Pulling Teeth: Why Advanced Technology Is Not Prevalent in the Classroom[¥]

Paul A. Gray
Department of Computer Science
University of Northern Iowa
Cedar Falls, IA 50614-0506
gray@cns.uni.edu

Abstract

The state-of-the-art in scientific computing has grown considerably over just the last decade. A commensurate advance in operating systems and programming paradigms has followed advances in computational hardware capabilities. Built upon this evolving curve of technological progress is a multitude of applications that have allowed us, as researchers, to model phenomena, visualize results, mine data, and interpret a wide breadth of computations in minimal time. The importance of developing and providing a technologically based curriculum at all levels is abundantly clear. When used intelligently, the incorporation of scientific tools in the curricula can advance learning and can lead to profound changes in both our teaching and in students' learning.

There are many obstacles that confront us, as educators, when bringing these advanced technologies to our classrooms. Some of these roadblocks are political, some physical, others financial or perhaps logistical. While these obstacles are easy to categorize, the most egregious encumbrances arise from *personal* influences. This paper discusses many of the common obstacles that we face in infusing the current curriculum with scientific tools. A specific experience in establishing a dedicated scientific-tools lab for students and instructors at UNI is an exemplar where appropriate.

Introduction

As researchers, the area of computational science has become for us a cornerstone; as educators, chalk and blackboards remains our daily staple. Network and workstation technologies are becoming widely available for our use in the classroom, and we are confronted with vastly untapped educational resources that go far beyond today's status quo of overheads and graphing calculators in the calculus courses. The foundation of this report is built around the importance of building technological content for incorporation into undergraduate curricula and K-12 instruction. The overall goal is to develop materials in concert with the "College of the Future," program (COTF[1]) and the NASA-supported "Standards for Technological Literacy[2]" as published by the International Technology Education Association.

A long-term objective is to establish technologically enhanced interdisciplinary curricula taught by educators proficient in these tools, and to establish materials for researchers and

[¥] This work supported in part by the Iowa Space Grant Consortium, Phase-I funds. Contact: Iowa Space Grant Consortium, 2271 Howe Hall, Room 2365, Ames, IA 50011-1667.

educators alike to turn to for classroom- and research-related resources.

In short, the objectives that guide our increased use of scientific software are consistent with the missions set out by NASA's Center for Educational Technologies, summarized here:

- To develop high-quality, technology-intensive curricular supplements;
- To incorporate in these supplements current research findings on the effective application of technology to educational settings;
- To deliver model, in-service programs that improve the technological competence of educators.
- To provide networked tools that ensure equitable access to the developed resources by all learning communities.

To this end, our institution was permitted the resources to establish a “scientific computing laboratory,” small in scale but extremely broad in scope. This laboratory is intended to serve as an important instructional tool, used for exposing students to the high-quality scientific software that is available today. In addition, the laboratory is to be specifically used to instruct faculty and secondary teachers in the use of scientific software for instruction.

The difficulties and successes associated with the implementation of the scientific laboratory known as the “*Storm Lab*” at the University of Northern Iowa is described in the remaining portion of this composition.

Difficulties and Issues Surrounding the Increased Use of Scientific Tools

There are many obstacles and roadblocks to infusing the use of scientific tools into the undergraduate curriculum. The establishment of the “*Storm Lab*” was aimed at providing the tools necessary for the task at hand. The *Storm Lab* puts a vast array of scientific tools at the disposal of the students and instructors. But would these tools be used?

One of the first obstacles that arose was *political.* A political impediment arises, for example, when colleagues' opinions differ greatly in the best “tool” to illustrate a specific topic. For example, should “PVM” or “MPI” be used to illustrate parallel programming concepts, or would *Maple* or *Matlab* be better suited as an overall visualization tool in the calculus sequence. Personal preferences often dominate and seldom wane. However, the outcome is often a diversity of tools used instead of an impasse.

Other limitations that arose could be described as *physical*. Physical limitations might include the fact that the room where a class meets lacks projection capabilities or network access, so the appropriate “tool” for the course might not be supported by the physical properties of the environment.

Related to the physical aspects are the “*logistical*” issues. Logistic issues might arise when considering where to physically locate the computer equipment that supports these tools. This also encompasses often-overlooked issues such as heating, cooling, power, and access to the equipment. To minimize these concerns, the *Storm Lab* began small.

The initial configuration consisted of four PC workstations, and was placed in a mathematics tutoring area where there was sufficient room and an established reasonable access policy.

Financial limitations also contributed to the size of the initial configuration. Financial issues are easily seen, for example, when one compares prices for multi-user software licenses with one's departmental budget. With scientific software costs ranging from free to several thousand dollars per license, having four workstations in the initial configuration meant that the lab could be uniformly configured and reasonably well stocked with software titles.

The problems associated with "*administration*" include not only who would purchase and maintain the software licenses, keep the software versions up to date, and make sure that the software functions as it should, but also who would configure the network, setup user accounts, review the logs, and related support-related tasks.   These are two distinct responsibility levels and are best managed as such.

While these obstacles are relatively "quantifiable," other barriers come into play that are more elusive and have just as much potential to limit success. These are colleagues that fear that incorporating technology into the curriculum would mean additional class preparatory time. These are the network and system administrators that are less than enthusiastic to configure and maintain an appropriate environment.   In the process of implementing the *Storm Lab*, apprehensions and individuals' reluctance to change presented *personal* obstacles that were just as detrimental to the implementation as any of the more quantifiable difficulties describe above.

A Workable Solution

In the implementation of the *Storm Lab*, logistical and financial limitations were foremost considerations. This would indicate that an initial configuration would be small in scale. Thus, planning of the laboratory began with four workstations, which was ideal for the space available. The costs were minimized by recouping four of the most capable workstations from a large pool of computers that were slated for replacement. These four workstations consisted of 350MHz Pentium-III class processors with 17 inch monitors and sufficient hardware configurations to support most scientific software packages. Money that was saved by adopting modest-performing workstations at little expense would in turn be used to purchase additional software and networking equipment.

With the equipment and location secured, approval for attaching to the University's network was in order. The nature of the *Storm Lab*'s usage and software licensing restrictions indicated that a self-contained network that was administrated by the department would best support the lab's functionality. The prospect of such a self-contained networked configuration attaching to the University's network infrastructure had not been suggested before. As such, a significantly detailed layout of administration, access, and network policies needed to be documented and reviewed by the University's technology experts. In the end, as the proposed networking approach was heretofore untested, the lab was provided access to a dedicated, provisional experimental network.

In the end, targeting a small number of workstations for the initial configuration was ultimately beneficial. With the objective of establishing a homogeneous distribution of software packages across the four machines, having only four workstations permitted the purchase of more individual software packages for all machines. Additionally, four workstations were readily integrated into an existing room so that the concerns over space, power, cooling, etc. were easily addressed.

Addressing More Difficult Issues

With the *Storm Lab*'s hardware, network, administration, and software set up, all that was needed was a group of users. The quantifiable issues had, for the most part, been addressed. There was ample interest in the laboratory by students, but colleagues seemed less than eager to devote time learning new ways to present traditional course content[§].

Textbooks abound that are devoted to integrating scientific tools into the content of the current curriculum. However, these textbooks are not being universally embraced by departments for the core courses or by colleagues for upper-level courses. The reasons for this appear to be:

a)The instructor is not familiar with the specific tools being utilized by the author of the text. For example, the author might choose Java to illustrate algorithms, whereas the instructor prefers Fortran. Or perhaps a mathematics textbook author would provide examples in Maple, whereas the instructor would prefer to use Mathematica. In this case, the instructor would not adopt the text because it would require learning a new tool, and a significant increase in preparatory work for the course.

b)Almost the direct opposite of the above consideration is the situation where the author attempts to cover the same topics with every conceivable tool. For example, a numerical analysis book that includes example codes in C, Fortran, Java, etc., or, in the context of the *Storm Lab*'s repertoire, a textbook that includes identical examples for Matlab, Mathematica, and Maple, and C lacks fails to leverage upon the inherent powers of specific tools. That is to say, that individual software packages have strengths and weaknesses, and diluting the examples down to a common functionality of all packages neglects the use of the *appropriate tool* for the *appropriate application*. In this situation the instructor is left to augment and expand upon the supplied examples, which often is more arduous of a task than starting from scratch.

c)The third situation is where the instructor is simply reluctant to adopt a textbook on that incorporates the use of scientific tools due to a reluctance to devote additional time to preparations for the course.

Note that the common theme that runs through the above bulleted items is reluctance on the instructor's part to devote a large amount of time in order to hone the tools of the textbook into modules that would be usable for the course. The difficulty then, is to allay the reservations of the instructor, while building curriculum based upon scientific tools that are expressly designed for illustrating the individual topics.

Students, on the other hand, are more open to devoting time to learn new tools, and

[§] It should be stressed that the objective of the *Storm Lab* was to *infuse* the current curriculum with the appropriate use of scientific tools, and not to revamp nor rewrite the curriculum with a scientific tools focus.

acclimate well to experimentation with new technologies. A natural solution would then be to pair up the enthusiasm of the students with the classroom savvy of the educators. The natural inquisitive interests of the students could be directed by the instructor to develop classroom modules, hand-in-hand, in a symbiotic manner. This premise is currently being investigated.

With the aide of the Iowa Space Grant Consortium[3], the development of classroom modules specifically aimed at the undergraduate curriculum common to majors in the natural sciences and education. Students learn and develop software modules tailored to the needs of the instructor. At the present, this provides instructors with a vested interest in the scientific tools and classroom modules that are being developed by the students.

Summary

This document has described our approach to overcoming the reluctance of many instructors to invest time in developing classroom modules centering upon the use of scientific tools. These efforts are specifically focused on infusing the current curriculum with the use of scientific tools as opposed to focusing on rewriting the curriculum to suit the use of scientific tools.

The preliminary results show great potential with enthusiastic students and openness of faculty members to work together with the students. At the present time, students have been arranged to work within calculus, differential equations, and linear algebra courses. Further efforts in arranging student/instructor relationships for physics, chemistry, biology, and computer science are expected to materialize as our work progresses.

Availability

As our course modules are developed, our efforts will be released under the Open Source license, and publicly accessible through http://www.stormlab.uni.edu. The anticipated date for which the web site will be accessible is no later than May, 2001.

[1] COTF: College of the Future, NASA's premier research and development program for educational technologies at the Center for Educational Technologies. http://education.nasa.gov/cotf/index.html

[2] Standards for Technological Literacy, by the International Technology Education Association. April 2000. ISBN: 1-887101-02-0.

[3] The *Iowa Space Grant Consortium*, http://www.public.iastate.edu/~isgc