Introduction

Discretization Error (DE)

- **What is DE?**
  - Numerical error that arises from the discretization of the governing equations
  - Typically the largest numerical error source and most difficult to estimate

- **Why is DE estimation important?**
  - Discretization errors can adversely affect engineering decision-making leading to poor system performance, system failure, or risk to public safety
  - Example: Even small errors in drag prediction can have a significant impact on aircraft performance [1]

Goals

- To obtain accurate DE estimates for the entire field solution and solution functionals at a low cost
  - Accuracy → Increase confidence in simulation results
  - Computational efficiency → More adoptable by broader CFD community
1 Introduction

2 Background

3 Methodology

4 Results

5 Conclusions & Future Work
Outline

1. Introduction
2. Background
3. Methodology
4. Results
5. Conclusions & Future Work
Preliminaries

Continuous Equations

\[ N(\tilde{u}) := |\Omega_i| \frac{d l_h^{\tilde{u}}}{dt} + \int_{\partial \Omega_i} F(\tilde{u}) \cdot \hat{n} \, dS - \int_{\Omega_i} s(\tilde{u}) \, d\Omega = 0 \]

Discrete Equations

\[ N^p_h(u_h) := |\Omega_i| \frac{d u_h}{dt} + \int_{\partial \Omega_i} F_h(l^+_h u_h, l^-_h u_h) \cdot \hat{n} \, dS - \int_{\Omega_i} s(l^+_h u_h) \, d\Omega = 0 \]

Truncation Error

\[ \tau^p_h(u) = N^p_h(l^h u) - l^h N(u) \]

Discretization Error

\[ \varepsilon_h = u_h - l^h \tilde{u} \]
Preliminaries

Continuous Equations

\[
N(\tilde{u}) := \left| \Omega_i \right| \frac{dI^h \tilde{u}}{dt} + \int_{\partial \Omega_i} F(\tilde{u}) \cdot \hat{n} \, dS - \int_{\Omega_i} s(\tilde{u}) \, d\Omega = 0
\]

Discrete Equations

\[
N^p_h(u_h) := \left| \Omega_i \right| \frac{dI^h u_h}{dt} + \int_{\partial \Omega_i} F_h(I^h u_h^+, I^h u_h^-) \cdot \hat{n} \, dS - \int_{\Omega_i} s(I^h u_h) \, d\Omega = 0
\]

Truncation Error

\[
\tau^p_h(u) = N^p_h(I^h u) - I^h N(u)
\]

Discretization Error

\[
\varepsilon_h = u_h - I^h \tilde{u}
\]
Preliminaries

Continuous Equations

\[ N(\tilde{u}) := |\Omega_i| \frac{d l^h \tilde{u}}{dt} + \int_{\partial \Omega_i} F(\tilde{u}) \cdot \hat{n} \, dS - \int_{\Omega_i} s(\tilde{u}) \, d\Omega = 0 \]

Discrete Equations

\[ N_h^p(u_h) := |\Omega_i| \frac{d u_h}{dt} + \int_{\partial \Omega_i} F_h(I_h u_h^+, I_h u_h^-) \cdot \hat{n} \, dS - \int_{\Omega_i} s(I_h u_h) \, d\Omega = 0 \]

Truncation Error

\[ \tau_h^p(u) = N_h^p(I^h u) - I^h N(u) \]

Discretization Error

\[ \varepsilon_h = u_h - I^h \tilde{u} \]
Preliminaries

**Continuous Equations**

\[
N(\tilde{u}) := |\Omega_i| \frac{dI^h\tilde{u}}{dt} + \int_{\partial\Omega_i} \mathbf{F}(\tilde{u}) \cdot \mathbf{n} \, dS - \int_{\Omega_i} \mathbf{s}(\tilde{u}) \, d\Omega = 0
\]

**Discrete Equations**

\[
N^p_h(u_h) := |\Omega_i| \frac{du_h}{dt} + \int_{\partial\Omega_i} \mathbf{F}_h(I^h_p u^+_h, I^h_p u^-_h) \cdot \mathbf{n} \, dS - \int_{\Omega_i} \mathbf{s}(I^h_p u_h) \, d\Omega = 0
\]

**Truncation Error**

\[
\tau^p_h(u) = N^p_h(I^h u) - I^h N(u)
\]

**Discretization Error**

\[
\varepsilon_h = u_h - I^h \tilde{u}
\]
Discretization Error Estimation

**Linearized Error Transport Equations (ETE)**

- Performing a Newton linearization of the discrete operator about $u_h$:

\[
N_h^p(I^h\tilde{u}) = N_h^p(u_h) - \frac{\partial N_h^p}{\partial u_h} (u_h - I^h\tilde{u}) + O\left(\|\varepsilon_h\|^2\right)
\]

\[
\Rightarrow \tau_h^p(\bar{u}) = 0 = \varepsilon_h
\]

- Rearranging and neglecting H.O.T.:

\[
\frac{\partial N_h^p}{\partial u_h} \varepsilon_h = -\tau_h^p(\bar{u})
\]

**Advantages of Linearized ETE**

- Minimal code modifications required for implementation
- Computationally cheaper than nonlinear ETE $\rightarrow$ 1 linear solve vs. nonlinear solve
- Like adjoints, provide error estimates in solution functionals
- Unlike adjoints, also provide local error estimates in all field variables
Outline

1. Introduction
2. Background
3. Methodology
4. Results
5. Conclusions & Future Work
Continuous Residual / p-TE Estimation

- Operate the continuous equations (or higher-order discrete equations) on a reconstruction of the discrete solution
  \[
  \tau_h^p(\tilde{\mathbf{u}}) \approx \tau_h^p(\mathbf{l}_h^r \mathbf{u}_h) = \mathbf{N}_h^p(\mathbf{l}_h^r \mathbf{u}_h) - \mathbf{l}_h^h \mathbf{N}(\mathbf{l}_h^r \mathbf{u}_h) \\
  = -\mathbf{l}_h^h \mathbf{N}(\mathbf{l}_h^r \mathbf{u}_h) \\
  \approx -\mathbf{N}_h^r(\mathbf{u}_h)
  \]

- Assumptions:
  - Reconstruction is representative of the exact solution: \( \mathbf{l}_h^r \mathbf{u}_h \approx \tilde{\mathbf{u}} \)
  - Reconstruction preserves the control volume average: \( \mathbf{u}_h = \mathbf{l}_h^r \mathbf{l}_h^r \mathbf{u}_h \)
  - Primal solution is iteratively converged: \( \mathbf{N}_h^p(\mathbf{u}_h) = 0 \)
  - Inter-cell jumps in the reconstruction are small: \( ||\mathbf{l}_h^r \mathbf{u}_h^+ - \mathbf{l}_h^r \mathbf{u}_h^-|| = O(h^r) \)

- Solution Reconstruction: \( k \)-Exact Least-Squares
  - Exactly reconstructs polynomials of degree \( \leq k \), i.e., \( ||\mathbf{u}_i^R(x) - u(x)|| = O(h^{k+1}) \)
  - Formulation: Expand \( u(x) \) in a Taylor series and enforce conservation of the mean in \( \Omega_i \) (exactly) and in a patch of surrounding cells \( \Omega_j \) (least-squares), e.g.
    \[
    \frac{1}{|\Omega_i|} \int_{\Omega_i} \mathbf{u}_i^R(x - x_i) \, d\Omega = u|_i + \frac{\partial u}{\partial x}|_i \overline{x}_i + \frac{\partial^2 u}{\partial x^2}|_i \overline{x}_i^2 + \cdots = \mathbf{u}_h, i
    \]
    \[
    \therefore \text{Reconstruction preserves control-volume averages: } \mathbf{l}_h^r \mathbf{l}_h^r \mathbf{u}_h = \mathbf{u}_h
    \]
Higher-Order Error Estimation

**ETE Relinearization**

**Goal:** To improve the accuracy of DE estimates (or equivalently the discrete solution)

**Approach:** Apply ETE correction iteratively

---

**Algorithm 1** ETE Relinearization

1. Compute: \( N_h^P(u_{h,0}) = 0 \)
2. for \( i = 1 \) to \( N_{relin} \) do
3. Compute: \( \tau_h^P(\bar{u}) \approx \tau_h^P(I_h'u_{h,i-1}) = -I_h^T N(I_h'u_{h,i-1}) \)
4. Compute: \( \frac{\partial N_h^P}{\partial u_h} \bar{e}_h = -\tau_h^P(I_h'u_{h,i-1}) \)
5. Compute: \( u_{h,i} = u_{h,i-1} - \bar{e}_h \)
6. end for
7. Compute: \( \bar{e}_h = u_{h,0} - u_h, N_{relin} \)
8. for \( j = 1 \) to \( N_{func} \) do
9. Compute: \( \varepsilon_j = \frac{\partial J_h,j}{\partial u_h} \bar{e}_h \)
10. end for

---

**Benefits**

- Can achieve error estimates (corrected solutions) of arbitrary order of accuracy

---

Outline

1. Introduction
2. Background
3. Methodology
4. Results
5. Conclusions & Future Work
Burgers’ Equation

2D Viscous Burgers’ Equation

\[
\frac{\partial u}{\partial t} + \frac{\partial}{\partial x} \left( \frac{u^2}{2} \right) + \frac{\partial}{\partial y} \left( \frac{u^2}{2} \right) = \nu \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right)
\]

Case Description

- **Computational Domain:**
  \[ x, y \in [-4.0m, +4.0m] \]

- **Description:**
  - Model problem for the coalescence of viscous shock waves

- **Boundary Conditions:**
  - Prescribe outer state + Nitsche’s Method

- **Primal Discretization:**
  - 2\textsuperscript{nd} order \( k \)-exact
Problem: 2D Burgers' Equation
Order Triplet: (p,q,r) = (2,0,0)
Variable: u
Grid Level: 256x256 (cells)

Base DE

Corrected DE: Pressure ($r = 4$)

DE Units: m/s
ETE Relinearization

Problem: 2D Burgers’ Equation
Order Triplet: (p,q,r) = (2,0,0)
Variable: u
Grid Level: 256x256 (cells)
DE Level: Base

Problem: 2D Burgers’ Equation
Order Triplet: (p,q,r) = (2,2,6)
Variable: u
Grid Level: 256x256 (cells)
DE Level: Correction #05

Corrected DE: Pressure ($r = 6$)

DE Units: m/s
2D Viscous Burgers’ Equation

ETE Relinearization

Problem: 2D Burgers’ Equation
Order Triplet: $(p,q,r) = (2,0,0)$
Variable: $u$
Grid Level: 256x256 (cells)
DE Level: Base

Problem: 2D Burgers’ Equation
Order Triplet: $(p,q,r) = (2,2,8)$
Variable: $u$
Grid Level: 256x256 (cells)
DE Level: Correction #08

Base DE

Corrected DE: Pressure $(r = 8)$

DE Units: m/s

W. Tyson (wtyson45@vt.edu)
2D Viscous Burgers’ Equation

ETE Relinearization

DE Convergence

Runtime Comparison w/ HO Solver

DE Units: m/s

W. Tyson (wtyson45@vt.edu)
Gaussian Bump Manufactured Solution

Euler Equations

\[ \frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{u}) = \mathbf{s}(\mathbf{u}) \]

\[ \mathbf{u} = \begin{bmatrix} \rho \\ \rho \mathbf{v} \\ \rho e_t \end{bmatrix}, \quad \mathbf{F}(\mathbf{u}) = \begin{bmatrix} \rho \mathbf{v}^T \\ \rho \mathbf{v} \mathbf{v}^T + \rho \mathbf{l} \\ \rho h_t \mathbf{v}^T \end{bmatrix}, \quad \mathbf{s}(\mathbf{u}) = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \]

Case Description

- Computational Domain:
  \( (x, y) \in [0.0m, 1.0m] \)

- Manufactured Solution:
  \[ \phi(x) = \alpha_0 + \alpha_1 \exp \left\{ - \left[ a_1(x - x_0)^2 + 2b_1(x - x_0)(y - y_0) + c_1(y - y_0)^2 \right] \right\} \]

- Boundary Conditions:
  - Prescribe exact outer state
ETE Relinearization

Problem: Gaussian Bump
Order Triplet: \((p, q, r) = (2, 0, 0)\)
Variable: Pressure
Grid Level: 256x256 (cells)
DE Level: Base

Problem: Gaussian Bump
Order Triplet: \((p, q, r) = (2, 2, 4)\)
Variable: Pressure
Grid Level: 256x256 (cells)
DE Level: Correction #06

Base DE: Pressure
Corrected DE: Pressure \((r = 4)\)

DE Units: Pa
Gaussian Bump Manufactured Solution

ETE Relinearization

Problem: Gaussian Bump
Order Triplet: (p,q,r) = (2,0,0)
Variable: Pressure
Grid Level: 256x256 (cells)
DE Level: Base

Problem: Gaussian Bump
Order Triplet: (p,q,r) = (2,2,6)
Variable: Pressure
Grid Level: 256x256 (cells)
DE Level: Correction #12

Base DE: Pressure
Corrected DE: Pressure (r = 6)

DE Units: Pa
Gaussian Bump Manufactured Solution

ETE Relinearization

Problem: 2D Gaussian Bump
Variable: Pressure
Grid Levels: 8x8 - 512x512 (cells)
Norm: $L_2$

Global Discretization Error Norm

$\frac{1}{10^3} - 10^0$

Base DE: (2,0,0)
Corrected DE: (2,0,0) + (2,2,4) (Step #06)
Corrected DE: (2,0,0) + (2,2,6) (Step #12)

$2^{nd}$ Order Slope
$4^{th}$ Order Slope
$5^{th}$ Order Slope

DE Convergence

Runtime Comparison w/ HO Solver

DE Units: Pa

W. Tyson (wtyson45@vt.edu)  ASME VVS2019  May 15, 2019 14 / 21
**Gaussian Bump Manufactured Solution**

**ETE Relinearization: Perturbed Grids**

**DE Convergence**

**Problem:** 2D Gaussian Bump  
**Variable:** Pressure  
**Grid Levels:** 8x8 - 512x512 (cells)  
**Norm:** $L_2$

\[ h = (N_{\text{cells}})^{-1/d} \]

**Global Discretization Error Norm**

- Base DE: (2,0,0)  
- Corrected DE: (2,0,0) + (2,2,4) (Step: #01 - #08)  
- Corrected DE: (2,0,0) + (2,2,4) (Step: #09)  
- Corrected DE: (2,0,0) + (2,2,6) (Step: #10 - #17)  
- Corrected DE: (2,0,0) + (2,2,6) (Step: #18)

**2nd Order Slope**  
**4th Order Slope**  
**5th Order Slope**

**Perturbed Grid (17 × 17)**

**DE Units:** Pa

---

W. Tyson (wtyson45@vt.edu)  
ASME VVS2019  
May 15, 2019
Karman-Trefftz Airfoil

Euler Equations

\[ \frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{u}) = \mathbf{s}(\mathbf{u}) \]

\[ \mathbf{u} = \begin{bmatrix} \rho \\ \rho \mathbf{v} \\ \rho e_t \end{bmatrix}, \quad \mathbf{F}(\mathbf{u}) = \begin{bmatrix} \rho \mathbf{v}^T \\ \rho \mathbf{v} \mathbf{v}^T + \rho \mathbf{l} \\ \rho h_t \mathbf{v}^T \end{bmatrix}, \quad \mathbf{s}(\mathbf{u}) = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \]

Case Description

- **Mapping:** Karman-Trefftz
- **Cylinder Center:** \((-0.1 + 0.0i)\)
- **TE angle:** \(\approx 5.7^\circ\)
- **Farfield Conditions:**
  \[ \alpha = 2^\circ, \quad M_\infty = 0.2 \]
  \[ \rho_\infty = 1\times10^5 \text{ Pa}, \quad T_\infty = 300 \text{ K} \]
- **Implementation Notes:**
  - Treated as manufactured solution
  - Results qualitative since curved surface geometry not available

W. Tyson (wtyson45@vt.edu)

ASME VVS2019

May 15, 2019 15 / 21
Problem: Karman-Trefftz Airfoil
Order Triplet: (p,q,r) = (1,0,0)
Variable: Pressure
Grid Level: 160x64 (cells)
DE: Exact

DE Units: Pa

W. Tyson (wtyson45@vt.edu)
Problem: Karman-Trefftz Airfoil
Order Triplet: (p,q,r) = (1,0,0)
Variable: Pressure
Grid Level: 160x64 (cells)
DE: Exact

Problem: Karman-Trefftz Airfoil
Order Triplet: (p,q,r) = (1,1,2)
Variable: Pressure
Grid Level: 160x64 (cells)
DE: ETE

DE Units: Pa
Karman-Trefftz Airfoil

Richardson Extrapolation vs. ETE

Problem: Karman-Trefftz Airfoil
Order Triplet: (p,q,r) = (1,0,0)
Variable: Pressure
Grid Level: 160x64 (cells)
DE: Exact

Problem: Karman-Trefftz Airfoil
Order Triplet: (p,q,r) = (1,1,2)
Variable: Pressure
Grid Level: 160x64 (cells)
DE: ETE (Correction #12)

Relinearized ETE Estimate

DE Units: Pa

W. Tyson (wtyson45@vt.edu)

ASME VVS2019

May 15, 2019
Problem: Karman-Trefftz Airfoil
Variable: Pressure
Grid Level: 160x64 (cells)
DE: Richardson Extrapolation

Error in Estimate: Richardson Extrapolation

DE Units: Pa

W. Tyson (wtyson45@vt.edu)

ASME VVS2019
May 15, 2019
Outline

1. Introduction
2. Background
3. Methodology
4. Results
5. Conclusions & Future Work
Conclusions & Future Work

Conclusions

- Developed a new methodology based on the linearized ETE which can iteratively compute higher-order error estimates
- Demonstrated methodology for several inviscid and viscous flow problems
- Obtained error estimates which were as high as 8\textsuperscript{th} order accurate
- Compared methodology to standard linearized ETE and Richardson extrapolation

Future Work

- Investigate accuracy differences between linearized ETE and nonlinear ETE
- Extend methodology to unsteady flow problems
- Add curved surface geometry capability
Acknowledgements

Special Thanks

- Dr. John Benek (AFRL)
- Dr. Michael Park (NASA LaRC)
- Dr. Joseph Derlaga (NASA LaRC)
Questions?


Backup Slides
Higher-Order Discretization Error Estimates

- **Goal:** Improve accuracy of DE estimates → Reduce numerical uncertainty
- **Approach:** Use adjoint/ETE equivalence to extend HO properties of adjoint to ETE
  - Select $u_h$ in $\Omega_i$ as the functional of interest and insert ETE error estimate:
    \[
    \varepsilon J_h = J_h(u_h) - J_h(I^h \tilde{u}) = \frac{\partial J_h}{\partial u_h} \varepsilon_h + O\left(\|\varepsilon_h\|^2\right)
    \]
    \[
    = u_{h,i} - (I^h \tilde{u})_i \approx \delta_{ij} \bar{\varepsilon}_{h,j} + O\left(\|\bar{\varepsilon}_h\|^2\right)
    \]
  - Assuming $\|\bar{\varepsilon}_h\| = O(h^p)$, then $\bar{\varepsilon}_h$ converges to $\varepsilon_h$ at a HO rate, i.e.,
    \[
    \|\varepsilon_h - \bar{\varepsilon}_h\| \approx O(h^{2p})
    \]
  
  Note: Requires $u_h$ to be sufficiently smooth and accurate TE estimate

Advantages of Higher-Order Error Estimates

- Generally more accurate in an asymptotic sense than LO error estimates
- Can correct the entire primal solution and all output functionals to HO
- Computationally cheaper than a conventional HO method


Quasi-1D Nozzle

Quasi-1D Euler Equations

\[
\frac{\partial (Au)}{\partial t} + \frac{\partial AF(u)}{\partial x} = s(u)
\]

\[
u = \begin{bmatrix}
\rho \\
\rho u \\
\rho e_t
\end{bmatrix}, \quad F(u) = \begin{bmatrix}
\rho u \\
\rho u^2 + p \\
\rho uh_t
\end{bmatrix}, \quad s(u) = \begin{bmatrix}
0 \\
p \frac{dA}{dx} \\
0
\end{bmatrix}
\]

Case Description

- Computational Domain:
  \[x \in [-1.0 \text{m}, +1.0 \text{m}]\]
- Gaussian area distribution:
  \[A(x) = 1 - 0.8 \exp(-12.5x^2)\]
- Inflow Conditions:
  \[p_0 = 300 \text{ kPa}, \quad T_0 = 600 \text{ K}\]
- Outflow Conditions:
  \[p_{\text{back}} = 297.485 \text{ kPa}\]
Quasi-1D Nozzle

ETE Relinearization

Global Discretization Error Norm

Problem: Quasi-1D Nozzle
Primal Problem: Sub.-Sup.
Variable: Pressure
Grid Levels: 32 - 2048 (cells)
Grid Spacing: Uniform

DE Convergence

Runtime Comparison w/ HO Solver

CPU Time (sec)

Global Discretization Error Norm

Problem: Quasi-1D Nozzle
Primal Problem: Sub.-Sup.
Variable: Pressure
Grid Levels: 32 - 2048 (cells)
Grid Spacing: Uniform
Nonlinear Error Transport (NETE)

Continuous NETE:
\[
\hat{N}(u, \varepsilon) = -\frac{\partial}{\partial t} \int_{\Omega_i} \varepsilon \, d\Omega + R(u - \varepsilon) - R(u) = -\left[ \frac{\partial}{\partial t} \int_{\Omega_i} u \, d\Omega + R(u) \right]
\]

Discrete NETE:
\[
\hat{N}_h(u_h, \bar{\varepsilon}_h) = -|\Omega_i| \frac{\partial \bar{\varepsilon}_h}{\partial t} + R_h^q(u_h - \bar{\varepsilon}_h) - R_h^q(u_h) = -\left[ |\Omega_i| \frac{\partial u_h}{\partial t} + R_h^r(u_h) \right]
\]

Linearized Error Transport (LET)

Continuous LETE:
\[
\hat{L}(u, \varepsilon) = \frac{\partial}{\partial t} \int_{\Omega_i} \varepsilon \, d\Omega + R(\varepsilon) = \frac{\partial}{\partial t} \int_{\Omega_i} u \, d\Omega + R(u)
\]

Discrete LETE:
\[
\hat{L}_h(u_h, \bar{\varepsilon}_h) = |\Omega_i| \frac{\partial \bar{\varepsilon}_h}{\partial t} + \frac{\partial R_h^q}{\partial u_h} \bar{\varepsilon}_h = |\Omega_i| \frac{\partial u_h}{\partial t} + R_h^r(u_h)
\]
Expected Order of Accuracy (OOA)

Nonlinear Error Transport (NETE)

\[ \| \bar{\varepsilon}_h - \varepsilon_h \| = O \left( h^{\min(p+q,r)} \right) \]

Linearized Error Transport (LETE)

\[ \| \bar{\varepsilon}_h - \varepsilon_h \| = O \left( h^{\min(2p,p+q,r)} \right) \]

Comments

- \( p \): Order of primal problem
- \( q \): Order of ETE discretization
- \( r \): Order of residual estimate

Note: Expected OOA should hold for structured and unstructured grids.
HO error estimates require \( r > p \)
ETE Relinearization: “Unstructured” Grids

- For steady-state problems, ETE relinearization is equivalent to:
  - Solving higher-order problem with a Newton method for set number of iterations
  - Defining the error estimate as the difference between the lower-order solution and the corrected solution
- Higher-order error estimates obtained after a sufficient number of iterations

![Graph showing DE convergence: Pressure](image)
ETE Relinearization: Perturbed Grids

Problem: Gaussian Bump
Order Triplet: \((p,q,r) = (2,0,0)\)
Variable: Pressure
Grid Level: 256x256 (cells)
DE Level: Base

Problem: Gaussian Bump
Order Triplet: \((p,q,r) = (2,2,4)\)
Variable: Pressure
Grid Level: 256x256 (cells)
DE Level: Correction #09

Base DE: Pressure
Corrected DE: Pressure \((r = 4)\)

DE Units: Pa

W. Tyson (wtyson45@vt.edu)
Gaussian Bump Manufactured Solution

ETE Relinearization: Perturbed Grids

Problem: Gaussian Bump
Order Triplet: (p,q,r) = (2,0,0)
Variable: Pressure
Grid Level: 256x256 (cells)
DE Level: Base

Problem: Gaussian Bump
Order Triplet: (p,q,r) = (2,2,6)
Variable: Pressure
Grid Level: 256x256 (cells)
DE Level: Correction #18

Base DE: Pressure
Corrected DE: Pressure (r = 6)

DE Units: Pa
Functional Error Estimation

Discrete Adjoint Method

- Performing a Newton linearization of a discrete solution functional, $J_h(\cdot)$, about $u_h$:
  \[ J_h(I^h \tilde{u}) = J_h(u_h) - \frac{\partial J_h}{\partial u_h} \varepsilon_h + O(\|\varepsilon_h\|^2) \]

- Rearranging and defining the functional error:
  \[ \varepsilon_{J_h} := J_h(u_h) - J_h(I^h \tilde{u}) = -\lambda^T_h \tau^p_h(\tilde{u}) + O(\|\varepsilon_h\|^2) \]

where
\[
\begin{bmatrix}
\frac{\partial N^p_h}{\partial u_h} \\
\frac{\partial J_h}{\partial u_h}
\end{bmatrix}^T \lambda_h = \begin{bmatrix}
\frac{\partial J_h}{\partial u_h}
\end{bmatrix}^T
\]

Advantages of Discrete Adjoint Method

- Provides error estimates & adaptation indicators for any functional of interest
- Application of error estimate as a correction increases the convergence order
- Simpler to implement than the continuous adjoint

Disadvantages of Discrete Adjoint Method

- Requires exact + dual-consistent linearization
- Requires an adjoint solve for each functional of interest
Functional Error Estimation

Discrete Adjoint Method

- Performing a Newton linearization of a discrete solution functional, $J_h(\cdot)$, about $u_h$:
  \[ J_h(I^h\tilde{u}) = J_h(u_h) - \frac{\partial J_h}{\partial u_h} \varepsilon_h + O(\|\varepsilon_h\|^2) \]

- Rearranging and defining the functional error:
  \[ \varepsilon_{J_h} := J_h(u_h) - J_h(I^h\tilde{u}) = -\lambda_h^T \tau_p^h(\tilde{u}) + O(\|\varepsilon_h\|^2) \]

  where
  \[ \left[ \frac{\partial N_p}{\partial u_h} \right]^T \lambda_h = \left[ \frac{\partial J_h}{\partial u_h} \right]^T \]

Advantages of Discrete Adjoint Method

- Provides error estimates & adaptation indicators for any functional of interest
- Application of error estimate as a correction increases the convergence order
- Simpler to implement than the continuous adjoint

Disadvantages of Discrete Adjoint Method

- Requires exact + dual-consistent linearization
- Requires an adjoint solve for each functional of interest
Functional Error Estimation

Discrete Adjoint Method

- Performing a Newton linearization of a discrete solution functional, $J_h(\cdot)$, about $u_h$:

$$J_h(I^h\tilde{u}) = J_h(u_h) - \frac{\partial J_h}{\partial u_h} \varepsilon_h + O(\|\varepsilon_h\|^2)$$

- Rearranging and defining the functional error:

$$\varepsilon_{J_h} := J_h(u_h) - J_h(I^h\tilde{u}) = -\lambda_h^T \tau_p^h(\tilde{u}) + O(\|\varepsilon_h\|^2)$$

where

$$\begin{bmatrix} \frac{\partial N_p^h}{\partial u_h} \end{bmatrix}^T \lambda_h = \begin{bmatrix} \frac{\partial J_h}{\partial u_h} \end{bmatrix}^T$$

Advantages of Discrete Adjoint Method

- Provides error estimates & adaptation indicators for any functional of interest
- Application of error estimate as a correction increases the convergence order
- Simpler to implement than the continuous adjoint

Disadvantages of Discrete Adjoint Method

- Requires exact + dual-consistent linearization
- Requires an adjoint solve for each functional of interest