PHYSICS TODAY ONLINE Your daily physics news & research source

URL: http://www.physicstoday.org/vol-58/iss-8/p12.html Published: August 2005

[Permission to reprint or copy this article/photo must be obtained from Physics Today. Call 301-209-3042 or e-mail rights@aip.org with your request.]

LETTERS

Validating the Need to Validate Code

The article "Computational Science Demands a New Paradigm" by Douglass Post and Lawrence Votta (PHYSICS TODAY, January 2005, page 35) makes some good points about the pitfalls of believing computed physics. The authors propose several criteria for verification and validation of large computational models for making public-policy decisions. Among other examples, they mention climate-change calculations.

The article's figure 5 shows several famous bridges, including as the third example the infamous "Galloping Gertie," the Tacoma Narrows Bridge in Washington State, which collapsed in 1940. In that case, previous designs were pushed too far, with just one little thing forgotten: resonance! The film of that collapse is still often used in physics classes to dramatize the importance of resonance. Post and Votta "assert that computational science is currently in the midst of" the stage where computing power makes it possible to outrun good engineering judgment.

The Kyoto Protocol to curtail CO_2 emissions was based on a global circulation model from 1994, now a full decade old. It left out the importance of clouds, because they were just too difficult to model. That model was never even successful in accurately describing the past: It violates the first validation criterion given by Post and Votta. Nevertheless, international policymakers have not attended to validation criteria, have never doubted the truth of the model, and have gathered momentum toward implementing the Kyoto Protocol. Consequently, large economic impact is associated with those computational results.

The Galloping Gertie of environmental science is the Kyoto Protocol. Unless computational scientists learn from its shortcomings, it will discredit future attempts to predict climate change.

Thomas P. Sheahen
(tsheahen@alum.mit.edu)
Deer Park, Maryland

The problems that concern Post and Votta were encountered decades ago in commercial software development and have been solved.

A familiar result is Microsoft's Windows operating system, a product of more than 20 years' work by thousands of people. It now consists of more than 200 million lines of

source code. At a smaller but no less impressive scale are numerical analysis programs from such companies as MSC, ABAQUS, Mentor, ANSYS, Dassault, and ALGOR. These firms' codes are commonly used to design bridges, automobiles, networks, buildings, and airplanes. Their development has presented exactly the set of issues Post and Votta describe.

Commercial efforts revolve around solid discipline and management. Perhaps for computational scientists that would qualify as a new paradigm; for software engineers it has become standard practice.

Craig Bolon
(cbolon@verizon.net)
Planwright Systems Corporation
Brookline, Massachusetts

We who work on the system side of high-performance computing development generally think our job is done when the first two challenges that Post and Votta mention, performance and programming, are addressed. I disagree that those two challenges are less urgent than the prediction challenge. I've heard too many complaints about the small percentage of peak that is reached and the dismal state of programming tools. However, from an application viewpoint, I can see that prediction is a formidable challenge.

The article reminds me of a paper I reviewed years ago for *IEEE Computational Science and Engineering*. The author compared several seis-mic processing packages, and each claimed to find oil in a different spot. Apparently, the results were often wrong. Nevertheless, users of the codes blindly trusted them and spent huge investments drilling for oil.

I wonder if the great importance of verification and validation could explain why some industries have not jumped more quickly into computational engineering. For example, one aircraft manufacturer is reportedly going back to using real-world wind tunnels for part of its development stage. It would be interesting to survey computational engineering companies about their experience with early computational technology.

One issue in successful validation is the availability of data for comparison. For example, to validate that an ocean simulator predicts correctly, one would need to place a huge number of sensors, which is probably impractical. So even if more project time is spent validating, I wonder how far the validation could go.

Rudolf Eigenmann
(eigenman@purdue.edu)
Purdue University
West Lafayette, Indiana

I am also concerned with the issues raised by Douglass Post and Lawrence Votta. A great deal of computational physics involves fitting parameters—for example, some turbulence constants and grid design parameters. Fifty physics models put together, each with a couple of free parameters, could yield 100 parameters that can be used to fit the code to whatever verification and validation tests it needs to pass. Yet we know that an

interpolation function fitted to a bunch of points can be wildly wrong between them. This concerns me, before we even start extrapolating the code to regions where its performance is completely untested.

The problem the article highlighted is serious. People who know the issues involved in computational physics are essential. Unfortunately, these days universities turn out users who employ codes as black boxes but do not understand what they do or when their results can be trusted. Moreover, analysis codes are often incorporated into multidisciplinary design optimization algorithms—for example, to design a better aircraft—but the optimization process drives the codes beyond any reasonable applicability. Expert guidance is usually needed to stay within implicit constraints of analysis codes.

Rigorous component testing, although necessary, is not sufficient. Software components can be combined, but their combination could be wrong even though the components test well individually. A combination that is insensitive to minor component errors could still give invalid results. Each component has an unstated region of applicability that is often horribly complicated to describe, yet the combination could unexpectedly exceed individual component limits.

Josip Loncaric
(jl-icase@comcast.net)
Los Alamos, New Mexico

Douglass Post and Lawrence Votta's article admirably stresses the need for validation of computation. If I understand their main point correctly, it's that computation is science, not mathematics.

The article motivates me to contrast two fields I work between that use very different paradigms to describe the electrodiffusion of charge. In computational electronics,[1,2] the electric field is traditionally calculated by solving Poisson's equation with far-field boundary conditions but at relatively low resolution. Poisson is solved anew whenever charges move. Computational chemistry, starting more or less with computer simulation of fluids[3] and computational biology,[4] computes the electric field at high resolution and does not deal clearly with far-field boundary conditions.

Electrodiffusion, which has been at the center of electro- and physical chemistry since Michael Faraday's time, is also at the center of electronics, where it describes the movement of charge in semiconductors and most of our digital devices. Electrodiffusion is no less important in biophysics, where it is responsible for the electrical properties of cells and tissues, and controls many biological functions.

It seems to me that these different treatments of similar physics are distinct and unlikely to be equally precise under all conditions. I hope the article helps motivate workers in each tradition to discuss other treatments beyond their own and try to understand the differences. I hope such workers can determine the conditions under which each treatment is accurate. That way we may learn to use each computational tradition of electrodiffusion only in appropriate situations.

References

1. C. Jacoboni, P. Lugli, *The Monte Carlo Method for Semiconductor Device Simulation*, Springer-Verlag, New York (1989).
2. S. Selberherr, *Analysis and Simulation of Semiconductor Devices*, Springer-Verlag, New York (1984).
3. M. P. Allen, D. J. Tildesley, *Computer Simulation of Liquids*, Oxford U. Press, New York (1987).
4. T. Schlick, *Molecular Modeling and Simulation*, Springer-Verlag, New York (2002).

Bob Eisenberg

(beisenbe@rush.edu)
Rush University Medical CenterChicago, Illinois

Post and Votta correctly point out the need for validation of software programs. Validation is often thought to mean the absolute prediction of measurements. However, new software often replicates measurement trends accurately long before predicting absolute values well. When time is critical, trends are frequently sufficient to guide new designs or predict outcomes. Sometimes the technology is no longer relevant once the model software finally replicates absolute measurements.

R. Casanova Alig
(alig@mailaps.org)
Princeton Junction, New Jersey

The part of the computational science article that uses bridges as examples contains a marginal error regarding the Széchenyi (note the correct spelling) Chain Bridge in Budapest. That bridge opened for traffic on 20 November 1849. Post and Votta write, "Although the retreating Germans blew it up in 1945, the bridge has since been rebuilt according to its original design." But that is not really the case.

The Széchenyi Chain Bridge was completely rebuilt from 1913 to 1915 and is essentially the same today as it was right after that reconstruction. During that rebuild, the biggest changes involved significantly strengthening the pillars and re-placing the chains with new high-strength steel ones. Each element of the new chain was roughly double the size of the original. Both types of elements were on display for many years in the hallways of the Technical University of Budapest and still may be there. A wonderful history of the bridge (in Hungarian but with many pictures) is available at http://www.idg.hu/expo/lanc.

I understand that the points made by Post and Votta do not in any way hinge on the accuracy of this minute detail. Yet it is an opportunity to marvel at the ingenuity of earlier generations of engineers. We owe it to them to get it right.

Denes Marton
(marton@uthscsa.edu)
University of Texas
Health Science Center
San Antonio

Post and Votta reply: The readers who responded to our article have interesting and

valid comments, some of which merit further discussion.

Thomas Sheahen makes several points supporting our premise that computational science's credibility needs improvement. Otherwise, the science cannot inform strategic decisions affecting society. With regard to Sheahen's specific criticisms, climate models have included clouds since the 1960s.[1]

Global warming is a fact. Identifying the causes and predicting future warming are active areas of research. A library search for papers on global warming turned up more than 5000 published since January 2000. The climate-modeling community has recognized that existing models are inadequate and has been working to improve them and to identify and add new effects.[2] The Community Climate System Model and Earth System Modeling Framework programs have been formed to coordinate the national and, to some extent, the international efforts in this area. One of us (Post) attended the 9th CCSM Workshop in Santa Fe, New Mexico, in 2004. There it was evident that the models are being improved, software engineering is becoming a key part of the CCSM program, and the verification and validation process is becoming a central part of model development. The international climate-modeling community has a program with a multibillion-dollar annual budget to gather, analyze, and store detailed data for continental, oceanic, atmospheric, and polar weather and climate phenomena.[3] The newer models generally confirm what the earlier models identified: the importance of CO_2 emissions in global warming. Other predictions of climate and weather phenomena also are making an impact. For example, predictions of El Niño effects are being used to make agricultural decisions.[4] Although the models have undergone tremendous progress and continual improvement, immense challenges remain, and work on addressing them continues.

Craig Bolon's assertion that computational science needs the injection of the project discipline employed by IT organizations like Microsoft is not totally correct. Computational science and engineering will benefit from improved software project-management processes, but not necessarily from the same kinds used in the IT industry. For instance, IT processes emphasize the importance of detailed and prescriptive requirements, but for scientific software such requirements are difficult to develop. Code development for computational science usually involves research to find the best solutions and most accurate models needed for credible answers. It's also not clear that Microsoft Windows is without problems. Windows XP recently had to be massively rewritten to minimize security vulnerabilities. Windows users now download security patches frequently—sometimes daily.

The paper Rudolf Eigenmann refers to is "The T Experiments: Errors in Scientific Software," by Les Hatton.[5] It illustrates the value of "code benchmarking"—comparing the results of many codes for a single problem and determining the reasons for divergent results.

Josip Loncaric expands on two key points in our paper. The first is that developers and users of scientific and engineering codes need considerable domain knowledge to ensure that their results are as accurate as possible. We think this situation is probably getting worse rather than better. The challenge of developing codes for very complex, massively

parallel computers has increased the emphasis on programming skills. As a result, the emerging generation of computational scientists is skilled in code development but much less so in the relevant scientific discipline.

The second point Loncaric highlights is that a model for a natural system—physical, chemical, biological, and so forth—is often much more than the sum of the individual components. For physical systems, Robert Laughlin recently pointed out that much of science today is inherently reductionist.[6] Present scientific research paradigms emphasize the detailed study of the individual elements that contribute to a complex system's behavior. High-energy physics, for example, involves the study of fundamental particles at progressively higher accelerator energies. Yet successful models of complex systems, such as low-temperature superconductors, are relatively insensitive to the detailed accuracy of the individual constituent effects. Laughlin stresses that successful models capture the emergent principles that determine the behavior of complex systems. Examples of these emergent principles are conservation laws, the laws of thermodynamics, detailed balance, and preservation of symmetries.

Since a computational simulation is only a model of nature, not nature itself, there is no assurance that a collection of highly accurate individual components will capture the emergent effects. Yet most computational simulations implicitly assume that if each component is accurate, the whole code will be accurate. Nature includes all of the emergent phenomena, but a computational model may not. This perspective underscores the importance of validation of the integrated code and of individual models.

Denes Marton takes us to task for possibly misspelling Széchenyi, Széchényi, or Szechenyi. Unfortunately, non-Hungarian speakers like us can find all three spellings for the bridge in various sources. We relied on the advice of a Hungarian colleague, who has admitted, on further inquiry, that Széchenyi is likely correct.

All the English-language accounts we could find mentioned the original bridge construction in the 1840s and the reconstruction after World War II. We don't doubt Marton's additional historical elements about the reconstruction in 1913–1915, but we couldn't find an English account of them.

References

1. C. Jacoboni, P. Lugli, *The Monte Carlo Method for Semiconductor Device Simulation*, Springer-Verlag, New York (1989).
2. S. Selberherr, *Analysis and Simulation of Semiconductor Devices*, Springer-Verlag, New York (1984).
3. M. P. Allen, D. J. Tildesley, *Computer Simulation of Liquids*, Oxford U. Press, New York (1987).
4. T. Schlick, *Molecular Modeling and Simulation*, Springer-Verlag, New York (2002).

Lawrence G. Votta
(larry.votta@sun.com)
Sun Microsystems Inc
Menlo Park, California

copyright © American Institute of Physics