Computational Software in Undergraduate Mathematics and Beyond – A Proposal for the 1804 Fund

Steve Chapin, Larry Snyder, and Todd Young
Department of Mathematics
Ohio University

April 12, 2000

Abstract

The invention of computational software (**CS**) over the last two decades is a very significant development in the use of mathematics. We propose to introduce **CS** into the mathematical life of Ohio University by making a particular software package, **MatLab**[1], widely available across the University and by introducing **MatLab** homework into most of the undergraduate mathematics curriculum. With this training and campus-wide availability of the software, **MatLab** will be a tool that students can use throughout their careers at Ohio. This project is based on book written by Young.

[1] **MatLab** is the registered trademark of The Math Works Inc.

1 Introduction

Computational software (**CS**) is characterized by its capability of both symbolic and numerical calculations. An example of a numerical calculation is solving the algebraic equation $x^2 - 2 = 0$ by giving the answer: $x = \pm 1.414213562\ldots$. On the other hand, a symbolic calculation can solve $x^2 - b = 0$ with the answer: $x = \pm\sqrt{b}$. Important computational software packages include **MatLab**, **Maple**[2] and **Mathematica**[3]. This technology revolutionizes the way that people use Mathematics. Experts in mathematics education write that "..., it has become increasingly evident that the technology altered the nature of the activity using it" [6].

We propose to introduce homework using **MatLab**, into the calculus/differential equations sequence (MATH 263A,B,C,D and MATH340) at Ohio University and to make the program widely available on campus. Our eventual goal would be to extend this initiative to most of the freshman and sophomore level mathematics courses at the University. These courses are taken by a large portion of students at the University. Once most freshmen and sophomore mathematics classes are using **MatLab** and the software is widely available on campus, the software will become a part of the culture of the University, professors from any department may take for granted that students have knowledge of the software, and students may use the software on their own for any math-related course-work. The goal is to introduce **CS** not only into the undergraduate mathematics curriculum, but to make it an integral part of mathematical life at the University.

While Ohio University is making very bold steps toward providing our students with computer experience, we are behind many universities in our use of (**CS**). The existence of a well-developed project, including a book by Young [8], puts Ohio in a unique position to become a leader in the use of **CS** in undergraduate mathematics and at the same time dramatically influence the use of mathematics around the University.

Young, who has had extensive experience using computers in teaching mathematics at other institutions, is writing a book about using programs like **MatLab** for homework in mathematics courses. His book promotes a "simple approach" to using computers which is distinct is several ways. We propose to introduce Young's approach into most undergraduate mathematics courses, beginning with calculus. A key feature of the approach is that individual instructors are able to use it with very minimal time investment.

2 Current use of technology in the Mathematics Department

Currently the Math Department has licenses for both **MatLab** and **Maple**. Several other departments also have site licenses for **MatLab**. Most notably the College of Engineering has licenses for over 100 concurrent users. These programs have been used sparingly for teaching purposes in the Math Department. Professor Snyder has used **Matlab** in a few courses, Professors Chapin and Young have used **Maple** in a few of sections and Ohio University Distinguished Professor S.K. Jain regularly teaches using **MatLab** (he is the author of a Linear Algebra text [4] which uses **Matlab**). Reasons that the software has not been more widely used are complex, but most stem from perceptions on the part of the faculty about how the software could be used. It was a commonly held notion that to use the software one must hold class in a "technology classroom" and completely restructure the class around the technology. Opposition to doing this ranged from practical to philosophical. Ohio University has not been unique with respect to the last few statements. Introduction of Young's methods largely change this problem. In the faculty retreat in October 1999, the mathematics faculty *unanimously* approved the principals of Young's approach.

Aside from the issue of **CS**, many faculty in the Mathematics Department use graphing calculators in lower level courses. While this is good, it is largely independent of the things we are proposing. Graphing calculators are a good classroom teaching tool, but are far less powerful than **MatLab**, or similar packages, in term of the types of calculations they can perform. Also, for

[2]**Maple** is the registered trademark of Waterloo Maple Inc.
[3]**Mathematica** is the registered trademark of Wolfram Research, Inc.

homework, and in the workplace, the visual advantage of a large screen and the speed of a personal computer are preferred over calculators. On the other hand, using **MatLab** directly in the classroom, is awkward and impractical on a large-scale basis. In light of these considerations, use of **MatLab** for *homework* should become standard, while use of graphing calculators in class should still be encouraged.

3 Why training students to use computational software is important

It has long been recognized that the development of technology should effect the curriculum of mathematics [3] [2]. A key point of Young's book is that these effects should not be merely pedagogical, but substantive. The speed of computers has fundamentally changed the kinds of computations that are possible and this changes the kinds of mathematics people do. Since the advent of computers, the ability to do numerical calculations has increased astronomically. In response, fields of numerical mathematics have blossomed in recent years and the use of numerical methods in the sciences and engineering has become ubiquitous.

The invention of symbolic manipulators in the mid 1980's added a whole new dimension to the interaction between mathematics and computers. For example, students can now find the integral of a function without learning lots of "tricks", which were a big part of traditional calculus courses. This example only scratches the surface; both students and researchers can now do many abstract operations on the computer, something that a generation ago was only a dream of a handful of people.

Precisely because the new **CS** is so remarkable, to use it properly requires training. Consider the now familiar example of a calculator. Even though a calculator alleviates the need for most people to do long division, a calculator is useless to the person who does not understand when to add and when to divide. Take this example and multiply its complexity be several thousand. **CS** is to a calculator as chess is to tic-tac-toe.[4] The user of **CS** must understand a number of things to take advantage of it. First, since both numerical and symbolic operations are possible, the user must understand the advantages and disadvantages of each. Not only must the user understand what she wants to calculate, but she must also have some understanding of how the computer is going about it's work. Moreover, with the new software it is quite possible to believe that the computer is doing what is desired, but in fact it is doing something completely different. This is because the new software does things for which fail-safe algorithms do not exist. Results of these computations can be misleading or even wrong. Fortunately, most problems of this nature can be avoided if students master a few basic mathematical principals about the software. To not make use of the software as an educational tool is bad enough in itself. However, it would be an even more serious problem if our graduates are ill-prepared to use computational software when they inevitably encounter it in the workplace.

4 Young's book and homework project

Young was an undergraduate at the beginning of the computer age and had many experiences with the experimental introduction of technology in undergraduate education. Later, Young was a graduate TA at the Georgia Institute of Technology at the time when math professors were first experimenting with the use of **Mathematica** for teaching. That experiment was going very badly, so Young developed and implemented his own approach. As a postdoc at Northwestern University, Young took part in the introduction of **Maple** into the calculus sequence. Based on these experiences he is writing a book "Computers in Undergraduate Math – A Simple Approach". In the book, Young promotes the use of **CS** in carefully designed homework assignments. Here are a few of the key points from the book:

[4] Adapted from a quote by Tuvak on Star Trek Voyager.

Widespread availability and widespread use are crucial. The obvious advantage of widespread availability is that students can use the software when and where they choose. This eliminates problems like crowding of specific computer labs when a project is due. Widespread availability also allows the students to make use of the software as a tool not only for math classes, but allows the software to become part of mathematics usage across the campus. The introduction of computers in the dorms is a great step, but, for students to get the full benefit from the computers, they must have access to good software.

Another clear advantage of widespread availability and use is that in subsequent Math classes and in math-using courses, professors are able to assume knowledge of the software. This leads to a type of snowball effect: widespread use encourages more widespread use.

A less obvious benefit of widespread use is that a culture or collective knowledge about the software develops. When this happens, the need for technical advise plummets, because students are able to obtain immediate help from classmates, friends and neighbors.

Others have observed the wisdom of widespread availability and use. According to Krantz [5],

> We have had catastrophic experience, at my own university, trying to introduce **MAPLE** or **MATHEMATICA** or some other software into isolated courses. Students are not stupid. They catch on right away to the fact that this software is specific to the particular course, and as soon as the semester is over they are unlikely to see it again. They resent having to learn a whole new language .... If we are going to introduce serious software into the lower-division curriculum then we should do it globally instead of locally. ... It makes sense to tell students from day one that their entire lower-division mathematics curriculum will depend on **MAPLE** ... and that they will need to master it right away. Having understood this dictum, they will comply ... and the software will become part of their *lingua franca*. They will (we hope) carry it (or the analytical skills attendant to it) on to the rest of their education, and their lives.

Assignments should focus on the principals needed to use the software well. Assignments in Young's approach are primarily aimed at training the students to understand the differences between numerical and symbolic computations and the underlying mathematics needed to use each effectively.

Use of the software can and should be simple for students. An instructor of an introductory math course should be careful to not unduly burden students with technology, as has often happened in experimental introduction of technology. It is an undeniable fact that math courses are difficult in themselves. Further, these courses are major career events for students interested in the Sciences and Engineering. To overburden students in these courses with additional difficulties borders on irresponsibility. Widespread availability and use are two keys to making assignments simple for students. Other steps which are important for this purpose in Young's approach include: 1. Use simple, basic calculations to demonstrate important principles, 2. Give very clear assignments, and, 3. Give very clear technical information.

Using the software can and should be easy for instructors. Many previous attempts (at other Universities) to incorporate software in math classes have required extensive time commitments on the part of instructors. This use of time is not only impractical for professors with other time demands, but it is unnecessary. Widespread availability and use of the software, and a clear, simple approach to assignments, make demands on professors minimal. Also important is that professors can make good use of the software without direct in-class use. Experience has shown that in-class use of software is not a good model for many professors. The simple homework approach can give students significant experience in using the software without burdening the faculty.

Writing should be incorporated in the assignments. There is a growing body of research which indicates that writing in the context of mathematics helps students to form and solidify abstract mathematical concepts [1] [7]. Writing is also an important skill in and of itself and should be encouraged whenever possible. Each homework assignment contains essay type questions where students are asked to describe and analyze results of the computations. It is emphasized that quality of writing will be an important part of the grade on the assignments.

5 About Matlab

MatLab, which is one of the most popular **CS** packages, was introduced before the invention of symbolic computation, but it now has symbolic capabilities via the **Symbolic toolbox**. **MatLab** is used heavily in Engineering schools around the country, including our own. It has excellent graphics which is important for pedagogical purposes. **Matlab** can also be used as a programming language and is the primary programming language used by the Department of Chemical Engineering. Another nice feature of **Matlab** is that it can interface with the program **Maple**, which has more advanced symbolic capabilities, and is also being used in the University. In fact, **MatLab** contains a scaled back form of **Maple** to do its symbolic computations. In comparison with the costs of hardware and other computer related expenses, the cost of a University site license for **MatLab** is absolutely trivial.

We are promoting **MatLab** over other software packages for a number of reasons. In addition to the feature mentioned above, **MatLab** is currently the most widely used package at Ohio University. It is very flexible and completely appropriate for use in the undergraduate mathematics courses. There exist many textbooks and supplements which use **MatLab**, including the one by Distinguished Professor Jain [4]. However, since the choice of software is one that influences education in so many departments of the University we propose that a University-wide committee be formed to make the final choice of software and this choice should be subject to approval by the Provost. Representatives from all affected entities in the University should participate. The committee should also be responsible for decisions about specific terms of the the licenses.

6 Implementation

First the University-wide committee should be formed to discuss and decide on the choice of software and specific details of the license. For instance, with **MatLab** one may also purchase certain "toolboxes" which perform more advanced tasks and the committee should consider which toolboxes should be included in the campus-wide distribution. Perhaps some departments will wish to continue to maintain small individual licenses because of specialized needs. Also, both annual and perpetual licenses are available and the committee should also consider which pricing option suits the overall needs of the University.

Once the decisions are made by the committee, 1804 Fund money will be used to purchase, or supplement the purchase of a University-wide site license for the chosen software. Once the licenses are obtained the software can be installed on any campus computer used for course related work, including dorm computers. The Office of Information Technology and Computer Services, with assistance from the authors, will oversee the distribution and installation of the software. The Math Department and Information Technology will be jointly responsible for advertisement of the software. Public demonstrations on accessing and using the software will be held around the University. Included in the license agreement for **MatLab** is that Ohio students can purchase the program, at a reduced price, for use on their home computers. The student release includes the **Symbolic** toolbox. With a site license, interested parties from all departments can begin to use the software immediately. A site license will also allow students immediate access to this powerful tool. Probably many of them will begin using it for classwork without any urging from the faculty.

As part of his book, Young has already prepared homework assignments for various levels of calculus, differential equations and linear algebra using **Maple**. (These are available on the Internet at `www.math.ohiou.edu/~simple`). Snyder and Young are currently adapting these assignments to **MatLab** and a sample homework assignment is attached. During the 2000-2001 school year, Chapin, Young and others will begin using the **MatLab** assignments and the assignments will be edited and improved. A PACE position has already been approved for the purpose of assisting in the development and posting of these assignments. Also during this year, the Mathematics department will evaluate the assignments. During this time the investigators will organize introductory demonstrations on accessing and using **MatLab** at various locations on campus.

Beginning with the Fall 2001 quarter, the homework assignments will be adopted by most of the calculus and differential equations sections (MATH 263, 340). Professors will have options of assigning the homework via web page or as handouts. We will make it possible for individual professors to easily modify the assignments to met the needs of their own classes. Demonstrations on accessing and using **MatLab** will be held at the beginning of each quarter.

Once the software is being widely used in the mathematics department, professors in other Math-using disciplines can begin to freely assume that incoming students can use the software.

7 Summary

The University has an opportunity to give students training in an important technology. This can be accomplished at minimal expense and with relative ease. Not only will introduction of **CS** effect undergraduate mathematics education, but will improve the use of mathematics across the University.

References

[1] E.S. Bell and R.N. Bell, Writing and Mathematical problem solving: arguments in favor of synthesis. *School science and mathematics*, March 1985, 210-221.

[2] T. Berger and H. Keynes. Everybody counts / everybody else. In N. Fisher, H. Keynes, and P. Wagreich, editors, *Changing the culture: Mathematics education in the research community*, volume 5 of *Issues in Mathematics Education*, pages 89–110, American Mathematical Society, Providence, RI, 1995.

[3] M. Heid. Resequenceing skills and concepts in applied calculus using the computer as a tool. *J. Res. Math. Ed.*, **19** (1988), 3–25.

[4] S.K. Jain. and A. Gunawardena *Linear Algebra with Matlab Drills*, with complete electronic text on CD. Springer-Verlag, New York, to appear Nov., 2000.

[5] S. Krantz. *How to Teach Mathematics – a Personal Perspective.* American Mathematical Society, Providence, RI, second edition, 1999.

[6] J. Kaput and P. Thompson. Technology in mathematics education research: The first 25 years in the JRME. *J. Res. Math. Ed.*, **25** (1994), 676–684.

[7] M. Williamson and D. McAndrew Writing in college developmental mathematics. *Research and Teaching in Developmental Education* **3** (1987), 14-21.

[8] T. Young. *Technology in College Mathematics – A Simple Approach.* In preparation. Draft available at: `www.math.ohiou.edu/~young`.

A Budget

Summer salary & benefits	$7,500
Hardware	$2,500
Software	$10,500
Total	**$20,500**

B Budget Justification

Summer salalies will be for the three investigators in the summer of 2001. The investigators will: develop assignments, coordinate use of the assignments within the Math Department, hold introductory demonstration sessions about the software around campus, and work with Computer Services on license and installation of the software. A PACE position has already been approved for the coming year for the purpose of assistance in the development and posting of assignments.

License arrangements are on a 'concurrent user' basis, that is, there is a limit to the number of computers using the software at any given moment. The funds for hardware would be used for a small PC and associated software to keep track of concurrent users of **MatLab**. Computer Services will maintain this computer.

Funds for software will be spent on a campus-wide license for **MatLab**. Although the exact details of the license will be worked out by a university committee, the requested amount is intended to cover a perpetual license for 500 concurrent users at a cost of $21 per user (license for **MatLab** and **Symbolic Math** toolbox). The Department of Mathematics will contribute up to $600 per year toward the estimated cost of $2,000 per year (if a perpetual license is chosen) for maintanance of the license.

C Attachments

- Price List for **MatLab**
- Sample homework assignment using **MatLab**