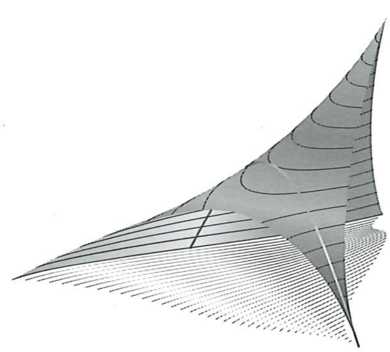
MapleTech

CONTENTS



Indexed/abstracted in: Current Contents, SciSearch, Research Alert, and CompuMath Citation Index.

Call for Papers : Maple In Industry

a Special issue of MapleTech

This is the first call for papers for a MapleTech special issue on Industrial Mathematics and Industrial Applications of Maple to be published in late 1998. The editors will be Dr. Ayowale Ogunye and Mr. Thomas Casselman.

In industry, symbolic and numeric computing environments have facilitated the design and development of complex engineering, technological and scientific systems. The synergy obtained from using a spectrum of computing environments, for example dedicated numeric computing environments like Matlab and Fortran, process simulation systems like Speedup, fluid visualization systems like Fluent, and modern general-purpose symbolic computing systems like Maple V have made previously intractable and impossible computations possible. The resulting gains in reducing code development time, execution times, and more rigorous and thorough analyses cannot be overemphasized.

Many universities are now creating "*industrial mathematics*" programmes. These programmes aim to expose students to industrial problems in the course of their study and to provide students with skills needed to tackle industrial applications. An important aspect of such programs is the use of computers and tools like Maple which are used to study and solve real problems that cannot feasibly be solved by hand.

We are interested in hearing from engineers and scientists who are involved in industrial applications which use Maple with or without other computing environments like Matlab, Scilab, Fortran, C, Excel, Speedup, Spice, Fluent, etc., in their projects. We are interested in papers that show clear, distinct and enabling advantages of using Maple in the overall computing scheme. Successful innovative results which use Maple for symbolic and/or numeric computations in industrial systems are of par ticular interest. Papers on industrial mathematics applications of Maple in education, and industrial applicationsplications of Maple in fields other than science and engineering, such as business related areas of economics, econometrics, operations research, and finance, are also encouraged.

All submissions will be refereed. The following guidelines will be important:

- Does the paper discuss an interesting application of Maple in an industrial setting?
- Does the paper show how Maple was useful or enabling in the analysis of the problem?
- Is the Maple code in the paper easy to follow?
- Is the paper easily understandable by other technical readers?

Important Deadlines

March 1, 1998	Papers submitted for review
May 2, 1998	Reviews completed & returned to authors.
June 1, 1998	Final versions of papers due at the editors.

Papers that we are unable to include in the special issue but which would otherwise be accepted, will be published in a regular issue of MapleTech.

Submission Details

Submission versions of papers must be either complete PostScript versions of papers, or Maple worksheets. Papers must be submitted either directly to the editors by electronic mail, or by anonymous ftp at ftp.cecm.sfu.ca and put in the directory pub/MapleTech/incoming under the authors name. Authors are requested to notify both editors of the special issue below of their submission. The editors may request authors to submit Maple code in their paper to verify results.

Final versions of papers must be in the form of Unix compatible LaTeX articles created using Maple V Release 3 or Release 4 using the MapleTech macros. LaTeX articles should be prepared using 10pt, two column format on 8.5 x 11 inch paper. A page charge of $100 per page in excess of 10 pages will apply. Detailed guideleines and tools for preparing articles for MapleTech, as well as LaTex style files can be obtained via anonymous ftp at ftp.cecm.sfu.ca in the directory pub/MapleTech at Simon Fraser University.

Dr. Ayowale B. Ogunye
Air Products and Chemical, Inc.
7201 Hamilton Boulevard
Allentown, PA 18195-1501
aogunye@ix.netcom.com
610-481-6570 (Phone)
610-481-4948 (Fax)

Mr. Thomas N. Casselman
C-Forth
11304 Rolling Hills Drive
Dublin, CA 94568
cassci@aimnet.com (Primary e-mail)
T. Casselman@lmco.com(Alternate e-mail)
510-551-5825 (Primary phone)
650-424-2661 (Alternate phone)
650-354-5400 (Fax)

A Note from the Editors Winter 1997, Vol.4, No.3

The response to the call for papers for the special issues of *MapleTech*, namely "Maple in the Mathematical Sciences" and "Maple in the Physical Sciences" was so swift and overwhelming that both of these special issues were completely "booked" in short order. Thus, we decided to have them published in sequence before returning to the regular issue. The positive response has given us the impetus to create yet another special issue devoted to *industrial applications* (see our call for papers).

After an interruption from two special issues in a row, it is a pleasure to see once again our section on news and announcements. Herein, you will find some interesting developments such as technologies "speaking" to each other, as in the case of the *Open Math* system, and in particular hybrid symbolic/numerical computation environments. As a matter of fact, the last (special) issue of MapleTech featured a number of articles from different avenues of research which use symbolic computation in Maple to prepare an efficient numerical computation to be done in Fortran, C, and Matlab. This trend seems to be part of a natural evolution towards more complete environments of computation with symbolic systems incorporating numerics and fast numerical systems incorporating symbolics (as in the case of the Maple/MATLAB link).

This issue will be the last issue in which Robert Lopez is editor of the Maple in Education section. Robert is now focussing his efforts and attention in writing a book. We welcome Robert Israel as the new section editor for the Maple in Education section, beginning with Vol 5 No 1. We also thank Robert Lopez for all his past contributions.

This particular issue includes a number of unconventional applications of Maple which are pleasing to our goal for pluralism. For example, we have an article on "state space dimension of the earth's surface mean temperature" by Rafael M. Gutierrez. We also have an article by Simon Woodward which illustrates how Maple is used to solve "Multi-Variable Optimal Control Problem in Fertiliser Economics". This qualifies as Maple's first article oriented towards agricultural science.

Tony Scott would like to thank Professor Yehuda Band of Ben-Gurion University and Dr. Claude Gomez INRIA-Rocquencourt, for allowing him the resources and time to contribute to the present issue of *MapleTech*.

Additional thanks to: Marc Rybowicz, Greg Fee, Bruno Salvy, Philippe Flajolet, Philippe Dumas, Manual Bronstein, Stephen Watt, Ann Kostant, Jacques Carette, Maurice Goursat, Stéphane Dalmas, Ramine Nikoukhah, Serge Steer, Francois Delebecque, Christophe Lavarenne, Yves Sorel, Michel Sorine, Paul Zimmerman, Andreas Sorgatz, Martine Verneuille, Jena-Pierre Quadrat, and Jacques Henry for their assistance and support; to the authors, to the members of the editorial board who make MapleTech possible; and to everyone at INRIA-Rocquencourt and Waterloo Maple Software who helped in in the preparation of this issue of MapleTech.

Editor-in-Chief : Tony C. Scott

News and Book Reviews

compiled by Blair Madore *

Scilab: new developments

Scilab is a free software developed at INRIA for numerical computation designed specifically for scientific applications. It provides an interactive user interface and a Matlab-like high level language for engineers and scientists, in particular, in Systems, Control and Signal Processing.

Scilab now includes Scicos, an interactive environment for modeling and simulation of hybrid systems. Scicos GUI can be used to construct complex block diagrams using system and user-defined basic blocks. A link with SynDEx is under development which should allow automatic distributed code generation from Scicos models onto a UNIX multi-workstation or real-time embedded multi-processor. This feature is particularly useful for rapid prototyping of control and signal processing systems.

Collaborative work with Waterloo Maple and Simon Fraser University has been undertaken to improve the link between Scilab and Maple, in particular with generation of fast BLAS Fortran code for definite integration problems.

In addition to UNIX versions, a Windows 95/NT version has been released.

A book entitled "Engineering and Scientific Computing with Scilab" edited by Birkhäuser will appear in 1998.

Scilab can be obtained from `ftp://ftp.inria.fr/INRIA/Scilab`. For more information about Scilab see `http://www-rocq.inria.fr/scilab`. For more information about SynDEx see `http://www-rocq.inria.fr/syndex`.

OpenMath

An important development in the world of mathematical applications is the need to communicate between different mathematical software. To communicate effectively the intended semantics of expressions must be known. OpenMath is being developed as a standard means of specifying the mathematical semantics. The actual semantics intended by two communicating applications are specified in an OpenMath dictionary. Communication is accomplished by having each application translate its own representation of the mathematical objects its manipulates into a common representation based on agreed upon OpenMath dictionaries prior to transmission.

Conventions for how to define dictionaries have been developed and several sample dictionaries exist at `http://www.openmath.org`. OpenMath has been used to demonstrate communication between various applications including Maple and Reduce at recent workshops and significant work is underway to make use of OpenMath to communicate between shipping commercial applications. A primary goal of OpenMath is to provide a truly effective international standard for communicating mathematics. To this end, members are interacting heavily with W3C's MathML (`http://www.w3c.org`) working group aimed at encoding mathematics on the Web.

The OpenMath organization consists of an international steering committee headed by Arjeh Cohen (The Netherlands) and various supporting members. The European Commission is funding a large ESPRIT project over the next three years (ESPRIT is a European program to support research and development in the area of information technologies) involving many of the European members (including NAG, Springer-Verlag, RIACA, The University of Bath and INRIA). This project is aimed at establishing OpenMath as an international standard, developing commercial applications and investigating the use of OpenMath in the publishing industry and for databases of mathematical information.

In North America, a number of organizations including IBM, the Centre for Experimental Mathematics at Simon Fraser University, the University of Western Ontario and Waterloo Maple Inc. have banded together to create NAOMI (North American OpenMath Initiative). This group, headed by Stephen Braham, was formed for the expressed purpose of developing real applications based on the OpenMath protocol. Maple has also taken steps to ensure that applications can be developed based on the OpenMath protocol by ensuring that the underlying OpenMath syntax and objects can be understood.

Using the IMSL/NAG-C library in a computer algebra system

Both libraries are widely known packages for numerical calculations. The computer algebra system MuPAD allows users to integrate C/C++ based algorithms utilizing IMSL or NAG-C routines and to use them during a MuPAD session. Symbolic and numerical computations can be used in combination to solve real world problems. This is realized with the concept of dynamic modules which is a very flexible and efficient concept of software integration based on dynamic linking. It will also be used to make the *OpenMath* protocol available in MuPAD. For more information and examples of dynamic modules see `http://www.mupad.de/PAPERS/MODULES/`. For an

*Thanks to Ben Friedman, Mike Monagan, Tony Scott, Carol Stewart, and everyone else who helped me compile this information.

announcement of the coming MuPAD Release 1.4.0 see http://www.mupad.de/R140/. This web page also contains information about downloading MuPAD.

Computer algebra group at SFU

The Center for Experimental and Constructive Mathematics (CECM) at Simon Fraser University (SFU) in Vancouver, Jon Borwein director, has recently formed three separate research groups. They are the Mathematics Research Group, Peter Borwein director, the Polymath Development Group, Stephen Braham director, and the Computer Algebra Group (CAG), Michael Monagan director. Please see http://www.cecm.sfu.ca for information about the CECM and the new groups. Information about the CAG may be found under http://www.cecm.sfu.ca/CAG .

In the Computer Algebra Group, we have three primary research projects. The first project is in the area of symbolic-numeric computing. Our main focus in on automatic differentiation and the creation of efficient C and Fortran codes. A first release of this work will appear in the `codegen` package in Maple V Release 5.

The second project is in the area of simplification of mathematical expressions. We are attempting to design an extensible simplifier based on field extensions a la Risch's structure theorem. This work involves a redesign of Maple's `normal` and `simplify` commands.

The third project is to develop algorithms for computing standard forms for systems of PDEs and DAEs (differential and algebra equations). The direction taken in this field has been to try to extend Buchberger's Grobner bases for algebraic equations to differential equations.

Students interested in persuing a PhD in computer algebra at Simon Fraser should contact Michael Monagan: monagan@cecm.sfu.ca .

ISSAC'98 Rostock

Following up on the very successful ISSAC'97 in Maui, the next International Symposium on Symbolic and Algebraic Computation (ISSAC) will be held August 13–15, 1998, at Universität Rostock in Germany.

ISSAC is a yearly international symposium that provides an opportunity to learn of new developments and to present original research results in all areas of symbolic mathematical computation.

The planned activities include invited presentations, research and survey papers, poster sessions, tutorial courses, vendor exhibits, and software demonstrations. Proceedings will be distributed at the symposium.

Proposals for workshops, tutorial courses, demonstrations, panel discussions or related activities are welcomed. User-groups, editorial boards or other associations desiring meeting space during the course of the symposium are encouraged to contact the conference organizers.

Papers must be received by Jan 20, 1998 for consideration. Acceptances will be sent out March 24 and camera ready copy must be received by April 28.

For more information please see
http://wwwteo.informatik.uni-rostock.de/ISSAC98.

CAMS/SCMA CSFD-98

The 19th annual meeting of the Canadian Applied Mathematics Society/Societe canadienne de mathematiques appliquees (CAMS/SCMA) will be held simultaneously with the 13th Canadian Symposium on Fluid Dynamics (CSFD) at Simon Fraser University at Harbour Centre in Vancouver, British Columbia, May 28-31,1998.

Contributed papers, in all areas of applied and industrial mathematics and fluid dynamics are invited. Abstracts should be submitted by April 1, 1998.

In addition to invited and contributed lectures the program includes minisymposia and workshops, the annual CAMS/SCMA doctoral dissertation award lecture, a poster session with a prize for the best graduate student poster, commercial displays, reception, banquet, social programs and more.

Further information will be posted at the CAMS/SCMA home-page on the World Wide Web, at
http://www.math.sfu.ca/cams .The e-mail address for general inquiries is: Penny_Southby@sfu.ca.

CAMS/SCMA 97

This year's Canadian Applied Mathematics Society meeting was one of the best ever. Hosted by the Fields Institute at the University of Toronto, May 30-June 1, it included over 16 different mini-symposia.

The mini-symposium on *Computer Technology and Undergraduate Mathematics Education*, organized by R.M. Corless was particularly exciting. Bruce Char's Plenary lecture brought everyone up to date on how the challenge to use technology in teaching has been met from the early 80's until now. Dr. Char (of Drexel University, Philadelphia) indicated we are at a crossroads, with the initial momentum to computerize mostly spent although a tremendous amount of work remains to be done. It seems many people have paused to debate whether we really know the proper uses of technology in the classroom. And yet the relentless spread of technology into every part of our society forces us to keep up - or our students will get there without us.

Jonathan Borwein, Director of the Centre for Experimental and Constructive Mathematics at Simon Fraser University, gave an incredibly compelling talk entitled "Doing Mathematics on the Web". This talk took place with slides physically located at Simon Fraser (3000km away) but projected live over the web at the Fields Institute. Dr. Borwein showed how the web is changing the way we all work and even the way we do and think about mathematics. Incredible challenges face us but, they seem to be dwarfed by the great opportunites available. Current struggles for control of the web and perhaps more importantly control of web publishing will certainly change all our professional lives.

Keith Geddes, of the University of Waterloo Dept. of Computer Science, explored the enhanced mathematical problem-solving capabilities of Maple which can be achieved via a hybrid symbolic-numeric approach in various problem areas. The objective is to achieve an appropriate combination of symbolic mathematical analysis and numerical computation.

Dr. Geddes showed us applications in algorithms for efficient numerical evaluation of functions, the evaluation of definite integrals in the presence of singularities, and the generation of problem-specific numerical methods for solving differential equations. The hybrid approach was seen to be especially suited to the teaching of scientific computation.

Hybrid symbolic numeric techniques are an important and currently very active area of research. This is certainly evidenced by news items on Scilab and Open Math in this issue and numerous recent MapleTech articles.

David Jeffrey, of the Applied Math Dept. at the University of Western Ontario, showed us that mathematicians and mathematics educators cannot ignore the ongoing revolution of using computers to do mathematics. Through several excellent examples Dr. Jeffrey illustrated how computer algebra has forced us to rethink what we mean by some mathematical notations and concepts. The new environment has exposed flaws and subtleties that we might otherwise have never noticed.

Some mathematicians and math educators have ignored the computer revolution in mathematics as much as possible. They treated computers as just another medium to replace pen and paper. That's not true and the problems exposed are the responsibility of the entire mathematics community to solve.

For more information about CAMS 97 see
`http://www.fields.utoronto.ca/cams97.html`

Maple ambassador program

Since Maple's inception there have always been "power users" who have led the way for other Maple users. Waterloo Maple Inc. (WMI) has always tried to identify and assist these individuals. Now this is being accomplished through a new Maple Ambassador program. The program was launched in the spring of 1997 and already many people have enrolled as Maple ambassadors.

Who are the Maple ambassadors? Here's a portion of WMI's description of an ambassador:

"Waterloo Maple Inc." (WMI) encourages and supports individuals who have become experts in using their products in an educational, research and commercial environment. Ambassadors share their expertise with others, through presentations at major events or giving workshops for other teachers, faculty and colleagues. The participants in this program provide a local face for Waterloo Maple Inc., a resource both for WMI and for other users.

Ambassadors in the educational field hold teaching or faculty positions in secondary or higher education and are instrumental in showing their peers how to integrate these software tools into their mathematics teaching and research.

Ambassadors in the commercial field have applied our technology in their field of work by showing peers how to integrate these software tools into their environment."

For more information on the opportunites and benefits of being a Maple Ambassador please contact Carol Stewart at WMI. Her email is `cstewart@maplesoft.com`.

News in brief

MapleTech is always pleased to announce Maple related shareware, books, websites, or other news of interest. Please direct any suggestions to Blair Madore, News and Book Reviews editor via email `madore@math.toronto.edu`.

New version of Coxeter and Weyl packages

Version 2.0 of the coxeter and weyl packages for are now available. You can dowload them (for Mac or Unix) at

`http://www.math.lsa.umich.edu/ ~jrs/ maple.html`

The coxeter package contains 37 Maple procedures for working with root systems and finite Coxeter groups. The weyl subpackage provides 8 additional procedures for manipulating weights and characters of irreducible representations of semisimple Lie algebras.

John Stembridge also maintains packages for working with symmetric functions and posets. They are available at the same website.

Handbook of ODEs online

A new online Computer-Handbook of ordinary differential equations is available at:

`http://lie.uwaterloo.ca/handbook _odes.html`

You can navigate the Handbook by following the links there found in connection with specific ODE types.

The Computer-Handbook of ODEs contain information related to most of the classifications of ODEs found in the literature, the related solving methods, the implementation of these methods in the context of the Maple ODEtools 3.0 package, and bibliographical references to textbooks where the ODEs are discussed in more detail.

The Maple input and output for tackling the ODEs, and/or hints for when the general solving method is not known, are also provided. One can mark the input using the mouse and paste it into a Maple worksheet to play around. For that purpose, ODEtools 3.0 is available at `http://lie.uwaterloo.ca/odetools.html`.

MapleTech bibliography

Thanks to Nelson Beebe there is now a complete bibliography available for MapleTech and its predecessors (the Maple Newsletter and the Maple Technical Newsletter). You can find the bibliography at

`ftp://ftp.math.utah.edu/pub/tex/bib/index.html`

This web site also contains many other bibliographies including Axiom, ISSAC conferences, Journal of Symbolic Computation, Maple, Mathematica, Reduce, and the SIGSAM Bulletin.

Nelson Beebe works at the Center for Scientific Computing at the University of Utah.

An electronic math-book

This exciting german site titled MMM@WWW (meaning `Mathematik mit Maple im WWW') can be found at `http://www.ikg.rt.bw.schule.de`. One feature of this site is the free program for managing or archiving your Maple worksheets. This tool is also available in english at `http://www.ikg.rt.bw.schule.de/virkla/mwsupl/mwspluse.html`.

Teaching chemical process calculations

This site at Clarkson University offers worksheets and lots of helpful information from people actually using Maple for a sophomore chemical engineering course. You can even see the assigments given for this term. Visit `http://people.clarkson.edu/~wilcox/cpc-indx.htm`. The worksheets were prepared by Professor Taylor at Clarkson University. He is an expert in the application of Maple to chemical engineering, and particularly thermodynamics.

Mass and energy balances for transport phenomena

A package of programs from Sam Davis in the Dept. of Chemical Engineering at Rice University are available on this topic. See `http://www.owlnet.rice.edu/~ceng303/Maple/maplemanual_ch5.html` and `http://www.owlnet.rice.edu:80/~ceng402/`.

Maple worksheets for complex analysis

The complete Maple software collection for the textbook "Complex Analysis: for Mathematics and Engineering" by John Matthews and Russell Howell is available at

`http://archives.math.utk.edu/software/msdos/complex.variables/complex_analysis/ca.html`

RSA encryptiondecryption tools

Harold Bien (`hbien@jhu.edu`) offers a free module on RSA encryption/decryption for Maple at `http://jhunix.hcf.jhu.edu/~hbien/public/index.html`. He developed this heavily commented code after a summer course with Professor Kaltofen at Rensaelear Polytechnic Institute. It is a much improved version of an original code by Eric Kaltofen.

New version of combstruct package

The new capabilities include automatic marked grammar generation, moment computation, and some algorithm analysis. The version is available at `http://www-rocq.inria.fr/algo/libraries`. Thorough introductory worksheets for the new capabilities can be found there as well.

Maple books

Below is a list of some recent publications about Maple V. The full booklist, now with over 200 entries in 11 languages, can be found on Waterloo Maple's website and has been recently updated to include links to online bookstores to provide users worldwide the ability to order almost any of the listed books. If you have corrections or additions to the full booklist, please send mail to Waterloo Maple at `authors@maplesoft.com`.

Waterloo Maple is committed to extending the selection of books about Maple V and offers assistance to all Maple authors and publishers. Contact Ben Friedman (`bfriedma@maplesoft.com`) for information, or, in Europe, contact Dr. Jenny Watson (`j.watson@maplesoft.co.uk`).

Recent Maple V Books

- Andersson, G., *Applied Mathematics with Maple*, Chartwell-Bratt, 1997, ISBN 0-862-38490-7
- Barrow, D., et. al, *Solving Ordinary Differential Equations with Maple V*, Brooks/Cole, 1997, ISBN 0-534-34402-X
- Brooks, D.R., *Introduction to Maple for College Algebra*, Brooks/Cole, 1997, ISBN 0-534-34757-6

- Burden, R. L., Faires, J. D., *Numerical Analysis, Sixth Edition*, Brooks/Cole, 1997, ISBN 0-534-95532-0
- Carlson, J., Johnson, J., *Multivariable Mathematics with Maple. Linear Algebra, Vector Calculus and Differential Equations*, Prentice Hall, 1997, ISBN 0-132-70315-7
- Coombes, K., Hunt, B., Lipsman, R., Osborn, J.,Stuck, G., *Differential Equations with Maple, Second Edition*, John Wiley and Sons, 1997, ISBN 0-471-17645-1
- Deeba, Gunawardeena, *Interactive Linear Algebra With Maple V : A Complete Software Package for Doing Linear Algebra*, Springer-Verlag, 1997, ISBN 0-387-98240-X
- Enns, R., McGuire, G., *Laboratory Manual with Maple for Nonlinear Physics for Scientists and Engineers*, Birkhäuser, 1997, ISBN 0-817-63841-5
- Enns, R., McGuire, G., *Nonlinear Physics with Maple For Scientists and Engineers*, Birkhäuser, 1997, ISBN 0-817-63838-5
- Etchells, T., Hunter, M., Monaghan, J., Pozzi, S., Rothery, A., *Mathematical Activities with Computer Algebra - a photocopiable resource book*, Chartwell-Bratt, 1997, ISBN 0-862-38405-2
- Gander, W., Hrebicek, J., *Solving Problems in Scientific Computing Using Maple and MATLAB*, Third Edition, Springer-Verlag, 1997, ISBN 3-540-61793-0
- Grinstead, C., Snell, J. L., *Introduction to Probability*, Second Edition, AMS, 1997, ISBN 0-821-80749-8
- Henderson, D., *Differential Geometry: A Geometric Introduction*, Prentice Hall, 1997, ISBN 0-135-69963-0
- Kamerich, E., *A Guide to Maple*, Springer-Verlag, 1997, ISBN 0-387-94116-9
- Klimek, G., Klimek, M., *Discovering Curves and Surfaces with Maple*, Springer-Verlag, 1997, ISBN 0-387-94890-2
- Kofler, M., *Maple: An Introduction and Reference*, Addison-Wesley Longman, 1997, ISBN 0-201-17899-0
- Lay, D., *Linear Algebra and Its Applications Second Edition*, Addison-Wesley, 1997, ISBN 0-201-82478-7
- Mathews, J., Howell, R., *Complex Analysis for Mathematics and Engineering*, Third Edition, Jones and Bartlett, 1997, ISBN 0-763-70270-6
- Meade, D., Bourkoff, E., *Engineer's Toolkit: Maple V for Engineers*, Addison-Wesley, 1997, ISBN 0-805-36445-5
- O'Neill, B., *Elementary Differential Geometry*, Second Edition, Academic Press, 1997, ISBN 0-125-26745-2
- Parker, R., *Maple for Algebra*, Delmar Publishing, 1996, ISBN 0-827-37407-0
- Parker, R., *Maple for Basic Calculus*, Delmar Publishing, 1997, ISBN 0-827-37408-9
- Parker, R., *Maple for Trigonometry*, Delmar Publishing, 1996, ISBN 0-827-37409-7
- Rosen, K. H., *Exploring Discrete Mathematics with Maple*, McGraw-Hill, 1997, ISBN 0-070-54128-0
- Shone, R., *Economic Dynamics: Phase Diagrams and Their Economic Application*, Cambridge University Press, 1997, ISBN 0-521-47973-8
- Williamson, R., *Introduction to Differential Equations and Dynamical Systems*, McGraw-Hill, 1997, ISBN 0-070-70594-1

Books in French

- Cornil, J., Testud, P., *Maple V Release 4, Introduction raisonné á l'usage de l'étudiant, de l'ingénieur et du chercheur*, Springer-Verlag, 1997, ISBN 3-540-631860
- Donato,P, *Maple: 15 themes mathématiques*, Diderot Editeur, 1997, ISBN 2-843-52043-6
- Dumas, P., Gourdon, X., *Maple, son bon usage en Mathématiques*, Springer-Verlag, 1997, ISBN 3-540-63140-2
- Leicknam, J.-C., *15 Leçons de Maple*, Masson, 1997, ISBN 2-225-85625-7
- Merceille, Nizard, Peronnet, *Exercices De Maple: calculs, tracés, programmation SUP et SPE*, Masson, 1997, ISBN 2-225-85444-0
- Rotaru, P., *Mathématiques avec Maple*, Diderot Editeur, 1997
- Rybowicz. M., Massias, J.-P., *Maple V Release 4*, Eyrolles, 1997, ISBN 2-212-08875-2

Hungarian

- Maróti, G., Mihály, K., *Maple a matematikai problémamegoldás Müvészaetéröl*, Movadat, 1997, ISBN 9-638-54172-5

Japanese

- Oguni, Tsutomu , *Maple V: Information & Computing*, Unknown, 1997, ISBN 4-781-90829-2

Spanish

- Pérez, C., *Métodos Matemáticos y Programación con Maple*, RA-MA Editorial, 1997, ISBN 8-478-97277-3

Russian

- Govorukhin, V., Tsybulin, V. , *Introduction to Maple: Mathematical package for everybody*, Mir, 1997, ISBN 5-030-03255-X

Books In Progress

- Articolo, J., *Partial Differential Equations with Maple*, Academic Press
- Betounes, *Partial Differential Equations with Vector Calculus and Maple (with CD-ROM)*, TELOS, Due: 12/97, ISBN 0-387-98300-7
- Char, B., Argabright, L., Busby, R., *Using Maple*, Springer-Verlag, ISBN 0-387-14233-9
- Corless, R, *Symbolic Recipes: Scientific Computing with Maple*, Springer-Verlag, Due: 9/97, ISBN 0-387-94210-6
- Greenberg, M., *Advanced Engineering Mathematics, Second Edition*, Prentice Hall, Due: 10/97, ISBN 0-133-21431-1
- Hale, M., Skidmore, A., *Differential Equations by Discovery, with Computer Activities*, Prentice Hall
- Kofler, M., *Maple: An Introduction and Reference*, Addison-Wesley Longman, Due: 10/97, ISBN 0-201-17899-0
- Prisman, E., *Derivative Securities with Maple V*, Academic Press
- Redfern, D., *The Practical Approach Utilities for Maple V Release 4*, Springer-Verlag, ISBN 0-387- 14225-8

Maple and the World Wide Web

Reid M. Pinchback *

Displaying equations

If you've ever tried to create web pages that contain mathematical notation for expressions and equations, you've learned just how inconvenient HTML[1] is for such tasks. If you haven't attempted this yet, then be forwarned that providing mathematical notation on a web page isn't as easy as providing ordinary text. In this month's column [1] we will look at some of the options for placing non-interactive mathematical content on the web.

BACKGROUND

Here is the crux of the problem. There are (at least) three important requirements for effectively representing and/or presenting mathematical content:

1. The freedom to specify and display the symbols most appropriate for the intellectual content.
2. A way to capture (or at least imply) the mathematical structure of the expressions.
3. A way to generate an appropriate two-dimensional rendering of the structure and content.

The idea behind the first requirement is that mathematical expressions often contain some special symbols that are used to represent operations or constants or other concepts within the intellectual domain being discussed. While it may be possible to substitute other symbols for the preferred ones, such as using an "L" in place of a lambda (λ), this is undesirable and creates a burden on the author to explain the unusual choice of symbols.

The second requirement tries to capture the notion that the placement of symbols in expressions is not arbitrary; the symbols are combined in specific ways to signify relationships or to reflect the algebraic or analytic steps that could be performed. For example, since we expect "1+2*3" to evaluate to 7, we wouldn't consider it acceptable for a browser to display "+123*". Even though all the correct symbols are present the placement of the symbols doesn't capture the intended interpretation of the expression. Each piece plays a specific role in the overall expression, and when we read and write expressions it is the relative placement of those pieces that gives us visual cues to their roles.

Finally the third requirement reflects the reality that not all expressions are presented with all the symbols in a linear order; fractions and subscripts and integral limits and suchlike all cause us to want to place symbols in a particular location with respect to other symbols. As the expressions increase in size and complexity, linearization becomes increasingly impractical and illegible. We want two-dimensional control over the rendering of the expression so that we can efficiently create visual cues about the relationships between the parts of the expression.

All this may sound straight-forward but HTML, which derives from SGML[2], was originally intended to deal mostly with the second requirement. The idea of a markup language was to insert some special codes — the markup tags — in a text file to indicate the structural relationships between the pieces of the document. Markup languages were not expected to replace WYSIWYG[3] editors and professional layout tools; the programs used to view markup files (like web browsers) were to have a great deal of lattitude in rendering the structured content in order to account for differences in document display across heterogeneous computing environments and to accomodate user-specified preferences. This has sometimes been interpreted as a prohibition on creating markup tags whose only purpose was to specify certain visual renderings, something an author might desire in order to address our third requirement of exerting two-dimensional control over the presentation of mathematical expressions.

Markup languages based on SGML didn't preclude the inclusion of mathematical notation or graphics, but until the explosion of interest in the web most of the R&D effort was directed at providing mechanisms for organizing and indexing plain text. Since (artistic arguments aside) the meaning of text is not dependent upon appearance-oriented controls, tags that support rendering concerns have historically been viewed a bit negatively by the established HTML/SGML community. The preferred way to support complex mathematical content would have been to substantially extend the specification of HTML to capture the structure of expressions; browsers implementing the extensions would be expected to somehow deal with the third requirement in a way that was consistent with the markup structure of the mathematical expressions. A "math mode" extension that was a hybrid of our second and third requirements was considered during the HTML 3.0 draft [2]; unfortunately little headway was made on the proposal and no mention of it remains in the current HTML 4.0 draft [3]. For our first requirement there is some support for special symbols within HTML by using markups called "entities", but the list of officially recognized entities

* I/T Delivery, M.I.T., reidmp@mit.edu, http://web.mit.edu/reidmp/www/home.html

[1] HyperText Markup Language

[2] Standard Generalized Markup Language

[3] What You See Is What You Get

MapleTech Vol. 4, No. 3, pp. 8–12; ISSN 1061-5733

is not sufficient for the variety of mathematical usages and as the list is static (it is part of a standard) there is no way for you to augment it to include the missing symbols that you would want to use.

Future HTML revisions

Periodically an effort is launched to revise HTML to improve its features and to cope with the compatability pressures of an active commercial browser market. While the math mode of the HTML 3 draft didn't survive — except for the <sup> and <sub> tags that provide simple superscripts and subscripts — there does seem to be a resurgence of this effort within the World Wide Web Consortium as something called "MathML", the Mathematical Markup Language [4]. The one aspect of math support that is present in the HTML 4 draft is an increased number of special-character entities. This still has the disadvantage of being a static list, but it is at least a much longer list containing many commonly-used mathematical symbols. Given current progress both in HTML standardization and independent vendor efforts to enable specialized font support in web pages [5] [6], the future looks good for eventually satisfying our first requirement in a port- able (i.e. non-browser-specific) way. Perhaps someday MathML may result in something that will address all three requirements.

OPTIONS

Getting back to the purpose of this article, there are many ways to try and cope with mathematical content on the web, so let's look at some of them and consider their strengths and weaknesses.

Preformatted text

Perhaps the simplest mechanism for dealing with equations is to treat HTML somewhat like a typewriter with memory. You can use the <pre> tags to contain text that you want presented literally the way you typed it, with no attempt by the browser to adjust or reformat it. This works much like the LaTeX verbatim environment and as such is only useful for simple notation and simple layouts. The layout is performed by manually positioning text instead of by interpreting the structure of the expression. The result is crude, doesn't readily scale to large expressions, and modifying or reorganizing the preformatted text is very error-prone and time-consuming.

Graphics

Given the obvious limitations of preformatted text, the next step that web-page authors can take is to present their mathematical content as graphic images on the web. This can be done simply by scanning printed pages, but with the large file sizes, the resulting binary-image printing problems, and the performance bottlenecks of slow internet connections and heavily-loaded web servers, you quickly have to move on to some better solution method. There are various conversion utilities—some of them available as freeware—that take a particular document format, extract ordinary text as text with HTML markup, and intersperse the text with small graphic images for those portions too complex for HTML (such as mathematical equations). Probably the best-known utility of this type is latex2html [7]. For LaTeX written specifically for latex2html processing the program is simple enough to use once installed and configured, but you have to be comfortable with the steps required to ftp, uncompress, untar, build, and install software in a Unix environment in order to use it. It only supports a subset of LaTeX so you will end up either substantially editing the source LaTeX, the resulting HTML, or the latex2html configuration information in order to get good results.

External viewers

Many browsers allow you to launch an external program to display particular types of files; these programs are often referred to as "viewers". The way they work is as follows:

- The user clicks on a link to a kind of file that normally can't be displayed by a web browser.
- The browser downloads the file from the web server, and as a result of the interaction between the browser, the server, and the extension on the filename (i.e. the part of the filename on the right side of "."), the browser determines something referred to as the "mime type" of the file.
- The browser looks at its own configuration information to see if it has been told about a viewer that is appropriate for the given mime type; if it finds one then the browser will launch the viewer program and give it the file to display.

From here whatever you do with the external viewer program is independent of the browser. It is like having two programs running at the same time, with no meaningful interaction or dependence between the two. In environments with a window-oriented graphical user interface (like X, MacOS, and MS Windows) you'll see two windows on your screen; you can use your mouse to navigate between them at will, and you should even be able to quit one program without affecting the other one.

External viewers are often used to make it easier to place files on the web that have some complicated kind of document format. For example, in the Unix environment you might modify your configuration of Netscape or Mosaic to launch "xdvi" whenever you try to download the DVI output of TeX or LaTeX. There are few limitations on external viewers; usually you just need some way to tell your browser how to launch the program and pass along the file that has been downloaded. The trickier part is in getting the mime information correct, but the details of that are beyond the scope of this article. Refer to the documentation available with your particular browser for more information.

Note that how good the appearance of the mathematical content is, and how easy it is to work with, now has nothing to do with the browser and is dependent entirely on the software used to produce the document and on the external viewer. While this sounds like a possible win, practically speaking it can be a clunky solution that requires an appropriate viewer program to be available on all the different kinds of computing platforms used by the readers of those web pages, and requires those readers to possess some relatively subtle knowledge about how web browsers are configured. It also has the disadvantage of losing the integration of the math content with other material that is being provided in HTML, comparable to situations where you have to continually bounce between different reference books for pieces of related information.

Plug-ins

Plug-ins are a refinement of the notion of an external viewer. A plug-in is a program that knows how to integrate its functionality with the web browser so completely that you might not even notice it is there. Usually no separate window will appear; whatever the plug-in will do for you, it displays within the confines of the browser window. While plug-ins provide a more integrated solution than external viewers, they tend to be available on very few platforms. Typically a given plug-in will only work on Mac, or on Windows, but rarely both and almost never in a Unix environment. With this caveat, there are some plug-ins that you really should look at if you want to place mathematical content on the web. None of these is in the public domain, but all can be downloaded for free.

- **Acrobat Reader** (by Adobe)

  One of the most flexible multi-platform solutions is provided by Adobe's Acrobat Reader [8]. Depending on your choice of computing environment (including Mac, MS Windows, and several versions of Unix), Acrobat Reader is available as either a plug-in or as an external viewer. The advantage of Acrobat is that you can convert any postscript file into Acrobat's Portable Document Format (PDF) using the "distiller" functionality (only available in the commercial versions of Acrobat). The distiller is easy to use and will let you create small document files that can provide web-like navigation features and good renderings of mathematical notation. Whenever you want to revise the PDF files you just revise the original documents (in Word, FrameMaker, LaTeX, or whatever you prefer), generate fresh postscript, and distill the postscript into PDF again.

  Notes:

  1. When distilling files be careful to know which fonts you've used, and either stick to common fonts or incorporate any unusual ones into the PDF document so that other people can render your files. This is something that even major publishing companies can forget; I once purchased the CD-ROM version of a book that had readable text but totally illegible mathematical notation because a key font hadn't been included in the PDF files.
  2. Use the most recent version of the Acrobat Reader (3.01 at the time of writing). Many new computers are shipped with Acrobat 2.0; the PDF file format changed in 3.0 and unfortunately Acrobat Readers aren't smart enough to spot the use of a newer file format. They will just indicate that the file is corrupt or unreadable for some reason.
  3. The plug-in versions of the Acrobat Reader are very convenient but they are newer and not yet as stable the external viewers. Don't be surprised to see a few bugs so keep an eye out for new upgrades on Adobe's web site.

- **techexplorer** (by IBM)

  This is an impressive plug-in that supports a large subset of LaTeX. Unfortunately it only exists on 32-bit MS Windows platforms [9], but if you only need a single-platform solution and you have large existing collection of LaTeX documents, then techexplorer is something to really pay attention to. Even without hand-editing my original LaTeX files I found techexplorer did a decent job of displaying them, which I have found is rarely the case with latex2html. I wish it had support for style files as they form the mainstay of most LaTeX-based publications, but otherwise I was quite pleased with this program.

  Notes:

  1. This plug-in integrates more smoothly with Netscape than it does with Microsoft's Internet Explorer, but it will work with both. Installation is easy but the web pages gloss over the steps. You can't just download the executable installer archive into any old directory you want; the directory has to be the same as the value of the "TEMP" environment variable or the installer gives an error about not having enough free

diskspace. The installer only looks for Netscape directories, so if you use Internet Explorer you'll need to tell the installer to use the MSIE's "Plug-ins" directory (which won't yet exist if you never installed a plug-in before; just create the directory yourself where you find the "Iexplore" application binary file and then proceed with the rest of the installation).

2. techexplorer doesn't really support keyboard-based navigation through displayed material, something which Netscape users probably won't notice but MSIE users will. You have to use the mouse for navigating through techexplorer materials displayed within MSIE.

- WebEQ (by the Geometry Center)

  As a Java applet WebEQ is the most cross-platform solution available for placing mathematical notation on the web [10]. It supports a very small subset of TeX and LaTeX called WebTeX which is mostly restricted to features that are used for displaying mathematical expressions. WebEQ actually began its existance by trying to implement the now-defunct HTML 3.0 math mode, and in the future the developers hope to add support for MathML. In the meantime, however, you will need to learn WebTeX and the basics of how to integrate Java applets into web pages in order to use WebEQ. Here is an example HTML page for displaying an equation containing radical signs. It assumes that the directory containing this web page has itself a subdirectory named "classes" containing the webeq fonts and compiled Java classes. The "size" parameter is the approximate point size of the font that will be used for display. The "eq" parameter is the equation or expression to be rendered.

```
<html>
  <head>
    <title>WebEQ Test</title>
  </head>
  <body>
    <applet codebase="classes"
        code="geom.webeq.app.mdraw"
        width=500 height=100>
        <param name=size value=36>
<param name=color value="#ffffff">
      <param name=eq
value="y=\sqrt{x}+\root{3}{2+\mu}">
    </applet>
  </body>
</html>
```

  which would result in the web page displaying (after a delay to load all the applet materials) in some suitable font:

  $y = \sqrt{x} + \sqrt[3]{2+\mu}$

- **Word Viewer** (by Microsoft)

  This may interest those of you who use Microsoft Word, particularly within either 16-bit or 32-bit MS Windows environments. The Word Viewer will display Word documents over the web, and those documents could contain whatever mathematical notation you've constructed within Word [11]. So far I've found that the viewer plug-in itself works well, but I haven't had much luck with getting the the Equation Editor included with Word in Microsoft Office 97 to do anything useful without crashing. Even without the use of this utility or Word itself, the Word Viewer could still be useful for anyone having a word processor that can save files in Rich Text Format (RTF); the viewer is equally happy reading either RTF or Word documents.

Conclusion

I began this column by saying there are at least three important requirements for effectively representing and/or presenting mathematical content. The reason for the qualification "at least" is because there are, of course, a host of intellectual or pedagogical objectives that have little to do with the mechanics of placing mathematical notation on the web. Ultimately it is these other concerns you want to be able to devote your energy to; if that were not the case we all might as well still be grinding our own ink and cutting quills to use as pens. While web-based technologies still have a little way to go before authors will have the simplicity of preparation and flexibility of presentation that we all desire, attainment of that goal is not far off in the future. We have some workable tools now and they are only going to get better over time.

References

[1] Reid M. Pinchback: *Displaying Equations*, http://web.mit.edu/maple/www/how-to/math-to-web-1.html (last modified September 16, 1997) is the on-line version of this article.

[2] World Wide Web Consortium: *HTML 3.0 Draft (Expired!) Materials*, http://www.w3.org/MarkUp/html3 (December 21, 1995).

[3] World Wide Web Consortium: *HTML 4.0 Working Draft Release*, http://www.w3.org/TR/WD-html40-970708 (July 25, 1997).

[4] World Wide Web Consortium: *HTML Math Overview*, http://www.w3.org/Math (May 15, 1997).

[5] Adobe Systems Incorporated: *The Best of Both Worlds: OpenType — Introduction*, http://www.adobe.com/newsfeatures/ opentype/ main.html (visited August 27, 1997).

[6] Bitstream Inc., *Dynamic Fonts*: http://www.bitstream.com/world/dynamic.htm (visited August 27, 1997).

[7] Nikos Drakos: *All About LaTeX2HTML*, http://cbl.leeds.ac.uk/nikos/tex2html/ doc/latex2html/ latex2html.html (November 29, 1996).

[8] Adobe Systems Incorporated: *Adobe Acrobat Overview*, http://www.adobe.com/acrobat (visited August 27, 1997).

[9] IBM: *IBM techexplorer product page for V1.2*, http://www.ics.raleigh.com/ics/techexp.htm (visited August 27, 1997).

[10] The Geometry Center: *WebEQ 2.0*, http://www.geom.umn.edu/software/WebEQ (visited August 27, 1997).

[11] Microsoft Corporation: *Microsoft Word Viewer*, http://www.microsoft.com/msword/internet/ viewer (visited August 27, 1997).

Tips for Maple Users and Programmers

Michael B. Monagan [§]

In the first part of this column I answer three questions from users on relatively straight-forward tasks in Maple, namely, how to plot the complex roots of a polynomial, how to flatten a nested structure, and how to test for a quadratic polynomial.

In the second part of the column I answer two more difficult questions from users where the solutions require a serious programming effort on the part of the user. These problems require some thought and care to find a good and short solution.

All these examples illustrate issues of concern in programs which manipulate formulae, issues the programmer should think about, hard, before coding. The solutions show the usage of helpful Maple commands like `op`, `seq`, `map`, `type` that will help users.

Finally I explain how to locate Maple library code and install fixes to bugs in the Maple library.

Graphing complex roots

Suppose we want to graph the complex roots of a polynomial. The `plot` command accepts as one of its inputs a list of [x,y] points. So if we can create a list of the complex roots then convert it into a list of [real,imaginary] points, we are done. Let's use a random polynomial of degree 10.

```
> f := randpoly(x,degree=10,dense);
```

$$f := -85\,x^{10} - 55\,x^9 - 37\,x^8 - 35\,x^7 + 97\,x^6 + 50\,x^5 + 79\,x^4 + 56\,x^3 + 49\,x^2 + 63\,x + 57$$

To get the complex roots we use the `fsolve` command with the complex option. I've put the sequence of roots returned by `fsolve` into a list. You should always put the result of `fsolve` (or `solve`) into something because it returns a sequence of zero or more roots. If you don't care about multiple roots then you can put the result into a set instead of a list.

```
> p := [fsolve(f,x,complex)];
```

$$p := [-1.205972191, -.6886654669 - .3711899572\,I, -.6886654669 + .3711899572\,I, -.2931711860 - .9147123100\,I, -.2931711860 + .9147123100\,I, .1070082661 - 1.066197851\,I, .1070082661 + 1.066197851\,I, .5794502798 - .6404455545\,I, .5794502798 + .6404455545\,I, 1.149669582]$$

The `Re` and `Im` commands can be used to select the real and imaginary parts of a complex number. For the first root, which happens to be real, we obtain

```
> [Re(p[1]),Im(p[1])];
```

$$[-1.205972191, 0]$$

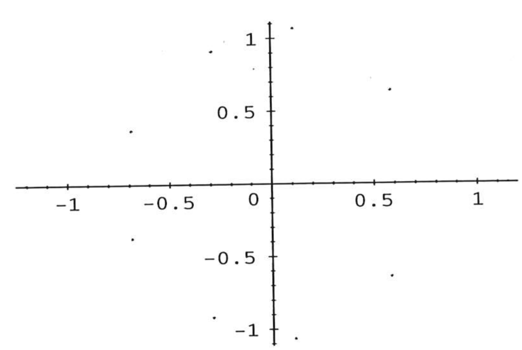
To create such a list for each complex root we use the `map` command. We map a Maple procedure onto the list of roots to obtain a new list of lists of the real and imaginary parts.

```
> r := map( z -> [Re(z),Im(z)], p );
```

$$r := [[-1.205972191, 0], [-.6886654669, -.3711899572], [-.6886654669, .3711899572], [-.2931711860, -.9147123100], [-.2931711860, .9147123100], [.1070082661, -1.066197851], [.1070082661, 1.066197851], [.5794502798, -.6404455545], [.5794502798, .6404455545], [1.149669582, 0]]$$

The `plot` command will plot a list of data. Use the style=point option, otherwise, Maple will draw lines between the points.

```
> plot(r,style=point);
```



The main Maple tool shown in this example is the `map` command. It is perhaps the single most useful command in Maple for manipulating data. It is useful for both interactive

§Department of Mathematics & Statistics, Simon Fraser University, Burnaby B.C., V5A 1S6 Canada, monagan@cecm.sfu.ca

usage of Maple and programming usage. Note, new in Release 4 is the `complexplot` command in the plots package. The same plot can be generated by

```
> plots[complexplot](p,style=point);
```

Flattening and associativity

A user asked how to flatten an object in Maple. This question appeared on the net recently and the user argued that it should be a built-in command in Maple. Here we show how to do it for simple cases with built-in Maple commands and how to program the more general cases.

Maple flattens sums, products, minimums and maximums automatically. For example

```
> a+(b+(c+d))+e;
```

$$a+b+c+d+e$$

```
> a*(b*(c*d)*e);
```

$$a\,b\,c\,d\,e$$

```
> min(a,min(b,min(c,d)),e);
```

$$\min(a,\ b,\ c,\ d,\ e)$$

The above flattened results are the simplified form that Maple uses. The action of flattening an object requires that it be associative. Thus two different groupings will give the same flattened result allowing one to detect equality. For example

```
> ( x*(y*z) ) / ( (x*y)*z );
```

$$1$$

```
> min(a,min(b,c)) - min(min(a,b),c);
```

$$0$$

Suppose we have the list of lists

```
> L := [[a,b],[c],[d,e,f]];
```

$$L := [[a,\ b],\ [c],\ [d,\ e,\ f]]$$

The simplest way to flatten L into one big list is to use the `op` and `map` commands. The `op` command returns a sequence of all operands of an object.

```
> op(L[1]);
```

$$a,\ b$$

```
> op(L[1]), op(L[2]), op(L[3]);
```

$$a,\ b,\ c,\ d,\ e,\ f$$

The `map` command applies a function to each operand of an object. Thus we have

```
> map(op,L);
```

$$[a,\ b,\ c,\ d,\ e,\ f]$$

This does not work if we have a non-homogeneous list. For example

```
> L := [[a,b],c+d,[e,[f,g]]];
```

$$L := [[a,\ b],\ c+d,\ [e,\ [f,\ g]]]$$

```
> map(op,L);
```

$$[a,\ b,\ c,\ d,\ e,\ [f,\ g]]$$

I use the following in this case.

```
> map( x -> if type(x,list)
>   then op(x) else x fi, L );
```

$$[a,\ b,\ c+d,\ e,\ [f,\ g]]$$

One way to flatten a list recursively is the following.

```
> flatten := proc(f)
> if type(f,list)
> then map(op@flatten,f)
> else [f] fi
> end:
> flatten(L);
```

$$[a,\ b,\ c+d,\ e,\ f,\ g]$$

The @ operator in Maple is the composition operator. That is, this is short for

```
map(proc(x) op(flatten(x)) end, f)
```

But lists of lists are not the only objects that one might want to flatten. What I also want to show you is how to make this flattening a property of your own function. Let us code our own MIN function which flattens itself. Miniums of real numbers are associative but they are also commutative. I use here a set instead of a list because sets are sorted and this will take care of commutativity. Also, because duplicates are removed from sets, it will take care of the additional property that MIN(a,a,b) = MIN(a,b).

```
> flatten := proc(x,f)
> if op(0,x)=f then op(x)
> else x fi end:
```

```
> MIN := proc() local A;
> A := {args};
> A := map(flatten,A,MIN);
> 'MIN'(op(A));
> end:
```

```
> MIN(a,MIN(b,c)) = MIN(MIN(a,b),c);
```

$$\mathrm{MIN}(a,\ b,\ c) = \mathrm{MIN}(a,\ b,\ c)$$

```
> MIN(a,MIN(d,MIN(c,b)),a);
```

$$\mathrm{MIN}(a,\ b,\ c,\ d)$$

Is this polynomial quadratic?

Suppose you want to test whether a formula f is a polynomial of the form $a\,x^2 + b\,x + c$, a quadratic polynomial in x, and if so, you want to get your hands on the coefficients a, b, c for further processing. Applications include integrals of special forms such as $\int_\alpha^\beta x^n\, e^{(a\,x^2+b\,x+c)}\,dx$.

Here is a one line program which takes as input the formula f and the variable x and outputs true if f is a quadratic polynomial in x, and false otherwise. We'll worry about returning the coefficients a, b, c in a minute.

```
> isquadratic := proc(f,x)
> type(f,polynom(anything,x)) and
>   degree(f,x)=2        end:
```

Our program first checks if f is a polynomial in x and if so, if it is degree 2 in x. In the type polynom(*anything*, x)*,* we are allowing coefficients of any kind. Let's run it on some examples

```
> ex1 := 3*x^2-2*x+1;
> isquadratic(ex1,x);
```

$$ex1 := 3\,x^2 - 2\,x + 1$$

$$true$$

```
> ex2 := x*(x-a)+x;
> isquadratic(ex2,x);
```

$$ex2 := x\,(x-a) + x$$

$$true$$

```
> ex3 := x*(x^2-a);
> isquadratic(ex3,x);
```

$$ex3 := x\,(x^2 - a)$$

$$false$$

```
> ex4 := x^2*exp(x);
> isquadratic(ex4,x);
```

$$ex4 := x^2\,e^x$$

$$false$$

```
> ex5 := a/x+b*x^2;
> isquadratic(ex5,x);
```

$$ex5 := \frac{a}{x} + b\,x^2$$

$$false$$

Now, is our isquadratic command correct? Unfortunately, the answer is no. Here is an example

```
> f := x^2/a-(x/a-1)*(x-2);
> isquadratic(f,x);
```

$$f := \frac{x^2}{a} - (\frac{x}{a} - 1)\,(x-2)$$

$$true$$

It looks like a quadratic polynomial, but if you simplify it, you will see that in fact it is linear as the quadratic terms cancel.

```
> s := simplify(f);
> isquadratic(s,x);
```

$$s := \frac{2\,x + a\,x - 2\,a}{a}$$

$$false$$

What is the problem? The degree function in Maple does not simplify before it computes the degree. Therefore, we must first write any input which satisfies the polynomial type test in standard form, namely $a_n\,x^n + a_{n-1}\,x^{(n-1)} + ... + a_1\,x + a_0$. We use the collect command to do this.

```
> c := collect(f,x);
```

$$c := (1 + \frac{2}{a})\,x - 2$$

Next we need to determine whether the leading coefficient is zero or not. In this example it is obvious that the leading coefficient is not zero so let me use another example where it is not so obvious.

```
> g := a*(a-1)*x^2+a^2*(1-x)*x+a*(x^2-1);
```

$$g := a\,(a-1)\,x^2 + a^2\,(1-x)\,x + a\,(x^2-1)$$

```
> collect(g,x);
> degree(",x);
```

$$(a\,(a-1) - a^2 + a)\,x^2 + a^2\,x - a$$

$$2$$

The main command to test for zero in Maple is normal. Applying normal to the coefficients using collect

```
> collect(g,x,normal);
> degree(",x);
```

$$a^2\,x - a$$

$$1$$

Here, collect first writes the input polynomal in the standard form. Then it applied normal to the coefficients. In this case the quadratic coefficient simplified to zero.

Let us now write the isquadratic procedure accordingly. We have also declared types for the input parameters.

```
> isquadratic := proc(f::algebraic,x::name)
> type(f,polynom(anything,x)) and
> degree(collect(f,x,normal),x)=2
> end:
```

```
> isquadratic(g,x);
```

$$false$$

Is `isquadratic` always correct now? The answer is, again, no, because the `normal` command in Maple is limited in its ability to recognize whether a given formula is equivalent to zero or not. Shouldn't Maple's `normal` command be improved then? Yes it should, and this is being done. In Release 4, the command `radnormal` is an extension of the `normal` command which handles radicals. But it will never be always correct because it's a fact of life that it is provably impossible to determine whether an arbitrary expression is zero or not. This important issue is further discussed in our Tips column in MapleTech Vol 3 No 2, 1996.

Now it is time to compute and return the coefficients. We could design our `isquadratic` procedure in the following way. If the input is a quadratic polynomial, then return a list of the three coefficients in descending degree, otherwise the special symbol FAIL. We could do this as follows

```
> isquadratic := proc(f,x) local g;
> if type(f,polynom(anything,x)) then
>    g := collect(f,x,normal);
>    if degree(g,x)=2 then
>       RETURN([coeff(g,x,2),coeff(g,x,1),
>               coeff(g,x,0)])
>    fi
> fi;
> FAIL
> end:
> isquadratic(ex1,x);
```

$$[3, -2, 1]$$

The problem with this design is that when you want to use it, you must first check to see if the result is FAIL before you can access the coefficients like this

```
> result := isquadratic(ex1,x):
> if result=FAIL then dothis else dothat fi;
```

It's simpler to use if you design the procedure instead as follows. Make it boolean valued and if the input is a quadratic polynomial, return the coefficients through a parameter. Since in this case there are only three, I returned them individually

```
> isquadratic := proc(f,x,a,b,c) local g;
> if type(f,polynom(anything,x)) then
>    g := collect(f,x,normal);
>    if degree(g,x)=2 then
>       a := coeff(g,x,2);
>       b := coeff(g,x,1);
>       c := coeff(g,x,0);
>       RETURN(true);
>    fi
> fi;
> false
> end:
```

The usage is now quite convenient

```
> if isquadratic(ex1,x,'a2','a1','a0') then
>    a2,a1,a0;
> else FAIL
> fi;
```

$$3, -2, 1$$

Well, now that you've learned how to do this, you'll be pleased to hear that Release 4 comes with this facility for you in the `ispoly` command which works for a polynomal of any degree. You can write ispoly(f, n, x, a_0, a_1, ...) which tests if the input formula f is a polynomial of degree n in the variable x and if so assigns the additional parameters the values of the coefficients in ascending order. The `ispoly` command uses Normalizer which is a variable assigned to the `normal` command. You may assign it to simplify or radnormal. Our example is therefore

```
> ispoly(ex1,quadratic,x,'a0','a1','a2');
> a2,a1,a0;
```

$$true$$

$$3, -2, 1$$

Splitting an integral with $\int_a^b f(t)\,dt = \int_a^c f(t)\,dt + \int_c^b f(t)\,dt$

A user asked if Maple can split up an integral into two parts. For example, given the integral

```
> f := x + int( h(t), t=-x..infinity );
```

$$f := x + \int_{-x}^{\infty} h(t)\,dt$$

how can we split it up at $t = 0$ into

```
> g := x + int( h(t), t=-x..0 ) +
>          int( h(t), t=0..infinity );
```

$$g := x + \int_{-x}^{0} h(t)\,dt + \int_{0}^{\infty} h(t)\,dt$$

Well, there is no facility to do this in Maple. So let's try to program it ourselves. I'm going to do it by walking through an input formula looking for any definite integrals and splitting them if I see them. Let me first write a program which finds and prints the definite integrals in a formula

```
> findintegral := proc(f) local g;
> if type(f,{name,numeric,procedure}) then f
> elif type(f,function) and op(0,f)=int and
>      type(op(2,f),name=range)
> then
>   print(`definite integral` = f);
>   f;
> else map(findintegral,f);
> fi
> end:
```

The procedure walks the input expression recursively. When it gets to a name of a variable, a number, or a Maple procedure, it returns because names and numbers are not integrals. If it sees a definite integral, it prints it and returns it. Otherwise it searches each operand of the input f for an integral recursively. The reason procedure is listed together with name and numeric is because procedures are the only object in Maple that you cannot map onto. So the search for definite integrals stops whenever it sees a name, number or Maple procedure. Let's try it out

```
> findintegral(f):
```

$$\mathit{definite\ integral} = \int_{-x}^{\infty} \mathrm{h}(t)\,dt$$

```
> findintegral(g):
```

$$\mathit{definite\ integral} = \int_{-x}^{0} \mathrm{h}(t)\,dt$$

$$\mathit{definite\ integral} = \int_{0}^{\infty} \mathrm{h}(t)\,dt$$

Now let's code the splitting action. Instead of just printing the integral, all we are going to do is to split it up. The rest of the program does not change.

```
> splitintegral := proc(f,c) local g,t,a,b;
> if type(f,{name,numeric,procedure}) then f
> elif type(f,function) and op(0,f)=int and
>      type(op(2,f),name=range) then
>    t := op([2,1],f);
>    a := op([2,2,1],f);
>    b := op([2,2,2],f);
>    g := op(1,f);
>    int(g,t=a..c) + int(g,t=c..b);
> else map(splitintegral,f,c)
> fi
> end:
```

```
> splitintegral(f,0);
```

$$x + \int_{-x}^{0} \mathrm{h}(t)\,dt + \int_{0}^{\infty} \mathrm{h}(t)\,dt$$

Our code has become quite a bit longer. We've used `op` to pick off the operands of the input as needed. This is a good example where the `typematch` command will shorten the code and also make the code more transparent. The above program is equivalent to this one

```
> splitintegral := proc(f,c) local g,t,a,b;
> if type(f,{name,numeric,procedure}) then f
> elif typematch(f,'int'(g::anything,
>        t::name=a::anything..b::anything))
> then int(g,t=a..c) + int(g,t=c..b);
> else map(splitintegral,f,c)
> fi
> end:
> splitintegral(f,0);
```

$$x + \int_{-x}^{0} \mathrm{h}(t)\,dt + \int_{0}^{\infty} \mathrm{h}(t)\,dt$$

Looks good so far. I wonder what subtleties we might have overlooked? My bet is that this program will be satisfactory for most purposes. But, here are some places where I think we might get into trouble. Firstly, a Maple detail. What if the input integral has more than two arguments? Maple integrals can have options, for example, the continuous option. The splitting fails in these cases

```
> f:=int(h(t),t=-x..infinity, continuous);
> splitintegral(f,0);
```

$$f := \mathrm{int}(\mathrm{h}(t),\, t = -x..\infty,\, \mathit{continuous})$$

$$\mathrm{int}(\mathrm{h}(t),\, t = -x..\infty,\, \mathit{continuous})$$

We have to be careful about the actual functionality that Maple provides. Reprogramming the code to handle an optional parameter is messy. The `type` and `typematch` commands do not support a value which can be there or not. So I test for each case separately.

```
> splitintegral := proc(f,c)
> local g,t,a,b,i,o;
> if type(f,{name,numeric,procedure}) then f
> elif typematch(f,'int'(g::anything,
>    t::name=a::anything..b::anything)) or
>       typematch(f,'int'(g::anything,
>    t::name=a::anything..b::anything,
>                          o::anything))
> then
>    if nops(f)=2 then o := NULL fi;
>    int(g,t=a..c,o) + int(g,t=c..b,o);
> else map(splitintegral,f,c)
> fi
> end:
```

```
> splitintegral(f,0);
```

$$\mathrm{int}(\mathrm{h}(t),\, t = -x..0,\, \mathit{continuous}) + \mathrm{int}(\mathrm{h}(t),\, t = 0..\infty,\, \mathit{continuous})$$

Secondly, what if $a = c$ or $b = c$? When I asked this question I supposed that if I applied splitintegral to g at 0 that I'd get a term $2 \int_0^0 \mathrm{h}(t)\,dt$. But to my surprise it didn't do that.

```
> splitintegral(g,0);
```

$$x + \int_{-x}^{0} \mathrm{h}(t)\,dt + \int_{0}^{\infty} \mathrm{h}(t)\,dt$$

The two new integrals were in fact created, but Maple simplified them to zero for us!

```
> int(h(t),t=0..0);
```

$$0$$

So this potential problem is resolved by Maple for us.

Third problem. I wonder if you noticed a more blatant mistake? Our program does not look for integrals inside the integrand or the limits of integration. It will not work on a multiple integral or nested integral. When you write these little procedures which walk through a formula looking for an object of a special type, it's easy to forget to look for nested objects. To do this we need to include the code

```
> g := splitintegral(g);
> a := splitintegral(a);
> b := splitintegral(b);
```

Getting the right form

$A\sin(\omega t) + B\cos(\omega t) = C\cos(\omega t + \phi)$

The solution to the second order equation below

```
> deq:=diff(y(t),t$2)-diff(y(t),t)+y(t)=0;
> sol:=dsolve({deq,y(0)=1,D(y)(0)=1},y(t));
```

$$deq := (\frac{\partial^2}{\partial t^2}\,\mathrm{y}(t)) - (\frac{\partial}{\partial t}\,\mathrm{y}(t)) + \mathrm{y}(t) = 0$$

$sol :=$

$$\mathrm{y}(t) = e^{(1/2\,t)}\cos(\frac{1}{2}\sqrt{3}\,t) + \frac{1}{3}\sqrt{3}\,e^{(1/2\,t)}\sin(\frac{1}{2}\sqrt{3}\,t)$$

is not expressed in the preferred engineering form $\mathrm{y}(t) = e^{(-k\,t)}\,C\cos(\omega\,t + \phi)$. No Maple command exists to put it in this form. In MapleTech Volume 3 No 3, Robert Lopez, in his Tips for Maple Instructors column, shows how to use Maple to determine what C and ϕ are. This is a very useful exercise for the student to make. What I want to do is show how to program Maple to effect the transformation $A\sin(x)+B\cos(x) = C\cos(x+\phi)$ where $C = \sqrt{A^2+B^2}$ and $\phi = \arctan(-A, B)$. These formulae can be determined by expanding

```
> A*sin(x)+B*cos(x)  = C*cos(x+phi);
```

$$A\sin(x) + B\cos(x) = C\cos(x+\phi)$$

```
> expand(");
```

$$A\sin(x) + B\cos(x) = C\cos(x)\cos(\phi) - C\sin(x)\sin(\phi)$$

and equating coefficients in $\sin(x)$ and $\cos(x)$ and solving the equations with Maple assuming A, B are both real. Our problem is complicated by the $e^{(\frac{t}{2})}$ common factor in this application. We don't want to put that inside the square-root only to have to simplify it out again. Let's have a go at it nevertheless. As in the previous examples, our first task is to locate a sine term and a cosine term in a sum with the same arguments, and to identify the coefficients present. I'm going to simplify the task by asking the user to specify the argument, $\frac{\sqrt{3}\,t}{2}$ in the above example. This is my first brute force attempt using nested for loops. I search the terms in a sum for a sine factor then a cosine factor with argument x.

```
> findsincos:=proc(f,x) local s,t,found,A,B;
> if type(f,{name,numeric,procedure})
> then f
> elif type(f,`+`) then
>   found := false;
>   for t in f while not found do
>     if t = sin(x)
>     then found := true; A := 1;
>     elif type(t,`*`) then
>       for s in t do
>         if s=sin(x) then found := true;
>            A := t/s; fi
>       od;
>     fi
>   od; print(found);
>   if not found then
>      RETURN( map(findsincos,f,x) )
>   fi;
>   found := false;
>   for t in f while not found do
>     if t = cos(x)
>     then found := true; B := 1;
>     elif type(t,`*`) then
>       for s in t do
>         if s=cos(x) then found := true;
>            B := t/s fi
>       od;
>     fi
>   od; print(found);
>   if not found then
>      RETURN( map(findsincos,f,x) )
>   fi;
>   f
> else map(findsincos,f,x)
> fi
> end:
```

```
> findsincos(sol,sqrt(3)*t/2);
```

$$true$$

$$true$$

$$\mathrm{y}(t) = e^{(1/2\,t)}\cos(\frac{1}{2}\sqrt{3}\,t) + \frac{1}{3}\sqrt{3}\,e^{(1/2\,t)}\sin(\frac{1}{2}\sqrt{3}\,t)$$

Well, despite the lengthy program, there is a bug. It will return true if one of the terms has $\sin(x) \times \cos(x)$ in it. If we find a sine term in the sum, we should not look at that term when looking for the cosine term. The program is also long. The search for the sine term duplicates the search for the cosine term, so we should be able to shorten the code. I did this but I was still not happy. Sometimes it pays to go for a coffee and think whether there might not be a simpler way to go about it. I looked at the `match` command in Maple but it it was not helpful. Perhaps you have asked yourself, why not just use `coeff`? Yes indeed. I didn't think of that at first. The simpler solution is to look for a non-zero coefficient of $\sin(x)$ and $\cos(x)$. Then the program is much simpler. Note the use the `degree` command below to check first that $\sin(x)$ appears linearly, and $\cos(x)$ also, but not a product of the two.

```
> findsincos:=proc(f,x) local s,t,found,A,B;
> if type(f,{name,numeric,procedure}) then f
> elif type(f,`+`)and degree(f,sin(x))=1 and
>        degree(f,cos(x))=1 and
>        degree(f,{sin(x),cos(x)})=1 then
>   A := coeff(f,sin(x),1);
>   B := coeff(f,cos(x),1);
>   f
> else map(findsincos,f,x)
> fi
> end:
```

Now that we have the coefficients, it's easy to effect the transformation that we want. The only detail is how to identify common factors present in the coefficients. This is actually not so simple. For one, x is a common factor between $a\,x + x$ and x^2. We need to compute the greatest common divisor of a general expression. Let's just code a simple direct comparison between the explicit factors present in the coefficient. This time, I'm not going to code it explicitly, I'm going to use Maple to do it for me by using the set operations.

```
> common := proc(f,g,newf,newg) local C,F,G;
> if type(f,`*`) then F := {op(f)}
>                 else F := {f} fi;
> if type(g,`*`) then G := {op(g)}
>                 else G := {g} fi;
> C := F intersect G; # factors in common
> newf := `*`( op(F minus C) ); # in F only
> newg := `*`( op(G minus C) ); # in G only
> RETURN( `*`( op(C) ) )
> end:
```

```
> common( exp(t/2), sqrt(3)/3*exp(t/2),
>          'A', 'B' );
> A, B;
```

$$e^{(1/2\,t)}$$

$$1, \frac{1}{3}\sqrt{3}$$

The final task is to put it all together and make the desired transformation.

```
> trigcombine := proc(f,x) local A,B,s,C,phi;
> if type(f,{name,numeric,procedure}) then f
> elif type(f,`+`) and degree(f,sin(x))=1 and
>      degree(f,cos(x))=1 and
>      degree(f,{sin(x),cos(x)})=1 then
>   A := coeff(f,sin(x));
>   B := coeff(f,cos(x));
>   s := common(A,B,'A','B');
>   C := sqrt(A^2+B^2);
>   phi := arctan(-A,B);
>   f - A*s*sin(x) - B*s*cos(x) +
>       s*C*cos(x+phi)
> else map(trigcombine,f,x)
> fi
> end:
> trigcombine(sol,sqrt(3)*t/2);
```

$$y(t) = \frac{2}{3}\,e^{(1/2\,t)}\,\sqrt{3}\sin(\frac{1}{2}\,\sqrt{3}\,t + \frac{1}{3}\,\pi)$$

Ugghhh! Maple has simplified the cosine to a sine on us. It has made the transformation $\cos(x - \frac{\pi}{6}) = \sin(x + \frac{\pi}{3})$ thinking that we'd prefer the result with a postive angle. Let's check it anyway.

```
> expand(");
```

$$y(t) = e^{(1/2\,t)}\cos(\frac{1}{2}\,\sqrt{3}\,t) + \frac{1}{3}\,\sqrt{3}\,e^{(1/2\,t)}\sin(\frac{1}{2}\,\sqrt{3}\,t)$$

Fixing the Maple library

Recently a user reported a bug in the code for solving linear systems modulo an integer. The bug is reproduced in the following. Since it is in fact my own bug, I have to apologize for it. Some kind soul located the code, fixed the error, and passed it on to the user. But you can install a fix to any Maple library routine relatively easily. I'd like to show you how to do that. In fact, being able to do this is one of the nicest features of Maple. It makes development of Maple painless, well, at least this aspect of it! Here is the bug (in Maple V Release 4)

```
> A:=matrix([
> [0,0,0,1,0,1],
> [1,1,1,0,0,0],
> [0,0,1,1,1,0],
> [0,0,0,1,1,1]]);
```

$$A := \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

```
> b := vector([1,1,1,1]);
```

$$b := [1, 1, 1, 1]$$

```
> x := Linsolve(A,b) mod 2;
```

$$x := [\_t_2 + \_t_6, \_t_2, \_t_6, \_t_6, 1, \_t_6]$$

Now the symbols _t2 and _t6 are arbitrary variables. They have been used to paramterize the solution space. They can take on any values in the field, in this case 0 or 1. We would expect at least two free variables in the solution because the matrix has only 4 rows but 6 columns. Now, if the solution is correct, $A\,x = b$ should hold modulo 2.

```
> evalm(A&*x-b);
```

$$[2\,\_t_6 - 1, 2\,\_t_2 + 2\,\_t_6 - 1, 2\,\_t_6, 2\,\_t_6]$$

```
> map(modp,",2);
```

$$[1, 1, 0, 0]$$

This should be the zero vector. To track down the bug you can first set printlevel := 1000; and reexecute the command. This will generate a trace (not displayed here because it's too long) indicating all Maple library procedures which are called.

You will see that the procedure `mod/Linsolve` is called. To see the code for this routine you may print it with

```
> print(`mod/Linsolve`);
```

proc(A, b, $rank$, v) ... **end**

The ... is the way the `print` command displays the body of a Maple procedure by default. To see the code in detail, do

```
> interface(verboseproc=2);
> print(`mod/Linsolve`);
```

Now you have to debug the routine. The bug is the $m + 1$ in the statement

```
x[j] := Normal((B[i,m+1]-t)*s) mod p;
```

The m should be an n. To make the correction, print the code into a file with the following

```
> writeto(Linsolve);
> print(`mod/Linsolve`);
> writeto(terminal);
```

Now edit the file. The first line should look like

> print(`mod/Linsolve`);proc(A, b, rank, v)

Fix it so that it reads

`mod/Linsolve` := proc(A, b, rank, v)

Now look at the end of the file. The last lines should look like

end
> appendto(terminal);

Fix it so that it is just

end;

so that the file contains valid Maple input. Now, make the correction(s) to the code. Next, read in the corrected code with

```
> read Linsolve;
```

The correction is now in Maple. Let us execute it

```
> x := Linsolve(A,b) mod 2;
```

$$x := [1 + \_t_2 + \_t_6,\ \_t_2,\ \_t_6,\ 1 + \_t_6,\ 0,\ \_t_6]$$

```
> evalm( A&*x-b );
```

$$[2\,\_t_6,\ 2\,\_t_2 + 2\,\_t_6,\ 2\,\_t_6,\ 2\,\_t_6]$$

```
> map(modp,",2);
```

$$[0, 0, 0, 0]$$

Note, in other cases you may not be able to get Maple to use your corrected code like this because of option remember. To have the new code used in your next Maple session do the following. First use the `save` command to create a ".m" file. This is Maple's internal format, the format used by the Maple library.

```
> save `mod/Linsolve`, `Linsolve.m`;
```

The file `Linsolve.m` contains the code for the `mod/Linsolve` procedure. It is not readable anymore. Now all that is left to do is to move the .m file to the appropriate place. If you want, you can update your copy of the Maple library. You will need to have write permissions on the Maple library to do this. If you are a single user on a personal computer, you may want to do this. First locate the directory of the Maple library on your computer. The libname variable specifies it.

```
> libname;
```

/maple/mapleR4/update, /maple/mapleR4/lib, /maple/mapleR4/lib/update, /maple/mapleR4/local/lib

The Maple library is under /maple/mapleR4/lib on my computer. Go to that directory. You should see the files

maple.hdb maple.ind maple.lib

in that directory. The file maple.lib contains the Maple library. It is a Maple library archive. One way to install the new code is to insert the new code into that library archive. Don't do this. If anything goes wrong you'll have to reinstall the library archive. Do the following which is simpler. Create the subdirectory mod and put the file Linsolve.m under that directory. Now when Maple searches this library, it will look for the file mod/Linsolve.m before searching the library archive maple.lib . Actually, Maple will first search the directory /maple/mapleR4/update because that directory is the first in the libname sequence. In fact, all the files under that directory are updates provided by the Maple company.

This is all valid if you are able to write to the directory /maple/maleR4/lib. If you are not able to write to that directory, you can still update Maple by creating your own library, putting the corrected code in your library, and arranging for Maple to search your library before it searchs the Maple library as follows. This is in fact how I develop Maple code. First create a directory for your Maple library. For example

/home/monagan/maple/lib

Now create the subdirectory mod and put the .m file Linsolve.m under that directory. That is the corrected code is in the file

/home/monagan/maple/lib/mod/Linsolve.m .

Next, start a fresh Maple session and execute the following command

```
> libname := `/home/monagan/maple/lib`,
>                   libname;
```

libname := /home/monagan/maple/lib, /maple/mapleR4/update, /maple/mapleR4/lib, /maple/mapleR4/lib/update, /maple/mapleR4/local/lib

This puts my library in front of the Maple library and the corrected code is now accessed directly. Put this assignment to the libname variable in your Maple initialization file so that it is done everytime you start Maple. That's it!