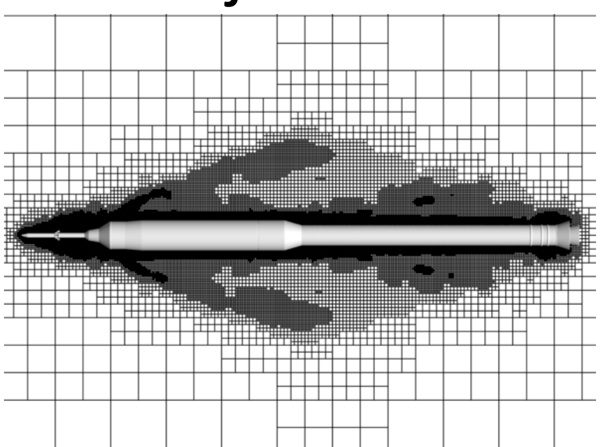
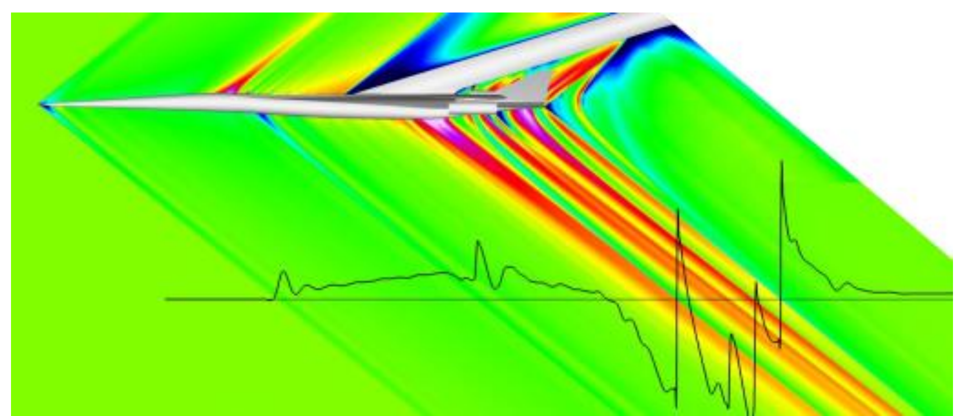
# Functional Error Estimation and Control for Time-Dependent Problems

Dimitri Mavriplis
Department of Mechanical Engineering
Univeristy of Wyoming

# Motivation

- Adjoint allows error estimation (and thus adaptive control) for specific objectives
  - Focus computational resources on output objective of interest
  - Conserve resources by de-emphasizing resolution/resources in regions that do not affect objective





# Motivation

- Adjoint error estimation well known for spatial error estimation and control (AMR) for steady-state problems
- **Extend to multidisciplinary time-dependent problems**
- Investigate formulations that can be used directly with existing discretizations/frameworks
  - Precludes space-time formulations, solver modifications
  - Lower potential, but more immediately applicable

# Outline

- Theoretical Formulation
  - Linear continuous formulation
  - Non-linear discrete formulation
- Formulation for Temporal-Algebraic error estimation in time-dependent ALE problems
  - Verification of error estimates
  - Adaptive control of temporal-algebraic error
- Combined spatial-temporal-algebraic error estimation and control
  - Equidistribution of error
  - Optimal cost  error control
- Generalized formulation for multidisciplianry problems
- Conclusions/Future Work

# Adjoint Error Estimation
# Continuous Linear Formulation

Consider solution of: with scalar output of interest:

$$Au = f \qquad\qquad L = (g, u)$$

# Adjoint Error Estimation
# Continuous Linear Formulation

Consider solution of: with scalar output of interest:

$$Au = f \qquad L = (g, u)$$

L can also be computed as (dual problem):

$$A^* v = g \qquad L = (v, f)$$

# Adjoint Error Estimation
# Continuous Linear Formulation

Consider solution of: with scalar output of interest:

$$Au = f \qquad L = (g, u)$$

L can also be computed as (dual problem):

$$A^{*}v = g \qquad L = (v, f)$$

*where A* is the adjoint operator of A defined as the operator that satisfies:*

$$(Au, v) = (v, A^{*}v)$$

# Proof

$$L = (g,u) = (A^* v,u)$$

$$L = (g,u) = (v, Au)$$

by definition of adjoint operator

$$L = (g,u) = (v, f)$$

Using Au=f

# Continuous Linear Case

For an approximate $\tilde{L} = L(\tilde{u})$ :

$$L - \tilde{L} = (g, u) - (g, \tilde{u})$$

$$L - \tilde{L} = (g, u - \tilde{u})$$

$$L - \tilde{L} = (A^* v, u - \tilde{u})$$

$$L - \tilde{L} = (v, A(u - \tilde{u}))$$

$$L - \tilde{L} = (v, f - A\tilde{u})$$

**Error in L is given by inner product of adjoint solution with primal residual exactly**

# Continuous Linear Case

- Adjoint problem same expense as primal problem
- Assuming cheaper approximate adjoint solution $\tilde{v}$

$$L - \tilde{L} = (\tilde{v}, A\tilde{u} - f) + (v - \tilde{v}, A\tilde{u} - f)$$

O(1) small small small

$$L \approx L_{corrected} = \tilde{L} + (\tilde{v}, A\tilde{u} - f)$$

- Provided:
  - $\tilde{u}$ converges to u (primal consistency)
  - $\tilde{v}$ converged to v (dual consistency)

# Non-Linear Discrete Case

- Use subscript h to denote discrete operator/solution
  - $u_h$ is exact discrete solution (unknown)
  - $\tilde{u}_h$ is approximate discrete solution (known)
- Exact (discrete) functional can be written as Taylor series about known approximate functional value as:

$$L_h(\mathbf{u}_h) = L_h(\tilde{\mathbf{u}}_h) + \left(\frac{\partial L_h}{\partial \mathbf{u}_h}\right)_{\tilde{\mathbf{u}}_h} (\mathbf{u}_h - \tilde{\mathbf{u}}_h) + \cdots$$

- Since residual of exact discrete solution must vanish

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

- Obtain expression for error in solution

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

# Non-Linear Discrete Case

- Substituting into Taylor series for L:

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

- Defining an adjoint variable as

$$\mathbf{\Lambda}_h^T = -\left(\frac{\partial L_h}{\partial \mathbf{u}_h}\right)_{\tilde{\mathbf{u}}_h} \left[\frac{\partial \mathbf{R}_h}{\partial \mathbf{u}_h}\right]_{\tilde{\mathbf{u}}_h}^{-1} \qquad \left[\frac{\partial \mathbf{R}_h}{\partial \mathbf{u}_h}\right]_{\tilde{\mathbf{u}}_h}^{T} \mathbf{\Lambda}_h^T = -\left(\frac{\partial L_h}{\partial \mathbf{u}_h}\right)_{\tilde{\mathbf{u}}_h}^{T}$$

- Obtain

$$L_h(\mathbf{u}_h) - L_h(\tilde{\mathbf{u}}_h) \approx \mathbf{\Lambda}_h^T \mathbf{R}_h(\tilde{\mathbf{u}}_h)$$

- Note: Even for exact discrete adjoint solution, estimate is approximate due to non-linear effects

# Interpretation of Adjoint Variable

$$\mathbf{\Lambda}_h^T = -\left(\frac{\partial L_h}{\partial \mathbf{u}_h}\right)_{\tilde{\mathbf{u}}_h} \left[\frac{\partial \mathbf{R}_h}{\partial \mathbf{u}_h}\right]_{\tilde{\mathbf{u}}_h}^{-1}$$

$$\mathbf{\Lambda}_h^T = -\frac{\partial L_h}{\partial \mathbf{R}_h}$$

Sensitivity of objective wrt residual

$$\delta L_h = -\mathbf{\Lambda}_h^T \delta \mathbf{R}_h$$

$$L_h(\mathbf{u}_h) - L_h(\tilde{\mathbf{u}}_h) \approx \mathbf{\Lambda}_h^T \mathbf{R}_h(\tilde{\mathbf{u}}_h)$$

Generalized Green's function

# Non-Linear Discrete Case

- As previously, exact (discrete) adjoint may be as costly to obtain as exact solution $u_h$
- Using approximate discrete adjoint $\tilde{\Lambda}_h$

$$L_h(\mathbf{u}_h) - L_h(\tilde{\mathbf{u}}_h) \approx \tilde{\mathbf{\Lambda}}_h^T \mathbf{R}_h(\tilde{\mathbf{u}}_h) + (\mathbf{\Lambda}_h^T - \tilde{\mathbf{\Lambda}}_h^T)\mathbf{R}_h(\tilde{\mathbf{u}}_h)$$

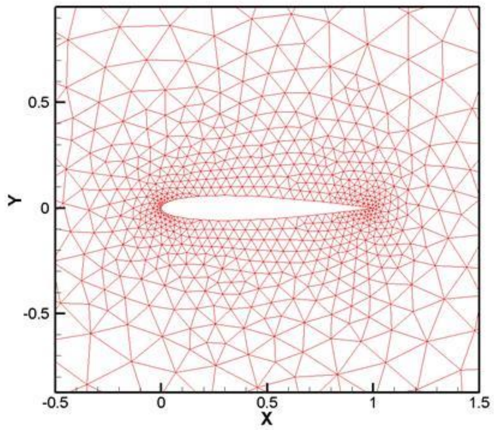
computable unknown

- Computable term will provide good error estimation if have asymptotically converging approximate $\tilde{\Lambda}_h$ , e.g.

$$\left[\frac{\partial \mathbf{R}_H}{\partial \mathbf{u}_H}\right]_{\mathbf{u}_H}^T \mathbf{\Lambda}_H^T = -\left(\frac{\partial L_H}{\partial \mathbf{u}_H}\right)_{\mathbf{u}_H}^T \longrightarrow \tilde{\mathbf{\Lambda}}_h = I_H^h \mathbf{\Lambda}_H$$

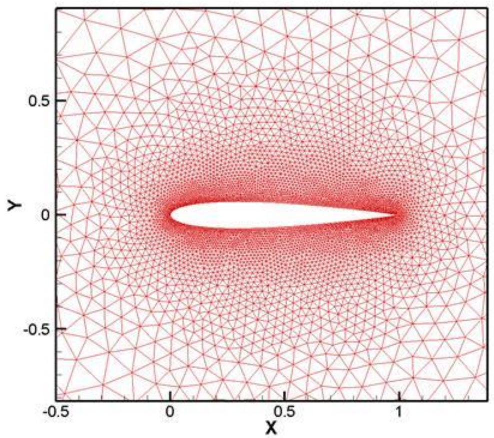
- Note 2 types of approximations
  - Approximate adjoint
  - Non-linear errors

# Functional Relevant Error
# A Simple Spatial Example




Arbitrary coarse resolution - H

Functional using real solution on H: $L_H(\mathbf{U}_H)$



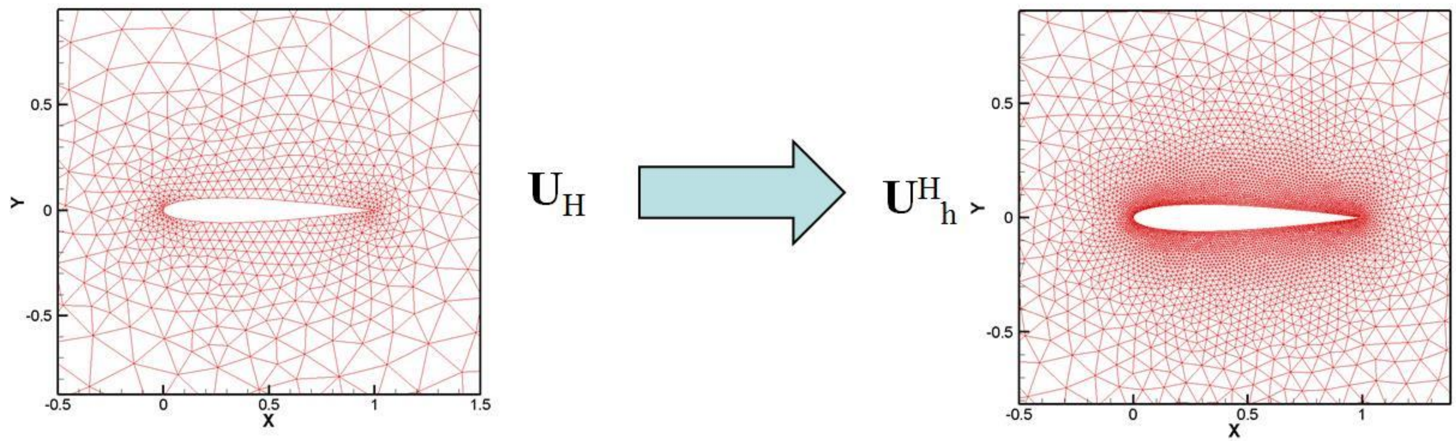

Arbitrary fine resolution - h

Functional using real solution on h: $L_h(\mathbf{U}_h)$

Can we estimate true fine level functional using only coarse level solution?

# Functional Relevant Error
# A Simple Spatial Example

- Here solution is approximate because is obtained from coarse grid $\tilde{\mathbf{u}}_h = I_H^h \mathbf{u}_H$

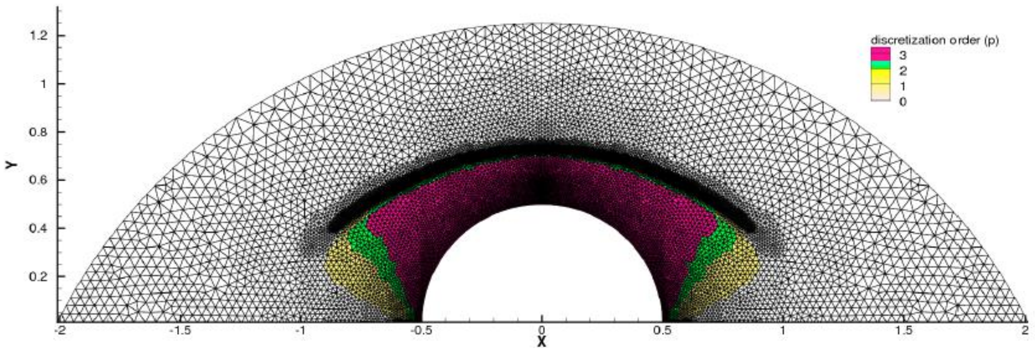


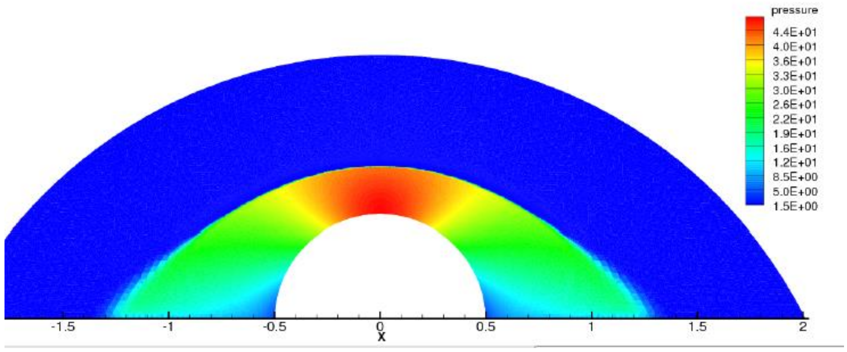

$$L_h(\mathbf{u}_h) - L_h(\tilde{\mathbf{u}}_h) \approx \tilde{\mathbf{\Lambda}}_h^T \mathbf{R}_h(\tilde{\mathbf{u}}_h) + (\mathbf{\Lambda}_h^T - \tilde{\mathbf{\Lambda}}_h^T)\mathbf{R}_h(\tilde{\mathbf{u}}_h)$$

- Use approximate adjoint computed on coarse grid

$$\left[\frac{\partial \mathbf{R}_H}{\partial \mathbf{u}_H}\right]^T_{\mathbf{u}_H} \mathbf{\Lambda}_H^T = -\left(\frac{\partial L_H}{\partial \mathbf{u}_H}\right)^T_{\mathbf{u}_H} \quad \Longrightarrow \quad \tilde{\mathbf{\Lambda}}_h = I_H^h \mathbf{\Lambda}_H$$

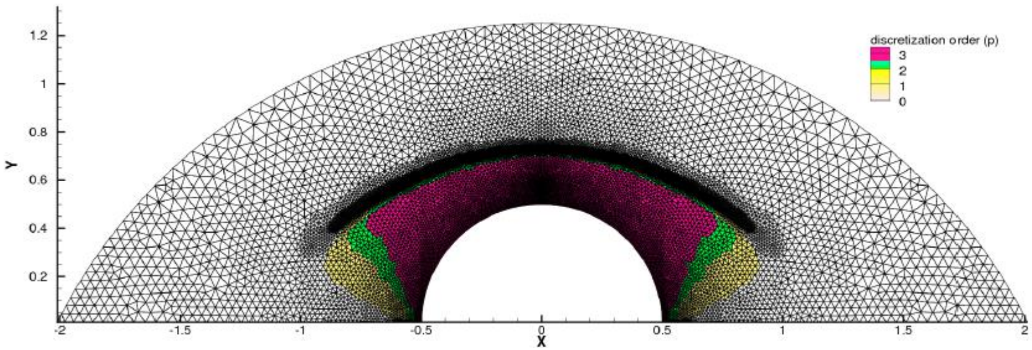
# Example: Spatial Discretization Error

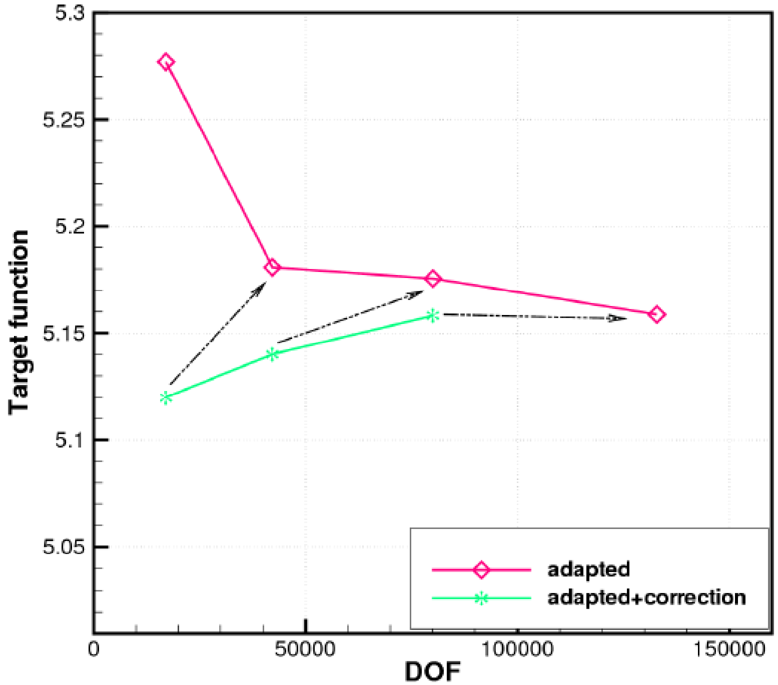



- Mach 6 flow over cylinder solved with h-p adaptive Discontinuous Galerkin scheme
- Objective is integral of surface temperature $\int T ds$ of cylinder
- Adjoint error estimates used to drive spatial h (mesh) and p (order) refinement
- **Refinement occurs ONLY in region of shock that affects objective**

# Example: Refinement History



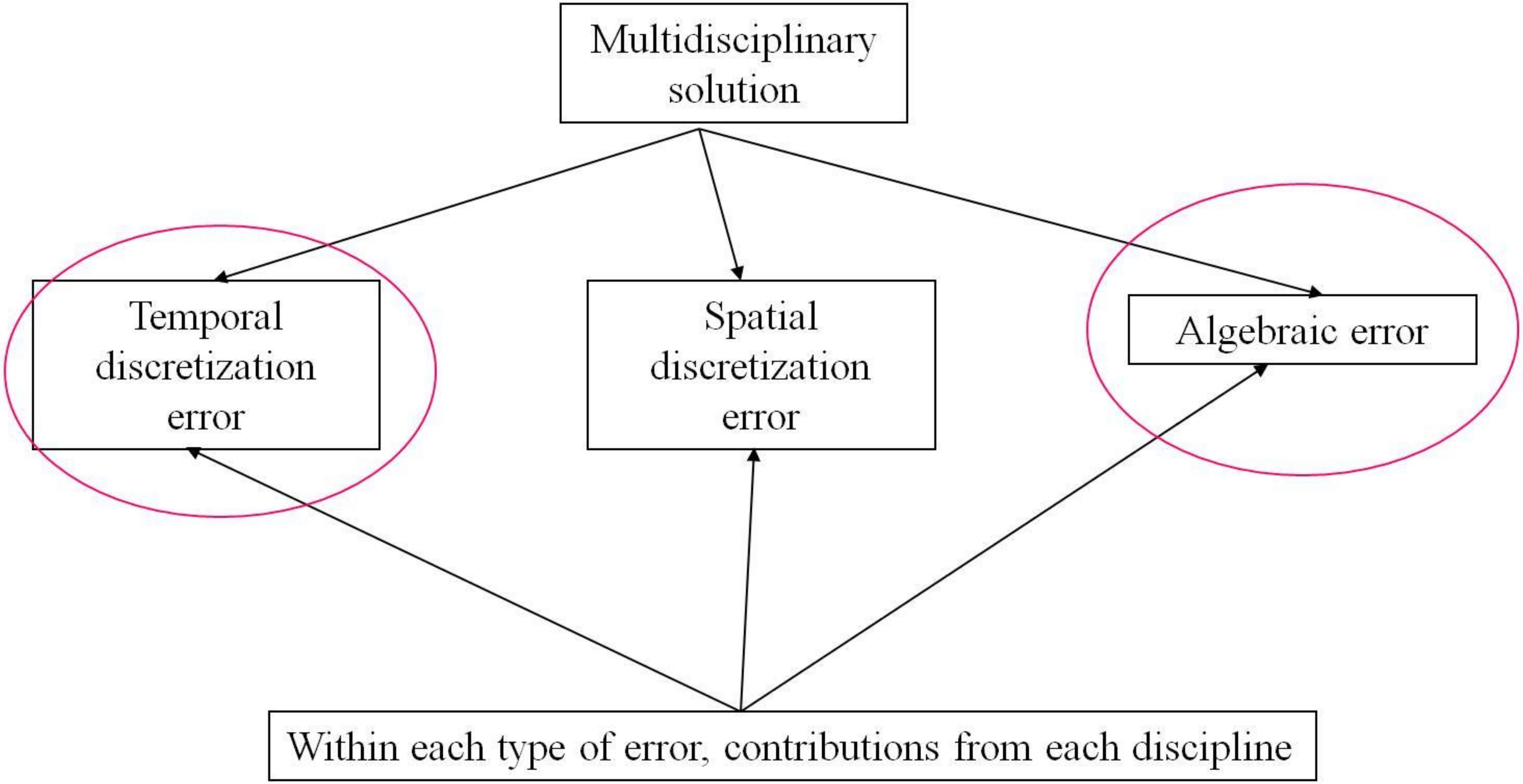




- Adjoint predicts discrete functional value on next refinement level
  - Not a predictor of total error/continuous functional value
- Error prediction improves at each refinement level
  - Decreasing non-linear error
  - Superconvergence of 2nd error term
- Final prediction is very accurate

# Different Error Sources

- Multidisciplinary time-dependent simulations contain many error sources
- Approximate nature of solution $\tilde{u}$ has not been specified
  - Computed on coarser grid (spatial disc. error)
  - Computed using larger time step (temporal error)
  - Not fully converged (algebraic error)
  - Computed using low fidelity model (modeling error)
  - Combinations of above
    - Can we use a single adjoint calculation to estimate (and control) different error types ?

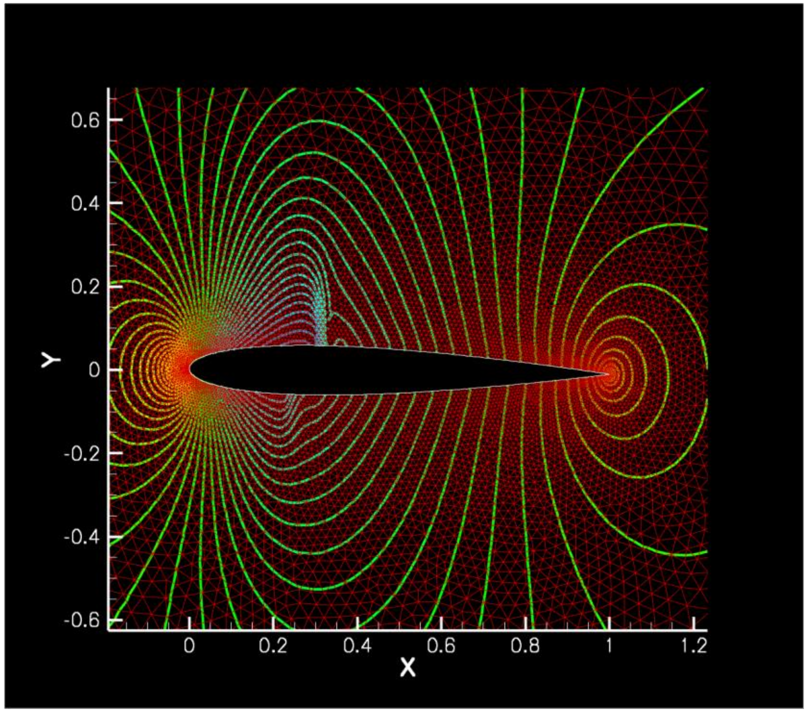
# Sources of Error

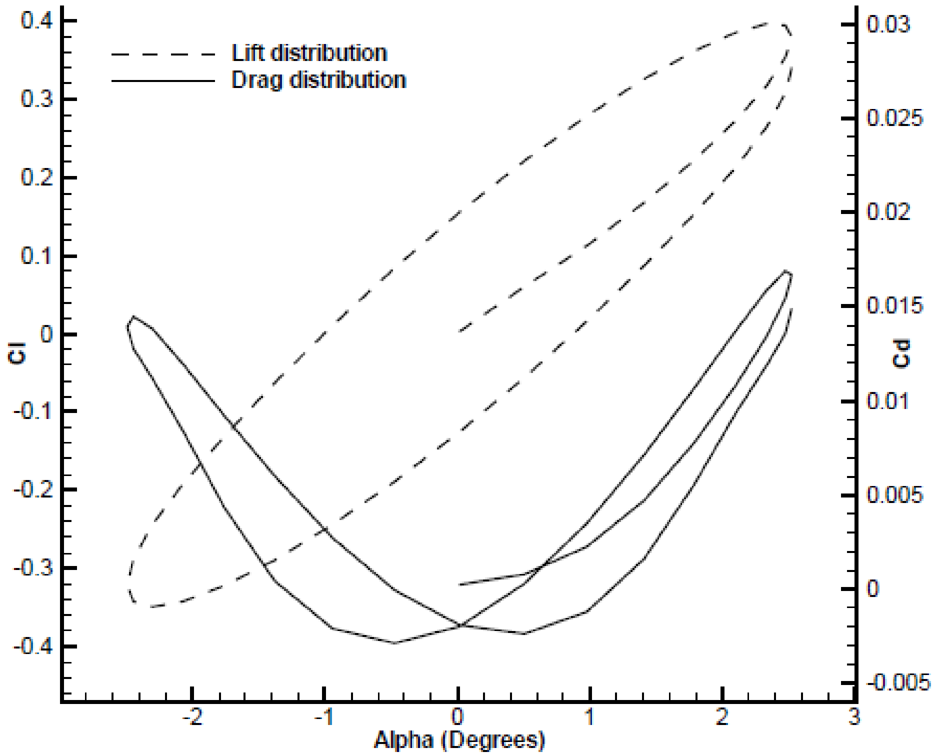


# Characteristics of Time Dependent Problems

- Ignore spatial error for now…
- Temporal error due to discrete (large) time steps
- Algebraic error more prevalent for time-dependent problems
  - Impractical to converge each implicit time step to machine precision
- Temporal and algebraic errors are intimately related for time dependent problems
  - Smaller implicit time steps converge faster
  - Algebraic error accumulates over all time steps
- Must be considered simultaneously

# Simple Multidisciplinary Time-Dependent Example




- Pitching airfoil with deforming mesh
- Estimate temporal and algebraic error
  - Ignore spatial discretization error for now

# Governing Equations

- Flow equations solved in ALE form at each time step
- Mesh deformation equations solved at each time step (prescribed airfoil motion)

$$\mathbf{R}_h(\mathbf{U}_h, \mathbf{x}_h) = 0$$
$$\mathbf{G}_h(\mathbf{x}_h) = 0$$

- Represents integration over all space and time
- At a given time step (BDF2)

$$\mathbf{R}^n = \mathbf{R}^n(\mathbf{U}^n, \mathbf{U}^{n-1}, \mathbf{U}^{n-2}, \mathbf{x}^n, \mathbf{x}^{n-1}, \mathbf{x}^{n-2}) = 0$$
$$\mathbf{G}_h(\mathbf{x}_h) = \frac{1}{\Delta t}\{[K]\,\delta x^n - \delta x^n_{surf}\} = 0$$

# Temporal Error Estimation

$$L_h(\mathbf{U}_h,\mathbf{x}_h) = L_h(\mathbf{U}_h^H,\mathbf{x}_h^H) + \left[\frac{\partial L_h}{\partial \mathbf{U}_h}\right]_{\mathbf{U}_h^H,\mathbf{x}_h^H} (\mathbf{U}_h - \mathbf{U}_h^H) + \left[\frac{\partial L_h}{\partial \mathbf{x}_h}\right]_{\mathbf{x}_h^H,\mathbf{U}_h^H} (\mathbf{x}_h - \mathbf{x}_h^H) + \cdots$$

$$\mathbf{R}_h(\mathbf{U}_h,\mathbf{x}_h) = \mathbf{R}_h(\mathbf{U}_h^H,\mathbf{x}_h^H) + \left[\frac{\partial \mathbf{R}_h}{\partial \mathbf{U}_h}\right]_{\mathbf{U}_h^H,\mathbf{x}_h^H} (\mathbf{U}_h - \mathbf{U}_h^H) + \left[\frac{\partial \mathbf{R}_h}{\partial \mathbf{x}_h}\right]_{\mathbf{x}_h^H,\mathbf{U}_h^H} (\mathbf{x}_h - \mathbf{x}_h^H) + \cdots = 0$$

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

$$L_h(U_h, x_h) = L_h(U_h^H, x_h^H) + \varepsilon_{cc1} + \varepsilon_{cc2}$$

$$\epsilon_{cc_1} = (\Lambda_{\mathbf{U}_h^H})^T R_h(\mathbf{U}_h^H,\mathbf{x}_h^H)$$

$$\epsilon_{cc_2} = (\Lambda_{\mathbf{x}_h^H})^T \mathbf{G}_h(\mathbf{x}_h^H)$$

Temporal error due to flow

Temporal error due to mesh

# Temporal Error Estimation

Temporal error due to flow

$$\epsilon_{cc_1} = (\Lambda_{\mathbf{U}_h^H})^T R_h(\mathbf{U}_h^H, \mathbf{x}_h^H)$$

$$\left[\frac{\partial \mathbf{R}_H}{\partial \mathbf{U}_H}\right]^T_{\mathbf{U}_H,\mathbf{x}_H} \Lambda_{\mathbf{U}H} = -\left[\frac{\partial L_H}{\partial \mathbf{U}_H}\right]^T_{\mathbf{U}_H,\mathbf{x}_H}$$

$$\Lambda_{\mathbf{U}_h^H} = I_h^H \Lambda_{\mathbf{U}H}$$

$R_h$ non zero because evaluated with approximate flow and mesh solution obtained using larger time step

Temporal error due to mesh

$$\epsilon_{cc_2} = (\Lambda_{\mathbf{x}_h^H})^T \mathbf{G}_h(\mathbf{x}_h^H)$$

$$[\mathbf{K}]^T \Lambda_{\mathbf{x}H} = -\left(\frac{1}{\Delta t}\right)(\lambda_{\mathbf{x}H})^T$$

$$\lambda_{\mathbf{x}} = \left\{(\Lambda_{\mathbf{U}})_h^T \left[\frac{\partial \mathbf{R}_h}{\partial \mathbf{x}_h}\right]_{\mathbf{U}_h^H \mathbf{x}_h^H} + \left[\frac{\partial L_h}{\partial \mathbf{x}_h}\right]_{\mathbf{x}_h^H}\right\}$$

$$\Lambda_{\mathbf{x}_h^H} = I_h^H \Lambda_{\mathbf{x}H}$$

$G_h$ non zero because evaluated with approximate mesh solution obtained using larger time step

# Algebraic Error Estimation

$$L_H(\mathbf{U}_H, \mathbf{x}_H) - L_H(\bar{\mathbf{U}}_H, \bar{\mathbf{x}}_H) \approx \left[\frac{\partial L_H}{\partial \mathbf{U}_H}\right]_{\bar{\mathbf{U}}_H, \bar{\mathbf{x}}_H} (\mathbf{U}_H - \bar{\mathbf{U}}_H) + \left[\frac{\partial L_H}{\partial \mathbf{x}_H}\right]_{\bar{\mathbf{x}}_H, \bar{\mathbf{U}}_H} (\mathbf{x}_H - \bar{\mathbf{x}}_H)$$

$$\mathbf{R}_H(\mathbf{U}_H, \mathbf{x}_H) = \mathbf{R}_H(\bar{\mathbf{U}}_H, \bar{\mathbf{x}}_H) + \left[\frac{\partial \mathbf{R}_H}{\partial \mathbf{U}_H}\right]_{\bar{\mathbf{U}}_H, \bar{\mathbf{x}}_H} (\mathbf{U}_H - \bar{\mathbf{U}}_H) + \left[\frac{\partial \mathbf{R}_H}{\partial \mathbf{x}_H}\right]_{\bar{\mathbf{x}}_H, \bar{\mathbf{U}}_H} (\mathbf{x}_H - \bar{\mathbf{x}}_H) + \cdots = 0$$

Similarly for mesh residual…..

$$L_H(U_H, x_H) = L_h(\widetilde{U}_H, \widetilde{x}_H) + \varepsilon_{cc1p} + \varepsilon_{cc2p}$$

$$\epsilon_{cc_{1p}} = (\Lambda_{\mathbf{U}H})^T \mathbf{R}(\bar{\mathbf{U}}_H, \bar{\mathbf{x}}_H)$$

$$\epsilon_{cc_{2p}} = (\Lambda_{\mathbf{x}H})^T \mathbf{G}(\bar{\mathbf{x}}_H)$$

Algebraic error due to flow

Algebraic error due to mesh

# Algebraic Error Estimation

Algebraic error due to flow

$$\epsilon_{cc1p} = (\Lambda_{\mathbf{U}H})^T \mathbf{R}(\bar{\mathbf{U}}_H, \bar{\mathbf{x}}_H)$$

$$\left[\frac{\partial \mathbf{R}_H}{\partial \mathbf{U}_H}\right]^T_{\bar{\mathbf{U}}_H, \bar{\mathbf{x}}_H} \Lambda_{\mathbf{U}H} = -\left[\frac{\partial L_H}{\partial \mathbf{U}_H}\right]^T_{\bar{\mathbf{U}}_H, \bar{\mathbf{x}}_H}$$

$R_h$ non zero because evaluated with approximate flow and mesh solution obtained partial convergence

Algebraic error due to mesh

$$\epsilon_{cc2} = (\Lambda_{\mathbf{x}_h^H})^T \mathbf{G}_h(\mathbf{x}_h^H)$$

$$[\mathbf{K}]^T \Lambda_{\mathbf{x}H} = -\left(\frac{1}{\Delta t}\right)(\lambda_{\mathbf{x}H})^T$$

$$with \quad \lambda_{\mathbf{x}H} = \left\{(\Lambda_{\mathbf{U}H})^T \left[\frac{\partial \mathbf{R}_H}{\partial \mathbf{x}_H}\right]_{\bar{\mathbf{U}}_H, \bar{\mathbf{x}}_H} + \left[\frac{\partial L_H}{\partial \mathbf{x}_H}\right]_{\bar{\mathbf{x}}_H, \bar{\mathbf{U}}_H}\right\}$$

$G_h$ non zero because evaluated with approximate mesh solution obtained using partial convergence

# Solution of Flow Adjoint Equation

$$\left[\frac{\partial \mathrm{R}_H}{\partial \mathrm{U}_H}\right]^T_{\bar{\mathrm{U}}_H,\bar{\mathrm{x}}_H} \Lambda_{\mathrm{U}H} = -\left[\frac{\partial L_H}{\partial \mathrm{U}_H}\right]^T_{\bar{\mathrm{U}}_H,\bar{\mathrm{x}}_H}$$

$$\mathrm{R}^n = \mathrm{R}^n(\mathrm{U}^n, \mathrm{U}^{n-1}, \mathrm{U}^{n-2}, \mathrm{x}^n, \mathrm{x}^{n-1}, \mathrm{x}^{n-2}) = 0$$

$$\begin{bmatrix} \begin{bmatrix} \frac{\partial \mathbf{R}^1(\mathbf{U})}{\partial \mathbf{U}^1} \\ \frac{\partial \mathbf{R}^2(\mathbf{U})}{\partial \mathbf{U}^1} \\ \frac{\partial \mathbf{R}^3(\mathbf{U})}{\partial \mathbf{U}^1} \end{bmatrix} & \begin{bmatrix} \frac{\partial \mathbf{R}^2(\mathbf{U})}{\partial \mathbf{U}^2} \\ \frac{\partial \mathbf{R}^3(\mathbf{U})}{\partial \mathbf{U}^2} \end{bmatrix} & \begin{bmatrix} \frac{\partial \mathbf{R}^3(\mathbf{U})}{\partial \mathbf{U}^3} \end{bmatrix} & & 0 & \\ & & \ddots & \ddots & & \\ & 0 & \begin{bmatrix} \frac{\partial \mathbf{R}^{n-1}(\mathbf{U})}{\partial \mathbf{U}^{n-3}} \end{bmatrix} & \begin{bmatrix} \frac{\partial \mathbf{R}^{n-1}(\mathbf{U})}{\partial \mathbf{U}^{n-2}} \\ \frac{\partial \mathbf{R}^{n}(\mathbf{U})}{\partial \mathbf{U}^{n-2}} \end{bmatrix} & \begin{bmatrix} \frac{\partial \mathbf{R}^{n-1}(\mathbf{U})}{\partial \mathbf{U}^{n-1}} \\ \frac{\partial \mathbf{R}^{n}(\mathbf{U})}{\partial \mathbf{U}^{n-1}} \end{bmatrix} & \begin{bmatrix} \frac{\partial \mathbf{R}^{n}(\mathbf{U})}{\partial \mathbf{U}^{n}} \end{bmatrix} \end{bmatrix}^T \begin{bmatrix} \Lambda^1 \\ \Lambda^2 \\ \Lambda^3 \\ \vdots \\ \Lambda^{n-1} \\ \Lambda^n \end{bmatrix} = - \begin{bmatrix} \frac{\partial L}{\partial \mathbf{U}^1} \\ \frac{\partial L}{\partial \mathbf{U}^2} \\ \frac{\partial L}{\partial \mathbf{U}^3} \\ \vdots \\ \frac{\partial L}{\partial \mathbf{U}^{n-1}} \\ \frac{\partial L}{\partial \mathbf{U}^n} \end{bmatrix}^T$$

Lower triangular form over time

# Solution of Flow Adjoint Equation

$$\left[\frac{\partial \mathbf{R}_H}{\partial \mathbf{U}_H}\right]^T_{\bar{\mathbf{U}}_H,\bar{\mathbf{x}}_H} \Lambda_{\mathbf{U}H} \quad = \quad - \left[\frac{\partial L_H}{\partial \mathbf{U}_H}\right]^T_{\bar{\mathbf{U}}_H,\bar{\mathbf{x}}_H}$$

$$\mathbf{R}^n = \mathbf{R}^n(\mathbf{U}^n, \mathbf{U}^{n-1}, \mathbf{U}^{n-2}, \mathbf{x}^n, \mathbf{x}^{n-1}, \mathbf{x}^{n-2}) = 0$$

$$\begin{bmatrix} \left[\frac{\partial \mathbf{R}^1(\mathbf{U})}{\partial \mathbf{U}^1}\right]^T & \left[\frac{\partial \mathbf{R}^2(\mathbf{U})}{\partial \mathbf{U}^1}\right]^T & \left[\frac{\partial \mathbf{R}^3(\mathbf{U})}{\partial \mathbf{U}^1}\right]^T & & & \\ & \left[\frac{\partial \mathbf{R}^2(\mathbf{U})}{\partial \mathbf{U}^2}\right]^T & \left[\frac{\partial \mathbf{R}^3(\mathbf{U})}{\partial \mathbf{U}^2}\right]^T & \left[\frac{\partial \mathbf{R}^4(\mathbf{U})}{\partial \mathbf{U}^2}\right]^T & 0 & \\ & & \ddots & \ddots & & \\ & & & \left[\frac{\partial \mathbf{R}^{n-2}(\mathbf{U})}{\partial \mathbf{U}^{n-2}}\right]^T & \left[\frac{\partial \mathbf{R}^{n-1}(\mathbf{U})}{\partial \mathbf{U}^{n-2}}\right]^T & \left[\frac{\partial \mathbf{R}^{n}(\mathbf{U})}{\partial \mathbf{U}^{n-2}}\right]^T \\ & 0 & & & \left[\frac{\partial \mathbf{R}^{n-1}(\mathbf{U})}{\partial \mathbf{U}^{n-1}}\right]^T & \left[\frac{\partial \mathbf{R}^{n}(\mathbf{U})}{\partial \mathbf{U}^{n-1}}\right]^T \\ & & & & & \left[\frac{\partial \mathbf{R}^{n}(\mathbf{U})}{\partial \mathbf{U}^{n}}\right]^T \end{bmatrix} \begin{bmatrix} \Lambda^1 \\ \Lambda^2 \\ \Lambda^3 \\ \vdots \\ \Lambda^{n-1} \\ \Lambda^n \end{bmatrix} = - \begin{bmatrix} \frac{\partial L}{\partial \mathbf{U}^1} \\ \frac{\partial L}{\partial \mathbf{U}^2} \\ \frac{\partial L}{\partial \mathbf{U}^3} \\ \vdots \\ \frac{\partial L}{\partial \mathbf{U}^{n-1}} \\ \frac{\partial L}{\partial \mathbf{U}^n} \end{bmatrix}^T$$

$$\left[\frac{\partial \mathbf{R}^k(\mathbf{U})}{\partial \mathbf{U}^k}\right]^T \Lambda^k = \frac{\partial L}{\partial \mathbf{U}^k} - \left[\frac{\partial \mathbf{R}^{k+1}(\mathbf{U})}{\partial \mathbf{U}^k}\right]^T \Lambda^{k+1} - \left[\frac{\partial \mathbf{R}^{k+2}(\mathbf{U})}{\partial \mathbf{U}^k}\right]^T \Lambda^{k+2}$$

# Combined Temporal and Algebraic Error

- For temporal error estimation, assumed coarse time solution was fully converged prior to projection to fine time level
- In practice, approximate solution is partially converged on coarse time step solution
  - Algebraic error estimate is unchanged (no projection)
  - Temporal error estimate includes temporal error and algebraic error

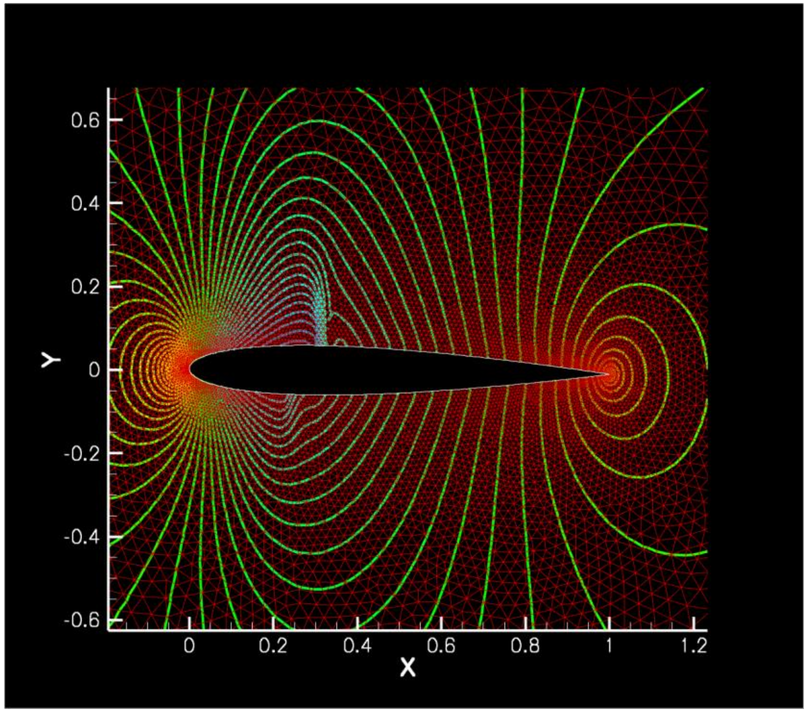
$$L_h(U_h, x_h) = L_h(U_h^H, x_h^H) + \varepsilon_{cc1} + \varepsilon_{cc2}$$

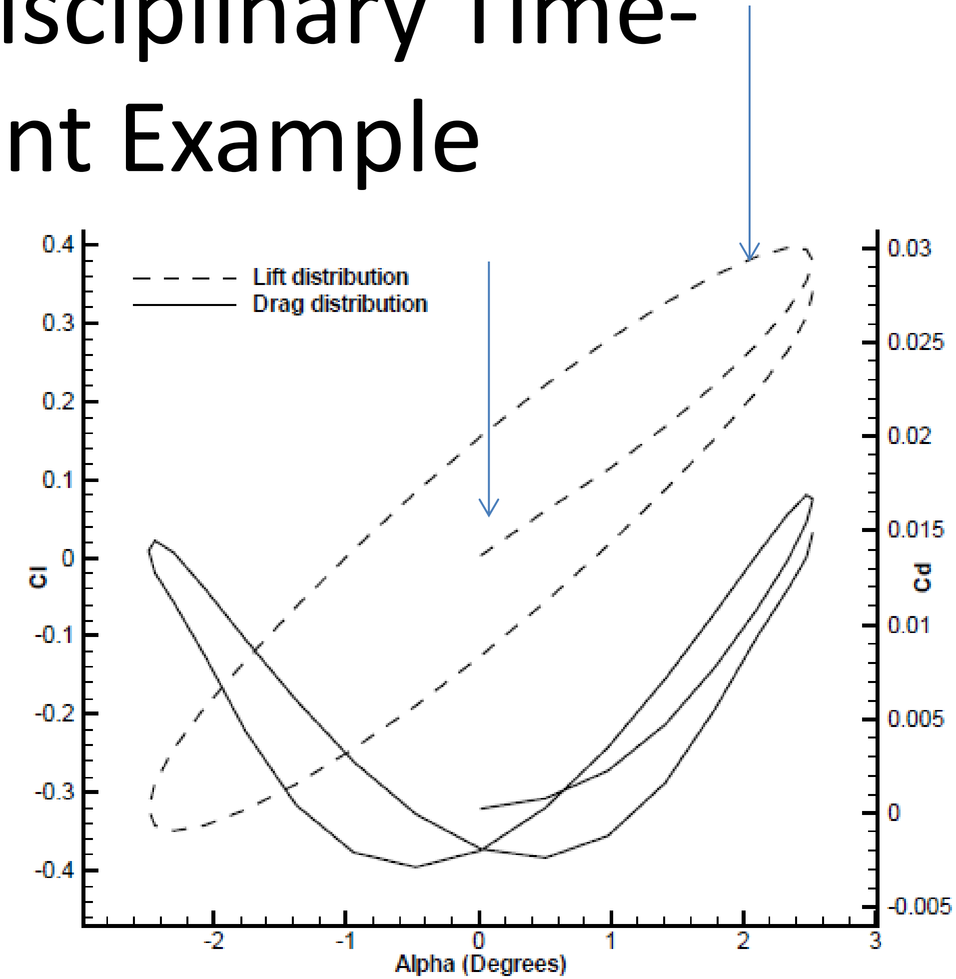
$$\varepsilon_{cc1} = \varepsilon_{cc1p} + \text{Temporal error due to flow}$$

$$\varepsilon_{cc2} = \varepsilon_{cc2p} + \text{Temporal error due to mesh}$$

- Additive error estimation is best we can do within context of adjoint formulation (a linearization)

# Simple Multidisciplinary Time-Dependent Example




- Pitching airfoil with deforming mesh
- Estimate Temporal/Algebraic error in time-integrated lift over 1st quarter period

# Validation of Error Estimates

Temporal discretization error only:

| Flow/Mesh Convergence Tolerances | H/h | $L_h(\mathbf{U}_h, \mathbf{x}_h)$ | $L_h(\mathbf{U}_h^H, \mathbf{x}_h^H)$ | Exact Error | Predicted Error | Ratio |
|---|---|---|---|---|---|---|
| 2e-14/1e-15 | 8/16 | 4.7627748 | 4.7419952 | 0.02077960 | 0.02222993 | 1.06979621 |
| 2e-14/1e-15 | 16/32 | 4.6769170 | 4.6602314 | 0.01668557 | 0.01616314 | 0.96868998 |
| 2e-14/1e-15 | 32/64 | 4.6352466 | 4.6257924 | 0.00945421 | 0.00945039 | 0.99959626 |
| 2e-14/1e-15 | 64/128 | 4.6149705 | 4.6097293 | 0.00524124 | 0.00524024 | 0.99980920 |

Flow algebraic error:

| Flow/Mesh Convergence Tolerance | H | $L_H(\mathbf{U}_H, \mathbf{x}_H)$ | $L_H(\bar{\mathbf{U}}_H, \mathbf{x}_H)$ | Exact Error | Predicted Error | Ratio |
|---|---|---|---|---|---|---|
| 1e-5/1e-15 | 8 | 4.9477907 | 4.9335132 | 0.0142775 | 0.0143015 | 0.9983222 |

Mesh algebraic error:

| Flow/Mesh Convergence Tolerance | H | $L_H(\mathbf{U}_H, \mathbf{x}_H)$ | $L_H(\mathbf{U}_H, \bar{\mathbf{x}}_H)$ | Exact Error | Predicted Error | Ratio |
|---|---|---|---|---|---|---|
| 2e-14/1e-5 | 8 | 4.9477907 | 4.9477703 | 2.0380075e-5 | 2.0631192e-5 | 1.0123217 |

# Validation of Error Estimates

Combined total error:

| Flow/Mesh Convergence Tolerance | H/h | $L_h(\mathbf{U}_h, \mathbf{x}_h)$ | $L_h(\bar{\mathbf{U}}_h^H, \bar{\mathbf{x}}_h^H)$ | Exact Error | Predicted Error | Ratio |
|---|---|---|---|---|---|---|
| 1e-5/1e-4 | 2/4 | 5.3287125 | 5.2361864 | 0.0925261 | 0.0897410 | 0.9699000 |
| 1e-6/1e-5 | 4/8 | 4.9477907 | 4.9142574 | 0.0335332 | 0.0364249 | 1.0862345 |
| 1e-7/1e-6 | 8/16 | 4.7627748 | 4.7419952 | 0.0207796 | 0.0222299 | 1.0697962 |
| 1e-8/1e-7 | 16/32 | 4.6769170 | 4.6602314 | 0.0166855 | 0.0161631 | 0.9686899 |
| 1e-9/1e-8 | 32/64 | 4.6352466 | 4.6257924 | 0.0094542 | 0.0094503 | 0.9995962 |
| 1e-10/1e-9 | 64/128 | 4.6149042 | 4.6097293 | 0.0051749 | 0.0051738 | 0.9997898 |

# Adaptation Results General Notes:

Targeted temporal adaptation compared against local error-based adaptation:

Local error estimated as:

$$e_{local} = \left\| \left[ \frac{dA\mathbf{U}}{dt} \right]_{BDF3} - \left[ \frac{dA\mathbf{U}}{dt} \right]_{BDF2} \right\|_2$$

<u>Adaptation strategy:</u>

Sort by time-steps by error contribution - decreasing order

Parse down list and flag time-steps for refinement until 99% error is covered

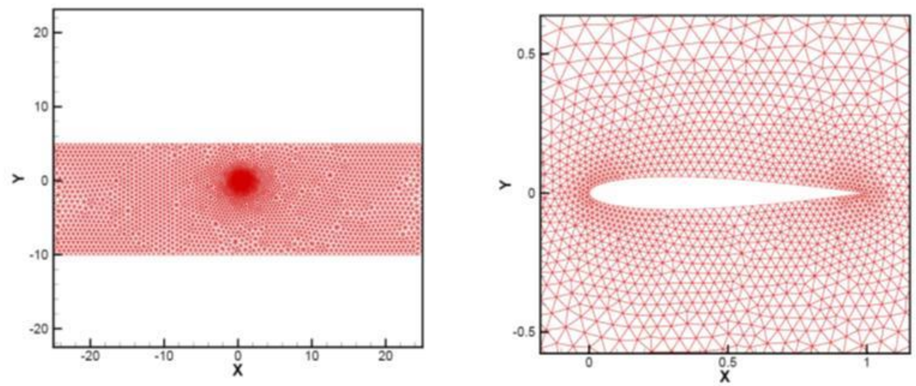
Same for all error components

Temporal resolution adaptation = divide time-step by two

Convergence tolerance adaptation = tighten by factor of 3

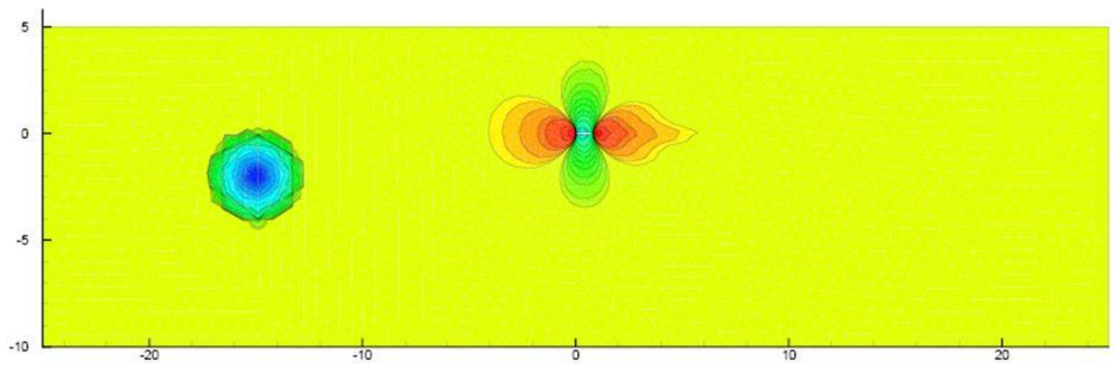
# Adaptation Results
# Time-Integrated Functional

Interaction of a convecting vortex with a slowly pitching airfoil. NACA0012 airfoil pitching at kc=0.001. Mach number is 0.4225. Starting at 50 steps uniform time-steps.

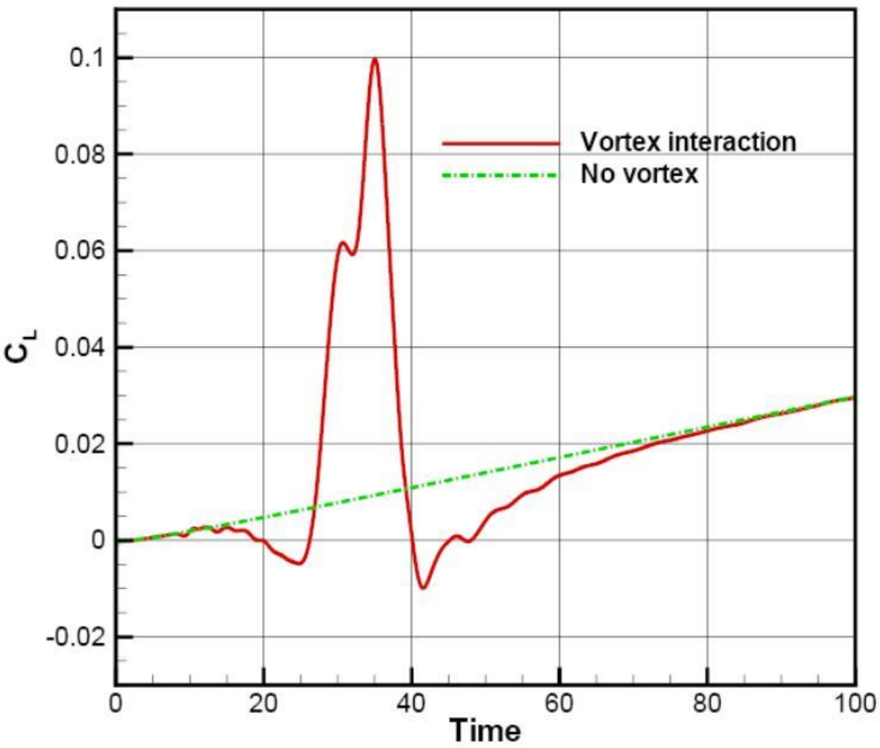



(a) Complete domain

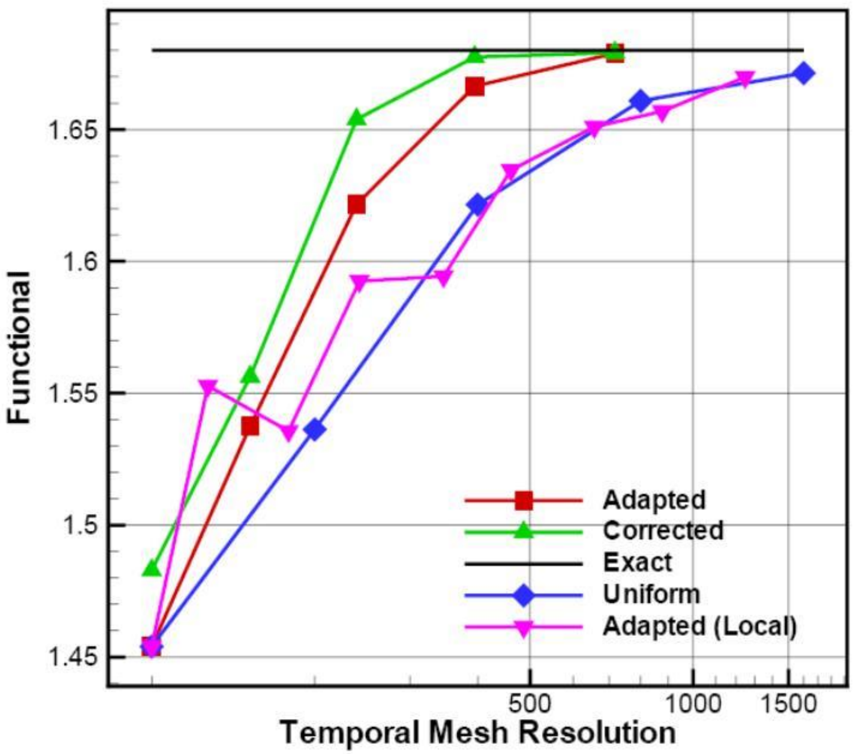
(b) Zoomed in view of airfoil within domain



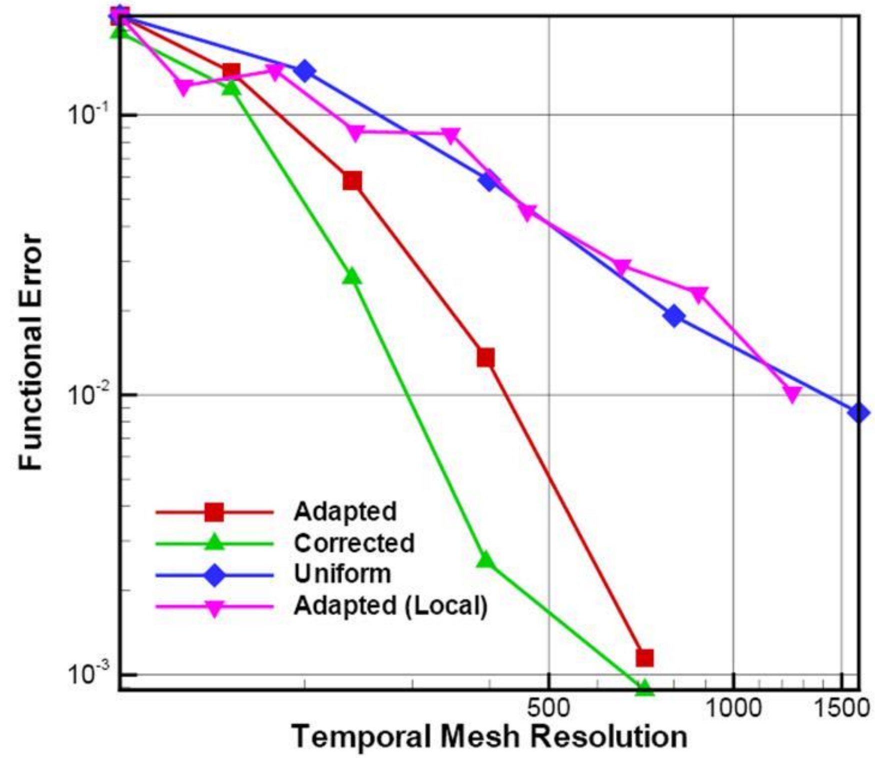
# Adaptation Results
# Time-Integrated Functional



# Adaptation Results
# Time-Integrated Functional

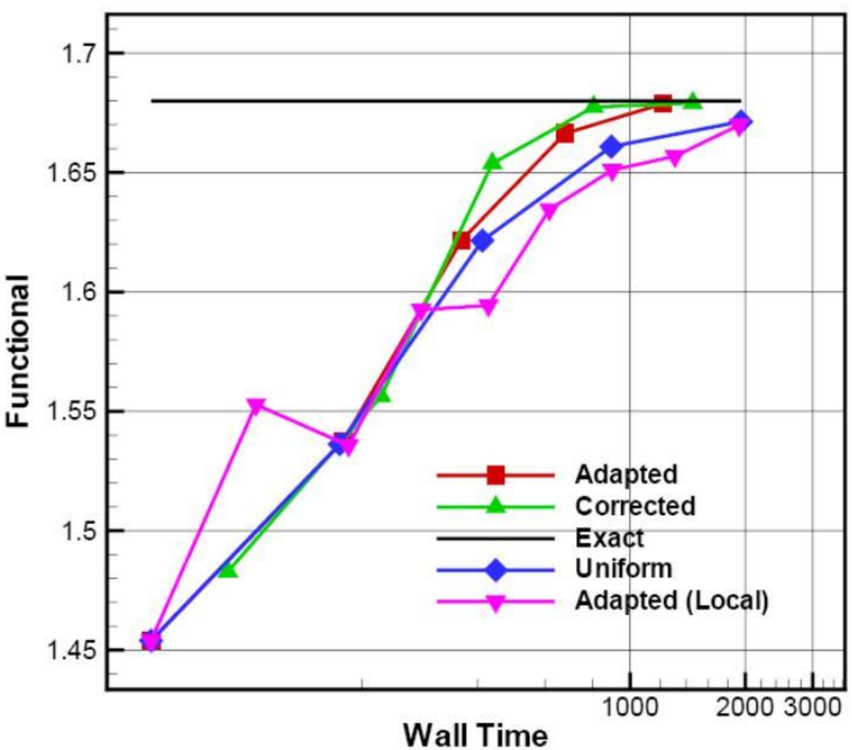



(a) Functional convergence

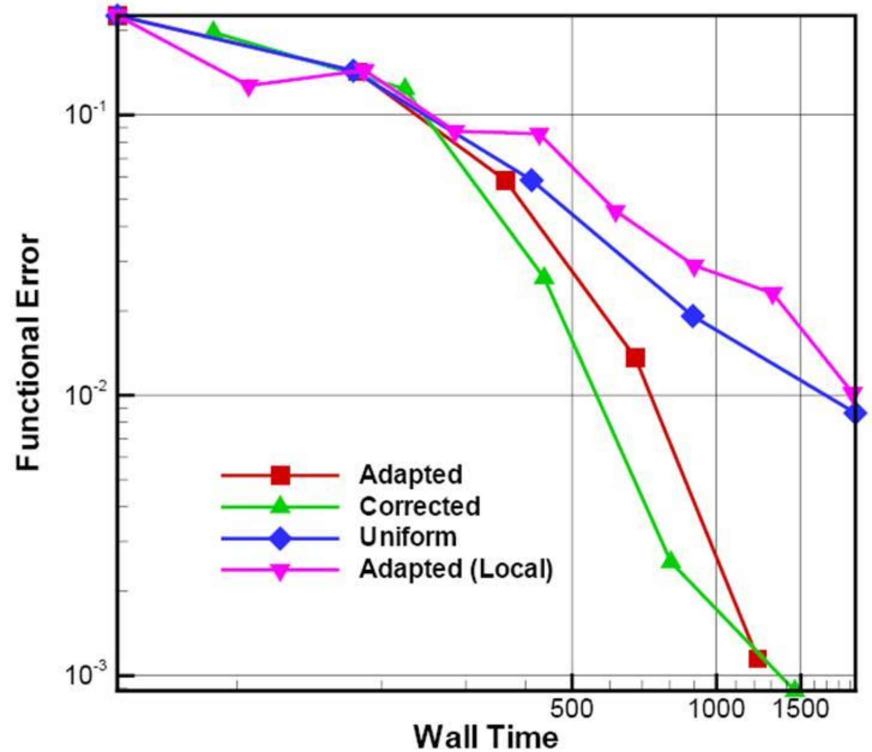



(b) Functional error convergence

# Adaptation Results
# Time-Integrated Functional

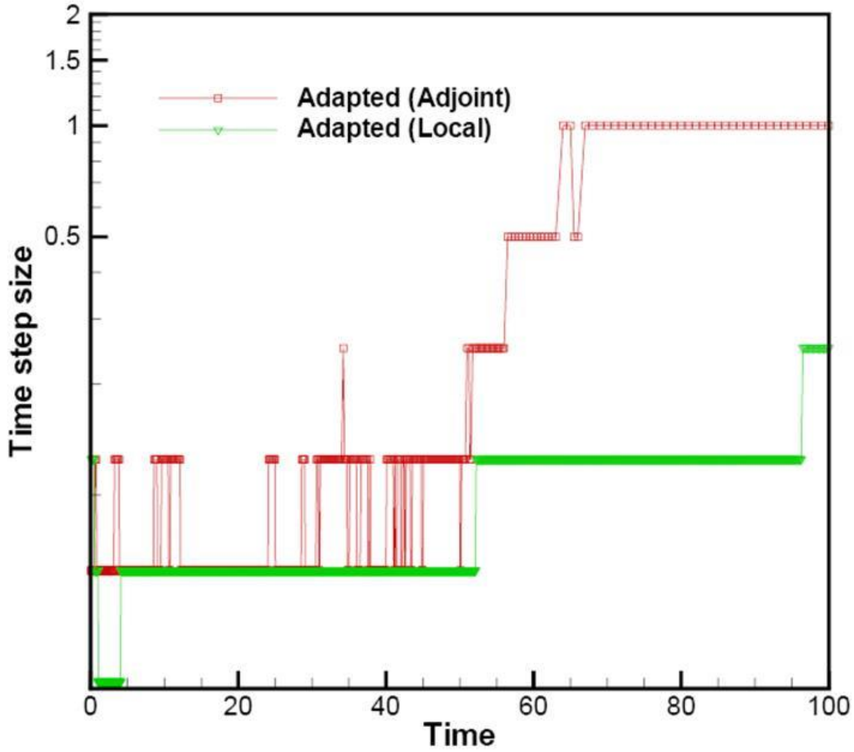



(a) Functional convergence

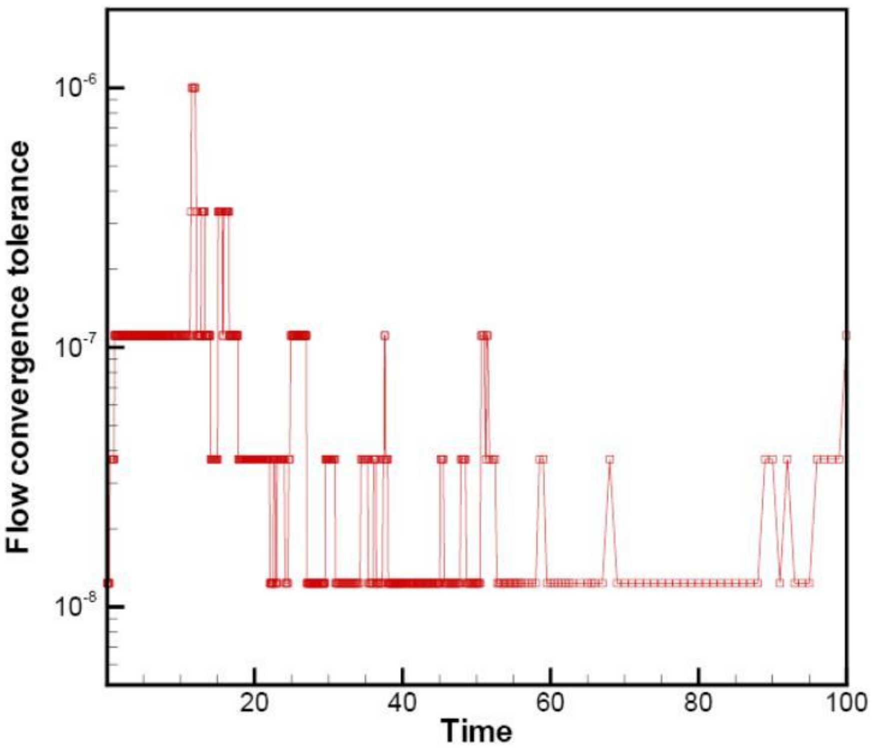



(b) Functional error convergence

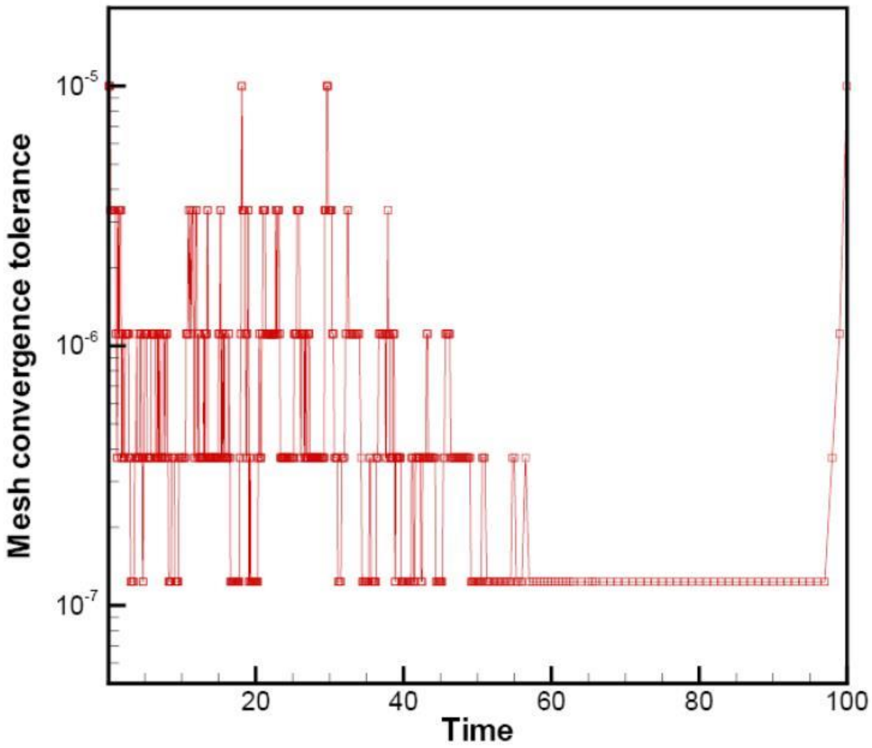
# Adaptation Results
# Time-Integrated Functional



# Adaptation Results
# Time-Integrated Functional




(a) Flow convergence tolerance




(b) Mesh convergence tolerance

# Combined Spatial-Temporal-Algebraic Error Estimation

- Previous example omitted spatial discretization error
  - Well known already
  - Complications for time-dependent mesh refinement (AMR)
- Must consider all 3 error sources simultaneously to reduce total simulation error
  - Use static mesh time-dependent case with exact solution
  - Time and convergence are 1-dimensional error spaces
    - Cost of adjoint is same as using twice as many time steps or twice the convergence tolerance
  - Maximum benefits come from reusing single adjoint calculation for all error sources

# Motivation

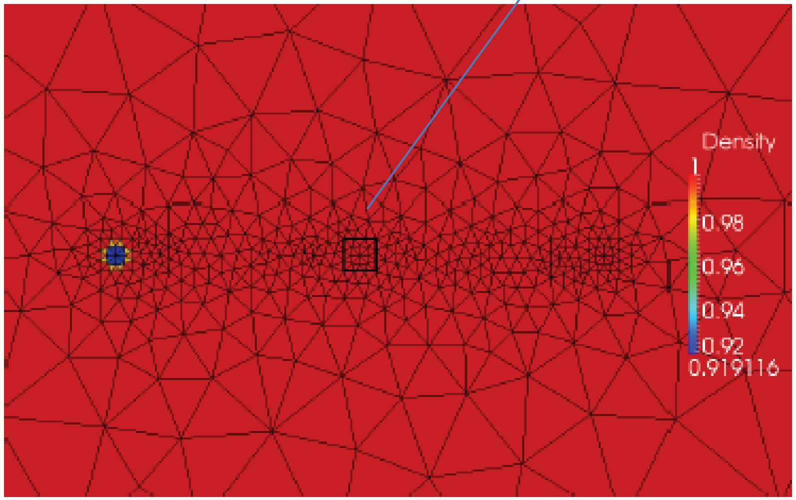
Isentropic Vortex

- Freestream ($M_\infty = 0.5$)
- Max Perturbation ($M = 0.2$)
- Core Radius ($R_c = 0.5$)

Objective Function

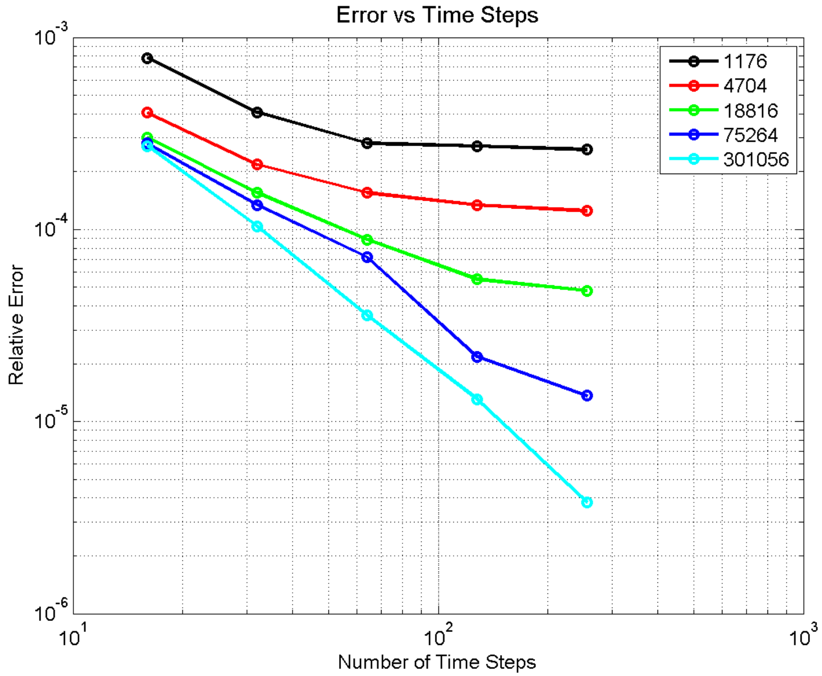
$$L(U) = \int_0^{60} \int_{-1}^{1} \int_{-1}^{1} \rho \, \mathrm{d}x\mathrm{d}y\mathrm{d}t$$

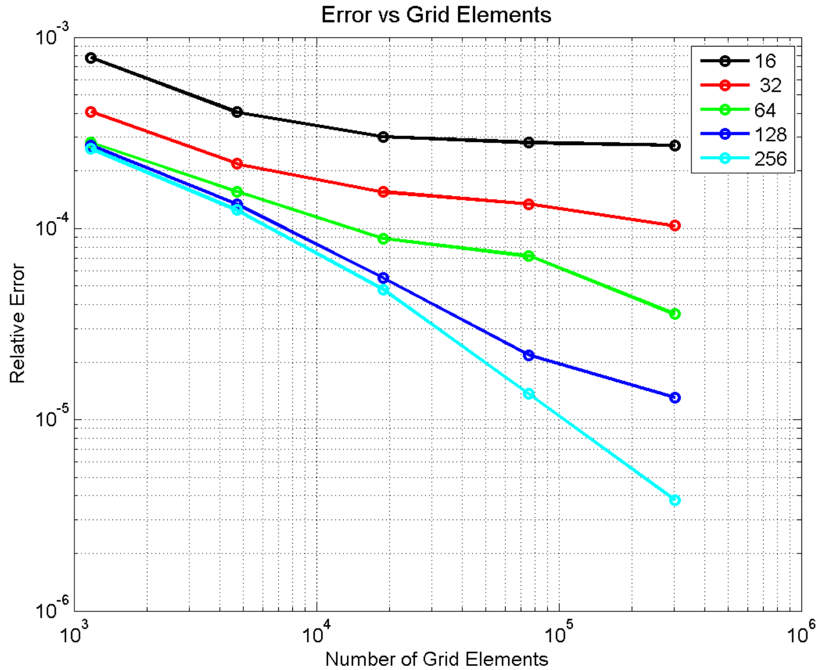
Exact (analytical) solution: 239.52558800471




- Examine total error (wrt exact functional) by refining in space, time and convergence tolerances

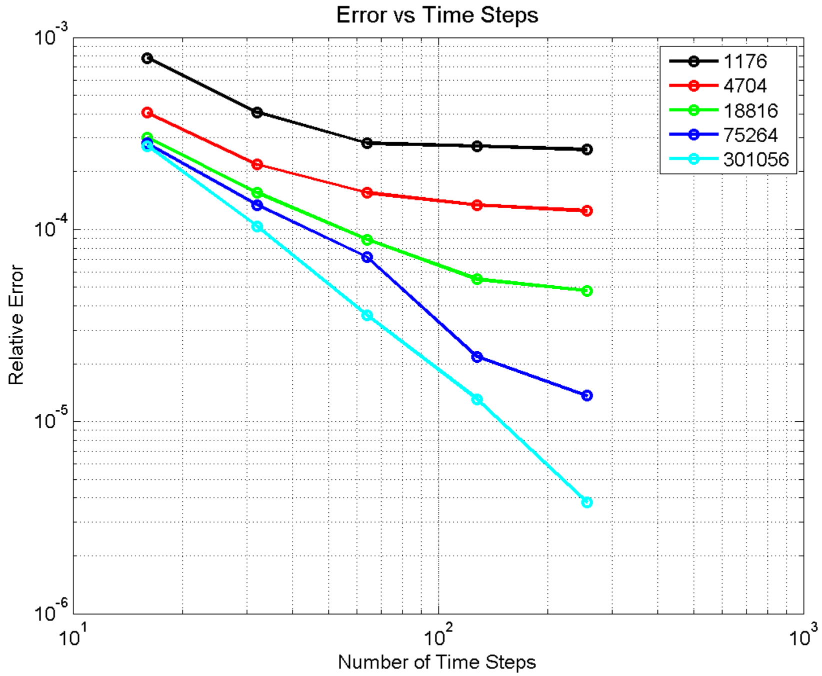
# Total Functional Error as Function of Refinement

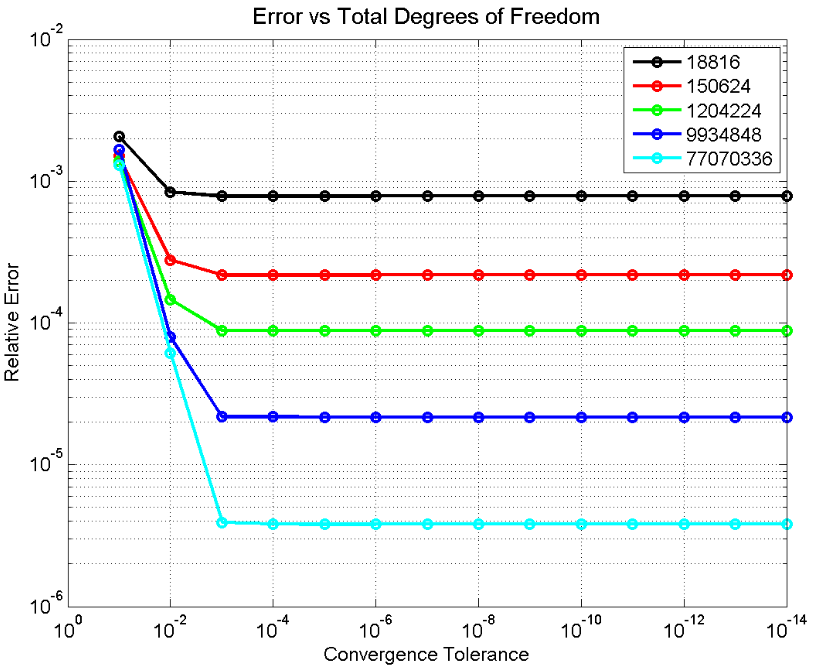



- Increasing temporal resolution ineffective at reducing total error on coarse grids
- Increasing spatial resolution ineffective at reducing total error using large time steps

# Total Functional Error as Function of Refinement




- Increasing temporal resolution ineffective at reducing total error on coarse grids
- Increasing convergence tolerance ineffective at reducing total error unless have fine mesh and time step resolution

# Combined Spatial-Temporal-Algebraic Error Estimation

- Equations over space and time:

$$R_h(U_h) = 0$$

- Goal is to estimate all error sources using a single adjoint solution (on coarse mesh, large time steps, partially converged)

$$\left[\frac{\partial \mathrm{R}(\mathrm{U})}{\partial \mathrm{U}}\right]^T \Lambda = -\left[\frac{\partial L(\mathrm{U})}{\partial \mathrm{U}}\right]^T$$

- Requires backwards integration in time

$$\begin{bmatrix}
\left[\frac{\partial \mathbf{R}^1(\mathbf{U})}{\partial \mathbf{U}^1}\right]^T & \left[\frac{\partial \mathbf{R}^2(\mathbf{U})}{\partial \mathbf{U}^1}\right]^T & \left[\frac{\partial \mathbf{R}^3(\mathbf{U})}{\partial \mathbf{U}^1}\right]^T & & & \\
 & \left[\frac{\partial \mathbf{R}^2(\mathbf{U})}{\partial \mathbf{U}^2}\right]^T & \left[\frac{\partial \mathbf{R}^3(\mathbf{U})}{\partial \mathbf{U}^2}\right]^T & \left[\frac{\partial \mathbf{R}^4(\mathbf{U})}{\partial \mathbf{U}^2}\right]^T & 0 & \\
 & & \ddots & \ddots & & \\
 & & & \left[\frac{\partial \mathbf{R}^{n-2}(\mathbf{U})}{\partial \mathbf{U}^{n-2}}\right]^T & \left[\frac{\partial \mathbf{R}^{n-1}(\mathbf{U})}{\partial \mathbf{U}^{n-2}}\right]^T & \left[\frac{\partial \mathbf{R}^{n}(\mathbf{U})}{\partial \mathbf{U}^{n-2}}\right]^T \\
 & 0 & & & \left[\frac{\partial \mathbf{R}^{n-1}(\mathbf{U})}{\partial \mathbf{U}^{n-1}}\right]^T & \left[\frac{\partial \mathbf{R}^{n}(\mathbf{U})}{\partial \mathbf{U}^{n-1}}\right]^T \\
 & & & & & \left[\frac{\partial \mathbf{R}^{n}(\mathbf{U})}{\partial \mathbf{U}^{n}}\right]^T
\end{bmatrix}
\begin{bmatrix} \Lambda^1 \\ \Lambda^2 \\ \Lambda^3 \\ \vdots \\ \Lambda^{n-1} \\ \Lambda^n \end{bmatrix}
= -\begin{bmatrix} \frac{\partial L}{\partial \mathbf{U}^1} \\ \frac{\partial L}{\partial \mathbf{U}^2} \\ \frac{\partial L}{\partial \mathbf{U}^3} \\ \vdots \\ \frac{\partial L}{\partial \mathbf{U}^{n-1}} \\ \frac{\partial L}{\partial \mathbf{U}^n} \end{bmatrix}^T$$

# Combined Error Estimation

- Spatial error estimate $\underbrace{L_s(\mathbf{U}_s) - L_s(\tilde{\mathbf{U}}_s)}_{\varepsilon_s} \cong \tilde{\mathbf{\Lambda}}_s^T \mathbf{R}_s(\tilde{\mathbf{U}}_s)$
  - Coarse solution projected onto fine mesh
  - Coarse adjoint projected onto fine mesh

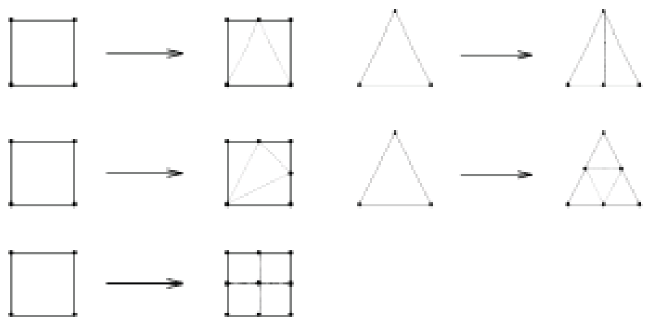
- Temporal error estimate $\underbrace{L_t(\mathbf{U}_t) - L_t(\tilde{\mathbf{U}}_t)}_{\varepsilon_t} \cong \tilde{\mathbf{\Lambda}}_t^T \mathbf{R}_t(\tilde{\mathbf{U}}_t)$
  - Coarse solution projected onto fine temporal domain
  - Coarse adjoint projected onto fine temporal domain

- Algebraic error estimate $\underbrace{L(\mathbf{U}) - L(\tilde{\mathbf{U}}_c)}_{\varepsilon_c} \cong \tilde{\mathbf{\Lambda}}_c^T \mathbf{R}(\tilde{\mathbf{U}}_c)$
  - No projections required

- Each error type estimated individually and used to drive adaptation of that error type

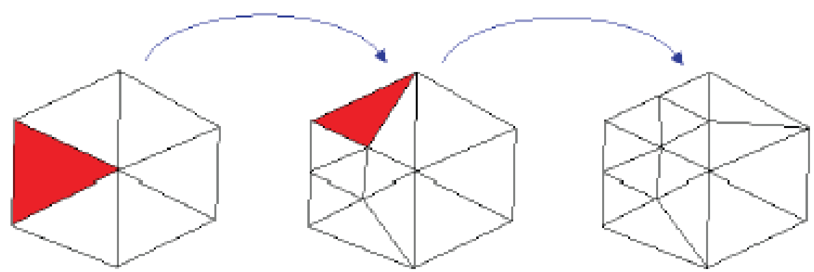
# Space-Time-Algebraic Refinement

- Allowable mesh patterns



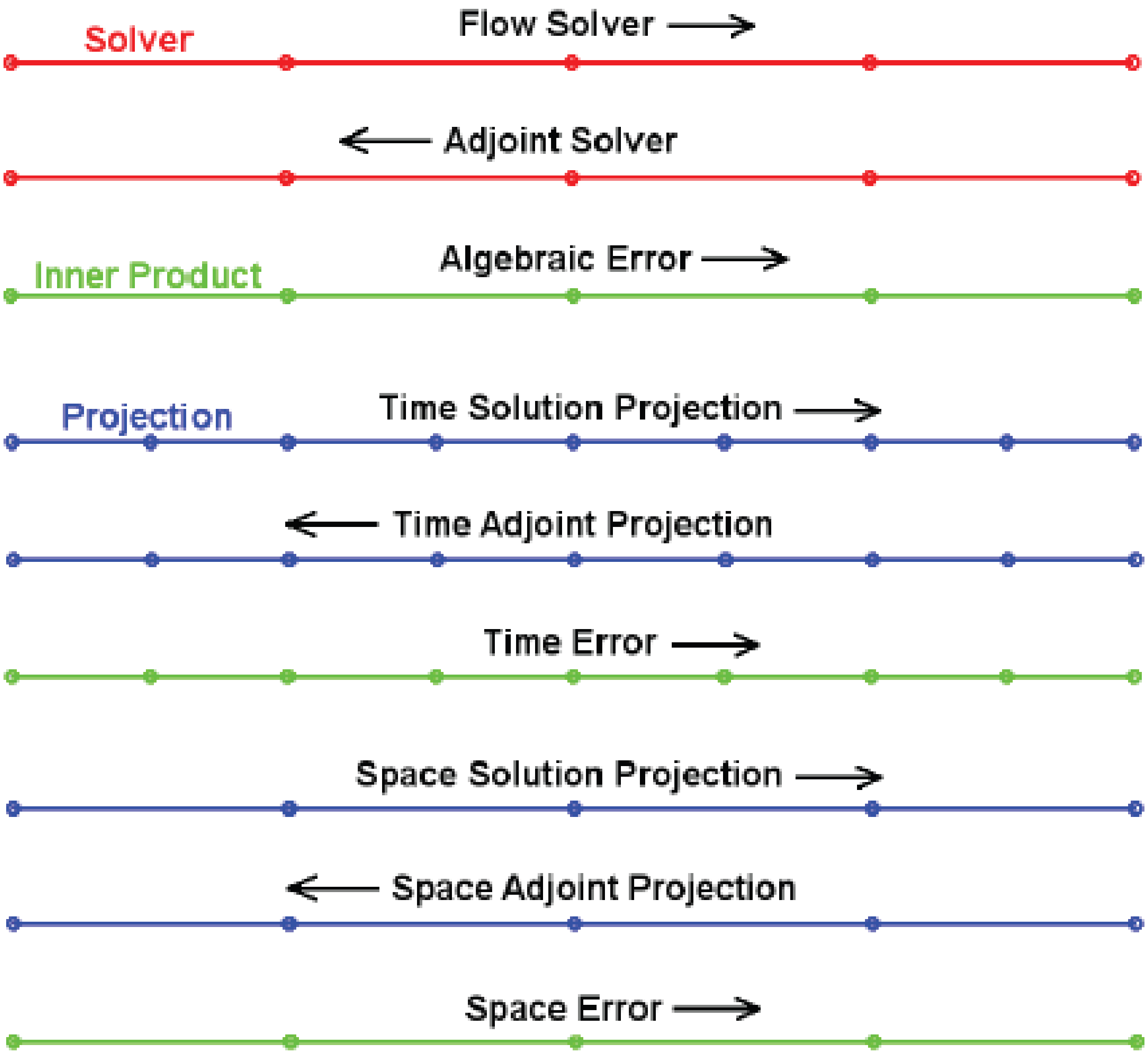
- Time is split 2:1
- 2:1 enforced between adjacent time intervals
- Variable time step BDF2

- Maintains 4:1 refinement



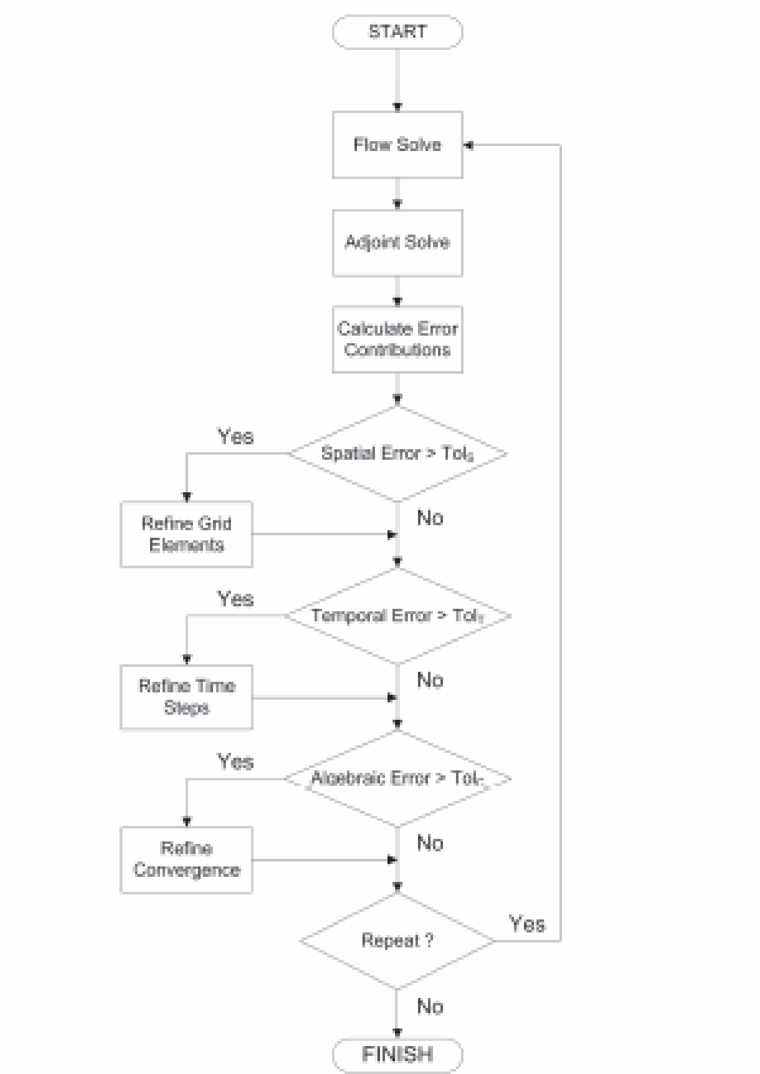
- Convergence tolerance reduced by 1 order of magnitude

# Computational Procedure




•

# Adaptive Error Control Strategy




- User specifies global error tolerance
  - Component error is equal fraction of global

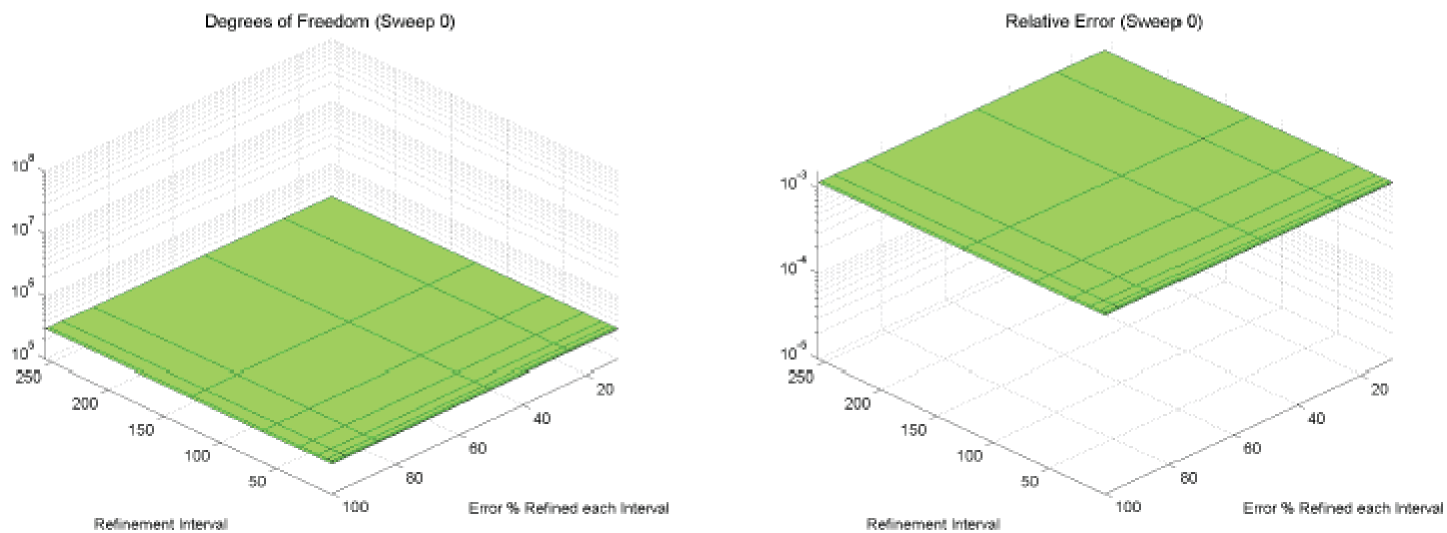
$$Tol_T = \frac{Tol_{Global}}{3}$$

$$Tol_S = \frac{Tol_{Global}}{3}$$

$$Tol_C = \frac{Tol_{Global}}{3}$$

# Spatial Refinement Strategies

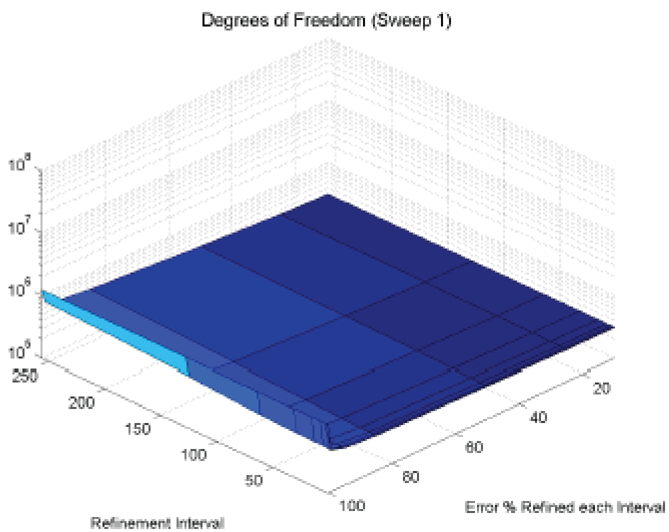
Initial Parameters (Sweep 0)

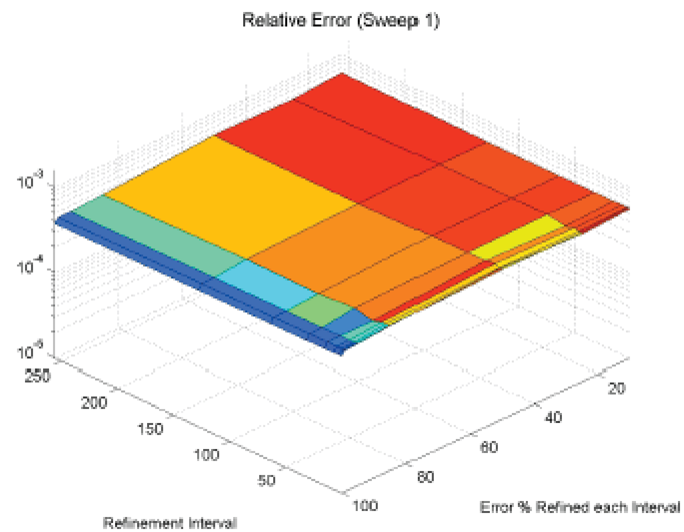



- Effect of refinement frequency and target error level

# Spatial Refinement Strategies

Sweep 1

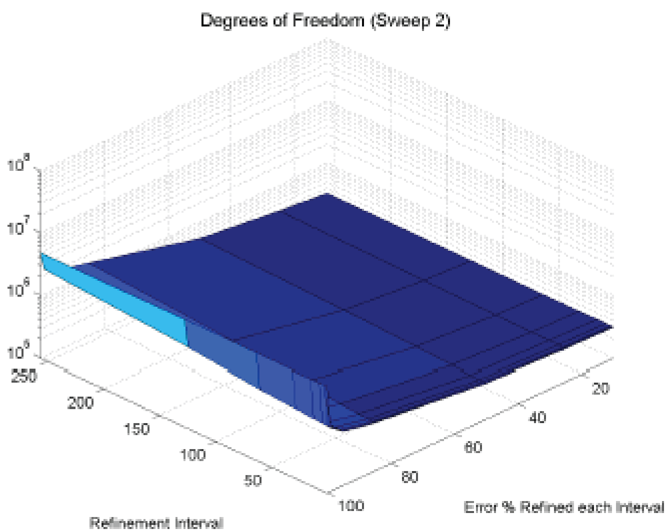


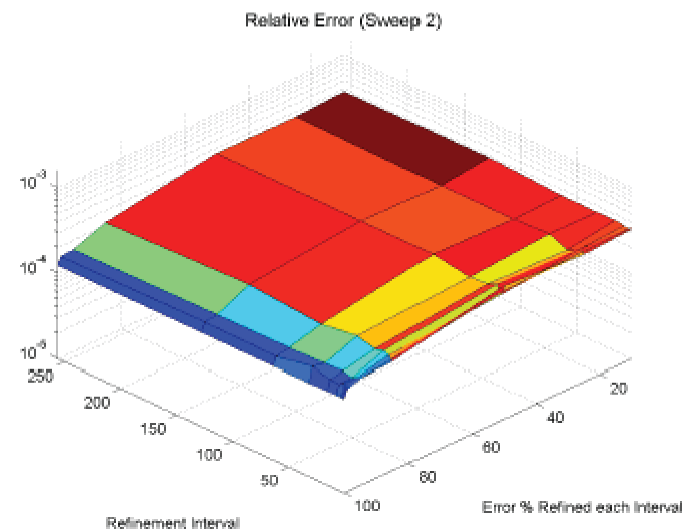




- Effect of refinement frequency and target error level

# Spatial Refinement Strategies

Sweep 2

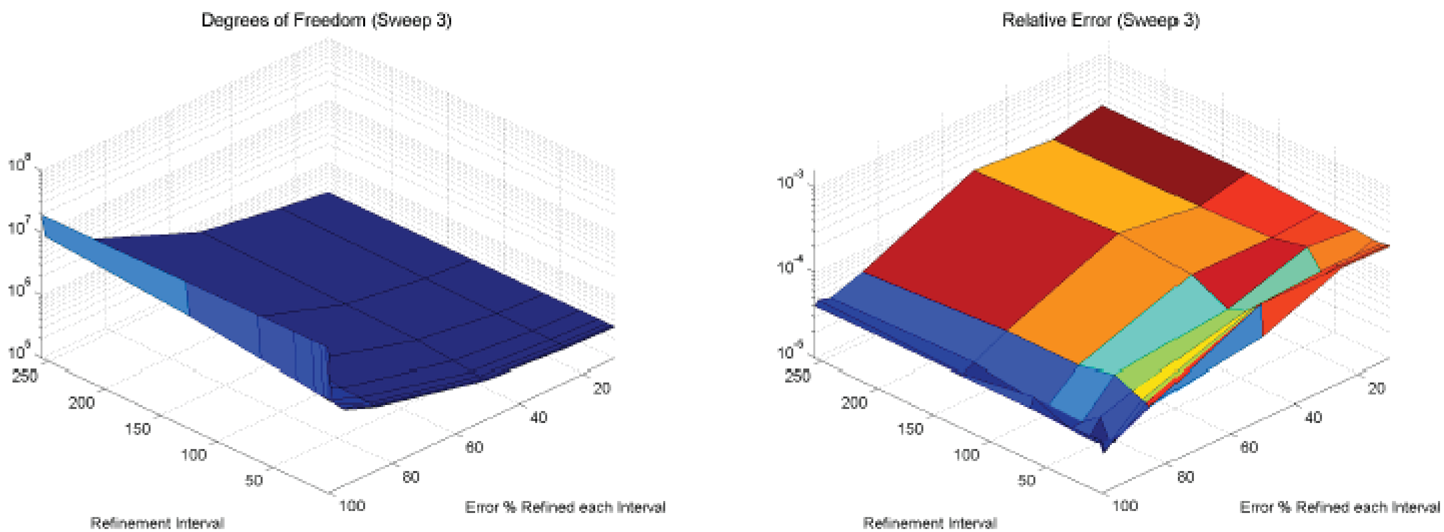



- Effect of refinement frequency and target error level

# Spatial Refinement Strategies

Sweep 3




- Effect of refinement frequency and target error level

# Spatial Refinement Strategies

- Lessons learned
  - Refine every time step
  - Initially refine only the largest errors then increase refinement near the end
- Previous work for steady state problems (Nemec, etc. AIAA-2008-0725)
  - Followed a very similar increasing refinement scheme
  - Showed an increasing error refinement tolerance always produced less costly computations
- Extend their work to unsteady solutions

# Spatial Refinement

- Define maximum allowable error $s$ for each cell

$$s = \frac{Tol_S}{\sum_{n=1}^{N} Elements_n}$$

- Refinement parameter $r_s^i$
  - Ratio of actual element error to allowed element error

$$r_s^i = \frac{\varepsilon_s^i}{s}$$

- Flag elements whose $r_s^i$ exceeds a threshold $\lambda_s$
  - Equidistribute error over every element of every time step $n$
  - Allows refinement every time step but does not force it

# Temporal Refinement

- Define maximum allowable error $t$ for each cell

$$t = \frac{Tol_T}{\sum_{n=1}^{N} Elements_n}$$

- Refinement parameter $r_t^i$
  - Ratio of temporal error per time step

$$r_t^n = \frac{\sum_{i=1}^{Elem^n} \varepsilon_t^i}{t \times Elements_n}$$

- Flag time steps whose $r_t^i$ exceeds a threshold $\lambda_t$
  - Equidistribute error over every time step $n$

# Algebraic Error Refinement

- Define maximum allowable error $c$ for each cell

$$c = \frac{Tol_C}{\sum_{n=1}^{N} Elements_n}$$

- Refinement parameter $r_c^i$
  - Ratio of convergence error per time step

$$r_c^n = \frac{\sum_{i=1}^{Elem^n} \varepsilon_c^i}{c \times Elements_n}$$

- Flag time steps whose $r_t^i$ exceeds a threshold $\lambda_t$
  - Equidistribute error over every time step $n$

# Test Case Initial Conditions

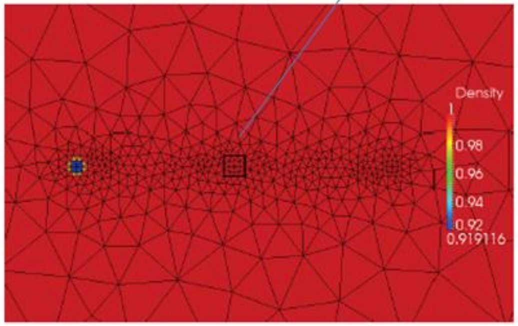
Isentropic Vortex

- Freestream ($M_\infty = 0.5$)
- Max Perturbation ($M = 0.2$)
- Core Radius ($R_c = 0.5$)

Objective Function

$$L(U) = \int_0^{60} \int_{-1}^{1} \int_{-1}^{1} \rho \, \mathrm{d}x\mathrm{d}y\mathrm{d}t$$

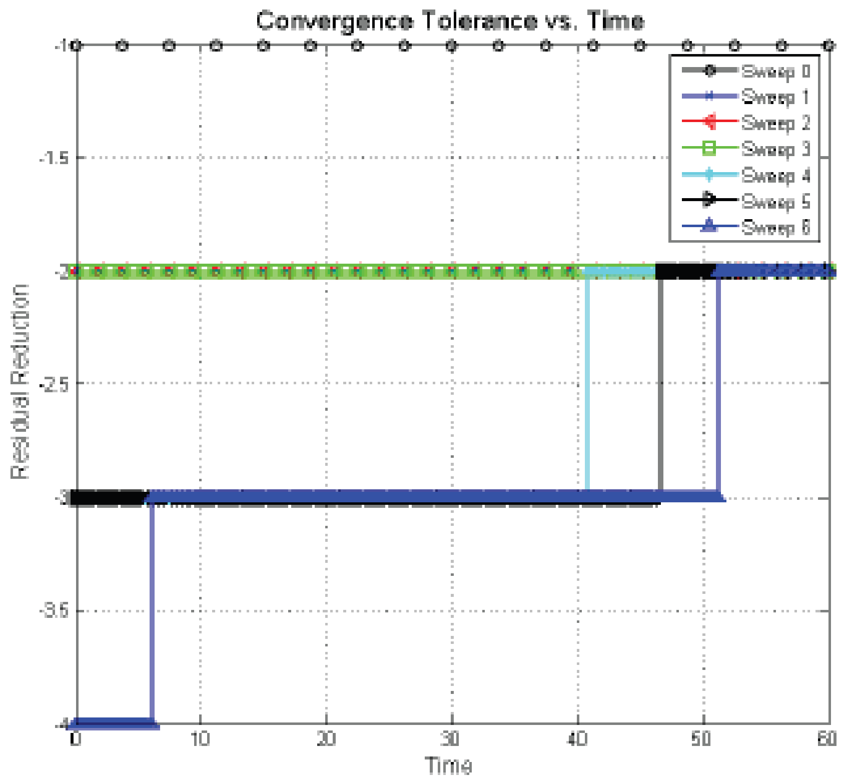
Exact (analytical) solution:
239.52558800471




- Convection of Isentropic Vortex
  - 16 initial time steps
  - 1176 grid elements at every time step
  - Initial residual converged 1 order of magnitude in $L_2$ norm
  - Threshold values 32, 16, 8, 4, 2, 1
  - $Tol_{Global} <= 10^{-3}$ for a relative error $<= 10^{-6}$

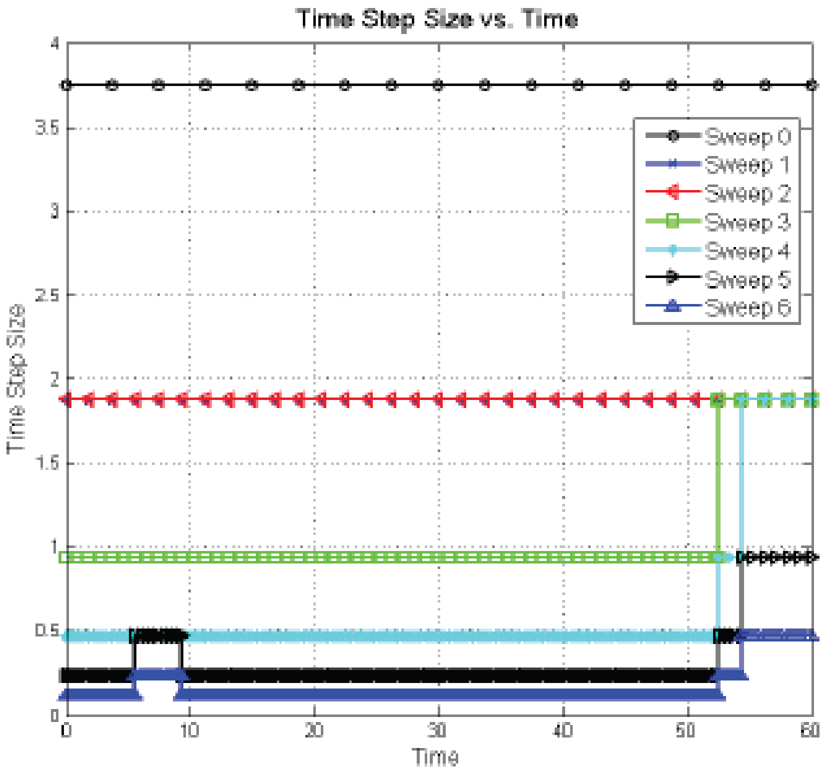
# Results
# Adaptive Convergence Tolerance

- Initial convergence tolerance refined
- Next 2 refinement sweeps the convergence tolerance was OK
- Last 3 refinement sweeps targeted the early time steps for refinement



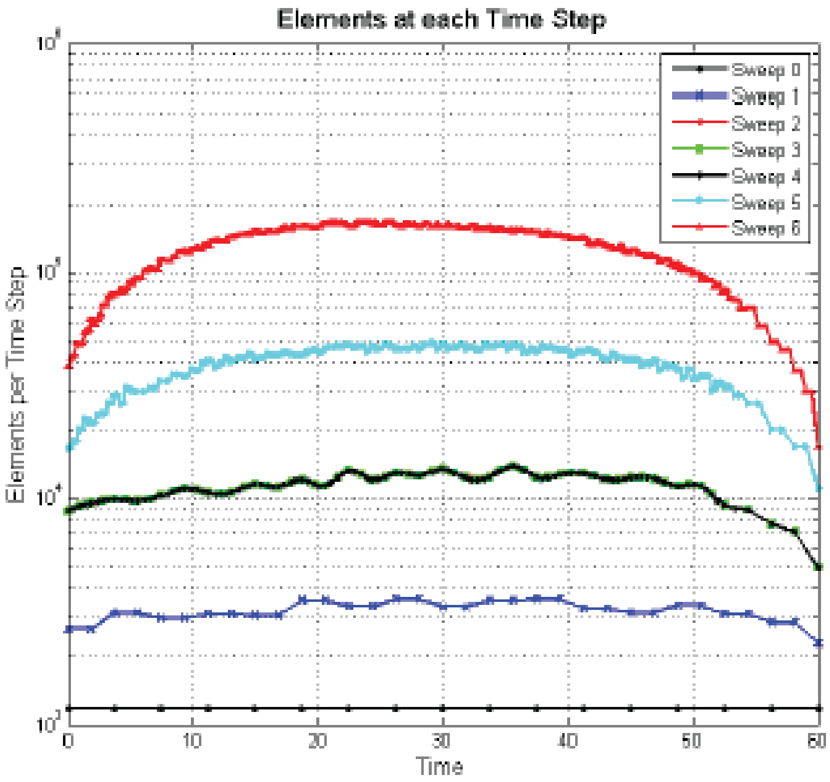
# Results
# Adaptive Time-Step Selection

- Initial time step size refined
- Next refinement sweep the time step size was OK
- Last 4 refinement sweeps targeted the early time steps for refinement



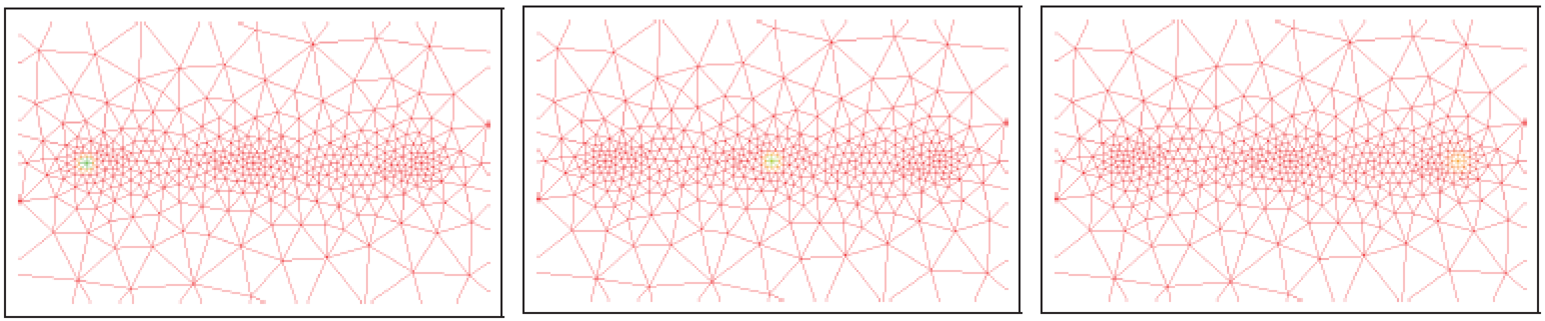
# Results
# Adaptive Mesh Refinement

- First 2 sweeps refined the mesh
- Next 2 sweeps mesh was OK
- Last 2 sweeps mesh was refined
- When the spatial error is above the tolerance all time steps had elements refined

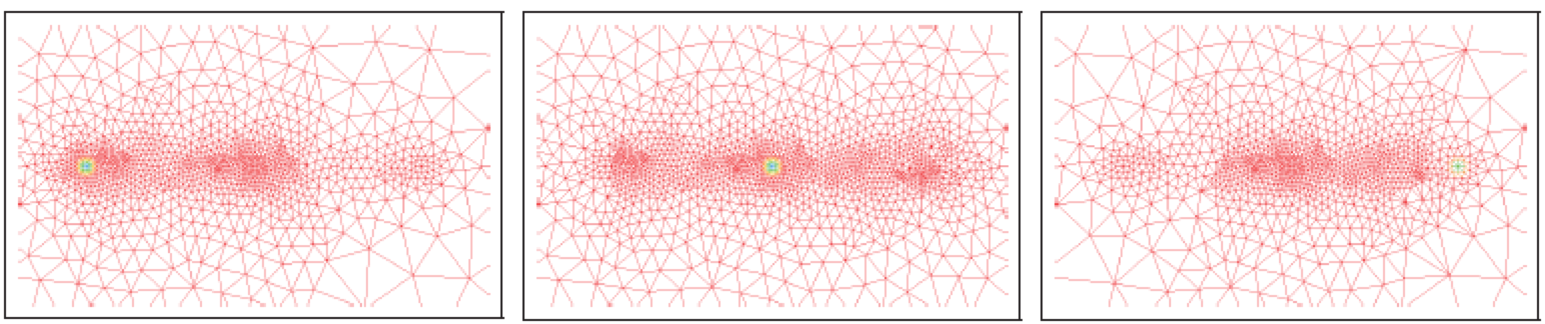


# Adaptive Mesh Refinement

Sweep 0

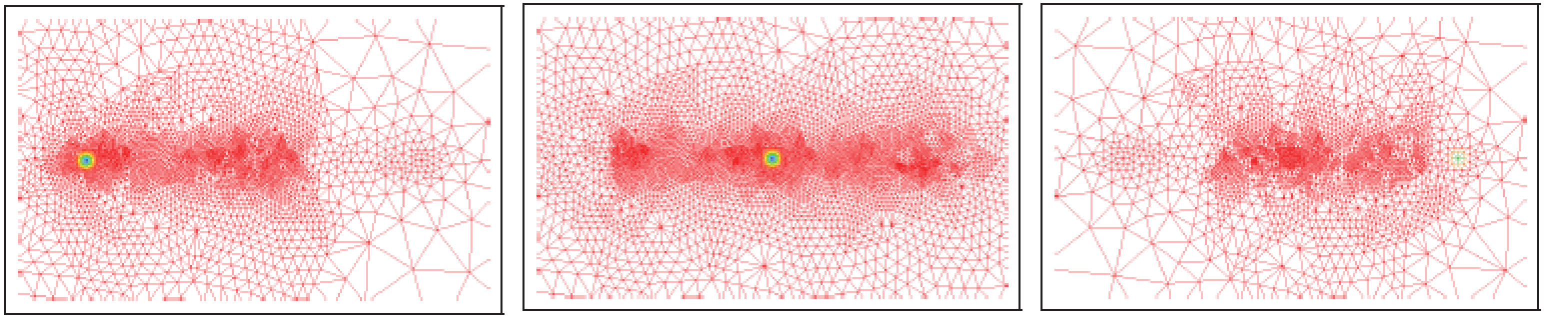


Sweep 1

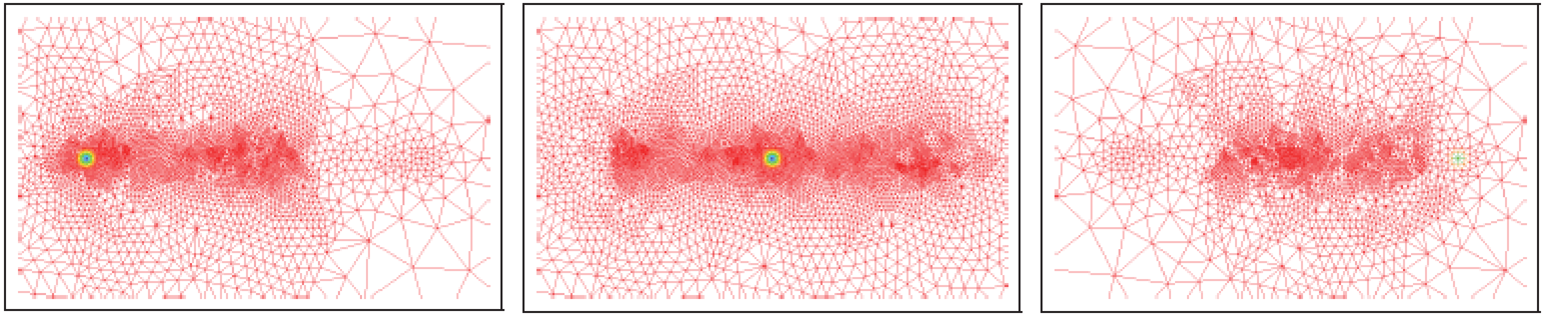


# Adaptive Mesh Refinement

Sweep 2

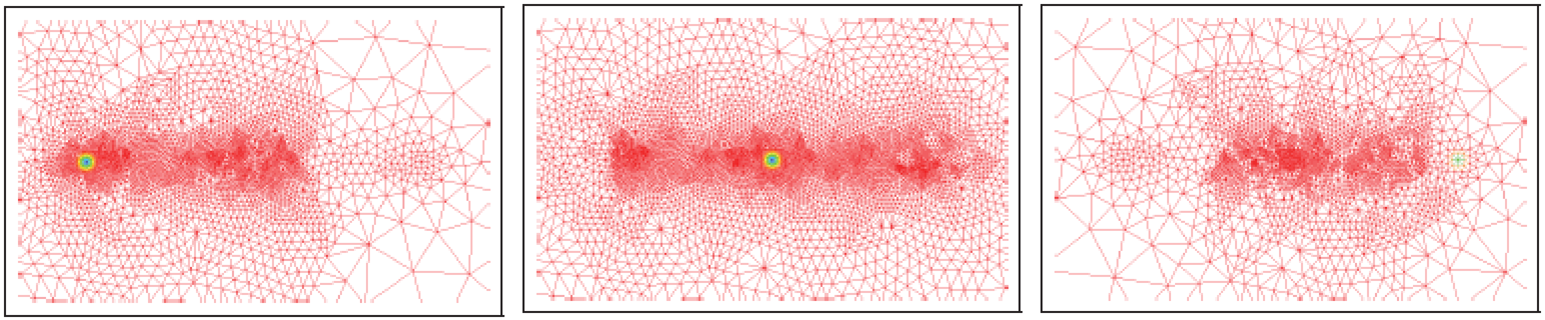


Sweep 3

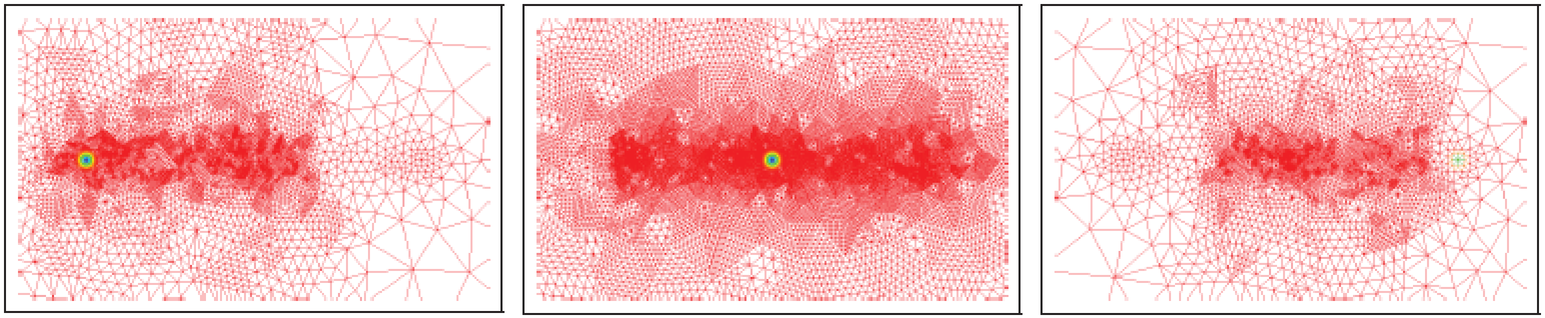


# Adaptive Mesh Refinement

Sweep 4

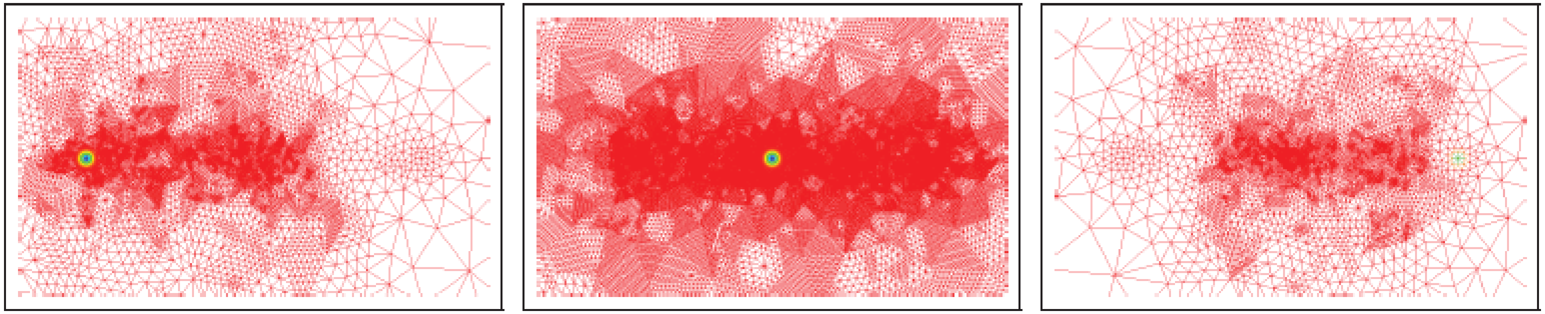


Sweep 5



# Adaptive Mesh Refinement

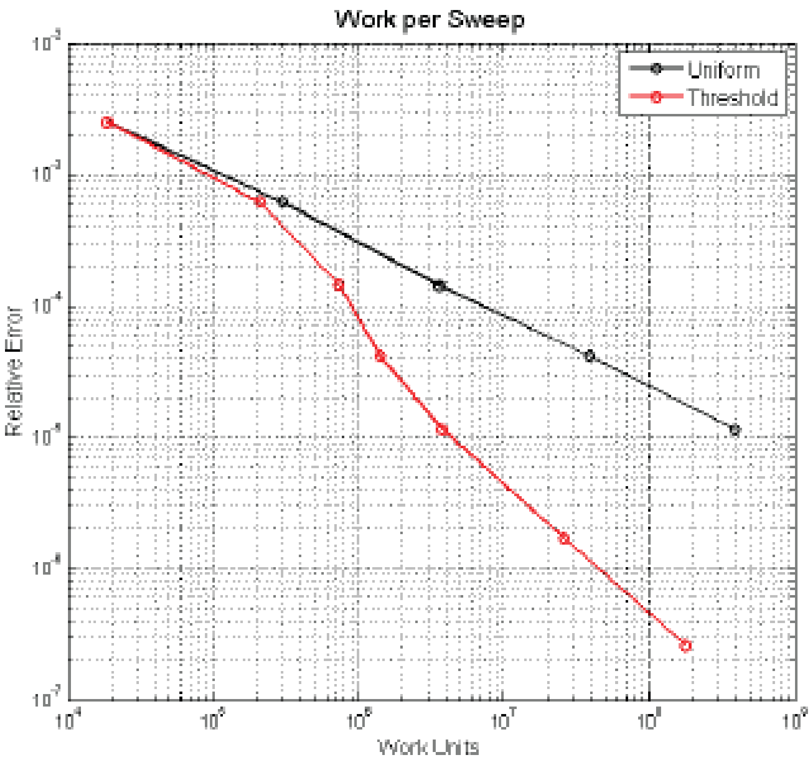
Sweep 6



- Refines mesh between vortex and integration region
  - Time integrated objective function
  - Heavy refinement when vortex is close to integrated region
  - Not as many refined elements at final time steps

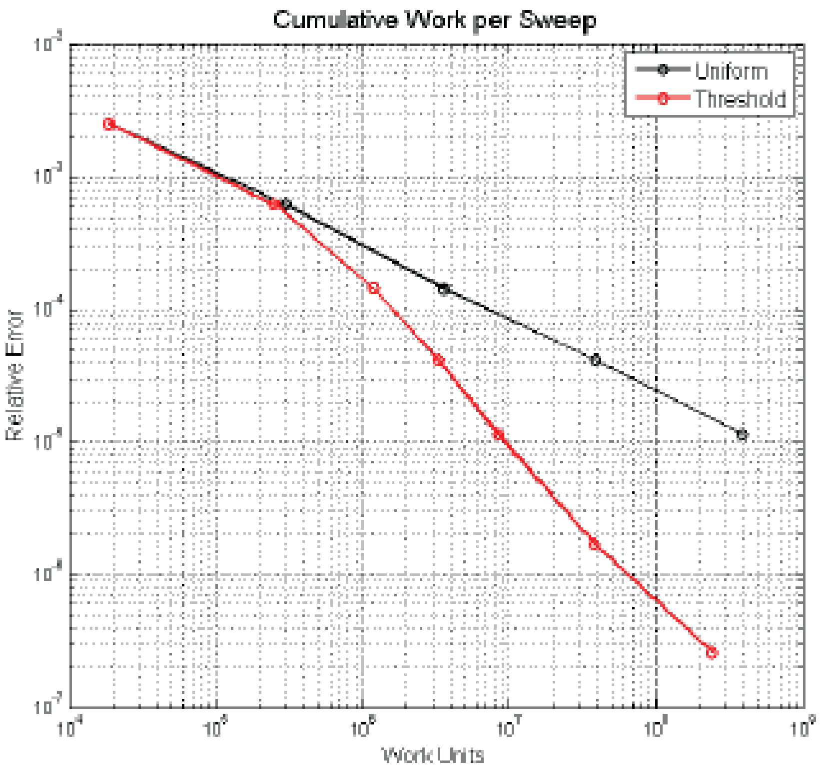
# Functional Error vs. Cost

- Work Unit
  - 1 Elements
  - 1 Time Step
  - 1 Order of Magnitude
- Uniform Refinement Sweep
  - Splits all time steps equally
  - Isotropically splits all elements
  - Increases convergence tolerance by 10
- Threshold
  - Threshold tolerance of 32, 16, 8, 4, 2, 1



# Functional Error vs Cost

- Uniform Refinement Sweep
  - Cost of only final solution
  - Same curve as previous
- Threshold
  - Total of all previous solution and adjoint costs
- Threshold
  - Final solution is most expensive so little added cost from previous steps
  - Large increase in accuracy



# Optimal Cost Error Reduction

- Obtain optimal total error reduction for given computational budget
  - Requires weighting of error with cost associate for reduction

Cost = Additional non-linear iterations required on an element

- Common parameter associated with all sources of error
- Non-linear solvers have a convergence rate
  - Newton's method ($q = 2$)

$$\lim_{n \to \infty} \frac{|\xi_{n+1}|}{|\xi_n|^q} = \mu$$

$$k = \frac{\log(\xi_{n+k})}{q \log(\xi_n)}$$

Where:

$$\begin{aligned} k &= \text{Non-linear steps} \\ \xi_n &= \text{Starting error} \\ \xi_{n+k} &= \text{Final Error} \end{aligned}$$

# Optimal Cost Error Reduction

Cost of Refinement ($CoR$) for each error type

$$CoR_c = (\text{Elements within Time Step}) \times \Delta k$$
$$CoR_t = (\text{Elements within Time Step}) \times k$$
$$CoR_s = (\text{New Elements} = 3) \times k$$

Use this cost to normalize all error types

- Allows comparison between discretization sources
- Simple application (Error/Cost) $Tol_R$

# Optimal Cost Error Reduction

Determine how much “excess“ error exists in solution

$$\varepsilon_{ex} = \varepsilon_{cst} - Tol_R$$

Refine largest error/cost refinements until $\varepsilon_{ex}$ is accounted for

- Previous research has shown an increasing error refinement tolerance always produces less costly computations.
  - Nemec, etc. AIAA-2008-0725

Modified using $(\lambda = 16, 8, 4, 2, 1, ...)$

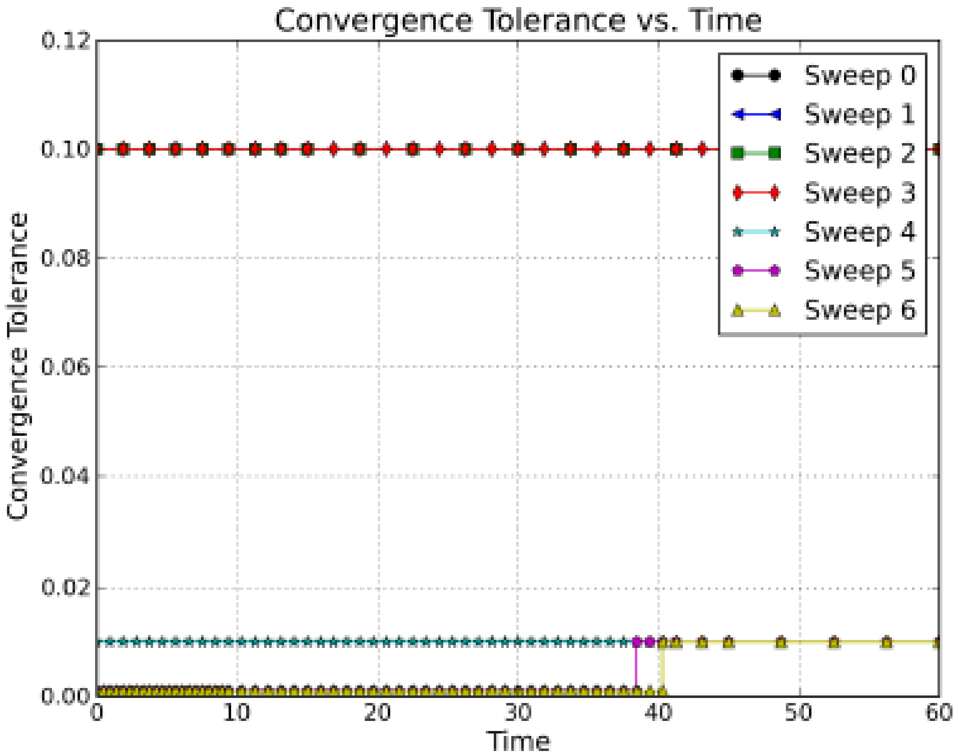
$$\varepsilon_{ex} = \frac{\varepsilon_{cst} - Tol_R}{\lambda}$$

# Optimal Cost Error Reduction

- Convection of Isentropic Vortex
  - 16 initial time steps
  - 1176 grid elements at every time step
  - Initial residual converged 1 order of magnitude in $L_2$ norm
  - Threshold values 16, 8, 4, 2, 1, 1
  - $Tol_{Global} <= 10^{-3}$ for a relative error $<= 10^{-6}$

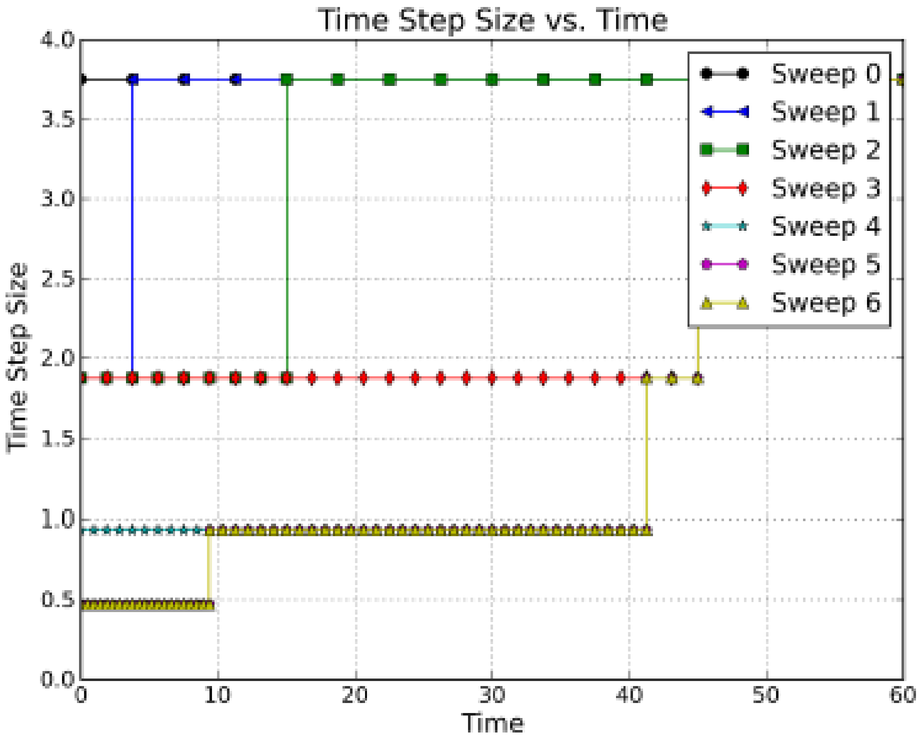
# Results
# Algebraic Error

- No refinement until after $4^{th}$ solution sweep
  - All steps refined
  - Reducing threashold value $(4 \rightarrow 2)$
- Refinement targets initial time steps

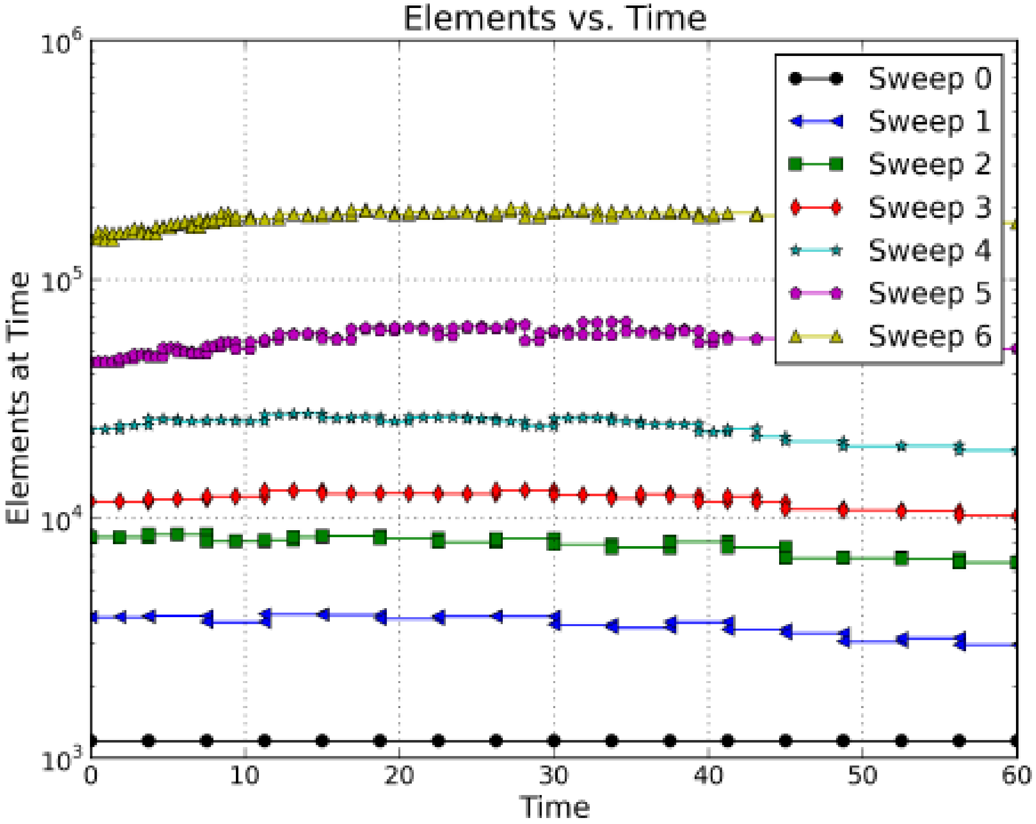


# Results
# Temporal Error

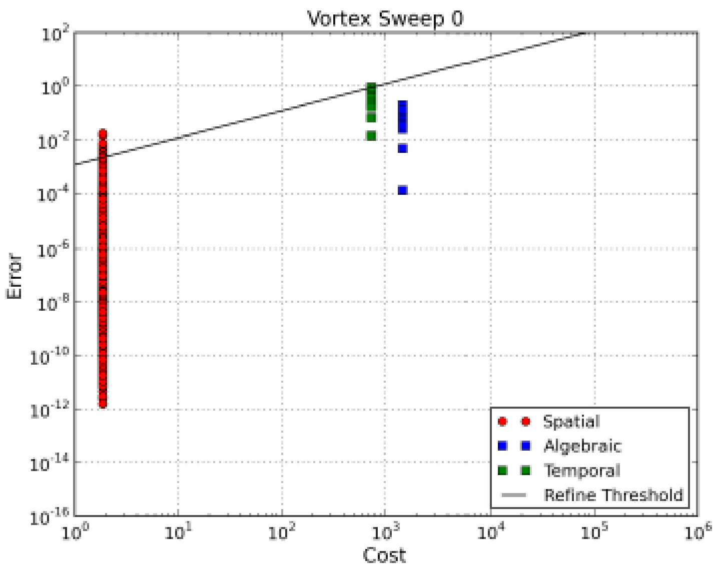
- Initial time step size refined
- Each refinement sweep has some time steps refined
- Maximum of 3 refinements

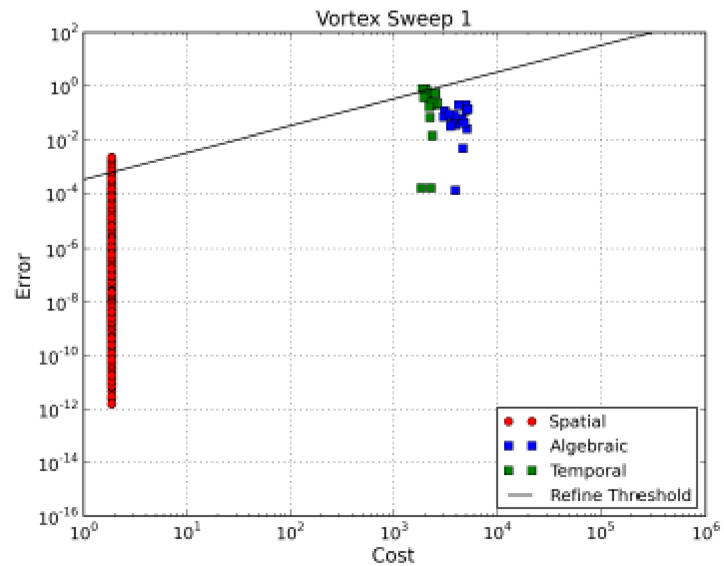


# Results
# Spatial Error



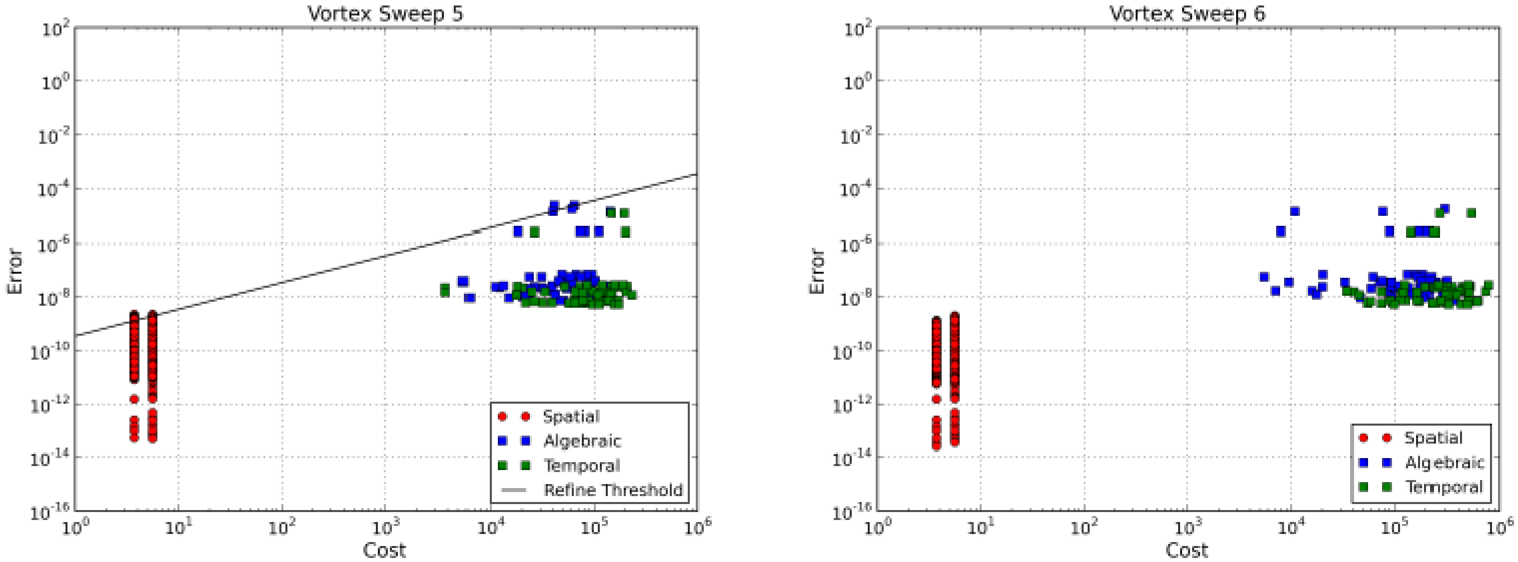
# Optimal Cost Error Control




- Line depicts constant Error/Cost Threshold
  – Refinement opportunities above line to be excercised
  – Refining a single spatial element is inexpensive
  – Temporal/Algebraic refinement apply to all elements (more expensive)

# Optimal Cost Error Control




- Line depicts constant Error/Cost Threshold
  - Refinement opportunities above line to be excercised
  - Refining a single spatial element is inexpensive
  - Temporal/Algebraic refinement apply to all elements (more expensive)

# Summary/Trends

- Cost weighted refinement tends to perform more spatial refinement because of lower cost
  - Can add individual new mesh cells
    - Inherit time step and convergence tolerances of parent cells
  - Temporal refinement results in 1 new time step for all mesh cells
  - Convergence tolerance refinement applies to all mesh cells
- Overall delivers lowest total error for fixed computational budget

# Generalized Error Form for Time-dependent Multidisciplinary Problems

- Consider multidisciplinary objective given by

$$L = L(\mathscr{U}) = L(\mathscr{U}_1, \mathscr{U}_2, \cdots, \mathscr{U}_m)$$

- With coupled disciplinary residual equations to be satisfied over all space and time

$$\begin{aligned}
\mathscr{R}_1(\mathscr{U}_1, \mathscr{U}_2, \cdots, \mathscr{U}_m) &= 0 \\
\mathscr{R}_2(\mathscr{U}_1, \mathscr{U}_2, \cdots, \mathscr{U}_m) &= 0 \\
&\vdots \\
\mathscr{R}_m(\mathscr{U}_1, \mathscr{U}_2, \cdots, \mathscr{U}_m) &= 0
\end{aligned}$$

# Generalized Error Form for Time-dependent Multidisciplinary Problems

- Taking Taylor series expansion of objective about approximate state

$$
\begin{aligned}
L(\mathscr{U}) = L(\tilde{\mathscr{U}}) &+ \left[\frac{\partial L}{\partial \mathscr{U}_1}\right]_{\tilde{\mathscr{U}}} (\mathscr{U}_1 - \tilde{\mathscr{U}}_1) \\
&+ \left[\frac{\partial L}{\partial \mathscr{U}_2}\right]_{\tilde{\mathscr{U}}} (\mathscr{U}_2 - \tilde{\mathscr{U}}_2) \\
&\vdots \\
&+ \left[\frac{\partial L}{\partial \mathscr{U}_m}\right]_{\tilde{\mathscr{U}}} (\mathscr{U}_m - \tilde{\mathscr{U}}_m) \\
&+ \mathcal{O}\left(\mathscr{U} - \tilde{\mathscr{U}}\right)^2 + \mathcal{O}\left(\mathscr{U} - \tilde{\mathscr{U}}\right)^3 \cdots
\end{aligned}
$$

$$
L(\mathscr{U}) - L(\tilde{\mathscr{U}}) = \left[\begin{array}{cccc} \frac{\partial L}{\partial \mathscr{U}_1} & \frac{\partial L}{\partial \mathscr{U}_2} & \cdots & \frac{\partial L}{\partial \mathscr{U}_m} \end{array}\right]_{\tilde{\mathscr{U}}} \left[\begin{array}{c} (\mathscr{U}_{1h} - \tilde{\mathscr{U}}_1) \\ (\mathscr{U}_{2h} - \tilde{\mathscr{U}}_2) \\ \vdots \\ (\mathscr{U}_{mh} - \tilde{\mathscr{U}}_m) \end{array}\right]
$$

# Generalized Error Form for Time-dependent Multidisciplinary Problems

- Linearizing multidisciplinary residual equations

$$\begin{bmatrix} \mathscr{R}_1(\mathscr{U}) \\ \mathscr{R}_2(\mathscr{U}) \\ \vdots \\ \mathscr{R}_m(\mathscr{U}) \end{bmatrix} \approx \begin{bmatrix} \mathscr{R}_1(\tilde{\mathscr{U}}) \\ \mathscr{R}_2(\tilde{\mathscr{U}}) \\ \vdots \\ \mathscr{R}_m(\tilde{\mathscr{U}}) \end{bmatrix} + \begin{bmatrix} \frac{\partial \mathscr{R}_1}{\partial \mathscr{U}_1} & \frac{\partial \mathscr{R}_1}{\partial \mathscr{U}_2} & \cdots & \frac{\partial \mathscr{R}_1}{\partial \mathscr{U}_m} \\ \frac{\partial \mathscr{R}_2}{\partial \mathscr{U}_1} & \frac{\partial \mathscr{R}_2}{\partial \mathscr{U}_2} & \cdots & \frac{\partial \mathscr{R}_2}{\partial \mathscr{U}_m} \\ \vdots & & & \\ \frac{\partial \mathscr{R}_m}{\partial \mathscr{U}_1} & \frac{\partial \mathscr{R}_m}{\partial \mathscr{U}_2} & \cdots & \frac{\partial \mathscr{R}_m}{\partial \mathscr{U}_m} \end{bmatrix}_{\tilde{\mathscr{U}}} \begin{bmatrix} (\mathscr{U}_1 - \tilde{\mathscr{U}}_1) \\ (\mathscr{U}_2 - \tilde{\mathscr{U}}_2) \\ \vdots \\ (\mathscr{U}_m - \tilde{\mathscr{U}}_m) \end{bmatrix} = 0$$

$$\begin{bmatrix} (\mathscr{U}_1 - \tilde{\mathscr{U}}_1) \\ (\mathscr{U}_2 - \tilde{\mathscr{U}}_2) \\ \vdots \\ (\mathscr{U}_m - \tilde{\mathscr{U}}_m) \end{bmatrix} = - \begin{bmatrix} \frac{\partial \mathscr{R}_1}{\partial \mathscr{U}_1} & \frac{\partial \mathscr{R}_1}{\partial \mathscr{U}_2} & \cdots & \frac{\partial \mathscr{R}_1}{\partial \mathscr{U}_m} \\ \frac{\partial \mathscr{R}_2}{\partial \mathscr{U}_1} & \frac{\partial \mathscr{R}_2}{\partial \mathscr{U}_2} & \cdots & \frac{\partial \mathscr{R}_2}{\partial \mathscr{U}_m} \\ \vdots & & & \\ \frac{\partial \mathscr{R}_m}{\partial \mathscr{U}_1} & \frac{\partial \mathscr{R}_m}{\partial \mathscr{U}_2} & \cdots & \frac{\partial \mathscr{R}_m}{\partial \mathscr{U}_m} \end{bmatrix}_{\tilde{\mathscr{U}}}^{-1} \begin{bmatrix} \mathscr{R}_1(\tilde{\mathscr{U}}) \\ \mathscr{R}_2(\tilde{\mathscr{U}}) \\ \vdots \\ \mathscr{R}_m(\tilde{\mathscr{U}}) \end{bmatrix}$$

# Generalized Error Form for Time-dependent Multidisciplinary Problems

$$\begin{bmatrix} (\mathscr{U}_1 - \tilde{\mathscr{U}}_1) \\ (\mathscr{U}_2 - \tilde{\mathscr{U}}_2) \\ \vdots \\ (\mathscr{U}_m - \tilde{\mathscr{U}}_m) \end{bmatrix} = - \begin{bmatrix} \frac{\partial \mathscr{R}_1}{\partial \mathscr{U}_1} & \frac{\partial \mathscr{R}_1}{\partial \mathscr{U}_2} & \cdots & \frac{\partial \mathscr{R}_1}{\partial \mathscr{U}_m} \\ \frac{\partial \mathscr{R}_2}{\partial \mathscr{U}_1} & \frac{\partial \mathscr{R}_2}{\partial \mathscr{U}_2} & \cdots & \frac{\partial \mathscr{R}_2}{\partial \mathscr{U}_m} \\ \vdots & & & \\ \frac{\partial \mathscr{R}_m}{\partial \mathscr{U}_1} & \frac{\partial \mathscr{R}_m}{\partial \mathscr{U}_2} & \cdots & \frac{\partial \mathscr{R}_m}{\partial \mathscr{U}_m} \end{bmatrix}_{\tilde{\mathscr{U}}}^{-1} \begin{bmatrix} \mathscr{R}_1(\tilde{\mathscr{U}}) \\ \mathscr{R}_2(\tilde{\mathscr{U}}) \\ \vdots \\ \mathscr{R}_m(\tilde{\mathscr{U}}) \end{bmatrix}$$

# Generalized Error Form for Time-dependent Multidisciplinary Problems

- Substituting into objective linearization

$$L(\mathscr{U}) - L(\tilde{\mathscr{U}}) =$$

$$\underbrace{-\begin{bmatrix} \frac{\partial L}{\partial \mathscr{U}_1} & \frac{\partial L}{\partial \mathscr{U}_2} & \cdots & \frac{\partial L}{\partial \mathscr{U}_m} \end{bmatrix}_{\tilde{\mathscr{U}}} \begin{bmatrix} \frac{\partial \mathscr{R}_1}{\partial \mathscr{U}_1} & \frac{\partial \mathscr{R}_1}{\partial \mathscr{U}_2} & \cdots & \frac{\partial \mathscr{R}_1}{\partial \mathscr{U}_m} \\ \frac{\partial \mathscr{R}_2}{\partial \mathscr{U}_1} & \frac{\partial \mathscr{R}_2}{\partial \mathscr{U}_2} & \cdots & \frac{\partial \mathscr{R}_2}{\partial \mathscr{U}_m} \\ \vdots & & & \\ \frac{\partial \mathscr{R}_m}{\partial \mathscr{U}_1} & \frac{\partial \mathscr{R}_m}{\partial \mathscr{U}_2} & \cdots & \frac{\partial \mathscr{R}_m}{\partial \mathscr{U}_m} \end{bmatrix}_{\tilde{\mathscr{U}}}^{-1}}_{\Lambda^T} \begin{bmatrix} \mathscr{R}_1(\tilde{\mathscr{U}}) \\ \mathscr{R}_2(\tilde{\mathscr{U}}) \\ \vdots \\ \mathscr{R}_m(\tilde{\mathscr{U}}) \end{bmatrix}$$

$$\begin{bmatrix} \frac{\partial \mathscr{R}_1}{\partial \mathscr{U}_1}^T & \frac{\partial \mathscr{R}_2}{\partial \mathscr{U}_1}^T & \cdots & \frac{\partial \mathscr{R}_m}{\partial \mathscr{U}_1}^T \\ \frac{\partial \mathscr{R}_1}{\partial \mathscr{U}_2}^T & \frac{\partial \mathscr{R}_2}{\partial \mathscr{U}_2}^T & \cdots & \frac{\partial \mathscr{R}_m}{\partial \mathscr{U}_2}^T \\ \vdots & & & \\ \frac{\partial \mathscr{R}_1}{\partial \mathscr{U}_m}^T & \frac{\partial \mathscr{R}_2}{\partial \mathscr{U}_m}^T & \cdots & \frac{\partial \mathscr{R}_m}{\partial \mathscr{U}_m}^T \end{bmatrix}_{\tilde{\mathscr{U}}} \begin{bmatrix} \Lambda_{\mathscr{U}_1} \\ \Lambda_{\mathscr{U}_2} \\ \vdots \\ \Lambda_{\mathscr{U}_m} \end{bmatrix} = - \begin{bmatrix} \frac{\partial L}{\partial \mathscr{U}_1}^T \\ \frac{\partial L}{\partial \mathscr{U}_2}^T \\ \vdots \\ \frac{\partial L}{\partial \mathscr{U}_m}^T \end{bmatrix}_{\tilde{\mathscr{U}}}$$

# Generalized Error Form for Time-dependent Multidisciplinary Problems

$$\begin{bmatrix} \frac{\partial \mathcal{R}_1}{\partial \mathcal{U}_1}^T & \frac{\partial \mathcal{R}_2}{\partial \mathcal{U}_1}^T & \cdots & \frac{\partial \mathcal{R}_m}{\partial \mathcal{U}_1}^T \\ \frac{\partial \mathcal{R}_1}{\partial \mathcal{U}_2}^T & \frac{\partial \mathcal{R}_2}{\partial \mathcal{U}_2}^T & \cdots & \frac{\partial \mathcal{R}_m}{\partial \mathcal{U}_2}^T \\ \vdots & & & \\ \frac{\partial \mathcal{R}_1}{\partial \mathcal{U}_m}^T & \frac{\partial \mathcal{R}_2}{\partial \mathcal{U}_m}^T & \cdots & \frac{\partial \mathcal{R}_m}{\partial \mathcal{U}_m}^T \end{bmatrix}_{\tilde{\mathcal{U}}} \begin{bmatrix} \Lambda_{\mathcal{U}_1} \\ \Lambda_{\mathcal{U}_2} \\ \vdots \\ \Lambda_{\mathcal{U}_m} \end{bmatrix} = - \begin{bmatrix} \frac{\partial L}{\partial \mathcal{U}_1}^T \\ \frac{\partial L}{\partial \mathcal{U}_2}^T \\ \vdots \\ \frac{\partial L}{\partial \mathcal{U}_m}^T \end{bmatrix}_{\tilde{\mathcal{U}}}$$

# Generalized Error Form for Time-dependent Multidisciplinary Problems

$$\varepsilon_{total} = L(\mathscr{U}) - L(\tilde{\mathscr{U}}) = +\begin{bmatrix} \Lambda^T_{\mathscr{U}_1} & \Lambda^T_{\mathscr{U}_2} & \cdots & \Lambda^T_{\mathscr{U}_m} \end{bmatrix} \begin{bmatrix} \mathscr{R}_1(\tilde{\mathscr{U}}) \\ \mathscr{R}_2(\tilde{\mathscr{U}}) \\ \vdots \\ \mathscr{R}_m(\tilde{\mathscr{U}}) \end{bmatrix}$$

$$\begin{aligned} \varepsilon_{total} = L(\mathscr{U}) - L(\tilde{\mathscr{U}}) = \\ + \left\{ \Lambda^T_{\mathscr{U}_1} \mathscr{R}_1(\tilde{\mathscr{U}}) + \Lambda^T_{\mathscr{U}_2} \mathscr{R}_2(\tilde{\mathscr{U}}) + \cdots + \Lambda^T_{\mathscr{U}_m} \mathscr{R}_m(\tilde{\mathscr{U}}) \right\} \end{aligned}$$

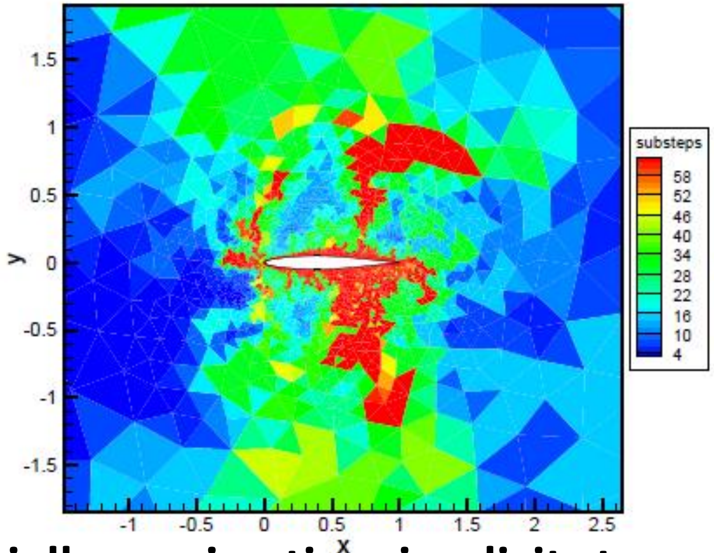
- Error is broken down into disciplinary contributions
  - Spatial, temporal, algebraic error of each discipline
  - Coupling error using fully converged disciplines with lagged values
  - Disciplinary modeling error possible if can project low fidelity model solution to high fidelity space

# Conclusions

- Adjoint methods allow estimation and control of error for specific simulation outputs
- Using a single adjoint solution it is possible to estimate and adaptively control various sources of error
  - Spatial
  - Temporal
  - Algebraic
- Techniques extend naturally to multidisciplinary problems

# Conclusions

- Focus has been on techniques that can be applied to existing production level simulation codes
- Further optimizations are possible if discretization/solvers are designed with adaptive error control in mind from the outset
  - Space-time formulations
  - Variable local solver tolerances
  - h-p discretizations




**Spatially varying time implicit steps**

# Conclusions

- Novel discretizations /solvers hold promise for large gains in efficiency and accuracy
- Extending even current spatial-temporal-algebraic error estimation and control techniques to 3D time-dependent multidisciplinary problems is challenging
  - Multidisciplinary adjoint solution
  - Dynamic AMR
  - Load balancing