

Let's Remember: The Noun is "Science"

Robert M. Panoff, Ph.D.
President, Shodor Education Foundation
Director, National Computational Science Institute

A Contributed Editorial to HPCWire, 1 July 2005

For several weeks I have been trying to digest the recent report by the President's Information Technology Advisory Council (PITAC) on computational science whose publication was punctuated by the nearly simultaneous dissolution of PITAC itself. In the same spirit that even a broken clock is right twice a day (well, at least the old analog clocks were!), maybe the President did us all a favor by making it clear that advice about information technology isn't advice that the administration wants to hear, at least not this advice. I want to spend as little energy as possible pointing out the shortcomings of a report that has fallen on deaf ears. Instead, let's open a fresh conversation among ourselves, with our colleagues, and even with our fellow citizens about the real roadmap for computational science and its potential to transform science and mathematics education for all.

Over many years, the high performance computing community in my opinion has lost its focus, because it has forgotten that the noun in "computational science" is "science." As supercomputer centers matured, those whose leadership failed to keep the quality of science as their driving motivation lost some of their relevance and some even closed. Those centers that kept science at the forefront fared better. The evaluation of the NSF advanced computational infrastructure program suggested that while there was still a need for well-run "cycle shops," the leading-edge high performance computing centers as then constituted, for the most part, were not likely to lead scientific teams to produce significant advances in science. They had gotten too caught up in boxes and wires, and not what the boxes and wires could do. And as budget priorities have changed, there hasn't been a huge outcry even from the science community because so much of computational science is being accomplished day in and day out at a scale that does not call for the biggest machines and a few centers. Consider one piece of evidence that suggests the PITAC authors missed the chance to make their case for a broader impact of computational science: the most compelling science challenges that face us—challenges that do, in fact, justify a national effort at the large end of the spectrum—were relegated to the appendices in the report. Their main recommendation is to sustain software centers, not science. As Stan Lee would say, "Nuf said,"

Dan Warner, a professor of mathematical sciences at Clemson University and one of the co-founders of Shodor, a national resource in computational science education, recently put the situation very clearly. In considering the vast oceans of data that are being generated by a variety of observational laboratories, he observed, "It isn't whether we have more chips processing the data, but whether we have more neurons." We need

many more people engaged in the conduct of science, and computational science is a wonderful way to bring people into science.

Our challenge is to see that computational science education is a most effective means for addressing a larger issue: quantitative reasoning. In simple terms, we still have to ensure our children actually grow up knowing how to compare quantities, even if it isn't being tested anymore by the SAT! In a relevant context of measuring and comparing, of observing and conjecturing, students need to master fractions, decimals, percents, and ratios, and reading and interpreting graphs. Not through repeated testing, but through minds-on exploration. With an administration caught up in sloganeering of "No Child Left Behind" (which is really "No Child Allowed to Get Ahead" don't get me started. . .) we have found that computational approaches to science education (the effective use of computational tools and visualization to teach the concepts of math and science) is as important or more important to stress as education in 'computational science' (teaching the process of building and testing a numerical model).

As reported in most major papers last week, it seems programming has lost its luster. Modeling the world hasn't. Our computational science classes at Shodor for middle school and high school students (<http://www.shodor.org/succeed>) are in full gear now, and the students learn everything from systems dynamics, to agent-based modeling, to data analysis, and visualization. But the focus is not the computer, but how the computer can help one learn about the world. Students want to focus more on content driven disciplines. And that is the strength of computational science, because modern math and science are more about pattern recognition and characterization than mere symbol manipulation. The tools of computational science can open up avenues of exploration for students in ways that even direct observation can't. The observation is paramount, but the observation is made in the context of a scientific model that is implemented on the computer. The science is at the heart of computational science.

For several years now, I have been sharing with faculty and teachers a simplification of the process referred to as "the scientific method." Basically, we can boil down the process of science –the acquisition of sure knowledge– to four basic questions:

What can I observe?
What can I learn from these observations?
How sure am I that I am right? and
Why should I care?

A well-balanced experience with an interactive model, or an exploration and visualization of a dataset can go along way for teaching the process of science with added benefit that more students will actually want to be scientists.

One approach to bring computational science to the masses is by enlisting the help of many to assist in the overwhelming task of analyzing the overwhelming data being generated by a number of space and land-based projects, from star surveys to earthquakes, to census data, to on-line archives of historical records. By incorporating the

exploration of real data –and there is so much of it yet to be explored– as part of the learning of math and science starting in the middle grades through high school and college, we can make education an adventure for the whole human race. Unfortunately, we have many math and science teachers at the elementary and middle school levels who choose to be teachers at this level because they “don’t do math!” Significant work to incorporate models and computational tools into the math education of many students has started to show its benefits, by easing some of the math anxiety, and showing how the math makes sense. Some materials also show how to seamlessly incorporate these tools into existing curricula in support of standards (see: <http://www.shodor.org/interactivate>). For these approaches to become more widespread, it will take a wholesale change in schools of education in the pre-service preparation of math and science teachers, which means a massive change in the attitudes of faculty in the sciences and in education.

Computational science is both content and method. Students should know the basics of the tools of computation, but also use computation to learn the basics of chemistry, biology, physics, and engineering. So many of the texts in use at all levels are wholly lacking. At the very least they fail to accurately communicate that much of what we know in the sciences is from computational models as much as from direct observation.

We have a long, long way to go. Eric Jakobsson, returning to Illinois from his tour of duty at the National Institutes of Health, reported that several years into the ten-year plan of the National Institute of General Medical Sciences (NIGMS) little progress has been made to open up an education and training pathway that integrates at every level physical science, mathematics, and computation with the learning of biology in a problem-solving environment. As a result, perhaps proving that merely having a ‘roadmap,’ does not guarantee success, he related that we are not training American biology researchers with quantitative skills at even close to a rate to sustain, let alone advance, American biology. Some of that biology requires “big iron” to manage exponentially growing databases; most of biology requires computational science that uses those databases remotely.

The same can be said for chemistry. Only ten years ago, without a supercomputer, no leading chemist could do significant computational chemistry in a reasonable amount of time; now most of the packages for much of the chemistry can be done without recourse to the biggest machines, and real computational chemistry can be part and parcel of every undergraduate –even high school– chemistry course. For instance, the Burroughs Wellcome Fund has recently awarded a grant for a computational chemistry server to be housed at Shodor so that North Carolina high school students would have precisely this resource and experience. True, there are significant problems that require massive computing, but there are many more problems that are more relevant to the education and training of computational chemists that don’t. Thom Dunning, a renowned chemist who as its new director has taken up the challenge of restoring the “applications” focus of the National Center for Supercomputing Applications, has set an even more challenging goal of incorporating computational chemistry into every undergraduate chemistry course at the University of Illinois, not just for a select few who may be computational chemists.

Even at the highest level, computational resources are limited and aren't "there yet." For instance, it would take about a dozen years on the fastest existing supercomputer merely to initiate the computation for a drop of water at the molecular level, that is, to assign initial values for each of the spatial components of position, velocity, and acceleration for each molecule in a single drop. That doesn't mean we shouldn't try to solve large problems, it's just a measure of how far we have to go.

So back to reality. If we keep thinking that computational science is only for the biggest problems, then it affects only a few who would be given limited access to limited resources concentrated in a few national centers. If that is the only way that "real science" will get done, we will never convince a doubting Congress the second time around, let alone an administration –that may not realize that only one of the three R's actually begins with "R"– of the relevance of computational science. To justify an appropriate appropriation for a long-range road map, we have to have a more wide-reaching goal of computational science for everyone at all levels, and that means developing an effective computational approach to science education as well as an effective education in computational science.