



Calibration contra Validation: Characterization and Consequences

Timothy Trucano and Laura Swiler

Optimization and Uncertainty Estimation Department, Org 9211

Sandia National Laboratories

Albuquerque, NM 87185

Takeru Igusa

Johns Hopkins University

Foundations for Verification, Validation and
Accreditation in the 21st Century

October 13-15, 2004

Phoenix, Arizona

Phone: 844-8812, FAX: 844-0918

Email: tgtruca@sandia.gov





Outline of talk.

- **Definitions**
- **Illustration (Double Mach Reflection)**
- **Benchmark formalisms**
- **Conclusions**



Perspective

- **Day-to-day challenges of the Sandia ASC V&V program.**
- **Computer codes that solve systems of partial differential equations for which quantitative solutions cannot generally be provided in other ways.**
- **High-consequence applications (being wrong has unpleasant consequences).**
- **“Validation” is synonymous with “experimental validation.”**
- **Not doing human-interaction modeling**



ASC V&V Definitions

- **Verification (ASC)** is the process of confirming that a computer code correctly implements the algorithms that were intended.
- **Validation (ASC)** is the process of confirming that the predictions of a code adequately represent measured physical phenomena.
- **Reference: United States Department of Energy, “Advanced Simulation and Computing Program Plan.” Sandia National Laboratories Fiscal Year 2005; Report SAND 2004-4607PP.**



Please contrast this with the AIAA definitions:

- **Verification (AIAA)** is the process of determining that a model implementation accurately represents the developer's conceptual description of the model and the solution to the model.
- **Validation (AIAA)** is 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.
- **Reference: AIAA, “Guide for the Verification and Validation of Computational Fluid Dynamics Simulations.” American Institute of Aeronautics and Astronautics, AIAA-G-077-1998, Reston, VA, 1998.**



Two other options that are perfectly appropriate:

- **IEEE:**
 - **Verification**: Software requirements are implemented correctly.
 - **Validation**: Software requirements are correct.
- **Pragmatic computational physicist (Roache):**
 - **Verification**: Equations are solved correctly.
 - **Validation**: Equations are correct.



Emphasis:

- **We don't intend to say anything that disagrees with any of these definitions.**
- **ASC pays the bills ...**

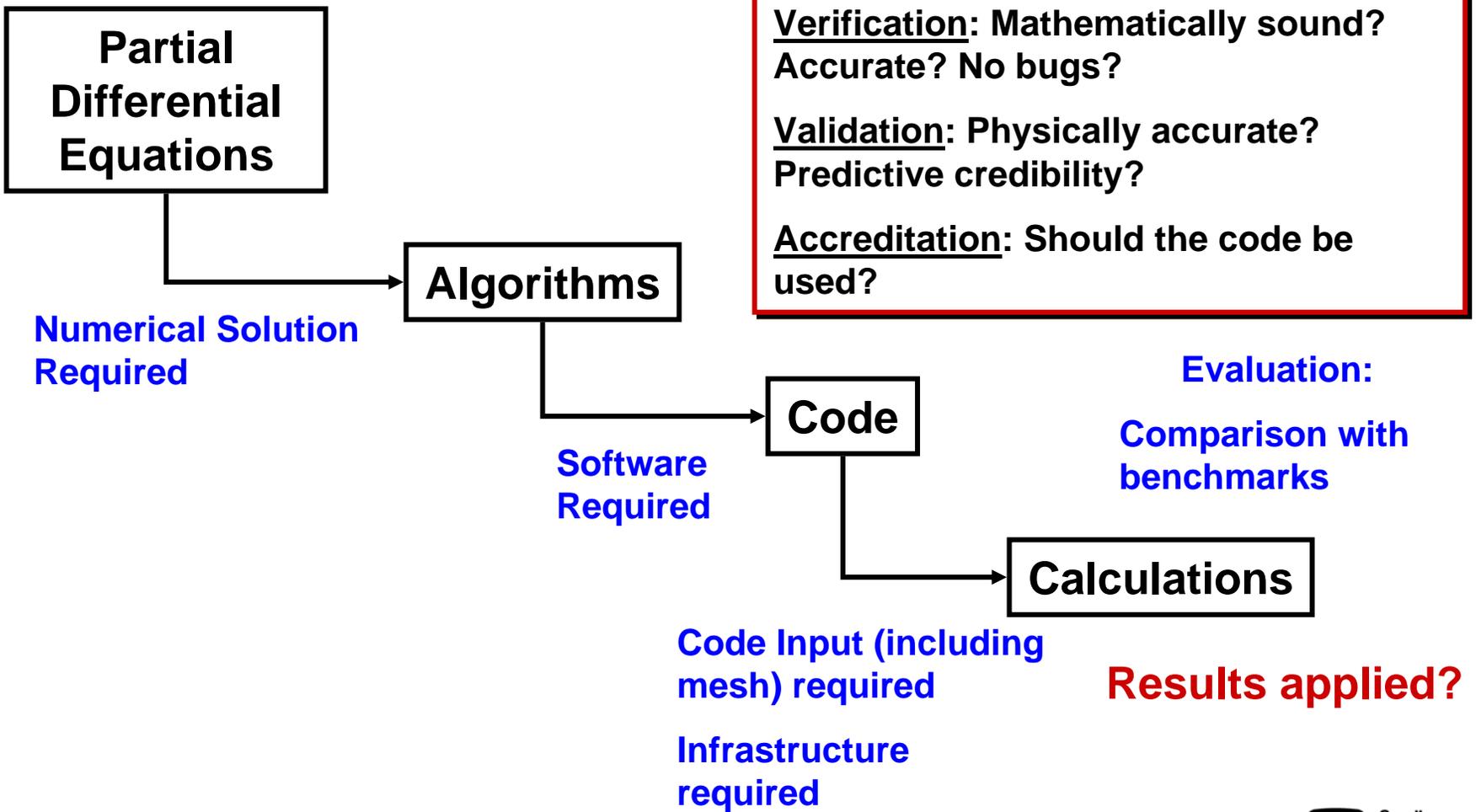
.... & A

- The ASC program does not define “Accreditation” nor is the term formally recognized with respect to the “codes” referred to in the V&V definitions.
- It is safe to assume that if the term was defined by ASC, the definition would look approximately like:

Accreditation is the process that determines whether or not a code will be applied to simulate (“solve,” “answer”) a specified problem and the answer used.

- The code is said to be “accredited” for that specific application.
- Accreditation will not be discussed in this presentation.

Codes and V&V





Benchmarks are required in V&V

- **Benchmark**: a choice of information for purposes of performing verification, validation (and calibration) via comparison with specific calculations.
 - Benchmarks are believed to be appropriate for the task of evaluation, that is drawing conclusions about verification and validation based on these comparisons.
 - V&V conclusions drawn from comparisons of code calculations with benchmarks centers on the quantitative meaning of these comparisons and subsequent inference about predictive (explanatory) credibility.



Benchmarks are required in calibration

- **Calibration**: the process of improving the agreement of a code calculation or calculations with respect to a chosen set of benchmarks through the adjustment of parameters implemented in the code.
 - Benchmarks do not have an evaluation role in calibration, but are assumed to be appropriate for the calibration task.
- **Calibration** is neither verification nor validation.



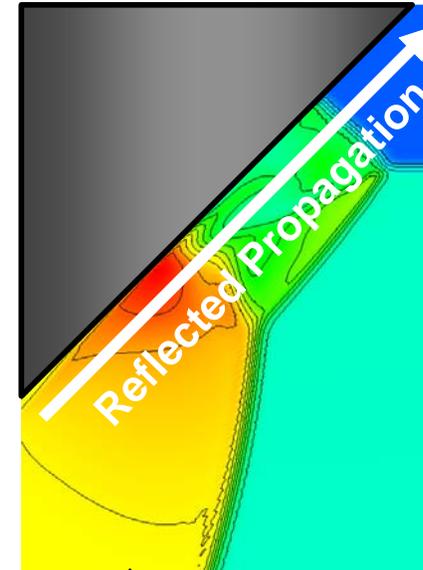
These thoughts are worth repeating

- Verification, validation and calibration involve uncertainty (variability and lack of knowledge).
- It is generally believed that validation is a harder problem than verification because of associated philosophical problems, as well as practical problems.
 - One of Hilbert’s problems is to “axiomatize” physics, which might offer a route to proving correctness of equations. This problem remains unsolved.
 - This does not mean that verification is easy; if it is a mathematical problem it should be held to that standard of rigor.
- Implications of validation depend upon verification.
- Implications of calibration depend upon verification and validation.

Example: Double Mach Reflection

- Shock in γ -law gas obliquely reflected from ideal wedge.
- Used as “qualitative” validation problem for the shock wave physics code ALEGRA: can the transition from regular to Mach to double Mach reflection be simulated?
 - We found qualitative agreement with data, but not quantitative.
 - Quantitative features were highly dependent on mesh resolution.
- See Chen and Trucano, “ALEGRA Validation Studies for Regular, Mach and Double Mach Reflection in Gas Dynamics,” SAND2002-2240.

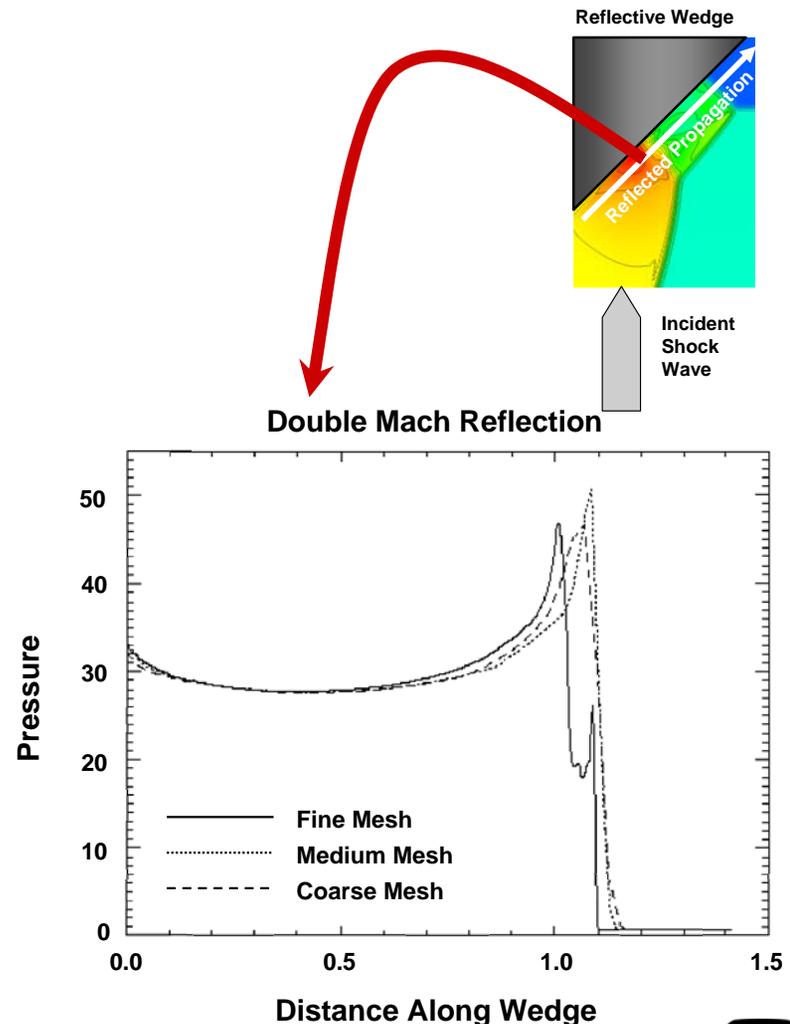
Reflective Wedge



Incident Shock Wave

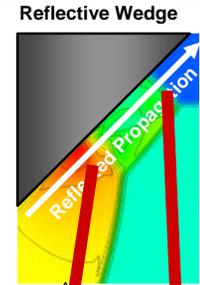
Verification: Are the equations solved correctly?

- This is mathematics:
 - Are the algorithms mathematically correct?
 - Are the algorithms implemented correctly in the code (no bugs)?
 - Do calculations converge to the mathematically correct solutions as the mesh is refined?
 - What is the numerical error for a given calculation?
- Calibration does not answer any of these questions.
- Calibration relies upon the answers to these questions.



Validation: Are the equations correct?

$$\frac{\Delta p_r}{\Delta p_{inc}} \equiv \frac{\text{Reflected Pressure} - \text{Ambient Pressure}}{\text{Incident Pressure} - \text{Ambient Pressure}}$$

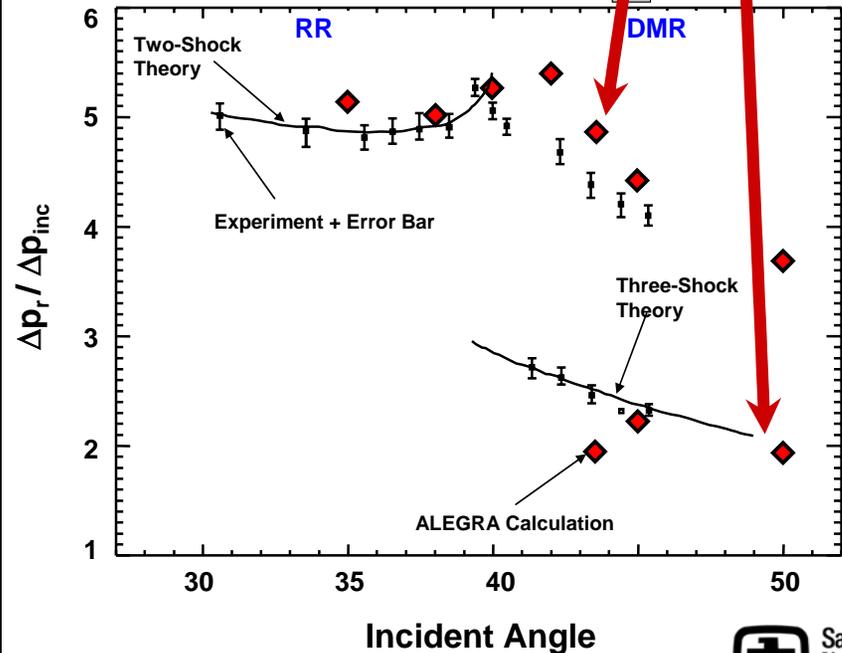


- This is physics:

- Experiment error bars mean what?
- What is the numerical accuracy of the calculations?
- Is the comparison good, bad, or indifferent? In what context?
- Why did we choose this means to compare the data and the calculation? Is there something better? *
- Why did we choose this problem to begin with? *
- What does the work rest on (such as previous knowledge)? *
- Where is the work going (e.g. what next)? *

- Calibration does not answer any of these questions.

- Calibration relies upon the answers to these questions.





To summarize: some operational considerations

- **Verification**: Are code bugs and/or numerical inaccuracies corrupting the comparison with experimental data?
- **Validation**: Are we performing the right calculations to compare with the right experiments in the right way to draw the right conclusions?
- **Validation requires verification**: computational errors in validation comparisons must be smaller than physical errors (experimental and physics in the code) to make these comparisons meaningful in the context of validation.
- Ask your favorite computational modelers what the numerical errors are in their calculations.
 - By the way, ask them to prove their answer. (After all, it IS a mathematics problem!!)
 - Adjusting the mesh to agree with experimental data is **calibration**, not **verification** or **validation**.
- **Verification has uncertainty, specifically lack of knowledge, if you don't know what the computational error is and can't prove that the code is bug free.**
- **Validation has uncertainty, both variability and lack of knowledge.**



Calibration considerations:

- **Performance of, and reliance upon, calibration depends upon verification and validation; it does not replace them.**
- **Calibration is therefore dangerous for high-consequence computational predictions to the degree that it replaces V&V.**
- **Confusion among verification, validation, and calibration **MUST BE AVOIDED** for high-consequence computing.**
- **Accreditation should theoretically rest only upon V&V; this seems to be unlikely in reality (for example, to the degree calibration replaces validation; to the degree risks can't be zeroed by perfect models; etc).**

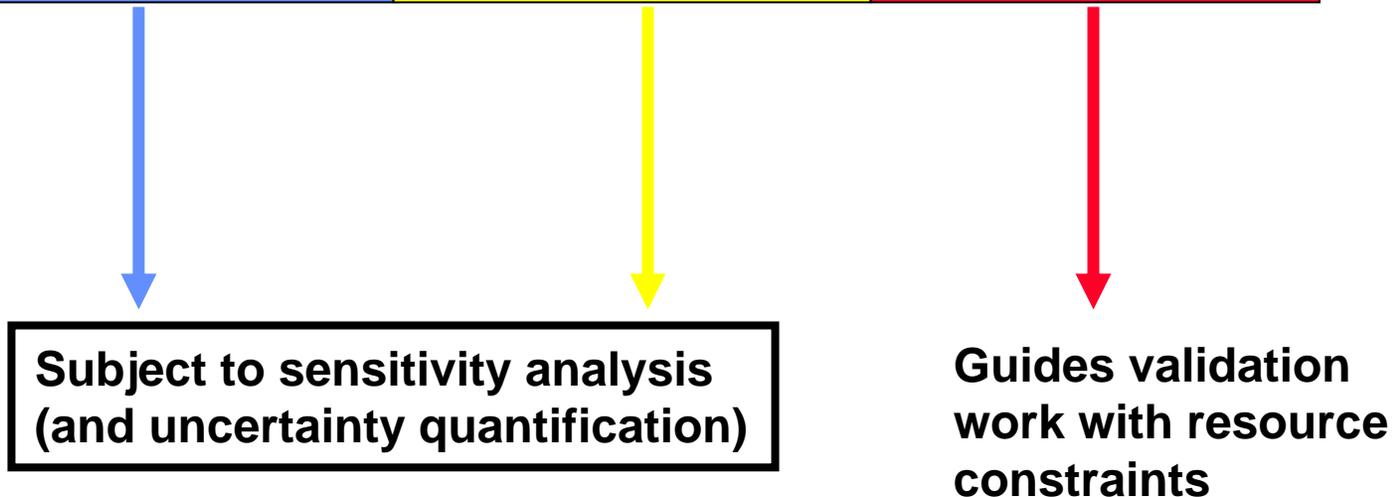


Further details next:

- **How do you do the right work?***
 - PIRT (and see the talk of Bill Oberkamp for more).
- **You don't know the answers (mathematical or physical) in the real application (otherwise why would you perform code calculations?).**
 - Benchmarks are necessary
 - Benchmarks are not likely to be sufficient.
- **How do you formalize “predictive credibility?”**
 - Baby steps: look at calibration and understand its dependence on verification and validation (CUU).

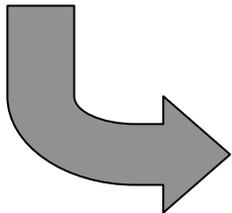
The Right Experiments*:

Phenomenology Identification and Ranking Table

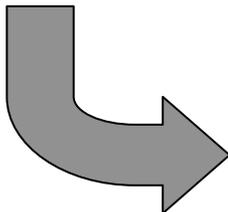


The PIRT is subject to iteration because sensitivity analysis (and uncertainty quantification) changes as work proceeds:

PIRT #1



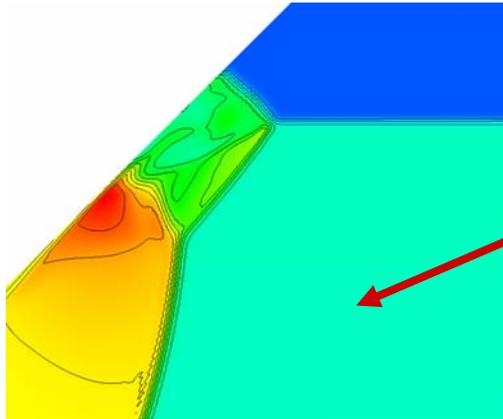
PIRT #2



PIRT #N



The right calculations:

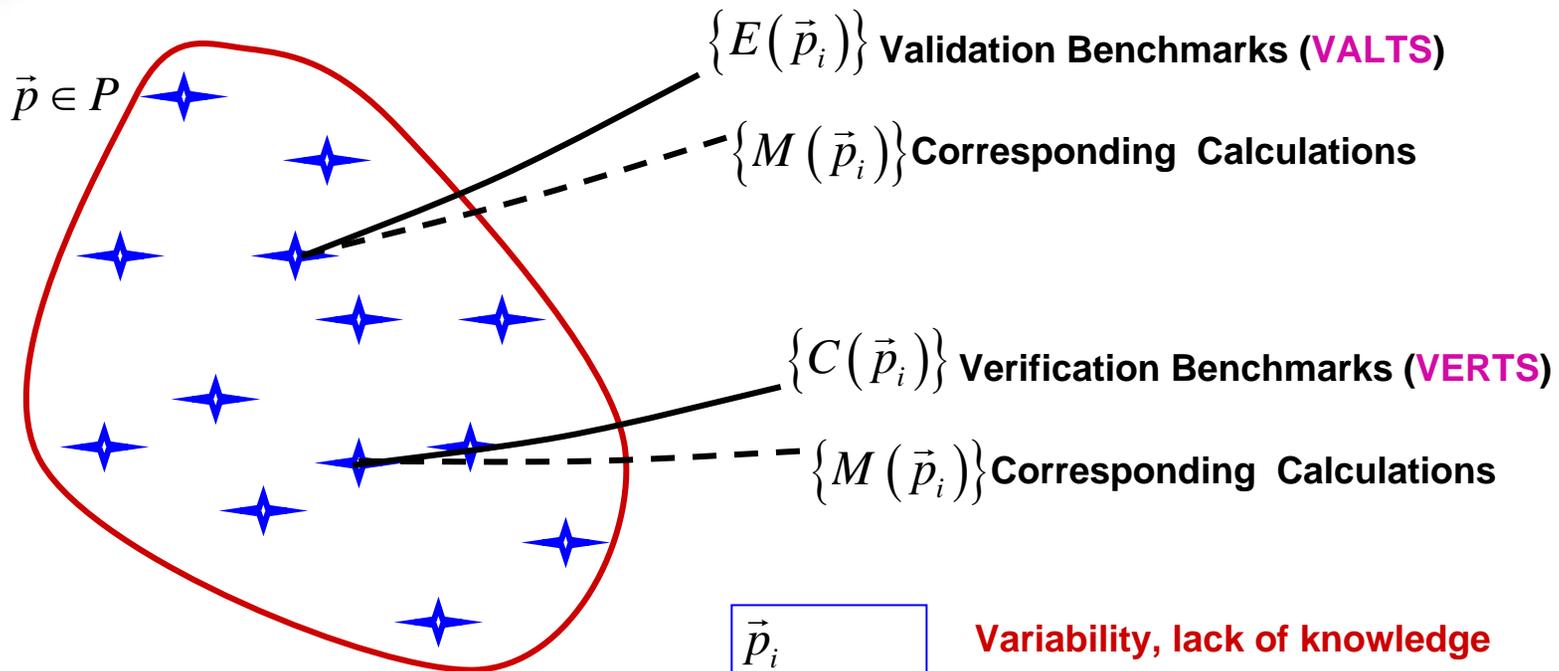


- Code bugs?
- Test what?
- Numerical performance (consistency, stability, convergence)?
- Numerical robustness?
- Calculations are sensitive to what?

The code: $M(\vec{p})$

- Multi-physics
- Multi-resolution
- \vec{p} is a (large) parameter including parameters required to specify physics, numerics, scenarios
 - The parameter vector \vec{p} is typically high-dimensional, especially if the grid specification is part of the parameter list
 - Verification centers on the numerics components of \vec{p} .
- Verification must be and is prioritized by the PIRT (verify what you are trying to validate).
- Sensitivity analysis is required.

Codes map parameters to outputs that can be compared with benchmarks



Uncertainty In →

- \vec{p}_i
- $\{E(\vec{p}_i)\}$
- $\{C(\vec{p}_i)\}$
- $\{M(\vec{p}_i)\}$
- $\{M(\vec{p}_i)\}$

Variability, lack of knowledge

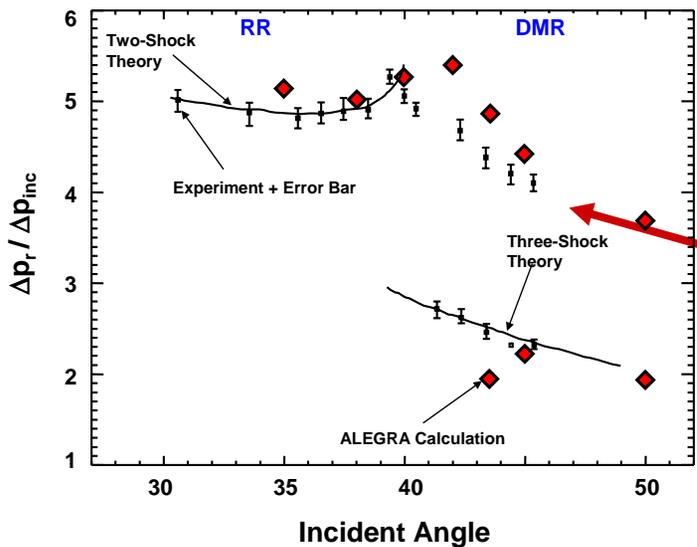
Variability, lack of knowledge

None

Lack of knowledge – numerics and software

Lack of knowledge - model

Comparing the right way: “Validation Metrics”



- Accurate calculations?
- Accurate experiments?
- Uncertainty accounted for in comparisons?
- Comparisons relevant?

- **Validation** compares code results $M(\vec{p}_1), \dots, M(\vec{p}_N)$ with experimental **benchmarks** $E(\vec{p}_1), \dots, E(\vec{p}_N)$ for a directed choice of \vec{p}_i

- Validation metrics quantify the difference, accounting for uncertainty.

$$D\{M(\vec{p}_i), E(\vec{p}_i)\}$$

- The parameters \vec{p}_i vary over the physics, not over the numerics.

- It is often the case that $N \ll \dim(\vec{p})$, so sensitivity analysis is very important for best leveraging limited experiments.

- Note that a simple definition of prediction is now any $M(\vec{p})$ for which $\vec{p} \neq \vec{p}_i, i = 1, \dots, N$; such values may be inputs into important decisions.

Think of validation metrics as metrics in this presentation.

- In the case of the Mach reflection benchmark, the metric may be defined by a norm on the sequences:

$$\left\{ \frac{\Delta p_r^E}{\Delta p_{inc}}(\theta_i) \right\}, \left\{ \frac{\Delta p_r^M}{\Delta p_{inc}}(\theta_i) \right\}$$

$$D\{M(\vec{p}_i), E(\vec{p}_i)\} = \left\| \left\{ \frac{\Delta p_r^M}{\Delta p_{inc}}(\theta_i) \right\} - \left\{ \frac{\Delta p_r^E}{\Delta p_{inc}}(\theta_i) \right\} \right\|_{l^1} = \sum_i \left| \frac{\Delta p_r^M}{\Delta p_{inc}}(\theta_i) - \frac{\Delta p_r^E}{\Delta p_{inc}}(\theta_i) \right|$$

- Suppose the calculation and the data are random fields: $M(\vec{p}_i) = \hat{M}_{\vec{p}_i}(\vec{x}, t)$, $E(\vec{p}_i) = \hat{E}_{\vec{p}_i}(\vec{x}, t)$ Then, e.g.

$$D\{M(\vec{p}_i), E(\vec{p}_i)\} = \left\| \text{Exp}(\hat{M}_{\vec{p}_i}) - \text{Exp}(\hat{E}_{\vec{p}_i}) \right\|_{L^p}$$

- Inference is complex because the distributions are usually empirical or poorly characterized.
- Non-metric validation metrics are beyond the scope of this talk.



The right conclusions:

- The goal of V&V is to measure credibility of the code for an intended application, usually involving prediction:

$$C_{red} \left[D\{M(\vec{p}_1), E(\vec{p}_1)\}, \dots, D\{M(\vec{p}_N), E(\vec{p}_N)\} \right]$$

- This puts a premium on the quality of the validation metrics:
 - Converged calculations?
 - Guaranteed no code bugs?
 - Experimental uncertainty (variability and bias) quantified?
Replicated in the calculations?
 - Experimental sensitivity matched by code?
- Decisions depend on our assessment of credibility.
- How sensitive are decisions to the various factors?

Example of a trivial credibility function:

- In the Mach reflection problem, credibility was not formally quantified.
 - The numerics was too problematic to make that task worthwhile (in otherwords lack of credibility was obvious).
- But, hypothetically, consider a verification problem: **Sod shock tube problem (LeBlanc shock tube problem)**.
 - Statement: ALEGRA converges to exact solution in L^1 .
 - Credibility: statement is necessary.
 - Translation:

$$C_{red} [D\{M(\vec{p}_1), Sod(\vec{p}_1)\}] = C_{red} [\|M(\vec{p}_1) - Sod(\vec{p}_1)\|_{L^1}] \\ = \begin{cases} \text{Pass, if } \|M(\vec{p}_1) - Sod(\vec{p}_1)\|_{L^1} < \varepsilon \\ \text{Fail, otherwise} \end{cases}$$

- Passing Sod (and LeBlanc!) is necessary – not sufficient.



What is a credibility function?

- **Interesting examples appear in statistical software reliability theory**
 - **For example, consider the number of “failures” in the time interval $[0,t]$, $N(t)$.**
 - **Assumptions lead to the description of $N(t)$ as a Poisson process, and allows the calculation of things like probability of k failures in $[0,t]$, probability of a failure in $[t,2t]$, probable time of $k+1^{\text{st}}$ failure, etc.**
 - **Credibility, for example, increases if probable time of next failure is large, or likely number of future failures is small.**
- **What is a “failure” for computational science? Probably an extension of reliability theory, such as:**
 - **A validation metric that is too large.**
 - **Too many failed experimental comparisons.**

Return to calibration:

- Credibility and calibration don't have to use the same formalism:

$$C_{red} \left[D\{M(\vec{p}_1), T(\vec{p}_1)\}, \dots, D\{M(\vec{p}_N), T(\vec{p}_N)\} \right]$$

$$\min_{\hat{p} \subset \vec{p} \in \Omega} C_{cal} \left[D_{cal} \{M(\vec{p}_1), T(\vec{p}_1)\}, \dots, D_{cal} \{M(\vec{p}_N), T(\vec{p}_N)\} \right]$$

- Calibration should acknowledge credibility, hence what is known about the results of validation:

$$\min_{\hat{p} \subset \vec{p} \in \Omega} C_{cal} \left[D_{cal} \{M(\vec{p}_1), T(\vec{p}_1)\}, \dots, D_{cal} \{M(\vec{p}_N), T(\vec{p}_N)\}; C_{red} \right]$$

- We are currently investigating calibration formalisms accounting for model uncertainty, such as that due to Kennedy and O'Hagan or found in machine learning theory, with this goal in mind.



Calibration and Validation: Who Cares?

- Scientists care: about calibration and V&V, and their role in R&D
 - Center of gravity is scientific progress.
 - **“It’s so beautiful it has to be right!”***
- Code developers care: about V&V
 - Center of gravity is testing their software (users are testers).
 - **“We built a really good code, but nobody used it!”***
- Decision makers care: about prediction
 - Center of gravity is spending money and risking lives.
 - **“We scientists do the best we can; we can’t be held legally liable for mistakes!”***
- Measures of success are not necessarily the same for these key groups.

***Quotes it’s been my displeasure to hear over the past seven years.**



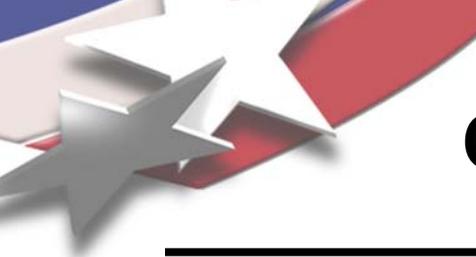
Conclusions:

- Anything dealing with code calculations starts with verification.
- Validation and calibration are different.
- Disguising calibration as validation is dishonest.
- Calibration is dangerous in high-consequence computing (latest example is the use of CRATER – **AN ALGEBRAIC MODEL** – in the Columbia flight); the danger may be reduced by careful acknowledgement of the results of a rigorous validation effort during calibration.
- Prediction with a quantified basis for confidence remains the most important problem.



Some references

- **L. P. Swiler and T. G. Trucano, “Treatment Of Model Uncertainty In Model Calibration,” 9th ASCE Specialty Conference on Probabilistic Mechanics and Structural Reliability, Albuquerque, July, 2004, SAND2004-2317C.**
- **Takeru Igusa and Timothy G. Trucano, “Role Of Computational Learning Theory In Calibration And Prediction,” 9th ASCE Specialty Conference on Probabilistic Mechanics and Structural Reliability, Albuquerque, July, 2004.**
- **L. P. Swiler and T. G. Trucano, “Calibration Under Uncertainty – A Critical Review,” Sandia Report (in progress).**
- **T. G. Trucano, L. P. Swiler, T. Igusa, W. L. Oberkampf, and M. Pilch, “Calibration, Validation, and Sensitivity Analysis: What’s What and Who Cares?,” 4th International Conference on Sensitivity Analysis of Model Output, March 8-11, 2004, Santa Fe, NM (in progress).**



Conclusion:

“We make no warranties, express or implied, that the programs contained in this volume are **FREE OF ERROR, or are consistent with any particular merchantability, or that they will meet your requirements for any particular application. **THEY SHOULD NOT BE RELIED UPON FOR SOLVING A PROBLEM WHOSE SOLUTION COULD RESULT IN INJURY TO A PERSON OR LOSS OF PROPERTY...**”**
[Emphasis Mine] (from Numerical Recipes in Fortran, Press, Teukolsky, Vetterling, and Flannery)

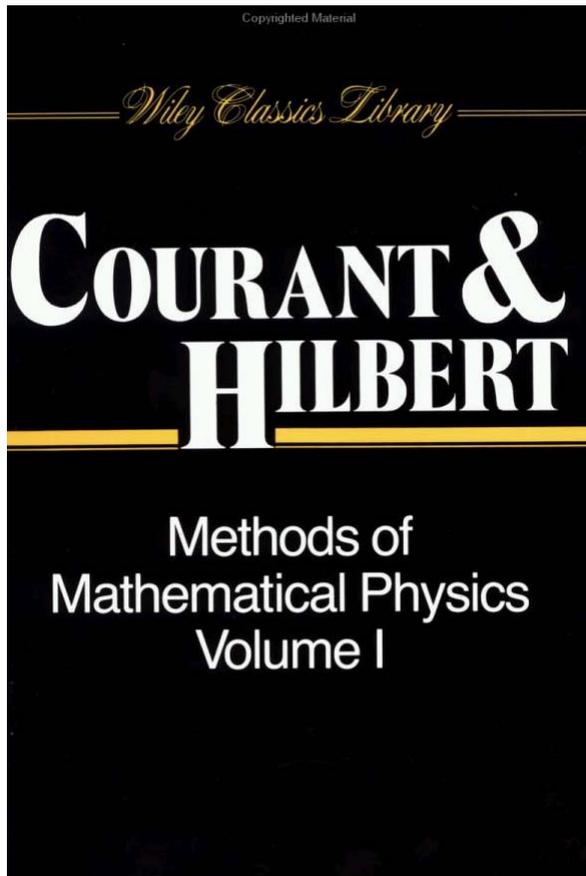
Will we be able to seriously claim that ASCII codes are any better than this?!

How absurd would the following be?



**We make no warranties,
express or implied, that the
bridge you are about to drive on
is free of error...**

How much more absurd would the following be?



**We make no warranties,
express or implied, that the
book you are about to read
is free of error...**