Newsjournal of the Society for Industrial and Applied Mathematics

siam news

Volume 42/ Number 5
June 2009

Challenges and Opportunities for Progress in CSE

It's a rare talk at a SIAM conference that begins with a discussion of the philosophy of science in general, and Karl Popper's principle of falsification in particular. J. Tinsley Oden, in an invited talk that set the scene for much that would follow at this year's SIAM Conference on Computational Science and Engineering, made the point that simulation-based engineering science, and the closely related CSE, are new areas of science that fit within Popper's general framework for the creation of scientific knowledge, with some differences. The scientific method, Oden pointed out, builds on observations, typically from experiment, to form theories that are tested by further observations. A hypothesis, by the popperian principle of falsification, becomes a legitimate scientific theory only if it can be refuted by observational evidence.

Shown in Miami at the 2009 SIAM Conference on Computational Science and Engineering are SIAG/CSE officers (from left) Kirk Jordan (vice chair), Tammy Kolda (chair), Carol Woodward (secretary/treasurer), and Uli Rüde (program director). Two of them—Jordan and Woodward—chaired the organizing committee for the conference, and all helped with coverage for SIAM News, including the accompanying article on Tinsley Oden's invited talk; a collection of articles on education-related conference sessions can be found in the May issue. Conference photos by Susan Whitehouse.

Clearly, Oden considers CSE an emerging area, "an important discipline in its own right," with far-reaching implications for all of science and technology: "CSE is going to permeate every area of science and technology, enriching applied math in the process."

Simulation-based engineering science (SBES) and CSE, used almost interchangeably by Oden (and in this article), make possible the discovery of things that cannot be observed directly and allow the testing of multiple hypotheses. (The two reports mentioned at the end of this article offer a good perspective on SBES.)

Today, Oden said, "epochal changes are occurring in science and engineering"—a reference in part to rapidly advancing computational capabilities—computing at the petascale now and at the exascale in the not-too-distant future—and in part to emerging "super algorithms" that enable performance improvements greater than those provided by computing power alone. These advances are making possible serious computational modeling and simulation of complex real-world systems. Mathematical models were long thought to provide *at most* qualitative information, Oden said. That has changed significantly in the last decade: People have begun to expect quantitative results, and in particular predictions whose uncertainty is quantifiable.

Validation and verification (V&V) and uncertainty quantification (UQ) were major themes both of Oden's talk and of the SIAM conference (which included a well-attended four-hour tutorial on techniques for UQ). Here again, Popper came to the fore. Parameters in models have errors, from uncertainties in measurements, among many sources, and variation. Just as a model (or a theory) cannot be proved, but only refuted, we cannot prove that a code (with specific parameters) provides the "correct" answer; rather, we must calibrate the models, by comparing model output with observed measurements, relying also on an a posteriori probability distribution function that characterizes the uncertainty in the answers.

Threaded throughout Oden's talk was a discussion of barriers to and opportunities

See **Progress in CSE** *on page 6*

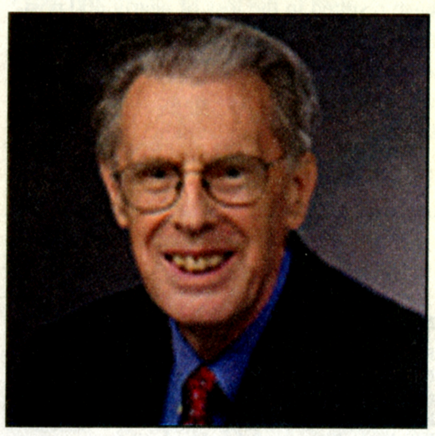
National Academy of Sciences Elects New Members

The U.S. National Academy of Sciences, created during the presidency of Abraham Lincoln to advise the federal government "upon request, in any matter of science and technology," held its annual meeting in Washington at the end of April. On the agenda, as always, was the election of new members. Representing SIAM among the 72 newly elected U.S. members are John Hopcroft of Cornell University, Thomas J.R. Hughes of the University of Texas, Austin, and Gilbert Strang of MIT.

A computer scientist, Hopcroft has been at Cornell since 1967. Apart from a foray into administration—mainly as dean of engineering, from 1994 to 2001—he has been a member of the computer science department, where he is currently the IBM Professor of Engineering and Applied Mathematics. Most recently, he has been studying information capture and access. Hopcroft chaired the SIAM Board of Trustees from 1989 to 1995; also a longtime member of the Financial Management Committee, he has helped keep SIAM fiscally stable through times both prosperous and turbulent. He was elected to the National Academy of Engineering in 1989 "for fundamental contributions to computer algorithms and for authorship of outstanding computer science textbooks."

Hughes is also a member of NAE, having been elected in 1995 "for contributions to the development of finite element methods for solid-structural and fluid mechanics." A professor of aerospace engineering and engineering mechanics, he is currently a member of the Institute for Computational Engineering and Sciences, known as ICES, at UT Austin. Among his recent research interests are variational and multiscale methods in studies of turbulence, in particular in large-eddy simulations—the subject of an invited talk he gave at the second SIAM Conference on CSE—and patient-specific cardiovascular modeling and simulation. His work has been recognized by many awards, including the 1997 John von Neumann Medal of the U.S. Association for Computational Mechanics, of which he is a founder, fellow, and past president.

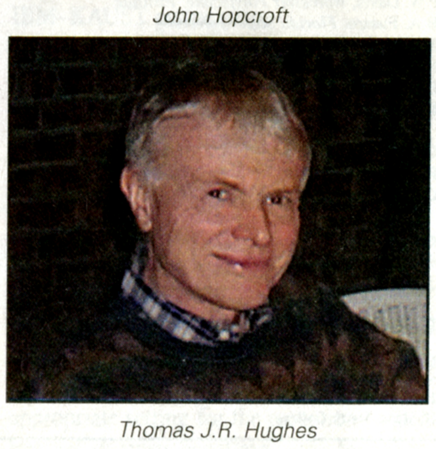
A professor of mathematics at MIT, Strang expressed thoughts about his books on linear algebra and computational science in a recent interview with *SIAM News* (April 2009). Through MIT's OpenCourseWare project, he has made lectures in those fields freely available to anyone in the world. As president of SIAM (1999–2000), he introduced applied and computational mathematicians worldwide to SIAM programs and ideas. The first Su Buchin Prize, which he received at ICIAM '07 in Zurich, recognized him in part for his visits to Asia (of which a tangible result is creation of the East Asia Section of SIAM), as well as to developing countries in other parts of the world. Among his other awards are the Henrici Prize, also presented at ICIAM '07, and the 2005 USACM John von Neumann Medal in computational mechanics, in recognition of his book (with George Fix) on finite elements.



John Hopcroft

Other mathematical scientists among the new members include Sun-yung Alice Chang, a professor of mathematics at Princeton, Percy Deift, a professor of mathematics at the Courant Institute of Mathematical Sciences, New York University, John Morgan, a professor of mathematics at Columbia, and Christos Papadimitriou, a professor of computer science at UC Berkeley.

The newly elected members were notified a day too late to be present for another highlight of the NAS meeting: President Barack Obama, addressing the members in attendance, announced new initiatives and investments in scientific research—with top priority for alternative energy and climate change—and in education. Along with support for high-risk, high-return research and for research by young scientists and engineers, he affirmed his administration's commitment to doubling the budgets, over the next ten years, of the National Science Foundation, the National Institute of Standards and Technology, and the Department of Energy's Office of Science. He also announced the creation of the Advanced Research Projects Agency–Energy—a DOE version of the Defense Department's DARPA. Only the fourth U.S. president to address the National Academy, Obama called for increasing the percentage of the federal budget allocated to R&D to 3%, from the current 2.7%.



Thomas J.R. Hughes

Gilbert Strang

Nonprofit Org
U.S. Postage
PAID
Permit No 360
Bellmawr, NJ

siam.
SOCIETY for INDUSTRIAL and APPLIED MATHEMATICS
3600 Market Street, 6th Floor
Philadelphia, PA 19104-2688 USA

*************************SCH 3-DIGIT 946
S49 P10
0010565?8 SINEWS/4205 00855
FERNANDO PEREZ
4508 FAIR AVENUE
OAKLAND CA 94619-2927

CSE 2009

Python for Scientific Computing at CSE 2009

By Fernando Pérez, Hans Petter Langtangen, and Randy LeVeque

Python for Scientific Computing, a three-part minisymposium we organized for CSE 2009, featured a mix of speakers from universities, government research laboratories, and industry who explained why they chose the open-source Python programming language and showed how they use it in their research and teaching. The sessions, all held on March 5, were well attended, as was a similar three-part minisymposium at last summer's SIAM Annual Meeting.

A Flexible Tool

Over the last few years, Python has experienced tremendous growth, becoming the tool of choice for many who do high-level scientific computing. It offers an effective mix of interactive and exploratory development approaches, direct access to libraries for many different tasks, and interfaces with high-performance numerical libraries in Fortran, C, or C++. Along with a few free add-on packages, Python provides basic capabilities similar to those offered by computational environments like MATLAB or IDL: numerical arrays with syntactic support for arithmetic and mathematical operations, a comprehensive library of common algorithms (in linear algebra, FFT, numerical integration, optimization, and special functions, among other areas), interactive control of data visualization, publication-quality plotting, and modules for interfacing with codes written in numerous other programming languages. What attracts many scientists to the language is its combination of flexibility, expressive power, and development possibilities that, in our experience, is unmatched by commercial tools.

Python was designed as a general-purpose language with an emphasis on a clear and readable syntax, high-level constructs that would not impede access to low-level resources, robust error handling, and portability. It ships with a comprehensive standard library that covers many common tasks, from text processing to network protocols or database access. Other free and open-source projects support most common computational needs, and provide bindings for use of a wide collection of Fortran, C, and C++ libraries. Today, Python is used extensively in industry, by companies like Google (the employer of Python's creator, Guido van Rossum), and in most U.S. federal research agencies, as made clear by the inclusion of speakers from several national laboratories and the National Institute of Standards and Technology in the minisymposia. Python requires no licensing fee and hence there are no license-manager hassles (important considerations for those running parallel codes on hundreds or thousands of processors or using cloud computing services). Highly portable, it can run on a cell phone or on the largest of supercomputers.

Research in Scientific Computing

The minisymposium speakers covered both general-purpose tools and domain-specific projects (all materials are available online*). One group of speakers focused on interactive, exploratory computing and data visualization: Fernando Pérez of UC Berkeley discussed the IPython system of components for interactive computing, and Brian Granger of Cal Poly San Luis Obispo covered IPython's applications for high-level parallel computing as well as the design of distributed data structures. John Hunter of TradeLink, the lead developer of the matplotlib plotting package, presented an overview and hands-on demonstration of the project, whose goal is to provide exceptionally high-quality two-dimensional plots (see Figure 1).

Hank Childs of Lawrence Livermore National Laboratory followed up with a discussion of VisIt, a package for the analysis and visualization of large-scale three-dimensional data sets, such as those produced via adaptive mesh refinement on massively parallel machines. VisIt has its own GUI interface, along with a Python interface that makes it particularly easy to integrate with other programs. Figure 2 shows a visualization done with VisIt in which the high-quality VTK 3D graphics library combined multiple rendering options to display elevation data for Mount St. Helens.

Speakers in another group covered Python tools for high-performance computing. Andreas Klöckner of Brown University demonstrated the easy access provided by his PyCuda library to the capabilities of modern high-performance graphics cards for numerical computing. Pearu Peterson of the Institute of Cybernetics at Estonia's Tallinn University of Technology presented his research on the algorithmic and data structure problems involved in the design of sympycore, a fast library for symbolic computing in Python. The mpi4py library, which permits the development of MPI codes in Python, was covered in a presentation by Lisandro Dalcín of the Argentinean CIMEC research laboratory and Brian Granger. With mpi4py, MPI primitives can be called in pure Python, and C or Fortran MPI codes (or even both in the same process) can be accessed directly, often with minimal performance loss.

Tony Drummond of Lawrence Berkeley National Laboratory described the PyACTS project, which provides Python interfaces to the ACTS collection of high-performance codes (Aztec, Hypre, PETSc, SLEPc, ScaLAPACK, SUNDIALS, SuperLU, TAO, and OPT++). Similar efforts of this type include PyTrilinos and petsc4py for use of Trilinos and PETSc from Python. Jon Guyer of NIST described the architecture of the FiPy project, a finite volume PDE solver developed by his group. With an easy-to-use syntax for model description, FiPy exploits Python tools like NumPy, SciPy, matplotlib, and PyTrilinos to tackle problems in materials science.

Aric Hagberg of Los Alamos National Laboratory presented the NetworkX project, a library of algorithms for studying complex networks; a good illustration of Python's strengths, the project couples an algorithmic core with rich functionality to multiple visualization libraries that provide alternative means of looking at networks. With Python, visualization systems written in other languages can be used to render results from NetworkX via a unified interface.

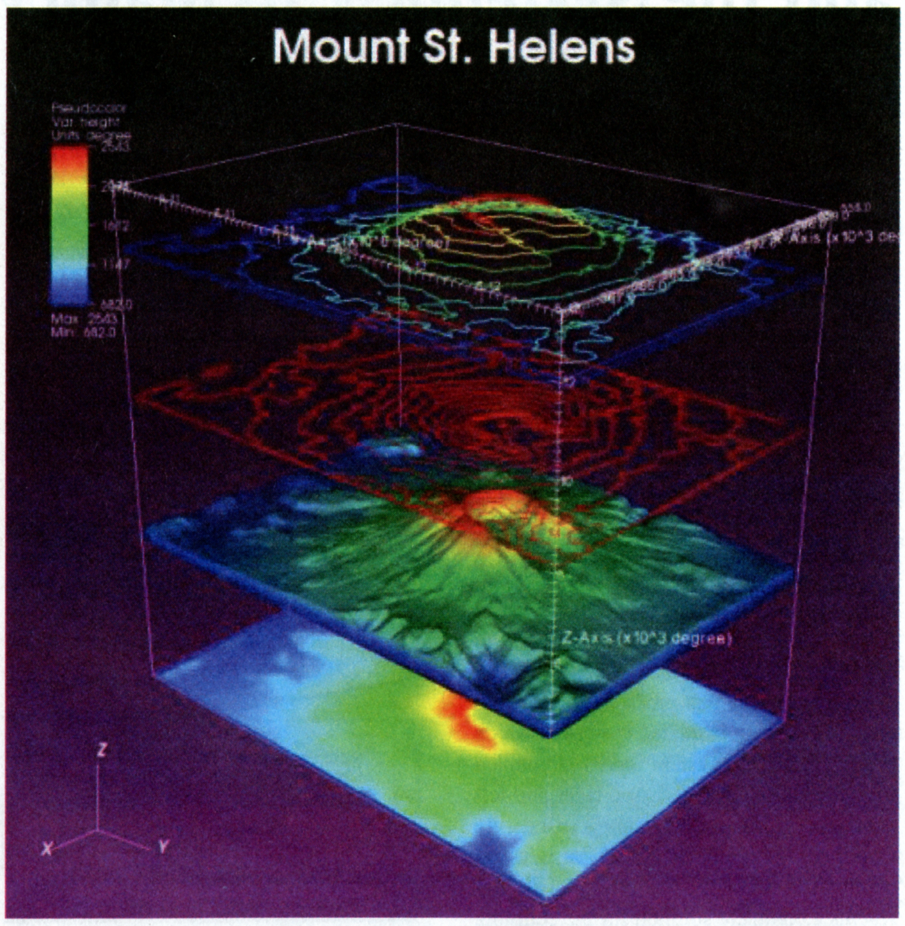


Figure 2. *Topographic visualization of Mount St. Helens performed with Lawrence Livermore Lab's VisIt software.*

CSE Education

Python is not only a research tool: The day was bracketed by two presentations illustrating the use of Python in scientific computing education. Hans Petter Langtangen of Norway's Simula Research Laboratory described the University of Oslo's implementation of a major reform of computational science teaching using Python as its foundation. Students learn Python and numerical methods in their first semester and apply these tools in a range of science courses across the university.

Toward the end of the day, Joe Harrington of the University of Central Florida described a 2007 attempt to replace IDL with Python in his course on astronomical data analysis, with poor results because of documentation issues. He responded by organizing and funding the SciPy Documentation Project during the summer of 2008. Results were dramatic, he said in Miami: Students in his fall 2008 class learned more in less time than students in the IDL-based class.

Python's use in education is growing. In the U.S. its impact can be seen at both ends of the spectrum: The National Science Foundation-funded SECANT project supports the development of a Python-based curriculum and workshops for interdisciplinary computational science education; the One Laptop Per Child project provides economically disadvantaged children with laptops loaded with Python-based software that they can inspect, learn from, and modify.

Open Tools and Reproducible Research

Throughout the SIAM CSE conference, speakers emphasized the growing awareness that computational research must be truly reproducible. Included in the registration packet of everyone who attended the conference, in fact, was the January issue of *Computing in Science and Engineering*, a special issue focused on reproducible research.

We believe that Python is an excellent platform for building a work flow in which every step can be validated and reproduced by anyone. Because it is open source and available at no cost, there are no financial barriers to its use and all of its internal components are open to inspection. By using open-source tools to build our computing foundation, we can facilitate this core principle of the scientific process in our discipline; accordingly, every project discussed in the minisymposia in Miami is freely available for all to download, use, verify, and improve. We hope that the community of scientists who build tools in this manner will continue to grow. The Python projects for scientific computing are rapidly maturing, becoming better integrated and documented, more powerful, and easier to use. Join us!

Readers interested in learning more about these tools can visit the SciPy Web site (http://www.scipy.org/), which hosts some of these tools and contains links and information to many related projects.

Fernando Pérez is a research scientist at the University of California, Berkeley. Hans Petter Langtangen is director of the Center for Biomedical Computing at the Simula Research Laboratory, in Norway. Randy LeVeque is a professor of applied mathematics at the University of Washington, Seattle.

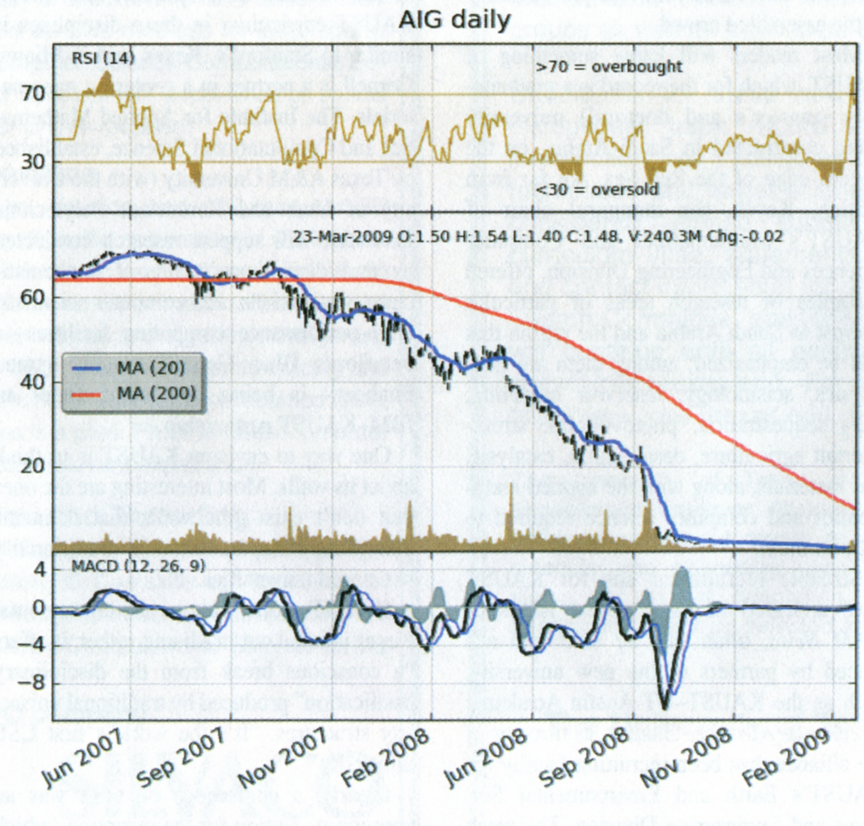


Figure 1. *AIG's recent stock performance, summarized by matplotlib.*

*https://cirl.berkeley.edu/fperez/s/cse09/.