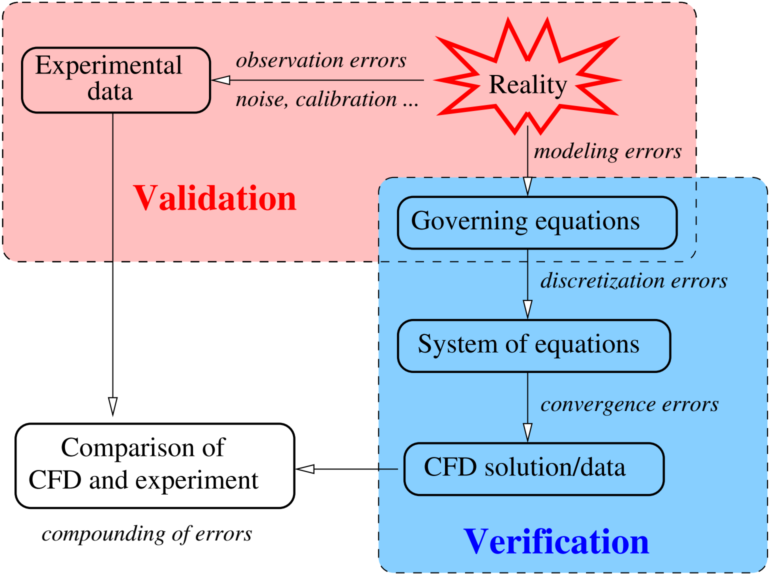


Differences in:

- Physical models
- Discretization
- **Mesh size distribution**

1 drag count ($.0001\,C_D$) $\approx$ 4-8 passengers for a large transport aircraft

Sources of Error



Verification: Control of Numerical Error

- Dominant source is *discretization error*
- Controlling error means answering
  1. How much error is present? (**error estimation**)
  2. How do I get rid of it? (**mesh adaptation**)
- Possible strategies:

	Error estimation?	Effective adaptation?
Resource exhaustion	**No**	**No**
Expert assessment	**Maybe**	**Maybe**
Convergence studies	**Yes**	**No**
Comparison to experiments	**Yes**	**No**
Feature–based adaptation	**No**	**Maybe**
Output–based methods	**Yes**	**Yes**

Outline

Why Outputs?

Output = scalar quantity computed from the CFD solution.

- A CFD solution may contain millions of degrees of freedom.
- Often of interest are only a few scalars (forces, moments, etc.)
- It is mathematically easier to speak of "error in an output" than "error in a CFD solution."

Output error = difference between an output computed with the discrete system solution and that computed with the exact solution to the PDE.

Output error estimation

- Identifies all areas of the domain that are important for the accurate prediction of an output.
- Accounts for error propagation effects.
- Requires solution of an *adjoint equation*.

Discrete Adjoint Definition

Consider N_H algebraic equations and an output,

$$\mathbf{R}_H(\mathbf{u}_H) = 0, \qquad J_H = J_H(\mathbf{u}_H)$$

- $\mathbf{u}_H \in \mathbb{R}^{N_H}$ is the vector of unknowns
- $\mathbf{R}_H \in \mathbb{R}^{N_H}$ is the vector of residuals (LHS of the equations)
- $J_H(\mathbf{u}_H)$ is a *scalar* output of interest

Adjoint definition

The discrete output adjoint vector, $\psi_H \in \mathbb{R}^{N_H}$, is the sensitivity of J_H to an infinitesimal residual perturbation, $\delta\mathbf{R}_H \in \mathbb{R}^{N_H}$,

$$\delta J_H \equiv \psi_H^T \delta\mathbf{R}_H$$

Discrete Adjoint Equation

The perturbed state, $\mathbf{u}_H + \delta\mathbf{u}_H$, must satisfy

$$\mathbf{R}_H(\mathbf{u}_H + \delta\mathbf{u}_H) + \delta\mathbf{R}_H = 0 \quad \Rightarrow \quad \frac{\partial \mathbf{R}_H}{\partial \mathbf{u}_H}\delta\mathbf{u}_H + \delta\mathbf{R}_H = 0,$$

Linearizing the output we have,

$$\delta J_H = \underbrace{\frac{\partial J_H}{\partial \mathbf{u}_H}\delta\mathbf{u}_H = \psi_H^T\delta\mathbf{R}_H}_{\text{adjoint definition}} \overbrace{\phantom{\psi_H^T\delta\mathbf{R}_H} = -\psi_H^T\frac{\partial \mathbf{R}_H}{\partial \mathbf{u}_H}\delta\mathbf{u}_H}^{\text{linearized equations}}$$

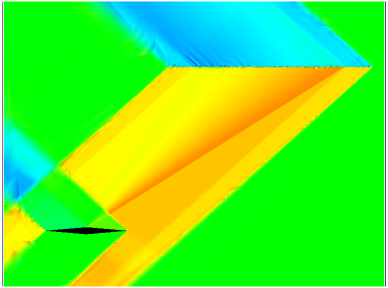
Requiring the above to hold for arbitrary perturbations yields the linear *discrete adjoint equation*

$$\left(\frac{\partial \mathbf{R}_H}{\partial \mathbf{u}_H}\right)^T \psi_H + \left(\frac{\partial J_H}{\partial \mathbf{u}_H}\right)^T = 0$$

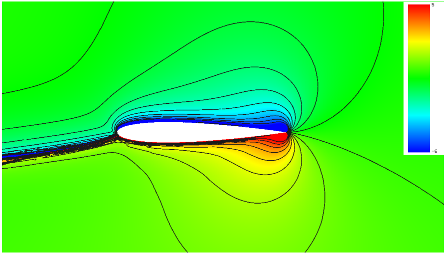
Continuous Adjoint

- If the following hold:
  1. the algebraic equations came from a consistent discretization of a continuous PDE, and
  2. the residual and output combination are *adjoint consistent*,

  then the discrete vector ψ_H approximates the *continuous adjoint* ψ.
- ψ is a Green's function relating source residual perturbations in the PDE to output perturbations.



y-momentum pres. integral adjoint: supersonic



x-momentum lift adjoint, $M_\infty = 0.4$, $\alpha = 5^o$

Adjoint Implementation

- The discrete adjoint, ψ_H, is obtained by solving a linear system.
- This system involves linearizations about the primal solution, $\mathbf{u}_H$, which is generally obtained first.
- When the full Jacobian matrix, $\frac{\partial \mathbf{R}_H}{\partial \mathbf{u}_H}$, and an associated linear solver are available, the transpose linear solve is straightforward.
- When the Jacobian matrix is not stored, the discrete adjoint solve is more involved: all operations in the primal solve must be linearized, transposed, and applied in reverse order.
- In unsteady discretizations, the adjoint must be marched backward in time from the final to the initial state.

Outline

Output Error Estimation

- Consider two discretization spaces:
  1. A **coarse** space with N_H degrees of freedom
  2. A **fine** one with $N_h > N_H$ degrees of freedom

  The fine discretization is usually obtained from the coarse one by refining the mesh or increasing the approximation order.

- The coarse state $\mathbf{u}_H$ will generally not satisfy the fine-level equations: $\mathbf{R}_h(\mathbf{I}_h^H \mathbf{u}_H) \neq \mathbf{0}$, where $\mathbf{I}_h^H$ is a coarse-to-fine prolongation operator.
- The fine-level adjoint, ψ_h, translates the residual perturbation $\delta \mathbf{R}_h \equiv -\mathbf{R}_h(\mathbf{I}_h^H \mathbf{u}_H)$ to an output perturbation:

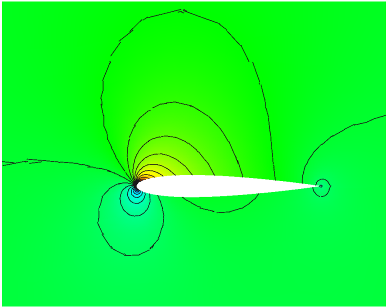
$$\delta J \approx \underbrace{-\left(\psi_h\right)^T \mathbf{R}_h \left(\mathbf{I}_h^H \mathbf{u}_H\right)}_{\text{adjoint-weighted residual}}$$

Approximation sign is present because $\delta \mathbf{R}_h$ is not infinitesimal.

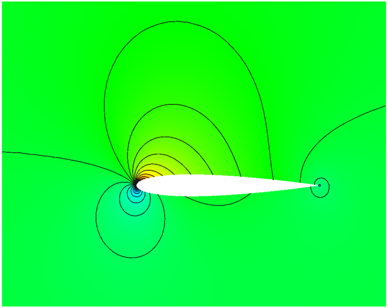
Adjoint-Weighted Residual Example

NACA 0012, $M_\infty = 0.5$, $\alpha = 5^o$

Interested in lift error in a $p = 1$ (second-order accurate) finite element solution. Using $p = 2$ for the fine space in error estimation.



$p = 1$ Mach contours

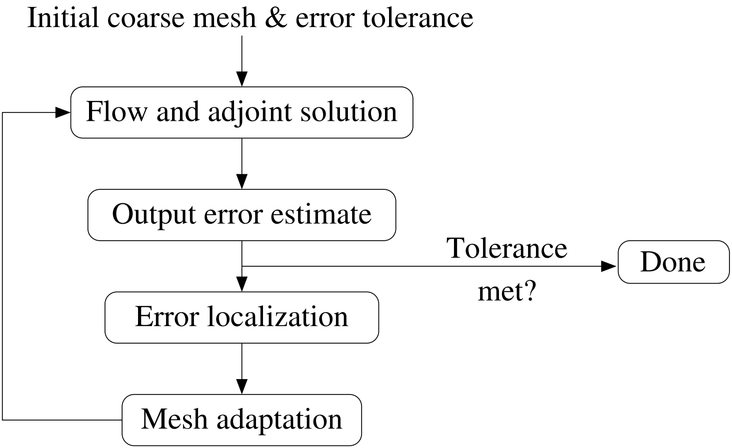


$p = 2$ Mach contours

- Adjoint-based error estimate: $-(\psi_h)^T \mathbf{R}_h \left(\mathbf{I}_h^H \mathbf{u}_H\right) = -.001097$
- Actual difference: $\delta J = -.001099$

Outline

Adaptive Solution Flowchart



Error Localization

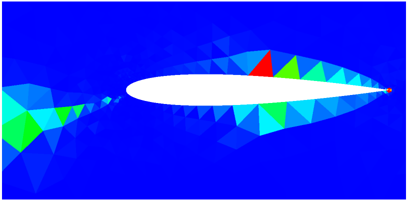
- Goal: need to identify problematic areas of the mesh
- The output error estimate,

$$\delta J \approx -\left(\psi_h\right)^T \mathbf{R}_h \left(\mathbf{I}_h^H \mathbf{u}_H\right)$$

is a sum over mesh elements (for finite volume/element methods)

<u>Error indicator on element κ</u>

$$\epsilon_\kappa = \left| -\left(\psi_{h,k}\right)^T \mathbf{R}_{h,k} \left(\mathbf{I}_h^H \mathbf{u}_H\right) \right|$$



Lift error indicator on a $p = 1$ DG solution

- Refinement in areas where ϵ_{κ_H} is large will reduce the residual there and hence improve the output accuracy.

Adaptation Mechanics

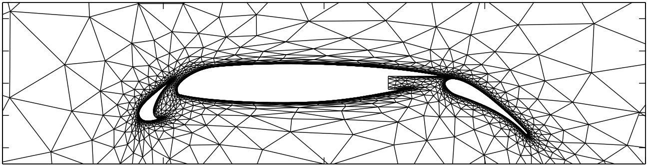
1. *h*-adaptation: only triangulation is varied
2. *p*-adaptation: only approximation order is varied
3. *hp*-adaptation: both triangulation and approximation order are varied

Given an error indicator, how should the mesh be adapted?

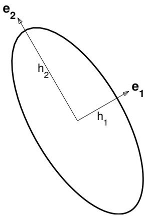
- Refine some/all elements?
- Incorporate anisotropy (stretching)?
- How to handle elements on the geometry?

Since mesh generation is difficult in the first place, adaptation needs to be automated to enable multiple iterations.

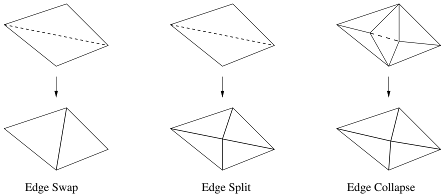
Meshing and Adaptation Strategies



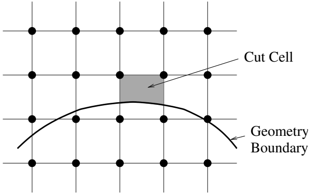
Metric-based anisotropic mesh regeneration (e.g. BAMG software)



Riemannian ellipse



Local mesh operators, and direct optimization



Cut-cell meshes: Cartesian and simplex

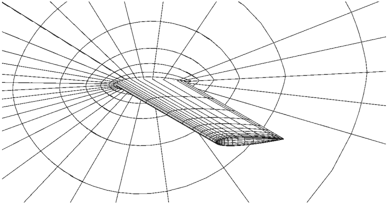
Outline

NACA Wing, $M = 0.4, \alpha = 3^o$

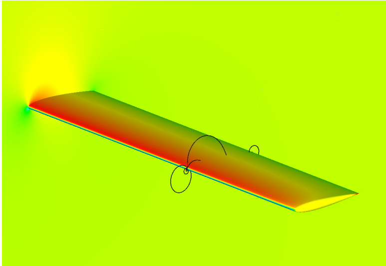
- Hanging-node adaptation
- Cubic curved geometry representation
- Hexahedral meshes
- $p = 2$ (third order) DG solution approximation
- Interested in lift and drag

Indicators

1. Drag and lift adjoints
2. Entropy adjoint
3. Residual
4. Entropy



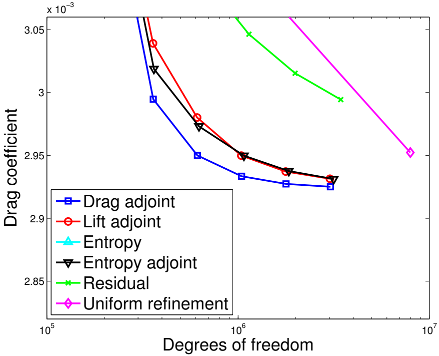
Initial mesh



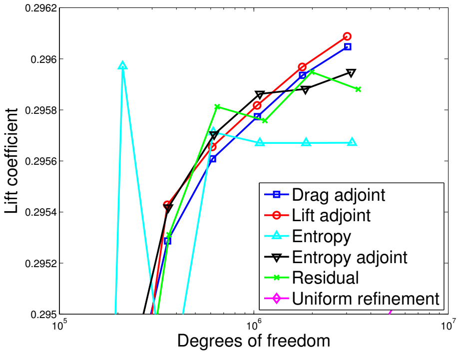
Mach number contours

NACA Wing, $M = 0.4, \alpha = 3^o$

- Degree of freedom (DOF) versus output error for $p = 2$
- Entropy adjoint performance again comparable to output adjoints

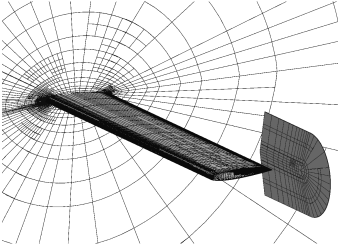


Drag

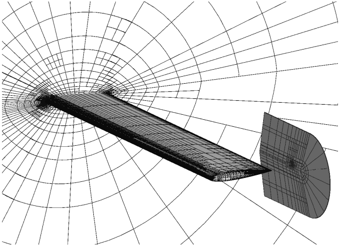


Lift

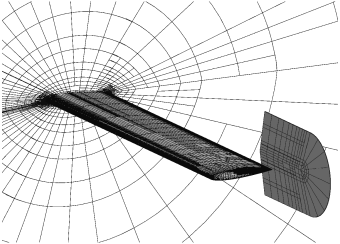
NACA Wing, $M = 0.4, \alpha = 3^o$, Final Meshes



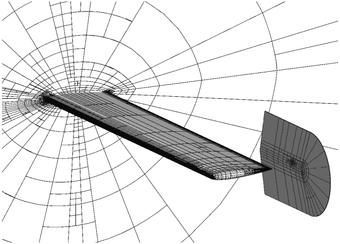
Drag Adjoint



Entropy Adjoint



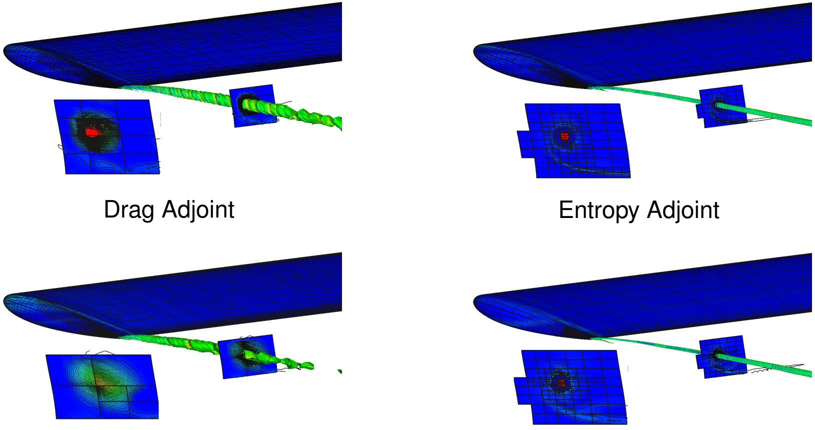
Lift Adjoint



Residual

NACA Wing, $M = 0.4$, $\alpha = 3^o$, Tip Vortex

Visualization of entropy isosurface and transverse cut contours



Drag Adjoint

Entropy Adjoint

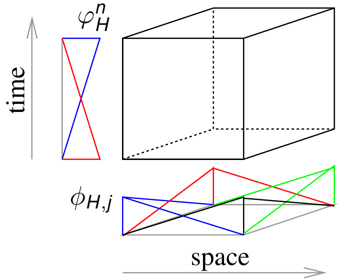
Lift Adjoint

Residual

Outline

Unsteady Extension

- The error estimation equations hold for unsteady problems.
- The adjoint is more expensive for nonlinear problems:
  - Adjoint solve proceeds backwards in time.
  - State vector is required at each time for linearization.
  - Must store or recompute state.
- Adaptation is trickier with the additional dimension of time.
- Current approach: **finite elements in space and time**.

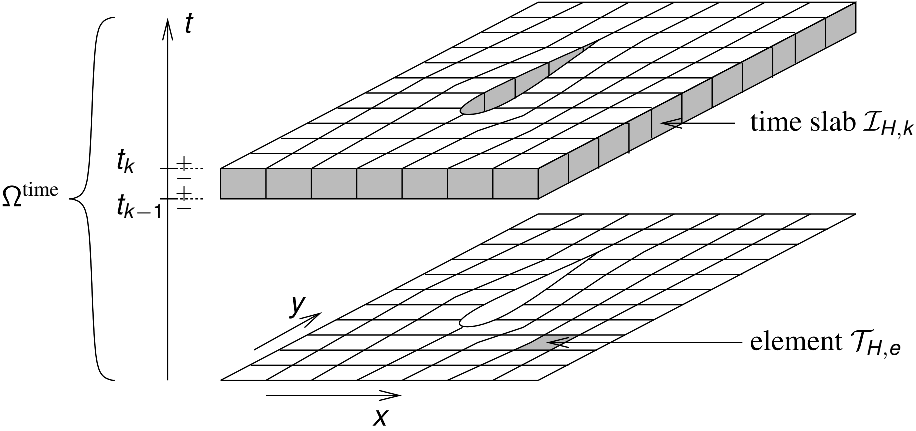


$$\mathbf{u}_H(\mathbf{x}, t) = \sum_n \sum_j \mathbf{u}_{H,j}^n \, \phi_{H,j}(\mathbf{x}) \varphi_H^n(t)$$

- $\phi_{H,j}(\mathbf{x})$ = j^{th} spatial basis function
- $\varphi_H^n(t)$ = n^{th} temporal basis function
- Basis functions are discontinuous in space and time (DG).

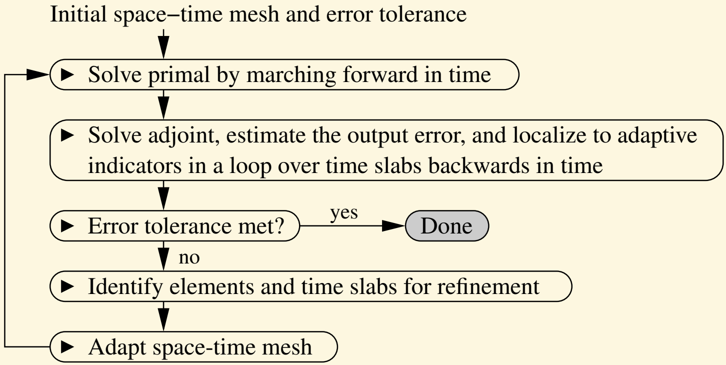
Discretization: Space-Time Mesh

- Time is discretized in slabs (all elements advance the same Δt)
- Each space-time element is prismatic (tensor product: $\mathcal{T}_e^H \otimes \mathcal{I}_k^H$)
- The spatial mesh is assumed to be invariant in time



Unsteady Adaptive Solution

Solution steps



The adaptation consists of hanging-node refinement in space and slab bisection in time.

Impulsively-Started Airfoil in Viscous Flow

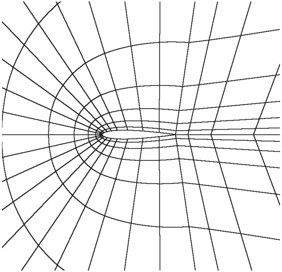
Governing equations (Navier-Stokes)

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial}{\partial x_i}\left[\mathbf{F}_i^I(\mathbf{u}) - \mathbf{F}_i^V(\mathbf{u}, \nabla\mathbf{u})\right] = 0$$

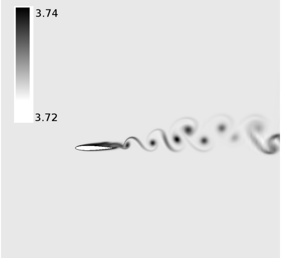
- $\mathbf{u} = [\rho, \rho u, \rho v, \rho E]^T$
- $\mathbf{F}_i^I(\mathbf{u})$ is the inviscid flux
- $\mathbf{F}_i^V(\mathbf{u}, \nabla\mathbf{u})$ is the viscous flux

Initial and boundary conditions

- At $t = 0$ the velocity is blended smoothly to zero in a circular disk around the airfoil
- The freestream conditions are $M_\infty = 0.25$, $\alpha = 8^\circ$, $Re = 5000$



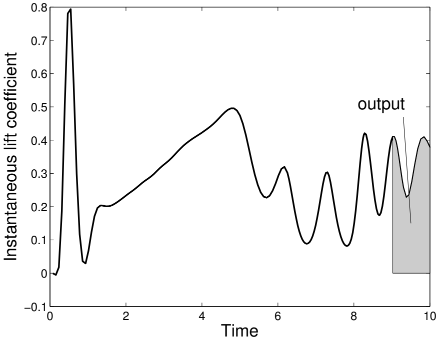
Initial condition and mesh



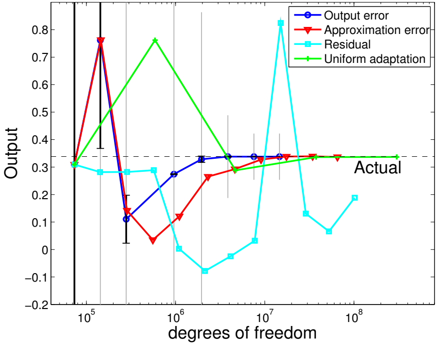
Entropy contours at $t = 10$

Impulsively-Started Airfoil: Output Convergence

The output of interest is the lift coefficient integral from $t = 9$ to $t = 10$



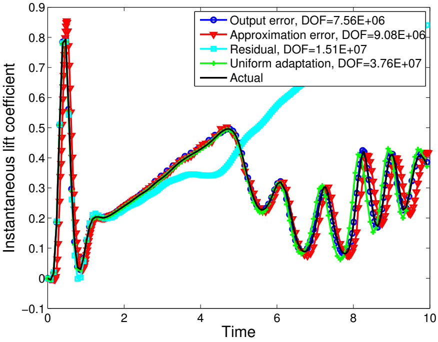
Time integral output definition.
A vortex-shedding pattern has been established by the time of the output measurement.



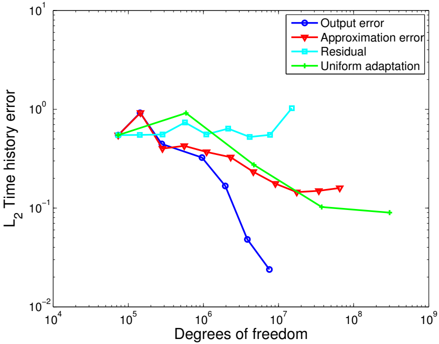
Convergence of output using various adaptive indicators. Shown on output-based results are:

- Error bars at $\pm\delta J$ (actual error est.)
- Whiskers at $\pm\epsilon$ (conservative error est.)

Impulsively-Started Airfoil: Time History Convergence



Lift coefficient time histories for adapted meshes with similar degrees of freedom. Values shown only at end of time slabs.

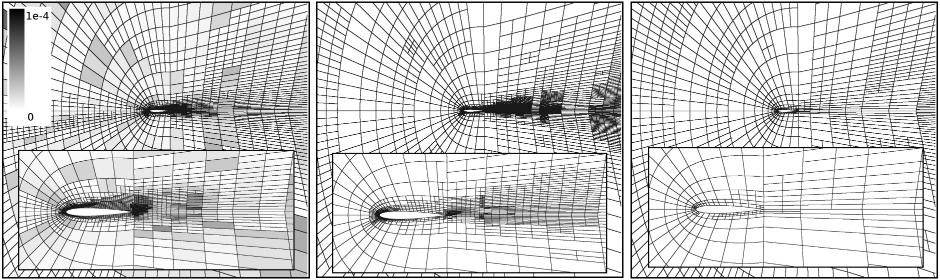


Convergence of the L_2 time history error for various adaptive indicators

Output-based adaptation yields not only an accurate scalar output, but also an accurate lift coefficient time history.

Impulsively-Started Airfoil: Adapted Spatial Meshes

- Meshes shown at iterations with similar total degrees of freedom.
- Spatially-marginalized output error indicator is shown on the elements of the output-adapted mesh.

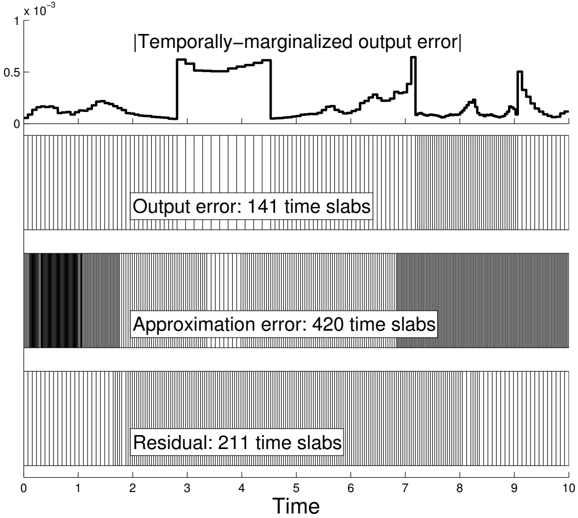


Adapted on output error (5956 elements)

Adapted on approximation error (4585 elements)

Adapted on residual (7929 elements)

Impulsively-Started Airfoil: Adapted Temporal Meshes



- Output error indicator yields a fairly-uniform temporal refinement.
- Approximation error focuses on the initial time (dynamics of the IC) and the latter 1/3 of the time, when the shed vortices develop.
- Residual creates a mostly-uniform temporal mesh as it tracks acoustic waves.

Outline

Conclusions

Conclusions

- An adequate CFD mesh is one that yields a sufficiently-low discretization error.
- The effect of discretization error on outputs can be quantified.
- Added cost: the solution of an adjoint problem.
- Benefit: error estimates *and* efficient meshes.
- Ideas apply to both steady and unsteady CFD problems.

What lies ahead

- Unsteady problems:
  - Dynamically-refined spatial meshes and grid motion
  - Forward solution checkpointing
  - Adjoint stability
- Entropy adjoint as a cheaper alternative
- Error bounds instead of estimates