Adjoint-Based Discretization Error Estimation for Time Dependent Problems

Dimitri Mavriplis
and
Karthik Mani

Department of Mechanical Engineering
University of Wyoming
Unsteady Problems

Total error in solution

• Temporal error (discretization/resolution)
  - Flow
  - Mesh
  - Other

• Algebraic error
  - Flow
  - Mesh
  - Other

• Spatial error (discretization/resolution)
  - Flow
  - Mesh
  - Other

• Solution of time-dependent adjoint: backwards integration in time
• Disciplinary adjoint inner product with disciplinary residual
Analysis Problem: 
Flow Equations

Conservative form of Euler equations
\[ \frac{\partial U(x, t)}{\partial t} + \nabla \cdot F(U) = 0 \]

Integrate over a moving control volume
\[ \frac{dA}{dt} U + \int_{B(t)} [F(U) - \dot{x}U] \cdot ndB = 0 \]

- Spatial discretization uses second-order accurate matrix dissipation scheme
- Temporal discretization uses second-order accurate BDF2 scheme

Define flow residual as:
\[ R^n(U^n, U^{n-1}, U^{n-2}, x^n, x^{n-1}, x^{n-2}) = \frac{dA}{dt} U + \int_{B(t)} [F(U) - \dot{x}U] \cdot ndB = 0 \]

Newton solver
\[
\begin{bmatrix}
\frac{\partial R(U^j)}{\partial U^j}
\end{bmatrix} \delta U^j = -R(U^j)
\]
\[ U^{j+1} = U^j + \delta U^j \]
Analysis Problem
Mesh Motion Equations

2 Spring equations (x and y)

Overall linear system: relates interior displacements to boundary displacements

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

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

Solve using Gauss-Seidel or linear multigrid
Analysis Problem
Mesh Motion Equations

Approximate edges as springs

2 Spring equations (x and y)

Overall linear system: relates interior displacements to boundary displacements

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

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

Solve using Gauss-Seidel or linear multigrid
Adjoint-Based Functional Error

Functional Taylor expansion about approximate values: \( U_h^H, x_h^H \)

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

Expand residual equations to obtain expressions for state error vectors:

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

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

Substitute back into functional expansion, define adjoint variables:

\[
\left[ \frac{\partial R}{\partial U} \right]_{U_h^H, x_h^H}^T \Lambda_{U_h} = -\left[ \frac{\partial L}{\partial U} \right]_{U_h^H, x_h^H}^T \\
\left[ K \right]^T \Lambda_{x_h} = -\lambda_{x_h}^T \\
\Lambda_{x_h} = I_h^H \Lambda_{x_h}
\]

Adjoint solution is a backward sweep in time
Adjoint problem solved in approximate space H (preferably)
Error in Functional

\[ L_h(U_h, x_h) - L_h(U_h^H, x_h^H) = -\Lambda_{U_h}^{H^T} R_h(U_h^H, x_h^H) - \Lambda_{x_h}^{H^T} G(x_h^H) \]

- Error in functional is inner product of adjoint and non-zero residual(s)
  - Flow adjoint . Flow residual \( \rightarrow \) Error due to flow equations
  - Mesh adjoint . Mesh residual \( \rightarrow \) Error due to mesh motion equations
- Inner product is over all space and all time
- Provides an error distribution which can be adapted on

- Approximate solutions: \( U_h^H, x_h^H \)
  - Obtained on coarser mesh \( \rightarrow \) Spatial error
  - Obtained using larger time steps \( \rightarrow \) Temporal error
  - Obtained by partially converging implicit system \( \rightarrow \) Algebraic error
Summary of Temporal Resolution Error Evaluation

- Compute unsteady flow solution on coarse time domain

- Compute adjoint variables on coarse time domain
  - Integrating backward in time

- Project adjoint variables, flow solution and mesh solution onto fine time domain

- Temporal resolution error is then dot product of adjoint with corresponding non-zero residual on fine time domain
  - Distribution in time is used to drive adaptation
Verification

*Functional = Lift after 1.5 periods*

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

- Adjoint is linearization about current state (16 time steps) to predict objective value on modified state (32 time steps)
Summary of Temporal/Algebraic Error Evaluation and Decomposition

• Compute partially converged flow and mesh solution on coarse time domain

• Compute adjoint variables on coarse domain using partially converged solution

• Compute partial convergence error on coarse level time domain
  – Inner product of adjoint with partially converged (non-zero) residual

• Project partially converged solution and adjoint variables onto fine time domain

• Evaluate fine level error estimate as previously
  – Combined temporal resolution and partial convergence error

• Determine temporal resolution error by subtracting partial convergence error from total error estimate on fine time domain
## Verification

<table>
<thead>
<tr>
<th>Description</th>
<th>Functional value</th>
<th>% Error vs. Target</th>
<th>% Predicted Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target functional - exact at 16 steps (fully converged)</td>
<td>-0.28576836616489</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Partial mesh convergence at 16 steps with limit = 1e-3 (flow fully converged)</td>
<td>-0.285995701438330</td>
<td>0.0796</td>
<td>-0.0743</td>
</tr>
<tr>
<td>Corrected for partial mesh convergence</td>
<td>-0.285783421090738</td>
<td>-0.0052</td>
<td>-</td>
</tr>
<tr>
<td>Partial flow convergence at 16 steps with limit = 0.8e-4 (fully converged)</td>
<td>-0.289033070451402</td>
<td>-1.142</td>
<td>-1.143</td>
</tr>
<tr>
<td>Corrected for partial flow convergence</td>
<td>-0.285765384379180</td>
<td>0.001</td>
<td>-</td>
</tr>
<tr>
<td>Partial flow and mesh convergence at 16 steps (flow=0.8e-4,mesh=1e-3)</td>
<td>-0.289270768414725</td>
<td>-1.226</td>
<td>-1.223</td>
</tr>
<tr>
<td>Corrected for partial flow and mesh convergence</td>
<td>-0.285778143449756</td>
<td>-0.003</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Description</th>
<th>Functional value</th>
<th>% Error vs. Target</th>
<th>% Predicted Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target functional - exact at 32 steps (fully converged)</td>
<td>-0.309957065250867</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Partially converged flow and mesh at 16 steps (flow = 0.8e-4, mesh = 1e-3)</td>
<td>-0.289270768414725</td>
<td>+6.674</td>
<td>+6.562</td>
</tr>
<tr>
<td>Corrected for partial convergence and also resolution from 16 to 32 steps</td>
<td>-0.3096094927905224</td>
<td>+0.112</td>
<td>-</td>
</tr>
</tbody>
</table>
Adaptation Strategy

**Adaptation strategy:**
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

Targeted temporal adaptation compared against local error-based adaptation:

Local error estimated as:

\[ e_{local} = \left\| \left[ \frac{dA}{dt} \right]_{BDF3} - \left[ \frac{dA}{dt} \right]_{BDF2} \right\|_2 \]
Time-integrated functional

- Interaction of isentropic vortex with slowly pitching NACA0012
- Mach number = 0.4225
- Reduced frequency = 0.001
- Center of pitch is quarter chord
- Functional is \[ L = \sum_{n=1}^{n_{\text{eps}}} \Delta t_n C_{L_n} \]

8,600 elements
Vortex-Airfoil Interaction Test Case
Adaptation Results
Time-Integrated Functional

![Graph showing time-integrated functional with curves for vortex interaction and no vortex.](image)
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
Spatial Variation of Temporal Error

• Adjoint error estimates give distribution of error in space and time for each component (discipline)
• Time step adaptively refined but uniform spatial time step value has been maintained
• Further gains by refining time step only in spatial locations with high temporal error
• *Note: could also devise spatially varying convergence tolerances...*
Traditional Uniform Time-Stepping

Traditional approach to solving unsteady problems (An illustration in 1D space):

Define residual operator $R$ at each time-step:

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

Linearize and solve using Newton iterations:

$$\left[ \frac{\partial R(U^k, x)}{\partial U^k} \right] \delta U^k = -R(U^k, x)$$

$$U^{k+1} = U^k + \delta U^k$$

$$\delta U^k \rightarrow 0, U^{k+1} = U^n$$
Spatially Non-Uniform Time-Stepping

The space-time slab-based unsteady solution process (An illustration in 1D space):

Define residual operator $R$ over whole slab and solve for all $U$ within slab (which are unknown) in one-shot using Newton iterations.
Formulation

Treat space and time together in integral form.

Start with conservative form of Euler equations:

\[ \frac{\partial U(x,t)}{\partial t} + \nabla \cdot F(U) = 0 \]

Integrate over moving control volume to get ALE form:

\[ \frac{dA}{dt} U + \int_{B(t)} [F(U) - \dot{x}U] \cdot ndB = 0 \]

Integrate once more, now over time to get Finite-Volume Space-Time form:

\[ \int_{T} \frac{dA}{dt} U dt + \int_{T} \int_{B(t)} [F(U) - \dot{x}U] \cdot ndBdt = 0 \]
Space-Time (S-T) Element

The space-time element is the control volume for the space-time FV formulation:

Constructed using 2 spatial elements at different times
(2D triangle at different times shown)

Space-time (S-T) elements bounded by temporal and space-time faces.
3D spatial elements used to construct space-time elements cannot be viewed easily.
Temporal Discretization

Just as in traditional Finite-Volume method, assume a constant solution within space-time element

\[ \int_T \frac{dA}{d} \frac{dU}{dt} dt = \frac{dA}{d} \frac{dU}{\Delta t} \]

First-order accurate backward-difference-formula (BDF1):

\[ \frac{dA}{d} \frac{dU}{\Delta t} = A^n U^n - A^{n-1} U^{n-1} \]

- Essentially amounts to a fully-upwinded flux balance in the time direction for the space-time element.
- BDF2 follows by extending temporal stencil one extra time level
Spatial Discretization

Spatial flux components are based on central differencing with artificial dissipation:

\[ S = \int_{T} \int_{dB(t)} [F(U) - \dot{x}U] \cdot ndBdt = \sum_{i=1}^{n_{edge}} F_{e_i}^{\perp}(V_{ei}, U, n_{ei})B_{ei}\Delta t_{ei} \]

\[ F_{e_i}^{\perp} = \frac{1}{2} \left\{ F_{L}^{\perp}(U_{L}, V_{e}, n_{e}) + F_{R}^{\perp}(U_{R}, V_{e}, n_{e}) + \kappa^{(4)}[T][\lambda][T]^{-1} \left\{ (\nabla^2 U)_L - (\nabla^2 U)_R \right\} \right\} \]

Undivided Laplacian for 2nd-order spatial accuracy:

\[ (\nabla^2 U)_i = \sum_{k=1}^{\text{spatial neighbors}} (U_k - U_i) \]

\( \dot{\lambda} \) (Grid speed terms) determined from grid point positions (mesh motion equations) and ensuring conservation in space-time-element (Geometric Conservation Law)
Convergence

Convergence comparisons flow equations:

Strong implicitness of ILU(0) leads to an almost mesh independent solver
Vortex-Airfoil Interaction Test Case
Vortex-Airfoil Interaction Test Case
Vortex-Airfoil Interaction Test Case
Functional and Error Convergence Comparison

Graphs showing the comparison of different convergence scenarios including Uniform, Adapted S-uniform, Corrected S-uniform, Adapted ST non-uniform, Corrected ST non-uniform, and Exact. The graphs illustrate the variation of functional and error with degrees of freedom (DOF) on a logarithmic scale.
Adjoint Driven Spatial Error Reduction

- 3 levels of refinement, targeting 75% of error at each level
  (objective=lift on airfoil) (Bryan Flynt, UWyoming PhD)
Conclusions and future work

• Adjoint error estimates
  – Target specific objectives of interest
  – Decompose error into disciplinary components
  – Decompose error into spatial, temporal, algebraic
  – Space time formulation enables further gains through spatially varying/optimal time steps
  – Extension to combined temporal-algebraic-spatial error (dynamic AMR)
Variation of lift coefficient in time