Interpolation Method needed for Numerical Uncertainty Analysis of Computational Fluid Dynamics

Curtis Groves
Ph.D. Candidate, University of Central Florida, Orlando, FL
Fluids Engineer, NASA Kennedy Space Center, FL

Marcel Ilie, Ph.D.
Former Assistant Professor, University of Central Florida, Orlando, FL

Paul Schallhorn, Ph.D.
Environments and Launch Approval Branch Chief, NASA Kennedy Space Center, FL
Problem

• Using Computational Fluid Dynamics (CFD) to predict a flow field is an approximation to the exact problem and uncertainties exist.

• There is a method to approximate the errors in CFD via Richardson’s Extrapolation.
  – This method is based off of progressive grid refinement.

• Unless using a Structured Grid with every other point, some interpolation method must be used.
Summary of Richardson’s Extrapolation

- Navier Stokes Equations
  - 2\textsuperscript{nd} order, non-homogeneous, non-linear partial differential equations
- Richardson’s Extrapolation is used to produce 4\textsuperscript{th} order accurate solution from separate 2\textsuperscript{nd} order accurate Navier Stokes Solutions
Summary of Richardson’s Extrapolation


• Assumptions
  1. Three discrete solutions are in the asymptotic range
  2. Meshes have a uniform spacing over the domain
  3. Meshes are related through systematic refinement
  4. Solutions are smooth
  5. Other sources of numerical error are small
• 5 Step Procedure for Uncertainty Estimation
  – Step 1: Representative Grid Size

\[ h = \left[ \frac{\sum_{i=1}^{N} \Delta V_i}{N} \right]^{1/3} \]

where
N = total number of cells used for the computations
\( \Delta V_i \) = volume of the \( i^{th} \) cell [4]

\[ h_1 < h_2 < h_3 \]
Step 2: Select 3 significantly \((r>1.3)\) different grid sizes

\[
\begin{align*}
    r_{21} &= \frac{h_2}{h_1} \\
    r_{32} &= \frac{h_3}{h_2}
\end{align*}
\]

Use CFD Simulation to analyze key variables, \(\varphi\):

\[
\begin{align*}
    \varepsilon_{32} &= \varphi_3 - \varphi_2 \\
    \varepsilon_{21} &= \varphi_2 - \varphi_1
\end{align*}
\]
– Step 3: Calculate observed order, \( p \)

\[
p = \left[ \frac{1}{\ln(r_{21})} \right] \ln \left| \frac{\varepsilon_{32}}{\varepsilon_{21}} \right| + q(p)
\]

\[
q(p) = \ln \left( \frac{r_{21}^p - s}{r_{32}^p - s} \right)
\]

\[
s = 1 \cdot \text{sign}(\frac{\varepsilon_{32}}{\varepsilon_{21}})
\]
– Step 4: Calculate extrapolated values

\[ \phi_{ext}^{21} = \left( r_{21}^p \phi_1 - \phi_2 \right) / \left\| r_{21}^p - 1 \right\| \]

\[ e_a^{21} = \left| \frac{\phi_1 - \phi_2}{\phi_1} \right| \]
– Step 5: Calculate Fine Grid Convergence Index & Numerical Uncertainty

\[ GCI_{\text{fine}}^{21} = \frac{Fs \cdot e_a^{21}}{r_p^{21} - 1} \]

The Factor of Safety, \( Fs = 1.25 \)

– Assumption that the distribution is Gaussian about the fine grid, 90% Confidence

\[ U_{\text{num}} = GCI / 1.65 \]
Solver Interpolation

- **FLUENT**
  - Includes a Mesh-to-Mesh Interpolation
  - Performs a zeroth-order (nearest neighbor) interpolation
  - Designed for initial conditions from a previous solution

- **OPENFOAM**
  - Mapfields fuction interpolation
  - Used for initialization of a solution from a previous model

- Using these ‘zeroth-order’ interpolation schemes is not sufficient for comparing errors from the mesh
Matlab Interpolation Schemes

- Matlab
  - High level language used for numerical computations
- CFD data is in various forms
  - 1D, 2D, 3D, uniform, non-uniform
  - Generic Scheme is sought for all CFD data

<table>
<thead>
<tr>
<th>Interpolation Method</th>
<th>interp1</th>
<th>interp2</th>
<th>interp3</th>
</tr>
</thead>
<tbody>
<tr>
<td>'nearest'</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>'linear'</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>'spline'</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>'pchip'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>'cubic'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>'v5cubic'</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Legend: X (uniformly-spaced only)
Example Problem

- Fully developed flow between parallel plates
  - Exact Solution to Navier Stokes
  - Provide a good example of errors that can be induced from interpolation

\[
\bar{V} = -\frac{1}{12\mu} \left( \frac{\delta P}{\delta x} \right) a^2
\]

\[
u = \frac{a^2}{2\mu} \left( \frac{\delta P}{\delta x} \right) \left[ \left( \frac{y}{a} \right)^2 - \left( \frac{y}{a} \right) \right]
\]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>a (m)</td>
<td>0.1 m</td>
</tr>
<tr>
<td>\rho (kg/m³)</td>
<td>1.225 kg/m³</td>
</tr>
<tr>
<td>\mu (Ns/m²)</td>
<td>0.0001789 Ns/m²</td>
</tr>
<tr>
<td>dp/dx (N/m³)</td>
<td>-0.004 N/m³</td>
</tr>
</tbody>
</table>
Example Problem

- Constructed a CFD Model in FLUENT
  - 3 Grids
    - Coarse, 7,140 Cells
    - Medium, 14,186 Cells
    - Fine, 24,780 Cells
Example Problem

- Interpolation Direction?
  1. Interpolate Coarse and Medium Mesh -> Fine

```
\varepsilon_{21} = \varphi_2 - \varphi_1
\varepsilon_{32} = \varphi_3 - \varphi_2
```

1. Interpolate Medium and Fine Mesh -> Coarse
Example Problem

1. Linearly Interpolate Coarse and Medium Mesh -> Fine

\[ \varphi_3 \quad \varphi_2 \quad \varphi_1 \]

coarse \quad medium \quad fine

Max % Error Extrapolated Values: 0.8950
Average % Error Extrapolated Values: 0.0596
Example Problem

2. Linearly Interpolate Fine and Medium Mesh -> Coarse

Max % Error Extrapolated Values       Average % Error Extrapolated Values
0.0792                                  0.0175
Example Problem

- Interpolation Direction?

1. Interpolate Coarse and Medium Mesh -> Fine

2. Interpolate Medium and Fine Mesh -> Coarse
Interpolating to the coarse grid was selected

Other interpolation methods

- “nearest” – Fluent’s Mesh-to-Mesh
- “linear” – Matlab
  
  \[
  y_{fi} = \text{interp1}(\text{fine}(:,2), \text{fine}(:,1), \text{coarse}(:,2), 'linear')
  \]
- “cubic” – Matlab
  
  \[
  y_{fi} = \text{interp1}(\text{fine}(:,2), \text{fine}(:,1), \text{coarse}(:,2), 'cubic')
  \]
Example Problem

- “nearest” – Fluent’s Mesh-to-Mesh

Zeroth Order Interpolation
Example Problem

• “linear” – Matlab

\[ yfi = \text{interp1}(	ext{fine(:,2)}, \text{fine(:,1)}, \text{coarse(:,2)}, 'linear') \]

1st Order Interpolation
Example Problem

- "cubic"
  - Matlab
  
  \[ yfi = \text{interp1}(\text{fine}(:,2), \text{fine}(:,1), \text{coarse}(:,2), \text{'}cubic\text{')} \]
Matlab Interpolation Schemes

• Extending the Interpolation Schemes to 2D and 3D
  • Interp2 and Interp3 Matlab Functions
    • Require use of MeshGrid
      • Transforms the domain of vectors into arrays
      • For Meshes in the 4 million to 8 million Cell Range
        • Error “Maximum variable size allowed by program is exceeded”
  • Griddata Function
    • Nearest, Linear, Natural, Cubic, and v4
    • Nearest, Linear, and Natural are the only options available in 2D and 3D
  • The only options available for 1D, 2D, and 3D
    • Interp1 and Griddata – ‘nearest’ and ‘linear’
3D Example

- Airflow around encapsulated spacecraft
  - Matlab griddata ‘linear’ option used
  - Interpolating Fine and Medium Grid onto Coarse Grid
3D Example

• Comparing using a Line Plot
3D Example

• Comparing using a Line Plot
Conclusion / Recommendation

- By comparing the interpolation schemes in one, two, and three dimensions and investigating the options that are readily available in Matlab
  - Recommend the “linear” option be used when comparing the error or uncertainty due to the grid
    - interp1 or griddata Matlab commands
- If coarse grid has the level of detail required
  - Recommend interpolating from the fine and medium grids onto the coarse grid
    - Lower Error in the Extrapolated Solution
    - Smaller Data Set
- Future Work include higher order interpolation schemes in 3D (Radial Basis Function Interpolation, 4th order)