Muddiest Points from Class 04/17

- *“Velocity correction term: Are we adding the same contribution of the correction factor to each u cells at the boundary or will the proportion depend on geometry and inlet conditions?”*
  - The same correction velocity is added to each outlet velocity
- *“I'm not sure I understand why, in slide 2, we sum the max values and not just all the values when determining the q-in and q-out.”*
  - because we want to separate inflow and outflow. The max statement takes 0 for outgoing velocities if we want to calculate the incoming volume flux and vice versa.
- *“I am a bit confused on how to solve two seperate equations together, like hyperbolic + parabolic problems. Can I assume it is as straightforward as writing FDEs and solving everything together?”*
  - Yes, see HW 10 solution
- *“GCI Analysis: In calculation of p = log[(f3-f2)/(f2-f1)]/log(r) if f1>f2<f3 or f1<f2>f3 we'll get complex value of p. So, in such cases how should we perform GCI Analysis? e.g. HW 10 - MAE 561 Solution, M = 80,160,320 case.”*
  - use absolute values for the difference in the numerator and denominator of the p formula, i.e., $p = \dfrac{\log|f_3 - f_2|}{\log|f_2 - f_1|}$
- *“Do you have any tips on how to code the solution for the Burger's equation in Matlab? The indexes and X and Y directions really messed with me while I was trying to solve HW10. I wasn't able to finish the assignment because of matrix indexing errors.”*
  - Yes: DO NOT cater to the linear algebra nature of Matlab. Forget that a 2D array is a matrix (Seriously, please just forget this!)
  - Code everything as in the slides: first index i for the x-direction, second index j for the y-direction, increasing i and j mean increasing x and y.
  - Plot using the transpose of the array providing vectors of x and y coordinates in the plot command.

• Muddiest Points from Class 04/17

- *"Just to be clear, we should first run a grid convergence test on different mesh sizes to find the optimum mesh size for the original chamber and then make modifications so as to bring a performance improvement in the chamber?"*
  – You should first find the answer for the original chamber (this includes solution verification)
  – Next, test potential improvements in the design (this will likely require fast turnaround so use coarse meshes)
  – Settle on an improved design and then verify it by performing solution verification on it

- *"Is there a preferred method for the mass fraction equation for undergrad students?"*
  – yes, the MAE561 method: WENO5-TVD-RK3. However, Lax-Wendroff will do very nicely as well (see HW10 solution)

- *"Optimization: Any suggestions better than "guess and check" for those of us who don't know as much about fluid dynamics?"*
  – No, except instead of guess: use your engineering skills to propose an improved chamber design

- *"For the project plots do we need to put physical error bars or just show the GCI analysis that goes with it?"*
  – no need to do error bars on the R(t) plot, but the steady state answer must have %accuracy ("error bars") with full solution verification (GCI analysis)

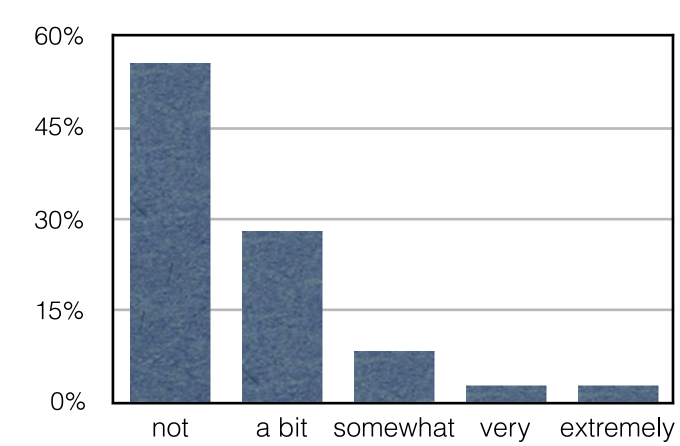
- *"Are we allowed to use MAC method for the final project or must we use fractional step?"*
  – I can devise no reason why you would want to code the MAC method from scratch if you can do fractional step, so no

- *"If I do all of the problems (bonus included) in the final project, how many core course outcomes will be possible?"*
  – six each in CCO2, 3, 4, 5

- *"To fill out CCO, does that mean I have to get 90/100 in this project?"*
  – the CCO threshold will be lower, no higher than 80%.

- *"Is the bonus HW available for graduate students?"*
  – Yes

- *"Can you upload the HW10 debug to the 'Old Assignments' folder when you get a chance?"*
  – Done



- 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, must make sure that “non-steady-state” error ϵ is much smaller than spatial errors

Verification & Validation

(Part II)

or

How can I trust my CFD results?

or

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

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

NNSA National Nuclear Security Administration

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.

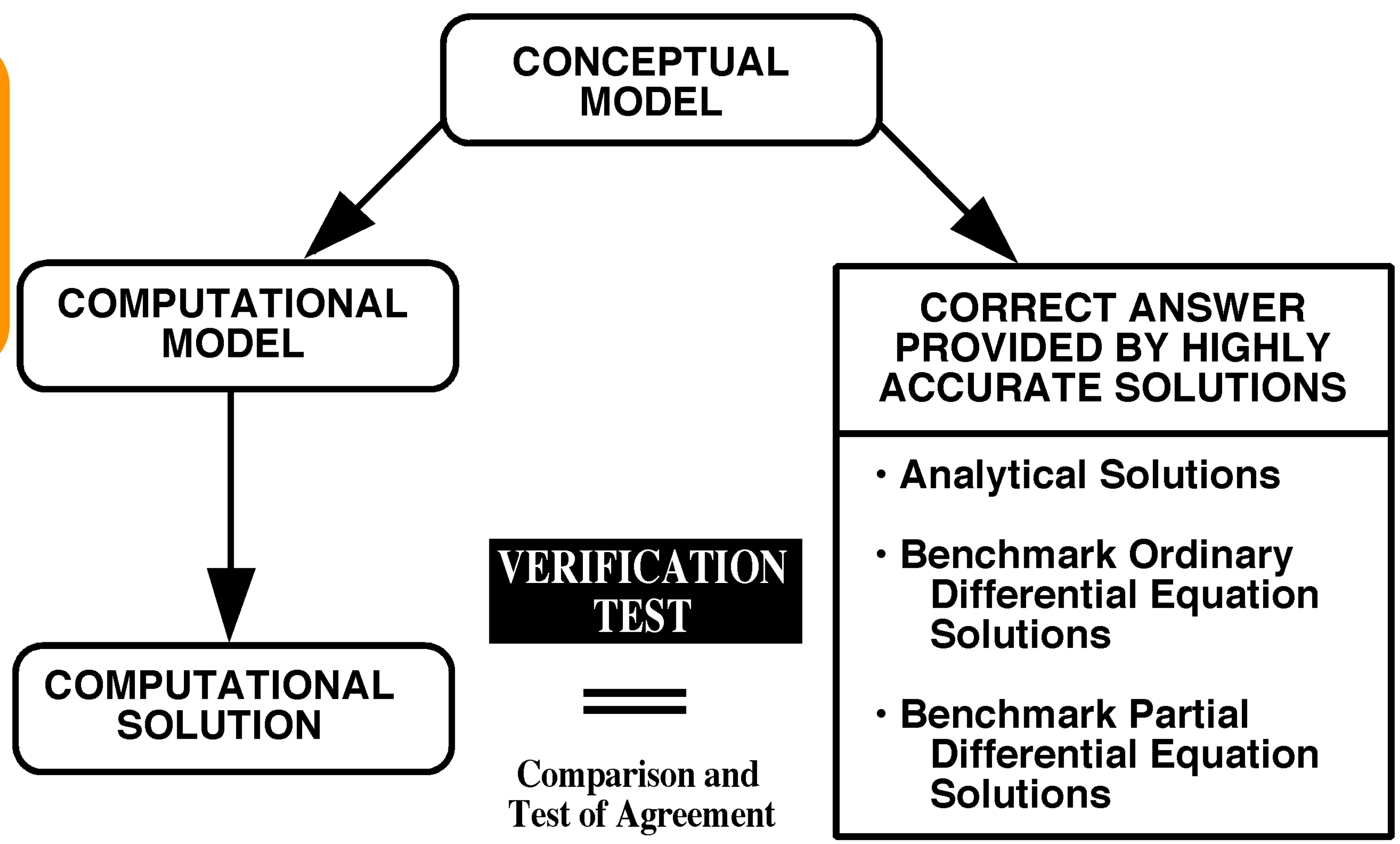
Sandia National Laboratories

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**



4

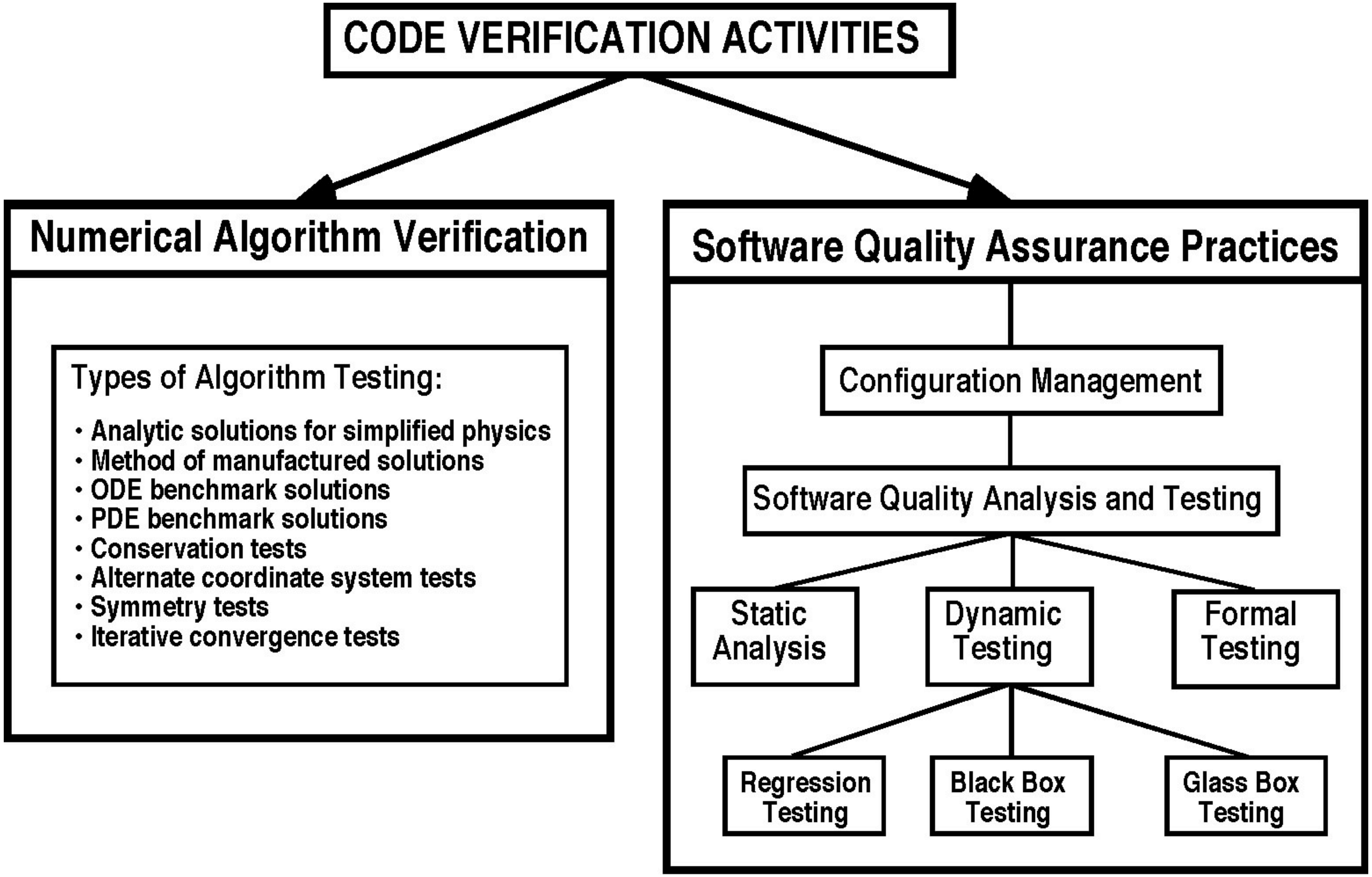
Two Types of Verification

- 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

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

Code Verification



9

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:

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

10

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

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 Manufactured Solutions (MMS) Example:

$$\frac{\partial u}{\partial t} = -u\frac{\partial u}{\partial x} + \nu\frac{\partial^2 u}{\partial x^2} \quad \Rightarrow \quad \mathcal{L}(u) = \frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} - \nu\frac{\partial^2 u}{\partial x^2} = 0 \quad \Rightarrow \quad \mathcal{L}(u) = 0$$

Idea: Modify PDE to: $\mathcal{L}(u) = Q(x,t)$

1. Select/define/**make up** an **exact** solution to the modified PDE

   for example: $U(x,t) = A + \sin(x + Ct)$

2. Find $Q(x,t)$ such that $U(x,t)$ is the exact solution!

   How? substitute $U(x,t)$ into the modified PDE: $Q(x,t) = \mathcal{L}(U) = \dfrac{\partial U}{\partial t} + U\dfrac{\partial U}{\partial x} - \nu\dfrac{\partial^2 U}{\partial x^2}$

   $$Q(x,t) = C\cos(x + Ct) + [A + \sin(x + Ct)]\cos(x + Ct) + \nu\sin(x + Ct)$$

3. Add $Q(x,t)$ as a source term to the CFD code

   $$\frac{\partial u}{\partial t} = -u\frac{\partial u}{\partial x} + \nu\frac{\partial^2 u}{\partial x^2} + Q(x,t)$$

4. Use exact solution as boundary conditions
   - Dirichlet: $u(0,t) = U(0,t) = A + \sin(Ct)$ $\qquad u(1,t) = U(1,t) = A + \sin(1 + Ct)$
   - Neuman: $\dfrac{\partial u}{\partial x}(1,t) = \cos(1 + Ct)$ etc.

Important:
- chosen solution need not be physical! example: $U(x,t) = \sin(t)e^x$
- chosen solution must involve all terms of the PDE! (sin,cos,tanh,exp ...)

Method of Manufactured Solutions

the following are examples from P. Brady & M. Herrmann, J. Comput. Phys. (2012).

- Finite volume temperature equation for multiphase flows

$$L(T) = \frac{1}{V_{cv}} \left[\int_{cv} \frac{\partial T}{\partial t} dV + \mathbf{u} \cdot \int_{cv} \nabla T dV - \frac{1}{(\rho c_p)_{cv}} \int_{cv} \nabla \cdot (k \nabla T) dV \right] = 0$$

with

$$(\rho c_p)_{cv} = \psi_{cv} (\rho c_p)_l + (1 - \psi_{cv}) (\rho c_p)_g \qquad k = k_g + (k_l - k_g) H(G)$$

- Source term for MMS is thus

$$L(T) = \frac{1}{V_{cv}} \left[\int_{cv} \frac{\partial T}{\partial t} dV + \mathbf{u} \cdot \int_{cv} \nabla T dV - \frac{1}{(\rho c_p)_{cv}} \left(\int_{cv, G=0} (k_l - k_g) \nabla G \cdot \nabla T dS \right.\right.$$
$$\left.\left. + \int_{cv, G>0} k_l \nabla^2 T dV + \int_{cv, G<0} k_g \nabla^2 T dV \right) \right]$$

- solve source term analytically, or with AMR marching tets

Example 1: 3D Unsteady Single Phase

- Material properties:

$$k_+ = k_- = 0.5, \quad c_{p_+} = c_{p_-} = 10.0, \quad \rho_+ = \rho_- = 5.0 + 1.5T$$

- Velocity field

$$(U, V, W) = (\cos(\pi x), \cos(\pi y), \cos(\pi z))/75$$

- Manufactured solution:

$$T_2(\boldsymbol{x}) = 10^{-4} \exp(2x) \cos(y) z^3 \sin(t) + 10 \cos^2(t)$$

- Analytical source term:

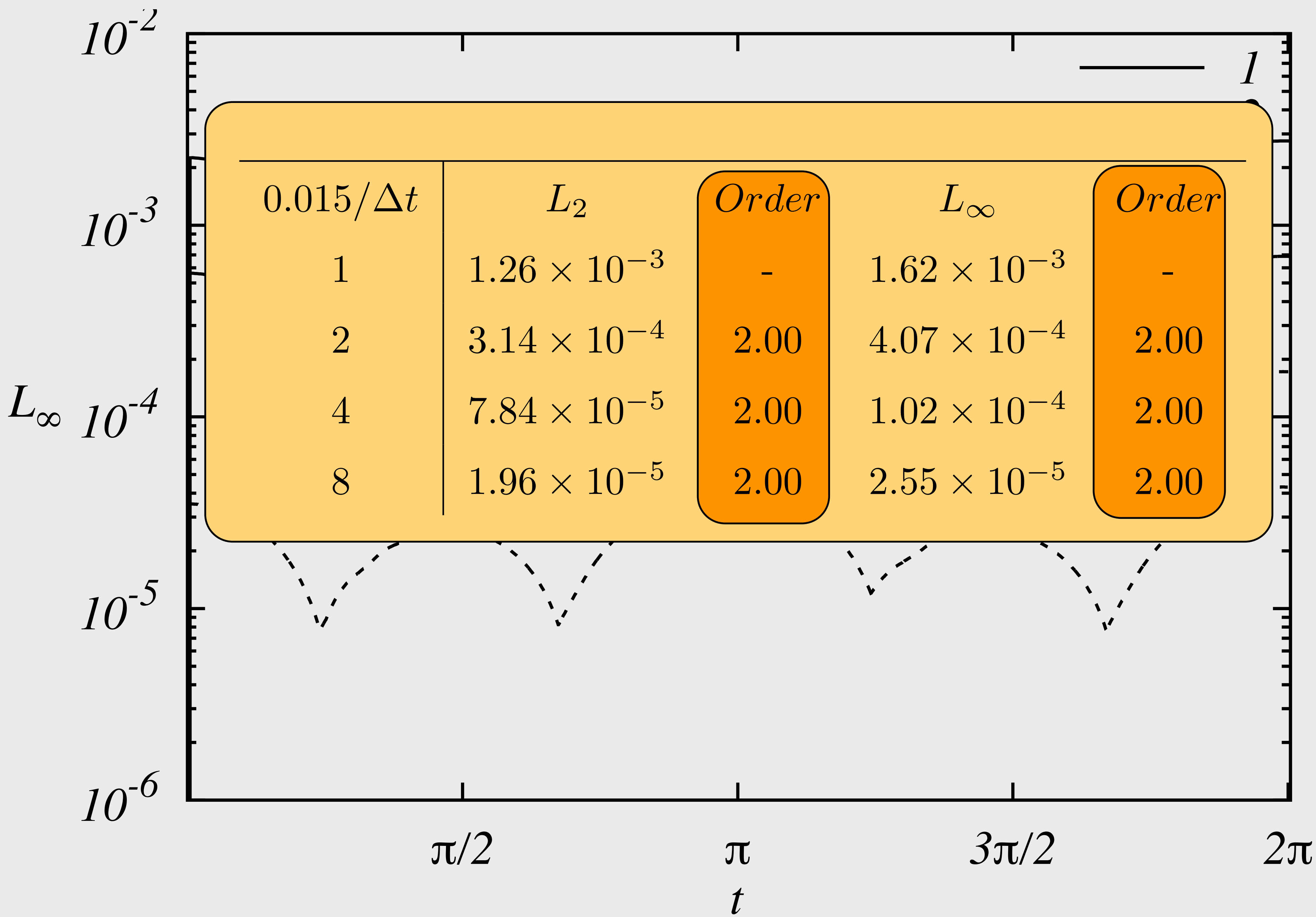
$$L_t = \int_{cv} \frac{\partial T_2}{\partial t} dV = 10^{-4} \frac{\exp(2x_2) - \exp(2x_1)}{2} (\sin(y_2) - \sin(y_1)) \frac{z_2^4 - z_1^4}{4} \cos(t) - 20 \cos(t) \sin(t) V_{cv}$$

$$L_c = \boldsymbol{u} \cdot \int_{cv} \nabla T_2 dV = 10^{-4} \frac{\exp(2x_2) - \exp(2x_1)}{2} \sin(t) \Big[U(\sin(y_2) - \sin(y_1)) \frac{z_2^4 - z_1^4}{2} + V(\cos(y_2) - \cos(y_1)) \frac{z_2^4 - z_1^4}{4} + W(\sin(y_2) - \sin(y_1))(z_2^3 - z_1^3) \Big]$$

$$L_{p/m} = \int_{cv} k_{+/-} \nabla^2 T_2 dV = 0.75 \times 10^{-4} k_+ \sin(t) [\exp(2x_2) - \exp(2x_1)] [\sin(y_2) - \sin(y_1)] [z_2^2 - z_1^2] \left[\frac{z_2^2 + z_1^2}{2} + 2 \right]$$

Example 1: 3D Unsteady Single Phase

- Infinity norm of error under Δt refinement and $N_{x,y,z} = 25$ for 2nd-order Adams-Bashforth



$0.015/\Delta t$	L_2	*Order*	L_∞	*Order*
1	1.26×10^{-3}	-	1.62×10^{-3}	-
2	3.14×10^{-4}	2.00	4.07×10^{-4}	2.00
4	7.84×10^{-5}	2.00	1.02×10^{-4}	2.00
8	1.96×10^{-5}	2.00	2.55×10^{-5}	2.00

Example 2: 3D Steady Single Phase

- Material properties:

$$k_+ = k_- = 0.5, \quad c_{p_+} = c_{p_-} = 10.0, \quad \rho_+ = \rho_- = 5.0 + 1.5T$$

- Manufactured solution:

$$T_3(\boldsymbol{x}) = \exp(2x)\cos(y)z^3$$

- Leading order error of modified equation (convective: 3rd order QUICK, diffusive: 2nd order central)

$$LE \propto \Delta x^2 \left(1 - \frac{U}{\alpha}\Delta x\right)$$

➡ formal order of accuracy is a function of U/α

- Velocity field 1 with $U/\alpha \gg 1 \Rightarrow LE \propto \Delta x^3$

$$(U, V, W) = 10(\cos(\pi x), \cos(\pi y), \cos(\pi z))$$

- Velocity field 2 with $U/\alpha \ll 1 \Rightarrow LE \propto \Delta x^2$

$$(U, V, W) = (\cos(\pi x), \cos(\pi y), \cos(\pi z))/75$$

Example 2: 3D Steady Single Phase

Table 6: Test 3: $(U, V, W) = 10(\cos(\pi x), \cos(\pi y), \cos(\pi z))$

N_x	L_2	*Order*	L_∞	*Order*
25	$1.45 \cdot 10^{-5}$	-	$9.03 \cdot 10^{-5}$	-
50	$1.70 \cdot 10^{-6}$	3.09	$1.13 \cdot 10^{-5}$	3.00
100	$2.00 \cdot 10^{-7}$	3.09	$1.36 \cdot 10^{-6}$	3.05
200	$2.28 \cdot 10^{-8}$	3.13	$1.59 \cdot 10^{-7}$	3.10

✓

Table 7: Test 4: $(U, V, W) = (\cos(\pi x), \cos(\pi y), \cos(\pi z))/75$

N_x	L_2	*Order*	L_∞	*Order*
25	3.13×10^{-5}	-	9.18×10^{-5}	-
50	7.44×10^{-6}	2.07	2.24×10^{-5}	2.03
100	1.81×10^{-6}	2.04	5.53×10^{-6}	2.02
200	4.45×10^{-7}	2.02	1.37×10^{-6}	2.01

✓

Example 3: Bug Finding

- Bug in diffusive term:

$$\nabla \cdot (k \nabla T)_i \approx \frac{\frac{k_{i+1} + k_i}{2} \frac{T_{i+1} - T_i}{\Delta x} - \frac{k_i + k_{i+1}}{2} \frac{T_i - T_{i-1}}{\Delta x}}{\Delta x}$$

Example 3: Bug Finding

- Bug in diffusive term:

$$\nabla \cdot (k\nabla T)_i \approx \frac{\frac{k_{i+1}+k_i}{2}\frac{T_{i+1}-T_i}{\Delta x} - \frac{k_i+k_{i+1}}{2}\frac{T_i-T_{i-1}}{\Delta x}}{\Delta x}$$

- Manufactured solution:

$$T_4 = \exp(2x)\cos(y)$$

N_x	L_2	$Order$	L_∞	$Order$
25	8.98×10^{-2}	-	3.90×10^{-1}	-
50	9.06×10^{-2}	-0.01	3.98×10^{-1}	-0.0319
100	8.79×10^{-2}	0.04	3.80×10^{-1}	0.0697

✗

Example 3: Bug Finding

- Bug in diffusive term:

$$\nabla \cdot (k \nabla T)_i \approx \frac{\frac{k_{i+1} + k_i}{2} \frac{T_{i+1} - T_i}{\Delta x} - \frac{k_i + k_{i+1}}{2} \frac{T_i - T_{i-1}}{\Delta x}}{\Delta x}$$

- Manufactured solution + all velocities = 0 (turn of convective terms):

$$T_4 = \exp(2x) \cos(y)$$

N_x	L_2	$Order$	L_∞	$Order$
25	2.29×10^{-1}	-	7.64×10^{-1}	-
50	2.33×10^{-1}	-0.02	7.93×10^{-1}	-0.06
100	2.24×10^{-1}	0.06	7.59×10^{-1}	0.06

Example 3: Bug Finding

- Bug in diffusive term:

$$\nabla \cdot (k \nabla T)_i \approx \frac{\frac{k_{i+1} + k_i}{2} \frac{T_{i+1} - T_i}{\Delta x} - \frac{k_i + k_{i+1}}{2} \frac{T_i - T_{i-1}}{\Delta x}}{\Delta x}$$

- Simpler manufactured solution + all velocities = 0 (turn of convective & x-diffusion terms):

$$T = \cos(y)$$

N_x	L_2	$Order$	L_∞	$Order$
25	4.92×10^{-4}	-	4.03×10^{-3}	-
50	1.63×10^{-4}	1.59	2.35×10^{-3}	0.78
100	8.30×10^{-5}	0.97	1.16×10^{-3}	1.02

$\Rightarrow$ Error must be in x-diffusion term

- Please fill out the course evaluation forms online!
- Questions about the Final Project?