Numerical Uncertainty Assessment

Verification = Solving the equations right

OBJECTIVE
Show that the numerical solution converges to the exact solution of the differential equation as $\Delta x \to 0$ and $\Delta t \to 0$.

APPROACH
Solve problem with different $\Delta x$ and $\Delta t$ and compare.
Validation = Solving the right equations

OBJECTIVE
Show that the differential equations accurately model the physics of the problem.

APPROACH
Compare computational solution to experimental results.
Verification

• Must always precede validation
• Check that physical features of the solution are adequately resolved (boundary layers, crack tips, etc.)
• Document how much solution changes when $\Delta x$ and $\Delta t$ are changed. Determine convergence rate.
• Compare to exact solution of PDE
• Compare solutions with two different numerical methods
Validation

• Account for uncertainty in both computational and experimental results.

• Important to validate using the features that are most sensitive to the model assumptions. (It is common that some features compare well to an experiment while other features of a numerical solution compare poorly.)
Example: Flow Past a Cylinder

Color shows vorticity field (red – positive, blue – negative)

Comparison Parameters

- Drag coefficient: \[ C_D = \frac{D}{\frac{1}{2} \rho U^2 A} \]
- Lift coefficient: \[ C_L = \frac{L}{\frac{1}{2} \rho U^2 A} \]
- Strouhal number: \[ St = \frac{f d}{U} \]
Verification: Compare solutions on different grids

- Fine Grid Computation: 21,580 points
- Course Grid Computation: 10,790 points

<table>
<thead>
<tr>
<th>Flow Measure</th>
<th>Course Grid</th>
<th>Fine Grid</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\bar{C}_D$</td>
<td>1.4 ± 0.1</td>
<td>1.4 ± 0.1</td>
</tr>
<tr>
<td>$C_{L,amp}$</td>
<td>0.6 ± 0.1</td>
<td>0.7 ± 0.1</td>
</tr>
<tr>
<td>$St$</td>
<td>0.21 ± 0.01</td>
<td>0.20 ± 0.01</td>
</tr>
</tbody>
</table>
Validation: Comparison to experimental results

<table>
<thead>
<tr>
<th>Flow Measure</th>
<th>Experiment $Re = 100$</th>
<th>Computation $Re = 100$</th>
<th>Experiment $Re = 300$</th>
<th>Computation $Re = 300$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_D$</td>
<td>$1.9 \pm 0.1$</td>
<td>$2.4 \pm 0.1$</td>
<td>$1.4 \pm 0.1$</td>
<td>$1.4 \pm 0.1$</td>
</tr>
<tr>
<td>$C_{L,amp}$</td>
<td>$0.6 \pm 0.25$</td>
<td>$0.4 \pm 0.1$</td>
<td>$0.6 \pm 0.25$</td>
<td>$0.7 \pm 0.1$</td>
</tr>
<tr>
<td>$St$</td>
<td>$0.15 \pm 0.01$</td>
<td>$0.15 \pm 0.01$</td>
<td>$0.21 \pm 0.01$</td>
<td>$0.20 \pm 0.01$</td>
</tr>
</tbody>
</table>

Comparison of computational results with reference experimental data compiled by Fleischmann and Sallet (1985).
Richardson Extrapolation

- A method to make a calculation more accurate

Let $f$ be a numerical solution on a grid with step size $h$. Expand $f$ about the exact solution to write

$$f = f_{\text{exact}} + g_1 h + g_2 h^2 + g_3 h^3 + ...$$

where $g_1$, $g_2$, $g_3$, etc., are independent of $h$. 
Richardson Extrapolation

Consider a second-order calculation, such that $g_1 = 0$. Perform calculations on two grids, with increment size $h_1$ and $h_2$, such that

\[ f_1 = f_{\text{exact}} + g_2 h_1^2 + g_3 h_1^3 + O(h_1^4) \]  
(1)

\[ f_2 = f_{\text{exact}} + g_2 h_2^2 + g_3 h_2^3 + O(h_2^4) \]  
(2)
Richardson Extrapolation

Taking $h_2^2(1) - h_1^2(2)$ and solving for $f_{exact}$ gives

$$f_{exact} = \frac{f_1 h_2^2 - f_2 h_1^2}{h_2^2 - h_1^2} + O(h^3) \quad (3)$$

Hence, we can decrease the error in the numerical solution from $O(h^2)$ to $O(h^3)$ by forming the ratio above. For grid doubling ($h_2 = 2h_1$) we have

$$f_{exact} = \frac{4}{3} f_1 - \frac{1}{3} f_2 + O(h^3)$$
Grid Convergence Index (GCI)

- A method to quantify the error in a computation

Let $p = \text{order of computational method}$

$r = \text{grid refinement ratio}$

$(h_2 = r \ h_1)$

We can then write (3) as

$$f_{exact} = f_1 + \frac{f_1 - f_2}{r^p - 1} + O(h^{p+1})$$

(4)
Grid Convergence Index (GCI)

Define the **Grid Convergence Index (GCI)** as the relative error, given by

$$GCI \equiv \frac{\varepsilon}{r^p - 1} \quad \text{where} \quad \varepsilon \equiv \left| \frac{f_1 - f_2}{f_1} \right|$$

Substituting this into (4) gives

$$GCI = \left| \frac{f_{\text{exact}} - f_1}{f_1} \right| + O(h^{p+1})$$

**GCI** is a measure of the relative discretization error of the computed solution.
Numerical Dissipation

In approximation of convection on a fixed grid, it is typically necessary to introduce additional “artificial” or “numerical” dissipation in order to stabilize the convective transport term. This numerical dissipation can be either implicit in the differencing method or added explicitly to the computation.
Example: Lax-Wendroff Method

Problem: Develop a numerically stable method to solve the advection equation to second-order accuracy.

\[ \frac{\partial \phi}{\partial t} + u \frac{\partial \phi}{\partial x} = 0 \]  \hspace{1cm} (5)
Lax-Wendroff Method

**Approach:** Expand \( \phi_{m}^{n+1} = \phi(x_m, t_{n+1}) \) about \( \phi_m^n \) using a Taylor series with local error \( O(k^3) \)

\[
\phi_{m}^{n+1} = \phi_{m}^{n} + k(\phi_{t})_{m}^{n} + \frac{1}{2} k^2 (\phi_{tt})_{m}^{n} + O(k^3) \tag{6}
\]

Differentiate (5) with respect to \( t \) to write

\[
\frac{\partial^2 \phi}{\partial t^2} = -u \frac{\partial^2 \phi}{\partial x \partial t} = u^2 \frac{\partial^2 \phi}{\partial x^2} \tag{7}
\]
Lax-Wendroff Method

Substitute (7) into (6) and solve for \((\phi_t)_m^n\)

\[
(\phi_t)_m^n = \frac{1}{k}(\phi_{m+1}^{n+1} - \phi_m^n) - \frac{1}{2}u^2k^2(\phi_{xx})_m^n + O(k^3) \tag{8}
\]

Plug (8) into (5) to get

\[
\frac{\phi_{m+1}^{n+1} - \phi_m^n}{k} = -u(\phi_x)_m^n + \frac{1}{2}u^2k^2(\phi_{xx})_m^n + O(k^3) \tag{9}
\]

FTCS Scheme (UNSTABLE)
Numerical Dissipation Term (STABLE IF CFL # < 1)
Effects of Numerical Dissipation

Causes regions of large $\phi$ to expand in size and decrease in strength

Example for advection of a vortex in an inviscid fluid:

Most commercial codes have high numerical dissipation because it makes the computations very stable.
Methods for Avoid Numerical Dissipation

- Use Lagrangian method (no numerical dissipation)
- Use high-order accurate method
- Use local grid refinement in regions with large values of $\phi$
- Use discretely-conservative method