Symbolic Manipulation Programs for the Personal Computer

KENNETH R. FOSTER AND HAIM H. BAU

The solution of scientific and engineering problems often requires setting up and solving a mathematical model. The use of a computer to assist in this task usually brings to mind finding a numerical solution (*1*), perhaps with a Fortran program or with an equation-solving program such as TK Solver Plus (*2*) or MathCAD (*3*). However, exact analytical solutions, if they can be obtained, are preferable in many respects to numerical solutions, but finding them with a computer requires a program that manipulates the symbols representing the equation. Although such symbolic manipulation (SM) programs have been in existence for many years, they have generally been designed for large mainframe computers that could handle their extensive memory requirements. In part because of the advent of inexpensive personal computers with large memory capabilities, a number of SM programs are now available for personal computers. We have reviewed four such programs: Derive (*4*), Maple (*5*), Mathematica (*6*), and Reduce (*7*). We also review Macsyma (*8*), which has long been available for larger computers and should soon be available for personal computers (Table 1).

At their simplest level, these programs can be used as symbolic calculators. After the user types in an expression, the program can perform the desired operation and return the answer. In addition, four of these programs incorporate a programming language that can be used to develop sophisticated applications (*9*). For example, investigators have used Macsyma and Reduce to evaluate partition functions in thermodynamics (*10*), generate algorithms for numerical solution of problems in fluid and solid mechanics (*11*), and find approximate solutions to complex problems by using perturbation techniques (*12*). All of these programs include a broad range of symbolic algebraic, calculus, vector, matrix, and equation-solving functions (see Table 2).

The following example illustrates the difference between SM and numerical programs. Consider the following set of nonlinear equations (*13*)

$$x1\ (x2 + x3) = 2a$$
$$x2\ (x1 + x3) = 2b$$
$$x3\ (x1 + x2) = 2c \qquad (1)$$

where a, b, and c are constants and $x1$, $x2$, and $x3$ are variables. A numerical solution requires assigning values to a, b, and c. In contrast, SM programs would solve this system of equations just as one would by hand, saving time, labor, and likelihood of error. The two solutions to this system of equations, as obtained with Macsyma, are shown in Fig. 1; obviously one would normally like to avoid solving problems such as this by hand.

Potential users need to be aware of some characteristics of SM programs. First, they can require a lot of memory. Macsyma (the most comprehensive SM program) requires more than 4 Mbyte of memory to load and still more (perhaps 8 Mbyte) to run effectively. For some problems it can exhaust the memory of a mainframe computer. Derive (for PC/MS-DOS computers) is limited by memory to calculations of modest length. The other programs can handle much larger and more diverse calculations than Derive, but require a Macintosh computer or a workstation with several megabytes of memory. The four programs we ran on a personal computer would, with greater or lesser frequency, cause the system to crash with loss of work, presumably because of memory overflow (it can be difficult to predict how much memory is "enough" for a given problem). The user needs to frequently store files of work in progress (an easy procedure) to protect against such an event. Another difficulty, the simplification of results, we discuss below.

We did not attempt speed comparisons. Execution time for any problem depends strongly on the computer used. Apart from this, pronounced differences might be expected among the programs when solving the same problem on the same computer. Extensive benchmark tests run on a Sun-3 workstation showed that Maple was about ten times faster than Macsyma (*14*) for typical SM calculators, with Mathematica performing at intermediate speeds. Neither did we attempt an exhaustive comparison of program capabilities, which are diverse and complex. Even for particular operations, such as integration, problems might be found that favor one program over the others. However, the programs have enor-

Table 1. Specifications of the software packages reviewed. The programs were tested on a Wyse PC computer, with MS-DOS 3.1 and 640 kilobytes (kbyte) of memory (Derive and Reduce), a Macintosh SE computer with 2.5 Mbyte of memory (Maple and Mathematica), and a Sun 3-60 computer with 8 Mbyte of memory (Macsyma). All of the programs are delivered in compiled form (however, Reduce is not delivered in executable form; see text). None of these programs is copy-protected. All of the programs except Derive are available in versions for many different kinds of computers, including personal computers (usually excluding PC/MS-DOS computers), workstations, minicomputers, and mainframe computers. Cost is for the version tested. Costs vary widely with version; in some cases substantial educational discounts are available.

Program	Version tested	Cost	Host computers	Memory requirements: Minimum	Memory requirements: Recommended
Derive	1.1	$200	IBM PC, PC-XT, AT, and compatibles*	512 kbyte	*640 kbyte
Macsyma	309.6	$1800	VAX, MicroVAX, Sun 3, Sun 4, Apollo, and Symbolics	4 Mbyte	8 Mbyte
Maple	4.2	$395†	Many computers, except PC/MS-DOS‡	1 Mbyte	1 to 2 Mbyte
Mathematica	1.1	$495† $795§	Macintosh, Macintosh II, Sun 3, NeXT, and 80386-based systems	2.5 Mbyte	5 to 8 Mbyte
Reduce	3.3	$495	Macintosh and PC/MS-DOS computers and many others	640 kbyte	

*Compatibles running PC/MS-DOS and PS/2. †Version for Macintosh computers. ‡For example, all of the non–PC/MS-DOS computers listed in this table except for Symbolics and the NeXT computer. §Version for Macintosh II and 80386-based computers.

K. R. Foster, Department of Bioengineering, University of Pennsylvania, Philadelphia, PA 19104.
H. H. Bau, Department of Mechanical Engineering and Applied Mechanics, University of Pennsylvania, Philadelphia, PA 19104.

mous differences in their ranges of capabilities and approaches, which we discuss. Extensive nontechnical descriptions of Macsyma, Maple, and Reduce have been published elsewhere (*15, 16*).

Mathematica

Mathematica, which was released in mid-1988, is an integrated mathemathics program developed by Wolfram that combines SM with many other capabilities. The program consists of two parts: the front end (a machine-dependent part that provides the superb user interface), and the kernel (a machine-independent part, which does the computations). The program requires at least 2.5 Mbyte of memory, but for effective use 5 Mbyte or more should be available. The documentation for this program consists of a well-written 749-page text (*17*); machine-specific instructions are provided in a separate booklet, which for the Macintosh occupies 179 pages. The program has extensive on-line help facilities.

Mathematica combines symbolic algebra, graphics, programming language, and numerical capabilities in a large, integrated package. It has more than 600 commands and built-in functions, including many special functions (for instance, it recognizes four different Bessel functions, which can have complex arguments). External "packages" supplied with the program provide additional capabilities, such as finding Laplace transforms.

Mathematica's superb graphics include two-dimensional (2-D) and 3-D plots, with other features. However, the default settings of the graphics package do not always lead to the most appealing results. For example, the default settings of Macsyma led to much more attractive plots of the "sombrero" function

$$3 \cos\,[(x^2 + y^2)/3]/(3 + x^2 + y^2) \quad (2)$$

These capabilities can be combined in useful ways. For example, the "notebook" allows one to combine text and graphics, together with input and output for the program. Using a notebook, one can develop a theory using SM, numerically evaluate and plot its results, write a document, and format that file for later typesetting. Mathematica's text and equations can be written in a form suitable for input to TEX; its figures are formatted in POSTSCRIPT.

Like most other SM programs, Mathematica contains an extensive programming language. A distinctive feature of Mathematica is the ability to define rules, which the system can implement where applicable. This somewhat resembles a rule-based programming language such as Prolog. For example, the system can be taught rules for manipulating logarithmic functions, such as:

$$\log[x\_\ y\_] := \log[x\_] + \log[y\_] \quad (3a)$$

$$\log[x\_^{a\_}] = a \log[x\_] \quad (3b)$$

where x_ refers to any expression (not just a variable). Whenever the log function is called, these rules are invoked. New rules can be introduced at any point, and functions already defined within Mathematica can be redefined after removing their protected status. This rule-based programming is more streamlined than the procedural programming used in other SM programs.

Although Mathematica has a broad range of SM functions, it lacks many of the features of Macsyma and Maple. For example, it can find Laplace transforms only for sim-

```
(c1) eqs: [x1*(x2+x3)=2*a, x2*(x1+x3)=2*b, x3*(x1+x2)=2*c]$

(c2) solve(eqs,[x1,x2,x3])$

(c3) ratsimp(%);

                 sqrt(- c + b + a) sqrt(c - b + a)
(d3) [[ x1 = - ---------------------------------,
                       sqrt(c + b - a)

                 sqrt(- c + b + a) sqrt(c + b - a)
          x2 = - ---------------------------------,
                       sqrt(c - b + a)

                          2    2             2
                         c  - b  + 2 a b - a
          x3 = - sqrt(- ---------------------) ],
                              c - b - a


                 sqrt(- c + b + a) sqrt(c - b + a)
        [ x1 =   ---------------------------------
                       sqrt(c + b - a)                ,

                 sqrt(- c + b + a) sqrt(c + b - a)
          x2 =   ---------------------------------
                       sqrt(c - b + a)                ,

                          2    2             2
                         c  - b  + 2 a b - a
          x3 = sqrt(- ---------------------)    ]]
                              c - b - a
```

Fig. 1. Use of Macsyma to solve the three coupled polynomial equations shown in the text. Line (c1) is the user input of the equations, (c2) the command to solve them, (c3) a command to simplify the results as rational functions, and (d3) the output of the program. The symbol $ included in lines (c1) and (c2) suppress the program's output in response to these statements.

Table 2. Selected capabilities of the programs reviewed. An x indicates that either the program or an add-on package supplied with can directly perform the function. Different programs perform these functions with varying capability. The programs have many additional functions beyond those listed here.

Function	Derive	Macsyma	Maple	Mathematica	Reduce
Integration-differentiation	x	x	x	x	x
Matrix manipulation	x	x	x	x	x
Systems of linear algebraic equations	x	x	x	x	x
Systems of nonlinear equations		x	x	*	
Ordinary differential equations		x	x		
Vector equations		x	†	†	
Laplace transforms		x	x	x	
Inverse Laplace transforms		x	x		
High-level language output (such as Fortran)	x	x	x	x	x
Generation of numerical code		x			x
Runs on a personal computer	x		x	x	x
2-D plotting	x	x	x	x	
3-D plotting	x	x		x	

*Only polynomial equations. †Only in specific coordinate systems.

ple cases and cannot find inverse transforms, nor can it perform vectorial calculations in general vector form (but only in specific coordinate systems). It has no routines for analytical solution of differential equations, nor built-in rules for manipulating complex symbolic expressions or simplifying trigonometric forms.

Other weaknesses include a **Solve** routine that can analytically solve only polynomial expressions, and a routine for numerical solution of differential equations (Runge-Kutta method) that can handle only a single first-order equation. Its abilities to simplify expressions are weak at times. (For example, an additional simplification step was needed to express most concisely the results of the differentiation operations shown in Fig. 2B). Some functions yield unexpected results; for example, converting the coefficients of a power series into floating point form (using the **N** function) yields results in a clumsy form. When run on a Macintosh, the program caused frequent crashes of the system with loss of work.

In Figs. 2 the use of Mathematica is shown. The program had no difficulty handling the singularities of the plotted function at $x = 0$ and ± 1 (Fig. 2A). Mathematica found both solutions to Eq. 1 in Fig. 1.

Macsyma

Macsyma is the only program we reviewed that does not presently run on a personal computer. We ran Macsyma on a Sun-3 workstation. Macsyma began development in 1969 in the Math Lab group at MIT Laboratory of Computer Science, and later was taken over by Symbolics, Inc. It consists of some 300,000 lines of code (in LISP) (*18*), and is probably the most powerful and comprehensive SM program available. It has a broad range of functions that include almost all of those in Mathematica, and many more.

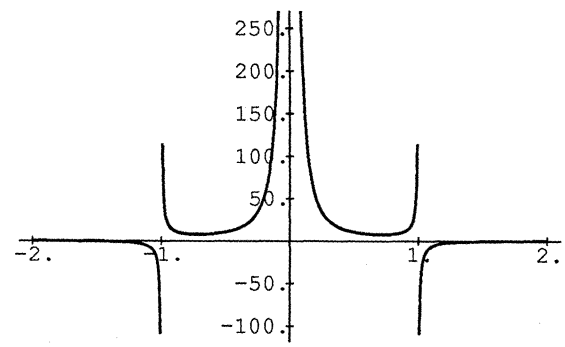
Macsyma's capabilities include solving ordinary differential equations, computing Fourier and Laplace transforms, and manipulating power series (to name a few). It has advanced features that are not found in other SM programs, including a library of all separable curvilinear coordinate systems and the ability to manipulate tensors. Recently added features include transforming differential equations into difference equations suitable for numerical computing, and perturbation analyses for finding approximate solutions to otherwise intractable equations. Macsyma can generate 2-D and 3-D plots and has extensive numerical capabilities. The use of Macsyma to solve a differential equation for a damped oscillator is shown in Fig. 3. Macsyma succeeded in evaluating both integrals in Fig. 2A.

A

```
Simplify[D[x^x^x,x]]

          x
 -1 + x + x                    2
x           (1 + x Log[x] + x Log[x] )

Integrate[x/(x^3-1),x]

                 1 + 2 x
Sqrt[3] ArcTan[-------]                          2
                Sqrt[3]    Log[1 - x]   Log[1 + x + x ]
----------------------- + ---------- - ---------------
           3                  3               6

Integrate[x*Sin[x]/(x^2+1),{x,0,Infinity}]

Pi
---
2 E

N[%,50]

0.57786367489546085895504659165634814956042551158221

Series[Sin[x]/x,{x,0,10}]

     2    4     6      8        10
    x    x     x      x        x           11
1 - -- + --- - ---- + ------ - -------- + O[x]
    6    120   5040   362880   39916800
```

B

```
Plot[Sqrt[4-x^2]/x^2/(1-x^2),{x,-3,3}]
```




```
Plot3D[Sin[2 x] Sin[2 y],{x,-Pi,Pi},
                         {y,-Pi,Pi}]
```

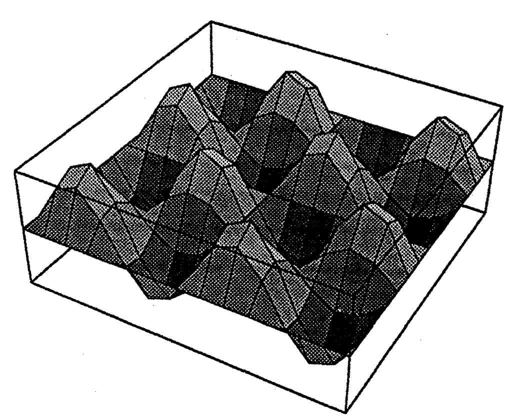


Fig. 2. (**A**) Use of Mathematica as a symbolic calculator to evaluate the derivative of x^{x^x} with respect to x and simplify the result, an indefinite integral; a definite integral; the last result expressed in floating point form with 50 significant digits; and a Taylor series to the tenth power in x. (**B**) Two- and three-dimensional plots from Mathematica. The plots were created with the default settings of the plotting routines.

Macsyma is a complex and heterogeneous program that is somewhat difficult to use. Its 487-page reference manual (version 11) is inadequate in several respects: it lacks sufficient detail, and appears to be misleading in places. (We did not review version 13 of the manual, which was released in December 1988.) Macsyma has a 259-page, well-written user's guide; the **describe** command calls up parts of the manual on-line, and the **apropos** command provides help with the spelling and syntax of commands. Macsyma has a cumbersome line editor. [We partially overcame this difficulty by running Macsyma on a Sun 3-60 workstation, with a Macintosh computer running Versa-Term (*19*) as a remote terminal.]

Maple

Maple has been under development at the University of Waterloo (Canada) since 1980. Versions are presently available for many computers that can address 2 Mbyte or more of memory. Its documentation consists of a well-written 115-page tutorial introduction, and a 403-page reference manual with many examples. The manual is also available on-line with the help command.

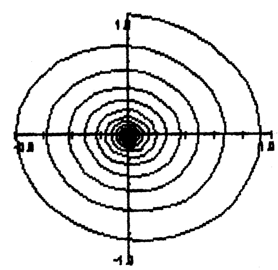
Maple consists of a system kernel (written in C) and a library of mathematical routines (written in the Maple Language). These libraries include many routines that perform algebraic manipulations, linear algebra and matrix manipulation, number theory, orthogonal polynomials, and so on. Other SM functions include solution of differential equations and Laplace and inverse Laplace transforms. A user can define additional functions and procedures with its extensive programming language.

The program combines extensive SM capabilities with comparative ease of use. It lacks many of the SM functions of Macsyma, such as a numerical root finder for nonlinear equations and the ability to manipulate vectors as objects. Maple has limited graphics capabilities compared with Macsyma and Mathematica; it can only generate 2-D plots, and cannot interleave graphics, Maple expressions, and text in the same window. (However, it allows parametric and polar plots, and the user can zoom in or out of plots or change their aspect ratios conveniently with a mouse.) When run on a Macintosh SE computer, Maple sometimes caused sudden and unexpected crashes in the system, apparently due to memory overflow.

Maple successfully evaluated the indefinite but not definite integral in Fig. 2A. The use of Maple to solve the damped oscillator problem is shown in Fig. 4, illustrating its use of windows. We failed to solve a set of two coupled linear second-order differential equations, corresponding to two coupled oscillators: Macsyma solved that problem with no difficulty. Maple found only one of the two solutions to Eq. 1.

Fig. 3. Use of Macsyma to calculate the motion of a damped oscillator as a function of time. The line numbers are provided by the program. The dollar sign ($) at the end of the statements suppress Macsyma's response. The commands are as follows: (c1) Set up initial conditions at time $t = 0$; (c2) set up the equation of motion with y as the displacement; (c3) calculate the Laplace transform of the equation of motion subject to the initial conditions; (c4) solve the linear algebraic equation (c3) for the Laplace transform of y; (c5) calculate the inverse Laplace transform of (c4); (c6) calculate the derivative of the displacement (velocity); and (c7) plot the phase diagram, with velocity plotted along the vertical axis and displacement along the horizontal axis.

```
(c1) ic: (atvalue(y(t),[t-0],0),atvalue(diff(y(t),t),[t-0],1))$

(c2) eq: diff(y(t),t,2)+0.1*diff(y(t),t)+y(t)=0$

(c3) leq: laplace(eq,t,s);

(c4) lsol: linsolve(leq,'laplace(y(t),t,s))$

(c5) sol: ilt(lsol,s,t);

(c6) dy: diff(part(sol,1,2),t)$

(c7) paramplot2(part(sol,1,2),dy,t,0,150), plotnum:700;
```



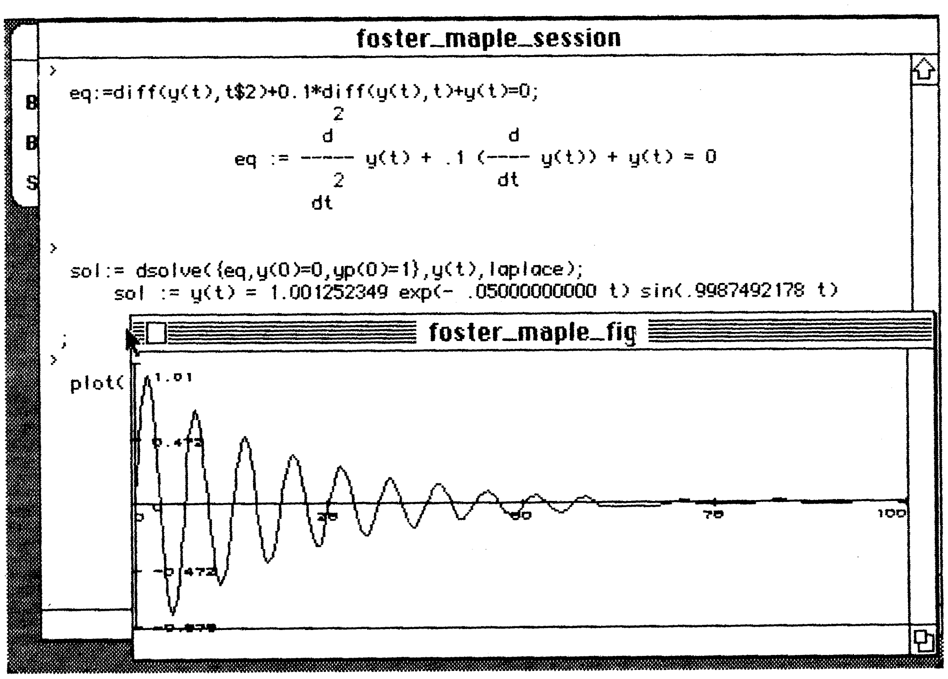


Fig. 4. Use of Maple to solve the damped-oscillator problem presented in Fig. 3. The displacement is plotted as a function of time in one of the windows.

Derive

Derive is a new program developed for the IBM PC and compatibles by Rich and Stoutemyer, the creators of muMATH, an early SM program for personal computers (*20*). Unlike the other programs considered here, Derive was designed to run on PC/MS-DOS systems. It has comparatively limited capabilities, but is an effective and easily used symbolic calculator and is supplied with a clearly written 98-page booklet.

Derive has menu-driven commands (reminiscent of Lotus 1-2-3) and extensive help files. Novices will find it simpler and more comfortable to use than the other programs considered here. Expressions entered by the

Fig. 5. (**A**) Screen printout from Derive. (1) is the indefinite integral shown in Fig. 2A, which is evaluated to yield (2). Differentiation of (2) yields (3); expanding the denominator of (3) yields (4). The command line and other information about the program are shown at the bottom. A cursor window, positioned by the user over the denominator of (3), instructs Derive to expand the indicated part of the expression. (**B**) Screen printout from Reduce. Line 6 evaluates the indefinite integral of Fig. 2A. The result is differentiated with respect to x to yield the result in line 7. This expression is repeated in line 9 in Fortran syntax.

A

1 $$\int \frac{x}{x^3 - 1}\, dx$$

2 $$\frac{\sqrt{3}\ \mathrm{ATAN}\left[\frac{2\sqrt{3}\,x}{3} + \frac{\sqrt{3}}{3}\right]}{3} - \frac{\mathrm{LN}\,(x^2 + x + 1)}{6} + \frac{\mathrm{LN}\,(x - 1)}{3}$$

3 $$\frac{x}{(x^2 + x + 1)\,(x - 1)}$$

4 $$\frac{x}{x^3 - 1}$$

```
COMMAND: Author Build Calculus Declare Expand Factor Help Jump soLve Manage
         Options Plot Quit Remove Simplify Transfer moVe Window approX
Enter option
User                C:REV.MTH          Free:100% Insert        Derive Algebra
```

B

```
6: int(x/(x^3-1),x);

                    2*X + 1            2
 2*SQRT(3)*ATAN(-----------) - LOG(X  + X + 1) + 2*LOG(X - 1)
                   SQRT(3)
-----------------------------------------------------------------
                              6

7: df(ws(6),x);

   X
--------
  3
 X  - 1

8: on fort$

9: ws(6);
      ANS=(2.*SQRT(3)*ATAN((2*X+1)/SQRT(3))-LOG(X**2+X+1)+
     . 2.*LOG(X-1))/6.
```

Fig. 6. Symbolic calculation of the real part of coth $[(u + i\,v)/2]$. (**A**) Output from Macsyma. Line (c1) is the input from the user, (c2) the command for trigonometric simplification of the results, (d2) the program output. The symbols %i denote $\sqrt{-1}$. The time required for the calculation (on a Sun-3 workstation) is shown also. (**B**) Results from the same calculation, from Derive. Tildes at end of each line indicate continuation to next line. This example illustrates the problem of effective simplification (see text).

A

```
(c1) x-%i*y=a*coth((u+%i*v)/2)$

(c2) trigsimp(realpart(%));
Totaltime= 3266 msec. GCtime= 1250 msec.

                         a sinh(u)
(d2)           x =  -----------------
                     cos(v) - cosh(u)
```

B

```
RE(COTH((u+#i*v)/2))

(#e^(2*u)+#e^(-2*u)-2*(2*COS(v)^2-1))*(COS(v/2)*(#e^(u/2)+#e^(-u/2))+#i*SIN(v/~
2)*(#e^(u/2)-#e^(-u/2)))*(-COS(v/2)*(#e^(u/2)-#e^(-u/2))+#i*SIN(v/2)*(#e^(u/2)~
+#e^(-u/2)))*(COS(v)*#e^u+COS(v)*#e^(-u)-2+#i*SIN(v)*(#e^(u/2)-#e^(-u/2))*(#e^~
(u/2)+#e^(-u/2)))*(-(COS(v)*#e^u+COS(v)*#e^(-u)+2)+#i*SIN(v)*(#e^(u/2)-#e^(-u/~
2))*(#e^(u/2)+#e^(-u/2)))/(2*(#e^(2*u)-(4*COS(v)*#e^u+4*COS(v)*#e^(-u)-(#e^(-2~
*u)+2*(2*COS(v)^2+1))))*(#e^(2*u)+4*COS(v)*#e^u+4*COS(v)*#e^(-u)+#e^(-2*u)+2*(~
2*COS(v)^2+1))*(#e^u+#e^(-u)-2*COS(v)))-(COS(v/2)*(#e^(u/2)+#e^(-u/2))+#i*SIN(~
v/2)*(#e^(u/2)-#e^(-u/2)))*(-COS(v/2)*(#e^(u/2)-#e^(-u/2))+#i*SIN(v/2)*(#e^(u/~
2)+#e^(-u/2)))/(2*(#e^u+#e^(-u)-2*COS(v)))
```

user, or results of previous calculations, are displayed in a window on the screen in an easy-to-read multiline format (Fig. 5A). A user can scroll through these equations, and (by positioning a cursor) manipulate expressions or parts of expressions, or print all or part of the file.

The program performs basic algebraic, calculus, and matrix operations on single expressions. It recognizes 55 functions, including the standard trigonometric, hyperbolic, matrix, and complex functions. It can solve systems of linear equations and many single nonlinear equations. Output can be written in Fortran or Pascal syntax, and transferred to programs with Sidekick (*21*) or similar programs.

In addition to SM, Derive has 2-D and 3-D graphics and numerical capabilities, including numerical evaluation of integrals (using an adaptive Simpson's rule) and numerical solution of equations (using a bisection technique). Derive has no programming capability; however, a user can define functions for use in later calculations.

Derive performs well for the problems for which it is intended. Its integration routine can evaluate many standard forms. Its graphics package has useful features, such as the ability to create parametric plots (plotting one function against another) or to "zoom" scales or easily relocate a plot on the screen. Derive effectively simplifies expressions, with the only obvious problem associated with hyperbolic functions. (It expands such functions in terms of exponential functions, leading to complex results.) Derive could evaluate the first but not second integral shown in Fig. 2 (Fig. 5A). It could not solve Eq. 1 nor the differential equation shown in Fig. 3.

Derive has other limitations that are likely to bother some users. Its graphics and numerical routines are slow and cumbersome to use (compared with other programs for PC/MS-DOS computers such as MathCAD). Lengthy expressions extend off the edge of the screen and require a clumsy scrolling maneuver to view or manipulate.

Nevertheless, Derive is a useful program for comparatively small problems. One of us (K.R.F.) extensively used a prerelease version of Derive in preparing lectures for a senior-graduate level course in biomedical signal analysis. It easily handled calculations that would be uncomfortably long by hand, for example, deriving transfer functions for filters. This work was accomplished with few cases of memory overflow. Derive was the only program considered here that ran effectively on PC/MS-DOS computers. It should find wide use as a symbolic calculator among science and engineering students.

Reduce

Reduce was developed beginning in 1963 by Hearn for application in high-energy physics (*22*), and subsequently extended by Hearn and others with applications as diverse as general relativity and wind turbine design (*23*). The program is written in LISP and is available for computers ranging from the Cray supercomputer to the IBM PC. We include it in this review because of its availability for the IBM PC, and only tested that version.

Reduce is advertised as a general-purpose SM program. However, it differs greatly in capability from Macsyma (which is in some respects a comparable program). Reduce has a variety of commands for manipulation of algebraic expressions (such as expansion and factoring of polynomials) and the basic algebraic, calculus, and matrix operations. However, it has no graphics capabilities, and few of the built-in SM functions found in the other programs reviewed.

Reduce appears to be aimed at programmers who can develop extensive application programs with its language. A few such programs supplied with Reduce can be used to generate Fortran code, perform specialized calculations in differential geometry, and compute symmetry properties of systems of partial differential equations. (Other packages are available from Hearn, for computing Laplace transforms, solving equations, and so forth.) The great portability of Reduce is a valuable feature: a programmer might develop an application program on a workstation or other relatively small computer, and later run it on a much larger machine. Reduce provides a pattern-matching facility for users not interested in extensive programming; many successful applications are based on this.

The PC/MS-DOS version of Reduce comes on 12 disks [including tutorial programs, test files, an appropriate version of LISP (UO-LISP), and application programs], together with several hundred pages of documentation and a comprehensive 329-page text (*24*). However, Reduce is difficult to run on PC/MS-DOS computers in several respects. Installation of Reduce requires several steps: the user must first run UO-LISP to build up a file from some or all of the 70 files that constitute the entire package. Any subsequent change in memory allocation (from adding or removing memory-resident programs or executing some DOS commands) prevents its running. Even limited versions of Reduce quickly exhausted the memory of the computer we used, causing the system to crash with loss of work. Some of these problems are likely to be avoided when running the program on a larger computer. Reduce successfully integrated the indefinite but not definite integral shown in Fig. 2 (Fig. 5B) and could not solve Eq. 1.

Discussion

Two programs for SM, Macsyma and Reduce, both date from the 1960s and are now mature products. The other programs considered here are important additions to this class of software. No single program is likely to satisfy all needs. These programs can be productively used as sophisticated calculators, and learned in a few hours. Of the large programs, Mathematica is the easiest to use in this mode. However, Macsyma and Maple have more symbolic functions, with the former being more complete and the latter being easier to use.

The effective simplification of results is the Achilles heel of SM programs. This problem can be striking in appearance and devastating in effect. For example, in Fig. 6 the output of Derive and Macsyma are compared for calculating the real part of the expression coth $[(u + i\,v)/2]$. Maple evaluated this expression without trouble. Mathematica could not, because it lacks routines for SM of complex expressions and trigonometric simplification capabilities. (A package for SM of complex expressions was separately supplied by the vendor.) Effective simplification routines are important in less obvious ways as well: a final result might be concisely written, but intermediate results (not visible to the user) might be very large, consuming much memory and time to manipulate. This effect (called "intermediate expression swell") makes it difficult to predict the memory and computation time needed for any SM task. The simplification capabilities of Macsyma and Maple are far superior to those of Mathematica and are a great advantage. Except for its handling of hyperbolic functions, those of Derive appear to be quite good.

For more advanced or specialized applications (such as generating differential equations from basic physical laws, coordinate transformations, perturbation analyses, and generation of numerical algorithms), a user would need the programming capability of these programs. The programming languages are broadly similar.

Mathematica is a special case, by virtue of its user-friendliness, rational packaging, consistency, and broad range of SM and numerical functions. Although the present version lacks many useful features, and has deficiencies in symbolic manipulation, most technical users would find it an attractive choice. Many add-on products will probably be developed to further enhance its usefulness. However, a complex program such as this is likely to have subtle bugs that will be corrected only slowly, and a user should be careful in accepting the results of the program (*25*). Macsyma and Maple have been in use much longer, and are likely to be more reliable. Improved numerical algorithms are needed, perhaps an adaptation of standard numerical analysis routines (*26*). Also needed are special purpose discipline-oriented materials, similar to Phaser (*27*), for modeling dynamical systems. Macsyma has a comparable (and perhaps larger) range of functions, but is less user-friendly and does not as yet run on a personal computer.

REFERENCES AND NOTES

1. K. R. Foster, *Science* **240**, 1353 (1987).
2. TK Solver Plus. Universal Technical Systems, Inc., 1220 Rock Street, Rockford, IL 61101.
3. MathCAD. MathSoft, Inc., One Kendall Square, Building 200, Cambridge, MA 02139.
4. Derive. Soft Warehouse, Inc., 3615 Harding Avenue, Suite 505, Honolulu, HI 96816.
5. Maple. Symbolic Computation Group, Computer Science Department, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1. [Vendor (Macintosh): Brooks/Cole Publishing Co., 511 Forest Lodge Road, Pacific Grove, CA 93950.)]
6. Mathematica. Wolfram Research, Inc., P.O. Box 6059, Champaign, IL 61821.
7. Reduce. A. C. Hearn, RAND Corp., P.O. Box 2138, Santa Monica, CA 90406. Personal computer version available from Northwest Computer Algorithms, P.O. Box 1747, Novato, CA 94948.
8. Macsyma. Symbolics, Inc., 257 Vassar Street, Cambridge, MA 02139. A version of Macsyma for the IBM 386 and compatible computers is expected to be released by late spring 1989.
9. A. C. Hearn, *Algebraic and Symbolic Methods* (School on Advanced Techniques in Computational Physics I.C.T.P., Trieste, 1986).
10. G. Jacucci and M. Rasetti, *J. Phys. Chem.* **91**, 4970 (1987).
11. S. L. Steinberg and P. J. Roache, *J. Comput. Phys.* 57, 251 (1985); in *Symbolic Computation in Fluid Mechanics and Heat Transfer*, H. H. Bau, T. Herbert, M. M. Yonvnovich, Eds. (American Society of Mechanical Engineers, New York, 1988), vol. HTD-105, pp. 81–86; P. S. Wang, *ibid.*, pp. 87–92.
12. R. H. Rand and D. Armbruster, *Perturbation Methods, Bifurcation Theory, and Computer Algebra* (Springer-Verlag, New York, 1987).
13. R. Pavelle, Ed. *Applications of Computer Algebra* (Kluwer Academic, Boston, 1985), p. 35.
14. P. Flajolet (INRIA, Rocquencourt, France), personal communication.
15. J. Fitch, *J. Symbolic Comput.* **1**, 211 (1985); R. Pavelle and P. S. Wang, *ibid.*, p. 69.
16. B. W. Char *et al.*, *ibid.* **2**, 179 (1986).
17. S. Wolfram, *Mathematica* (Addison-Wesley, Redwood City, CA, 1988).
18. R. Pavelle, in *Applications of Computer Algebra*, R. Pavelle, Ed. (Kluwer Academic, Boston, 1985), p. 3.
19. Versa-Term Pro, Version 1.20, developed by Abelbeck Software. Peripherals Computers and Supplies, Inc., 2457 Perkiomen Avenue, Mt. Penn, PA 19606.
20. muMATH. Soft Warehouse, Inc., 3615 Harding Avenue, Suite 505, Honolulu, HI 96816.
21. Sidekick. Borland International, 4585 Scotts Valley Drive, Scotts Valley, CA 95066.
22. A. C. Hearn, *Commun. ACM* **9**, 573 (1966).
23. A. D. Garrad and D. C. Quarton, *J. Sound Vib.* **109**, 65.
24. G. Rayna, *Reduce, Software for Algebraic Computation* (Springer-Verlag, New York, 1987).
25. J. Barwise, Ed., *Not. Am. Math. Soc.* **35**, 1333 (1988). The incorrect results noted (version 1.03) appear to have been corrected in version 1.1.
26. For example, W. H. Press, B. P. Flannery, S. A. Teukolsky, W. T. Vetterling, *Numerical Recipes* (Cambridge Univ. Press, Cambridge, 1986).
27. H. Kocak, *Differential and Difference Equations Through Computer Experiments* (Springer-Verlag, New York, 1986).