APPLICATIONS OF COMPUTER ALGEBRA

APPLICATIONS OF COMPUTER ALGEBRA

edited by

Richard Pavelle
Symbolics, Inc.

KLUWER ACADEMIC PUBLISHERS
Boston/Dordrecht/Lancaster

Distributors for North America:
Kluwer Academic Publishers
190 Old Derby Street
Hingham, MA 02043, U.S.A.

Distributors for all other countries:
Kluwer Academic Publishers Group
Distribution Centre
P.O. Box 322
3300 AH Dordrecht
The Netherlands

Consulting Editor: Jonathan Allen

Library of Congress Cataloging in Publication Data
Main entry under title:

Applications of computer algebra.

Papers based on the proceedings of a symposium on symbolic algebraic manipulation in scientific computation presented by the ACS Division of Computers in Chemistry at the 188th meeting of the American Chemical Society, Aug. 26–31, 1984.
Includes index.
1. Algebra—Data processing—Congresses. I. Pavelle, Richard. II. American Chemical Society. Division of Computers in Chemistry.
QA155.7.E4A67 1985 512′.0028′5 85-9760

ISBN-13: 978-1-4684-6890-8 e-ISBN-13: 978-1-4684-6888-5
DOI: 10.1007/978-1-4684-6888-5

Softcover reprint of the hardcover 1st edition 1985

Second printing, 1986

CONTENTS

CONTRIBUTORS

Professor Carl M. Bender
Department of Physics
Washington University
St. Louis, MO 63130

Dr. John T. Bendler
General Electric Corporate Research and Development
Schenectady, NY 12301

Professor R. Stephen Berry
Department of Chemistry
The James Franck Institute
5735 South Ellis Avenue
University of Chicago
Chicago, IL 60637

Dr. Robert H. Berman
Research Laboratory of Electronics
Massachusetts Institute of Technology
Cambridge, MA 02139

Dr. Walter Bloss
GTE Laboratories, Inc.
40 Sylvan Road
Waltham, MA 02154

Richard L. Brenner
Symbolics, Inc.
11 Cambridge Center
Cambridge, MA 02142

Dr. Gene Cooperman
GTE Laboratories, Inc.
Fundamental Research Laboratory
40 Sylvan Road
Waltham, MA 02154

Dr. William J. Frawley
GTE Laboratories, Inc.
Fundamental Research Laboratory
40 Sylvan Road
Waltham, MA 02154

Dr. Lionel Friedman
GTE Laboratories, Inc.
Fundamental Research Laboratory
40 Sylvan Road
Waltham, MA 02154

Mohammad Golnaraghi
Theoretical and Applied Mechanics
Cornell University
Ithaca, NY 14853

C. Gomez
INRIA
Domaine de Voluceau
Roquencourt B.P. 105
78153 LE CHESNAY Cedex
FRANCE

Professor W. H. Hui
Department of Applied Mathematics
University of Waterloo
Waterloo, Ontario, N2L 3G1
CANADA

Dr. M. A. Hussain
General Electric Corporate Research and Development
Schenectady, NY 12301

Dr. W. L. Keith
Theoretical and Applied Mechanics
Cornell University
Ithaca, NY 14853

Professor Jeffrey L. Krause
Department of Chemistry
The James Franck Institute
5735 South Ellis Avenue
University of Chicago
Chicago, IL 60637

Professor Donald R. McLaughlin
Department of Chemistry
University of New Mexico
Albuquerque, NM 87131

Professor F. C. Moon
Theoretical and Applied Mechanics
Cornell University
Ithaca, NY 14853

Professor B. Noble
Mathematics Research Center
University of Wisconsin
610 Walnut Street
Madison, Wisconsin 53705

Dr. Andrew M. Odlyzko
AT&T Bell Laboratories
Room 2C-370
Murray Hill, NJ 07974

Dr. Richard Pavelle
Symbolics, Inc.
11 Cambridge Center
Cambridge, MA 02142

J. P. Quadrat
INRIA
Domaine de Voluceau
Roquencourt B.P. 105
78153 LE CHESNAY Cedex
FRANCE

Professor Thomas E. Raidy
Department of Chemistry
University of South Carolina
Columbia, SC 29208

Professor R. H. Rand
Theoretical and Applied Mechanics
Cornell University
Ithaca, NY 14853

Dr. Patrick Roache
Ecodynamics Research Associates, Inc.
Post Office Box 8172
Albuquerque, NM 87198

Dr. Michael F. Shlesinger
Physics Division
Office of Naval Research
Arlington, VA 22217

Professor Stanly Steinberg
Department of Mathematics and Statistics
University of New Mexico
Albuquerque, NM 87131

A. Sulem
INRIA
Domaine de Voluceau
Roquencourt B.P. 105
78153 LE CHESNAY Cedex
FRANCE

Professor G. Tenti
Department of Applied Mathematics
University of Waterloo
Waterloo, Ontario, N2L 3G1
CANADA

Professor Carl Trindle
University of Virginia
Department of Chemistry
Charlottesville, VA 22901

Professor Paul S. Wang
Department of Mathematical Sciences
Kent State University
Kent, OH 44242

Professor Stanley J. Watowich
Department of Chemistry
The James Franck Institute
5735 South Ellis Avenue
University of Chicago
Chicago, IL 60637

PREFACE

Today, certain computer software systems exist which surpass the computational ability of researchers when their mathematical techniques are applied to many areas of science and engineering. These computer systems can perform a large portion of the calculations seen in mathematical analysis. Despite this massive power, thousands of people use these systems as a routine resource for everyday calculations. These software programs are commonly called "Computer Algebra" systems. They have names such as MACSYMA, MAPLE, muMATH, REDUCE and SMP. They are receiving credit as a computational aid with increasing regularity in articles in the scientific and engineering literature.

When most people think about computers and scientific research these days, they imagine a machine grinding away, processing numbers arithmetically. It is not generally realized that, for a number of years, computers have been performing non-numeric computations. This means, for example, that one inputs an equation and obtains a closed form analytic answer. It is these Computer Algebra systems, their capabilities, and applications which are the subject of the papers in this volume.

On August 26–31, 1984, the American Chemical Society held their 188th national meeting in Philadelphia. On August 26–27, 1984, the ACS Division of Computers in Chemistry held a symposium on Symbolic Algebraic Manipulation in Scientific Computation. This was the first symposium ever organized on applications of Computer Algebra. The symposium was broken into four sessions. The first session gave an introduction to Computer Algebra and explained the uses of these systems as opposed to numeric computation systems. Also included were discussions of the interface between algebraic and numeric systems as well as the application of algebraic systems to the field of mathematics. The second session mainly dealt with applications of Computer Algebra to the field of chemistry and chemical education. The chemistry community is the most recent major scientific group to discover the benefits of using Computer Algebra systems. The third session dealt with the engineering applications of Computer Algebra systems to problems in spectral analysis, robotics, finite element methods, and optimal control. The fourth session dealt primarily with applications of Computer Algebra to computations in physics and applied mathematics.

This volume provides a broad introduction to the capabilities of Computer Algebra systems and gives many examples of applications to real problems in engineering and the sciences. It is my hope that this information will create new users of Computer Algebra systems by showing what one might expect to gain by using them and what one will lose by not using them.

Richard Pavelle
Cambridge, MA

1

MACSYMA: CAPABILITIES AND APPLICATIONS TO PROBLEMS IN ENGINEERING AND THE SCIENCES

Abstract

MACSYMA™ is a large, interactive computer system designed to assist engineers, scientists, and mathematicians in solving mathematical problems. A user supplies symbolic inputs and MACSYMA yields symbolic, numeric or graphic results. This paper provides an introduction to MACSYMA and provides the motivation for using the system. Many examples are given of MACSYMA's capabilities with actual computer input and output. Also presented are several applications where MACSYMA has been employed to deal with problems in engineering and the sciences.

Richard Pavelle
Symbolics, Inc.
MACSYMA Group
11 Cambridge Center
Cambridge, MA 02142

1. INTRODUCTION

1.1 MACSYMA: A Personal Perspective

My purpose in this paper is to provide a broad introduction to the capabilities of MACSYMA and to give some examples of its applications to real problems in engineering and the sciences. It is my hope that this information will create new users of Computer Algebra systems by showing what one might expect to gain by using them and what one will lose by not using them.

MACSYMA output is used and CPU times are often given. In some cases I have modified the output slightly to make it more presentable. The CPU times correspond to a Symbolics 3600 and to the MACSYMA Consortium machine (MIT-MC) which is a Digital Equipment KL10. These are about equal in speed and about twice as fast as a Digital Equipment VAX 11/780 for MACSYMA computations. When CPU times are not given one may assume the calculation requires less than 10 CPU seconds.

I have been involved with the MACSYMA project since 1973. Between 1972 and 1973, as a postdoctoral research fellow at the University of Waterloo, I worked on a calculation (by hand of course) that took me about 3 months to complete. I did not submit it for publication because I was not sure the result was correct. I had been working in General Relativity and tensor analysis for several years and massive hand calculations were the normal means of getting a result. Indeed, even today, many mistakenly believe this is the only way to do analysis. In 1973 I joined a small firm outside of Boston to work on a DARPA contract on gravitation. The terms of the contract required us to build component and indicial tensorial capabilities into MACSYMA. At that time MACSYMA was in its early development, and I remember my first encounter with it vividly. I had heard about MACSYMA's algebraic capabilities but, not trusting computers, I thought it was trickery. I recall my amazement when, at that first encounter, one of the MIT hackers had MACSYMA compute the components of the Einstein tensor for a spherically symmetric metric in about 15 minutes of real time. It had a sign error that we later corrected, but it was fantastic to see a machine perform a complicated algebraic calculation that most experts would require several hours to accomplish. I was sold and since then I have been promoting MACSYMA. I also entered into an agreement with myself to never again spend more than 15 minutes on a hand calculation. This agreement is still in effect. Some time later I repeated my earlier hand calculation on MACSYMA, verified my computation in about 2 minutes, and published the paper [Pa1].

Computer Algebra systems can actually boost a person's ability to perform analysis and get useful or publishable results. I shall illustrate this with a personal experience. In 1974 Yang [Ya1] proposed a new gauge theory of gravitation that was meant to incorporate Einstein's General Relativity as a special case. The paper was published in Physical Review Letters, one of the premier journals in the sciences. Yang did not give any solutions to his equations and this seemed to me like a perfect opportunity to test MACSYMA. I knew that Yang's equations were identical, under certain conditions, to others proposed by Kilmister and Newman several years before and that they had studied them carefully [Ki1]. Nevertheless, even if Yang had some solutions and Kilmister and Newman had others, I believed that with MACSYMA I could quickly discover something that might have been missed. After two days of analysis with MACSYMA, I did find solutions that enabled me to criticize the theory on physical grounds [Pa2]. As a consequence I was fortunate to publish three papers in Physical Review Letters about Yang's theory [Pa3,Pa4], and I believe the criticisms I offered are generally accepted to be valid. One of the papers [Pa4] contains an exact solution of a partial differential equation of 4th degree in two variables containing 404 terms. This is the largest and most complex differential equation I have seen for which exact solutions have been found. Does the Guinness Book of World Records accept solutions of PDEs?

I have not used other Computer Algebra systems extensively because MACSYMA has provided all the tools I have needed. I shall try to give the reader ideas about the capabilities of Computer Algebra systems as a whole. However, one should bear in mind that not all of the capabilities of MACSYMA that I present reside on other systems.

Of the general-purpose systems [Pa5,va1,Yu1], MACSYMA offers the most features. It has been licensed to about 400 sites worldwide since 1982. REDUCE is the most widely distributed system. It has been licensed, over the last 20 years, to about 1000 sites worldwide, and its clones (a polite term for unlicensed programs) may be on several thousand additional machines. I do not know about the distribution of the other general systems. However, MAPLE (inexpensive) and SMP (very expensive) will likely have a distribution inversely proportional to their cost. muMATH has had a very large distribution; several thousand copies have been sold to the micro market. The distribution of the special-purpose systems, those that have a very restricted range of capabilities, has been limited.

Computer Algebra systems are very far from saturating the user base. I estimate that perhaps only 20% of the people who need these systems are even aware of their existence and less than 1/4 of these actually use them. I gave a MACSYMA talk at MIT about three years to two engineering departments. Only half of the 50 attendees had even heard of MACSYMA. It is clear that the field of Computer Algebra needs a good deal of public relations.

1.2 What is MACSYMA

The development of MACSYMA began at MIT in the late 60s, and its history has been described elsewhere [Mo1]. A few facts worth repeating are that a great deal of effort and expense went into MACSYMA. There are estimates that 100 man-years of developing and debugging have gone into the program. While this is a large number, let us consider the even larger number of man-years using and testing MACSYMA. At MIT, between 1972 and 1982, we had about 1000 MACSYMA users. If we had 50 serious users using MACSYMA for 50% of their time, 250 casual users at 10% and 700 infrequent users at 2% then the total is over 600 man-years. MACSYMA has been at 50 sites for 4 years and at 400 sites today. Well, we can conclude that at least 1000 man-years have been spent in using MACSYMA. MACSYMA is now very large and consists of about 3000 lisp subroutines or about 300,000 lines of compiled lisp code joined together in one giant package for performing symbolic mathematics.

These huge numbers show why it is difficult to construct new general purpose Computer Algebra systems. A new system must be very large if it is to have many capabilities. Given the technology of the 1980's, it will still require a substantial fraction of the time it took to build MACSYMA, a minimum of 1/5th say, to construct, test and debug a system with MACSYMA's capabilities. One would be left with a system perhaps 1/4 to 1/2 the size. Perhaps the most reasonable approach is to build upon the current knowledge base rather than starting a new system from the ground up.

While this paper is directed towards MACSYMA, the development of MACSYMA and other Computer Algebra systems has really been the result of an international effort. There are many systems, world-wide, of various sizes and designs which have been developed over the past fifteen to twenty years [val]. Research related to the development of these systems has lead to many new results in mathematics and the construction of new algorithms. These results in turn helped the development of MACSYMA as well as other systems.

1.3 Why MACSYMA is Useful or Necessary

Here are some of the more important reasons for using MACSYMA:

1. The answers one obtains are exact and can often be checked by independent procedures. For example, one can compute an indefinite integral and check the answer by differentiating; the differentiation algorithm is independent of the integration algorithm. Since exact answers are given, the statistical error analysis associated with numerical computation is unnecessary. One obtains answers that are reliable to a high level of confidence.

2. The user can generate FORTRAN expressions that allow numeric computers to run faster and more efficiently. This saves CPU cycles and makes computing more economical. The user can generate FORTRAN expressions from MACSYMA expressions. The FORTRAN capability is an extremely important feature combining symbolic and numeric capabilities. The trend is clear, and in a few years we will have powerful, inexpensive desktop or notebook computers that merge the symbolic, numeric and graphic capabilities in a scientific workstation.

3. The user can explore extremely complex problems that cannot be solved in any other manner. This capability is often thought of as the major use of Computer Algebra systems. However, one should not lose sight of the fact that MACSYMA is more often used as an advanced calculator to perform everyday symbolic and numeric problems. It also complements conventional tools such as reference tables or numeric processors.

4. A great deal of knowledge has gone into the MACSYMA knowledge base. Therefore the user has access to mathematical techniques that are not available from any other resources, and the user can solve problems even though he may not know or understand the techniques that the system uses to arrive at an answer.

5. A user can test mathematical conjectures easily and painlessly. One frequently encounters mathematical results in the literature and questions their validity. Often MACSYMA can be used to check these results using algebraic or numeric techniques or a combination of these. Similarly one can use the system to show that some problems do not have a solution.

6. MACSYMA is easy to use. Individuals without prior computing experience can learn to solve fairly difficult problems with MACSYMA in a few hours or less. While MACSYMA is written in a dialect of LISP, the user need never see this base language. MACSYMA itself is a full programming language, almost mathematical in nature, whose syntax resembles versions of ALGOL.

There are two additional reasons for using MACSYMA that are more important than the others.

7. One can concentrate on the intellectual content of a problem leaving computational details to the computer. This often results in accidental discoveries and, owing to power of the program, these occur at a far greater rate than when calculations are done by hand.

8. But the most important reason is that, to quote R. W. Hamming, "The purpose of computing is insight, not numbers." This exemplifies the major benefit of using MACSYMA, and I will demonstrate the validity of this statement by showing not only how one gains insight but also how one uses MACSYMA for theory building. However, a second quotation reputed to be by Hamming is correct as well, namely that "The purpose of computing is not yet in sight."

In the chapter on applications I shall illustrate each of these points with specific examples.

2. Capabilities and Uses of MACSYMA

2.1 Capabilities of MACSYMA

It is not possible to fully indicate the capabilities of MACSYMA in a few lines since the reference manual itself occupies more than 500 pages [MA1]. However, some of the more important capabilities include (in addition to the basic arithmetical operations) facilities to provide analytical tools for

LIMITS
DERIVATIVES
INDEFINITE INTEGRATION
DEFINITE INTEGRATION
ORDINARY DIFFERENTIAL EQUATIONS
SYSTEMS OF EQUATIONS (NON-LINEAR)
SIMPLIFICATION
FACTORIZATION
TAYLOR SERIES (SEVERAL VARIABLES)
POISSON SERIES
LAPLACE TRANSFORMATIONS
INDEFINITE SUMMATION
MATRIX MANIPULATION
VECTOR MANIPULATION
TENSOR MANIPULATION
FORTRAN GENERATION

There are other routines for calculations in number theory, combinatorics, continued fractions, set theory and complex arithmetic. There is also a share library currently containing about 80 subroutines. Some of these perform computations such as asymptotic analysis and optimization while others manipulate many of the higher transcendental functions. In addition one can evaluate expressions numerically, to arbitrary precision, at most stages of a computation. MACSYMA also provides extensive graphic capabilities to the user.

To put the capabilities of MACSYMA in perspective we could say that MACSYMA knows a large percentage of the mathematical techniques used in engineering and the sciences. I do not mean to imply that MACSYMA can do everything. It is easy to come up with examples that MACSYMA cannot handle, and I will present some of these. Perhaps the following quotation will add the necessary balance. It is an exit message from some MIT computers that often flashes on our screens when logging out. It states: "I am a computer. I am dumber than any human and smarter than any administrator." MACSYMA is remarkable in both the questions it can and cannot answer. It will be many years before it evolves into a system that rivals the human in more than a few areas. But until then, it is the most useful tool that any engineer or scientist can have at his disposal.

2.2 Uses of MACSYMA

It is difficult to list the application fields of MACSYMA because users often do not describe the tools that helped them perform their research. However, from Computer Algebra conferences [MUC1,MUC2,MUC3] and publications in the open literature we do know that MACSYMA has been used in the following fields:

ACOUSTICS
ALGEBRAIC GEOMETRY
ANTENNA THEORY
CELESTIAL MECHANICS
COMPUTER-AIDED DESIGN
CONTROL THEORY
DEFORMATION ANALYSIS
ECONOMETRICS
EXPERIMENTAL MATHEMATICS
FLUID DYNAMICS
GENERAL RELATIVITY
NUMBER THEORY
NUMERICAL ANALYSIS
PARTICLE PHYSICS
PLASMA PHYSICS
SOLID-STATE PHYSICS
STRUCTURAL MECHANICS
THERMODYNAMICS

Researchers have reported using MACSYMA to explore problems in:

AIRFOIL DESIGN
ATOMIC SCATTERING CROSS SECTIONS
BALLISTIC MISSILE DEFENSE SYSTEMS
DECISION ANALYSIS IN MEDICINE
ELECTRON MICROSCOPE DESIGN
EMULSION CHEMISTRY
FINITE ELEMENT ANALYSIS
GENETIC STUDIES OF FAMILY RESEMBLANCE
HELICOPTER BLADE MOTION
LARGE SCALE INTEGRATED CIRCUIT DESIGN
MAXIMUM LIKELIHOOD ESTIMATION
NUCLEAR MAGNETIC RESONANCE
OPTIMAL CONTROL THEORY
POLYMER MODELING
PROPELLER DESIGN
RESOLVING CLOSELY SPACED OPTICAL TARGETS
ROBOTICS
SHIP HULL DESIGN
SPECTRAL ANALYSIS
UNDERWATER SHOCK WAVES

3. Examples of MACSYMA

3.1 Polynomial Equations

Here is an elementary example that demonstrates the ability of MACSYMA to solve equations. In MACSYMA, as with most systems, one has user input lines and computer output lines. Below, in the input line (C1), we have written an expression in an ALGOL like syntax, terminated it with a semi-colon, and in (D1) the computer displays the expression in a two dimensional format in a form similar to hand notation. Terminating an input string with $ inhibits the display of the D lines.

```
(C1) X^3+B*X^2+A^2*X^2-9*A*X^2+A^2*B*X-2*A*B*X-

                9*A^3*X+14*A^2*X-2*A^3*B+14*A^4=0;


         3      2    2 2        2   2                 3
(D1) X   + B X  + A  X  - 9 A X  + A  B X - 2 A B X - 9 A  X

                                    2       3      4
                           + 14 A  X - 2 A  B + 14 A  = 0
```

In (C2) we now ask MACSYMA to solve the expression (D1) for X and the three roots appear in a list in (D2).

```
(C2) SOLVE(D1,X);

                                        2
(D2)             [X = 7 A - B,   X = - A ,   X = 2 A]
```

Notice that MACSYMA has obtained the roots analytically and that numeric approximations have not been made. This demonstrates a fundamental difference between a Computer Algebra system and an ordinary numeric equation solver, namely the ability to obtain a solution without approximations. I could have given MACSYMA a "numeric" cubic equation in X by specifying numeric values for A and B. MACSYMA then would have solved the equation and given the numeric roots approximately or exactly depending upon the specified command.

MACSYMA can also solve quadratic, cubic and quartic equations as well as some classes of higher degree equations. However, it obviously cannot solve equations analytically in closed form when methods are not known, eg. a general fifth degree equation or one of higher degree. It can however, find approximate real solutions of polynomial equations of any degree to arbitrary precision.

3.2 Differential Calculus

MACSYMA knows about calculus. In (D1) we have an exponentiated function that is often used as an example in a first course in differential calculus.

```
                                     X
                                    X
(D1)                               X
```

We now ask MACSYMA to differentiate (D1) with respect to X, using the DIFF command, to obtain this classic textbook result of differentiation. Notice how fast, 3/100 CPU seconds, MACSYMA computes this derivative.

```
(C2) DIFF(D1,X);
Time= 30 msec.

                  X
                 X   X                          X - 1
(D2)            X   (X  LOG(X) (LOG(X) + 1) + X      )
```

Below is a more complicated function, the error function of the tangent of the arc-cosine of the natural logarithm of X. Notice that MACSYMA does not display the identical input. This is because the input in (C1) passes through MACSYMA's simplifier. MACSYMA recognizes that the tangent of the arc-cosine of a function satisfies a trigonometric identity, namely TAN(ACOS(X)) = SQRT(1-X^2)/X. It takes this into account before displaying (D1).

```
(C1) ERF(TAN(ACOS(LOG(X))));
                                        2
                           SQRT(1 - LOG (X))
(D1)                   ERF(-----------------)
                                LOG(X)
```

Now when MACSYMA is asked to differentiate (D1) with respect to X, it does so in a straightforward manner and simplifies the result using the canonical rational simplifier RATSIMP. This command puts the expression in a numerator-over-denominator form canceling any common divisors. In (D2) the symbols %E and %PI are MACSYMA's representations for *e* (the base of the natural logarithms) and π.

```
(C2) DIFF(D1,X),RATSIMP;
Time= 1585 msec.
                                        1
                                  1 - -------
                                         2
                                      LOG (X)
                             2 %E
(D2)          - ------------------------------------
                               2                  2
               SQRT(%PI) X LOG (X) SQRT(1 - LOG (X))
```

3.3 Trigonometry

MACSYMA knows about elementary mathematics. For example, it can manipulate expressions containing multiple angles of trigonometric functions. In (D1) below is an expression that does not simplify in any obvious way. First we use the TRIGEXPAND command on (D1) which expands out trigonometric (hyperbolic) functions of sums of angles and of multiple angles resulting in (D2). We then EXPAND out (D2) and find that (D2) vanishes identically.

This is an example showing how to use MACSYMA to do "experimental mathematics". I suspected that (D1) vanished but trying to verify this by hand was both time consuming and error prone because of the multitude of signs. This identity does not appear to be known. It would be interesting if someone could find a geometrical interpretation of it.

```
(D1) SIN(Y - X) SIN(Z - X) SIN(Z - Y) + COS(Y - X) COS(Z - X) SIN(Z - Y)

   - COS(Y - X) SIN(Z - X) COS(Z - Y) + SIN(Y - X) COS(Z - X) COS(Z - Y)

(C2) TRIGEXPAND(D1);
Time= 1400.0 msec.

(D2) (COS(X) SIN(Y) - SIN(X) COS(Y)) (SIN(X) SIN(Z) + COS(X) COS(Z))

     (SIN(Y) SIN(Z) + COS(Y) COS(Z)) - (SIN(X) SIN(Y) + COS(X) COS(Y))

     (COS(X) SIN(Z) - SIN(X) COS(Z)) (SIN(Y) SIN(Z) + COS(Y) COS(Z))

     + (SIN(X) SIN(Y) + COS(X) COS(Y)) (SIN(X) SIN(Z) + COS(X) COS(Z))

     (COS(Y) SIN(Z) - SIN(Y) COS(Z)) + (COS(X) SIN(Y) - SIN(X) COS(Y))

     (COS(X) SIN(Z) - SIN(X) COS(Z)) (COS(Y) SIN(Z) - SIN(Y) COS(Z))

(C3) EXPAND(D2);
Time= 1350.0 msec.

(D4)                                  0
```

MACSYMA can also go back the other way, so to speak, beginning with an expression such as (D5) and finding an equivalent form for it in terms of multiple angles using the TRIGREDUCE command.

```
                 5          2        3            4
(D5)      2 SIN (X) - 8 COS (X) SIN (X) + 6 COS (X) SIN(X)

(C6) EXPAND(TRIGREDUCE(D5));

(D6)                        SIN(5 X) + SIN(X)
```

3.4 Very Large Problems

A frequent question asks the size of the largest problem that MACSYMA can solve. The size of a problem that can be handled is, of course, to a large extent a function of the underlying computer hardware. I shall go into this question in more depth in Chapter 7. Here is a preliminary answer; here is a very large problem:

```
                        100          100          700
(D1)             (X - 1)     (2 X - 1)     (3 X - 1)
```

This is a polynomial in X and if one were to expand it out without collecting terms one would have

```
(C2) NTERMS(D1);

(D2)                          7150901
```

NTERMS(*expr*) gives the number of terms that *expr* would have if it were fully expanded out and no cancellations or combination of terms occurred. This number, in (D2), is precisely 101^2*701 which one would expect from the binomial expansion formula. We now ask MACSYMA to expand the expression. The command RATEXPAND uses a very efficient algorithm for expanding polynomials. After waiting about 13 CPU minutes MACSYMA returns:

```
(C3) RATEXPAND(D1);
Time = 764150 msec.

(D3)  122427186804066208416145044327206249704010297
      794616687779023569766751053609544125677306330
      282403068164313282073928984176895647340305797
      701184895851976317431643022541219564575431932
      361445842322265611989432234712383670344350435
      476671159706350681167253578453360950549918549
      395826481540244545237495806672865354745228669           900
      444407778719348014578179086676206584907014351093 76 X        + ...

                          3            2
      ...  - 2295846600 X   + 2876600 X   - 2400 X + 1
```

For obvious reasons I have included only the first term and the last 4 terms of (D3). Notice the leading coefficient of X^900 in (D3). MACSYMA can manipulate integers of arbitrary size, and floating point numbers to arbitrary precision. In fact this very large integer is exactly 2^100*3^700 which is also expected from the binomial expansion.

3.5 Factorization

MACSYMA can factor expressions. Below is a multivariate polynomial in four variables.

```
                 2  7  4  8      2  6  3  8      3  7  4  6
(D1) - 36 W  X  Y  Z  + 3 W  X  Y  Z  - 24 W  X  Y  Z

        3  6  3  6       2  8  6  5       4  7  6  5
 + 2 W  X  Y  Z  + 96 W  X  Y  Z  - 168 W  X  Y  Z

         2  7  6  5        2  10  5  5      2  7  5  5     7  5  5
 + 12 W  X  Y  Z  - 216 W  X   Y  Z  - 8 W  X  Y  Z  + 9 X  Y  Z

         4  6  5  5    2  6  5  5       2  9  4  5      7  3  5
 + 14 W  X  Y  Z  - W  X  Y  Z  + 18 W  X  Y  Z  + 87 X  Y  Z

        2  6  3  5       7  5  3        7  3  3     3  6  3  3
 - 3 W  X  Y  Z  + 6 W X  Y  Z  + 58 W X  Y  Z  - 2 W  X  Y  Z

         8  7  2        2  7  7  2      7  7  2       10  6  2
 - 24 X  Y  Z  + 42 W  X  Y  Z  - 3 X  Y  Z  + 54 X   Y  Z

          8  5  2         2  7  5  2       7  5  2       4  6  5  2
 - 232 X  Y  Z  + 414 W  X  Y  Z  - 29 X  Y  Z  - 14 W  X  Y  Z

      2  6  5  2        10  4  2       2  9  4  2
 + W  X  Y  Z  + 522 X   Y  Z  - 18 W  X  Y  Z
```

We now call the function FACTOR on (D1) and

```
(C2) FACTOR(D1);
Time= 111998 msec.

           6  3  2     3                 2       2  2    2       3
(D2) - X  Y  Z  (3 Z  + 2 W Z - 8 X Y  + 14 W  Y  - Y  + 18 X  Y)

                                       2      3     2  3       2          2
                               (12 W  X Y Z  - W  Z  - 3 X Y  - 29 X + W )
```

MACSYMA factors this massive expression in about 2 CPU minutes. One can also extend the domain of factorization to the Gaussian integers or other algebraic fields [Wal].

The time required to factor expressions on a particular machine depends strongly upon the number of variables and the degree of the exponents. It is a simple matter to construct examples that look simple yet that take inordinate amounts of time to factor. For example X^2026 - X^1013 + 1 factors into two parts. One factor is X^2 - X + 1 but the second is an irreducible polynomial with 1351 terms. The current factorization algorithms do not return after several CPU hours on this example. It is possible to enhance the factorization program to run faster on this