Our experience at the University of Ulster seems to dispel these beliefs except for the case of employment in the finance and commerce sector.

Overall, a majority of the firms who responded to our market research survey placed great value on the sandwich component and were keen to consider students of degrees such as these for placement. But in the finance and commerce sector, (about 32% of the firms contacted and 33% those who responded), the opposite was true and a majority never accepted students for sandwich placement.

At the time of writing of this article our first cohort of 23 students are just completing their sandwich year. All were placed with relative ease and all have reported in glowing terms on the value of the experience obtained. Of these 23 placements: 4 (17%) were in finance and commerce, 8 (35%) involved extensive use of mathematics, 10 (43%) involved extensive use of statistics, and 7 (30%) involved extensive use of operational research. All 23 placements naturally involved some computing, but in only 6 cases (26%) was it the principal element.

4 Conclusion

In his paper O'Reilly [1] poses 22 questions. We hope that our experiences at the University of Ulster, described in this paper, will help him find answers.

References

[1] Maurice O'Reilly, Mathematics at Third Level - Questioning How we Teach, Bulletin Irish Math Soc, No 22, (1989), 50-54.

[2] S. K. Houston, Mathematical Comprehension, in M S Arora, F. Mina and A Rogerson, editors, Mathematical Education: The Present State of the Art, to appear (1990).

[3] J. R. McCartney, A Comprehension Paper in A-level Mathematics, Teaching Mathematics and its Applications, 9, No 1 (1990), 6-14.

Department of Mathematics
University of Ulster
Jordanstown

Use of MACSYMA and MAPLE in Mathematics teaching in UCG

Raymond A. Ryan

Introduction

Several years ago the Honours Mathematics program in UCG began to undergo major reform. At the centre of the changes which our courses are still undergoing was the introduction of modern computing techniques, and in particular the use of symbolic computation software. There were a number of reasons for this reform; two in particular stand out — a desire to reverse a trend of falling numbers of good Mathematics students, and a process of reeducation among members of the Mathematics Department themselves.

The number of students taking honours degrees in Mathematics in UCG had been decreasing through the late '70's and early '80's. Departments like Electronic Engineering and Medicine were full of frustrated mathematicians, but it was not just the distorting effects of the points system that robbed us of students. There was — and still is — an image problem associated with Mathematics. This problem has several facets. On the one hand, most school leavers could picture themselves in the role of an Engineer or Accountant or Solicitor, but few could imagine themselves as a Mathematician. Also, the divorce between Mathematics and Computer Science meant that many Mathematics programs were frozen in outdated modes of content and presentation, lacking the vital interplay with computing which would have ensured growth and change.

The second factor was the changing attitudes of the members of the Mathematics Department. Of course, computers had been used, and taught, for the past 25 years or so, but "Pure" Mathematics had remained largely untouched. Then, over the course of a few years, people became acquainted with Pascal, Lisp, electronic mail, TEX, CAYLEY, REDUCE, MACSYMA, and so on. The

computer had really arrived in Mathematics! And as computing techniques began to find a place in our own research, it became clear that all our courses would eventually have to reflect this new fact of mathematical life. One read constantly of the reform of Calculus in the USA, and if one was still sceptical, there was the spectacle of seeing our own students with calculators which could sketch graphs and compute derivatives and invert matrices. Primitive though these devices were, they were clearly a sign of things to come. Questions began to be asked that wouldn't have made sense five years previously, such as: why should I spend several weeks teaching methods of integration when my students are going to get their integrals from a computer or calculus calculator in "real life"? or: if I use CAYLEY to help me understand the structure of this group, shouldn't my students have the same opportunity?

The response to these stimuli took place at various levels. In this account we look mainly at the introduction of symbolic computation in first year courses.

MACSYMA and MAPLE

MACSYMA was introduced in the first year Analysis course in 1987. While REDUCE was also available, the manuals were so cryptic as to be unusable. MACSYMA was somewhat better in this respect, but it was still necessary for us to write our own introductory manual.

Software such as this can be taught at two levels. The first is to view it as a sort of symbolic supercalculator. The second is as a full programming language, in which the user will write their own procedures, define their own environments etc. At the first year level we concentrated mainly on the first level. Our aim was to make it possible for the student to be able to interact with MACSYMA with sufficient ability to be able to carry out the manipulations that they would be likely to meet in their courses. Initially we dealt with Calculus applications only; in our second year we brought MACSYMA into the Algebra course as well, dealing mainly with matrix-related computations.

The effect on the students was remarkable. Mathematics suddenly became something immediate, rather than a sequence of theorems alternating with daunting examples. The ability to cut quickly through tedious, repetetive calculations meant that more time could be spent exploring the ideas behind the calculations. To give one simple example, consider the partial fraction expansion of a rational function. This can be taught in a formal lecture situation, the various tricky cases outlined and so on, but such knowledge is

best absorbed when acquired actively. It was in precisely such a situation that MACSYMA came to the fore. The student could now compute these expansions at will, and could experiment freely with changes in the structure of the function to see how they would affect the expansion. Similarly, it became possible to deal with more interesting problems in Linear Algebra. No longer were we confined to 3×3 matrices! The courses themselves began to change to reflect the new presence of the computer, and the effect of this has been invigorating for all concerned. Our first year Analysis and Algebra are, in their philosophy and content, and in the way they are taught, quite different from what they were in 1986. There are also implications for the way courses are examined. The traditional three-hour written exam is giving way to a combination of written and computer work.

Unfortunately, it was not all plain sailing, as those who have experience of MACSYMA in a multi-user environment will understand. Each user essentially loads a full copy of the program when MACSYMA is invoked, and this places great strains on the computer. Our experience was that once seven or eight students had started to do computations in MACSYMA, the whole system (a VAX 11/785 in this case) was reduced to a snail's pace, leading to great frustration on the part of our own students and other users. The following year, we tried working with smaller groups of students. This still did not eliminate the problem with speed of response, and the additional supervisory burden created its own problems.

MACSYMA is an excellent program, but is not suitable for simultaneous use by groups of students. Alternatives were sought. MATHEMATICA looks very promising, but the cost, under present circumstances, is prohibitive. MAPLE was acquired instead, and installed on the same computer, alongside MACSYMA (and REDUCE). Its outstanding advantage is its adaptability to a multi-user situation. Each user is given only those parts of the program which are required at the time, other modules being loaded as needed. The capabilities at the level at which we use this software seem to be at least as good as those of MACSYMA, and the level of documentation is also good. So, within the next year or two, we foresee a situation in which all our Honours students will learn to use MAPLE in their first year, and will continue to develop their skills in its use in succeeding years, and will come to take it for granted as a normal mathematical skill which is available when needed.

Conclusions

As a result of these and other changes, enrolment in the honours Mathematics programs began to increase sharply in recent years, and the first year numbers are now three times their previous levels. Although there are other factors at work here, the introduction of symbolic computation in first year has certainly contributed to this development. There is a noticeable improvement in the attitude of the students to Mathematics. The computer is clearly acting as a bridge for them into an area that they otherwise would not feel they could reach.

Finally, the Mathematics courses themselves are changing, and the use of software such as MAPLE is driving this change. New types of problems are now accessible which could not be tackled by hand. Some parts of our courses have become obsolete, and must be ruthlessly pruned. New branches of Mathematics are emerging. The long-term effect of this will be interesting to see. One thing is clear: if we are not preceived by our students as leading in this revolution rather than being dragged along, then Mathematics will, by the turn of the century, be a neglected backwater.

NOTES

A public key cryptosystem as hard as factorisation

M. Christopher W. Jones

1 Introduction

The idea of a public-key cryptosystem was first put forward by Diffie & Hellman in their 1976 paper [7]. Since then various descriptions of it have appeared [3,11,14,24,27] including a recent Bulletin article [10]. The idea behind a public-key cryptosystem is that it allows secret messages to be sent across an open channel without it being necessary for some additional piece of information to be previously exchanged between sender and receiver.

Briefly, the idea is this. If Mr. X wishes to receive secret communications he constructs an *encryption function* E and a *decryption function* D. These should possess the following properties: (i) $D(E(m)) = m$ for all messages m, (ii) both E and D should be easily computable, (iii) it should not be possible to determine D from a knowledge of E alone, (iv) $E(D(m)) = m$ for all messages m. (Actually property (iv) is not absolutely essential, but is useful for purposes of authentication - for more details consult the above references.)

Mr. X then publishes the encryption function E (the *public key*) and keeps the decryption function D to himself (the *secret key*). Anyone wishing to send him a message m then transmits the encrypted message $E(m)$. On receiving this, Mr. X is able to recover the original message using D and property (i). However any eavesdropper who intercepts $E(m)$ is unable, because of property (iii), to discover m, even if he knows the encryption function E.

In order to put the above scheme into practice it is necessary to construct suitable encryption/decryption functions. One way this has been attempted is by the use of a "trapdoor" function f: this is a function for which it is easy to compute $f(x)$ but very difficult to compute $f^{-1}(x)$ without some additional