Taking advantage of the presence of many small computers with large memories, the authors have embarked on a new program of teaching mathematics to engineers. Until complete, the experiment leaves unanswered the question of whether such computer-aided techniques will produce students with fewer math skills.

Teaching Engineering Analysis Using Symbolic Algebra and Calculus

R. H. Lance, R. H. Rand, and F. C. Moon
Cornell University

For the past year and a half limited numbers of sophomore and junior students in the College of Engineering at Cornell have been taught engineering mathematics using new computer software (MACSYMA, muMATH) that performs exact symbolic algebraic and calculus operations. The students use traditional textbooks and the usual syllabi, but their learning of the material is enhanced by computer programs that perform many routine (and some not-so-routine) mathematical operations. Beginning in the fall of 1985 all 550 students in the sophomore engineering mathematics course are using such a program.

We shall describe how we have incorporated a new type of computer-based "mathematics laboratory" into these courses, and how the teaching and learning environment has changed. We shall also comment on some of the immediate effects on the students and speculate on this innovation's long-range educational effects.

Beyond Number-Crunching

The use of symbols and the need to manipulate them in engineering teaching and practice is universally accepted. We all derive and demonstrate fundamental results in symbols and often illustrate applications in such terms. It is not surprising that with the availability of fast, compact and inexpensive computers, programmers have recently paid serious attention to the development of programs that can do more than "crunch" numbers. Such programs as muMATH[1] and REDUCE[2] can perform symbolic as well as numerical manipulations and can be used on computers with smaller memories, such as an IBM PC-XT. SCRATCHPAD[3], requires a larger computer with a CMS operating system, such as an IBM 4341.

At the Massachusetts Institute of Technology's computer science laboratory in 1975, work was supported to develop a large symbolic manipulation program, now known as MACSYMA (project MAC's SYmbolic MAnipulation System).[4] MACSYMA, a large computer programming system written in LISP,

This article is based on a presentation made at the 1985 ASEE Annual Conference in Atlanta and published in the Proceedings.

Some Administrative Details

Although symbolic manipulation programs have received some attention in the literature,[5,6] their use in teaching has not been widely discussed. There is, however, ample documentation on MACSYMA,[4] muMATH[1] and the other programs of the same type, to support the innovative teacher, and Rand[7] has published a book on the use of computer algebra in the context of advanced dynamics and applied mathematics.

Of course the regular use of computers in teaching requires some planning to accommodate all the students doing the assigned mathematical experiments. In the first use of muMATH, in the spring of 1984, we had one lecture section of 65 students. Each student was expected to complete four experiments; each experiment was designed to last no more than three hours. Our math lab had seven IBM PC-XTs and was open three hours a day on a walk-in basis. We had no difficulty meeting the students' computer needs, except during the inevitable last-minute rushes.

The use of MACSYMA in the junior course of approximately fifty students places a small additional load on the DECSYSTEM-2060, which is used extensively by students and faculty in many courses. It has the capacity to carry all but the heaviest demand for computing time, which occurs during the few hours before every due-date. (As an aid to instructors and students, the office of academic computing posts due-dates for large courses.)

Beginning this fall we will have 38 IBM PC-XTs, available in two clusters, open to the sophomores in the math class. Our experience suggests that to conduct four labs a semester, it is necessary to have one PC available eight hours a day, five days a week, for every 40 students. The labs must be attended by trained personnel, usually graduate teaching assistants, for instruction and security.

performs essentially the same functions as muMATH and similar programs but on a much more comprehensive scale. Because of the size of the MACSYMA program, it can be supported at present only by a large-scale computer.

In 1982 MACSYMA was installed on Cornell's DECSYSTEM-2060, the principal instructional mainframe computer on campus. The faculty soon saw the potential for this symbolic manipulator to ease the tedium of lengthy algebraic and other mathematical manipulations in research and, more important, to perform them without error. We in mechanics, who are responsible for a major portion of the teaching of sophomore mathematics to Cornell's engineering students, also recognized the potential value of such programs in enhancing our teaching in these and other courses.

MACSYMA was first introduced to graduate students at Cornell in an applied mathematics course taught by Richard Rand in the fall of 1983. In the spring of 1984 he introduced MACSYMA into his junior course on differential equations (Engineering Analysis I). The following fall, with a segregated set of students in the second semester of the sophomore course, he introduced muMATH, which fits into the memory of a desktop computer. Last spring he continued to develop experiments suitable for sophomores, and we enlarged the MACSYMA component of the junior-level course. This fall, following the installation of a personal computer facility of adequate size, the one-year sophomore engineering mathematics course based on muMATH is being offered to all 550 students enrolled in the course.

All graduate students in mechanics are now encouraged to learn MACSYMA so they can use it in their research and advanced courses in engineering analysis.

Engineering Mathematics for Sophomores

The sophomore math course for engineers at Cornell has a traditional syllabus, which includes partial derivatives, multiple integrals, first- and second-order ordinary differential equations, vector spaces and linear algebra, matrices, eigenvalue problems with applications to systems of linear differential equations, vector calculus, boundary value problems and an introduction to Fourier series. All these topics are associated with applications in the physical sciences and engineering sciences; all provide ample opportunity to assign homework problems that can become so tedious that the point of the problem can be lost in the detail of the solution.

The tactic we have adopted takes advantage of the opportunity to assign "tough" homework problems and to reinforce and clarify the mathematical principles with practice on the computer.

In a typical mathematics experiment students will have had lectures on a topic and will have seen several simple examples worked in class. They will also have worked examples independently for homework. As an example of the muMATH computer algebra system, here is a problem that is easy (in principle) but tedious and thus difficult to do by hand correctly. The problem is to find the indefinite integral of the 10th power of the natural log of x. This computation requires 9 integrations-by-parts, but muMATH computes it in seconds:

```
?  INT( LOG(X)^10, X );
@:  3628800 X + 1814400 X LN X^2 -
604800 X LN X^3  + 151200 X LN X^4 -
30240 X LN X^5 + 5040 X LN X^6
- 720 X LN X^7 + 90 X LN X^8 -
10 X LN X^9 + X LN
X^10 -
3628800 X LN X
```

As a second example, we use muMATH to solve a typical problem in ordinary differential equations: to find the general solution of the equation $x'' + x' + x = 0$. We first declare x to be a function of t:

```
? DEPENDS( X(T) )$

? SOLVE( DIF(X,T,2) + DIF(X,T) +
X == 0, X);

@: {X == ARB (1) SIN (3^(1/2) T/2)/
#E^(T/2) + ARB (2) COS (3^(1/2) T/2)/
#E^(T/2)}
```

Note that the arbitrary constants of integration are represented by *ARB(1)* and *ARB(2)*, and that e is represented by *#E*.

If it is necessary to solve a specific type of problem repetitively, a program can be written to prompt the user to supply essential data. The printout shown at right outlines this approach and shows the response of a student solving a first-order, linear ODE.

Advanced Engineering Analysis for Juniors

The junior course in engineering analysis covers ordinary differential equations in an engineering context. The topics covered include analytical and numerical methods, special functions, initial and boundary value problems and eigenvalue problems in linear partial differential equations. An introduction to nonlinear ordinary differential equations is also given.

This course offers many chances for students to operate with a symbolic manipulator in the interactive (or calculator) mode, i.e., sitting at a terminal and giving the computer instructions to, say, differentiate or integrate a complicated expression that has arisen in their study or homework. There are also general problems of engineering analysis that can be well served by the development of a program, in the above muMATH sense, that can be called upon by the user.

The MACSYMA system is similar to muMATH but much more powerful. As an example of how MACSYMA can be used to solve several coupled nonlinear algebraic equations, take the problem of finding the points of intersection of the following three surfaces:

i) the cone $x^2 + y^2 = z$
ii) the sphere $x^2 + y^2 + z^2 = 1$, and
iii) the plane $y = 2x$.

Here is the record of a MACSYMA run which solves this problem:

```
(C12) SOLVE([X^2+Y^2 =
Z,X^2+Y^2+Z^2 = 1,Y = 2*X], [X,Y,Z]);
(D12)
```

$$[[X = \frac{SQRT(SQRT(5) + 1)\ \%I}{SQRT(10)},$$

$$Y = \frac{2\ SQRT(SQRT(5) + 1)\ \%I}{SQRT(2)\ SQRT(5)},$$

$$Z = - \frac{SQRT(5) + 1}{2}],$$

$$[X = - \frac{SQRT(SQRT(5) + 1)\ \%I}{SQRT(10)},$$

$$Y = - \frac{2\ SQRT(SQRT(5) + 1)\ \%I}{SQRT(2) SQRT(5)},$$

$$Z = - \frac{SQRT(5) + 1}{2}],$$

$$[X = \frac{SQRT(SQRT(5) - 1)}{SQRT(10)},$$

$$Y = \frac{2\ SQRT(SQRT(5) - 1)}{SQRT(2)\ SQRT(5)},$$

$$Z = \frac{SQRT(5) - 1}{2}],$$

$$[X = - \frac{SQRT(SQRT(5) - 1)}{SQRT(10)},$$

$$Y = - \frac{2\ SQRT(SQRT(5) - 1)}{SQRT(2)\ SQRT(5)},$$

$$Z = \frac{SQRT(5) - 1}{2}]]$$

MACSYMA has obtained four solutions, the first two of which are complex. Although the preceding solutions involve exact arithmetic (with no round-off error), in some applications it may be convenient to convert the results to floating point form:

Example: A Program to Execute Repetitive Commands Automatically

Both muMATH and MACSYMA are computer algebra languages that support programming. Programming offers a convenient alternative to the interactive (or calculator) mode: if a series of commands must be given repeatedly, it is best to build them into a program and execute them automatically.

Students will have learned in lecture that to solve the first order linear ODE:

$$y' + P(x)\, y = Q(x)$$

one may use an integrating factor, leading to the formula

$$y = e^{-\int P\,dx}[\int Q\, e^{\int P\,dx} dx + c]$$

They have worked a number of examples by hand for homework and realize that the process, while straightforward, is time-consuming and repetitive. This experiment contrasts the traditional pencil and paper approach with a short (student-written) computer algebra program.

Omitting the program listing, here is a typical run, which begins with the assignment of values to the functions P and Q, and the initial conditions XO and YO:

```
STUDENT : ? P:3/X;
          ? Q:X SIN(X);
          ? XO: #PI;
          ? YO: O;
```

Next the previously-written program, named LINODE, is called:

```
? LINODE( );
```

The program begins by "echoing" the problem back, as a check:

```
COMPUTER: Y' + P Y = Q
          P = 3/X, Q = X SIN X
          INITIAL CONDITION:  Y = 0, X = #PI
          Y = -X COS X + 12 #PI^2/X^3 - #PI^4/X^3
          - 24 COS X/X^3 - 24 SIN X/X^2 + 12 COS X/X
          - 24/X^3 + 4 SIN X
```

That is,

$$y' + \frac{3}{x} y = x \sin x,\ y(\pi) = 0$$

has the solution

$$y = -x \cos x + \frac{12\,\pi^2}{x^3} - \frac{\pi^4}{x^3} + \cdots$$

"It is clear that all engineering curricula will have to provide some training in this new software tool to keep abreast of industrial mathematics needs and practices."

```
(C13) %,NUMBER;

(D13) [[X = 0.56886448 %I,
Y = 1.13772896 %I
Z = -1.61803399],
[X = -0.56886448 %I,
Y = -1.13772896 %I,
Z = -1.61803399], [X = 0.351577584,
Y = 0.70315517, Z = 0.61803399],
[X = -0.351577584, Y = -0.70315517,
Z = 0.61803399]]
```

A further example is to find the solution of an n-th order, linear ordinary differential equation with constant coefficients by the Laplace transform method. As in the sophomore course the student can be prompted to supply the coefficients, the RHS (the forcing function) and the initial conditions. The program—which is really a function in MACSYMA—produces the solution. An example of such a MACSYMA terminal session is given at left.

The Educational Outcome

We are often asked whether computer-aided algebra and calculus will produce a student with fewer mathematics skills. We think not, since one has to know mathematics to ask the computer to perform the operations. But this question, a serious one that cannot be ignored, waits for an answer. We are convinced that engineers in the future will use both numerical and symbolic computation and now is the time to start training them accordingly.

Like most innovations in engineering education, this one has been slowly introduced by teachers enthu-

Example: A MACSYMA Terminal Session

Laplace transforms offer a popular, efficient method of solving initial value problems involving constant coefficient linear differential equations. The main difficulty in their use is the inversion process. Students are assailed by a list of special techniques to assist inversion, such as shift theorems, partial fraction expansions and formulas for residues. In contrast to these, MACSYMA offers a work-saving function, ILT (for inverse Laplace transform), that automates the inversion process.

In this experiment students first use MACSYMA in the interactive (non-program) mode to work through a sample problem using Laplace transforms. They then write a program to automate the entire procedure, including inputting the coefficients and initial conditions, solving the problem and displaying the resulting solution.

We illustrate the procedure with the following sample problem:

$$x'' + x = e^t,\ x(0) = 1,\ x'(0) = 2$$

Here is part of a MACSYMA session in which this problem is solved using Laplace transforms. In MACSYMA each line is labeled. "C" labels represent the computer's replies. We begin by declaring X to be a function of T:

```
(C6) X:X(T);
(D6)                X(T)
```

Next we use the MACSYMA function ATVALUE to set the initial conditions:

```
(C7) ATVALUE(X,T=0,1);
(D7)                                1
(C8) ATVALUE(DIFF(X,T),T=0,2);
(D8)                                2
```

Then we state the differential equation and take its Laplace transform:

```
(C9) DIFF(X,T,2)+X=%E^T;
```

$$\text{(D9)}\quad X(T)_{TT} + X(T) = \%E^T$$

```
(C10) LAPLACE(D9,T,S);
```

$$\text{(D10)}\quad S^2\ \text{LAPLACE}(X(T),\ T,\ S) + \text{LAPLACE}(X(T),T,S) - S\ -2 = \frac{1}{S-1}$$

We then use the MACSYMA function SOLVE to algebraically solve for the Laplace transform of x:

```
(C11) SOLVE(D10,LAPLACE(X,T,S));
```

$$\text{(D11)}\quad [\text{LAPLACE}(X(T),\ T,\ S) = \frac{S^2 + S - 1}{S^3 - S^2 + S - 1}]$$

Finally the transform is inverted:

```
(C12) ILT(PART(D11,1),S,T);
```

$$\text{(D12)}\quad X(T) = \frac{3\ \text{SIN}(T)}{2} + \frac{\text{COS}(T)}{2} + \frac{\%E^T}{2}$$

The advantage of writing a program to accomplish the foregoing kind of calculation is that it saves a great deal of typing when there are numerous problems to solve. The following MACSYMA session shows how it works. After the program, named LAPTR, is called, the user is cued to enter the order of the equation, its coefficients and so on:

```
(C13) LAPTR( );

THIS PROGRAM USES LAPLACE TRANSFORMS TO SOLVE THE LINEAR CONSTANT

COEFFICIENT NTH ORDER NONHOMOGENEOUS ODE.

ENTER ORDER OF HIGHEST DERIVATIVE

2;
ENTER COEFFICIENT OF 2 TH DERIVATIVE OF X(T)
1;
ENTER COEFFICIENT OF 1 TH DERIVATIVE OF X(T)
0;
ENTER COEFFICIENT OF 0 TH DERIVATIVE OF X(T)
1;
ENTER THE NONHOMOGENEITY (RHS) F(T)
%E^T;
ENTER INITIAL VALUE OF X AT T=0
1;
ENTER INITIAL VALUE OF 1 TH DERIVATIVE OF X AT T=0
2;
```

$$\text{(D13)}\quad X(T) = \frac{3\ \text{SIN}(T)}{2} + \frac{\text{COS}(T)}{2} + \frac{\%E^T}{2}$$

siastic about the computer's potential to change the way we do things and the way learning takes place. We have not constructed an educational experiment that measures carefully and scientifically the outcome of our methods—the effect on our students of exposure to a new approach to teaching mathematics. We are, however, conducting a continuous experiment that produces new "data" every day.

We can report only anecdotally at this point on the results of this new way of teaching mathematics, based primarily on evaluations conducted in each course near the end of the semester. While the results of such surveys lack statistical validity, we pay attention to the written remarks when assessing the results of teaching experiments. We expect to report in the future more completely, and with more certainty, that our students' behavior is favorably changed, and that their understanding of mathematics and their performance in future courses—our real testing ground—has improved compared to that of students taught conventionally.

Conclusion

We have embarked on a new program of teaching mathematics that we believe can only improve understanding, enhance utility in applications and increase teaching efficiency. The presence of many small-scale computers with large memories will make it possible for more of us to exploit the computer programs available to teach more and, we think, better.

Although we think this innovation is unique in engineering education, we know that industrial engineering analysts are already making use of symbolic computations in the engineering workplace. At Lockheed, for example, K. C. Park has used MACSYMA to derive advanced finite element models for numerical codes. Computer algebra has also been used in industry to derive complex equations for robot manipulations. We expect engineering analysis in the future will be done completely on the computer, from derivation of equations for simulations to numerical simulation itself. Future CAD/CAM programs will incorporate symbolic methods and applied mathematics will reappear after a long absence in the engineering industrial laboratory. While we are in an experimental stage as far as teaching symbolic mathematical techniques is concerned, it is clear that all engineering curricula will have to provide some training in this new software tool to keep abreast of industrial mathematics needs and practices.

References

1. *muMATH-83 Reference Manual*, The Soft Warehouse, Honolulu, Hawaii, Sept. 1983.
2. Hearn, A.C., *REDUCE 2 Users Manual* (2nd edit.), Univ. of Utah, Salt Lake City, March 1973.
3. *SCRATCHPAD Users Manual*, IBM Research Report RA70, IBM Thomas J. Watson Research Center, Yorktown Heights, N.Y., June 1975.
4. *MACSYMA Reference Manual*, Version 10, MACSYMA Group, Dec. 1984. Available from Symbolics, Inc., 11 Cambridge Center, Cambridge, Mass. 02142.
5. Stoutmyer, D.R., "LISP Based Symbolic Math Systems," The Soft Warehouse, *Byte*, Aug. 1979, pp. 176-192.
6. Douglass, B.P., "Review of Extended muMATH," *80 Micro*, Nov. 1982, pp. 42-45.
7. Rand, R.H., *Computer Algebra in Applied Mathematics—An Introduction to MACSYMA*, Pitman, Boston, 1984.

Richard H. Lance is associate dean of engineering for undergraduate programs at Cornell University, where he previously served as associate dean of engineering for external affairs. He worked in industry for two years before joining the Cornell faculty in 1962.

Richard H. Rand is professor of mechanics at Cornell, where he has taught since 1967. He was a visiting professor at the University of Berkeley in 1981.

Francis C. Moon, chairman of the Department of Theoretical and Applied Mechanics at Cornell, has also taught at Princeton. He has published widely on the dynamics of solids and structures and recently directed some of his work to magneto-mechanics problems in fusion reactors.