Computer Algebra Support Project

A Guide to Computer Algebra Systems

David Harper Computer Laboratory
Chris Wooff Computer Laboratory
David Hodgkinson Applied Mathematics and Theoretical Physics

The University, P.O. Box 147, Liverpool L69 3BX

Fourth edition (4.01)
March 1990

DISCLAIMER

The information in this document was compiled on the basis of the authors' experiences with the systems described and information provided by the developers of the systems. The tables in section 3 describe those versions of the eight computer algebra systems that were in use on machines at the University of Liverpool Computer Laboratory at the time this document was being prepared.

All of the developers were sent a copy of this document and asked to comment upon its accuracy. However, the authors retained final editorial discretion. By its nature, this document is a concise summary which should not be regarded as a complete description of any of the systems. For full information about the capabilities of the latest release of each of the computer algebra systems, readers should contact the developers at the addresses given in section 4 of this document.

The authors of this document and organisations associated with the Computer Algebra Support Project have no commercial interest in any of the systems described herein.

Publication history

1st edition: July 1988.

2nd edition: July 1989, for IUSC worskhop on algebraic computing, University of Liverpool, 4-5 July 1989.

3rd edition: September 1989, reprinted with minor changes February 1990.

4th edition: March 1990. First TEX edition.

TABLE OF CONTENTS

1 Introduction

This document describes the eight main general-purpose computer algebra systems currently available. It begins by discussing the general capabilities of the systems and their applications in research and teaching where mathematics is involved. The second section gives a more detailed description of the features of computer algebra systems and presents a comparison of the eight systems. Tables showing their relative merits are included.

The final section contains information on the availability of the various systems including the addresses of the companies or institutions which distribute them and the latest prices.

The idea of producing such a guide is due to Dr John Hosack of Colby College, Waterville, Maine 04901, USA.

This document was formatted using the TEX typesetting system.

2 Introduction to computer algebra

In the academic world and in industry, one of the main uses of computers is to solve complex mathematical problems using numerical techniques. Software packages such as the Numerical Algorithms Group (NAG) library are used to obtain numerical solutions in many areas of science and engineering. It is less well-known that computers can also be used to obtain algebraic solutions to many classes of mathematical problem. Computer algebra systems (CAS's) developed during the last 10-20 years can solve a wide variety of problems in algebra and calculus. They typically provide the following facilities:

- Manipulation of power series and expressions involving functions such as sine, logarithm and exponential.
- Symbolic differentiation and integration of a large number of functions.
- Algebraic solutions of:
  - Polynomial equations up to 4th order.
  - Sets of simultaneous linear equations, including over-determined systems.
  - Some sets of simultaneous non-linear equations.
  - Some ordinary differential equations.
- Manipulation of matrices whose components may be algebraic expressions includes symbolic matrix inversion.
- Vector algebra and calculus in general orthogonal curvilinear coordinates systems.
- Arbitrary-precision arithmetic, integer and floating-point.
- Number-theoretic calculations.
- The ability to save the results of calculations for subsequent use.
- Facilities for plotting functions in 2 and 3 dimensions.

This list is not exhaustive. There are eight major CAS's and they vary in their abilities to solve particular types of problem. A detailed comparison is given in Section 3.

The use of CAS's in research and teaching is growing rapidly due to two reasons. Firstly, there is an increasing awareness in the research community that computer algebra provides a very powerful tool to help to solve problems that may previously have been considered intractable due, perhaps, to the size and complexity of the expressions to be manipulated. Secondly, CAS's are now available for a wide range of machines: not only mainframes, but also minicomputers, a large number of workstations and microcomputers such as PC-compatibles, the Atari ST and the Apple Macintosh. Thus computer algebra is being brought within easy reach of a greater number of potential users. Among the most important recent developments in computer algebra is Derive, a system designed for use on IBM/PC-compatible machines, and Mathematica, a system with extremely powerful graphics capabilities.

In the next section, the facilities which are available to the user of a general-purpose CAS are discussed in more detail. A comparison of the eight major systems and their relative capabilities in various branches of algebra and calculus is provided. These systems are:

- Derive
- MACSYMA
- Maple
- Mathematica
- muMATH (RIEMANN)
- REDUCE
- Scratchpad
- SMP

3 Capabilities of computer algebra systems

3.1 Overview of CAS's

All of the general-purpose CAS's are designed to be able to manipulate and evaluate a wide variety of mathematical expressions including power series, transcendental functions, derivatives, integrals, systems of equations and matrices. This has two consequences. Firstly, the syntaxes of the various systems are very similar, so it is quite easy to use one system after becoming familiar with another. Secondly, all the systems share a common core of basic facilities. Elementary algebraic operators are available together with basic simplification rules, calculus and equation solving.

In addition to these facilities, each system has its own special features. For example, Mathematica is capable of producing graphical representation of functions (with the appropriate hardware) while MACSYMA includes a suite of routines for handling tensors. At a more fundamental level, the different systems vary in their ability to solve certain common types of problem. Integration is an important example: the range of functions which each system is capable of integrating differs greatly between systems, as will be shown in the following sections.

The main facilities available in all (or most) of the general-purpose CAS's are given below.

Interactive use

Expressions may be entered at the terminal and the result will be displayed on the screen, usually in a two-dimensional notation in order to show superscript indices correctly. For example, `(1+X)**3` might result in the display

```
                 2     3
1 + 3 X + 3 X  + X
```

File handling

Expressions and commands can be read from files. This is useful both for batch processing of large programs and for loading frequently-used expressions or definitions without the need to re-type them.

Results can also be written to files, either in the two-dimensional 'pretty-printed' format shown above or in the same format as input, enabling expressions to be saved and then read during a subsequent session.

In addition, some systems can write algebraic expressions as FORTRAN or C statements or in TEX or troff/eqn format.

Polynomial manipulation

This is the most basic aspect of all CAS's. It includes simple algebraic expansion and simplification of expressions by extracting common factors. Some systems may also be able to factorise multinomial expressions.

Recognition of elementary special functions

All systems recognise a number of transcendental functions and their various algebraic properties. These functions are:

- Sine, cosine and tangent (and their inverses)
- Natural logarithm and exponential
- Hyperbolic sine, cosine and tangent (and their inverses) The systems can differentiate and integrate certain expressions involving these functions, as well as applying some simplification rules and values for special arguments (e.g. $\sin 0 \mapsto 0$).

Arithmetic

Numbers within expressions (coefficients and indices) are usually represented as integers or ratios of integers. These may have as many digits as are required to give the exact value. Thus for example, exact values of large factorials may be evaluated using CAS's. 200! is

```
78865786736479050355236321393218506229513597768717
32632947425332443594499634033429203042840119846239
04177212138919638830257642790242637105061926624952
82993111346285727076331723739698894392244562145166
42402540332918641312274282948532775242424075739032
40321257405579568660226031904170324062351700858796
17892222278962370389737472000000000000000000000000
0000000000000000000000000
```

a number 375 digits long.

It is also possible to evaluate numbers in floating-point format. The user can generally specify how many significant figures are to be used in floating-point calculations. Most systems do not limit the number of significant figures. Such arbitrary-precision arithmetic is carried out in software and is relatively slow compared to floating-point operations carried out in hardware. Some systems allow numerical evaluation of algebraic expressions and functions and special constants such as π and e.

Note that SMP is an exception to the general rule. By default, it represents polynomial coefficients as floating-point numbers with a precision equal to that of the local double-precision type in C. Since the coefficients are not exact, problems may arise when nearly-equal terms are compared or subtracted or when a very small number is added to a very large one.

Differentiation

Differentiation can be applied to any expression involving polynomials and other functions which are recognised by the CAS. The appropriate rules for differentiation of products and quotients are applied automatically, as is the chain rule.

Mixed and ordinary derivatives can generally be calculated using a single instruction which is a variant on that for single derivatives.

Integration

The integration of complicated expressions is one of the main applications of CAS's. All of the general-purpose systems are able to integrate expressions involving polynomial terms and elementary transcendental functions and certain combinations of these. Some substitutions and the rule for integration by parts are applied where necessary.

The repertoire of integrable functions varies between systems depending upon the algorithms which are built into the system and the set of mathematical functions that the system recognises. Most systems use a combination of table look-up and methods such as the Risch-Norman algorithm.

Equation solving

All of the general-purpose systems can solve polynomial equations up to degree 4 in the independent variable. Thus algebraic solutions may be obtained for cubic and quartic equations very easily, though the answers are often quite cumbersome.

Systems of simultaneous linear equations with both numeric and algebraic coefficients can also be solved exactly without recourse to numerical approximations. Again, the results tend to be rather unwieldy in all but the simplest cases. Some systems are able to obtain algebraic least-squares solutions to over-determined sets of equations.

Most of the systems can solve some first- and second-order ordinary differential equations. As in the case of integration, this capability varies between systems.

Substitution

The ability to substitute an expression or numerical value in place of a given variable is very useful. It allows expressions to be built up or altered in a carefully-controlled way. All systems have the substitution facility.

Matrix manipulation

All the systems provide a method for declaring and initialising matrices. Multiplication and addition of matrices can be carried out simply by reference to the names of the matrices. In some systems, the scalar algebra operators can also be applied to matrices. For example, two compatible matrices A and B can be multiplied simply by entering `A*B`.

Algebraic inversion of matrices can also be performed and this is another important application of CAS's, though as in the case of equation solving, the result may often be a set of large and complicated expressions.

Defining new rules and procedures

All modern programming languages provide a facility for defining new functions and procedures to carry out frequently-used sequences of operations. CAS's also allow the user to define procedures. These may be used to perform some algebraic operation upon an expression or to introduce a new mathematical function to the system. In the latter case, it is generally possible to define certain properties of the function such as its derivative and integral and the simplification rules which can be applied to expressions containing the new function.

3.2 General description of the eight CAS's

Note: a list of the machines and operating systems for which each system is available is given in Section 4, and in tabular form in Appendix A.

3.2.1 REDUCE

REDUCE is the oldest of the general-purpose CAS's. It was developed originally by A.C. Hearn at the RAND Corporation and it is being continuously upgraded and improved by Hearn and his colleagues and also by a number of other contributors, notably J.P. Fitch and J.H. Davenport at Bath University, A.C. Norman at Cambridge University and E. Schrüfer at GMD, Bonn, West Germany. It is written in LISP and is available for a wide range of machines from micros to supercomputers. It is the most widely-used CAS in the UK and Europe.

REDUCE has rather few built-in functions, though it allows the user to add new simplification and substitution rules and to define new functions and procedures very easily. It can be tailored to meet specific needs if the existing facilities are insufficient. Definitions can also be added at the LISP level, allowing the system to be extended in more sophisticated ways, for example by adding new data types to REDUCE.

Unlike other CAS's, REDUCE does not allow algebraic simplification rules to be applied selectively to an expression. It evaluates an expression by repeatedly applying *all* currently active rules until no further change can be made to the expression. For example, consider the rules which allow products of sine and cosine to be re-written in terms of sine and cosine of compound angles. One such rule is

$$\sin A \cos B \mapsto \frac{1}{2}(\sin(A+B) + \sin(A-B))$$

and other CASs have procedures which apply these rules to an expression. In REDUCE however, the rules must first be defined using a command such as

```
FOR ALL A,B LET SIN(A)*COS(B)=1/2*(SIN(A+B)+SIN(A-B));
```

After that, they will be applied to all subsequent expressions until the user removes the rules using an appropriate `CLEAR` command.

Another characteristic feature of REDUCE is the use of switches to control the simplification and display of results. The mode of calculation and output can be altered so that, for example, numerical coefficients in an expression will be printed in floating-point form rather than as integers or rational fractions.

REDUCE is normally supplied with an on-line user manual which describes in all the facilities and functions provided by the system. For the beginner there is a set of interactive lessons which can be run within a REDUCE session and which guide the user through the system step-by-step.

There is a book entitled "REDUCE: Software for Algebraic Computation" by Gerhard Rayna (Springer, 1987) which describes the REDUCE system with the help of a large number of examples. This is a useful introduction to REDUCE for the beginner who may find the user manual somewhat daunting. In addition, the book "Computer Algebra" by Davenport, Siret and Tournier (Academic Press, 1987) also provides information for experienced REDUCE users.

3.2.2 Maple

Maple is a powerful and compact system developed at the University of Waterloo, Ontario. It was designed for use on workstations and mini-computers where a number of people would be using it at the same time. It is therefore an ideal multi-user system.

Maple is available for a wide range of machines from the Atari ST to CRAY supercomputers.

Maple has a large number of built-in functions and procedures for manipulating expressions. There are packages of routines for a variety of special applications such as group theory, linear algebra, Gröbner bases, number theory, formal power series, linear optimisation and statistics.

Facilities exist for analysing expressions and manipulating parts of expressions. The user can add procedures to carry out operations and apply algebraic rules that are not already defined in Maple.

Maple is supplied with a tutorial book and reference manual plus a comprehensive on-line help facility. A Maple user guide has been written by David Harper and is available from the University of Liverpool Computer Laboratory.

3.2.3 muMATH

muMath was designed for use on microcomputers and so it is a very compact system. It was written by David Stoutmeyer and Albert Rich of Soft Warehouse Inc to run on CP/M-80 and PC-DOS (MS-DOS) machines with as little as 64k of RAM. It has many of the features of larger systems and includes facilities such as vector algebra and calculus which are not to be found in other systems in such a convenient form.

The muMATH system fits easily onto a single 360k floppy disk and is provided with a user manual. There is a useful introductory book describing the system, "muMATH: a Microcomputer Algebra System" by C Wooff and D Hodgkinson, published by Academic Press.

muMATH is no longer marketed for CP/M and PC-DOS systems but an Apple II version is distributed and supported by Dr Edwin Dickey of the University of South Carolina. In addition, a much-enhanced version re-named RIEMANN has been developed for use on Atari ST computers by Alexander Niemeyer. The information given in Section 3 for muMATH also applies to RIEMANN. Contact addresses for Edwin Dickey and for Alexander Niemeyer are given under the muMATH heading in Section 4.

3.2.4 MACSYMA

MACSYMA is a very large and powerful system. It was written originally at MIT and it is now marketed by Symbolics Inc. It is available for a limited number of machines, principally UNIX workstations.

MACSYMA makes heavy demands on the machines on which it is run. For example, on a VAX 11-780 it requires 20Mb of virtual memory in order to run the basic system, and the 386-based PC version requires 4Mb. However, it is able to solve a very wide range of mathematical problems because it is well-equipped with utilities and library procedures, many of which have been contributed by MACSYMA users in the United States and Europe. Its strengths play a larger role in research work rather than in teaching.

MACSYMA is provided with a set of documents including an introductory primer, a complete reference manual and a quick-reference card. There is also a good on-line help system which includes 500 executable examples and can even recognise parts of (half-remembered) keywords.

3.2.5 SMP

SMP is the predecessor of Mathematica. It recognises a wide range of mathematical functions, though this is largely limited to numerical evaluation. It was developed at the California Institute of Technology and is now marketed by Innovus International Inc.

3.2.6 SCRATCHPAD

The SCRATCHPAD system is significantly different from the other seven packages reviewed. Its design is based upon abstract datatypes which are organised into programmable algebraic hierarchies. The authors of Scratchpad claim that such an organisation of algebraic code is useful for building a general purpose computer algebra system. The system provides for modular code. While the system was originally intended for more sophisticated applications, recent developments have placed emphasis upon the user interface to make it more accessible for general use.

The IBM research division has previously made Scratchpad available to a number of universities as part of a joint study contract. At the time of writing, it is being considered as a possible IBM product.

3.2.7 Derive

Derive is the successor to muMath and is a menu-driven system. Many commands and operations can be carried out with just two or three keystrokes. Derive incorporates most of the mathematical facilities of muMath and in addition it has a powerful graphics package which can plot functions in two and three dimensions. It runs on PC-compatible machines under PC-DOS (MS-DOS) and it is provided with a user manual. It is a very easy system to learn and has great potential as a teaching aid.

3.2.8 Mathematica

Mathematica is the successor to SMP and is the product of Wolfram Research Inc., based in Illinois. It was announced amid much publicity in summer 1988 and it is available for a number of popular machines, notably the Apple Macintosh. One of its most noteworthy features is its excellent colour graphics capability, though this depends upon the hardware on which it is run. It requires at least 2Mb of memory in order to operate, and reports indicate that 4Mb or more are needed for serious work with this system.

3.3 Comparison of the eight CAS's

In this section the eight general-purpose CAS's are compared to show their relative merits. Since these systems are subject to frequent upgrades (especially the newer systems), it is important to note that this comparison used the most recent versions available for the hardware used at Liverpool University. The version numbers of the various systems are as follows:

System	Version	Date
REDUCE	3.3	1988
Maple	4.2	1988
MACSYMA	412.6	1988
muMATH	4.12	1983
SMP	1.5.4	1987
SCRATCHPAD	2	1987
Derive	1.61	1990
Mathematica	1.1a	1988

In the tables which follow, some features are either present or absent from a particular system whilst other features are partially present, dependent perhaps upon the hardware. For example, REDUCE has a set of interactive introductory lessons whereas Maple does not. In addition, some systems have the capability to produce graphical output but only when used in conjunction with the appropriate operating system and hardware. Thus the entries in the tables are not always a simple matter of YES/NO. Special cases will be explained in the text below the tables.

The column headed 'SPAD' refers to Scratchpad, and 'Math' refers to Mathematica.

The column headed 'muMATH' may also be taken to describe RIEMANN which is 98% compatible with muMATH.

Table 1: General features

Feature	REDUCE	Maple	MACSYMA	muMATH	SMP	SPAD	Math	Derive
Primer document	Yes	Yes	Yes	Yes[1]	Yes	Yes	Yes[10]	Yes[11]
Reference manual	Yes	Yes	Yes	Yes	Yes	Yes	Yes[10]	Yes[11]
On-line manual	Yes	No	Yes	No	No	Yes	No	No
On-line help system	No	Yes	Yes	No	Yes	Yes	Yes	Yes
Interactive lessons	Yes	No[2]	Yes	Yes	No	Yes	No	No
Access to OS commands[3]	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Access to OS editor[4]	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes
Access to source code	Yes[5]	Yes[6]	No	Yes[7]	No	Yes	Yes[12]	No
Previous command recall[8]	Yes	No	Yes	No	Yes	Yes	Yes	Yes
Previous result recall depth[9]	All	3[13]	All	1	All	All	All	All
Undo previous commands	No	No	No	No	No	No	No	No

Notes

[1] The book *muMATH: A Microcomputer Algebra System* by C Wooff and D Hodgkinson is a good reference for newcomers to muMATH.

[2] Maple has no interactive lessons, but a set of lessons has been developed as part of the Computer Algebra Support project and can be obtained from the University of Liverpool Computer Laboratory.

[3] Only a subset of the host operating system commands may be available. This depends on the operating system.

[4] The editor may not be able to load large files since some CAS's reserve all available memory for their own use, leaving little or none for other tasks. This also depends upon the operating system.

[5] The REDUCE source code is written in RLISP, the LISP meta-language which resembles Algol.

[6] Many of the top-level Maple procedures are written in the Maple language itself - in particular, the routines which constitute the linear algebra, number theory and orthogonal polynomial libraries. The source code for these procedures is provided with the software. Lower-level procedures, such as the basic expression simplifier, are not available in source form.

[7] The muMATH source code is written in the muSIMP language which is similar to LISP.

[8] This refers to the facility whereby previous commands can be re-issued. In REDUCE, for example, the command `INPUT N;` has the effect of re-issuing the N^{th} command of the current session. MACSYMA also allows this by numbering each input line (called a 'c' line) so that the symbol `c1` contains the first command of the current session, `c2` the second and so forth. Instructing MACSYMA to evaluate `c1` is thus equivalent to re-issuing the first command.

[9] All the systems save one or more of the previous results so that they can be recalled for use in subsequent commands. The number of previous results saved is shown in the table. 'All' means that all previous results are available via special variable names. In REDUCE, the variable `WS` contains the result of the previous command whilst `WS(N)` contains the result of the N^{th} command. MACSYMA numbers each output line (called a 'd' line) so that the symbol `d1` contains the results of the first command of the session, `d2` the result of the second and so forth.

[10] The principal documentation for Mathematica is the book *Mathematica: A System for doing Mathematics by Computer* by Stephen Wolfram.

[11] Derive is accompanied by a user guide which serves as a primer and reference manual.

[12] Much of Mathematica is written in the Mathematica language itself. The kernel is written in C, and this is not available in source form.

[13] Maple has a library function named `showtime` which provides a history mechanism with full previous-result-recall.

Table 2: File handling

Feature	REDUCE	Maple	MACSYMA	muMATH	SMP	SPAD	Math	Derive
Read commands and expressions from files	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Write results to file	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Output formatting								
'Pretty-printing'[1]	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes
Machine-readable form	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes
FORTRAN code[2]	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes
C code[2]	Yes	No	Yes	No	Yes	Yes[3]	Yes	No
TEX typesetting code	Yes[4]	Yes	Yes	No	No	Yes	Yes	No
troff (eqn) typesetting code	No	Yes	No	No	No	No	No	No

Notes

[1] This refers to the ability to print superscript indices in a raised position, usually on the line above the expression, to give a pseudo-mathematical notation. Some implementations offer even greater sophistication by use of graphics terminals and can write expressions in true mathematical notation, but this is uncommon.

[2] REDUCE and MACSYMA use the GENTRAN package written by Barbara Gates. GENTRAN can generate FORTRAN and C code and can translate statements in the CAS source language (i.e. REDUCE or MACSYMA statements) to their equivalent in FORTRAN or C, including conditional statements and loop constructs.

[3] On IBM 6150 (RT PC) versions.

[4] A TEX code generation package is available from the REDUCE network library, details of which are given in the Bibliography.

Table 3: Basic algebra

Feature	REDUCE	Maple	MACSYMA	muMATH	SMP	SPAD	Math	Derive
Controlled automatic expansion of expressions[1]	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Controlled automatic expansion of expressions[1]	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Factorisation of polynomials[6]	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes
Complex arithmetic	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Substitution	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Numerical evaluation of expressions	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Arbitrary-precision integer arithmetic	Yes	Yes	Yes	Yes	Yes[2]	Yes	Yes	Yes
Arbitrary-precision real arithmetic	Yes	Yes	Yes	Yes	Yes[2]	Yes	Yes	Yes
Evaluation of limits	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Taylor (Maclaurin) series expansions	No[3]	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Summation of a sequence over an indefinite or infinite range[4]	No	Yes	Yes	Yes	Yes	Yes	No	Yes
Product of a sequence over an indefinite or infinite range[4]	No	Yes	Yes	Yes	Yes	Yes	No	Yes
Asymptotic expansions[5]	No	Yes	Yes	No	Yes	Yes	Yes	No
Re-ordering of expressions with respect to specific variables	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
User-defined operators	Yes	Yes	Yes	No	No	Yes	No	No

Notes

[1] Expansion and simplification are performed by CAS's according to the rules in effect at the time. In REDUCE, MACSYMA and muMATH the rules are controlled at the lowest level by flags. For example, flags exist in all three systems to determine whether a polynomial such as

$$(1 + x)^5$$

is to be expanded to yield

$$1 + 5x + 10x^2 + 10x^3 + 5x^4 + x^5$$

or left in its factorised form. By default, such expansion is carried out by REDUCE but not by muMATH and MACSYMA. Maple never expands such an expression. In all the systems except REDUCE, there are procedures which may be used to apply specific expansion or simplification rules to an expression. In Maple and MACSYMA, expansion of a polynomial is achieved using the `expand` function while in muMATH the equivalent is `expd`.

Similarly, in muMATH, MACSYMA, SMP and Maple there exist procedures to simplify expressions using rules which change only certain type of mathematical function. For example, muMATH has a general-purpose procedure called `trgexpd` which can be used to apply any combination of several trigonometric simplifications to an expression. In the case of muMATH and MACSYMA, some of these procedures operate by temporarily altering the appropriate flags which control the application of these rules. Thus muMATH's `expd` procedure saves the current state of the relevant flags and then alters them to enable expansion of polynomials. It then re-evaluates the argument expression and resets the flags to their original values before returning the expanded result.

[2] SMP uses floating-point arithmetic by default to represent coefficients. The precision of the floating-point numbers is the same as the (machine-dependent) double-precision representation of the local C compiler. This is usually about 16 significant figures.

The other systems use arbitrary-precision integers or ratios of integers to represent coefficients. Hence coefficients are always exact. It is possible to instruct SMP to use integer arithmetic in calculations and this is advisable since the use of real numbers as coefficients may cause problems when comparing or differencing nearly-equal quantities.

[3] A number of Taylor series functions exist for REDUCE. Users will find it very easy to define such a function.

[4] Summation and multiplication of sequences over indefinite or infinite ranges allows the evaluation of a number of important types of expression including geometric sums and the sum of the N^{th} powers of the first M natural numbers. In the latter case, the result is given by the CAS in terms of M which may be indeterminate. By contrast, REDUCE can only calculate such sums and products over a specified finite range.

[5] Some systems perform asymptotic expansions by substituting $x \mapsto 1/x$ and calculating a Taylor expansion in the new variable. This method fails to give a correct result in the presence of many types of singularity.

[6] This table entry indicates ability to factorise multivariate polynomials over the rationals. Some systems can factorise polynomials $Z \ mod \ p$, but only the univariate case is implemented.

Table 4: Equation solving

Feature	REDUCE	Maple	MACSYMA	muMATH	SMP	SPAD	Math	Derive
Univariate polynomials up to degree 4	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Simultaneous linear equations	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Simultaneous non-linear equations	Yes[1]	Yes[2]	Yes	No	Yes	Yes	Yes	No
Equations involving transcendental functions[3]	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Gröbner bases	Yes	Yes	Yes	No	No	No	Yes	No

Notes

[1] Simultaneous non-linear equations may be solved in REDUCE by using the Gröbner basis package supplied with the system.

[2] Maple often returns the solution of simultaneous non-linear equations in terms of another equation (or set of equations) to be solved. For example, the solution to the pair of equations

$$x - y = 1$$
$$x^2 + y^2 = 9$$

is expressed by Maple in terms of the roots of the quadratic

$$X^2 + X - 4 = 0$$

where X is a variable created by the system and distinct from x. The answer is left in this form even though Maple can easily solve the quadratic. This enables the user to see the form of the result more easily, especially when explicit solutions would be complicated expressions involving radicals. The user can often find explicit solutions with a little further manipulation.

[3] Most of the systems can solve simple polynomials in transcendental functions such as

$$\log^2 x + 5 \log x + 6 = 0$$

but will not be able to tackle cases such as

$$\log x + \sin x = 0$$

where no rational solution exists.

Table 5: Calculus

Feature	REDUCE	Maple	MACSYMA	muMATH	SMP	SPAD	Math	Derive
Differentiation								
Differentiation of expressions	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Ability to distinguish between partial and total derivative[1]	No	No	Yes	No	Yes	Yes	Yes	No
Integration								
Polynomials	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Transcendental functions	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Inverse transcendental functions	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Rational functions	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes
Algebraic functions with radicals[2]	Yes[3]	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Definite integrals over an infinite range	No	Yes	Yes	Yes	Yes	No	Yes	Yes
Numerical integration	No	Yes	Yes	No	Yes	No	Yes	Yes
Solution of ordinary differential equations								
1^{st} order	No	Yes	Yes	Yes	Yes	No	No	Yes
2^{nd} order homogeneous	No	Yes	Yes	Yes	Yes	No	No	Yes
2^{nd} order inhomogeneous	No	Yes	Yes	Yes	Yes	No	No	Yes
Perturbation methods	No	No	Yes	No	No	No	No	No
Integral equations	No	No	Yes	No	No	No	No	No

Notes

[1] Most applications of CAS's will involve differentiation of known functions of one or more variables. In such circumstances, the derivatives that are produced are correct. However, in a recent paper by Wester and Steinberg, it was noted that implicit dependencies such as $F(x, y(x))$ may not be treated correctly by most of the systems. In particular, none of the systems except MACSYMA, SMP and Mathematica make the distinction between $\partial F/\partial x$ which is the derivative of F with respect to its first argument, and the total derivative dF/dx which involves both arguments, the second argument via the dependency of y upon x. Wester and Steinberg argue that these systems are limited by their ambiguous notation for derivatives. In order to evaluate the total derivative using Maple and an early version of MACSYMA, they were forced to resort to non-standard techniques which are not to be recommended. They suggest that a concise notation, such as that of SMP, is to be preferred.

[2] Most of the systems can only integrate functions involving radicals of quadratics. By contrast, REDUCE and Scratchpad can handle radicals of more complicated expressions. Scratchpad has the most comprehensive

implementation of algorithms (due to Risch, Trager and Bronstein) for integrating arbitrary mixtures of logarithms, exponentials and algebraic expressions.

[3] REDUCE implements integration of functions involving radicals using the package ALGINT developed by James Davenport. This package is only available for REDUCE version 3.3 and later.

Table 6: Matrix and vector algebra

Feature	REDUCE	Maple	MACSYMA	muMATH	SMP	SPAD	Math	Derive
Matrix algebra								
Use of infix notation[1]	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes
Addition and multiplication	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Inversion	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Determinant	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Rank	No	Yes	Yes	No	Yes	Yes	No	Yes
Eigenvalues and eigenvectors	Yes	Yes	Yes	No[2]	Yes	Yes	Yes	Yes
Definitions of special matrices (e.g. Jacobian)	No	Yes	Yes	No	No	No	No	No
Vector algebra								
Addition	Yes[3]	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Dot and cross products	No[3]	Yes	Yes	Yes	No	Yes	Yes[4]	Yes
Vector calculus								
Cartesian coordinates	No[3]	Yes	Yes	Yes	No	Yes	Yes	Yes
General orthogonal curvilinear coordinates	No[3]	No	Yes	Yes	No	Yes	Yes	Yes

Notes

[1] Most of the systems allow matrix algebra to be performed using scalar algebraic operators. For example, if A and B have been declared to be matrices then their sum can be entered as `A + B` and their product as `A*B` (in MACSYMA, this is `A.B`) assuming that the dimensions of the two matrices are correct in each case. Furthermore, the inverse of a square matrix A can be entered as `1/A` and square matrices can be raised to integer powers (positive or negative) using the exponentiation operator.

Maple does not allow this in general, but it possesses a function called `evalm` which is used to evaluate matrix expressions using infix operator notation. For example, if A and B have been declared as Maple arrays (matrices) then `A * B` would cause an error, but `evalm(A &* B)` would evaluate the matrix product correctly.

[2] muMATH does not possess a routine which calculates the eigenvalues and eigenvectors of a matrix A. However, it can calculate the determinant of the matrix

$$A - x.I$$

where I is the identity matrix of the same dimension as A. The equation-solving can then be called to yield the roots of the characteristic equation which are the eigenvalues.

[3] REDUCE has no built-in capability to perform vector algebra and calculus beyond vector (matrix) addition. However, a comprehensive vector package has been written for use with REDUCE version 3.3 by David Harper. This package introduces a new data type, the 3-vector, and defines vector algebra and calculus in general orthogonal curvilinear coordinate systems.

[4] Mathematica can evaluate vector cross products after the file `crossproducts.m` has been read.

Table 7: Special features

Feature	REDUCE	Maple	MACSYMA	muMATH	SMP	SPAD	Math	Derive
Orthogonal polynomials	No	Yes	Yes	No	Yes	Yes	Yes	No
Bessel functions (of the first kind and integer order)	No	Yes	Yes	No	Yes	No	Yes	No
Number theory functions	No	Yes	Yes	No	Yes	Yes	Yes	No
Laplace transforms	No[3]	Yes	Yes	No	Yes	Yes	Yes	No
Inverse Laplace transforms	No[3]	Yes	Yes	No	Yes	Yes	No	No
Fourier transforms	No	No	Yes	No	No	No	Yes	No
Graphics[1]								
2-dimensional plots (curve drawing)	No	Yes	Yes	No	Yes	Yes	Yes	Yes
3-dimensional plots (surface drawing)	No	No	Yes	No	Yes	Yes	Yes	Yes
Parametric plotting	No	Yes	Yes	No	Yes	Yes	Yes	Yes
Colour graphics	No	Yes	No	No	No	Yes	Yes	Yes
Surface shading	No	No	No	No	No	Yes	Yes	No
Postscript output	No	Yes	Yes	No	No	Yes	Yes	No

Notes

[1] The ability to display functions and surfaces using graphics depends upon the hardware which is used to run the CAS. Some implementations may be unable to produce graphical output if appropriate hardware is not installed.

[2] Derive can produce graphical output using a variety of popular IBM/PC graphics cards including EGA, CGA, VGA, MCGA and Hercules. In addition, it can produce low-resolution character-based graphics in the absence of a suitable graphics card.

[3] REDUCE can calculate Laplace transforms and inverse transforms using a package written by C Kazasov, M Spiridonova and V Tomov at Sofia University, Bulgaria. This package is not yet part of the official version of REDUCE but it may be obtained from the REDUCE network library, details of which are given in the Bibliography.

4 Availability of CAS's

This section contains information on the availability of the eight general purpose CAS's including addresses of suppliers and (where available) prices. This information is subject to change, especially in the case of systems which are under continuous development. In particular, much work is being carried out on new versions of REDUCE, Maple and MACSYMA and the developers of these systems are very active. **Price information should only be regarded as a guide**, although prices have been quite steady.

MACSYMA

Availability:

- DEC VAX 7xx, 82xx, 83xx, 84xx series (under VMS and UNIX)
- MicroVAX II
- Symbolics 3600 series workstations
- Sun workstations (under UNIX)
- Apollo workstations
- 386-based PC-compatibles

Price:

For a single CPU, MACSYMA costs £1250 for a SUN-3 workstation and £1050 for a 386-based PC. Prices for VAX systems depend upon the machine: for example, £1600 for a VAX 11/780 and £3000 for a VAX 8700. These prices apply only to educational establishments.

Suppliers:

Symbolics Limited
St John's Court
Easton Street
High Wycombe
Bucks HP11 1JX

Telephone: (0494) 443711

In the United States:

Computer Aided Mathematics Group
Symbolics Inc.
8 New England Executive Park East
Burlington
MA 01803

Telephone: (617) 221-1250
FAX: (617) 221-1099

REDUCE

Availability:

Machine	Operating System	Notes
Acorn Archimedes A440		Needs at least 4Mb memory
Apollo	Berkeley UNIX	
Atari 1040ST or Mega	TOS	Needs at least 1Mb memory
CDC Cyber 170	NOS or NOS/BE	
CDC Cyber 180	NOS/VE 1.3 or 1.4	PSL version
Common LISP		Generic Common LISP version
Cray 1, 2, X-MP, Y-MP	COS or UNICOS	PSL version
Data General Eclipse MV	AOS/VS	PSL version
DEC VAX	VMS	PSL version
DEC VAX	UNIX 4.2 BSD or ULTRIX	PSL version
DEC VAX	UNIX 4.2 BSD	Franz LISP version
DECStation 2100, 3100	UNIX	MIPS PSL version
Hewlett-Packard 9000/300	HP-UX	300 series only
HLH Orion	UNIX 4.2 BSD	Cambridge LISP version
IBM 370	VM/CMS	PSL version
IBM 370	MVS or TSO or UTS	Standard LISP version
IBM/PC compatibles	MS-DOS	Two versions available
ICL 2900 series	VME	Cambridge LISP version
Siemens X20,MX2,MX300 etc	SINIX	
Silicon Graphics IRIS series	IRIX	MIPS PSL version
SONY NEWS	UNIX 4.2 BSD	Tuneup LISP version
Sord M68MX	CP/M-68K	Cambridge LISP version
SUN 3	SunOS	PSL version
SUN 3	SunOS	KCL version
SUN 3	SunOS	Part of Mathscribe
SUN 4	SunOS	PSL version
SUN 386i	UNIX	Generic 80386 version
Tektronix 4300	UNIX	Part of Mathscribe
Toshiba AS3000	OS/AS	KCL version
StaffLisp/68000		Generic 68000 version
StaffLisp/86		Generic 80x86 version

Price and supplier:

REDUCE is supplied by various organisations under license from the RAND Corporation. The names and addresses of the suppliers are given in the *REDUCE Information Form* published by RAND, a copy of which is reproduced as Appendix B of this document. Please contact the suppliers for information on prices and licensing options.

Maple

Availability:

Machine	Operating System	Version	Price (CDN$)
386-based systems	PC or MS DOS	4.3.1	835
386-based systems	386/ix or AT&T UNIX or SCO UNIX	4.3	1075
Amiga 1000	Amiga DOS	4.2	475
Apollo 3xxx, 4xxx	Unix	4.2	1075
Apollo DN10000	Unix	4.2	1495
Atari ST	TOS	4.2	475
AT&T 7300, 3B1	Unix V.2	4.1	595
AT&T 3B2/500	Unix V.2	4.2	1075
Convex C series	Convex Unix	4.2	1495
Cray 2	Unicos	4.0	1255
DECStation 2100, 3100	Ultrix	4.3	1075
DECStation 5400, 5800	Ultrix	4.3	1495
DEC Microvax	Ultrix 32m	4.3	835
DEC VAX	4.2 or 4.3 BSD Unix	4.3	1075
DEC VAX	VMS	4.3	1010[1]
Gould NPI	Unix	4.1	595
Gould DN 9082	Unix	4.2	595
Hewlett-Packard 9000/300	HP-UX	4.3	1075
IBM 386-based systems	PC / MS DOS	4.3.1	835
IBM PS/2, Model 70,80	AIX	4.3	1075
IBM RISC System/6000	AIX	4.3	1285
IBM RT	AIX 1.1	4.2	1075
IBM S/370 family	VM/Sp CMS	4.2	1195[1]
Macintosh Plus,SE,II	Finder	4.2	475
Masscomp 5xxx	Unix	4.0	595
MIPS RS2030	Unix (BSD or Sys V)	4.3	1075
MIPS RC3240, RC3260	Unix (BSD or Sys V)	4.3	1285
MIPS M/2000, RS6280	Unix (BSD or Sys V)	4.3	1495
NCR Tower	Unix	4.2	595
PCS Cadmus	Munix	4.1	595
Pyramid 9800	Unix	4.1	595
Sequent Balance	Dynix	4.1	835
Sequent Symmetry	Dynix	4.3	1495
Silicon Graphics IRIS	IRIX	4.3	1075
Sun 3	Sun Unix	4.3	1075
Sun 4, SPARCstation	Sun Unix	4.3	1075
Sun 386i	Sun Unix	4.2	595
Unisys Tahoe	Unix	4.2	835
Library source tape		4.3	2400

Price:

[1] Annual license fee.

(Continued on next page)

Price:

The license fees quoted in the table allow the use of Maple on a single CPU for an indefinite period, except for VM/CMS and VAX/VMS versions where the fee is an annual license fee. Updates to the Maple system are available free of charge during the first year. Site licenses are available for most versions of Maple, and these provide substantial discounts.

Supplier:

Waterloo Maple Software Inc
160 Columbia Street, W.
Waterloo, Ontario
Canada N2L 3L3

Phone: (519) 747-2373 FAX: (519) 747-5284

Email: wmsi@daisy.uwaterloo.ca *or* wmsi@daisy.waterloo.edu

muMATH and RIEMANN

Availability:

- Apple II (muMATH)
- Atari ST (RIEMANN)

Price:

Please contact the suppliers listed below for price information.

Suppliers:

muMATH for Apple II:

Dr Edwin Dickey
Educational Technology Center
College of Education
University of South Carolina
Columbia
SC 29208
United States of America

RIEMANN for Atari ST:

Begemann & Niemeyer Softwareentwicklung
Schwarzenbrinker Str. 91
4930 DETMOLD
West Germany

Notes:

[1] muMATH has been superseded by Derive. Support for muMATH has been discontinued by Soft Warehouse Inc and the IBM/PC version is no longer officially available, though some suppliers (see entry for Derive) may have copies in stock.

[2] RIEMANN is 98% compatible with muMATH and existing muMATH users will have no difficulty using it. RIEMANN has several enhancements including graphics, improved floating-point capability, sub-expression selection using a mouse, and menus to select options for functions such as `trgexpd`. The RIEMANN user guide is currently only available in German, although the authors of RIEMANN are planning to make an English translation available in late 1989. A version of RIEMANN for the Apple Macintosh is also planned for Spring 1990.

SMP

Availability:

- VAX VMS
- VAX UNIX
- MicroVAX
- Sun workstations
- Apollo workstations

Price:

Approximately US $300 (for VAX VMS version).

Supplier:

Innovus International Inc
250 W. First Street
Suite 240
Claremont
CA 91711
Telephone: (714) 624-7907

Note

Current SMP licenses issued to UK Universities include a clause restricting its use to staff, students and researchers *at the licensee institution only*, excluding its use from other sites via JANET. In addition, the license is subject to a US Department of Trade restriction which forbids the use of SMP by nationals of a number of Eastern Bloc and East Asian countries and also forbids dissemination of results obtained using SMP to those countries.

SCRATCHPAD

Availability

- IBM VM/CMS
- IBM 6150 (RT PC)
- IBM PS/2

Academic users wishing to undertake a joint study contract on SCRATCHPAD should write a letter describing the intended applications to:

Richard D. Jenks,
Computer Algebra Group,
IBM Research Division,
T.J. Watson Research Center,
P.O. Box 218,
Yorktown Heights, NY 10598.
U.S.A.

More information about SCRATCHPAD can be obtained by writing to:

Computer Algebra Group,
c/o Sandra Wityak

at the above address.

Derive

Availability:

- All IBM/PC compatible machines running MS-DOS

Price:

Derive is approximately £125 per copy when obtained from most suppliers. Some suppliers offer a discount on orders for a large number of copies. There are also special license arrangements for the use of Derive on LANs.

Supplier:

In the UK:

System Science
3-5 Cynthia Street
London N1 9JF

Telephone: (01) 833 1022
FAX: (01) 837 6411

GreyMatter Ltd
2 Prigg Meadow
Ashburton
Devon TQ13 7DF

Telephone: (0364) 53499
FAX: (0364) 53071

Chartwell-Bratt Ltd.
Old Orchard
Bickley Road
Bromley
Kent BR1 2NE

Telephone: (01) 467 1956
FAX: (01) 467 1754

Mathematica

Availability:

Machine	Operating System	Price (US$)
Apollo DN2500 to 4500	SR10	2000
Apple Macintosh		620 to 995
Cray X-MP,Y-MP,2	UNICOS	*Call WRI*
DEC VAX	VMS and ULTRIX	2000 to 31250
DECstation 2100,3100	ULTRIX	2250
DECsystem 3100,5400,5800	ULTRIX	3875 to 6500
Hewlett-Packard 9000/3xx	HP-UX	2000
Hewlett-Packard 9000/8xx	HP-UX	4625 to 8875
IBM AIX/RT	AIX	2000
MIPS RC2030,RS2030,RC3240, M/120,M/1000,M/2000	UNIX	2250 to 6500
386-based MS-DOS systems	MS-DOS	870 to 1620
Silicon Graphics IRIS	IRIX	2250 to 5875
Sony NEWS		2000 to 2750
SUN 3	SunOS	1745
SUN 4	SunOS	2000
SUN 386i	SunOS	2000

Price:

The prices given in the table above apply to educational institutions only. Commercial prices are generally about 50% higher. For certain machines, such as DEC VAX, the price depends upon the model of the hardware. Full price information is given in the Mathematica price list published by Wolfram Research. A copy of the most recent price list is reproduced as Appendix C of this document.

Suppliers: (in the U.K.)

Colin Grace Associates
Clifton Mews
62 High Street
Saffron Walden
Essex
Telephone: (0799) 22532

In Touch Software
2 Clive Chambers
Clive Street
Caerphilly
Mid Glamorgan CF8 1GE
Telephone: (0222) 887521

5 Examples in the use of the eight CAS's

The following pages show how each of the six major computer algebra systems are used to solve a number of mathematical problems. Seven problems have been chosen to illustrate a range of techniques and to demonstrate the capabilities of the systems.

Example problems

1 Expand $(1+x)^6$, differentiate the expanded polynomial and then factorise the derivative.

2 Evaluate the function obtained by integrating $1/x$ three times.

3 Solve the cubic equation $x^3 - 4x^2 - 11x + 30$. This equation has integer roots, namely 2, 5 and -3.

4 Evaluate the inverse and the determinant of the matrix

$$\begin{pmatrix} 1 & p \\ q & 3 \end{pmatrix}$$

5 Solve the inhomogeneous second-order ordinary differential equation

$$\frac{d^2y}{dx^2} + y = \sin 2x$$

6 Evaluate the factorial of 30.

7 Evaluate π to 30 significant figures.

For each computer algebra system, the sequence of commands required to solve the problems is given, followed by the output which resulted from executing these commands.

5.1 REDUCE

Program

```
% Example 1
Y := (1+X)**6;
DY := DF(Y,X);
W := FACTORIZE(DY);

% Example 2
Y := 1/X;
INT(WS,X);
INT(WS,X);
INT(WS,X);

% Example 3
Y := X**3 - 4*X**2 - 11*X + 30;
SOLVE(Y,X);

% Example 4
MATRIX A;
A := MAT( (1,P), (Q,3) );
1/A;
DET A;

% Example 5
%
% REDUCE cannot solve differential equations

% Example 6
FOR J:=1:30 PRODUCT J;

% Example 7
ON BIGFLOAT;
ON NUMVAL;
PRECISION 30;
4*ATAN(1);

QUIT;
```

Results

```
REDUCE 3.3, 15-Jul-87 ...
% Example 1
Y := (1+X)**6;

      6      5       4       3       2
Y := X  + 6*X  + 15*X  + 20*X  + 15*X  + 6*X + 1

DY := DF(Y,X);

          5      4       3       2
DY := 6*(X  + 5*X  + 10*X  + 10*X  + 5*X + 1)

W := FACTORIZE(DY);

W := {6,

      X + 1,

      X + 1,

      X + 1,

      X + 1,

      X + 1}


% Example 2
Y := 1/X;

      1
Y := ---
      X

INT(WS,X);

LOG(X)

INT(WS,X);

X*(LOG(X) - 1)

INT(WS,X);

  2
 X *(2*LOG(X) - 3)
-------------------
         4


% Example 3
```

```
Y := X**3 - 4*X**2 - 11*X + 30;

      3      2
Y := X  - 4*X  - 11*X + 30

SOLVE(Y,X);

{X=5,X=2,X=-3}


% Example 4
MATRIX A;

A := MAT( (1,P), (Q,3) );

A(1,1) := 1

A(1,2) := P

A(2,1) := Q

A(2,2) := 3

1/A;

                   3
MAT(1,1) :=  - ---------
                P*Q - 3

                P
MAT(1,2) := ---------
             P*Q - 3

                Q
MAT(2,1) := ---------
             P*Q - 3

                   1
MAT(2,2) :=  - ---------
                P*Q - 3

DET A;

 - (P*Q - 3)


% Example 5
%
% REDUCE cannot solve differential equations

% Example 6
FOR J:=1:30 PRODUCT J;
```

```
    265252859812191058636308480000000


    % Example 7
    ON BIGFLOAT;

    ON NUMVAL;

    PRECISION 30;

    30

    4*ATAN(1);

    3.141 59265 35897 93238 46264 33832 8


    QUIT;
*** END OF RUN
```

5.2 Maple

Program

```
writeto(`example.out`);

# Example 1
y := (1+x)^6;
y := expand(y);
dy := diff(y,x);
factor(dy);

# Example 2
y := 1/x;
int(",x);
int(",x);
int(",x);

# Example 3
y := x^3 - 4*x^2 -11*x + 30;
solve(y,x);

# Example 4
with(linalg);
a := array(1..2, 1..2, [ [1,p], [q,3] ]);
inverse(a);
det(a);

# Example 5
p := dsolve(diff(f(x),x,x)+f(x)=sin(2*x),f(x));
readlib(`trig/reduce`);
`trig/reduce`(p);

# Example 6
30!;

# Example 7
Digits := 30;
evalf(Pi);

quit;
```

Results

```
# Example 1
# y := (1+x)^6
                                                    6
                                       y := (1 + x)

# y := expand(y)

                                  2        3        4       5    6
                y := 1 + 6 x + 15 x  + 20 x  + 15 x  + 6 x  + x

# dy := diff(y,x)

                                     2        3        4      5
                 dy := 6 + 30 x + 60 x  + 60 x  + 30 x  + 6 x

# factor(dy)

                                                   5
                                         6 (1 + x)


# Example 2
# y := 1/x
                                          y := 1/x


# int(",x)
                                           ln(x)

# int(",x)

                                        x ln(x) - x

# int(",x)

                                        2               2
                                   1/2 x  ln(x) - 3/4 x

# Example 3
# y := x^3 - 4*x^2 =11*x +30
                                       3      2
                                 y := x  - 4 x  - 11 x + 30


# solve(y,x)
                                          -3, 5, 2

# Example 4
# with(linalg)
Warning: new definition for   trace
```

```
# a := array(1..2, 1..2, [ [1,p], [q,3] ])

a := array ( 1 .. 2, 1 .. 2,

                                    [1, p]

                                    [q, 3]

)
# inverse(a)

array ( 1 .. 2, 1 .. 2,

                                 1            p
                           [3 ---------, - ---------]
                               3 - p q      3 - p q

                                  q          1
                            [- ---------, ---------]
                                3 - p q    3 - p q

)

# det(a)

                                   3 - p q

# Example 5
# p := dsolve(diff(f(x),x,x)+f(x)=sin(2*x), f(x))

             p := f(x) = - 2/3 sin(x) cos(x) + C sin(x) + C1 cos(x)

# readlib(`trig/reduce`)
# `trig/reduce`(p)

                  f(x) = - 1/3 sin(2 x) + C sin(x) + C1 cos(x)

# Example 6
# 30!
                       265252859812191058636308480000000

# Example 7
# Digits := 30
                                  Digits := 30

# evalf(Pi)

                        3.14159265358979323846264338328
```

5.3 MACSYMA

Program

```
writefile("example.out");
y:(1+x)^6;
y:expand(y);
dy:diff(y,x);
factor(dy);

y:1/x;
integrate(%,x);
integrate(%,x);
integrate(%,x);

y:x^3 - 4*x^2 - 11*x + 30;
solve(y,x);

a:matrix( [1,p], [q,3] );
invert(a);
determinant(a);

depends(f,x);
eq:diff(f,x,2)+f=sin(2*x);
ode(eq,f,x);

30!;

fpprec:30;
bfloat(%pi);

closefile();
quit();
```

Results

```
(c2) y:(1+x)^6;

                                              6
(d2)                                   (x + 1)

(c3) y:expand(y);

                   6      5       4       3       2
(d3)              x  + 6 x  + 15 x  + 20 x  + 15 x  + 6 x + 1

(c4) dy:diff(y,x);

                      5       4       3       2
(d4)                6 x  + 30 x  + 60 x  + 60 x  + 30 x + 6

(c5) factor(dy);

                                               5
(d5)                                  6 (x + 1)

(c6) y:1/x;

                                            1
(d6)                                        -
                                            x

(c7) integrate(%,x);

(d7)                                   log(x)

(c8) integrate(%,x);

(d8)                                x log(x) - x

(c9) integrate(%,x);



                                       2             2
                                      x  log(x)   3 x
(d9)                                  --------- - ----
                                          2        4

(c10) y:x^3 - 4*x^2 - 11*x + 30;



                                     3      2
(d10)                               x  - 4 x  - 11 x + 30

(c11) solve(y,x);
```

```
(d11)                        [x = - 3, x = 5, x = 2]

(c12) a:matrix( [1,p], [q,3] );


                                    [ 1  p ]
(d12)                               [      ]
                                    [ q  3 ]

(c13) invert(a);


                             [     3          p    ]
                             [  -------   - ------- ]
                             [  3 - p q     3 - p q ]
(d13)                        [                      ]
                             [      q        1      ]
                             [ - -------   -------  ]
                             [   3 - p q   3 - p q  ]

(c14) determinant(a);

(d14)                                3 - p q

(c15) depends(f,x);

(d15)                                [f(x)]

(c16) eq:diff(f,x,2)+f=sin(2*x);


                               2
                              d f
(d16)                         --- + f = sin(2 x)
                                2
                              dx

(c17) ode(eq,f,x);


                           sin(2 x)
(d17)              f = - -------- + %k1 sin(x) + %k2 cos(x)
                              3

(c18) 30!;
```

```
(d18)                   265252859812191058636308480000000

(c19) fpprec:30;

(d19)                                  30

(c20) bfloat(%pi);

(d20)                   3.14159265358979323846264338328b0
```

5.4 muMATH

Program

```
wrs('example,'out,'b);
% Example 1                                             %
y:(1+x)^6;
y:expd(y);
dy:dif(y,x);
% muMATH can't factorise polynomials                    %

% Example 2                                             %
y:1/x;
int(@,x);
int(@,x);
int(@,x);

% Example 3                                             %
y:x^3 - 4 x^2 - 11 x + 30;
solve(y,x);

% Example 4                                             %
a:{ [1,p], [q,3] };
a^(-1);
det(a);

% Example 5                                             %
depends(f(x));
solve(dif(f,x,x)+f-sin(2 x),f);

% Example 6                                             %
30!;

% Example 7                                             %
% muMATH cannot evaluate pi in floating-point form      %
```

Results

```
% Example 1                                              %
?@: (1 + X)^6
?@: 1 + 6 X + 15 X^2 + 20 X^3 + 15 X^4 + 6 X^5 + X^6
?@: 6 + 30 X + 60 X^2 + 60 X^3 + 30 X^4 + 6 X^5


% Example 2                                              %
?@: 1/X
?@: LN X
?@: -X + X LN X
?@: X^2 LN X/2 - 3/4 X^2


% Example 3                                              %
?@: 30 - 11 X - 4 X^2 + X^3
?@: {X == 4/3 + 49/3/(-143 + 180 3^(1/2) #I)^(1/3)
            + (-143 + 180 3^(1/2) #I)^(1/3)/3,
     X == 4/3 + 49/6 3^(1/2) #I/(-143 + 180 3^(1/2) #I)^(1/3) -
            3^(1/2) #I*(-143 + 180 3^(1/2) #I)^(1/3)/6 -
            49/6/(-143 + 180 3^(1/2) #I)^(1/3) -
            (-143 + 180 3^(1/2) #I)^(1/3)/6,
     X == 4/3 - 49/6 3^(1/2) #I/(-143 + 180 3^(1/2) #I)^(1/3)
            + 3^(1/2) #I*(- 143 + 180 3^(1/2) #I)^(1/3)/6 -
            49/6/(-143 + 180 3^(1/2) #I)^(1/3) -
            (-143 + 180 3^(1/2) #I)^(1/3)/6}


% Example 4                                              %
?@: {[1, P],
     [Q, 3]}
?@: {[1 + P Q/(3 - P Q), -P/(3 - P Q)],
     [-Q/(3 - P Q), 1/(3 - P Q)]}
?@: 3 - P Q


% Example 5                                              %
?@: TRUE
?@: {F == 2 ARB (3) SIN X INT (1/(1 - COS (2 X)), X)
            + ARB (4) SIN X - SIN X LN COS (X/2)/2
            + SIN X LN SIN (X/2)/2 -
            SIN X INT (SIN (3 X)/(1/2 - COS (2 X)/2),X)/6}


% Example 6                                              %
?@: 265252859812191058636308480000000


% Example 7                                              %
% muMATH cannot evaluate pi in floating-point form       %
```

5.5 Scratchpad

Program

```
/* Example 1 */
(1+x)**6
pderiv(%,x)
factor %

/* Example 2 */
y := 1/x
integrate(y,x)
integrate(%,x)
integrate(%,x)
integrate(%,x)

/* Example 3 */
solve(x**3-4*x**2-11*x+30,x)

/* Example 4 */
a := Matrix([ [1,p], [q,3] ])
determinant(a)
inverse(a)

/* Example 5 */
/* Currently, Scratchpad can only solve differential equations
   using power series */

/* Example 6 */
*/[1..30]

/* Example 7 */
precision := 30
pi()
```

Results

```
   Scratchpad

    /* Example 1 */

(1+x)**6

         6     5      4      3      2
   (1)  x  + 6x  + 15x  + 20x  + 15x  + 6x + 1

pderiv(%,x)

          5      4      3      2
   (2)  6x  + 30x  + 60x  + 60x  + 30x + 6

factor %

                5
   (3)  6(x + 1)

    /* Example 2 */

y:=1/x




        1
   (4)  -
        x

integrate(%,x)

   (5)  log(x)

integrate(%,x)

   (6)   x log(x) - x

integrate(%,x)



           2            2
         2x  log(x) - 3x
   (7)   ----------------
                4

    /* Example 3 */

solve(x**3-4*x**2-11*x+30,x)
```

```
   (8)  [- 3,2,5]


    /* Example 4 */

a:=Matrix([[1,p],[q,3]])



       [ 1  p ]
   (9) [      ]
       [ q  3 ]

Type: M P I

determinant(a)

   (10)  - p*q + 3

inverse(a)



      [      3          p      ]
      [  - -------    -------   ]
      [    p q - 3    p q - 3   ]
      [                         ]
      [      q            1     ]
      [    -------    - ------- ]
      [    p q - 3      p q - 3 ]

    /* Example 5 */

    /* Currently, Scratchpad can only solve differential equations
       using power series */

    /* Example 6 */

factorial(30)

  (12)  265252859812191058636308480000000


    /* Example 7 */

digits 30

  (13)  20

pi()

  (14)  3.1415926535 8979323846 264338328
```

5.6 SMP

Program

```
/* Example 1 */
Ex((1+x)**6)
D(%,x)
Fac(%)

/* Example 2 */
Int(1/x,x)
Int(%,x)
Int(%,x)

/* Example 3 */
Sol(x**3-4*x**2-11*x+30=0,x)

/* Example 4 */
Minv({{1,p},{q,3}})
Det({{1,p},{q,3}})

/* Example 5 */
/* SMP cannot solve differential equations */

/* Example 6 */
30!

/*Example 7 */
N(Pi,30)
```

Results

```
/*  Example 1 */
#I(1)::
Ex((1+x)**6)

                      2       3       4      5    6
#O(1):   1 + 6x + 15 x  + 20 x  + 15 x  + 6 x  + x

#I(2)::
D(%,x)

                       2       3       4      5
#O(2):   6 + 30x + 60 x  + 60 x  + 30 x  + 6 x

#I(3)::
Fac(%)

                  5
#O(3):   6 (1 + x)

/* Example 2 */
#I(4)::
Int(1/x,x)

#O(4):   Log(x)

#I(5)::
Int(%,x)

#O(5):   -x + x * Log(x)

#I(6)::
Int(%,x)

          2               2
         x  * Log(x)   3 x
#O(6):   ----------- - ----
              2         4

/* Example 3 */
#I(7)::
Sol(x**3-4*x**2-11*x+30=0,x)

#O(7):   (<x -> -3,x -> 2,x -> 5>)

/* Example 4 */
#I(8)::
Minv({{1,p},{q,3}})

                 3        -p            -q         1
#O(8):   (<(<---------,--------->),(<---------,--------->)>)
             3 - p * q 3 - p * q     3 - p * q 3 - p * q
```

```
#I(9)::
Det({{1,p},{q,3}})

#O(9):   3 - p * q

/* Example 5 */
/* SMP cannot solve differential equations */

/* Example 6 */
#I(10)::
30!

#O(10):*  (265252859812191058636308480000000)

/* Example 7 */
#I(11)::
N(Pi,30)

#O(11):*  (3.14159265358979323846264338328)
```

5.7 Derive

Program

Note: since Derive is a menu-driven system, most operations can be carried out using single keystrokes. In the example program which follows, we denote menu options using square brackets. For example, `[A]uthor` means that the menu option **A** (Author) is chosen. Note also that Derive does not allow comments, so the '/* ... */' notation should not be entered. It is used in this document to explain the examples.

```
/* Example 1 */
[A]uthor
(1+x)^6
[E]xpand
[C]alculus [D]ifferentiate
[F]actor

/* Example 2 */
[A]uthor  1/x
[C]alculus  [I]ntegrate
[C]alculus  [I]ntegrate
[C]alculus  [I]ntegrate

/* Example 3 */
[A]uthor  x^3-4*x^2-11*x+30
so[L]ve

/* Example 4 */
[A]uthor
A:=[[1,p],[q,3]]
[E]xpand
A^(-1)
[F]actor
[A]uthor
DET(A)

/* Example 5 */
/* Derive cannot solve differential equations */

/* Example 6 */
[A]uthor
30!

/* Example 7 */
[O]ption  [N]otation  [D]ecimal  30
[O]ption  [P]recision  [A]pproximate
[A]uthor  <ALT+P>
/* this is the key sequence ALT+P, a synonym for pi */
[E]xpand
```

Results

```
/* Example 1 */

               6
1:    (1 + x)


      6       5        4        3        2
2:   x  + 6 x  + 15 x  + 20 x  + 15 x  + 6 x + 1

        5        4        3       2
3:   6 x  + 30 x  + 60 x  +60 x  + 30 x + 6

                 5
4:   6 (1 + x)

/* Example 2 */

      1
5:   ---
      x


6:   LN (x)


7:   x LN (x) - x



       2                2
      x  LN (x)      3 x
8:   ----------- - ------
          2          4


/* Example 3 */

      3      2
9:   x  - 4 x  - 11 x + 30


10:  x = -3


11:  x = 2


12:  x = 5


/* Example 4 */
```

```
          [ 1  p ]
13:  A :=  [      ]
          [ q  3 ]


     [       3             3        1 ]
     [  - ---------   ------------ + - ]
     [     p q - 3    q (p q - 3)    q ]
14:  [                                 ]
     [       q             p           ]
     [    ---------    - ---------     ]
     [     p q - 3        p q - 3      ]


     [       3             p           ]
     [  - ---------      ---------     ]
     [     p q - 3        p q - 3      ]
15:  [                                 ]
     [       q             1           ]
     [    ---------    - ---------     ]
     [     p q - 3        p q - 3      ]

16:  3 - p q

/* Example 5 */
/* Derive cannot solve differential equations */


/* Example 6 */

17:  265252859812191058636308480000000


/* Example 7 */

18:  3.14159292035398230088495575221
```

5.8 Mathematica

Program

```
/* Example 1 */
Expand[(1+x)^6]
D[%,x]
Factor[%]

/* Example 2 */
Integrate[1/x,x]
Integrate[%,x]
Integrate[%,x]

/* Example 3 */
Solve[x^3-4*x^2-11*x+30==0,x]

/* Example 4 */
A:={{1,p},{q,3}}
Inverse[A]
Det[A]

/* Example 5 */
/* Mathematica cannot solve differential equations */

/* Example 6 */
30!

/* Example 7 */
N[Pi,30]
```

Results

```
/* Example 1 */
In[1]:=Expand[(1+x)^6]

                         2        3        4       5    6
Out[1]=  1 + 6 x + 15 x  + 20 x  + 15 x  + 6 x  + x

In[2]:=D[%,x]

                          2        3        4    5
Out[2]=  6 + 30 x + 60 x  + 60 x  + 30 x  + x

In[3]:=Factor[%]

                   5
Out[3]=  6 (1 + x)

/* Example 2 */
In[4]:=Integrate[1/x,x]

Out[4]=Log[x]

In[5]:=Integrate[%,x]

Out[5]=- x + x Log[x]

In[6]:=Integrate[%,x]



               2    2
         - 3 x     x  Log[x]
Out[6]=  ------ + ---------
           4          2

/* Example 3 */
In[7]:=Solve[x^3-4*x^2-11*x+30==0,x]

Out[7]={{x -> -3}, {x -> 5}, {x -> 2}}

/* Example 4 */
In[8]:=A:={{1,p},{q,3}}

In[9]:=Inverse[A]



              3          -p            -q           q
Out[9]=  {{---------, ---------}, {---------, ---------}}
            3 - p q    3 - p q      3 - p q    3 - p q

In[10]:=Det[A]
```

```
Out[10]= 3 - p q

/* Example 5 */
/* Mathematica cannot solve differential equations */

/* Example 6 */
In[11]:=30!

Out[11]=265252859812191058636308480000000

/* Example 7 */
In[12]:=N[Pi,30]
Out[12]= 3.14159265358979323846264338328
```

6 Bibliography

Some of the information presented here is based on a similar bibliography generated by Jane Bryan-Jones and Francis Wright, whose work we gratefully acknowledge.

6.1 Books on computer algebra

Buchberger, Collins, Loos and Albrecht, *Computer Algebra: Symbolic and Algebraic Computing*, Springer (1983)

Pavelle, R. (editor), *Applications of Computer Algebra*, Kluwer Academic Publishers (1985)

Wooff, C.D., Hodgkinson, D.E., *muMATH: A Microcomputer Algebra System*, Academic Press (1987)

Davenport, J.H., Siret, Y., Tournier, E., *Symbolic Algebra, Systems and Algorithms for Algebraic Computation*, Academic Press (1988)

Rayna, G., *REDUCE – Software for Algebraic Computation*, Springer (1987)

Wolfram, S., *Mathematica – a system for doing mathematics by computer*, Addison-Wesley (1988)

Maeder, R., *Programming for Mathematica*, Addison-Wesley (1989)

6.2 Books on Applications

Rand, R.H., *Computer Algebra in Applied Mathematics: an Introduction to MACSYMA*, Pitman Research Notes in Mathematics **94** (1984)

Rand, R.H. and Armbruster, *Perturbation Methods, Bifurcation Theory and Computer Algebra*, Springer Applied Mathematical Science Series **65** (1987)

Crandall, R., *Mathematica for the Sciences*, Addison-Wesley (1990)

6.3 Books on Mathematics and Algorithms

van der Waerden, B.L., *Modern Algebra* (2 volumes), Frederick Ungar (1953)

Knuth, D.E., *The Art of Computer Programming, Vol. 2: Seminumerical Algorithms*, Addison-Wesley (1969)

Aho, Hopcroft, Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley (1974)

Lipson, J.D., *Algebra and Algebraic Computing*, Addison-Wesley (1981)

Sims, C.S., *Abstract Algebra: A Computational Approach*, Wiley (1984)

6.4 Review Articles

Davis, M.S., *Review of Non-Numerical Uses of Computers*, in Tapley, B.D, Szebehely, V. (editors), **Recent Advances in Dynamical Astronomy**, 351-391, Reidel (1973)

Wester, M., Steinberg, S., *A Survey of Symbolic Differentiation Implementations* in Golden, E.V. (editor) **Proceedings of the 1984 MACSYMA Users' Conference**, General Electric, Schenectady, New York (1984)

Analytic calculations on digital computers for applications in physics and mathematics, **Sov. Phys. Usp., 23**, 59-77 (1980)

Yun, Stoutemyer, *Symbolic Mathematical Computation*, in Belzer, Holzmann, Kent (editors), **Encyclopaedia of Computer Science and Technology, 15**, 235-310, Mercel Dekker (1980)

Algebra made mechanical, **Nature, 290**, 198-200 (March 1981)

Steen, L.A., *Computer Calculus*, **Science News, 119** (1981)

Pavelle, R., Rothstein, Fitch, J., *Computer algebra*, **Scientific American, 245**, 102-113 (December 1981)

Fraser, C., *Algebraic manipulation by computer – a sign of things to come ?*, **IMA Bulletin, 21**, 167-168 (1985)

Sloane, N.J.A., *My friend MACSYMA*, **Notices of the American Mathematical Society**, 40-43 (1985)

MacCallum, M.A.H., *Computer algebra – tomorrow's calculator ?*, **New Scientist, 112**, 52-55 (23 October 1986)

Hosack, J.M., *A Guide to Computer Algebra Systems*, **The College Mathematics Journal, 17,5**, 434-441 (1986)

Wooff, C.D., Hodgkinson, D.E., *Computer algebra systems in physics teaching*, **European Journal of Physics, 9**, 145-148 (1988)

MacCallum, M.A.H., *Pocket Calculus*, **Physics World, 2,6**, 27-29 (June 1989)

Harper, D., *Maths on the Menu*, **Physics World, 3,2**, 43 (February 1990)

6.5 Articles Primarily on Teaching Use

Whole double issue of **SIGSAM Bulletin**, 18,4 and 19,1 issues 72,73 (1984,1985)

Lane, Ollongren, Stoutemyer, *Computer-based symbolic mathematics for discovery*, in **The influence of computers and informatics on mathematics and its teaching**,133-146 (Proc. ICMA symposium, Strasbourg, 1985)

Lance, Rand, Moon, *Teaching engineering analysis using symbolic algebra and calculus*, **Engineering Education**, (November 1985)

6.6 Journals and Conference Series

Communications ACM, 9,10 (1966) and **14,8** (1971)

Journal ACM, 18,4 (1971)

SIAM J. Comp, 8,3 (1979)

SIGSAM Bulletin of the ACM

SYMSAC Proceedings, ACM

EUROCAM and **EUROCAL Proceedings**, Springer Lecture Notes in Computer Science

Journal of Symbolic Computation, Academic Press

Computer Physics Communications

6.7 Newsletters

The Department of Mathematics at Colby College, Waterville, Maine, publish a CASE (Computer Algebra Systems in Engineering) newsletter. To receive copies, contact:

CASE Newsletter
Department of Mathematics
Colby College
Waterville
ME 04901
United States of America

Telephone: (207) 872-3255

A Maple newsletter is published by Brooks-Cole Publishing Company. Requests to be added to the mailing list should be sent to:

Brooks/Cole Publishing Co.
511 Forest Lodge Road
Pacific Grove
CA 93950-5098
United States of America

Telephone: (408) 373-0728

A Derive bulletin board has been created to enable users to communicate with one another. The telephone number of the bulletin board in the United States is (217) 337-0926.

6.8 Electronic Mail Discussion Groups

There are a number of user groups and discussion fora which may interest users of computer algebra systems. Each group has two addresses, one for contributions and another for requests to be added to the mailing list. If you want to send a contribution, make sure you mail it to the correct address since the 'request' userid cannot forward your query to the group moderator. It may even add your name to the mailing list again!

The addresses for contributions are on networks such as ARPA and InterNet. If in doubt, contact your local computing service for advice on how to send electronic mail to such sites.

REDUCE discussion forum

This is a forum for the exchange of ideas and suggestions about REDUCE. It is not moderated. Contributions are automatically distributed to everyone on the forum mailing list. Many of the issues raised in this forum are quite technical.

Contributions to:	reduce-forum@rand.org (InterNet)
Mailing requests to:	reduce-forum-request@uk.ac.nsfnet-relay

Symbolic Math bulletin

This newsletter is edited by Lawrence Leff (Southern Methodist University) and appears two or three times a month. It carries items on several of the more popular computer algebra systems, notably MACSYMA.

Contributions to:	leff@smu (UUCP)
Mailing requests to:	symalg-request@uk.ac.qmw.maths

Mathematica users group

This newsletter is edited by Steve Christensen (University of Illinois at Urbana-Champaign) and is dedicated to the exchange of ideas and comments about Mathematica.

Contributions to:	mathgroup@yoda.ncsa.uiuc.edu (InterNet)
Mailing requests to:	mathematica-request@uk.ac.qmw.maths

Maple users group

This newsletter is edited by George Labahn (University of Waterloo) and is a forum for the exchange of ideas and comments on Maple.

Contributions to:	maple_group@daisy.uwaterloo.ca (InterNet)
Mailing requests to:	maple-request@uk.ac.qmw.maths

UseNet

There are several discussion groups which are relevant to computer algebra on the UseNet service. The news group **sci.math.symbolic** is devoted to computer algebra, but news groups such as **sci.math** and **comp.sys.handhelds** may also carry interesting items. The latter includes much discussion about the Hewlett-Packard HP28-S and HP48-SX programmable calculators which have limited algebraic capability.

6.9 Software libraries

A library of utilities and information relating to REDUCE is maintained at the RAND Corporation by Dr Anthony C Hearn. Users can obtain files from the library by sending requests by electronic mail to an automatic library server. The server replies by sending the required file(s) by electronic mail. The address of the server is **reduce-netlib@rand.org** on the InterNet network. To obtain instructions on how to use the server together with an index of utilities, send an electronic mail message to the server containing the following two lines:

```
help
send index
```

6.10 User guides

Harper, D., *Maple User Guide*, University of Liverpool. Published by Manchester Computing Centre as NorthWest Document 75 (1989)

Appendix A: Availability of Computer Algebra Systems

This table provides information on the availability of the eight computer algebra systems in a compact quick-reference format. An up-to-date version of this table is available from the Computer Algebra Support Officer at any time.

In the following tables, an asterisk (*) indicates that the computer algebra systems is available for the machine and operating system as indicated. An X denotes that a version exists but may no longer be available as it has been superseded by a more recent computer algebra system. A plus (+) indicates that a version is available under a different name (the Atari ST version of muMATH is called RIEMANN).

Key: **RED** = REDUCE, **MPL** = Maple, **MTH** = Mathematica, **MAC** = MACSYMA, **DER** = Derive, **SPD** = Scratchpad, **SMP** = SMP, **MUM** = muMATH.

Manufacturer Machine	Operating system	RED	MPL	MTH	MAC	DER	SPD	SMP	MUM
Acorn									
Archimedes A440		*							
Amdahl									
S/370	UTS 1.1		*						
Apollo									
Apollo 3xxx,4xxx	UNIX	*	*	*	*			*	
Apollo DN1000	UNIX		*						
Apple									
Apple II									*
Macintosh Plus	MPW	*	*	*					
Macintosh Plus	Finder		*						
Atari									
Atari 1040ST	TOS	*	*						+
AT&T									
AT&T 7300,3B1	UNIX		*						
AT&T 3B2/500	UNIX		*						
Commodore									
Amiga	Amiga DOS		*						
Convex									
Convex C series	Convex UNIX		*						
Cray									
Cray 1	Cray UNIX	*							
Cray 2	Cray UNIX		*	*					
Cray XMP	Cray UNIX	*		*					
Cray YMP	Cray UNIX			*					
CDC									
Cyber 170, 180	NOS	*							
Data General									
Eclipse MV	AOS/VS	*							
DEC									
VAX	VMS	*	*	*	*			*	
VAX	ULTRIX	*	*	*	*			*	
VAX	UNIX 4.2 BSD	*	*	*	*			*	
MicroVAX	VMS	*	*	*	*			*	
MicroVAX	ULTRIX	*	*	*	*			*	
MicroVAX	UNIX 4.2 BSD	*	*	*	*			*	
DECstation 3100	ULTRIX			*					
Gould									
NPI	UNIX		*						
DN 9082	UNIX		*						

Key: **RED** = REDUCE, **MPL** = Maple, **MTH** = Mathematica, **MAC** = MACSYMA, **DER** = Derive, **SPD** = Scratchpad, **SMP** = SMP, **MUM** = muMATH.

Manufacturer Machine	Operating system	RED	MPL	MTH	MAC	DER	SPD	SMP	MUM
Hewlett-Packard									
HP 9000/300	HP/UX	*		*					
HP 9000/500	HP/UX			*					
HP 9000/8xx	HP/UX			*					
High Level Hardware									
Orion	UNIX 4.2 BSD	*							
IBM									
370 VM/SP	VM/CMS	*	*				*	*	
RISC System/6000	AIX		*						
6150 (RT/PC)	AIX		*	*			*		
PS/2	PC-DOS	*	*	*	*	*			*
PS/2	AIX		*				*		
IBM/PC and compatibles									
PC-compatibles	PC-DOS	*				*			X
PC-compatibles	MS-DOS	*				*			X
386-based machines	PC-DOS	*	*	*	*	*			X
386-based machines	MS-DOS	*	*	*	*	*			X
ICL									
2900 series	VME	*							
MASSCOMP									
5000	UNIX		*						
MIPS									
M/120				*					
M/1000				*					
M/2000				*					
PCS									
Cadmus	Minix		*						
Pyramid									
9800	UNIX		*						
Sequent									
Balance	Dynix		*						
Symmetry	Dynix		*						
Siemens									
PC MX2	Sinix		*						
SONY									
NEWS	UNIX	*		*					
Sord									
M68MX	CP/M-68K	*							

Key: **RED** = REDUCE, **MPL** = Maple, **MTH** = Mathematica, **MAC** = MACSYMA, **DER** = Derive, **SPD** = Scratchpad, **SMP** = SMP, **MUM** = muMATH.

Manufacturer Machine	Operating system	RED	MPL	MTH	MAC	DER	SPD	SMP	MUM
SUN									
SUN 3	SUN UNIX	*	*	*	*				
SUN 4	SUN UNIX	*	*	*	*			*	
386i			*						
Tektronix									
4400		*							
Toshiba									
AS3000	OS/AS	*							
Unisys									
Tahoe	UNIX		*						

Date of last revision of this table: 19 March 1990

Appendix B: Reduce Information Form

REDUCE INFORMATION FORM

REDUCE is an interactive program designed for general algebraic computations of interest to physicists, mathematicians and engineers. Its capabilities include:

1. expansion and ordering of polynomials and rational functions,
2. substitutions and pattern matching in a wide variety of forms,
3. automatic and user controlled simplification of expressions,
4. calculations with symbolic matrices,
5. arbitrary precision integer and real arithmetic,
6. rational, complex, modular and algebraic number arithmetic,
7. facilities for defining new functions and extending program syntax,
8. analytic differentiation and integration,
9. factorization of polynomials,
10. Dirac matrix calculations of interest to high energy physicists.

The most recent release of REDUCE (Version 3.3 with corrections) is dated January 15, 1988. This release includes a manual and tutorial aids to help with the understanding of the system. Two versions of this release are currently distributed by The RAND Corporation, and include all REDUCE source files. These are as follows:

1. Common Lisp Version: This is designed for use under any version of Common Lisp that conforms to the definition given in the book "Common Lisp - The Language" by Guy L. Steele Jr. (Digital Press, 1984). It has been installed under Franz Extended Common Lisp, HP Common Lisp, Kyoto Common Lisp, Lucid Common Lisp and Symbolics Common Lisp. Documentation is supplied outlining the steps necessary to build the system under a prototypical Common Lisp. Familiarity with the particular Common Lisp being used is desirable.

2. IBM Mainframe Version: This is designed for use on IBM System/360 or derivative computers (e.g., System 370, 4300 series, Amdahl 400 series and so on). It requires a minimum memory partition of 500K bytes for assembly and operation. This version includes a Standard Lisp interpreter and compiler. Documentation is supplied describing the implementation of the system under OS-based systems (such as MVS and TSO), CMS and MTS. However, CMS users are strongly advised to obtain the IBM CP/CMS version described later since the latter version is better integrated into the operating system.

The Common Lisp version is distributed in UNIX tar format on a 9 track magnetic tape at 1600 bpi. The IBM version is distributed on a 9 track magnetic tape in Standard Label EBCDIC format at a recording density of 6250 bpi. Processing of other recording modes requires an additional handling fee.

To obtain either of the above versions, please complete the accompanying

REDUCE Distribution Form and return it to the address shown. A fee of $500 is charged for each version, and includes the costs of documentation, magnetic tape and postage. Your order should be accompanied by a check for the appropriate amount for each version ordered. To keep distribution costs as low as possible, we request no purchase orders. Domestic orders will be shipped by first class mail, foreign orders by air mail.

European customers may also obtain a copy of either version from:

Volker Winkelmann
Rechenzentrum
Universitaet zu Koeln
Robert-Koch-Strasse 10
D-5000 Koeln 41, WEST GERMANY
Telephone: (0221) 478-5526
Electronic Mail: a0078@dk0rrzk0.bitnet.

Japanese customers may also obtain a copy of either version from:

SOFTEC Corp
3687-4 Obayashi, Shobumachi
Saitama, 346-01, JAPAN
Telephone: (0480) 85-6565
Telex: 2977460 SOFTECJ
Facsimile: (0480) 85-3662.

In addition, a number of other organizations are marketing versions of REDUCE under a third party licensing agreement. If you are interested in any of these, please direct your inquiries to the appropriate contact. A description of the currently available versions and the name of the distributor follows. The full addresses of these distributors are given later in this form.

1. Acorn Archimedes Version: This version, designed for use on the Acorn Archimedes A440 computer with 4 Mbytes of memory and a hard disk, is an implementation of REDUCE in Cambridge Lisp.

   Distributor: Codemist Limited.

2. Apollo Version. This is designed for use on Apollo workstations under the Berkeley UNIX, Apollo Release SR9.5 and higher. It is based on Portable Standard Lisp (PSL). The tape includes the PSL files needed to run REDUCE.

   Distributor: Department of Computer Science, University of Utah.

3. AS3000 Version: This is designed for use on a Toshiba AS3000 workstation under OS/AS V1.0 or higher and also on the Sun Microsystems Sun 3 workstation under Sun OS R3.0 or higher. It is based on Kyoto Common Lisp (KCL).

   Distributor: System Control Corporation.

4. Atari ST Version: This version, designed for use on the Atari 1040ST and

Mega machines with at least one megabyte of memory, and preferably a hard disk, is an implementation of REDUCE in Cambridge Lisp.

Distributor: Codemist Limited.

5. CDC 170 Version: This implementation is designed for Cyber 170 series computers and requires a minimum of 200,000 (octal) words of central memory to run. It can be installed under NOS or NOS/BE.

   Distributor: Rechenzentrum, University of Cologne.

6. CDC 180 Version: This implementation is designed for Control Data Cyber 180 computers running the NOS/VE operating system version 1.3. It will also run under NOS/VE 1.4. It includes a complete Portable Standard LISP (PSL) distribution upon which it is based.

   Distributor: Rechenzentrum, University of Cologne.

7. Cray Versions: These implementations of REDUCE are designed for the Cray 1, Cray X-MP, Cray Y-MP and Cray 2 computers. Implementations for both COS and UNICOS are supported. They are based on a version of Portable Standard Lisp (PSL) with vectorized big integer arithmetic. The relevant tape includes the PSL files needed to run REDUCE.

   Distributor: Konrad-Zuse-Zentrum fuer Informationstechnik, Berlin.

8. Data General Eclipse Version. This version, designed for the Data General Eclipse MV series machines running the AOS/VS operating system, is based on Kyoto Common Lisp (KCL).

   Distributor: R.I.M.S., Kyoto University.

9. Franz Lisp Version. This implementation, designed to run with the version of Franz Lisp distributed with the UNIX 4.2 BSD distribution for the DEC VAX series, was developed at the Tektronix Computer Research Laboratory in Beaverton, Oregon. This is an experimental version only, not a supported Tektronix product. The distribution tape includes the standard REDUCE sources and documentation, plus the Franz Lisp equivalent source code.

   Distributor: Computer Research Laboratory, Tektronix, Inc.

10. HP 9000 HP-UX Version. This version is currently available on the HP 9000 Series 300 under HP-UX Version 6.1. It is based on the HP Common Lisp Execution Environment and is supported under X Windows. Special features include a graphics module providing 2D and 3D graphics. The REDUCE documentation has been enhanced and completely rewritten. Future availability under Lucid LISP and on the HP 9000 Series 800 machines is planned.

    Distributor: Innovus Inc.

11. IBM VM/CMS Version. This is intended for use on VM/CMS systems. It is

based on an implementation of Portable Standard LISP (PSL) 3.2 for VM/CMS systems, and is fully integrated with the CMS operating system. The distribution tape includes the PSL files necessary to run REDUCE.

Distributor: I.M.S.S.S., Stanford University.

12. ICL Version: This is a version based on Cambridge LISP for use on ICL mainframes with the VME operating system.

    Distributor: Codemist Limited.

13. Intel 80386 Version. This is designed for Intel 80386-based machines running UNIX, with or without the 80387 coprocessor. It is based on Portable Standard Lisp (PSL), and the necessary PSL files are included in the distribution. This version has been tested on several such systems, including the Sun Microsystems 386i.

    Distributor: Konrad-Zuse-Zentrum fuer Informationstechnik, Berlin.

14. MathScribe. This is a visually oriented scientific computing environment which uses REDUCE as the algebra engine. The system has been developed at the Tektronix Computer Research Laboratory in Beaverton, Oregon. It includes symbolic and numerical computations, interactive plotting, and mouse-aided two-dimensional expression editing. The system runs under UNIX/X Windows environment on Sun 3 and Tek 4300 workstations.

    Distributor: Computer Research Laboratory, Tektronix, Inc.

15. MIPS Version. This is designed for machines using the MIPS microprocessor, such as the DECstation 2100 and 3100 and the Silicon Graphics IRIS machines. It is based on Portable Standard Lisp (PSL), and the necessary PSL files are included on the system tape.

    Distributor: Konrad-Zuse-Zentrum fuer Informationstechnik, Berlin.

16. Orion Version. This version, for the High Level Hardware 32 bit user microprogrammable superminicomputer running UNIX 4.2 BSD and Cambridge Lisp, has been developed at the University of Bath in collaboration with High Level Hardware.

    Distributor: High Level Hardware Ltd.

17. Siemens SINIX Version. This version is designed for use on Siemens Computers (including X20, MX2 and MX300) running SINIX Version 5.2 or later.

    Distributor: SINIX-Referenzzentrum, TU Braunschweig.

18. Sony NEWS Version. This is a version based on Tuneup Lisp for use on Sony NEWS workstations under UNIX 4.2 BSD.

    Distributor: Forbs System Co. Ltd.

19. Sord Version. This version, which runs on Sord's M68MX computer under CP/M-68K, is an implementation of REDUCE in Cambridge Lisp.

    Distributor: Sord Computer Corporation.

20. StaffLisp/68000 Version. This version, which runs on several Motorola 68000 based machines, uses the StaffLisp interpreter and compiler developed by B U G, Inc. B U G, Inc. can also provide help to anyone interested in implementing REDUCE on other systems.

    Distributor: B U G, Inc.

21. StaffLisp/86 Version. This version, which runs under MS-DOS on IBM-PC compatible machines and other Intel 8086-based machines with at least 640KB of main memory and a 640KB floppy disk, uses the StaffLisp/86 interpreter and compiler developed by B U G, Inc. In this version, the REDUCE test program runs in 40 seconds on an 8 mHz 80286.

    Distributor: B U G, Inc.

22. Sun 3 Version. This is designed for Sun Microsystems Inc. Sun 3 workstations. It is based on Portable Standard Lisp (PSL). The tape includes the PSL files needed to run REDUCE.

    Distributors: Konrad-Zuse-Zentrum fuer Informationstechnik, Berlin,
    B U G, Inc.

23. Sun 4 Version. This is designed for Sun Microsystems Inc. Sun 4 workstations. It is based on Portable Standard Lisp (PSL). The tape includes the PSL files needed to run REDUCE.

    Distributors: Konrad-Zuse-Zentrum fuer Informationstechnik, Berlin,
    B U G, Inc.

24. UO-Lisp Version. This complete REDUCE implementation runs on all IBM-PC compatible microcomputers, with MS-DOS 2.0 or higher. It requires a minimum of 512K bytes of RAM and dual floppy disks. A hard disk is recommended. The system includes an Emacs-like screen editor.

    Distributors: MicroMath Systems,
    Codemist Limited.

25. VAX/UNIX Version: This is designed for use on a Digital Equipment Corporation VAX series computer (including the MicroVAX) running UNIX 4.3 BSD or ULTRIX. It uses a virtual memory partition of 4 megabytes for assembly and operation. It is based on Portable Standard Lisp (PSL). The tape includes the PSL files needed to run REDUCE.

    Distributor: Department of Computer Science, University of Utah.

26. VAX/VMS Version: This is designed for use on a Digital Equipment Corporation VAX series computer (including the MicroVAX) running VAX/VMS

Version 4.1 or later, and uses a minimum virtual memory partition of 3.2 megabytes for assembly and operation. It is based on Portable Standard Lisp (PSL). The tape includes the PSL files needed to run REDUCE.

Distributor: Department of Computer Science, University of Utah.

The full addresses of these distributors are as follows:

B U G, Inc.:

B U G, Inc.
31-33, Shimonopporo
Atsubetsu-cho
Shiroishi-ku
Sapporo 004, JAPAN
Telephone: (011) 807-6666
Facsimile: (011) 507-6645.

Codemist Limited:

Codemist Limited
"Alta", Horsecombe Vale
Combe Down
Bath BA2 5QR, UNITED KINGDOM
Telephone: (0225) 837430.

Computer Research Laboratory, Tektronix, Inc.:

Dr. Guy Cherry
Symbolic Computation Program
Computer Research Laboratory
Tektronix, Inc.
P.O. Box 500 - MS 50/662
Beaverton OR 97077.

Department of Computer Science, University of Utah:

Loretta Cruse, PASS Secretary
Department of Computer Science
University of Utah
Salt Lake City UT 84112
Telephone: (801) 581-5017
Electronic Mail: cruse@cs.utah.edu
Facsimile: (801) 581-5843.

Forbs System Co. Ltd.:

Forbs System Co. Ltd
Wakao Building 4-34
Honcho, Naka-ku
Yokohama 231, JAPAN
Telephone: (045) 212-5020.

```
High Level Hardware Ltd.:

          Sales Office, High Level Hardware Ltd
          P.O. Box 170
          Windmill Road
          Oxford OX3 7BN, UNITED KINGDOM
          Telephone: (0865) 750494.

I.M.S.S.S., Stanford University:

          Tryg A. Ager
          Institute for Mathematical Studies in the Social Sciences
          Ventura Hall, Stanford University
          Stanford CA 94305
          Telephone: (415) 723-4117
          Electronic Mail: tryg@suwatson.stanford.edu

Innovus Inc:

          Innovus Inc.
          200 James St. South
          Hamilton
          Ontario L8P 3A9, CANADA
          Telephone: (416) 529-8117
          Facsimile: (416) 572-9586.

Konrad-Zuse-Zentrum fuer Informationstechnik, Berlin:

          Herbert Melenk
          Konrad-Zuse-Zentrum fuer Informationstechnik Berlin
          Heilbronner Str. 10
          D 1000 Berlin 31, WEST GERMANY
          Telephone: (030) 89604 195
          Electronic Mail: melenk@sc.zib-berlin.dbp.de
          Facsimile: (030) 89604 125.

MicroMath Systems:

          MicroMath Systems
          1057 Amoroso Place
          Venice CA 90291
          Telephone: (213) 399-7612
          Facsimile: (213) 305-1003.

Rechenzentrum, University of Cologne:

          Volker Winkelmann
          Rechenzentrum, Universitaet zu Koeln
          Robert-Koch-Strasse 10
          D-5000 Koeln 41, WEST GERMANY
          Telephone: (0221) 478-5526
          Electronic Mail: a0078@dk0rrzk0.bitnet.
```

R.I.M.S., Kyoto University:

Dr. Sin Hitotumatu
Research Institute for Mathematical Sciences
Kyoto University
Kyoto, 606, JAPAN
Telephone: (075) 7512111 Ext. 7226
Telex: 05422020 RIMS J.

SINIX-Referenzzentrum, TU Braunschweig:

Technische Universitaet Braunschweig
Institut fuer Betriebssysteme und Rechnerverbund
SINIX-Referenzzentrum fuer technisch-wissenschaftliche
Anwendersoftware
Bueltenweg 74/75
D-3300 Braunschweig, WEST GERMANY
Telephone: (0531) 391-3245
Facsimile: (0531) 391-4577.

Sord Computer Corporation:

Koji Fukuhara
Sord Computer Corporation
5-20-7, Masago, Chiba-shi
Chiba, 260, JAPAN
Telephone: (0472) 79-2611.

System Control Corporation:

Kazuo Sarashino
System Control Corporation
Hasegawa Building 2-5-2
Higashigotanda
Shinagawa-ku
Tokyo 141, JAPAN.

A mailing list is maintained for persons interested in receiving periodic information about REDUCE. Recipients of the software from RAND are automatically added to this list.

The RAND Corporation
Attn: Dr. Anthony C. Hearn
1700 Main Street
P.O. Box 2138
Santa Monica, CA 90406-2138
Telephone: (213) 393-0411, Ext. 6615
TWX: 910-343-6878
Facsimile: (213) 393-4818. Verify Number: (213) 393-0411, Ext. 6527
Electronic Mail: reduce@rand.org 21/12/89

% End of file redinfo.tex

Appendix C: Mathematica Price List

```
SYSTEM                        STANDARD  EDUCATIONAL   SPECIFICATIONS

Apollo
======

DN 2500 through 4500            $3,000   $2,000   Runs under SR10. Supports
                                                  Display Manager window
                                                  system. Distributed on
                                                  on 1/4" tape.

Cray
====

Cray X-MP, Cray Y-MP, Cray 2                      Runs under UNICOS. Call
                                                  for information on
                                                  pricing.

DEC
===

VAXstation II, 2000, 3100,                        VMS and ULTRIX versions
 3200, 3520, 3540, 8000;                          available for all VAX
 MicroVAX II, 2000;                               systems. VMS requires
 VAX 11/730                     $3,000   $2,000   release 5.0 or higher.
                                                  Distributed on TK50
VAX 11/750, 11/780, 11/785;                       cartridges (Digital
 MicroVAX 3300,3400             $6,000   $3,625   CompacTape). 9 track
                                                  tape available.
VAX 8200, 8250, 8300, 8350;
 MicroVAX 3500, 3600, 3800,
 3900                          $10,750   $6,500

VAX 6210, 6310, 8530;
 MicroVAX 6210                 $15,750   $9,500

VAX 6220, 6320, 8550, 8600,
 8650, 8700, 8810              $22,500  $13,750

VAX 6230, 6240, 6330, 6340,
 6350, 8800, 8820              $32,500  $19,500

VAX 6360, 8830, 8840           $52,500  $31,250

DECstation 2100, 3100           $3,500   $2,250   Runs only under ULTRIX
DECsystem 3100                  $6,500   $3,875   on RISC-based systems.
DECsystem 5400, 5800           $10,750   $6,500   ULTRIX version requires
                                                  release 2.2 or higher.
                                                  Distributed on TK50
                                                  cartridges (Digital
```

```
                                                 CompacTape).

Hewlett-Packard
===============

HP 9000/300 Series             $3,000   $2,000   Supports X Windows.  For
HP 9000/825 & 835              $7,750   $4,625   Series 300, requires HP
HP 9000/840, 850 & 855        $14,750   $8,875   ID module.  Runs under
                                                 HP-UX release 6.0 or
                                                 higher on Series 300;
                                                 3.0 or higher on Series
                                                 800. Distributed on HP
                                                 16 track 1/4" tape.

IBM
===
AIX/RT                         $3,000   $2,000   Supports X Windows. Runs
                                                 under 2.2.1 operating
                                                 system. Distributed on
                                                 3 1/2" floppy disks.

MIPS
====

RC 2030, RS 2030               $3,500   $2,250   Runs under operating
RC 3240 & M/120                $6,500   $3,875   system 4.0 or higher.
M/1000 & M/2000               $10,750   $6,500   Includes X Windows
                                                 driver. Distributed on
                                                 1/4" tape.

Silicon Graphics
================

Personal Iris                  $3,500   $2,250   Runs under operating
Professional Series                              system release 3.1
 IRIS/50 through 80GT          $7,750   $4,625   Runs under 4Sight
Power Series                                     network window system.
 IRIS/120 through 240GTX       $9,750   $5,875   Includes real-time 3D
                                                 graphics rendering
                                                 package.  Distributed
                                                 on 1/4" tape.

Sony
====

NEWS 700, 800, 1700, 1800       $3,000   $2,000  Runs under X Windows.
NEWS 1900                       $4,250   $2,750  Distributed on 3 1/2"
                                                 diskettes and 1/4" tape.

Sun
===
Sun-3                           $2,815   $1,745  Runs under SunView,
Sun-4 and compatibles           $3,000   $2,000  NeWS and X Windows
```

```
Sun386i                        $3,000   $2,000  network window
                                                systems.  Runs under
                               SunOS versions 3.5 and above. On Sun-3,
                               supports 68881 and FPA numeric coprocessors.
                               For Sun-3 and Sun-4: distributed on high-
                               density 1/4" cartridge tape. For Sun-386i:
                               distributed on 3.5" diskettes.

==============================================================================

Apple
=====

Standard Macintosh Version          $620        Standard Macintosh
Enhanced Macintosh Version                      version runs on any
(numeric coprocessor required)      $995        Macintosh; does not
                                                take advantage of any
                               numeric coprocessor. Enhanced Macintosh
                               version: requires numeric coprocessor;
                               runs on Macintosh II, IIx, SE/30, IIcx and
                               accelerator cards. 4 megabytes recommended.
                               2 megabytes suffice if a virtual memory system
                               is used. Supports Mathematica notebook
                               user interface. Supports connectivity to remote
                               Mathematica kernel.


MS-DOS Systems
====== =======

386 Version                        $870         640k memory & 1 megabyte
                                                extended memory required.
386/7 Version                                   Supports CGA, EGA, VGA,
(287 or 387 coprocessor required)$1,245         Hercules and AT&T VDC
                                                VDC 750 graphics standards,
386/Weitek                                      PostScript, LaserJet,
(Weitek coprocessor required)    $1,620         Epson FX, Epson LQ,
                                                IBM Proprinter, Proprinter
                                                X24, IBM QuickPrinter and
                                                Toshiba P3 compatible
                                                printers, and Encapsulated
                                                PostScript form.


Front Ends
==========

Macintosh                     $231.25 $156.25   1 megabyte required;
                                                2 megabytes recommended.
                                                Requires remote Mathematica
                                                kernel. Supports TCP/IP
                                                and serial connectivity.
```

```
==============================================================================


General Information
======= ===========

All versions (except Front Ends) include a hardbound copy of the book,
"Mathematica: A System for Doing Mathematics by Computer,"
together with version-specific documentation.

All workstation versions include T4010 and character-based graphics
support, as well as support for PostScript-based printers. All versions
licensed on a per CPU basis.

All prices listed are in US dollars. Payment to be made in US funds,
drawn on US bank. Import duties and local taxes are not included.
Shipping charges may be added, depending on distance and location.
Credit terms subject to approval.

Volume discounts are available.

Educational discounts are available for accredited universities
and colleges, and for high schools.

Upgrade contracts are available.

Wolfram Research, Inc.
P.O. Box 6059
Champaign, IL  61826-6059
Telephone:    217-398-0700
Fax:          217-398-0747
Orders only:  800-441-MATH

Effective: January 1990. This price list supersedes all previous
price lists. Prices and specifications subject to change without
notice. Prices do not apply to US and Canada.

Mathematica is a trademark of Wolfram Research, Inc.
All other trademarks are trademarks of their respective companies.
% End of file mathinfo.tex
```

Appendix D: Announcement: New Book

The authors of this document are currently preparing a book, to be published in early 1991 by John Wiley & Sons. It will be entitled *A Guide to Computer Algebra Systems* and will include a detailed comparison of the five most popular computer algebra systems (REDUCE, Maple, Mathematica, Derive and MACSYMA) plus several case studies which illustrate how computer algebra systems can be used to solve problems in science, engineering and mathematics. In addition, there will be a comprehensive bibliography of books, papers and articles on computer algebra.

In writing this book, the authors have drawn upon their extensive experience of using computer algebra systems. They have also made use of the resources of the highly successful U.K. Computer Algebra Support Project to guarantee that the book contains information that is comprehensive and up-to-the-minute in the rapidly-changing world of computer algebra systems.

The book will be invaluable to anyone who wants to know more about this fascinating subject. Computer algebra promises to revolutionise mathematics during the 1990s in the same way as pocket calculators in the 1970s. Engineers, mathematicians, scientists, teachers – anyone whose work involves the use of mathematics in research or in teaching will find the book a stimulating introduction to the many possibilities that these systems offer.

A Guide to Computer Algebra Systems by David Harper, Chris Wooff and David Hodgkinson. To be published by John Wiley & Sons in 1991.