DOCUMENT RESUME

ED 364 431 SE 053 963

AUTHOR	Senechal, Lester
TITLE	Perspectives for Computing in Mathematics. Technology in Higher Education: Current Reflections. Fifth in a Series.
INSTITUTION	North Carolina Univ., Chapel Hill. Inst. for Academic Technology.
REPORT NO	ISBN-1-878643-06-1; ISSN-1046-6274
PUB DATE	91
NOTE	22p.
AVAILABLE FROM	Institute for Academic Technology, The University of North Carolina at Chapel Hill, P.O. Box 12017, Research Triangle Park, NC 27709.
PUB TYPE	Books (010) -- Viewpoints (Opinion/Position Papers, Essays, etc.) (120)
EDRS PRICE	MF01/PC01 Plus Postage.
DESCRIPTORS	Calculators; *Calculus; *College Mathematics; *Computer Assisted Instruction; *Computer Uses in Education; Curriculum Development; *Educational Change; Higher Education; *Instructional Innovation; Mathematics Education; Mathematics Instruction; Teaching Methods
IDENTIFIERS	Computer Integrated Instruction

ABSTRACT

This document promotes a better awareness of available technological tools, their implementation, and their advantages and applications in teaching and learning mathematics. Principles for successful implementation are: (1) computation should be an integral part of instruction; (2) software and curriculum development should go hand-in-hand; (3) implementing computation without radically altering curriculum is possible (but not optimal) as long as principle number 1 is adhered to. Computers should be distributed across classrooms, and physical design of facilities is important. Calculus reform projects are cited as examples of how curricular changes can be brought about. The major outcome of a workshop entitled "Computers in the Mathematics Classroom" was that faculty are in great need of models for curriculum change and the improvement of teaching through the use of technology. Several mathematics software packages have "scripting" or "notebook" features which allow the teacher to alter the scripting to suit individual tastes and needs, allow students to work individually or in groups, and reduce the need for lecturing. The use of projection technology integrated with computer applications for use in large classrooms is discussed. The technologies of teaching and learning in mathematics and science and in the humanities are drawing closer together with developments in hypertext and modular toolkits. (MAZ)

iat

Institute for Academic Technology

Perspectives for Computing in Mathematics

Lester Senechal

Fifth in a Series
Technology in Higher Education: Current Reflections

Perspectives for Computing in Mathematics

Technology in Higher Education: Current Reflections

Perspectives for Computing in Mathematics

By Lester Senechal
Mount Holyoke College

Support and funding for
the Institute for Academic Technology is provided by
International Business Machines Corporation.

Some of the materials appearing in this publication are from the November/December 1990 issue of *Academic Computing* and are reprinted with permission.

ISSN 1046-6274

ISBN 1-878643-06-1

Contents

The Software is the Solution!

In fact, it doesn't seem to be; not for most of us; not by itself at least. Relevant here is a story about a man who was asked whether he could play the violin. He replied that he didn't know; he had never tried. So it is, too, with computing in mathematics and in other disciplines. Most of us have never tried, but those who have find that the art of using computational instruments well in teaching is not a trivial matter. Training and practice are required; maybe even talent. The successful use of computing in teaching does not, in any case, tolerate gladly a blatantly casual approach.

The effort required to use computers well tends to make faculty overly conservative. In every department there is someone who has "tried," but with discouraging results, reverting then to the old ways. Usually the attempt has taken the form of a computer overlay to an established course, say calculus. Students perceive, most often correctly, that a new burden is being superimposed on already abundant burdens and rebel to whatever extent they can. Those among them who are predisposed to be afraid of computers will have their fears confirmed. Others, more neutral initially, will be inspired to avoid computers in the future.

It is a truism that the computer is changing mathematics, but it is easy to underestimate the magnitude and extent of this change. One profound difference already perceived is that significantly more people are becoming users of advanced mathematics. This is especially the case in engineering where many practicing engineers routinely apply principles of advanced mathematics, relying on computers to expedite calculations quickly and accurately. It is of paramount importance, however, that the underlying mathematical principles be clearly understood by the practitioner, in order that the results of the calculations be meaningful and trustworthy.

Herein lies a major shift which seems to elude some academic mathematicians: the computer allows the stripping away of trivial detail (what the Greeks disdainfully referred to as logistics) and permits us to concentrate on important theoretical principles which inform the calculations. The curriculum, however, is slow to respond to this profoundly different circumstance and, in typical introductory courses in calulus, differential equations, and linear algebra, the paucity of organizational principles gets submerged in a sea of poorly organized calculations. Such courses will become much more attractive to all users of mathematics once powerful computational aids get properly integrated. When this happens, we will see the beautiful and powerful theoretical side of mathematics playing a larger role.

A mathematics curriculum is sick if its sophomore-level courses are only accessible or of interest to mathematics specialists. Mathematics (like history and literature and music) is for everyone! It is an important part of our culture. The discipline has allowed itself to become introspective and exclusive, but truly it could be accessible to a large proportion of humanity. Everyone (and this was one of the very important and correct messages of the "new math" of the 1960s) should understand not only concepts of number, but notions of set and mapping as well. The trick is to make it possible for the learner to construct a fabric of meaningful reality around these basic concepts, a web of metaphor. Just as the ruler provides a metaphor for number, so there is also a variety of visual and emotional metaphors for mappings. The computer provides us with rich means for displaying the metaphorical possibilities.

The computer also gives new power and meaning to a major in mathematics. Whereas, previously, this major was mainly preparation for graduate study or a teaching career or an option for general education, it now can take the shape of a professional degree. A computer-literate mathematics graduate can be assured of well-compensated employment in a scientific, business, or engineering field. The utility of computers survives graduation day, unlike the utility of many mathematics textbooks, which rarely survives the semester. Just as was alluded to above in relation to engineering, proper mathematical education provides general philosophical orientation, discrimination, and taste.

The role of the computer is to free us from the banal, the repetitive, the mechanical. It is not, as some suppose, to enslave us, but to free us to follow pleasant and profitable pursuits. When we look at our academic colleagues, we see that the ones who make use of computing are precisely the more philosophically adventurous among us.

Most of us are, however, faced with the dilemma of extreme conservatism on the part of our academic communities as regards instructional methods, just when the potential for change is the greatest. There is nothing paradoxical here: it is just that there is an inevitable time lag, reflecting a physical principle that the motion is stationary when the potential is at a maximum. Some of us, however, want to be assured that this lag will persist for just a few years, not for decades. It is high time to push for change, to intervene. Herein lies a role, at least in part, for federal and private funding agencies and for new constructs, such as the Institute for Academic Technology (IAT), to bring to academics a better awareness of the available technological tools, their implementation, and their advantages and applications in teaching and learning.

The Software is the Syllabus!

Hardly! In fact, on the contrary, software (the name itself is repugnant — the French *logiciel* sounds nicer to the English ear) seems like something terribly commercial, like an advertisement in an airline or computer magazine. Most software, we know too well, just sits passively on a shelf in a department closet or in a repository in an academic computing center, or in a drawer, or on a computer's hard disk in a faculty office, its power going unrealized.

Ideally, software and curriculum should evolve together — the curriculum taking the shape which optimizes the use of computation, the software accommodating itself to this form. At the moment, however, software development is in the lead, and the curriculum is barely moving. The pressure point is calculus, through the National Science Foundation's sponsorship of a reform movement. But the resistance to change is great since calculus instruction serves diverse purposes and demanding clients, whose expectations often diverge and sometimes clash.

It is easier to implement computation when the situation is simpler, for example, in an introductory course in number theory or in abstract algebra. Although these courses primarily serve mathematics majors at present, it would be possible to make them sufficiently interesting and useful for science students that they will begin to draw new clients. Another example is algebraic geometry, where computer algebra systems are being used to good advantage, with the effect that material previously taught only at the graduate level is becoming accessible to undergraduates.

Number theory deserves special mention because it goes deeply into our cultural history and is potentially a rich source of experience for the general student. At Mount Holyoke College, a module in number theory was developed for the introductory cultural history course "Pasts and Presences" by my number-theorist colleague, Giuliana Davidoff. Excellent coordinated software (really a laboratory for number theory for students of all stages of development, from elementary school to graduate school) was written by Mark Peterson, an eclectic mathematician/physicist/computer scientist, whom an enlightened dean had the good sense to support, by allowing Mark to devote a fourth of his contractual time to academic software development — an arrangement which must be extremely rare in the academic world.

In mathematics, we currently have the benefit of enough experience to make it possible to recognize some guiding principles:

1) in order to be successful, computation should be an integral part of instruction;

2) software development and curriculum development should ideally go hand-in-hand, although

3) it is possible (but not optimal) to implement computation without radically altering the curriculum, as long as principle one is adhered to.

There are also principles for physical facilities:

1) computing facilities should ideally be distributed, i.e., the computing takes place, optimally, in the normal teaching environment of the department;

2) physical design is important: there is something a bit

dehumanizing and off-putting about rows and rows of computer benches.

George Francis, a mathematician at the University of Illinois, suggests the placement of computers around the perimeter of the room, so that students face a wall when using the machines. In this way, it is possible to accommodate ten machines in a small classroom, leaving 80 percent of the room free for noncomputational purposes, e.g., discussion sessions. Computers do not need to dominate the physical environment in order to be used effectively; they can be quiet and unobtrusive.

An extremely important psychological aspect of distributed facilities is that, when students spend their time in a department's own facilities, they become invested in the department and bonded to it. N. ·ural science departments have known this for a century, and the purpose of science laboratories transcends scientific method. (It cuts both ways, however, and many students are repulsed from science by the boredom and fatigue experienced in laboratory courses).

The Grand Calculus Reform Projects

These are very well known to mathematicians, so well in fact that mathematicians are becoming tired of hearing about them. They are, however, prototypical examples of how curricular changes can be brought about when there are adequate priorities set, adequate collaborations realized, and adequate funding support bestowed. They are, historically, the greatest curriculum reform efforts ever undertaken in mathematics.

For our attention here, it is most interesting to note that every one of the large projects incorporates computation as a fundamental pedagogical method. The implementations vary: some use established software such as *MathCad* or *Mathematica*; some projects have produced customized software. All, however, have woven the thread of computation into the intellectual fabric of mathematical analysis. This is a fascinating outcome, since the groups of mathematicians who pursue these developments represent divergent tastes and sensibilities. It provides strong evidence for the conjecture that other parts of the curriculum would become computationally intensive, too, if priorities were so directed and if sufficient resources were available to bring about change.

A Model Workshop

In July of 1990, a workshop entitled "Computers in the Mathematics Classroom" was held at the IAT and supported by funding from the Alfred P. Sloan Foundation. It was intended as a model workshop in the sense that it sought to demonstrate the current range of possibilities in the use of computers in teaching the mathematical sciences. It focused both on technology and on curricular issues. It was interactive in that feedback was continually solicited from the participants, and there was much informal discussion.

Twenty-six participants were selected from an applicant pool of nearly 300. The selection was based partly on considerations of diversity, especially with regard to type of institution: liberal arts college, small or large university, community college, historically black, inner city. Its purpose was to demonstrate the possibilities which computer technology has in the teaching and learning of mathematics. One of the features in the design was that each participant would receive a copy of each piece of software used in the workshop. For this purpose we had budgeted $250 per participant and, even though vendors were willing to supply software at dealer prices, keen decisions had to be made on what to include or exclude. There was some freeware, and there were some low-cost noncommercial programs. Within this restricted budget, we were able to assemble a collection of approximately twenty-five excellent programs — each a marvel in itself when properly used. Their range of applicability covers virtually the entire undergraduate curriculum from precalculus to group theory and complex variables.

Thus, without having intended to do so in advance, we arrived at a selection of an affordable collection of high-quality software that is capable of serving many of the present needs of teachers of college mathematics. Our greatest regret is that the user interfaces are of varying degrees of elegance and convenience. All would benefit from being put into Windows 3.0, something we hope to be able to expedite, or at least encourage, over the course of the next year. We intend, furthermore, to periodically update and improve this "core collection," always willing to replace good with better so long as the total cost remains roughly the same. At list prices, this total cost would be about $500; still, a bargain for such powerful and wide-ranging tools.

The major outcome of this workshop experience for the organizers is the acquisition of strong evidence that faculty are in great need of models for curriculum change and the improvement of teaching through the use of technology. Mere possession of hardware and software will not turn the trick.

We are extremely grateful to the Sloan Foundation for its support; yet, we are at the same time cognizant that the need which was satisfied in this one workshop (and a complementary one which took place at Dartmouth) is only a small portion of the need which exists. Therefore, we are proposing a system of "exemplary sites" for consideration by the mathematics community. These are to be distributed geographically and by type of institution. At these sites, computer classrooms would be set up in mathematics departments for the purpose of allowing faculty, who would be beneficiaries of released-time arrangements, to experiment with the incorporation of computing into the teaching process. A group of faculty mentors with extensive experience, selected nationally, would provide advice and support. Furthermore, the sites would serve as regional locations for workshops and briefings when courses are not in session.

Interactive Texts for Mathematics

Several of the larger mathematics software packages have "scripting" or "notebooks" features which allow for the production of a new type of teaching/learning package, vastly superior and richer than a traditional textbook. That is, they allow for the writing and reading of text within the computational environment so that scripts containing information and instructions can be given, problems solved, and experiments performed without interruption and without the necessity of reference to other sources.

The efficiency is striking, not just in principal, but in actual practice. What is more, the materials prepared in this way have a hypertext aspect so that many branchings are possible which pedagogically is a great contrast to the traditional book. The result is a new paradigm for teaching and learning (and even for research) which parallels nicely a similar paradigm for the use of hypertext in the humanities.

Two prototypical examples of such interactive courseware have come to our attention, both magnificent models for showing the power of this new medium. These are the "calculus notebooks" by Horacio Porta and Jerry Uhl of the University of Illinois (Brown, Porta, and Uhl 1990) and the "projective geometry notebooks" under development by Dana Scott of Carnegie Mellon University.

An additional advantage of such courseware is its ability to allow for customization. The teacher will have it within her power to alter the scripting to suit her particular tastes and needs (and the legal right to do so within the classroom situation). She might, for instance, want to trim down the size of a standard product in order to restrict the attention of her students to the points of interest and thereby increase efficiency, or she might want to customize the exercises, or bring to the textual parts her own insights and her own tastes. Thus, the medium is empowering to the faculty member and enhances individual inclinations and perceptions in ways that a traditional textbook has no chance of accommodating.

The "notebooks" software obviates to a great extent the need for lecturing: students can work individually or in groups while the instructor plays the role of consultant. In fact, a new concept of "teaching to the backs" of students has arisen in connection with this and other interactive software. The importance of lecturing is decreased, if not eliminated entirely. The role of instructor becomes that of coach, consultant, and advisor. Authority lines are eliminated where this is desirable: students can become friends and co-investigators with their teachers.

There is a pressing need for "notebook" development in many areas of the curriculum in order to produce self-contained learning tools which can be readily adopted and adapted by faculty and used effectively by students. There is enormous potential for the painless teaching (and effective learning) of material which is preparatory to calculus. There is potential, too, for courses such as complex variables, differential equations, and numerical analysis. There is also much potential for building on Dana Scott's pioneering work in notebooks for geometry in a program for revitalizing the role of geometry in the introductory curriculum. The mathematical community is well-advised to pursue these possibilities with vigor; the funding agencies and publishers are urged to aid in underwriting this important curricular change which could realistically be fully implemented within the next decade.

A Toolkit for Constructing Mathematical Software

In spite of the attractiveness and promise of interactive texts, other types of software will undoubtedly retain their importance. This is especially true of smaller-scale programs written to standardizing graphical interfaces, like those provided by the Macintosh and by Microsoft Windows 3.0. The dilemma here is that programming in graphical environments, at least under present circumstances, goes beyond the capabilities of most amateurs. The key to a viable direction lies in the object-oriented possibilities of assembling programs out of cleanly separated "small" modules capable of communicating with each other, e.g., function interpreters, expressions editors, and graphers capable of communicating through Windows Dynamic Data Exchange.

However, there is nothing which forces this programming style; and the production of such modules, despite their evident advantages, is presently being promoted only by a tiny group of mathematical *avant garde*. The modules would facilitate cooperation among mathematicians who program — they would share modules. The modules would contribute to the uniformity of the interface since the same code would run pieces of many different applications. This kind of organization would also be amenable to a high-level scripting language like Asymetrix's *ToolBook* so that eventually Windows programming within the universe of the existing modules would become easy. The modules would have to be fully encapsulated, incapable of being altered from the outside so that they would perform reliably and consistently.

The main argument for this approach, and it is a telling one, is that it is open-ended and truly limitless in its potential. It can make use of the creative energies of anyone who wishes to participate, and it offers many levels of participation. It will start out at the programming level with demonstration projects exploring, developing, and refining the idea of modularity and reusability. Then it will progress to prototype modules — prototype in the way they interconnect, not necessarily in content — and to scripting capabilities at a higher level, provided perhaps by *ToolBook*.

The Software Toolkit modules will be usable by everyone for building programs. It will be a kind of software equivalent of open computer architecture. The challenge in assuming leadership in such a project, whether at the IAT or elsewhere, will lie in determining good module styles, promoting them, and bringing a variety of prototype modules into existence.

Presentation Aids

Most of us mathematicians are devoted to chalk and slate and abhor the use of any advanced technology in lecturing. Most even dislike using overhead projectors. However, we really need to be aware of the possible advantages technology can provide, especially since some great improvements have been made in recent years, both in projection technology and in software for presentations.

The lecture format is presently much maligned, and perhaps rightly so; but as long as it is a necessary practice in teaching, especially in front of large audiences, we would do well to pay some attention to the advantages which technology offers. Projection systems presently are either inconvenient, unsatisfactory, or expensive, but they are getting better and cheaper. Assuming the availability of a good projector attached to a good computer, we can do some interesting things. In fact, the same type of software which is capable of producing interactive texts is also excellent for presenting mathematics. *ToolBook* is especially well-designed for this purpose and allows for the use of hypermedia, say for historical anecdotes or just to relieve tedium. *Mathematica*, in its notebooks form is also a good lecturing tool. Some non-hypertext programs also make good lecturing aids: the program *GASP*, which is designed for teaching probability, incorporates sound (including songs which relate to chance, e.g., *It's my lucky day!*), cartooning, and a lot of good humor and fun.

Huge Classes

In classical times, classes were small (a small group, ordinarily, judging from Plato). Of course, rich folks were involved, and modern notions of democracy require efficiencies of various kinds, so that it's not entirely uncommon for a hundred or more students to be enrolled in a class. Can technology save this unsatisfactory situation?

I think it can, at least to a certain extent. Suppose, for instance, that a class has 180 students. We can design the course along the following lines: two lectures a week making use of projection technology. Then we can break the group up into sections of thirty for one discussion session and one computer laboratory per week. In the typical large university situation, teaching assistants would be used for some of the discussion and laboratory sessions. It is not an ideal situation, but one in which the technology can be a staunch ally. (One of the Sloan Workshop participants is actually going to follow such a procedure this fall in a differential equations course in which 240 students are enrolled.)

Conclusion

There is no final prophecy for computing in mathematics. Five years ago, what is happening now would have been difficult to predict; there have been many surprises and there will continue to be. There have been some disappointments, too. One general trend is noticeable, however, and can be expected to continue. Five years ago, most mathematical computation was number crunching and the humanities weren't engaged. Now we observe that the technology of teaching and learning in both areas (and in the natural sciences, too) are drawing closer in their strategies.

The humanist's interest in hypertext has infected the mathematician somewhat, and our interest in a modular toolkit may be of interest in the humanities. A fairly safe prediction for the future is that these areas will influence each other even more profoundly and that the Two Cultures of C.P. Snow will be drawn a bit closer together in the process.

Note: I am indebted to my IAT colleagues Bill Graves and Jim Noblitt for their many-faceted influence and for exposure to the humanistic side of technology; to Dana Scott and Jerry Uhl for discussions of the possibilities of interactive coursebooks, such as notebooks in Mathematica; *and to Mark Peterson and Jim White for their penetrating and prophetic views on modular tools for Windows programming.*

About the Author

Lester Senechal is Mathematics Fellow at the Institute for Academic Technology and John Steward Kennedy Professor of Mathematics at Mount Holyoke College where he has taught for twenty years. He has also held positions at the Universities of Arizona and Massachusetts and was Fulbright Lecturer in Brazil. He is interested in all aspects of computation in relation to mathematics, from the elementary to the research level. He has been active in the calculus reform movement and in the promotion of undergraduate research as a primary educational mission of the colleges. He is presently involved with a project, funded in part by the Dana Foundation, for increasing participation in mathematics with special attention to women and minorities. He is especially interested in constructing a new entry level course which will stress the interplay between the discrete and the continuous and in which computation will be totally integrated.

RESOURCES

Brown, D. T., H. Porta, and J. J. Uhl. *Calculus and Mathematica, Part I.* Reading, MA: Addison Wesley (1990).

Maeder, R. *Programming in Mathematica.* Reading MA: Addison Wesley (1989).

iat

Institute for Academic Technology

Institute for Academic Technology
University of North Carolina at Chapel Hill

Telephone	919 560-5031
BITNET	IAT@UNC
Facsimile	919 560-5047

U.S. Mail
P.O. Box 12017
Research Triangle Park, NC 27709

Deliveries
2525 Meridian Parkway, Suite 50
Durham, NC 27713