NDIA #19703: Verification and Validation in CREATE Multi-Physics HPC Software Applications

DOD HPC MODERNIZATION PROGRAM

Lawrence G. Votta Ph.D. with

K.P. Kendall, D.E. Post, E.T. Moyer and S.A. Morton

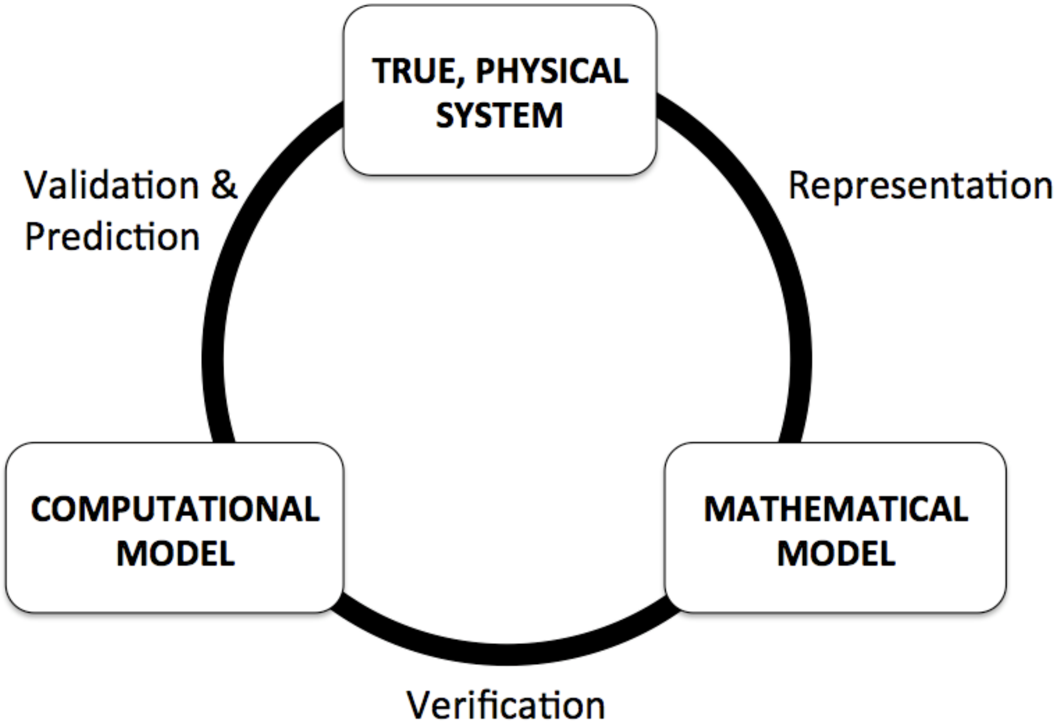
October 2017

Outline

- Introduction
  - VVUQ
  - Design-Analyze-Build
  - Observations
- CREATE Verification and Validation Principles
- Verification Practices
- Validation Practices
- Examples
- Observations and Conclusions

Introduction: Verification & Validation

The Modeling and Simulation Ecosystem



Verification, validation, and prediction as they relate to the true, physical system, the mathematical model, and the computational model. (Adapted from American Institute for Aeronautics and Astronautics. 1998.)

Introduction: Verification & Validation

Important Terms and Concepts - 1

- **Quantity of Interest (QOI) –** are the output(s)/result(s) of computational models, and are used in the engineering and study of modeled systems.
- **Verification –** how accurately a computer program ("code") correctly solves the equations of the mathematical model.
- **Validation –** the degree to which a model is an accurate representation of the real world from the perspective of the intended uses of the model.
- **Uncertainty Quantification (UQ) –** quantifying uncertainties associated with model calculations of true, physical QOIs.

Introduction: Verification & Validation

Important Terms and Concepts - 2

- **Community of Interest –** A community of domain experts, computational users and modelers that maintain detailed domain knowledge, shared validation test suites, and benchmarks for problems of interest.
- **Intended Use –** A computational model cannot “be proven” correct. Usually a community of interest defines problems in a domain and sets an acceptable level of testing to insure that the computational model is validated. An intended use is defined by the set of problems.

Introduction: CREATE Project

Design – Analyze – Build

Concept Development	Engineering Development	Post Development

Design→Build→Test

Virtual Environment

Multi-physics based performance analysis

Build Final Product → Physical Test

Manufacture And Sustainment & Modification

21st Century Goal – Rapid and Agile Systems Development

Potential for Large Productivity Gains	Potential for Large Productivity Gains	Additional Productivity Gains

Introduction:
Observations

- "essentially, all models are wrong, but some are useful" (https://en.wikipedia.org/wiki/George_E._P._Box)
- "Since it isn't possible to prove that the complex multi-physics software in the Computational Research and Engineering Acquisition Tools and Environments (CREATE) program is mathematically "correct", there's a risk that without adequate testing, it won't be trusted to provide accurate predictions of weapon system performance".

Core CREATE Verification and Validation Practices - Principles

Align testing with National Research Council best-practices for scientific software, supplemented and refined by the CREATE staff's collective experience in DoD, DOE, industry, and academia. (ISBN 978-0-309-25634-6)

- Verification Principles:
  - Solution verification is well-defined only in terms of specified QOIs.
  - The efficiency and effectiveness of code and solution verification can often be enhanced by exploiting the hierarchical composition of codes and mathematical models, with verification performed first on the lowest-level building blocks, and then on successively more complex levels.
  - The goal of solution verification is to estimate, and control if possible, the sources of error in the implementation of the models for each QOI for the problem at hand.

Core CREATE Verification and Validation Practices - Principles

- Validation Principles:
  - A validation assessment is well-defined only in terms of specified QOIs and the accuracy needed for the model's intended use.
  - A validation assessment provides information of model accuracy only in the domain of applicability "covered" by the physical observations employed in the assessment.
  - The efficiency and effectiveness of validation and prediction assessments are often improved by exploiting the hierarchical composition of computational and mathematical models.
  - Validation and prediction often involve specifying or calibrating model parameters.
  - The uncertainty in the prediction of a physical QOI must be aggregated from uncertainties and errors introduced by many sources, including discrepancies.
  - Validation assessments should consider the uncertainties and errors in physical observations (measured data).

Core CREATE Verification Practices

"… code and solution both …"

1. *Design code with hierarchical code verification in mind*.
2. *Develop a verification test plan*.
3. *Verify the code prior to validation*.
4. *Verify the code, as much as is practical, and document the coverage*.
5. *Conduct hierarchical testing (that is, unit, integration, system, and regression tests), and document the results. Automate testing to the greatest degree possible*.
6. *Document the domain and range of intended use of the code.*
7. *Use as many types of verification tests as are feasible*.
8. *Test for software integrity*.

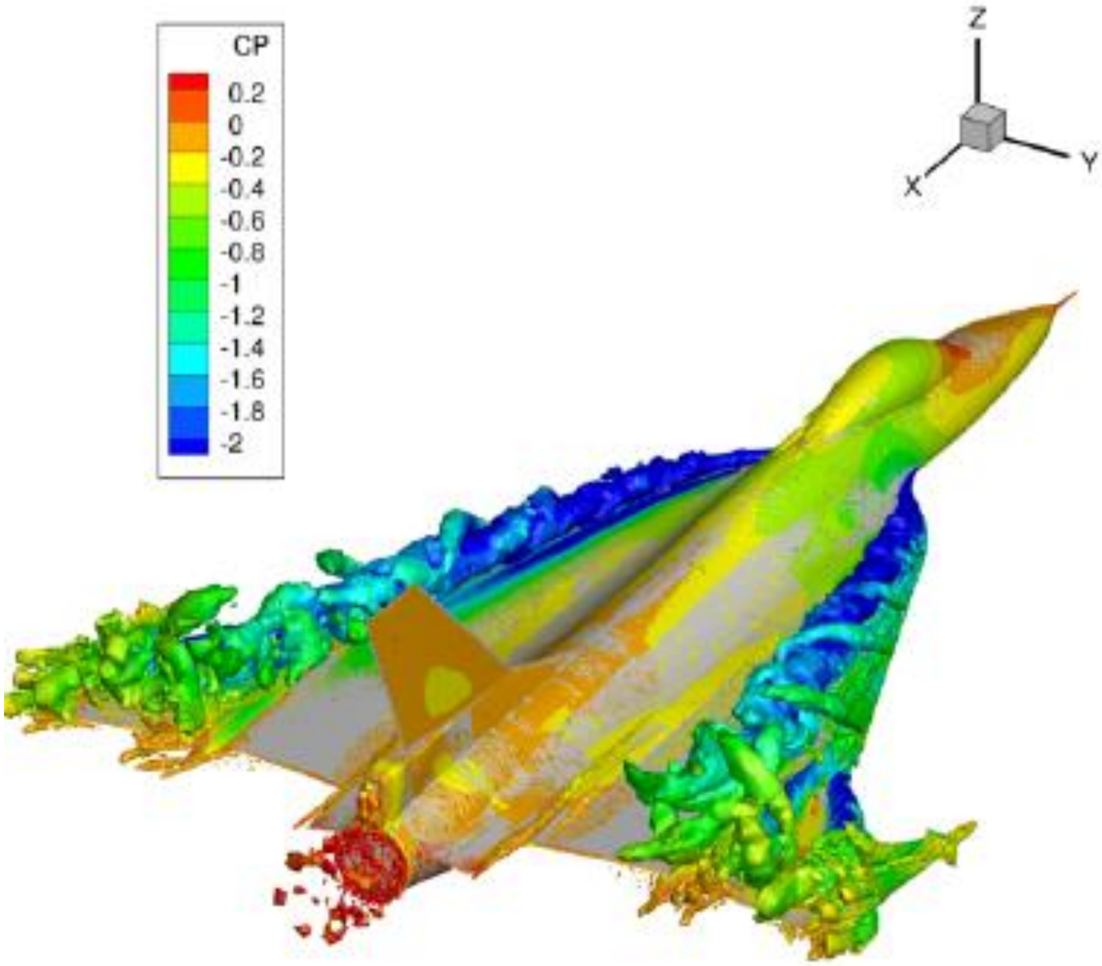
Core CREATE Validation Practices

"… model represents reality …"

9. *Validate for the full range of the code's intended use.*
10. *Develop an archival database for validation.*
11. *Validation should be focused on the behavior and accuracy of QOIs associated with use-cases*.
12. *When metrics are used to assess the difference between model and experiment, they should only measure the mismatch between computational and experimental results*.
13. *Develop validation project plans, review them with independent experts and users, and execute them*.
14. *Formally assess the V&V status and progress.*

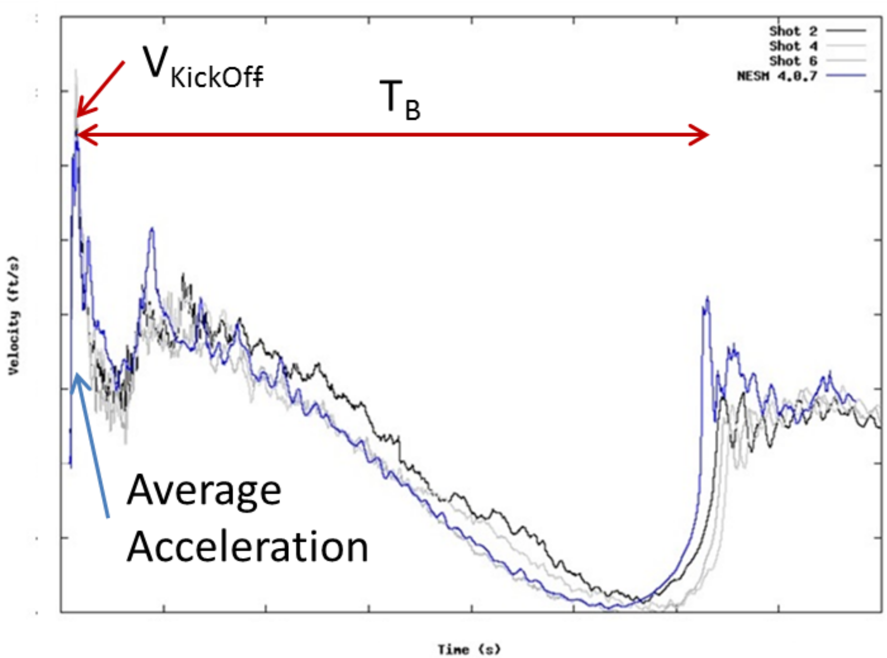
Example

CREATE AV Turbulence Model Tests



Example

Floating Shock Platform (FSP) and typical FSP underwater explosion (UNDEX) test



Distribution A: Approved for Public release; distribution is unlimited.

Observations and Conclusions

- Automate testing as much as possible.
- As intended uses for computational models evolve, so do the V&V test case suites. There is a continual need for maintenance and evolution of test cases and tools that support automated testing.
- The principles, as discussed in the NRC, have led to a cost-beneficial set of practices that lead to high-quality supercomputer software applications.