Muddiest Points from Class 03/01

- *"Using ADI will not require us to use ghost cells correct?"*
  - You will need ghost cells for cell centered meshes (they are needed for the explicit direction terms)

- *"The reverse order for the sort on the 2nd part of the ADI was a little unclear."*
- *"The ordering of the mesh points into the vectors was confusing. Aren't the vectors supposed to contain diagonals?"*
- *"The creation of 1D vectors from 2D vectors for tridiagonalization is still in clear to me. Do these 1D vectors only contain the information from the diagonals, or the whole 2D vector just put into 1D form?"*
- *"Will the 2D code require a sorting and de-sorting operator to bring the solution array values into and out of the 1D arrays?"*
  - in 2D the solution variables are arrays: u(i,j)
  - for the tridiagonal solve, the 2D solution variable arrays have to be sorted into 1D vectors (since we solve A**x** = **b**)
  - to use the tridiagonal method, non-zero values can only be on the main diagonal and the 2 neighbor diagonals

Step 1: $-d_1 u^{n+1/2}_{i-1,j} + (1+2d_1) u^{n+1/2}_{i,j} - d_1 u^{n+1/2}_{i+1,j}$

i-1,j must be next to i,j must be next to i+1,j

```
do j = 1, N
    do i = 1,M
        d((j-1)*M+i)=…
        …
    end do
end do
```

Step 2: $-d_2 u^{n+1}_{i,j-1} + (1+2d_2) u^{n+1}_{i,j} - d_2 u^{n+1}_{i,j+1}$

i,j-1 must be next to i,j must be next to i,j+1

```
do i = 1, M
    do j = 1,N
        d((i-1)*N+j)=…
        …
    end do
end do
```

  - one needs at least 1 sorting/desorting step (Fortran/Matlab: step 2, C/C++: step 1)
  - the other direction can be done with a reshape call since it is already sorted correctly in memory

Muddiest Points from Class 03/01

- *"Is the ADI more accurate than the Crank-Nicholson? It includes the 4th order terms, so would we be able to observe that difference at all?"*
  - ADI and Crank-Nicolson have the same order of accuracy. All 2nd-order methods have 4th-order error terms as well (and 3rd, 5th, 6th, etc.)

- *"In the code for parabolic solvers (last slide) it mentions calculating a stable time step at the first of each while loop. Does this mean explicit methods have a varying time step with each solution to guarantee stability? Or is this just referring to "hitting the exact output times?"*
- *"In the final slide, is the 'if' loop presented first within the 'calculate stable time step' function in the second box?"*
  - Here, it refers to hitting the exact output time, since for parabolic PDEs we study, the stable time step is a constant
  - Yes

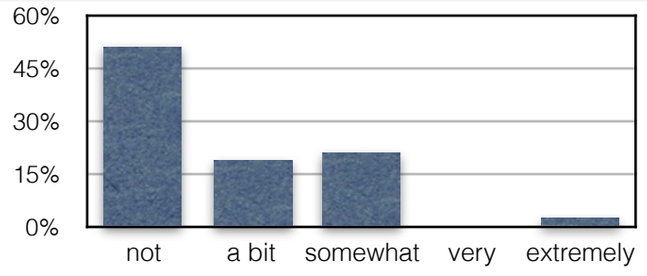
- *"I am actually quiet confused about the meaning of δx, if △x=δx ,then d1*d2*δx^2*δy^2 is no longer O(h^4) since there is a △x^2 in d1 as well and the result should be (α^2 *△t^2)/4. Then ADI is no long the same as Crank-Nicholson."*
  - δx is not equal to △x
  - δ_x^2 is simply a shorthand for

$$\delta_x^2 u_{i,j} = u_{i+1,j} - 2u_{i,j} + u_{i-1,j}$$

$$\delta_y^2 u_{i,j} = u_{i,j+1} - 2u_{i,j} + u_{i,j-1}$$

```
if (time < outputTime) .and. &
   (time + dt >= outputTime) then
    dt = outputTime - time;
    setFlagforOuput;
end if
```

```
setInitialConditions;
applyBoundaryConditions;
time = intialTime;
while (time < endTime)
     dt = calculateStableTimeStep;
     calculateNewTimeStepSolution;
     applyBoundaryConditions;
     time = time+dt;
     doOutputIfRequired;
end if
```



Verification & Validation
(Part I)

or

How can I trust my CFD results?

or

Can I quantify the error/uncertainty of a CFD simulation?

Definition of Terms:

- **Uncertainty**

  *"A **potential** deficiency in any phase or activity of the modeling process that is due to the lack of knowledge." (AIAA G-077-1998)*

➡ Aleatory uncertainty is the inherent variation associated with the physical system or the environment
(irreducible uncertainty, variability, and stochastic uncertainty)

- ✓ Variation in thermodynamic properties due to manufacturing
- ✓ Variation in joint stiffness and damping in structures
- ✓ Random vibrational input to a structure

➡ Epistemic uncertainty is a potential deficiency in any phase of the modeling process that is due to lack of knowledge
(reducible uncertainty, model form uncertainty, and subjective uncertainty)

- ✓ Poor understanding of fracture dynamics
- ✓ Poor understanding of primary atomization mechanisms
- ✓ Poor understanding of chemical reaction mechanisms

Definition of Terms:

- **Error**

  "*A **recognizable** deficiency in any phase or activity of modeling and simulation that is **not** due to lack of knowledge.*" *(AIAA G-077-1998)*

➡ Acknowledged errors are errors that can be estimated, bounded, or ordered

✓ Finite precision arithmetic in a digital computer
✓ Insufficient spatial discretization
✓ Insufficient temporal discretization
✓ Insufficient iterative convergence

➡ Unacknowledged errors are mistakes or blunders

✓ Computer programming errors (source code or compiler)
✓ Use of incorrect input files (geometry, material properties)

The following slides rely/copy heavily on the following tutorial/talk:

Verification and Validation in Computational Simulation

Dr. William L. Oberkampf
Distinguished Member Technical Staff
Validation and Uncertainty Quantification Department
Sandia National Laboratories, Albuquerque, New Mexico
wloberk@sandia.gov

2004 Transport Task Force Meeting
Salt Lake City, Utah
April 29, 2004

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Motivation

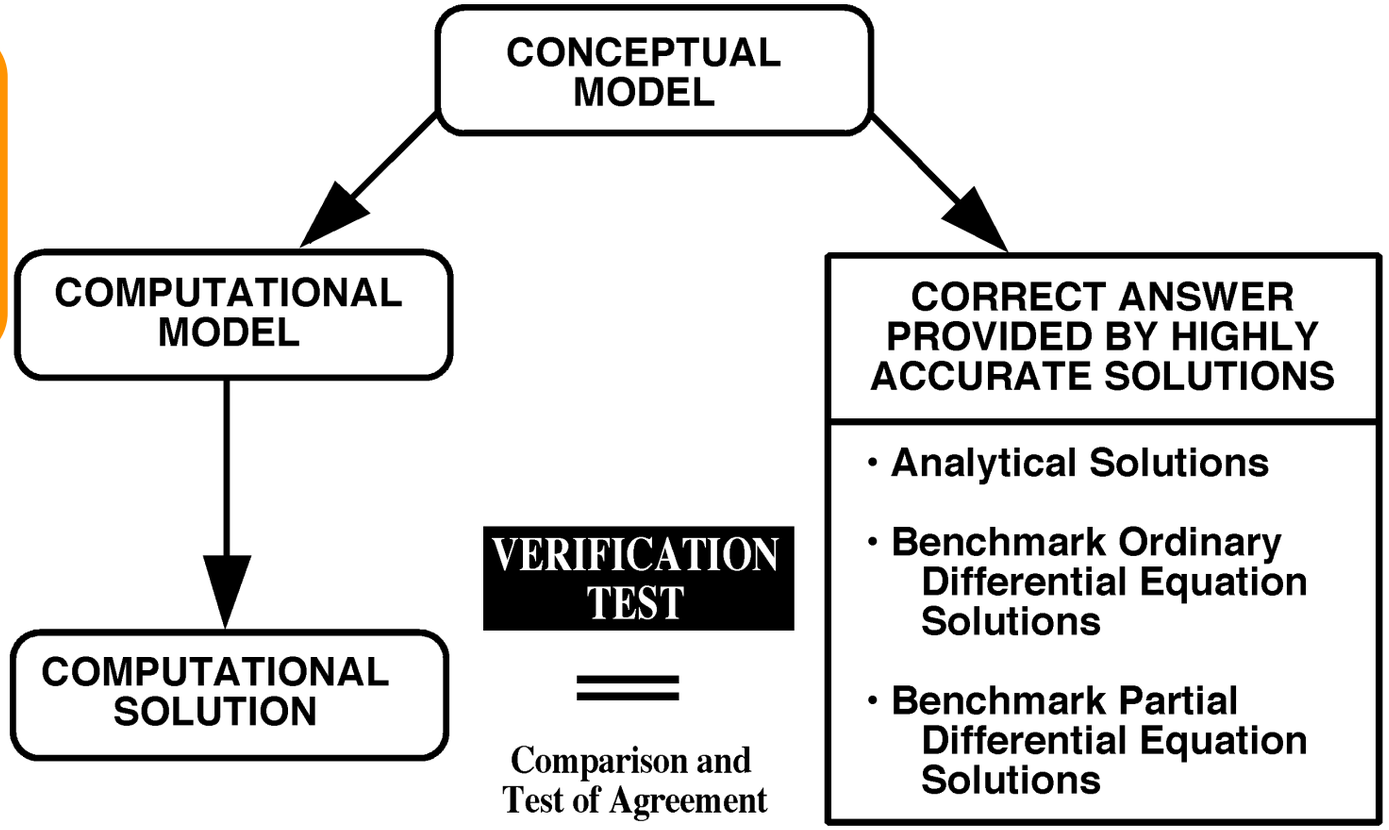
- **Why is verification and validation (V&V) important?**
  - **V&V procedures are the primary means of assessing accuracy in computational simulations.**
  - **V&V procedures are the tools with which we build confidence and credibility in computational simulations.**

Terminology: Verification

American Institute of Aeronautics and Astronautics, Committee on Standards in Computational Fluid Dynamics definition (1998):

Verification: The process of determining that a model implementation accurately represents the developer's conceptual description of the model and the solution to the model

! Verification deals with mathematics



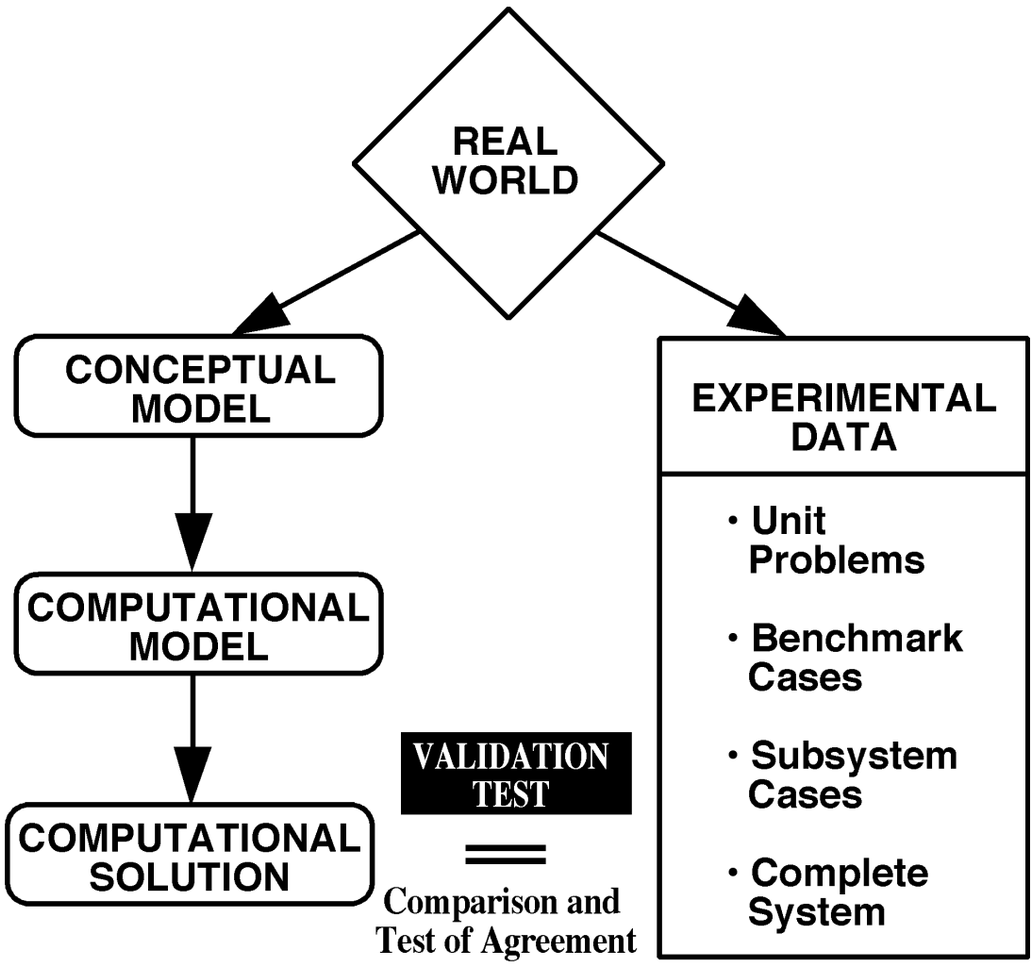
Two Types of Verification

- **Verification is now commonly divided into two types:**
- **Code Verification: Verification activities directed toward:**
  - **Finding and removing mistakes in the source code**
  - **Finding and removing errors in numerical algorithms**
  - **Improving software using software quality assurance practices**
- **Solution Verification: Verification activities directed toward:**
  - **Assuring the accuracy of input data for the problem of interest**
  - **Estimating the numerical solution error**
  - **Assuring the accuracy of output data for the problem of interest**

Definition of Validation

Validation: The process of determining the degree to which a model is an accurate representation of the real world from the perspective of the intended uses of the model

! Validation deals with physics



Important Features of Verification and Validation

- Both definitions stress "process of determining":
  - Each process provides evidence (substantiation)
  - The veracity, correctness, or accuracy of all possible solutions to the conceptual model cannot be proven
- Both definitions stress comparison with a reference standard:
  - A measurement of accuracy, or error, must be available
  - For verification, the standard is the "conceptual model"
  - For validation, the standard is the "real world"

Verification provides evidence that the computational model is solved correctly and accurately.

Validation provides evidence that the mathematical model accurately relates to experimental measurements.

Numerical Algorithm Verification

- **Formal order of accuracy of a numerical method is determined by:**
  - **Taylor series analysis for finite-difference and finite volume methods**
  - **Interpolation theory for finite-element methods**
- **Consider the 1-D unsteady heat conduction equation:**

$$\frac{\partial T}{\partial t} - \alpha \frac{\partial^2 T}{\partial x^2} = 0$$

- **Using a forward difference in time and a centered difference in space, the Taylor series analysis results in:**

$$\frac{\partial T}{\partial t} - \alpha \frac{\partial^2 T}{\partial x^2} = \left[-\frac{1}{2} \frac{\partial^2 T}{\partial t^2} \right] \Delta t + \left[\frac{\alpha}{12} \frac{\partial^4 T}{\partial x^4} \right] (\Delta x)^2 + O\left(\Delta t^2\right) + O\left(\Delta x^4\right)$$

Observed Order of Accuracy

- Computed solutions do not typically reproduce the formal order of accuracy
- Factors that can degrade the formal order of accuracy include:
  - Mistakes in the computer code, i.e., programming errors
  - Δx, Δy, Δz, Δt are not sufficiently small for the solution to be in the asymptotic convergence region, i.e., truncation errors
  - Singularities or discontinuities in the solution domain and on the boundaries
  - Insufficient iterative convergence for solving nonlinear equations
  - Round-off error due to finite word length in the computer
- We use the term "observed" order of accuracy for the actual accuracy determined from computed solutions

What's needed?

- We need the exact solution to compare our numerical solution to
- Ideally we would have an analytical exact solution!

Methods for Determining the Observed Order of Accuracy

- **Method of Exact Solutions (MES):**
  - **MES involves the comparison of a numerical solution to the exact solution to the governing PDEs**
  - **MES is the traditional method for code verification testing**
  - **Number and variety of exact solutions is extremely small**
- **Method of Manufactured Solutions (MMS):**
  - **MMS is a more general and more powerful approach for code verification**
  - **Rather than trying to find an exact solution to a PDE, we "manufacture" an exact solution *a priori***
  - **It is not required that the manufactured solution be physically real**
  - **Use the PDE operator to analytically generate source terms in a new PDE**
  - **The manufactured solution is the exact solution to a new (modified) equation: original PDE + source terms**
  - **MMS involves solving the backward problem: given an original PDE and a chosen solution, find a modified PDE which that chosen solution will satisfy**
  - **Initial & boundary conditions are determined from the solution, after the fact**

Method of Exact Solution (MES) Example:

- 2D unsteady, imcompressible Navier-Stokes equations
- Exact solution due to Chorin (1967) & Pearson (1964)

$$u_1(x_1, x_2, t) = -\cos x_1 \sin x_2 \, e^{-2t}$$

$$u_2(x_1, x_2, t) = \sin x_1 \cos x_2 \, e^{-2t}$$

$$p(x_1, x_2, t) = -\tfrac{1}{4}(\cos 2x_1 + \cos 2x_2) \, e^{-4t}$$

from Kim & Moin (1985)

Method of Exact Solution (MES) Example:

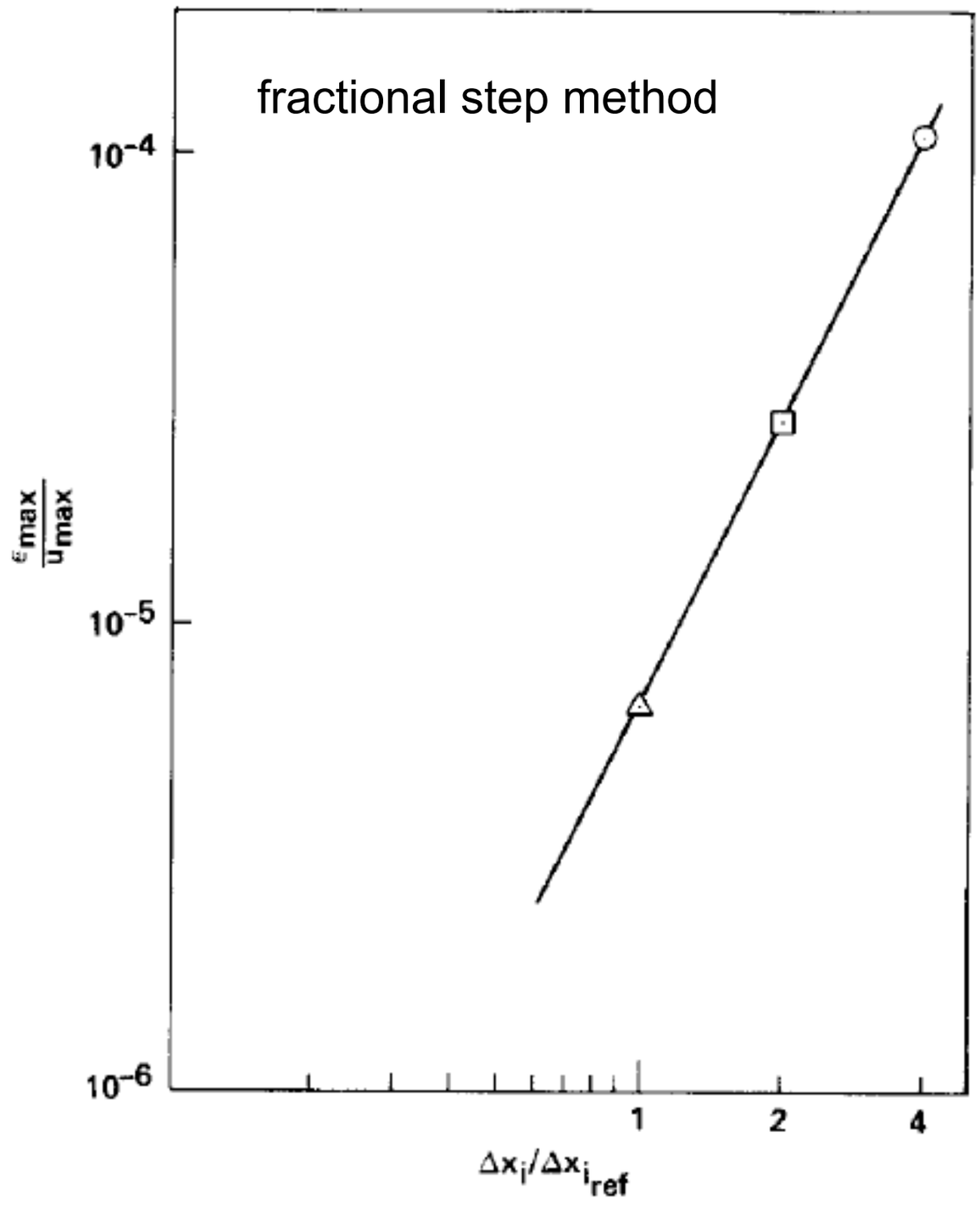


FIG. 2. Maximum error as a function of mesh refinement.

from Kim & Moin (1985)

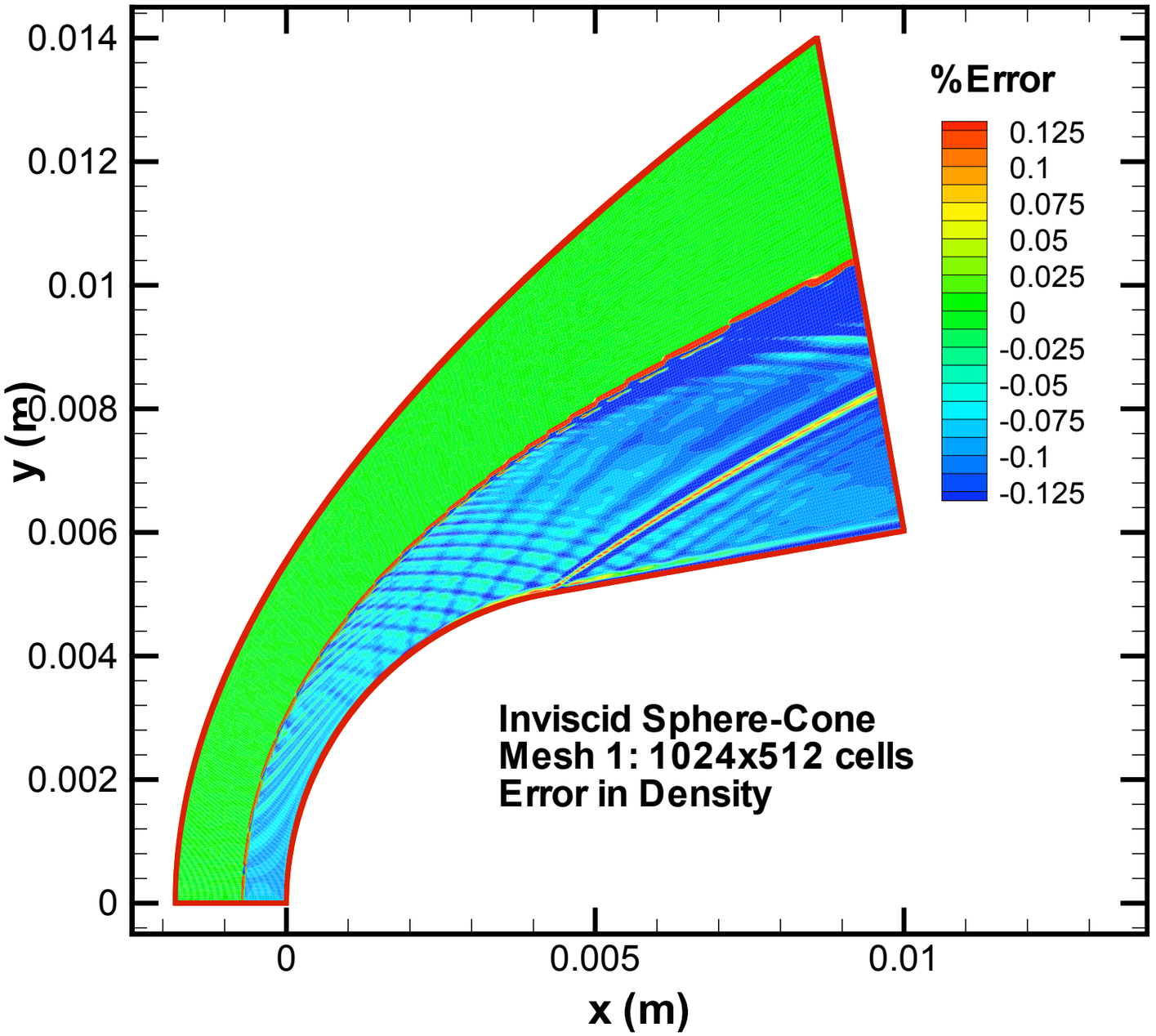
Solution Verification

- **Three aspects of solution verification:**
  1. **Verification of input data**
     - **Ensuring correct input files, grids, physical and material data, etc.**
  2. **Numerical error estimation of the solution**
     - **Mapping from continuum mathematics to discrete mathematics**
     - **Non-zero Δx, Δy, Δz, Δt**
     - **Insufficient iterative convergence for solving nonlinear equations**
     - **Round-off error due to finite word length in the computer**
  3. **Verification of output data**
     - **Ensuring that the correct files are used and post-processing steps taken**
- **Solution verification must be performed for every simulation that is sufficiently different from previous solutions**

Numerical Solution Error

- Discretization error (DE) arises due to the mapping of PDEs to discretized equations
- The DE can be clearly related to the truncation error (TE) using a Taylor series expansion for linear PDEs
- For nonlinear problems, the relation between DE and TE is not as straightforward (Celik and Hu, 2003)
- Discretization of the boundary conditions can dominate the numerical accuracy if the the order of accuracy is less than the interior scheme
- The total (or global) DE is made up of two components
  - Local DE due to the local element size
  - Error that has been transported from other regions (also known as pollution error)

Local and Transported Error (Roy, 2003)



- Mach 8 flow over a spherically-blunted-cone
- Local DE sources:
  - Capturing of the bow shock wave
  - Sphere-cone tangency point
- Error is also transported
  - By convection along streamlines
  - Along Mach waves in supersonic flow

Sandia National Laboratories

Approaches for Estimation of Discretization Error

- ***a priori*** **error estimation:**
  - **Estimated before the numerical solution is computed**
  - **Estimated by truncation error analysis for finite difference or finite volume scheme**
  - **Estimated by interpolation theory for finite element schemes**
  - **Not useful for practical problems because the magnitude of the error is only know within a (unknown) constant**

- ***a posteriori*** **error estimation**
  - **Estimated after at least one numerical solution is computed**
  - **Finite-element-based error estimation**
    - **Recovery methods: e.g., Zienkiewicz-Zhu (1992)**
    - **Residual methods, adjoint methods**
  - **Extrapolation-based error estimation**
    - **Richardson extrapolation (h-extrapolation)**
    - **Order extrapolation (p-extrapolation)**

Richardson Extrapolation

Result of a simulation, f, can be expressed as:

$$f = f_{h=0} + g_1 h + g_2 h^2 + g_3 h^3 + \dots$$

- assume a p-th order method, i.e. all $g_{i<p}$ are zero
- run a fine grid simulation (h_1) and coarse grid simulation (h_2) with $r = h_2/h_1$

$$f_1 = f_{h=0} + g_p h_1^p + \dots \quad |\cdot h_2^p$$ we would like to know exact solution: $f_{h=0}$

$$- \quad f_2 = f_{h=0} + g_p h_2^p + \dots \quad |\cdot h_1^p$$ ⇒ eliminate g_p

$$f_1 h_2^p - f_2 h_1^p = f_{h=0}\left(h_2^p - h_1^p\right)$$

$$f_{h=0} = \frac{f_1 h_2^p - f_2 h_1^p}{h_2^p - h_1^p} = \frac{f_1 h_2^p - f_1 h_1^p + f_1 h_1^p - f_2 h_1^p}{h_2^p - h_1^p} = f_1 + \frac{h_1^p (f_1 - f_2)}{h_2^p - h_1^p}$$

$$= f_1 + \frac{f_1 - f_2}{\frac{h_2^p}{h_1^p} - 1} = f_1 + \frac{f_1 - f_2}{r^p - 1} \quad \Rightarrow \quad f_{h=0} \approx f_1 + \frac{f_1 - f_2}{r^p - 1} \qquad r = \frac{h_2}{h_1}$$

Richardson Extrapolation

Result of a simulation, f, can be expressed as:

$$f = f_{h=0} + g_1 h + g_2 h^2 + g_3 h^3 + \dots$$

- assume a p-th order method, i.e. all $g_{i<p}$ are zero
- run a fine grid simulation (h_1) and coarse grid simulation (h_2) with $r = h_2/h_1$

$$f_{h=0} \approx f_1 + \frac{f_1 - f_2}{r^p - 1} \qquad \text{also} \quad f_{h=0} \approx f_2 + \frac{r^p}{r^p - 1}(f_1 - f_2)$$

- can apply Richardson extrapolation for every grid point or for solution functionals
- estimate fractional errors E_1 on fine grid and E_2 on coarse grid

$$E_1 = \frac{f_1 - f_{h=0}}{f_1} = \frac{f_1 - f_1 - \frac{f_1 - f_2}{r^p - 1}}{f_1} = -\frac{f_1 - f_2}{f_1}\frac{1}{r^p - 1} = \frac{\epsilon}{1 - r^p}$$

$$E_2 = \frac{f_2 - f_{h=0}}{f_1} = \frac{\epsilon r^p}{1 - r^p} \qquad \text{with relative error: } \epsilon = \frac{f_1 - f_2}{f_1}$$

But what's p? Order of Grid Convergence

- let's define an error E:

$$E = f(h) - f_{exact} = Ch^p + O(h^{p+1})$$

$$\Leftrightarrow \ \log(E) = \log(C) + p\log(h)$$

- ratio r between different grids should be $r >= 1.1$, best is $r = 2$
- should use at least 3 simulations
- use least squares fitting to determine p
- with constant r:

$$p = \frac{\log\left(\frac{f_3 - f_2}{f_2 - f_1}\right)}{\log(r)}$$

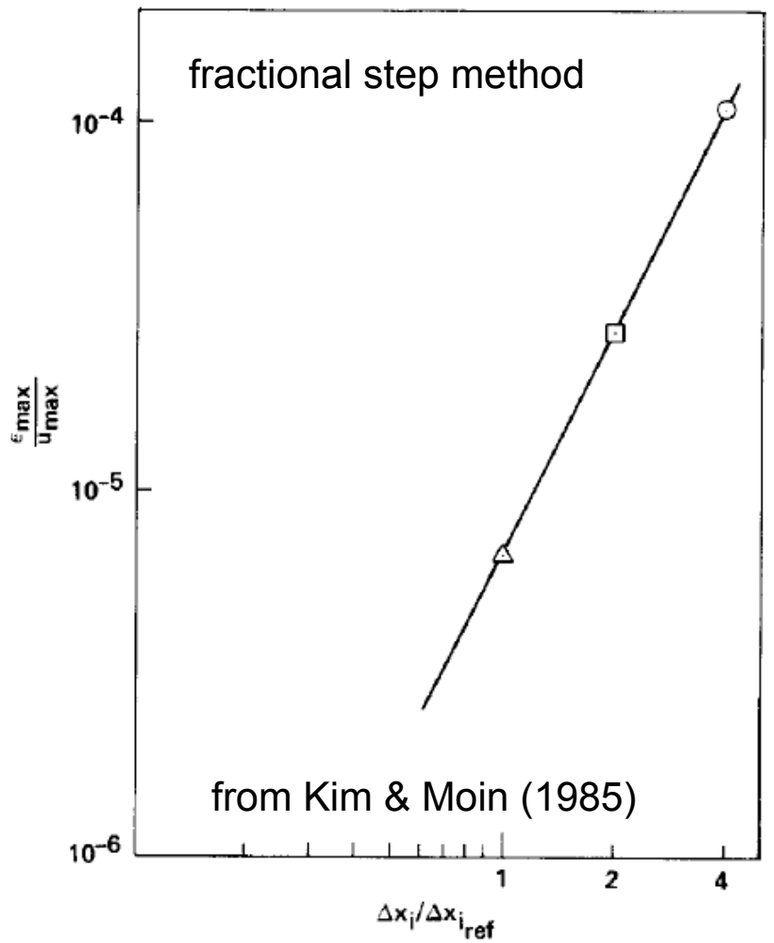


FIG. 2. Maximum error as a function of mesh refinement.

- must be in asymptotic convergence region, i.e. C is constant in $C = \frac{E}{h^p}$

$$C = \frac{E_1}{h_1^p} \qquad C = \frac{E_2}{h_2^p} \qquad \text{divide:} \qquad \frac{\frac{E_1}{h_1^p}}{\frac{E_2}{h_2^p}} = \frac{E_1}{E_2}\frac{h_2^p}{h_1^p} = \frac{E_1}{E_2}r^p = \frac{C}{C} = 1$$

Grid Convergence Index (Roache 1993)

methodology for uniform reporting of grid refinement studies

- use at least 2 solutions on different meshes, better are 3 or more
- GCI is a measure of the percentage the computed value is away from the asymptotic value (see fractional error)

$$GCI_{21} = F_{sec} \frac{|\varepsilon|}{r^p - 1}$$

2 simulations: $F_{sec} = 3$

3 or more simulations: $F_{sec} = 1.25$

- Check for asymptotic convergence region

$$\frac{GCI_{i,i+1}}{GCI_{i+1,i+2}} r^p \approx 1 \, ?$$

- Estimate for required grid resolution for desired level of accuracy, i.e. GCI*

$$r^* = \left(\frac{GCI^*}{GCI_{21}} \right)^{1/p}$$

Example: CFD Analysis of Pressure Recovery in an Inlet (NPARC Alliance)

grid	grid spacing h	pressure recovery
1	1	0.9705
2	2	0.96854
3	4	0.96178

grid spacing h
normalized to finest grid spacing

- Determine order of convergence p (theoretical p = 2):

$$p = \ln\left(\frac{f_3 - f_2}{f_2 - f_1}\right) / \ln(r) \qquad p = \ln\left(\frac{0.96178 - 0.96854}{0.96854 - 0.97050}\right) / \ln(2) = 1.78617$$

- Richardson extrapolation with 2 finest grids to estimate solution at *h=0*:

$$f_{h=0} \approx f_1 + \frac{f_1 - f_2}{r^p - 1} \qquad f_{h=0} \approx 0.9705 + \frac{0.9705 - 0.96854}{2^{1.78617} - 1} = 0.9713$$

- Grid Convergence Index:

$$GCI_{21} = F_{sec}\frac{|\varepsilon|}{r^p - 1}$$

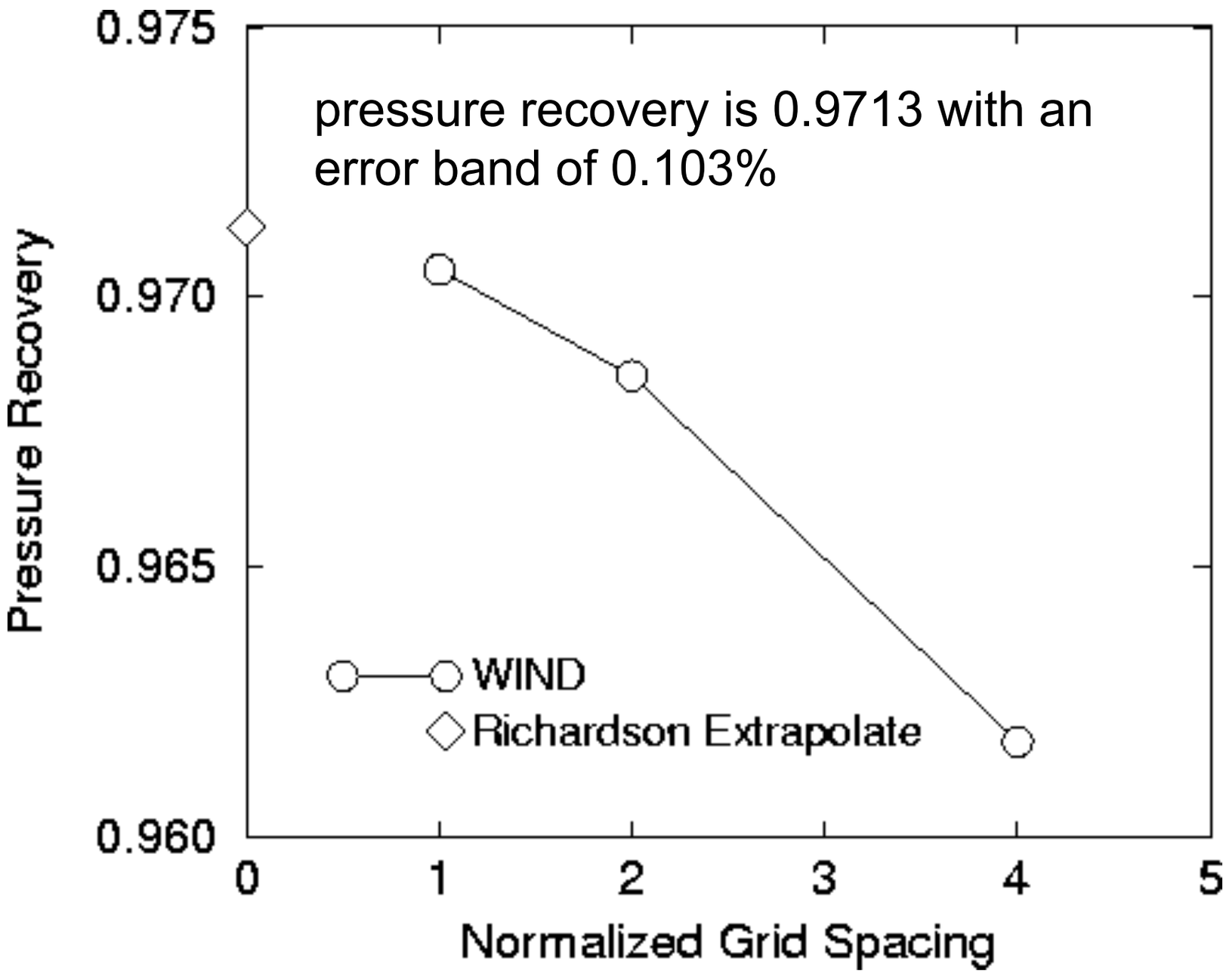
$$GCI_{12} = 1.25\frac{\left|\frac{0.9705 - 0.96854}{0.9705}\right|}{2^{1.78617} - 1} = 0.103\%$$

$$GCI_{23} = 1.25\frac{\left|\frac{0.96854 - 0.96178}{0.96854}\right|}{2^{1.78617} - 1} = 0.356\%$$

- Asymptotic range of convergence?

$$\frac{GCI_{i,i+1}}{GCI_{i+1,i+2}} r^p \approx 1\,? \qquad \frac{0.103}{0.356} 2^{1.78617} = 0.9979$$

Example: CFD Analysis of Pressure Recovery in an Inlet (NPARC Alliance)



- Comment on GCI analysis for steady state solutions
  - Common way to reach a steady state solution for time dependent PDEs is to time advance the solution until in discrete form

$$\frac{\partial \phi}{\partial t} < \epsilon$$

  - To perform GCI analysis for spatial discretization errors, make sure that "non-steady-state" error ϵ is much smaller than spatial errors

- Comment on GCI analysis for unsteady solutions
  - superposition of time and spatial errors
  - GCI analysis as presented can deal with one type of error at a time only
  - for spatial error GCI
    - make temporal errors much smaller than spatial errors (very small Δt)
    - vary mesh spacing h only
  - for temporal error GCI
    - make spatial errors much smaller than temporal errors (very small h)
    - vary time step size only