Draft

Simulation Verification and Validation

Christopher J. Roy
Associate Professor
Aerospace and Ocean Engineering Department
Virginia Polytechnic Institute and State University
Blacksburg, Virginia 24061 USA

and

William L. Oberkampf
Consulting Engineer
Albuquerque, New Mexico 87112 USA

Abstract

With the advent of digital computers in the middle of the 20th century, simulation joined theory and experiment as an approach for predicting the behavior of engineering and natural systems. Advances in processor speed and in the ability to compute on a large number of processors at once (parallel processing) have resulted in the world's fastest computers doubling their computing power roughly every fourteen months. These remarkable advances in computing power have lead to an increased reliance on simulation in the decision-making process for researchers, project managers, and policy makers. The weaknesses of modern-day simulations are only now beginning to be understood. This fact should not be surprising given the relatively short 50 year history of simulation versus that of theory (~300 years) and experimentation (>3000 years). Over the last two decades, Verification and Validation have emerged as a framework for rigorously assessing the accuracy and reliability of computer simulations.

Introduction

While the term simulation could be broadly interpreted to include a wide variety of activities, here we will focus exclusively on those computational simulations which involve numerical solutions to partial differential equations, which are ubiquitous in the physical sciences and engineering. We further distinguish between a model and a simulation. The term model is used to represent the partial differential equations, initial conditions, boundary conditions, and any auxiliary relations or submodels that are used. Since these models rarely have closed-form mathematical solutions, we are generally required to use numerical methods to obtain approximate solutions using computers, i.e., simulations.

Sources of Simulation Error

In general, one would like to know the error in the simulation relative to the true value in nature. The simulation error can thus be written as

$$E_{sim} = f_{sim} - f_{nature}$$

where f_{sim} is the simulation result and f_{nature} is the true value found in nature. Knowledge (or estimates) of the simulation error can give us insight into the predictive capability of the simulation. However, the above equation does not explicitly show the many possible sources of error which can be broadly classified into mathematical errors (Verification) and physical modeling errors (Validation).

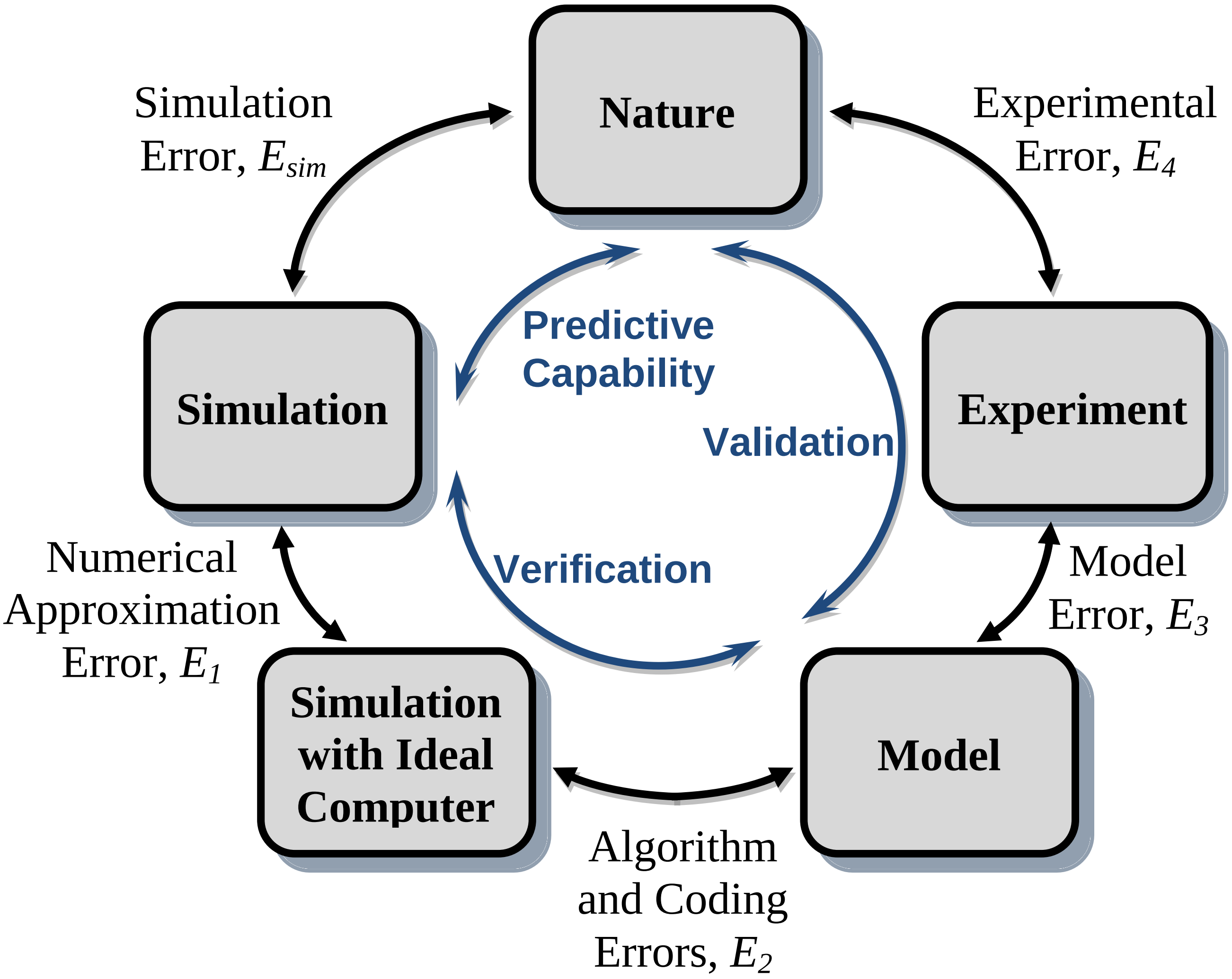
In order to identify the different sources of error, it is helpful to expand the simulation error equation as

$$E_{sim} = (f_{sim} - f_{sim}^{C\to\infty}) + (f_{sim}^{C\to\infty} - f_{model}) + (f_{model} - f_{exp}) + (f_{exp} - f_{nature})$$

where $f_{sim}^{C\to\infty}$ is the result one could hypothetically obtain on an ideal computer with infinite speed and memory (this allows us to assume that all sources of numerical error related to computer round-off, domain discretization, etc. are zero), f_{model} is the exact solution to the model, and f_{exp} is the result of an experimental measurement of this system. This expanded form of the simulation error equation highlights the possibility of error cancellation when comparing the results of a simulation to experimental data. A more compact form of the simulation error equation is given by:

$$E_{sim} = E_1 + E_2 + E_3 + E_4$$

These error sources represent numerical approximation errors (E_1), algorithm and computer programming errors (E_2), modeling errors (E_3), and experimental measurement errors (E_4). In order to estimate the accuracy and reliability of a simulation, each of these different error sources must be estimated. These error sources are briefly discussed below.


Nature
Simulation
Experiment
Simulation with Ideal Computer
Model
Simulation Error, E_{sim}
Experimental Error, E_4
Numerical Approximation Error, E_1
Model Error, E_3
Algorithm and Coding Errors, E_2
Predictive Capability
Validation
Verification

Schematic of the Verification and Validation process for computational simulation showing the different error sources.

Numerical Approximation Errors (E_1)

Solution verification deals with the estimation of the numerical approximation errors in computational simulation. The numerical errors that occur in every single simulation are round-off and discretization errors. Round-off errors are due to the fact that computers are only able to retain a finite number of significant figures, usually either 7 (single precision) or 14 (double precision). Discretization error comes from the fact that the domain of interest (usually a region of space and/or a period in time) must be broken into discrete values. For example, a climate simulation might decompose the earth's atmosphere into computational cells that are 15 km wide in the horizontal directions and 1 km high in the vertical direction. The numerical accuracy of the simulations will depend on the resolution (e.g., the number of cells) used to decompose the computational domain. After the discretization step, the original partial differential equations have now been converted into a set of algebraic equations that can be solved on the computer. For nonlinear models (and large linear models), these algebraic equations cannot be solved exactly in an efficient manner, and are instead solved using approximate iterative methods. The difference between the current solution during this iterative procedure and the exact solution to the algebraic equations is the iterative error, another source of numerical error.

Discretization error is usually the largest and most difficult numerical error to estimate. While there are many approaches for estimating discretization error, the simplest and most broadly applicable is Richardson extrapolation, which makes use of the fact that the discretization errors will be reduced as more and more computational cells are used (i.e., as the mesh is refined) and as smaller time steps are used. For a representative cell length of h, the discretization errors generally reduce at a rate proportional to h^p, where p is the order of accuracy of the numerical approximation method. If this rate is either known or estimated from solutions on successively refined meshes, then the discretization error can be estimated by extrapolating to the limit as $h \to 0$. However, when using Richardson extrapolation, it is important to ensure that round-off and iterative errors are much smaller than the discretization errors to ensure that the extrapolation process is reliable.

Algorithm and Coding Errors (E_2)

Ensuring that no algorithm or coding errors are present is called code verification. Algorithm errors can occur when a numerical algorithm is chosen which does not reproduce the exact solution to the model as the mesh is refined (i.e., as $h \rightarrow 0$). Algorithm errors are related to algorithm consistency and stability, and can be particularly troublesome since they can sometimes exhibit the proper convergence rates with mesh refinement while converging to the wrong solution. Coding errors are also called programming mistakes or simply “bugs.” For more general software applications, system-level testing is used to check whether or not the software produces the correct answer based on a given set up inputs. A key difficulty in applying such general practices to computational simulation software is that the output of the code is only an approximate numerical solution to the model, and thus there is no straightforward system-level software test that can be applied.

Both algorithm and coding errors can be addressed by instead solving a different problem for which we know the exact solution to the model. One approach is to solve a simplified problem where exact analytic solutions to the partial differential equations are known. A more general approach is the method of manufactured solutions wherein the differential equations themselves are altered in order to accommodate a chosen exact solution. In either case, the code is verified by examining the rate at which the approximate numerical solutions produced by the code converge to the exact solution to the model at the proper rate (i.e., at the formal order of accuracy p of the numerical algorithm).

Model Error (E_3)

The model error is the difference between the exact solution to the model (i.e., the partial differential equation along with the initial and boundary conditions as measured in the experiment) and the experimentally measured value. The comparison between the model and the experimental result is conceptually and mathematically more difficult than estimating E_1 and E_2 for two reasons. First, the experimental result will contain uncertainty, both random (stochastic) measurement uncertainty, as well as systematic (bias) measurement uncertainty. The random measurement uncertainty is usually characterized as a random variable, which can be represented as an empirical distribution function for the measurements. Second, critical experimental information, such as initial conditions, boundary conditions, and system characteristics, must be measured and provided as input data to the model so that a simulation can be conducted. This information is usually characterized by the experimentalist as either a random variable or as an interval-valued quantity. This uncertain input data must then be propagated through the model, usually with Monte Carlo sampling, in order to compute the uncertain response quantities from the model for comparison with the experimental measurements. As a result, a quantitative comparison of the model results and the experimental results must deal with a difference measure between either cumulative distribution functions or between probability-boxes; one from the experiment and one from the model. This new field of research is now referred to as the construction of validation metric operators. It should be stressed that these operators serve a very different purpose than parameter estimation methods used in model calibration or model updating.

Experimental Error (E_4)

The difference between the experimentally-measured value and the true value found in nature is the experimental error, which is the focus of the field of metrology. The value from nature can be considered as either a fixed but unknown quantity, or as a random variable that is the result of a stochastic process due to initial and boundary conditions. The experimentally-measured value f_{exp} always contains both random and systematic errors due to the measurement process. While the random errors can be reduced by increasing the number of experimental measurements, it is the systematic errors which are most challenging to characterize. Systematic errors in the experimental measurements are also the most damaging since they directly affect the model evaluation process, or worse, are transferred to the model during model calibration or model updating procedures. The design of experiments approach has proven very effective in quantifying both random and systematic errors. Using randomization and blocking techniques, one can convert systematic errors into random errors, thereby providing control over experimental error by increasing the number of measurements.

Predictive Capability

The errors that arise in each phase of the computational simulation process cannot be computed directly because the ideal computer does not exist and the exact value of the model and the true value in nature are not known. These errors must therefore be estimated, and this estimation process converts these errors into uncertainties. The various sources of uncertainty affect the model’s capability to predict the behavior of a system. In addition, the preceding discussion is appropriate for cases where experimentally-measured data are present. However, when computational simulations are used to predict the behavior of a system for cases where experimental data are not available, then there is an additional source of uncertainty associated with the interpolation between or extrapolation from the model validation database. In addition, the model must also address other possible sources of uncertainty in the prediction, e.g., uncertainties in the actual system of interest, its surroundings, possible environments (such as system failure conditions), and possible event scenarios of interest.

Bibliography

ASME, “Guide for Verification and Validation in Computational Solid Mechanics,” American Society of Mechanical Engineers, ASME V&V 10-2006, 2006.

Oberkampf, W. L., Trucano, T. G., and Hirsh, C., “Verification, Validation, and Predictive Capability in Computational Engineering and Physics,” *Applied Mechanics Reviews*, Vol. 57, No. 5, 2004, pp. 345-384.

Roache, P. J., *Verification and Validation in Computational Science and Engineering*, Hermosa Publishers, New Mexico, 1998.

Roy, C. J., “Review of Code and Solution Verification Procedures in Computational Simulation,” *Journal of Computational Physics*, Vol. 205, No. 1, 2005, pp. 131-156.

Additional Reference for On-Line Version

AIAA, “Guide for the Verification and Validation of Computational Fluid Dynamics Simulations,” American Institute of Aeronautics and Astronautics, AIAA-G-077-1998, 1998.

Knupp, P. and Salari, K., *Verification of Computer Codes in Computational Science and Engineering*, Rosen KH (ed.), Chapman and Hall/CRC, Boca Raton, FL, 2003.

Keywords

Verification and Validation, Modeling and Simulation, Uncertainty Quantification, Numerical Error, and Risk Assessment