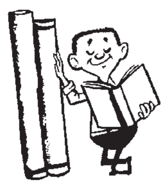
Introduction



Reading **Problems**

Computer Algebra Systems

Computer Algebra Systems (CAS) are software packages used in the manipulation of mathematical formulae in order to automate tedious and sometimes difficult algebraic tasks. The principal difference between a CAS and a traditional calculator is the ability of the CAS to deal with equations symbolically rather than numerically. Specific uses and capabilities of CAS vary greatly from one system to another, yet the purpose remains the same: manipulation of symbolic equations. In addition to performing mathematical operations, CAS often include facilities for graphing equations and programming capabilities for user-defined procedures.

Computer Algebra Systems were originally conceived in the early 1970's by researchers working in the area of artificial intelligence. The first popular systems were ***Reduce, Derive*** and ***Macsyma***. Commercial versions of these programs are still available. The two most commercially successful CAS programs are ***Maple*** and ***Mathematica***. Both programs have a rich set of routines for performing a wide range of problems found in engineering applications associated with research and teaching. Several other software packages, such as ***MathCAD*** and ***MATLAB*** include a Maple kernal for performing symbolic-based calculation. In addition to these popular commercial CAS tools, a host of other less popular or more focused software tools are availalbe, including ***Axiom*** and ***MuPAD***.

The following is a brief overview of the origins of several of these popular CAS tools along with a summary of capabilities and availability.

Axiom

Axiom is a powerful computer algebra package that was originally developed as a research tool by Richard Jenks' Computer Mathematics Group at the IBM Research Laboratory in New York. The project began in 1978 under the name Scratchpad and the first release of Axiom was launched in 1991. The research group is widely recognized for some outstanding achievements in the field of computer algebra and their work was augmented by contributions from a multitude of experts around the world. This collaboration continues to play an important part in the development of Axiom.

The object oriented design of Axiom is unique in its recognition of a 'universe' of mathematical components and the relationships amongst them. The strong typing and hierarchical structure of the system leads to it being both consistent and robust. Many users are wary of strongly typed languages but Axiom employs comprehensive type-inferencing techniques to overcome problems encountered in some older packages. It is a very open system with all library source code held on-line. This means that all the algorithms employed and the type-inferencing performed on behalf of the user are readily visible - making Axiom particularly attractive for teaching purposes.

The package includes an interface that allows users to develop their own object oriented programming code and add it in compiled form to the Axiom library which leads to significant performance improvements. Axiom is the only computer algebra package providing such a facility. Code can also be 'lifted out' of the Axiom system to form stand-alone programs which are independent of the Axiom environment.

Axiom was available commercially for several years through NAG (Numerical Algorithm Group) but their support of the product terminated in December 2001. It is now distributed as an open license through several ftp sites worldwide.

Web Site: http://home.earthlink.net/∼ jgg964/axiom.html

Derive

Derive had its roots in the Mu-Math program first released in 1979, with the first available version appearing in 1988. Its main competitive advantage to programs such as Maple and Mathematica was its limited use of computer resourses including RAM and hard disk space. While Derive does not have the same power and breadth of features as Maple or Mathematica, it is a more suitable package for the casual or occasional user and probably for many professional users whose needs are for standard rather than esoteric features.

Derive is the ideal program for demonstration, research, and mathematical exploration because of its limited use of computer resources. Users appreciate the easy to use, menu-driven interface. Just enter a formula, using standard mathematical operators and functions. Derive will display it in an easy-to-read format using raised exponents and built up fractions. Select from the menu to simplify, plot, expand, factor, place over a common denominator, integrate, or differentiate. Derive intelligently applies the rules of algebra, trigonometry, calculus, and matrix algebra to solve a wide range of mathematical problems. This non-numeric approach goes far beyond the capabilities of mere statistics packages and equation solvers that use only approximate numerical techniques. Powerful capabilities exist for 2D (Cartesian, polar, and parametric) and 3D graphing.

The current version of Derive for PC's is Derive 5, available from Texas Instruments for $199US or $99US for an Educational version. System Requirements include, Windows 95, 98, ME, NT, 2000 or XP compatible PC (minimum RAM and processor requirements are the same as the operating system requirements), CD ROM Drive, and less than 4 MB of disk space.

Web Site: http://education.ti.com/us/product/software/derive/features/features.html

Macsyma

Macsyma evolved from research projects funded by the U.S. Defense Advanced Research Projects Agency at the Massachusetts Institute of Technology around 1968. By the early 1970's, Macsyma was being widely used for symbolic computation in research projects at M.I.T and the National Labs. In the late 1970's the U.S. government drastically reduced the funding for symbolic mathematics software development. Authorities reasoned then that faster vectorized supercomputers and better numerical mathematical software could solve all U.S. mathematical analysis needs. Around 1980 M.I.T. began seeking a commercial licensee for Macsyma and in 1982, licensed Macsyma to Symbolics, Inc., an early workstation spin-off from M.I.T.

In April of 1992, Macsyma Inc. acquired the Macsyma software business from Symbolics, Inc. Bolstered by private investors from across the U.S. who understand the potential of the software, Macsyma Inc. has been reinvesting in Macsyma to make the software a suitable tool for a wide range of users and has already brought the PC user interface and scientific graphics up to modern windows standards.

Macsyma's strength lies in the areas of basic algebra and calculus, O.D.E.s, symbolic-numerical linear algebra and, when combined with PDEase, in symbolic and numerical treatment of P.D.E.s. The current version of PC Macsyma 2.4 is available for MS-Windows.

MAXIMA is a COMMON LISP implementation due to William F. Schelter, and is based on the original implementation of Macsyma at MIT. This particular variant of Macsyma was maintained by William Schelter from 1982 until he passed away in 2001. In 1998 he obtained permission to release the source code under GPL. Since his passing a group of users and developers has been formed to keep Maxima alive and kicking. Maxima itself is reasonably feature complete at this stage, with abilities such as symbolic integration, 3D plotting, and an ODE solver, but there is a lot of work yet to be done in terms of bug fixing, cleanup, and documentation.

Web Site: http://www.scientek.com/macsyma/main.htm

Maple

The MAPLE project was conceived in November 1980 with the primary goal to design a computer algebra system which would be accessible to large numbers of researchers in mathematics, engineering, and science, and to large numbers of students for educational purposes. One of the key ideas was to make space efficiency, as well as time efficiency, a fundamental criterion. The vehicle for achieving this goal was to use a systems implementation language from the BCPL family rather than LISP. Another aspect of making the system widely accessible was to design for portability, so that the system could be ported to the various microcomputers which were appearing in the marketplace. A very important aspect of achieving the efficiency goal was to carry out research into the design of algorithms for the various mathematical operations.

Maple is a comprehensive general purpose computer algebra system that can do both symbolic and numerical calculations and has facilities for 2 and 3-dimensional graphical output. Maple is also a programming language. In fact almost all of the mathematical and graphical facilities are written in Maple and not in a systems implementation language like other computer algebra systems. These Maple programs reside on disk in the Maple library and are loaded on demand. The programming language supports procedural and functional programming. Because of the clean separation of the user interface from the kernel and library, Maple has been incorporated into other software packages, such as Mathcad and MatLAB, to allow the symbolic functionality of the program to be accessable to as wide and audience as possible.

At the University of Waterloo, Maple 8 is available to all students through the Nexus system or through the University dial up system using an X-Windows package.
(see http://ist.uwaterloo.ca/cs/chip/gs/newgs.html for further details)

Web Site: http://www.maplesoft.com/

Mathematica

Mathematica is a product of Wolfram Research Inc. founded by the 'architect' of the system, Stephen Wolfram. Stephen Wolfram was a MacArthur Prize recipient in 1981. During this period (in the early 1980's) Stephen Wolfram developed a language called SMP (Symbolic Manipulation Processor) in C. This evolved into another program, Mathematica.

Mathematica, like Maple, offers capabilities for symbolic and numerical computations. Numeric computations can be carried out to 'arbitrary' precision, though obviously the higher the precision, the more time required to complete the calculation. There is a full suite of functions supporting 2- and 3-dimensional plotting of data and functions. Mathematica incorporates a graphics language capability which can be used to produce visualisations of complex objects.

There is an Applications Library available which includes a range of application-tailored tools written in Mathematica's language. These include: a 3-D real-time graphics tool; a control-system tool; an experimental data analyser; mechanical, electrical, and signal analysers.

Web Site: http://www.wolfram.com/

MuPAD

MuPAD is a general purpose (parallel) computer algebra system, developed at the University of Paderborn in Germany. MuPAD is available via FTP for several operating systems. The net version is limited in its memory access and cannot be used to solve real hard problems. But all non-commercial users can get the full-version for free by obtaining a MuPAD license (key) that unlocks all memory.

Web Site: http://www.mupad.de/

Reduce

The first version of REDUCE was developed and published by Anthony C. Hearn more than 25 years ago. The starting point was a class of formal computations for problems in high energy physics (Feynman diagrams, cross sections etc.), which are hard and time consuming if done by hand. Although the facilities of the current REDUCE are much more advanced than those of the early versions, the direction towards big formal computations in applied mathematics, physics and engineering has been stable over the years, but with a much broader set of applications.

Like symbolic computation in general, REDUCE has profited by the increasing power of computer architectures and by the information exchange made available by recent network developments. Spearheaded by A.C. Hearn, several groups in different countries take part in the REDUCE development, and the contributions of users have significantly widened the application field.

Today REDUCE can be used with a variety of hardware platforms from the Windows-based personal computer up to the Cray supercomputer. However, the primary vehicle is the class of advanced UNIX workstations.

Although REDUCE is a mature program system, it is extended and updated on a continuous basis. Since the establishment of the REDUCE Network Library in 1989, users take part in the development, thus reducing the incompatibilities encountered with new system releases.

REDUCE is based on a dialect of Lisp called “Standard Lisp”, and the differences between versions are the result of different implementations of this Lisp; in each case the source code for REDUCE itself remains the same. The complete source code for REDUCE is available through ftp sites worldwide.

Web Site: http://www.uni-koeln.de/REDUCE/ or http://www.zib.de/Symbolik/reduce/

Others

Mathcad is an easy to use tool for basic matematical calculations that has a Maple engine for doing symbolic computation. It is fully WYSYWIG.

Web Site: http://www.mathsoft.com/products/mathcad/

MatLab is a good number crucher. It also has a Maple engine for doing Symbolic operations. A Student version is available.

Web Site: http://www.mathworks.com/

GNU-calc runs inside GNU Emacs and is written entirely in Emac Lisp. It does the usual things: arbitrary precision integer, real, and complex arithmetic (all written in Lisp), scientific functions, symbolic algebra and calculus, matrices, graphics, etc. and can display expressions with square root signs and integrals by drawing them on the screen with ASCII characters. It comes with a well written 600 page on-line manual. You can FTP it from any GNU site.

Web Site: http://www.gnu.org/home.html

Glossary of Symbols

Symbol	Quantity
$\mathbf{arg(z)}$	argument or phase of the complex number $\boldsymbol{z}$
$\mathbf{bei(z),\ ber(z)}$	Kelvin functions of order zero
$\mathbf{bei}_{\boldsymbol{\nu}}\mathbf{(z),\ ber}_{\boldsymbol{\nu}}\mathbf{(z)}$	Kelvin functions of order $\boldsymbol{\nu}$
$\mathbf{B}(\boldsymbol{a},\boldsymbol{b})$	Beta function
$\mathbf{B_n}(\boldsymbol{a},\boldsymbol{b})$	Incomplete Beta function
$\mathbf{B_n}$	Bernoulli number
$\mathbf{B_n}(\boldsymbol{x})$	Bernoulli polynomial
$\mathbf{C}(\boldsymbol{x})$	Fresnel integral
$\mathbf{Ci}(\boldsymbol{x})$	Cosine integral
$\mathbf{D_n}(\boldsymbol{x})$	Debye functions
$\mathbf{D}_{\boldsymbol{\nu}}(\boldsymbol{x})$	Parabolic cylinder (or Weber's) function
$\mathbf{E}(\boldsymbol{k},\boldsymbol{\phi})$	Elliptic integral of the second kind
$\mathbf{E}(\boldsymbol{k})$	Complete elliptic integral of the second kind
$\mathbf{Ei}(\boldsymbol{x}),\ \mathbf{E_1}(\boldsymbol{x})$	Exponential integrals
$\mathbf{E}_{\boldsymbol{n}}(\boldsymbol{k})$	Generalized error function
$\mathbf{F}(\boldsymbol{k},\boldsymbol{\phi})$	Elliptic integral of the first kind
$\mathbf{F}(\boldsymbol{a},\boldsymbol{c},\boldsymbol{x}),\mathbf{F_1}(\boldsymbol{a},\boldsymbol{c},\boldsymbol{x})$	Kummer's function (or confluent hypergeometric function)
$\mathbf{F_1}(\boldsymbol{a},\boldsymbol{b},\boldsymbol{c},\boldsymbol{x})$	Gaussian hypergeometric function
$\mathbf{H}_{\boldsymbol{n}}(\boldsymbol{x})$	Hermite polynomial
$\mathbf{H}_{\boldsymbol{n}}^{\mathbf{(1)}}(\boldsymbol{x}),\ \mathbf{H}_{\boldsymbol{n}}^{\mathbf{(2)}}(\boldsymbol{x})$	Hankel functions (or Bessel functions) of the third kind

Symbol	Quantity
$\mathbf{h}_{n}^{(1)}(x),\ \mathbf{h}_{n}^{(2)}(x)$	Spherical Hankel functions
$\mathbf{I}_{\nu}(x)$	Modified Bessel Functions of order ν
$\mathbf{Im}(z)$	Imaginary part of the complex number z
$\mathbf{J}_{\nu}(x)$	Bessel function of the first kind of order ν
$\mathbf{j}_{n}(x)$	Spherical Bessel function
$\mathbf{K}(x)$	Complete elliptic integral of the first kind
$\mathbf{K}_{\nu}(x)$	Modified Bessel function of the second kind or order ν
$\mathbf{kei}(\mathrm{z}),\ \mathbf{ker}(\mathrm{z})$	Kelvin functions or order zero
$\mathbf{kei}_{\nu}(\mathrm{z}),\ \mathbf{ker}_{\nu}(\mathrm{z})$	Kelvin functions or order ν
$\mathbf{L}_{n}(x)$	Laguerre polynomial
$\mathbf{L}_{n}^{k}(x)$	Associated Laguerre polynomial
$\ell_{i}(x)$	Logarithmic integral
$\mathbf{M}(a,c,x)$	Confluent hypergeometric function
$\mathbf{P}_{n}(x)$	Legendre (or spherical) polynomial
$\mathbf{P}_{n}^{m}(x)$	Associated Legendre function of the first kind
$\mathbf{Q}_{n}(x)$	Legendre polynomial
$\mathbf{Q}_{n}^{m}(x)$	Associated Legendre function of the second kind
$\mathbf{q}$	Jacobi nome
$\mathbf{Re}(z)$	Real part of the complex number z
$\mathbf{S}(x)$	Fresnel integral
$\mathbf{Si}(x),\ \mathbf{si}(x)$	Sine integrals

Symbol	Quantity
$\mathbf{T}_n(x)$	Chebyshev polynomial of the first kind
$\mathbf{U}_n(x)$	Chebyshev polynomial of the second kind
$\mathbf{Y}_\nu(x)$	Bessel function of the second kind of order ν
$\mathbf{Y}_n(x)$	Spherical Bessel function
Greek Characters	
γ	Euler-Mascheroni constant
$\Gamma(x)$	Gamma function
$\gamma(x, a)$	Incomplete Gamma function
$\Gamma(x, a)$	Incomplete Gamma function
$(\lambda)_n$	Pochhammer symbol
$\binom{\lambda}{n}$	Binomial coefficient
$\Phi(x)$	Probability integral
$\zeta(x)$	Riemann's zeta function
$\Pi(k, \phi, a)$	Elliptic integral of the third kind
$\Pi(k, a)$	Complete elliptic integral of the third kind
$\Psi_n(x)$	Weber-Hermite function
$\psi(x)$	Psi function (or the logarithmic derivative of the Gamma function)

Selected References

1. **Abramowitz, M. and Stegun, I.A.**, *Handbook of Mathematical Functions*, Dover, New York, 1965.

2. **Artin, E.**, *The Gamma Function*, Holt, Rinehart and Winston, New York, 1964.

3. **Erdelyi, A., Magnus, W., Oberhettinger, F. and Tricomi, F.G.**, *Higher Transcendental Functions*, Bateman Manuscript Project, Vols. 1-3, McGraw-Hill, New York, 1953.

4. **Fletcher, A., Miller, J.C.P., Rosehead, L. and Comrie, L.J.**, *An Index of Mathematical Tables*, Vols. 1 and 2, 2 edition, Addison-Wesley, Reading, Mass., 1962.

5. **Hobson, E.W.**, *The Theory of Spherical and Ellipsoidal Harmonics*, Cambridge University Press, London, 1931.

6. **Hochsadt, H.**, *Special Functions of Mathematical Physics*, Holt, Rinehart and Winston, New York, 1961.

7. **Jahnke, E., Emdw, F. and Losch, F.**, *Tables of Higher Functions*, 6th Edition, McGraw-Hill, New York, 1960.

8. **Lebedev, A.V. and Fedorova, R.M.**, *A Guide to Mathematical Tables*, Pergamon Press, Oxford, 1960.

9. **Lebedev, N.N.**, *Special Functions and Their Applications*, Prentice-Hall, Englewood Cliffs, NJ, 1965.

10. **MacRobert, T.M.**, *Spherical Harmonics*, 2nd Edition, Mathuen, London, 1947.

11. **Magnus, W., Oberhettinger, F. and Soni, R.P.**, *Formulas and Theorems for the Functions of Mathematical Physics*, 3rd Edition, Springer-Verlag, New York, 1966.

12. **McLachlan, N.W.**, *Bessel Functions for Engineers*, 2nd Edition, Oxford University Press, London, 1955.

13. **Rainville, E.D.**, *Special Functions*, MacMillan, New York, 1960.

14. **Relton, F.E.**, *Applied Bessel Functions*, Blackie, London, 1946.

15. **Slater, L.J.**, *Confluent Hypergeometric Functions*, cambridge University Press, London, 1960.

16. **Sneddon, I.N.**, *Special Functions of Mathematical Physics and Chemistry*, 2nd Edition, Oliver and Boyd, Edinburgh, 1961.

17. **Watson, G.N.**, *A Treatise on the Theory of Bessel Functions*, 2nd Edition, Cambridge University Press, London, 1931.

18. **National Bureau of Standards**, *Applied Mathematics Series 41, Tables of Error Function and Its Derivatives*, 2nd Edition, Washington, DC, US Government Printing Office, 1954.

19. **Franklin P.**, *A Treatise on Advance Calculus*, Chapter 16, Dover Publications, New York, 1940.

20. **Hancock H.**, *Elliptic Integralsd*, Dover Publications, New York, 1917, p69 and p81.

21. **Tranter, C.J.**, *Integral Transforms in Mathematical Physics*, 2nd Edition, Methuen, London, 1956, pp 67-72.

22. **Stroud, A.H. and Secrest, D.**, *Gaussian Quadrature Formulas*, Prentice-Hall Inc., Englewood Cliffs, NJ, 1966.

23. **Bauer, F.L., Rutishauser, H. and Stiefel, E.**, "New Aspects in Numerical Quadrature," *Symposia in Applied Mathematics*, 15, Providence, RI, American Mathematical Society, 1963, pp. 199-218.

24. **Ralston, A.**, *A First Course in Numerical Analysis*, McGraw-Hill, New York, 1965.

25. **Gerald, C.F.**, *Applied Numerical Analysis*, 2nd Edition, Addison-Wesley, Reading, Mass., 1978.