Solution Verification Using the Robust Multi-Regression Approach

V. Gregory Weirs
Verification is about numerical error

Code Verification
- Goal: software quality and algorithmic improvement
- Have exact solution, so can compute exact error
- Hard estimates of convergence properties
- Metrics defined by numerical analysis

Solution Verification
- Goal: Estimate numerical error for problems with unknown solutions ("real" problems)
- Soft estimates of numerical error
- Metrics defined by analyst
- Also called “calculation verification”
Motivation

This work is about bridging the gap between code verification and solution verification:

- Provide appropriate confidence in solution verification results (error estimates)
- Better understanding of the weak points in the solution verification process

How do we use what we know from code verification to improve and inform solution verification?

This is work in progress so results are more suggestive than definitive.
Field variable solution verification

1. approximate values of field variables on suitably refined meshes
2. Spatial interpolation or “sampling” to a common mesh
3. At each node, extrapolated solution at h=0, or an error estimate on the fine mesh
4. Approximate global error norm

- Caveat: code verification does not make strong statements on the convergence of point values
Last year’s talk: no surprises

- 2-mesh Richardson Extrapolation behaves well and gives good estimates for (easy) smooth MMS and oblique shock test problems.
- Target mesh tradeoff: interpolate finer solutions to coarse mesh for more accurate error estimate (but with less spatial detail) – “completed” Richardson extrapolation
- Global norms of error estimates converge at design rates, even when less refined solutions are prolonged to fine meshes
Isentropic Vortex test problem

- Air at 300K, 1atm, and $M_x=0.5$; vortex rotates CCW, $M_{r,max} \approx 0.4$. Based on Yee, Sandham, Djhomeri, JCP 150.

- Quadrilateral meshes: regular refinement, coarse mesh nodes are collocated with fine mesh nodes.
  - Mesh 5: 64000 elements, 64551 nodes
  - Mesh 4: 16000 elements, 16281 nodes
  - Mesh 3: 4000 elements, 4141 nodes
  - Mesh 2: 1000 elements, 1071 nodes

- Why it is useful for exploring numerical error:
  - smooth, well behaved
  - straightforward BCs
  - exact solution
Richardson extrapolation, 2 meshes

- We have approximate solution values $\hat{u}_1, ..., \hat{u}_N$ on each node of a common mesh.
- Use two-mesh Richardson extrapolation to generate an error estimate:
  \[
  \hat{u} = \hat{u}_3 + a h_3^p \\
  \hat{u} = \hat{u}_4 + a h_4^p
  \]
- Use the expected convergence rate of $p=2$, and solve for the coefficient, $a$, and extrapolated solution, $\hat{u}$.
- Error estimates are then
  \[
  \hat{e}_3 = \hat{u} - \hat{u}_3 \\
  \hat{e}_4 = \hat{u} - \hat{u}_4
  \]
2-Mesh Rex solutions are well behaved

Specify the convergence rate, $p$ ($p=2$ in this case)

Extrapolated solution is well behaved for this smooth problem

Density field of isentropic vortex on quad meshes 3, 4 interpolated to mesh 3
2-Mesh Rex error estimates are reasonable

Exact and estimated errors are comparable

Estimated error field is smooth for this smooth problem

Density errors of isentropic vortex on quad meshes 3, 4 interpolated to mesh 3
Richardson extrapolation, 3 meshes

- Given computed solutions at three resolutions,
  \[ \ddot{u} = \ddot{u}_2 + a h_2^p \]
  \[ \ddot{u} = \ddot{u}_3 + a h_3^p \]
  \[ \ddot{u} = \ddot{u}_4 + a h_4^p \]

- In general, need to iterate to solve for \( p \), then compute \( \ddot{u} \) and \( a \)

- For a constant refinement ratio:
  \[ p = \log\left( \frac{(\ddot{u}_3 - \ddot{u}_2)}{(\ddot{u}_4 - \ddot{u}_3)} \right) / \log(h_3/h_4) \]
  - But need to handle \( \Delta \ddot{u} = 0 \)
  - Here, \( p = \log\left( \frac{|\ddot{u}_3 - \ddot{u}_2|}{\max(|\ddot{u}_4 - \ddot{u}_3|, 1e-20)} \right) / \log(h_3/h_4) \)
  - And, \( p = 0 \) if \( \ddot{u}_3 - \ddot{u}_2 = 0 \) and \( \ddot{u}_4 - \ddot{u}_3 = 0 \)
3-Mesh Rex solutions are not well behaved at some points

Same scale as for Rex2: [0.93, 1.18]
Black contours outside this range: 1.18, 1.23, 1.28, 1.33, 1.38

Density field of isentropic vortex on quad meshes 3, 4, and 5 interpolated to mesh 3
Ad hoc treatment is not the main cause of poor extrapolated solutions

\[ p = [-11.36, 12.93] \]

Black contours: as before, poor behavior of extrapolated solution

Red contour: \( \mathbf{u}_5 - \mathbf{u}_4 = 0 \)

Blue contour: \( \mathbf{u}_4 - \mathbf{u}_3 = 0 \)

Density convergence rate of isentropic vortex on quad meshes 3, 4, and 5 interpolated to mesh 3
Extrapolated solution can be reasonable even when the rate is not 13.

The design convergence rate is observed for very little of the flowfield, but the extrapolated solution is not nearly as bad as the convergence rate suggests.

Density convergence rate and extrapolated solution of isentropic vortex on quad meshes 3, 4, and 5 interpolated to mesh 3.
Getting past 3-mesh Rex

- Rex on 3 meshes is sensitive:
  - solving for p introduces problems
  - free stream problem
- One strategy: more data (computed solutions) to fit, via least squares; and more error models (Eça and Hoekstra, JCP 262)
- A similar strategy: Robust Multi-regression (Rider, et al., JCP 307)
  - Multiple error models: partial bootstrapping, multiple norms, specified bounding convergence rates
  - Optimization to solve multiple regression problems
  - Robust (median) statistics to diminish role of outliers
  - Include expert judgement (bounding rates, screening results)
Robust Multi-Regression (RMR)

Multiple error models based on \( \bar{u} = \hat{u}_i + a h_i^p \):

- Unspecified rate; theoretical rate; upper and lower bound rates
- Bounded optimization to fit using several norms (here: \( L_1, L_2, L_{\infty} \))
- Partial bootstrapping leaves out coarser meshes
- In this work, for 3 computed solutions (at 3 resolutions):
  - Two finest solutions: 6 models (\( L_2 - p_{th}, L_2 - p_L, L_2 - p_U, L_1, L_2, L_{\infty} \))
  - All three solutions: 6 models
- For 4 computed solutions, 18 models
Robust Multi-Regression, cont’d

- Each model is a separate data fit (unique values of parameters); use robust statistics to choose best fit and describe uncertainty
  - Each model provides a value for ũ; choose the median ũ as the extrapolated solution
  - Here: pick the median of 12 values for ũ, and retain the associated p
  - Numerical uncertainty estimate is based on median deviation and need not be symmetric. (This part of the algorithm is not used in this presentation)
- Subject matter expertise enters through bounding rates and screening intermediate results
  - Here: \( p_{\text{th}}=2.0, p_{\text{L}}=0.1, p_{\text{U}}=4.0 \); no detailed screening
RMR appears to give better results than 3-mesh Rex

RMR extrapolated solution:
- Well behaved in free stream
- Some noise in the vortex

Density field of isentropic vortex on quad meshes 3, 4, and 5 interpolated to mesh 3
RMR estimates of error and convergence rate are also improved

- Error is a bit noisy in and around the vortex
- Convergence rate is noisy away from the vortex but seems not to affect the extrapolated solution

Estimated density error and density convergence rate for isentropic vortex on quad meshes 3, 4, and 5 interpolated to mesh 3
Errors in the different error estimates each show weaknesses

- Rex2 misses the center and edge of the vortex
- Rex3 does the best within the vortex but shows spurious behavior
- RMR exhibits large errors within the vortex

Estimated density error and density convergence rate for isentropic vortex on quad meshes 3, 4, and 5 interpolated to mesh 3
Global error norms for density

- Global error norms of density on mesh 3

<table>
<thead>
<tr>
<th></th>
<th>exact</th>
<th>Rex2 (p=2)</th>
<th>Rex3</th>
<th>RMR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>e_rhoo</td>
<td></td>
<td>_1</td>
</tr>
</tbody>
</table>

- Errors of density error estimates on mesh 3

<table>
<thead>
<tr>
<th></th>
<th>Rex2 (p=2)</th>
<th>Rex3</th>
<th>RMR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>e_rhoo</td>
<td></td>
</tr>
<tr>
<td>e_rhomin</td>
<td>-1.2473e-3</td>
<td>-0.21050</td>
<td>-7.1169e-4</td>
</tr>
<tr>
<td>e_rhomin</td>
<td>9.4708e-3</td>
<td>0.021654</td>
<td>0.027708</td>
</tr>
</tbody>
</table>
Errors in the different error estimates each show weaknesses
Global error norms for density

- Global error norms of density on mesh 3

<table>
<thead>
<tr>
<th></th>
<th>exact</th>
<th>Rex2 (p=2)</th>
<th>Rex3</th>
<th>RMR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>3 to 4 on 3</td>
<td>2 to 4 on 3</td>
<td>3 to 5 on 3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>e_rhoo</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Errors of density error estimates on mesh 3

<table>
<thead>
<tr>
<th></th>
<th>Rex2 (p=2)</th>
<th>Rex3</th>
<th>RMR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3 to 4 on 3</td>
<td>2 to 4 on 3</td>
<td>3 to 5 on 3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>e_rhoo</td>
<td></td>
</tr>
<tr>
<td>e_rhomin</td>
<td>-1.2473e-3</td>
<td>-1.47373</td>
<td>-0.21050</td>
</tr>
<tr>
<td>e_rhomin</td>
<td>9.4708e-3</td>
<td>14.4987</td>
<td>0.021654</td>
</tr>
</tbody>
</table>
Concluding remarks

- The implementation of RMR as tested here is work in progress
  - Not mature – library choices might be improved, possibly bugs
  - Solution verification for field variables is not the common use case
- As implemented, RMR appears to be more robust than 3-mesh Richardson extrapolation, but not as globally accurate or well-behaved as 2-mesh Richardson extrapolation
  - Questionable set of models included; median selection breaks down if there are too many outliers
  - Is there a more accurate way of choosing the extrapolated solution than median selection?

Acknowledgement: Kevin Copps for an early implementation of RMR, Bill Rider and Brian Carnes for helpful discussions
BACKUP SLIDES
Review of code verification

1. Exact and approximate values of field variables on suitably refined meshes

2. Exact errors on each mesh

3. “Exact” global error norm on each mesh

\[
\|e_1\|, \ldots, \|e_N\|
\]
Sierra Aero – compressible flow code

- Solves compressible Euler and Navier-Stokes equations, including RANS turbulence models
- Demonstrated parallel scaling to tens of thousands of cores
- 2D and 3D unstructured meshes, several element types
- Numerical method:
  - Edge-based finite elements a la Barth
  - Nodal values are interpreted as point values or cell averages of a dual mesh
  - TVD methods compute fluxes normal to cell edges
  - First or second order spatial accuracy
  - Implicit or explicit time advancement methods
Isentropic Vortex, triangle meshes

<table>
<thead>
<tr>
<th></th>
<th>exact</th>
<th>ET/DC</th>
<th>Rex2 (p=2)</th>
<th>Rex2 (p=1)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>3 to 4 on 3</td>
<td>3 to 4 on 4</td>
<td>4 to 5 on 4</td>
</tr>
<tr>
<td>$</td>
<td></td>
<td>e_{\text{rho}}</td>
<td></td>
<td>$</td>
</tr>
<tr>
<td>$</td>
<td></td>
<td>e_{\text{u}}</td>
<td></td>
<td>$</td>
</tr>
<tr>
<td>$</td>
<td></td>
<td>e_{\text{v}}</td>
<td></td>
<td>$</td>
</tr>
<tr>
<td>$</td>
<td></td>
<td>e_{\text{T}}</td>
<td></td>
<td>$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>exact</th>
<th>ET/DC</th>
<th>Rex2 (p=2)</th>
<th>Rex2 (p=1)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>3 to 4 on 3</td>
<td>3 to 4 on 4</td>
<td>4 to 5 on 4</td>
</tr>
<tr>
<td>$</td>
<td></td>
<td>e_{\text{rho}}</td>
<td></td>
<td>$</td>
</tr>
<tr>
<td>$</td>
<td></td>
<td>e_{\text{u}}</td>
<td></td>
<td>$</td>
</tr>
<tr>
<td>$</td>
<td></td>
<td>e_{\text{v}}</td>
<td></td>
<td>$</td>
</tr>
<tr>
<td>$</td>
<td></td>
<td>e_{\text{T}}</td>
<td></td>
<td>$</td>
</tr>
</tbody>
</table>
Isentropic Vortex, quad meshes

<table>
<thead>
<tr>
<th></th>
<th>exact</th>
<th>ET/DC</th>
<th>Rex2 (p=2)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>3 to 4 on 3</td>
<td>3 to 4 on 4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>e_rho</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>e_u</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>e_v</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>e_T</td>
<td></td>
</tr>
</tbody>
</table>

|                  | exact | ET/DC     | Rex2 (p=1)                                      |
|                  |       | 3 to 4 on 3 | 3 to 4 on 4           | 4 to 5 on 4           | 4 to 5 on 5           |
| ||e_rho||        | 1.63648e-4 | 1.75451e-4 | 6.93274e-4           | 7.14159e-4           | 2.77578e-4           | 2.90957e-4           |
| ||e_u||          | 0.140608 | 0.154432   | 0.566488           | 0.590624           | 0.235755           | 0.246528           |
| ||e_v||          | 0.147940 | 0.157604   | 0.598565           | 0.624239           | 0.237982           | 0.247484           |
| ||e_T||          | 0.0274052 | 0.0280981  | 0.129465           | 0.129403           | 0.0472372          | 0.0474375          |