Lecture 22 – Verification and Validation

Prof. Brendan Kochunas

NERS/ENGR 570 - Methods and Practice of Scientific Computing (F22)

COLLEGE OF ENGINEERING
NUCLEAR ENGINEERING & RADIOLOGICAL SCIENCES
UNIVERSITY OF MICHIGAN

Outline

- Motivation & Review of Testing
- Fundamental Concepts and Definitions
- Order of Accuracy
- Code Verification
- Modeling Errors

Learning Objectives: By the end of Today's Lecture you should be able to

- (*Knowledge*) explain the difference between verification and validation
- (*Knowledge*) perform order of accuracy study
- (*Knowledge*) qualitatively order sources of numerical errors in simulations

Motivation – Why V&V?

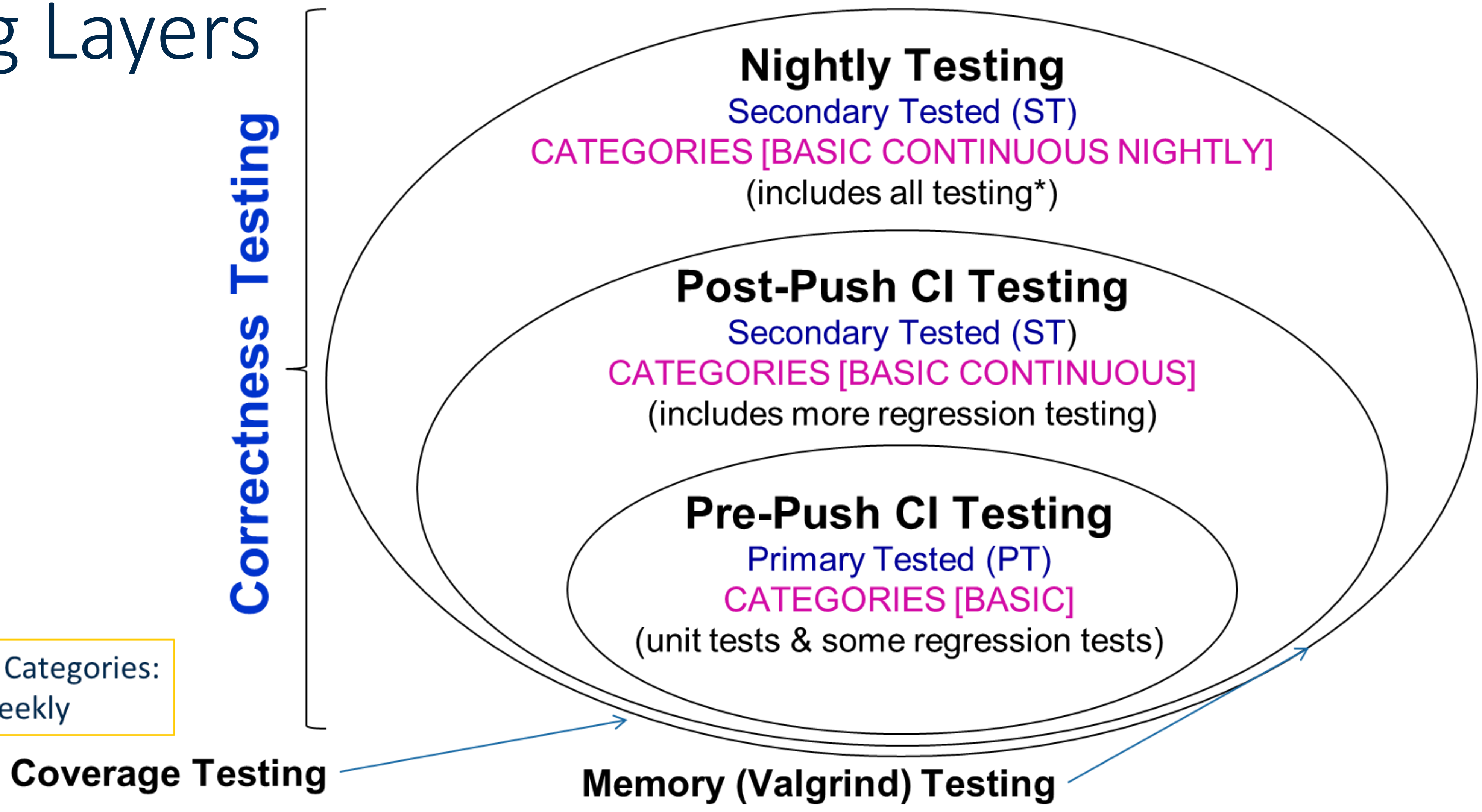
Review of Testing

Ross's Taxonomy of Testing

A Taxonomy of Testing

- Testing is the backbone of software quality assurance (SQA).
- Types of testing
  - *Unit Testing* – Test individual units of program *in isolation*
    - Should run very fast: < 1 second (a couple seconds is ok)
  - *Integral Testing* – Testing program components together
    - Should run fast: < 1 minute (a couple minutes is ok)
  - *Regression Testing* – Test whole program for changes in program output
    - Should run fast: < 1 minute (a couple minutes is ok)
  - *Verification Testing* – Test that you are “doing things right”
    - Can happen at unit or integral or regression level. Comparison analytic solutions or manufactured solutions.
  - *Validation Testing* – Whole program testing “doing the right thing”; simulating reality, comparison to experiment.
    - May be long running: minutes to hours
  - *Memory Testing* – Expensive testing that does detailed memory simulations to detect errors (valgrind)
  - *Coverage Testing* – Figure out how much of your source code is actually covered by testing
  - *Portability Testing* – test on different platforms and with different compilers
- Other types of testing exist

Testing Layers



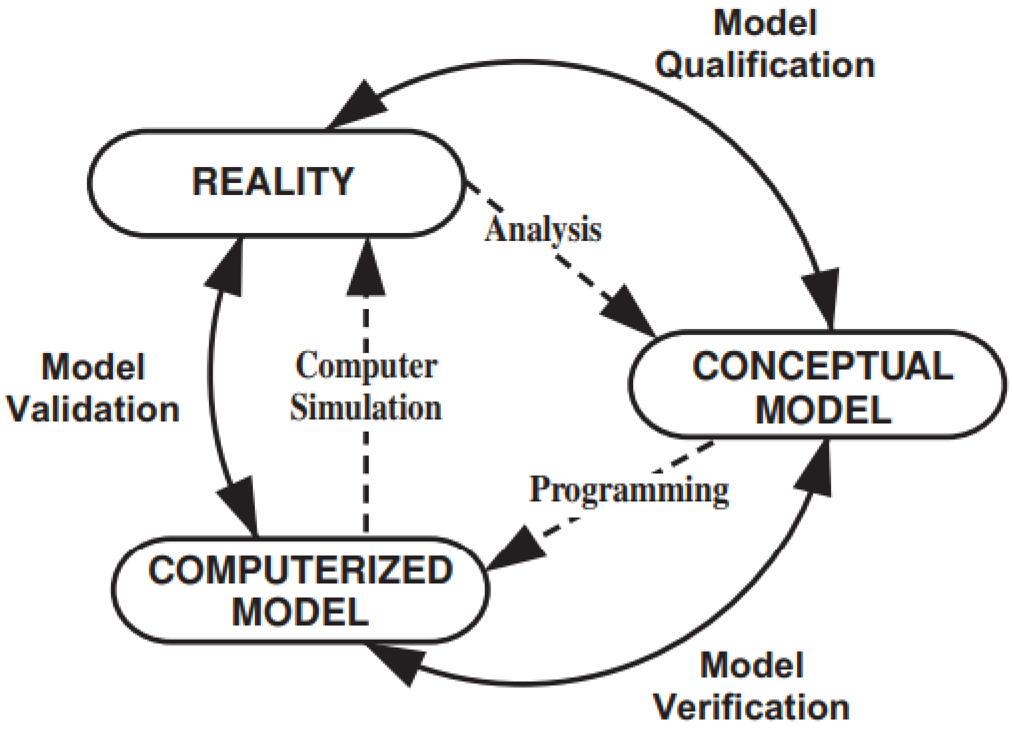
*Additional Categories:
Heavy or Weekly

Terminology

Lets get pedantic

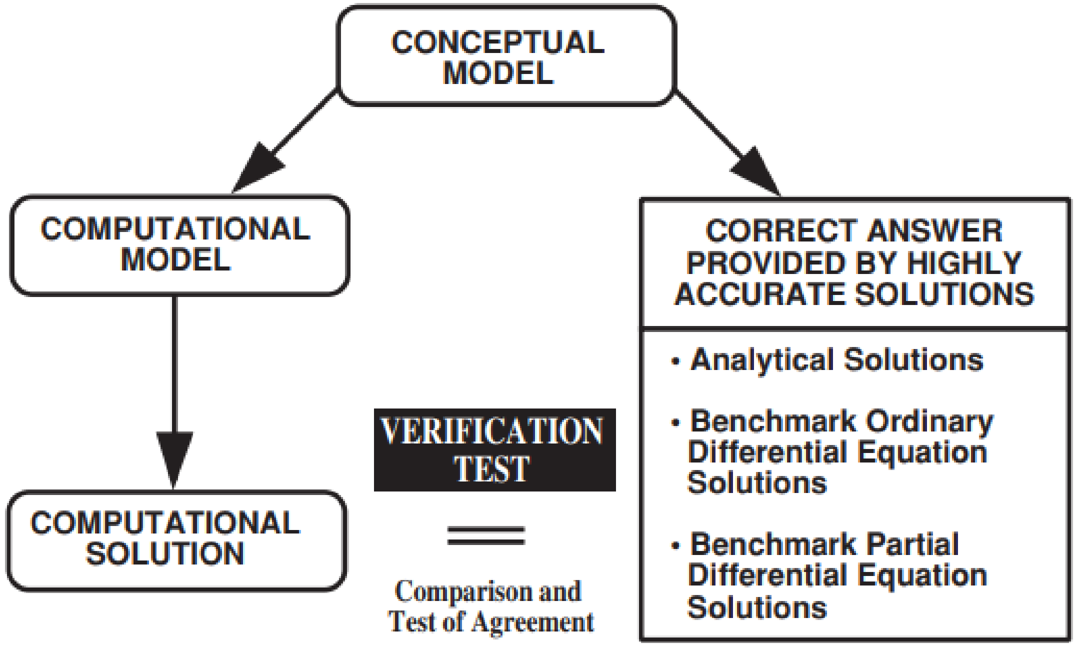
Early Definitions of Verification and Validation

- *Model verification*: substantiation that a computerized model represents a conceptual model **within specified limits of accuracy**.
- *Model validation*: substantiation that a computerized model within its domain of applicability possesses a **satisfactory range of accuracy** consistent with the **intended application** of the model.

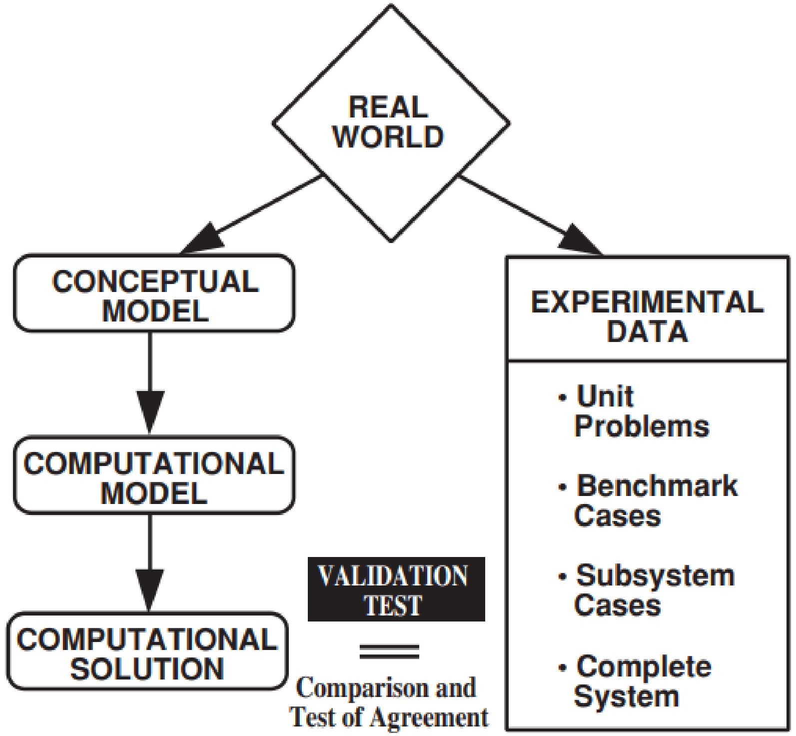


American Institute of Aeronautics & Astronautics

Verification

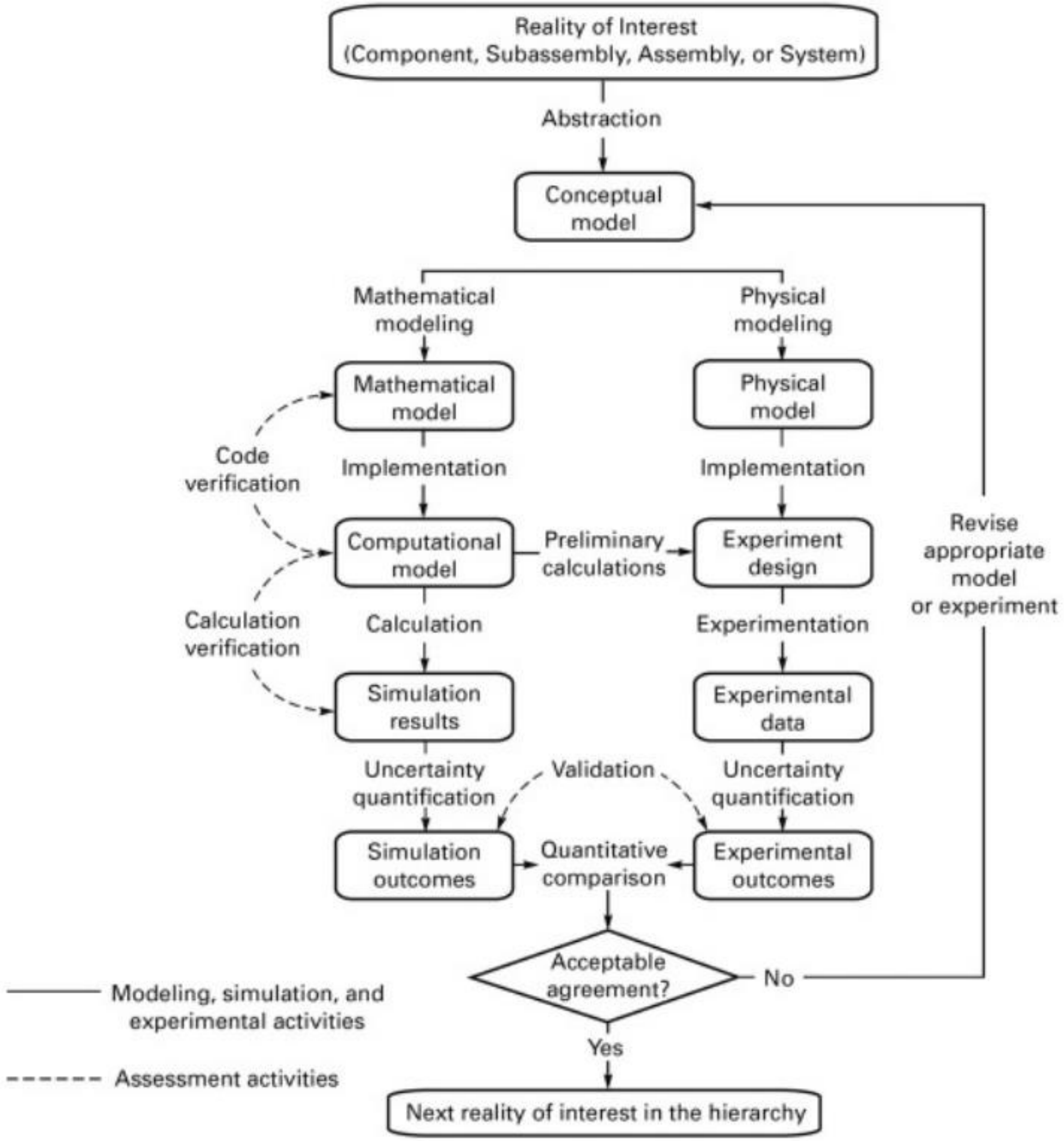


Validation



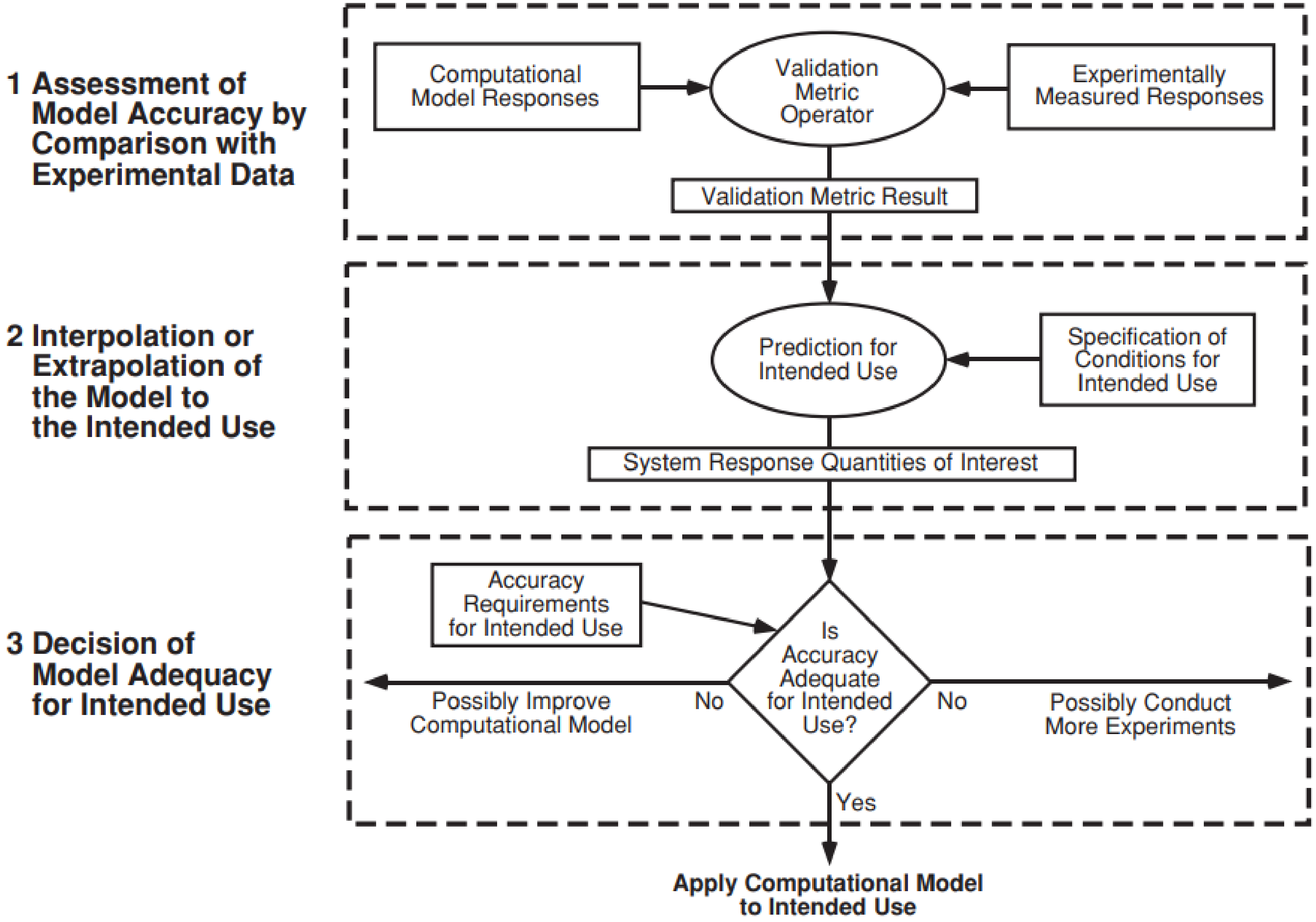
ASME

- *Code verification*: the process of determining that the numerical algorithms are correctly implemented in the computer code and of identifying errors in the software.
- *Solution verification*: the process of determining the **correctness of the input data**, the numerical accuracy of the solution obtained, and the **correctness of the output data** for a particular simulation.



Aspects of Validation

- Quantification of the accuracy of the computational model results by comparing the computed system response quantities (SRQs) of interest with experimentally measured SRQs
- Use of the computational model to make predictions, in the sense of interpolation or extrapolation of the model, for conditions corresponding to the model's domain of intended use.
- Determination of whether the estimated accuracy of the computational model results satisfies the accuracy requirements specified for the SRQs of interest.

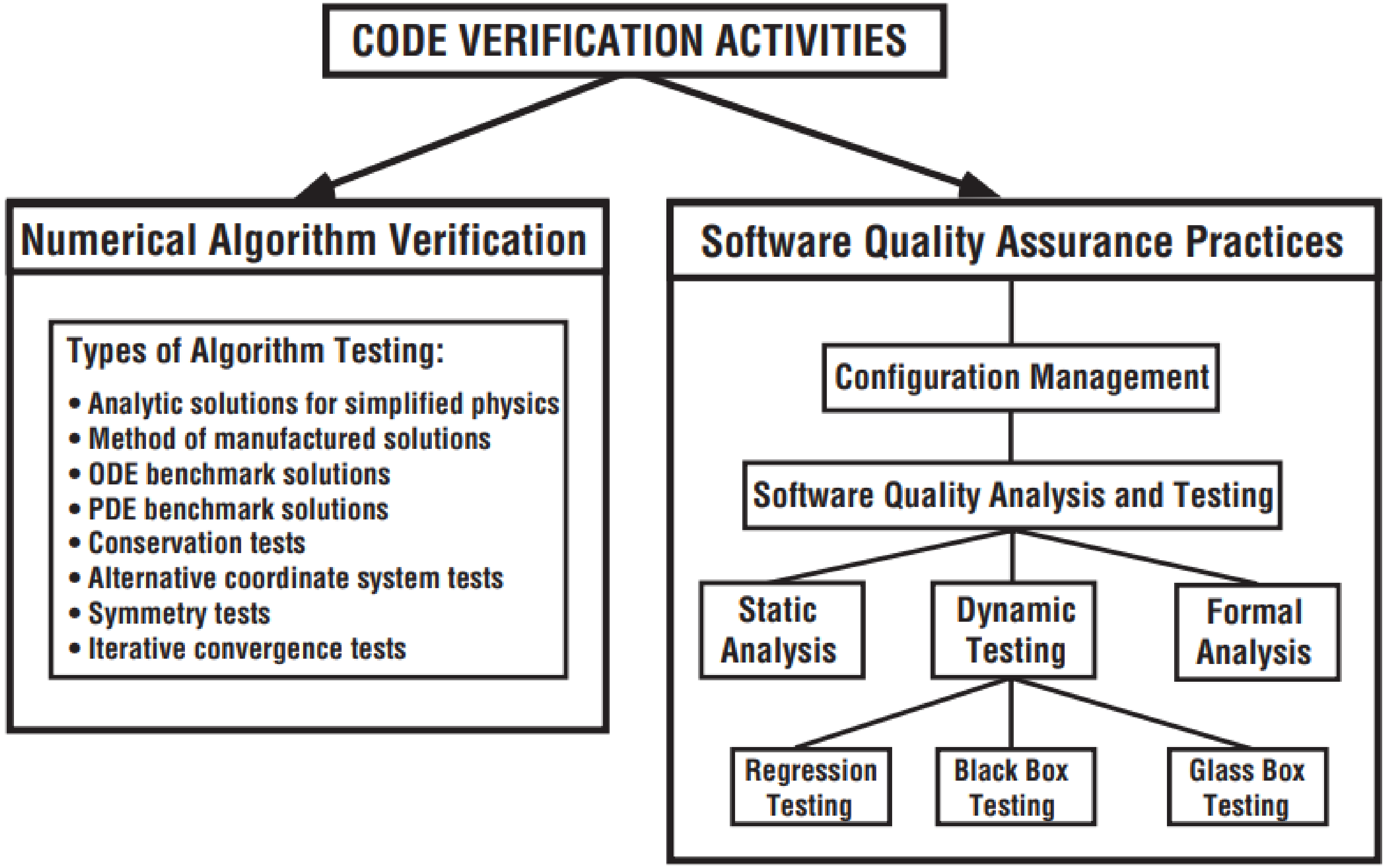


Case Study of Verification & Validation

Code Verification

How do you prove that your program is a faithful representation of the original mathematical model?

V&V In the Context of SQA

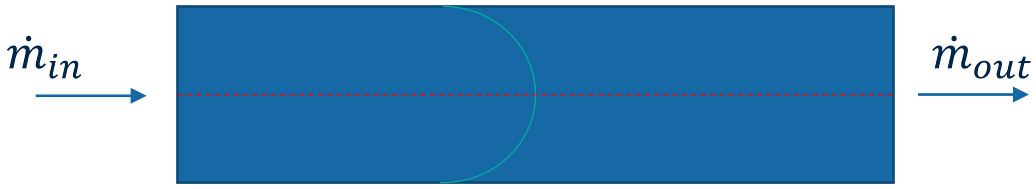


Types of Code Verification Tests

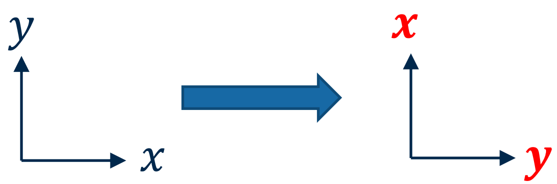
- Simple Tests
- Manufactured Solutions
- Code-to-Code Verification
- Discretization Error / Convergence Tests
  - Require some reference analytic solution
- Order of Accuracy Tests
- Iterative Convergence

Simple Test Example

- Symmetry
  - Define a problem with geometry and boundary conditions symmetric about a plane
  - Can also look at periodic problems



- Rotational Invariance
  - Similar to the above except change coordinates

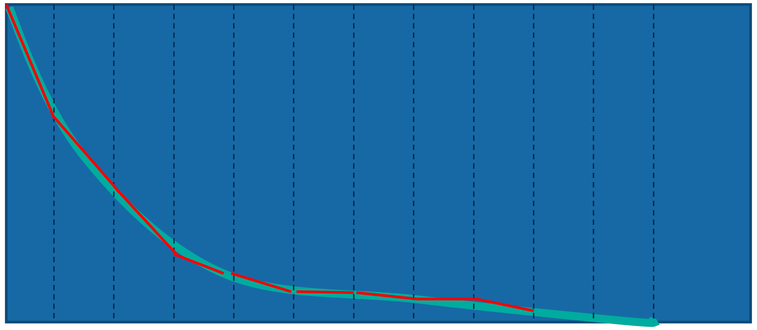


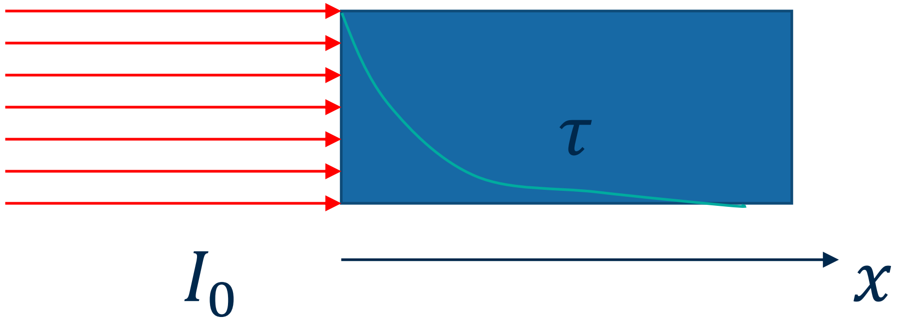
- Conservation (perform global integration)
  - of energy in heat transfer
  - of mass or momentum in fluid flow

$$\dot{m}_{in} = \dot{m}_{out}$$

Discretization Error Tests

- Compare the numerical solution to an exact solution
- Quantitative assessment of code output using a single mesh
- Example: beam attenuation





$$I(x) = I_0 \exp(-\tau x)$$

Order of Accuracy Tests

Procedure for Order-of-Accuracy Tests

Establishing Formal Order of Accuracy

Computing Observed Order of Accuracy

Computing Observed Order of Accuracy

Plotting Observed Order of Accuracy

Making the solution more “right”

A qualitative assessment of Numerical Errors

Modeling Errors

Closing Remarks

- There is well defined theory and procedures for many methods in scientific computing to assess:
  - the code verification,
  - the solution verification,
  - and validation of the model

- Many researchers are beginning to focus on topics of artificial intelligence, machine learning, or other “learning” based methods.
  - Significant open question: **How does one verify or validate these methods?**