A computer-based approach to teaching applied math using linkway

R. KOK

Agricultural Engineering Department, Macdonald Campus of McGill, 21,111 Lakeshore Blvd, Ste. Anne de Bellevue, QC, Canada H9X 3V9. *Received 4 August 1992; accepted 8 February 1994. CSAE Paper No. 92-502.*

Kok, R. 1994. **A computer-based approach to teaching applied math using linkway.** Can. Agric. Eng. 36:001-007. A second-year engineering course on applied mathematics is being taught using a computer-based approach. The course objectives are to familiarize the students with a number of software tools and to teach them to do mathematics with these. The course material is delivered via a Novell network on IBM Model 50Z machines with VGA screens. The interface software is *LinkWay*, an object-oriented authoring package marketed by IBM. *LinkWay* is used to navigate through course and lecture menus and to execute scripts. The system does not require extensive prior computing knowledge; the students access course material such as text files and graphics by clicking with a mouse on buttons. Software tools accessed are *Handidat, 123, CurveFit, Derive*, and *Matlab*. With *LinkWay*, the course author controls software access (eg., environment variables, paths, defaults, etc.) and the students can use software without needing excessive computer knowledge. During execution of the courseware and the software tools, *Sidekick* is used in the background to deliver additional material and to transfer information from the screen into the students' answer and report files.

Une approche basée sur l'informatique est utilisée pour l'enseignement d'un cours de mathématique appliquée pour les étudiants de deuxième année en ingénierie. Les objectifs du cours sont de familiariser les étudiants avec un certain nombre de logiciels et de leur apprendre à les employer pour les mathématiques. Le matériel du cours est livré via un réseau Novell sur des machines de modèle IBM Model 50Z équipées d'écrans VGA. L'interface logicielle est *LinkWay*, qui consiste en une plate-forme orientée-objet conçue par IBM. *LinkWay* sert à se promener dans les menus des cours et pour exécuter des scénarios. Le système ne nécessite pas de connaissance préalable en informatique; les étudiants accèdent simplement au matériel des cours à l'aide d'une souris. Les logiciels accédés sont *Handidat, 123, CurveFit, Derive* et *Matlab*. L'auteur du cours contrôle l'accès aux logiciels à l'aide de *LinkWay*, en ajustant, e.g., des variables d'environnement et des valeurs par défaut. Durant le déroulement d'une session de travail, *Sidekick* est utilisé en arrière-plan pour délivrer du matériel additionnel et pour transférer l'information de l'écran à des fichiers contenant les réponses de l'étudiant et des rapports.

INTRODUCTION

In the Province of Quebec, children normally attend primary school for six years and secondary school for five years. Then, before going to University, they attend junior college, CEGEP, for another two years. Most university undergraduate programs are credit-based but are structured so that they can be completed in about three years. In our case (McGill University), the agricultural engineering program has a requirement of 106 credits and most students take seven terms to complete it. The course described in this paper is worth three credits and is usually taken by students in their 3rd term; ie., it is a 2nd year course in our system and would correspond to a 3rd year course in provinces with a 4-year university program. The course number and title are 336-319A *Applied Mathematics*. For two years the course has been taught by electronic means, via the computer network on the Macdonald Campus of McGill University.

After a brief literature review, the course, the methods used, and some overall reflections on this approach are presented below.

LITERATURE REVIEW

The use of computers in education is growing rapidly; it is by no means confined to the official institutions. Within this field there are two major, overlapping areas of application: training and scholarship. In the former, the emphasis is upon the transfer of specific skills and factual knowledge, whereas in the latter the main objective is the development of personal abilities within a more general knowledge framework. Especially in the military and industrial spheres, but also in vocational instruction, a large number of training applications has been developed based on computer technology and these have generally been quite cost effective. In scholarship, however, the computer-based approach has not yet achieved as deep a penetration as many of its early proponents had predicted, except for some limited tasks such as information retrieval. This has been due to a number of factors, including reluctance on the part of teaching personnel, as well as some severe shortcomings in both the hardware and software aspects of computer technology. During the last ten years, however, many of the technological limitations have been eroded. The convenience and practicality of computer-aided scholarship have therefore continued to improve as computers and telecommunications have become cheaper, more powerful, and almost ubiquitous. As a result, it is now possible in some subjects to provide a better educational experience using a computer-based approach than by traditional means. Because its study requires a mixture of training and scholarship, engineering mathematics is particularly suited to this method of teaching. The same applies to many other engineering subjects.

As an area of study, the use of computers in teaching and learning has been the domain of workers in education and educational psychology. Although dynamic and growing, it is not a new field; the potential of computer technology was recognized very early on. Consequently, there is a vast literature which treats of the role and potential impact of computers in education, as well as their performance. A detailed review of these areas is not in order in this paper; the intent here is to merely provide the reader with some general reference material for further reading. In the Library of Congress system (used in most university libraries and for the indexing of their on-line catalogues) the subject areas pertaining to this field are: "computer-aided learning"; "computer-assisted instruction"; "computer-managed learning"; "computer networks"; "education -- data processing"; "educational technology"; "mathematics -- computer-assisted instruction". A number of relevant works are reviewed below. These have been helpful to the author in developing the course material and are deemed to be representative of the literature at large. Together, they are a good source of information on practically all aspects of computer-aided learning. As well, many of them contain lists of references to aid those interested in yet further study.

In a much-cited work, Papert (1980) discussed how computers affect thinking and learning. He suggested that computers have a conceptual impact, rather than being mere calculating tools, and that they make the assimilation of knowledge easier through their capacity to use flexible models and simulations. He also expressed his feeling that students develop through programming and that it gives them mastery over technology and ideas. Taylor (1980) reviewed the philosophies and accomplishments of some of the most influential thinkers in the area of computers in education. Three aspects are discussed: the computer as a flexible interactive tutor; the computer as an intellectual tool which frees the student from tedious mental tasks; the computer as a "tutee", whereby students program or instruct the computer to perform the first two tasks and thus are encouraged to manipulate and understand facts instead of merely acquiring them.

Godfrey and Sterling (1982) wrote a practical guide to the creation of educational software dealing with all aspects of the process, from conception through to testing and maintenance. In 1984, the yearbook of the National Council of Teachers of Mathematics (Hansen and Zweng 1984) was devoted entirely to the topic of computers in mathematics education. In this collection of papers, the learning of mathematics through programming is discussed, as well as many surrounding issues; the computer is dealt with as a teaching aid as well as a diagnostic tool for educators. Solomon (1986) reviewed both the theoretical and practical foundations of two major schools of thought regarding computers in education. The first advocates the use of the computer as an interactive textbook; rote learning or Socratic-style question-and-answer tutorials are emphasized. The second school supports the development of a more free-form computer environment in which the students are encouraged to discover and explore mathematical ideas for themselves.

Doll (1987) discussed how to evaluate educational software. He considered software in the categories of drill and practice programs, tutorials, games, and simulations. He outlined a strategy for the grading of products, basing this on criteria such as presentation, educational quality, documentation and support materials, user interaction, and instructional management capabilities. Rushby (1987) edited a collection of papers on a range of topics associated with computers in education. In this, the incorporation of computers into school curricula is a major topic and there are several discussions regarding the use of developing technologies like teleconferencing, expert systems, and artificial intelligence. Also, some case studies are included on the use of computer education systems in industry, schools, and the military.

Hannafin and Peck (1988) have written a general text on the design and development of educational software, as well as the evaluation thereof. This book is very useful for those interested in producing their own software. Mason and Kaye (1989) have edited a book in which various experts discuss computer-mediated communication and the issues surrounding the use of "virtual classrooms". Harasim (1990) has edited a compilation of works in which computer-based education methods are examined from several theoretical, design, and methodological perspectives. There is some discussion regarding the distance education system used by the British Open University, which makes use of computer networks and telecommunications facilities to create a learning environment in which students and instructors can interact, despite being separated by large distances. Larkin and Chabay (1992) have also edited a collection of contributions. In these, the use of computers in education is discussed from the perspective of two groups: teachers and educational researchers, whose approach is termed "computer-assisted instruction", and researchers in psychology and computer science who are involved in developing "intelligent tutoring systems". The two approaches are compared and their shared goals and issues are discussed. Several case studies are also included.

COURSE DESCRIPTION

Subject area of the course

Most of the subjects dealt with in the course fall within the area of linear algebra and its associated fields. Students taking the course have learned about vectors and matrices and their algebras before, during their CEGEP education, but are not aware of the uses to which such mathematics can be put, nor do they know how to do real calculations with them. The subjects treated are: data conversion, data modeling and curve fitting, 3D geometry, vector algebra, matrix algebra, filtering, and filter design.

Course objectives

Apart from the obvious objective of teaching the students about the subject area of the course, the aim is to familiarize them with a number of computer-based mathematical engineering tools so that they can effectively do mathematics with these. Students who have completed the course, when confronted with a mathematical task, should automatically consider the option of using computer-based tools to generate a solution and know what type of tool is best suited (eg., symbolic mathware, a simple spreadsheet, a specialized matrix algebra package, etc.). The intent is that the students learn to think in terms of using computer-based tools to do mathematics, whether it be vector calculus, ordinary algebra, differential equations, numerical methods, etc.

Approach to the course

Before the advent of electronic calculators, an engineering student regularly spent considerable effort on arithmetic manipulations using tables of logarithms, slide rules, mechanical calculation devices, and other computational aids. This severely limited the scope of problems that could be solved within the context of a course. Similarly, the difficulties in both symbolic and numerical manipulation are now limiting (eg., looking up integrals in tables, transforming equations, solving sets of simultaneous differential equations, etc.). It is not the students' comprehension that is lacking; it is the mere mechanics of dealing with the situation that holds them back. By providing easy access to the appropriate mathematical engineering tools, the personal computer will become to mathematics what the calculator has become to arithmetic. Thus, students will be freed from the drudgery of manipulation to explore more conceptually.

The approach to the course has been to introduce students to mathematical possibilities without dwelling on details of procedure. Learning the underlying principles of the mathematics occurs by *doing* mathematics rather than in the isolation of theoretical considerations. In most instances, students are presented with an explanation of a principle or method, a data set or equation set, a sample solution and problems together with answers, followed by reference to how the principle or method was applied. Part of the course material is presented graphically and almost all the software tools are able to display results in graphic format. This approach simultaneously stimulates various areas of the students' brains to learn. This is especially effective in areas such as 3D geometry and filtering because the effect of a transformation can be visualized instantly. The students can therefore easily link symbolic manipulation in algebraic terms to its geometric or graphical equivalent. Because it is the software tools that do the manipulating, real data sets can be used for the examples and problems. For example, the students can analyze, transform, and filter 10-year weather data sets. Also, because the system operates on a network to which a weather station is connected, in the near future each student will be able to access on-line data and obtain a unique data set.

Course delivery

The course is delivered using Novell NetWare via the faculty computer network, which uses an IBM PS/2 Model 80 as its main file server. Diskspace occupied by the entire course totals 25 Mbytes: 1 Mb for delivery software (*LinkWay* and *Sidekick*), 4 Mb for the support software, and 20 Mb for courseware. The students receive the course in a lab containing 20 IBM Model 50Z's (80286 CPU's) with VGA screens running the DOS 5 operating system. Also, all courseware files (but not the copyright restricted delivery and support software!) can be copied to the students' personal computers so that they can partly do the course at home. This option is dealt with in the discussion below.

Although it is quite possible for a student to take the course completely independently, a certain amount of guidance is provided during delivery. In the present format, the class meets formally in a lecture theatre for 1 hour/week and the instructor reviews the course material. Although the majority of students thought this was unnecessary, some felt it was essential. Most students also attend 2-hour tutorials which are held twice weekly in the laboratory with the instructor present.

Course presentation

The course was authored using *LinkWay* (IBM Corp., Boca Raton, FL, version 1.01, 1989), an object-oriented authoring package which can be run in several modes. During course delivery the user operates inside a limited version of *LinkWay* (ie., the menu bar is not available and all files are automatically locked). A special run-time version is also available for unlimited distribution to students who want to install the course on their own computers. The author creates folders which are password protected, each lecture consisting of one or more folders which can be linked together. The transfer between folders is transparent to the user. Folders consist of pages which contain objects such as pictures, buttons, or text blocks. Buttons can be used for a variety of functions (eg., page forward, page backward, transfer to another folder, etc). Within *LinkWay* a complete command language is also available and a script can be connected to a button so that clicking on the button causes the script's execution. By clicking on buttons, the user accesses text files, graphic display files, notes, pictures, other software, etc. Also, via script execution, *LinkWay* itself can be unloaded from memory to free up space for other software. Upon termination of that software, the user is automatically brought back into the lecture. Such memory management is entirely transparent to the user. As part of the *LinkWay* system, several editors are supplied for text, graphics, icons, and fonts.

Students access the course from the network menu with a single command: APPLMATH. This executes a batch file which sets a number of DOS environment variables and paths, loads *LinkWay*, and starts the course. The student arrives at the course menu and from there navigates by clicking on buttons with a mouse. From the course menu one can proceed to any of the lectures, tests or exams, or to a print menu. The course is divided into 37 lectures, each containing a lecture menu from which the student can access course resources, other lectures, the course menu, DOS, messages, etc. Transfer to the lectures can be done in Standard or in Fast mode, as selected from the course menu. In Standard mode the student is shown an introductory panel, ancillary information about the course, and then the lecture menu. In Fast mode one is brought directly to the lecture menu. At the beginning of the course, Fast mode is completely disabled and then is gradually enabled as the academic term progresses. At any time during the term, previous lectures can be accessed in either mode and future lectures only in Standard mode. Typical graphics panels used to illustrate the lecture material are shown in Figs. 1 and 2. They are from the section on 3D geometry.

One option available in each lecture is the installation of *Sidekick* (Borland International Inc., Scotts Valley, CA, version 1.5, 1985). This is a memory-resident pop-up utility. It contains a mini word-processor which allows one to look at several text files at a time or to transfer information from the screen to a file. The students can use it to compose their answers or to look at instruction files even after they have

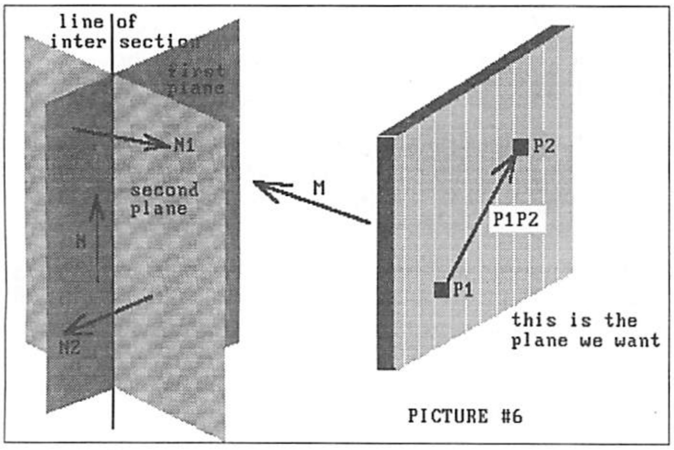

Fig. 1. A graphic panel illustrating a plane parallel to the line of intersection of two other planes.

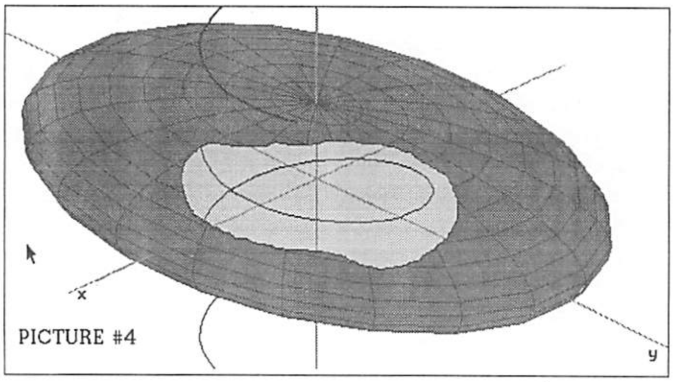

Fig. 2. A graphic panel illustrating an ellipsoid pierced by a spiral.

temporarily unloaded *LinkWay* from memory. For instance, to follow an example instruction sequence in a lecture, students might do the following: a) startup the course, b) select the lecture in question, c) install *Sidekick*, d) point *Sidekick* at the text file containing the instructions, e) activate the application program of interest (*LinkWay* would be unloaded from memory at this point; this is transparent to the user), and f) pop *Sidekick* up and down, alternating between the application program and the instruction file. While doing an assignment, students might also open a second file with *Sidekick* and transfer part of the screen contents, eg., equations or results, to their answer file to hand in via electronic mail. All of this can be achieved with several mouse clicks and by pressing keyboard function keys to select *Sidekick* options.

All course materials can be shown on the computer screen and all text files can be printed out on a dot-matrix or a laser printer via the print menu. As well, most course materials are also available as ASCII and as WordPerfect files so that students can process them further, if they wish. When they start the course, practically all students are already familiar with selecting from menus by clicking a mouse and the startup familiarization period is generally less than 10 minutes. Many students from other departments use the course too, usually without training.

Support software

Apart from *LinkWay* and *Sidekick*, a number of software products are used in delivery of the course: *Handidat*, *123*, *CurveFit*, *Derive*, and *Matlab* with its Signal Processing Toolbox.

Handidat (Logi-Soft, Edmonton, AB, version 2.0, 1986) allows one to convert data and to look up a large number of physical/chemical/mechanical constants in a variety of units (eg., Bohr radius, enthalpy of water and steam, etc.). It is very simple to operate and is used mostly to train the students at the beginning of the course. Similarly, because most students are already familiar with it, *123* (Lotus Development Corp., Cambridge, MA, version 2.01nw, 1986) is used at the very beginning for elementary curve fitting.

CurveFit was written in Basic by Professor A.T. Johnson (University of Maryland, Darlington site, version 3.8 slightly modified by the author) and is intended to be used for multidimensional linear regression. The program operates in six major steps: data entry, model choice, variable transformation and/or permutation, equation solution, display of predicted values, and display of the fitted data. It is used to introduce the students to the concepts of data modeling, least-square regression, and correlation. It allows these to be explored without delving into the details of matrix algebra. The program forces the student to follow a highly structured path and it tightly connects decisions to numerical results and to graphical displays.

Derive (Soft Warehouse Inc., Honolulu, HI, version 2.02, 1990) is categorized as a "mathematical assistant" and can simplify, solve, and plot mathematical expressions of many types. With it, a user can do arithmetic, algebra, and calculus and can work either symbolically or numerically with vectors and matrices, as well as with systems of expressions or equations. The package can plot expressions in a variety of screen formats, either in 2D or 3D. A large number of functions are also supplied in utility files. In the course, *Derive* is used to introduce students to doing symbolic mathematics on a computer and to teach 3D geometry using vectors. The ability of this type of software to display expressions in standard algebraic form on the screen at the same time as their 3D plots is invaluable in teaching this material. As discussed below, the link between symbolic and graphical representation will be further enhanced by adding an animation engine to rotate the plot about any axis.

Matlab (The MathWorks Inc., South Natick, MA, version 3.5g, 1989) is an interactive software package for numeric computation which deals natively with matrices. The instruction syntax is very similar to the way matrix algebra is written; variable names do not need dimensioning and operations such as multiplication, division, exponentiation, etc. are denoted with single operator symbols. It is easy to use and has a good quality graphic interface to display 2D or 3D results. Even on the small computers used to teach the course, *Matlab* is fast enough to allow the use of real-sized data sets in sample problems (eg., hourly temperature data for one year). Because *Matlab* is written for matrix computation it is readily applicable to fields such as numerical analysis, signal processing, and optimization. A number of "toolboxes" are marketed together with the main package (eg., for signal processing, for control system design, and for state-space identification). In the course, the signal processing toolkit is used for filter design and analysis.

Courseware

All course resources reside on the faculty file server in a number of directories and are maintained there by the course author and instructor. The courseware material (as distinct from the delivery and support software) is present in several kinds of files: DOS batch, *LinkWay* folders, ASCII text, bit-mapped graphics, *WordPerfect*, and *Encapsulated PostScript*. In total, the courseware consists of about 1000 files.

In the standard course delivery approach, a 3-credit course is usually assigned three lecture hours per week and, depending on the subject, some laboratory time as well. The academic term lasts 13 weeks so that nominally 39 hours are available for lectures. A 5% time loss is common and one can reasonably expect to give 37 lectures. A similar format is followed for this course: it is divided into 37 lectures. This allows subdivision of the course material into parcels of a convenient size and also helps students pace themselves at a rhythm with which they are already familiar from other courses. A total "captured time" of 3 hours/week per credit is deemed reasonable; this gives the student a 45-hour work week when taking a 15-credit program. In the present delivery format, 5 hours/week are structured, leaving the student about 4 hours/week of unstructured time.

The courseware contains the course menu, the print menu, 37 lectures including 10 assignments, a term test, and a final examination. The term tests and final exams are cumulative, so that students can review the tests and exams of all previous years. In the first year the students wrote the term test as well as the final exam via the network; in the second year they submitted their answers on paper. This is discussed further below.

Table I lists the subjects covered in each group of lectures.

Table I. Lecture groups and subjects

Lecture Group	Subject
1	Course introduction, course topics etc.
2	*Sidekick* and printing
3	*Handidat*
4	*Lotus 123*
5 to 7	Regression and *CurveFit*
8 to 22	*Derive* (includes theory sections of course on 3D geometry, vectors and vector calculus):
	Introduction and demos
	Structure, windows, cursors, precision, notation, expressions, functions
	Algebra in *Derive*
	Derive commands: Declare, Manage, Substitute
	Derive variables: Domains and Values
	Entering relations, solving relations, linear systems of equations
	Calculus, vectors and matrices with *Derive*
	Plotting in 2D and 3D
	3D geometry - equations of common 3D objects
	Vectors - dot and cross products
	Vector calculus
	Vector algebra used in 3D geometry problems
	Problems with lines and planes, curved lines and 3D surfaces
23 to 37	*Matlab* (includes theory sections of course on matrix algebra, filtering & signal processing):
	Review of matrix techniques
	Matlab - demos and help
	Matlab - tutorial
	Matrix operations
	Matrix functions and matrix manipulation
	Least-square analysis and regression curve fitting
	Eigenvalues and eigenvectors
	Graphics and plotting in *Matlab*
	Elementary signal processing
	Filters in use
	Filter performance
	Filter magnitude response and phase
	The frequency domain; the FFT; spectral power density

DISCUSSION

Student considerations

In most undergraduate courses, the instructor is the single most important course resource as well as the delivery vehicle, supported in these two functions by text books, handouts, video, etc. Much of the knowledge transfer occurs verbally in a highly structured situation, ie., at set times in a particular location. Students who need structure and who are good listeners feel comfortable with this traditional format. In the course described, most of the knowledge is delivered via the computer system, supported by the instructor in a tutorial setting. Some advantages, as well as some disadvantages, of this approach have come to light. For learning, students depend heavily on reading and computer skills (including keyboarding) and those students with good aptitude in these areas have done well in the course. Hypothetically, it is possible to do the course work at any time, but the majority of students seems to prefer following a set schedule and the tutorial sessions were judged to be an essential course component by at least 50% of the students. However, the computer-based approach is rather flexible in this regard and a small number of students have done the course almost entirely on their own, hardly ever attending either the lectures or the tutorials. Thus, it gives the students independence and allows those with greater aptitude in this subject area to proceed at a faster than average pace. Also, in this course format, a large amount of ancillary information can be readily made available so that the better students can easily work beyond the course requirements. As well, it provides opportunity for the students with lesser aptitude to proceed more slowly, but still have the chance to ask questions about material that others may already have finished. With this format there is no need to maintain tight synchronization between students and, because very little time is spent on verbal delivery, much more time is available for individual consultation. It is felt that one of the greatest advantages of the present format, ie., computer based with tutorial support, is that it readily accommodates a wide spectrum of students; one can create sufficient challenge for the gifted students and, at the same time, provide support to those with lower aptitude. Presently, students can take home an execute-only version of the delivery software and copies of the courseware, but not of all the support software. Thus, they must do part of the course work in a specific place, the faculty computing laboratory. At times, equipment may be occupied (especially near the end of the term) and access is not always convenient. This is a limitation of the approach which will hopefully be overcome when CD-ROM becomes the standard information distribution medium, all students have their own portable micros, and limited-volume site licenses for software are the norm. The whole course, including support software, easily fits on a 3.5" CD-ROM and this will probably be the publication vehicle for the course in another few years.

This paper is oriented to the presentation of a computer-based course, but there are two interrelated aspects to be considered: a) the transfer of knowledge with the computer-based approach and b) the extensive reliance on support software to do mathematics. The latter readily, although not necessarily, follows in the wake of the former, but it does lead to an issue which has surfaced in the teaching of this course. This is the "loss of skills" on the part of the students, often cited by critics of the computer-based method. The subject is complex and cannot be discussed in detail here, but should be addressed. The criticism has come from both staff and students and usually from individuals who feel uncomfortable with digital technologies. The term refers to skills which would have been imparted in the traditional approach and that are no longer taught as a result of the abilities of the support software. In short, it is the same problem which was experienced when electronic calculators were introduced. Critics claimed that students would no longer be able to multiply and divide or know how to take a square root if allowed to use calculators in school. These dire predictions have turned out to be entirely correct and most students no longer possess the arithmetic skills of which previous generations were proud. Students do not, however, seem to have less of an understanding than before of what a square root is, although they are rather hard pressed to extract one without the requisite tools. Once equipped, they extract such roots about a thousand times faster than was possible even a short thirty years ago. Similarly, computer programs like *Derive* and *Matlab* will probably make many of the still current skills in algebra, calculus, and geometry obsolete within the next ten years. For example, the linear algebra course the author taught fifteen years ago contained extensive material on various methods used to invert matrices, but the applied math course simply relies on the support software to carry out these operations.

Instructor considerations

The time investment in authoring the course has been rather substantial. Part of this has been due to inexperience in CAL (Computer Aided Learning), leading to inefficiency, but a fairly heavy price must initially be paid, regardless of experience level. When the course is in its "final" form, it will have cost about 50 hours per lecture. After that, just like any other course, it will require regular updating at an estimated cost of about 1 to 2 hours per year per lecture. Hopefully, annual improvements will be cumulative and the course quality will increase. One advantage of the approach is that once a mistake is corrected, it remains corrected. Another advantage is that the course in its entirety is available to other members of the academic community, so that questions about the subject matter or the support software can easily be answered by giving a reference to a lecture resident on the faculty network. At the same time, there are some disadvantages. One of these came to light under rather unpleasant circumstances. In his initial enthusiasm, the instructor administered the final examination via the computer network and all students submitted their answers on diskette only. Unfortunately, once the marks had been posted, one student, after requesting a copy of his/her files, made the claim that these files were not the ones submitted. A number of schemes are under investigation to circumvent this problem, but all subsequent term tests and examinations have had to be answered handwritten, in standard booklets.

Software considerations

The computer-based approach readily allows both the delivery of course material and the intensive use of software tools via a single vehicle. In this case, *LinkWay* has been used as the "glue" between text files, graphic illustrations, support software, etc. With this method, the environment that the student experiences can be controlled and one can deal with much more complex problems than is possible with other approaches. For example, because the student is relieved from overhead tasks, real sets of equations can be solved, large data matrices can be processed, realistic simulations can be run, many regression curves can be fitted to full-sized data sets, frequency spectra of on-line weather data can be obtained, etc.

As a software tool, *LinkWay* 1.01 was adequate, although it might be improved in many ways to increase author productivity as well as the quality of the product (although in later versions of *LinkWay* some shortcomings were addressed, the comments below apply to these as well). Fairly major irritants in using *LinkWay* are: a) the graphic, text, and field editors which are part of the *LinkWay* package are rather rudimentary and at times awkward to work with, and b) the font set is very small and does not include a sufficient range of font sizes and styles. Although these problems can be overcome in part by using other software to create graphics and text, it makes authoring more cumbersome. *LinkWay* is flexible enough to accept text and graphic files from other editors and, because the text files are in ASCII, almost any wordprocessor can be used to create them. The graphic files are of the bit-mapped, PCX type and any program able to output according to this standard can be used to create acceptable graphic panels. With *LinkWay* a graphic capture program is also supplied to transfer graphic information into the correct format. These methods have been used extensively in the creation of the course. The native font set can be extended by two means: a) more fonts can be bought from the company which wrote *LinkWay* (Washington Computer Services, Bellingham, WA), and b) the user can create new fonts with the font editor supplied as part of the *LinkWay* package; the author has followed both routes. A most useful addition to *LinkWay* would be a library of standard objects. During several years of authoring with *LinkWay*, the author has developed a large collection of useful buttons, pictures, fields, etc. which, in effect, form such a library. Inclusion of a library in the package might, however, save others considerable work 'reinventing the wheel'. Although it would obviously be of interest, no detailed comparison with other authoring systems has been attempted. To gain sufficient familiarity with a tool to fairly assess its adequacy in a particular context like teaching applied mathematics (as opposed to accepting the assertions in company literature) requires a very substantial time investment and the resources for this have not been available.

It is anticipated that during the third teaching of the course, it will reach its "final" state. It is planned to add a few more support software products, especially to improve graphic representation of 3D geometric figures. Both *AcroSpin* (Acrobits, Salt Lake City, UT) and *GyroGraphics* (Cipher Systems, Stillwater, OK) are under consideration. The former is an animation engine. With it one can translate and rotate a 3D geometric figure (as well as sets of points or lines or an *AutoCad* drawing) in any direction at any speed and zoom in and out, etc. The display of a rotating rather than a static image is useful because it improves the connection between the algebraic symbolic description and visual comprehension. *GyroGraphics* can generate surfaces of revolution which can then be fed to *AcroSpin* for display.

CONCLUSIONS

As with any technology, early adoption brings with it a number of problems and subjects the promoters and adopters to criticism. However, it usually also brings compensating benefits and, in the long run, computer-based teaching will probably prove to be advantageous in a number of situations. Education, and especially re-training, will not necessarily remain time and location-bound. With this approach the delivery and consumption of course material can be de-synchronized and can be separated in space. Historically, this trend started with textbooks and the computer-based approach is an extension of that. It might be particularly effective in less-developed and under-developed regions where qualified teaching staff is simply not available. It may also prove to be a feasible way to increase knowledge levels much more rapidly than via the oral tradition; it takes 20 years to educate a PhD-level engineer but less than a second to print all she or he knows onto a CD-ROM. In particular, the method's effectiveness may be very good because educational backgrounds vary widely in such areas and this approach allows each person to proceed at his or her own speed.

Overall, the cost efficiency (in terms of time) of the course depends on the number of students and the useful life of the course. In this instance, the class size is about 20 and the expected course life is 10 years (while parts of the course are continually replaced via updating). The ratio of delivery time to creation time is therefore 4:1. Of course, this ratio depends directly on class size and once the course is set up it can be distributed to an unlimited number of students with no extra time cost to the author. In terms of cost effectiveness this delivery method is therefore best suited to large classes.

The author looks forward to migrating the course to a true multitasking operating system such as OS/2 2.0, thus eliminating the need for *Sidekick*, and to a hypermedia context in which high-quality animation, video and speech will be available. As well, in the near future it will be possible for students to access the course with their personal computers via a network dial-in facility and thereafter, in the intermediate future, they will probably receive a complete copy of the course on magneto-optical or compact disc. A second course, Advanced Applications of Microcomputers in Agriculture, is currently being taught with the same technology.

All course materials developed by the author are available free of charge and some of the support software, eg., *Handidat*, *CurveFit*, and the limited version of *LinkWay*, can also be distributed. None of the other software mentioned is in the public domain, however, and anyone wishing to install the course would have to acquire their own copies of these products.

REFERENCES

Doll, C. 1987. *Evaluating Educational Software*. Chicago, IL: American Library Association.

Godfrey, D. and S. Sterling. 1982. *The Elements of CAL*. Victoria, BC: Press Porcépic.

Hannafin, M. and K. Peck. 1988. *The Design, Development, and Evaluation of Instructional Software*. New York, NY: Macmillan.

Hansen, V. and M. Zweng (eds.). 1984. *Computers in Mathematics Education*. National Council of Teachers in Mathematics, Restor, VA.

Harasim, L. (ed.). 1990. *Online Education: Perspectives on a New Environment*. New York, NY: Praeger.

Larkin, J. and R. Chabay (eds.). 1992. *Computer-Assisted Instruction and Intelligent Tutoring Systems: Shared Goals and Complementary Approaches*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Mason, R. and A. Kaye (eds.). 1989. *Mindweave: Communication, Computers and Distance Education*. Toronto, ON: Pergamon.

Papert, S. 1980. *Mindstorms: Children, Computers, and Powerful Ideas*. New York, NY: Basic Books.

Rushby, N. (ed.). 1987. *Technology Based Learning*. New York, NY: Nichols.

Solomon, C. 1986. *Computer Environments for Children*. Cambridge, MA: MIT Press.

Taylor, R. (ed.). 1980. *The Computer in the School*. New York, NY: Teachers College Press.