PSFC/JA-09-34

Tutorial: Verification and Validation for Magnetic Fusion

Greenwald, M.J.

December 2009

Plasma Science and Fusion Center
Massachusetts Institute of Technology
Cambridge MA 02139 USA

This work was supported by the U.S. Department of Energy, Grant No. DE-FC02-99ER54512. Reproduction, translation, publication, use and disposal, in whole or in part, by or for the United States government is permitted.

Submitted for publication to *Physics of Plasmas.*

Tutorial: verification and validation for magnetic fusion

Martin Greenwald
Massachusetts Institute of Technology
175 Albany St.
Cambridge, MA 02139

Abstract

Dramatic progress in the scope and power of plasma simulations over the past decade has extended our understanding of these complex phenomena. However, as codes embody imperfect models for physical reality, a necessary step towards developing a predictive capability is demonstrating agreement, without bias, between simulations and experimental results. While comparisons between computer calculations and experimental data are common, there is a compelling need to make these comparisons more systematic and more quantitative. Tests of models are divided into two phases, usually called verification and validation. Verification is an essentially mathematical demonstration that a chosen physical model, rendered as a set of equations, has been accurately solved by a computer code. Validation is a physical process which attempts to ascertain the extent to which the model used by a code correctly represents reality within some domain of applicability, to some specified level of accuracy. This paper will cover principles and practices for verification and validation including lessons learned from related fields.

I. Introduction and Background

There is a strong motivation to develop and utilize ever more powerful computational models for magnetic fusion energy. Codes capable of making reliable predictions already have enormous practical value for designing experimental scenarios in existing or planned devices and for interpreting experimental results. In addition to their immediate uses, these codes can demonstrate and embody our state of knowledge for complex and dynamic plasma systems. Since predictions and, more particularly, errors in predictions can have far reaching consequences, it is critical to assess our models in a systematic and rigorous way as we move beyond purely empirical predictive approaches in meeting programmatic goals. Other related fields have already begun to address these issues. For example, Computational Fluid Dynamics (CFD) codes are used for design of commercial airplanes, fission reactor cooling systems and skyscrapers (where wind loads are critical). In these fields, with impacts on public safety, the environment and the economy, tests of code reliability can be part of regulatory regimes and professional engineering standards [1, 2] as well as editorial policy statements from prominent engineering journals. With the construction of ITER, a licensed nuclear facility, fusion will be confronting a similar set of concerns. Overall, the challenge is to assess and improve the confidence in predictions derived from complex simulations.

The challenges to theory are well known. Governed by the Boltzmann-Maxwell equations, plasma physics encompasses a huge range of temporal and spatial scales, with essential nonlinearities and with extreme anisotropies arising from the imposed magnetic field and complex three-dimensional geometry. Exact analytic solutions are possible only for a few isolated classes of highly simplified problems, while direct numerical integration over the full range of scales is far out of reach. Despite the challenges, dramatic progress has been achieved. These are due to advances in theory, improved algorithms and numerical methods, new generations of powerful computers and improved comparisons with experiments, paced by increasingly sophisticated measurement systems. Still, “virtual reality” from simulations is nowhere in sight and for the foreseeable future, we must rely on approximate solutions to approximate equations. The equations, algorithms, approximations and assumptions comprise a conceptual (or physical) model for a particular problem. For plasmas, typically the approximations involve temporal or spatial domain decomposition. The main branches of plasma models can be categorized principally by the range of temporal scales that they cover, running from the shortest: RF (Radio Frequency) models through turbulence, MHD and transport, the longest. Spatial domain decomposition for a confined plasma would typically address the plasma core, pedestal scrape-off layer (SOL), divertor and wall as separate, but coupled problems.

It is worth considering formal definitions for the term *model* in this context. From Eykhoff [3] we have “a representation of the essential aspects of an existing system (or a system to be constructed) which presents knowledge of that system in usable form”. In as similar vein, from Huber we have “a model can be defined as “a representation of the essential aspects of some real thing, in an idealized or exemplary form, ignoring the inessential aspects”. In both definitions the challenge is to identify and demonstrate what is essential and what is not. Model testing is meant to demonstrate that developers have correctly understood the underlying physics and have made the right set of choices. This paper is about how that testing might be done. The issues raised when considering the reliability of computer simulations are pragmatic versions of a more general question of epistemology, “how do we know what we know”. Historically, what we now call theory, on the one hand, and experiments, on the other were embodiments of two opposed schools of philosophy. Rationalism, championed by Plato, Descartes and Spinoza, was based on the logical development of a model based on indisputable axioms – pure logic. They believed that knowledge gained through the senses is always confused and impure. In contrast, Empiricism, espoused by John Locke and Francis Bacon among others, required that every axiom, deduction, assumption or outcome be empirically confirmed. The empiricists only trusted knowledge gained through the senses. In the early days of CFD simulation, this dichotomy was sharpened into a debate over whether experiments (wind tunnels) would soon be obsolete [4].

The scientific method, though usually associated with empiricism, defines an approach which tries to combine the best features of both schools. Theory, now augmented with modern computation, can provide predictive capabilities and fundamental understanding with near perfect diagnostics – that is, with all of its internal states exposed. Computation offers a high degree of flexibility and is often cheap and fast (to run, if not to develop). However, for complex problems it can provide only imperfect models and solutions. Experiments work with a “perfect” model, that is reality itself, though measurements of that reality are necessarily imperfect and incomplete. Experiments invariably discover unpredicted phenomena and for fusion energy, are the only method to demonstrate concretely the required levels of performance. (To quote from a 2002 presentation by Bill Nevins, “Achieving ignition with pencil and paper doesn't count, achieving ignition in a computer doesn't count".) The two approaches are clearly complementary rather than competitive.

II. Verification And Validation

The particular terms verification and validation were adopted by the CFD community to describe two distinguishable sets of model-testing activities [5, 6 7, 8]. *Verification* assesses the degree to which simulations correctly implement a physical (conceptual) model while *validation* assesses the degree to which a physical model captures "reality" through comparison with experimental measurements. These words are almost synonyms in ordinary usage and their use for the specific activities described here is arbitrary, but not unimportant. Model testing is a collective activity, using a common vocabulary allows us to interact more effectively and share knowledge gained more effectively. Verification and validation (V&V) can be seen as a logical extension of standard scientific method, called out as distinct processes to emphasize a more rigorous, systematic and quantitative approach to model testing than had previously been applied. As a practical matter, verification and validation can be viewed as essential confidence building activities – an accumulation of evidence that codes are correct and useful. Strictly speaking we do not verify and validate codes. Technically we verify a set of calculations, then draw inferences about the validity of a code. Similarly, one validates a set of simulations, then draws inferences about the underlying model.

Several other terms are worth discussing at this point. *Qualification* can be defined for our purposes as the theoretical specification for the expected domain of applicability for a model. This is critical since extensive model testing outside of this domain is generally not useful. *Calibration* is the adjustment of parameters in a computational model in order to improve agreement with data. Calibration may be justified *only* to account for physics beyond the scope of the model; it should never be used to obscure essential errors in a model or its implementation. Finally, we can define *prediction* as the use of a code outside its previously validated domain to foretell the state of a physical system, or more generally as a simulation result for any specific case that is different from cases that have been validated. The most reliable sort of prediction is where the application is entirely inside the validation domain. In this case predictions is equivalent to interpolation and is the basis for sound engineering practice [1]. The interpolation is not trivial however, since the domains involved are typically of high dimensionality (in various parameters of interest) with validation data only available sparsely throughout. Estimating prediction uncertainties from validation error is relatively straightforward. For fusion applications, there is interest in extrapolating into new parameter ranges. In this case, the application domain would only partially overlap the validation domain. One must be cautious since large errors could arise from transitions or bifurcations or any other

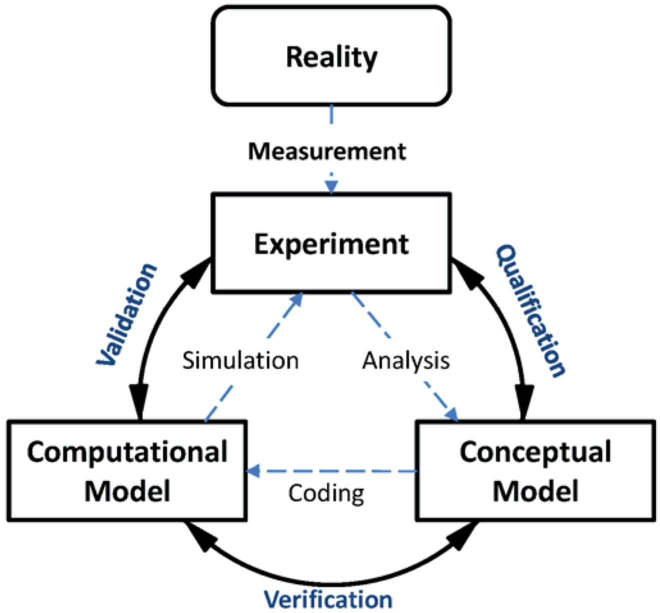

Figure 1. This figure shows the relationship between models and experiments and the verification and validation processes. It is modified from [5] (color online).

unanticipated change in behavior. Accuracy of prediction is not proven, but inferred from V&V activities. Figure 1 shows the relationship between these various activities in graphical form. A key point, and one that we will return to, is that the processes described are elements in a continuous cycle of improvement.

III. Verification Issues and Approaches

As a more formal definition, we can state that Verification is the substantiation that a computerized model and its solution represent a conceptual model within specified limits of accuracy. Figure 2 shows the relation between a conceptual model, computational model and a computational solution. Verification assesses the computational model (code) to determine if it correctly embodies the conceptual model and determines the reliability and accuracy of particular solutions. (An assessment of whether the approximations and assumptions that make up the conceptual model correspond to a real-world physical problem, is left to the validation process.) Verification is essentially a problem in applied math and logic, though generally not one with a rigorous solution. For practical problems of interest, it is not possible to prove

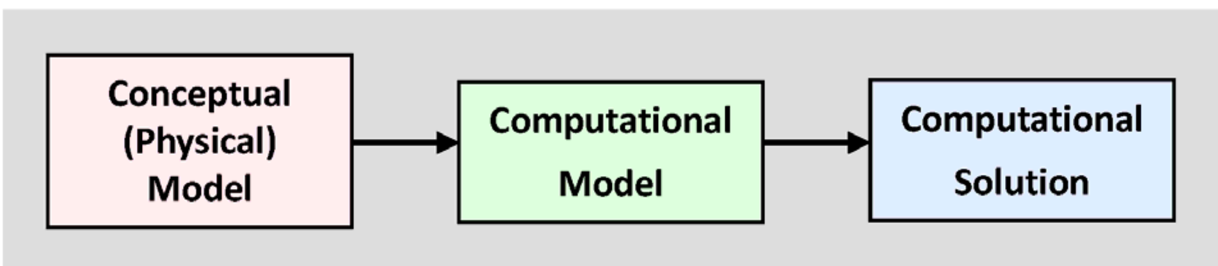

Figure 2. The steps from conceptual model to computational model to computational solution are illustrated. Verification concerns itself with the correctness of the computational model and solution (color online).

mathematically that a code is “correct”, rather we rely on extensive testing to demonstrate the absence of proof that the code is “incorrect”. Since the absence of proof is easy to come by through inaction, this rather unsatisfactory state of affairs requires considerable discipline on the part of practitioners. There are two general sources of error that must be addressed by verification: problems with the code and problems with the solution. The first arise from errors in implementing the conceptual model including those in algorithms or numerical methods as well as simple coding errors or compiler bugs. Solution errors can arise from the spatial or temporal gridding (that is discretization errors which are inevitable for a digital calculation); convergence difficulties; numerical noise, the accumulation of round-off errors and problems or uncertainties in boundary or initial conditions.

A set of formal and informal methodologies have emerged to address the potential problems listed above. At their heart is the question of whether the continuum mathematics of the model has been correctly solved by the discrete mathematics of the computer code [8]. The first step is to follow good software engineering and software quality assurance practices throughout the development process [9, 10]. These practices would cover design, coding, testing, source control, version management and release. The goal is to produce software that is error-free, through testing and design. (Studies have shown that “typical” scientific software has mistakes at the discomforting rate of about one error per hundred lines of code [11].) Since scientific software is usually under continuous development, designing for testability and maintainability are critical, though often overlooked requirements. Regression testing, that is the regular execution of a set of well-defined problems, can help ensure that new errors are not introduced over time. The ideal tool for solution verification is comparison between simulations and highly accurate solutions – either analytic or well known numeric benchmarks, if they exist. Comparison with analytic solutions is not always possible since they are often obtained in the extremes of parameter ranges where codes can have numeric problems. Even if the comparison is satisfactory for simplified problems, strong physics coupling, which may put analytic solutions out of reach, are often exactly the problems for which the codes are designed. Additional methods for verification are employed including formal convergence tests to assess spatial and temporal discretization errors along with conservation and symmetry tests [1, 8]. The latter two methods test whether the solutions obey the same conservation and symmetry properties as the underlying equations and geometry. The method of manufactured solutions [1] can be applied to some problems of interest to fusion. In this method, a made-up (manufactured) analytic solution is constructed for the set of equations used in a code. This solution is then evaluated and all excess terms grouped into a pseudo-source. This source term is then added to the simulation which, if the algorithms and numerics are correct, should return the manufactured solution. This method is rarely in fusion codes, but is in wider use for CFD simulations [12, 13].

An approach which is in common use in plasma physics is code to code comparisons, sometimes called benchmarking. It is entirely plausible that successful comparison between calculations solving the same set of equations, with the same set of initial and boundary conditions can build confidence in the codes [14]. It is argued that this is particularly true if the calculations implement very different methods. In a very well documented example [15] thermal diffusivity calculated with PIC and continuum codes were compared in detail. While valuable, these exercises must be built on systematic verification studies of individual codes (as they were in Reference 15 above) – code benchmarking should not be viewed as a substitute for more basic verification activities [16]. Disagreement demonstrates that at least one of the codes under consideration are wrong, but by itself, agreement does not prove that all are correct.

An activity closely related to verification is uncertainty quantification (sometimes referred to in the literature as UQ) [6, 8, 17, 18]. *Uncertainty quantification* is the quantitative assessment of the degree to which a computed solution differs from an exact one. This is obviously a challenging problem, since the exact solution is not known. Since numeric solutions are always approximate, the distinction between verification and uncertainty quantification is really a matter of emphasis; many of the same tools are employed. Convergence studies through grid refinement, comparison of different grid geometries, comparison of low and high-order solutions on the same grid as well as conservation and symmetry tests are all used to estimate the probable errors. Methods of quantitative analysis using traditional or Bayesian statistical techniques [19] have been developed and employed. An entirely separate source of solution error arises from uncertainties in the boundary or initial conditions used for a particular calculation. These errors may arise from measurement uncertainties when particular physical cases are being run or from uncertainties in other codes coupled to the simulation under consideration. Examples might include the MHD equilibrium used as input for stability calculation, or the temperature and density gradients used in a turbulence code. A common method for evaluating the resulting uncertainty is *ensemble computing*, the execution of a set of code runs which encompass the possible range of errors in the boundary or initial conditions imposed [20, 21]. In a multi-dimensional parameter space, a very large number of runs might be required. The art in this method is to reduce the computation of the full uncertainty space by a huge factor to something computationally tractable. Carrying this out requires a careful sensitivity analysis of reduced spaces and finer sampling in the important parameters. It is worth noting that taken together, verification, uncertainty quantification and validation consume substantial computational (and human) resources – this needs to be taken into account when simulation projects are planned.

IV. Validation Issues and Approaches

Validation is the process of determining the degree to which a conceptual model is an accurate representation of the real world from the perspective of its intended uses. Validation is an essentially physical problem, one that can be as difficult and complex as the underlying science itself. Despite the implications from the standard English usage for the word, validation should not be viewed as a one-time, binary process where codes are accepted for all time or rejected and discarded. Validation is instead part of a cyclic process, whereby models are continuously improved. Validation failures should not be seen as personal failures but rather a normal part of scientific development. In fact, even relatively early in a model's development, where physics elements are known to be deficient, comparison with experiments can be valuable. Developers may be able to identify and concentrate on improving the most important or uncertain elements of a model based on solid evidence rather than guesswork. The conditional nature of the definitions of both verification and validation should be noted. To avoid unbounded philosophical questions, verification and validation are best defined 1. for a targeted class of problems and applications, 2. for a set of specified variables and 3. at a specified level of accuracy. Together, the goal of validation and verification is an assessment of the extent to which a simulation represents true system behavior sufficiently to be useful. Both should be viewed as ongoing, open-ended processes.

A critical element of validation is to confront the significance of the comparisons made. What constitutes agreement? And what inferences can we draw from the level of agreement observed? We are particularly interested in parameters which are important discriminators between models. A well-known results is that for some parameters, very different models may predict essentially the same values. A related characteristic is sensitivity. Some measureable quantities vary more strongly with input parameters than others and variation in some input parameters cause more variation in the results of a simulation than others. As example consider the relation between the normalized temperature gradient R/L_T and the energy flux. Experiments and simulations follow a marginal stability behavior where the flux increases dramatically when the gradient is raised above some critical level. Thus it is easy to predict the gradient from the flux, but hard to predict the flux from the gradient. We want to take advantage of these varying sensitivities to improve the assessment of models and to avoid less meaningful tests.

One successful strategy to deal with these issues is to compare simulations to several measurements at different levels on what has been termed the *primacy hierarchy* [22]. The primacy hierarchy ranks the degree to which parameters of interest are basic or derived from other quantities. For example, in transport the lowest level on the primacy would be the rapidly fluctuating turbulent quantities, perhaps characterized by their spectra or other characteristics of their time series. At the next level are quantities like fluxes, which can be derived from the fluctuation amplitudes, cross-phase and coherence. The highest level would be profiles, which

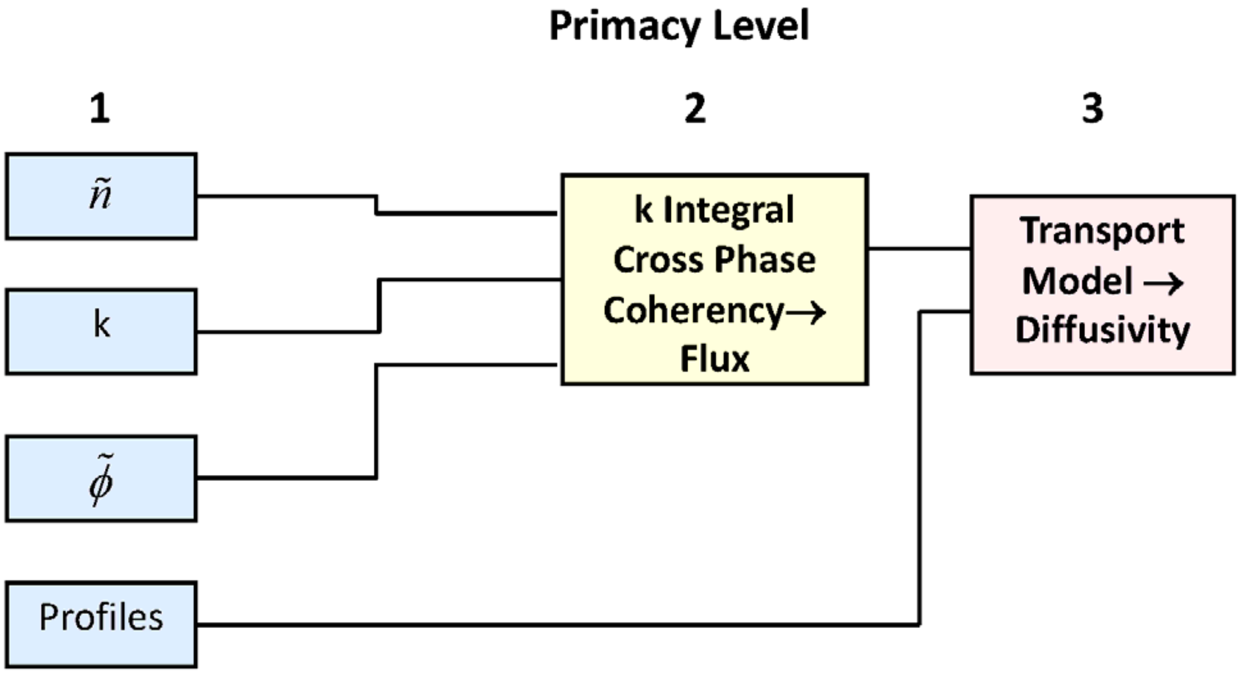

Figure 3. The primacy hierarchy [22] for turbulent transport is illustrated. In this case, fluctuating quantities combine to produce fluxes and fluxes combine with profiles into a transport model (color online).

come from transport analysis using the fluxes. (see Figure 3) Another example comes from RF physics where the lowest level would be computed and measured wave fields, the second level would be the perturbed particle distribution functions and the third level would be local heating (or current drive) profiles. In general, discrimination between models is reduced as one goes up the primacy hierarchy, though comparison at several levels in the hierarchy is best practice. Through analysis, it may be possible to identify ways in which physics cause uncertainties and errors to cancel. One should note that the form of the hierarchy is not necessarily unique – the important thing is to come to grips with the associated issues.

High quality diagnostics are fundamental to the validation process. Fortunately there has been significant process in the scope and accuracy of measurement techniques including measurement of most important 1D profiles, along with advances in fluctuation and imaging techniques [23, 24]. A more recent advancement, which has improved our capabilities for quantitative comparisons with simulations, is the development of synthetic diagnostics. Validation requires comparison of identical quantities and diagnostics often can't make fully resolved, local measurements of fundamental quantities. Mathematical inversions may be impossible or may introduce artifacts. To solve this problem, synthetic diagnostics have been developed as post-processors for several important code-diagnostic combinations, for example [25, 26]. The *synthetic diagnostic* attempts to replicate numerically, in all its detail, the physical processes including viewing geometry, temporal, spatial and spectral averaging that characterize the actual measurement [27]. Afterwards comparison with the real measurements is direct. Development of each synthetic diagnostic is essential a complex exercise in matching phase-space geometries requiring thorough and careful characterization of the physical diagnostic. The synthetic diagnostic code may be quite complex and must be carefully tested.

As noted above, comparison of time series data provide some of the most fundamental validation tests. Mathematics has provided a large tool-box which we have only begun to exploit for these purposes. The most commonly used, the auto-power or spectral density function is easy to compute, even with limited amounts of data, but often does not discriminate sufficiently between competing models. This is not unsurprising perhaps, since the underlying mathematical assumption is that the time series can be described as a superposition of statistically uncorrelated waves. For many problems of interest in fusion plasma physics, nonlinear wave interactions dominate and linear spectral analysis will be of limited use. In these cases, higher order spectral analysis like the bispectrum or bicoherence have been shown to be applicable [28, 29, 30]. A variation of these traditional linear methods, the fractional fourier transform may be useful for studies of RF wave coupling and propagation. This function can be thought of as an oblique slice in the time, frequency plane and has been applied to optics and signal processing. Wavelet analysis has been successfully applied to problems in fluid turbulence [31] allowing for a compact mathematical description of simulated flow fields. With this expansion, it has been shown that virtually all coherent structures can be described by less than 1% of the computed wavelet coefficients [32]. A distinctly different set of tools come under the heading of chaotic analysis. These include calculation of the fractal (or correlation) dimension, recurrence or cyclic analysis and computation of Lyapunov exponents [33, 34]. It seems likely that more complete exploitation of powerful time series analysis methods may provide better sensitivity and discrimination.

V. Quantitative Analysis

Validation requires careful quantitative consideration of uncertainties and errors. These can be divided broadly into two categories as random or systematic . Random errors, also referred to as irreducible or aleatory uncertainty, arise from inherent statistical properties of the physical system. Systematic errors, also known as reducible or epistemic uncertainty, are due to incomplete knowledge. The former can be estimated and treated by classical statistical methods. The latter, by their nature are more difficult to quantify. Bayesian statistical approaches ,with its concepts of *prior* probability distributions and subjective probabilities, offer some means to incorporate systematic uncertainty into a quantitative framework [19, 35]. In the experimental measurements, uncertainties can arise through conceptual errors with the measurement techniques, statistical or counting errors, difference arising from (unaccounted) temporal, spatial or spectral averaging, calibration errors, electronic noise, data acquisition errors or data reduction. Characterization of the errors can also be important. For example, are the errors provided normally distributed, uniformly distributed, or are they confidence intervals or maximum errors?

Validation metrics address a key challenge by making quantitative rather than qualitative assessments of the comparison between simulations and experiments [22. 36. 37. 38]. There is no “correct” or unique way to define these metrics; as we will see, there are various methods to construct them. The goal is not mathematical rigor, but rather an attempt to identify salient elements of the models under testing and to confront disagreement in detail. Validation metrics should take account of uncertainties in both measurements and simulations and account for the breadth of the comparisons. When constructing validation metrics, it is important to identify and to reduce the impact of quantities that have sensitivity to poorly measured parameters used by the models.

The mathematics of the metric can take many forms. In the examples below, y_i will represent a set of n simulated values and Y_i will be the corresponding set of experimental measurements. As simple metric using the L_1 norm, V, could take the form [8]:

$$V = 1 - \frac{1}{n}\sum_{i=1}^{n}\tanh\left|\frac{y_i - Y_i}{Y_i}\right|$$

The tanh function results in a metric which takes the value 1 if agreement is perfect and decreases exponentially as experiment and simulation diverge. If a sufficient set of data is available, the experimental values could be replaced by their mean $\hat{Y}_i$.

$$V = 1 - \frac{1}{n}\sum_{i=1}^{n} \tanh\left|\frac{y_i - \hat{Y}_i}{\hat{Y}_i}\right|$$

Following this line farther, we can account for experimental errors. Here $S_{\hat{Y}}$ is the estimated standard error.

$$V = 1 - \frac{1}{n}\sum_{i=1}^{n} \tanh\left(\left|\frac{y_i - \hat{Y}_i}{\hat{Y}_i}\right| + \left|\frac{S_{\hat{Y}}}{\hat{Y}}\right|\right)$$

Estimated errors in the simulations could be accounted for in a similar manner. Another approach to metrics uses the chi-squared statistic. In this case, lower values of V indicate better agreement. Here, the σ's are the standard deviations for the simulations and measurements.

$$V = \chi_y^2 = \frac{1}{N_{\text{degrees}}}\sum_{i=1}^{n}\left(\frac{y_i - Y_i}{\sigma_y + \sigma_Y}\right)^2$$

A non-trivial point is estimation of the number of degrees of freedom. Typically selection of simulation and experimental data is not "random" in the statistical sense so that the degrees of freedom do not necessarily equal the number of observations. Making the same measurements on the same system over and over again does not reduce all types of errors. Mathematical techniques to account for this include bootstrap or jackknife techniques[39]. Confidence intervals can be used to construct a metric. Using $E \equiv \bar{y} - \bar{Y}$ to represent the estimated error, the confidence interval for the true error is given by

$$\tilde{E} - t_{0.05,N_{DOF}} \frac{s}{\sqrt{n}} < E \text{ (true error)} < \tilde{E} + t_{0.05,N_{DOF}} \frac{s}{\sqrt{n}}$$

Where *t* is from Student's *t* statistic for N degrees of freedom, leading to a validation metric

$$V = \frac{\tilde{E} + 2t_{0.05,N} \frac{s}{\sqrt{n}}}{\bar{Y}}$$

In some situations, average errors are less important than the maximum error. In this case the L_∞ norm would be appropriate, defining

$$V = \left|\frac{\tilde{E}}{\bar{Y}}\right| = \max\left|\frac{y_i - \bar{Y}}{\bar{Y}}\right|$$

Classical hypothesis testing, which returns a binary answer, is probably not appropriate for validation which seeks to inform the development cycle [36].

Composite metrics attempt to assess an overall comparison across several parameters [22]. As with the examples above, these can be constructed in various ways. The combination should rate the validation higher when more quantities are compared, when more experiments or more tests are performed and when measures of sensitivity and uniqueness are high. While the emphasis here is on quantitative methods, the power of good graphical techniques should not be underestimated – especially for data exploration. Best practices probably combine both approaches.

VI. Validation Hierarchy

It's generally not sufficient to implement model validation only on the most complex problems, there are usually too few opportunities to test the basic physical phenomena which underlie the full system behavior. Consider as an example, the field of aerodynamics [8]. Codes are first tested against the simplest wind tunnel experiments – say laminar flow around a sphere. They would then move on to more complex flow regimes and more complicated shapes. Once successful, would be followed by analysis of the full airframe and finally to the full system including control surfaces, control actuators and engines. This stepwise approach is often called the *validation hierarchy* and is illustrated in Figure 4 [40]. As we move from the simplest unit problems to subsystem cases to complete systems, the degree of realism along with the complexity and coupling all increase. At the same time, the quality and quantity of experimental data, the information on initial and boundary conditions and the number of code runs all decrease. Cases at the bottom of the hierarchy provide opportunities for more complete testing of less realistic problems while cases near the top have the opposite characteristic. Prudence would suggest at least as much effort on the bottom of the hierarchy as to the bottom, but this has not been the case in our field historically, where by far the most attention is lavished on the most complex (typically highest performance) devices at the top of the hierarchy. While many basic linear processes were originally observed on low temperature devices (in both linear and toroidal plasmas), we have not validated many of the basic nonlinear phenomena that are critical elements of newer simulations. There are admirable exceptions

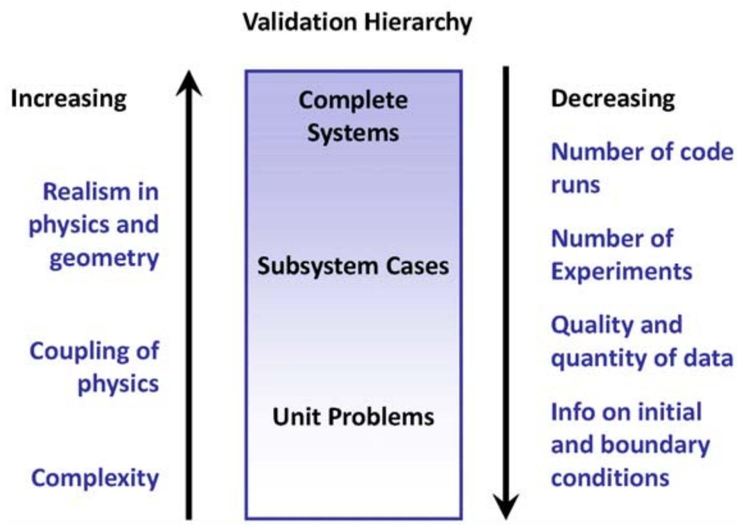

Figure 4. The validation hierarchy running from the most basic problems with simplified physics and geometry to the most complex [2, 40] (color online).

fortunately which should be emulated [43, 44, 45, 46, 47, 48]. This effort requires configuring codes to work in simpler geometry and in unfamiliar parameter ranges and require significant effort to diagnose the experiments, but may be the only way to put our more advanced models on a solid footing.

VII. Best Practices for Validation

While much can be learned from mining databases of previous experiments, the most useful comparisons are carried out in experiments designed and dedicated to the purpose. Older data is often not well enough documented or characterized and in any event direct interaction with experimentalists is essential to the process. Principles for the design and execution of these experiments have been thoroughly discussed [8, 47, 48, 49]. To paraphrase, these would include:

1. Verify codes first.
2. Plan a hierarchy of experiments beginning with the simplest physics and geometry.
3. Design experiments jointly by experimentalists and computationalists.
4. Design experiments to test crucial features of the model, especially its assumptions or important simplifications. Perturbing effects should be minimized. Geometry, boundary and initial conditions must be well characterized and documented. Critical measurements should be defined and limitations, uncertainties, and sources of error discussed with openness and candor.
5. Document code predictions in advance including estimates of uncertainties.
6. While they are jointly designed, carry out experiments and code runs independently.

7. Make as complete measurements as possible when carrying out experiments. Multiple diagnostics to measure the same quantities are desirable. Statistically sufficient data sets should be collected, repeating runs as required. It can be valuable to conduct experiments at more than one facility if this is practical.
8. Pay special attention to analysis of errors and uncertainties. Use modern statistical techniques to design experiments and to identify random and bias errors.
9. When analyzing results, don't paper over differences. The goal is not to prove that a code is correct, but to assess its reliability and point the way towards improvement.
10. Document the process and results including data reduction techniques and error analysis.

VIII. Summary

Despite dramatic advances in computational plasma physics, we are still far from solving the critical problems required to achieve practical fusion energy. The science benefits from a continuous and ongoing collaboration between experiments and simulation, which need to be seen as complementary rather than competitive approaches. Verification and Validation can provide a framework for carrying out the collaboration in a methodical and systematic way with the goals of increasing confidence in the predictive capability of computer simulations. This will require new modes of interaction, especially a greater openness about uncertainties, errors and limitations of methods.

Acknowledgements

The work presented here was supported by DOE awards DE-FC02-99ER54512 and DE-SC0001682

This tutorial grew out of efforts by a joint BPO and TTF working group which included Paul Terry, Jean-Noel Leboeuf, George McKee, David Mikkelsen, Bill Nevins, David Newman and Daren Stotler.

References and Bibliography

[1] Roache, P. J. *Verification and Validation in Computational Science and Engineering*, Hermosa Publishers, Albuquerque, NM. (1998).

[2] W.L. Oberkampf and T.G. Trucano, Progress Aerospace Sci **38**, 209 (2002).

[3] Eykhoff, Pieter *System Identification: Parameter and State Estimation*, Wiley & Sons, (1974).

[4] D.R. Chapman, H. Mark, M.H. Pirtle, Astronautics and Aeronautics **13**, 22 (1975).

[5] S. Schlesinger, Simulation **32**, 103 (1979).

[6] P.J. Roache, Annual Rev. Fluid Mech. **29**, 123 (1997).

[7] AIAA, *"Guide for the verification and validation of computational fluid dynamics simulations"* AIAA-G-077-1998, American Institute of Aeronautics and Astronautics, Reston, VA, 1998.

[8] W.L. Oberkampf and T.G. Trucano, Progress in Aerospace Sciences **38**, 209 (2002).

[9] R.E. Fairley, "Software Engineering Concepts, New York, McGraw-Hill (1985).

[10] J. Sanders and E. Curran, *"Software Quality"*, Reading, MA, Addison-Wesley (1994).

[11] L. Hatton, IEEE Computer Science Engineering **4,** 27 (1997).

[12] S. Steinberg, PJ Roache, J. Computational Physics **57**, 251 (1985).

[13] TM Shih, CH Tan, BC Hwang, International Journal of Numerical methods for fluids **9**, 193 (1989).

[14] A.M. Dimits, G. Bateman, M.A. Beer, et al., Phys. Plasmas **7**, 969 (2000).

[15] W.M. Nevins, J. Candy, S. Cowley, Phys. Plasmas 1**3**, 122306 (2006).

[16] T. Trucano, M. Pilch, W. Oberkampf, *"On The Role Of Code Comparisons In Verification And Validation"* Sandia Report SAND2003-2752, (2003).

[17] B. DeVolder, J. Glimm, J. Fluids Eng. **124**, 129 (2002).

[18] W. Oberkampf and F. Blottner, AIAA Journal, **36**, 687, (1998).

[19] S. Mahadevan, R. Rebba, Reliability Engineering & System Safety, **87** 223, (2005).

[20] R. Klein, S. Doebling, F. Graziani, M. Pilch, and T. Trucano, *"ASC Predictive Science Academic Alliance Program Verification and Validation Whitepaper"*, UCRL-TR-220711 (2006).

[21] Saltelli, A., K. Chan and M. Scott, Eds., *"Handbook of Sensitivity Analysis"*, John Wiley & Sons publishers, Probability and Statistics series (2000).

[22] P.W. Terry, M. Greenwald, J-N. Leboeuf, et al., Physics of Plasmas **15**, 062503, (2008).

[23] R. Boivin, RSI 75, 3381, (2004).

[24] I Hutchinson, *"Principles of Plasma Diagnostics 2^nd^ Edition"*, New York, Cambridge University Press (2002).

[25] C. Holland, A. White, G. McKee et al., Phys. Plasmas, **16** 052301, (2009).

[26] D. Ernst, J. Lang, W. Nevins, M. Hoffman, Y. Chen, W. Dorland, Sparker, Phys. Plasmas **16**, 055906 (2009).

[27] R. V. Bravenec and W. M. Nevins, Rev. Sci. Instrum. 77, 015101, (2006).

[28] Y.C. Kim and E.J. Powers, IEEE Trans. Plasma Sci **7**, 120 (1979).

[29] C.P. Ritz, E.J. Powers, R.D. Bengtson, Phys. Fluids **B 1**, 153 (1989).

[30] J. S. Kim, R. J. Fonck, R. D. Durst, E. Fernandez, P. W. Terry, S. F. Paul, and M. C. Zarnstorff, Phys. Rev. Lett. **79**, 841 (1997)

[31] M. Farge, Annual Rev. Fluid Mech. **24**, 395 (1992).

[32] M. Farge, K. Schneider, G. Pellegrino, A.A. Wray, R.S. Rogallo, Phys. Fluids, **15**, 2886 (2003).

[33] B.A. Carreras, V.E. Lynch, D.E. Newman, G.M. Zaslavsky, Phys. Rev E **60**, 4770 (1999).

[34] B. Ph. van Milligen, R. Sánchez, B. A. Carreras, Phys. Plasmas 11, 2272 (2004)

[35] W.L. Oberkampf, T. Trucano, C. Hirsch, Appl. Mech Rev **57**, 345 (2004).

[36] W.L. Oberkampf and M.F. Barone, J. Comp. Phys. **217**, 5 (2006).

[37] W.L. Oberkampf and T. Trucano, Nuclear Eng. And Design **238**, 716, (2008).

[38] L. Schwer, Engineering with Computers 23, **295** (2007).

[39] B. Efron, The Annals of Statistics, 7, 1, (1979).

[40] M. Greenwald, Computer Physics Communications, **164**, 1, (2004).

[41] U. Stroth, F. Greiner, C. Lechte, N. Mahdizadeh, K. Rahbarnia, and M. Ramisch, Phys. Plasmas **11**, 2558 (2004)

[42] M A Pedrosa, C Hidalgo, E Calderón, T Estrada, A Fernández, J Herranz, I Pastor and the TJ-II team, Plasma Phys. Control. Fusion **47** 777, (2005).

[43] G R Tynan, C Holland, J H Yu, A James, D Nishijima, M Shimada and N Taheri, Plasma Phys. Control. Fusion **48** S51 (2006).

[44] T. Carter, Phys. Plasmas **13**, 010701 (2006)

[45] A. Fasoli, B. Labit, M. McGrath, S. H. Müller, G. Plyushchev, M. Podestà, and F. M. Poli, Phys. Plasmas **13**, 055902 (2006).

[46] K. Gentle, H. He, Plasma Sci. Technol. **10,** 284 (2008).

[47] W. Oberkampf in “*Verification and Validation of Computational Fluid Dynamics*”, F. Grasso, J. Perlaux, H. Deconinck editors, von Karman Institute for Fluid Dynamics publishers, Belgium, (2000).

[48] D.P. Aeschliman and W.L. Oberkamf, AIAA Journal **36**, 733 (1998).

[49] Rizzi, J. Vos, AIAA Journal **36**, 668 (1998).

Figures

1. This figure shows the relationship between models and experiments and the verification and validation processes. It is modified from [5] (color online).
2. The steps from conceptual model to computational model to computational solution are illustrated. Verification concerns itself with the correctness of the computational model and solution (color online).
3. The primacy hierarchy [22] for turbulent transport is illustrated. In this case, fluctuating quantities combine to produce fluxes and fluxes combine with profiles into a transport model (color online).
4. The validation hierarchy running from the most basic problems with simplified physics and geometry to the most complex [2, 40] (color online).