Chapter 1 Introduction of Computer Algebra

Introduction

- What is computer algebra?
- Why do algebraic calculations?
- Applications of computer algebra
- Chronology of computer algebra systems

What is computer Algebra?

" Computer Algebra is that part of computer science which designs, analyzes, implements, and applies algebraic algorithm"

R. Loos, "Computer Algebra: Symbolic and Algebraic Computation(1982)".

Comparison of numerical to algebraic computation

Numeric	Symbolic
$2/6 \longrightarrow 0.3333$	$2/6 \longrightarrow 1/3$
$2+3 \longrightarrow 5$	$x+2x \longrightarrow 3x$
$\cos(3.14159) \longrightarrow -0.99999$	$\cos(\pi) \longrightarrow -1$
$\int_0^{1/2} \frac{x}{x^2-1} d\,x \longrightarrow 0.1438$	$\int \frac{x}{x^2-1} \longrightarrow \frac{\ln\lvert x^2-1\rvert}{2}$
	$a^2-b^2 \longrightarrow (a+b)(a-b)$
numerical evaluation	algebraic simplification

Numeric	Symbolic
normally inexact	always exact
usually produces some sort of answer (may suffer loss of accuracy, fail to converge)	frequently cannot produce an answer (problem is intractable, expression swell)
arithmetic relatively cheap	arithmetic often expensive
zeroes usually not given special treatment except to avoid computational singularities	zeroes can greatly simplify calculations by making some of them unnecessary
zero detection: number sufficiently small?	zero detection: expression equivalent to zero?
maximize size of pivots to promote stability	minimize complexity of pivots to stunt expression growth

Typical features of algebraic computation

1. Computation with arbitrary precision numbers–no rounding
2. Computation with symbols and variables (e.g., x, y)
3. Computation with functions (e.g.,$\sin, \cos, f$)
4. Manipulation of formulas
5. Symbolic operations (differentiation, integration, factorization)

Why do algebraic calculations?

1. In many research fields, one often needs to process very large algebraic expressions (of perhaps hundreds of thousands or more terms, e.g., collision integrals, computations from the general theory of relativity, celestial mechanics calculations) or perform long analytical operations

2. Compared to a human, a computer does not make errors (assuming the programming is correct! This unfortunately is not always the case ...)

3. There exist algorithms which cannot easily be performed by a human with pencil and paper (e.g., factorization, quantifier elimination)

4. Algebraic solutions are usually more compact than a set of numerical solutions; algebraic solution gives more direct information about the relationship between the variables then figures

5. Algebraic solutions are always exact, numerical solutions will normally be approximations; this can arise from rounding and truncation errors, further errors can creep in when the user

interpolates data given in tabular form

6. Computer algebra can save both time and effort in solving a wide range of problems; much larger problems can be investigated than by using traditional methods

7. Computer algebra reduces the need for tables of functions, series and integrals; symbolic computation using computers have highlited many errors in such materials

8. Traditional teaching of applied mathematics has to involve much time in teaching techniques of solution; computer algebra systems tend to produce solutions quickly and without errors, so they

enable more time to be devoted to studying the properties of the solution

9. Using of computer algebra allows also very effective construction of numerical algorithms and their semiautomatic programming by code generation, the effectiveness of the work and reliability of the results can be strongly increased

Why use approximate solutions at all if we can have exact solutions?

1. An exact solution is not possible (for certain ordinary differential equations, one can prove that no closed form solution is possible)

2. Efficiency

Applications of Computer Algebra

1. Mathematics (abstract algebra, game theory, logic, PDEs)
2. Education (Economics,case studies, quantum mechanics,)
3. Engineering (chemical, control, electrical, signal processing)
4. Science (biology, physics)
5. Computer Science (cryptography, graph theory, error correction)

History

1. The field originated from the needs of physics researchers
2. The first programs dealing with formulas were written by physicists in order to save them from performing long, tedious, error prone calculations by hand
3. 1955 - first programs for formula derivation
4. 1965 - first general purpose computer systems working with algebraic expressions

5. 1975 - new research field with its own conferences, journals, etc.

6. 1990 - general spreading of integrated mathematical systems into almost all branches of science

Chronology of computer algebra systems

Three goals for research and development of computer algebra

1. wide functionality

2. easy of use

3. speed

Specialized systems

- TRIGMAN 1970, celestial mechanics
- SCHOONSCHIP 1971, quantum physics (1999, Nobel prize on physics)
- CAMAL 1975, celestial mechanics, general theory of relativity
- SHEEP 1977, general theory of relativity, very fast

General purpose systems

- REDUCE 1968, classical system, delivered with sources
- MACSYMA 1970, classical system, mainly in USA
- MAPLE 1985, fast and good system
- DERIVE 1988, on PC, even on pocket computers
- MATHEMATICA 1988, nice graphics and user interface (notebook)

- AXIOM 1991, former Scratchpad II, modern "object oriented" concepts, most complete system

System name	Year	Related systems	e-mail address
ALPAK	1964	ALTRAN	Bell Labs
ALTRAN	1968		Bell Labs
FORMAC			IBM
MATHLAB	1968	MACSYMA	DEC
REDUCE	1968		reduce-netlib@rand.org
MACSYMA	1970		macsyma-service@symbolics.com
MuMath	1979	Derive	
VAXIMA	1980		
SMP	1982	Mathematica	NOT ON MARKET
Maple	1985		wmsi@daisy.waterloo.edu
MathCAD	1985	Mathcad	
Derive	1988		Soft Warehouse Inc.
Mathematica	1988		info@wri.com
Theorist	1988		Prescience Corp
PARI	1988		ftp to math.ucla.edu
Mathcad	1991		1-800-MATHCAD
SymbMath	1991		chen@deakin.oz.au
Axiom	1991		

Mathematical computing on the Internet

1. NetSolve (by University of Tennessee's Innovative Computing Laboratory,1999) is a client-serve system that enables users to solve complex scientific problems remotely. The system allows users to access both hardware and software computational resources distributed across a network. NetSolve searches for computational resources on a network, choose the best one available. Using retry for fault-tolerance solves a problem and returns the answers to the user.

2. MathML (by W3C world wide web consortium, 1998) MathML is an XML application for describing mathematical notation and capturing both its structure and content. The goal of MathML is to enable mathematics to be served, received, and processed on the world wide web just as HTML has enabled this functionality for text.

3. OpenMath (Helsinki, Finland, 2000) is emerging standard for representing mathematical objects with their semantics, allowing them to be exchanged between computer programs, stored in databases, or publised on the worldwide web.

Five parts in the book

1. Euclid: Euclid algorithm for calculating the gcd and presents the subresultant theory for polynomials, modular algorithms, continued functions, Diophantine approximation, the Chinese Remainder Algorithm, secret sharing, and the decoding of BCH codes.

2. Newton: basis of fast arithmetics. FFT-based multiplication, division with remainder and polynomial equation solving vie

Newton iteration, and fast methods for the Eucliden algorithm and the solution of system of linear equations.

3. Gauss: We start with univariate factorization over finite fields, then we will discuss polynomials with rational coefficients. Two basis algorithmic ingredients are Hensel lifing and short vectors in lattices.

4. Fermat: Primality testing and factorization of integers

5. Hilbert Gröbner basis, Ritt-Wu method, symbolic integration, symbolic summation.