The computer algebra system SIMATH

Preliminary remarks

Since 1985, SIMATH is developed by the research group of Prof. Dr. H.G. Zimmer at the Universitaet des Saarlandes in Saarbruecken (Germany), partially supported by the Siemens AG/Munich.

The main area of application is **algebraic number theory**.

SIMATH is written **in C**, thus it is not necessary to learn a new programming language.

SIMATH is an **open system**, the **sources are being left open**, i.e. it is easy to add the user's own algorithm at any point within the system.

Up to now SIMATH runs on the following machines:

SUN 3 and SUN SPARCstations
Apollo DN 3000, DN 4500 and DN 10000
Solbourne 5E/900
SGI Challenge and Indigo
HP 9000 series 7xx
Linux (3/86 and 4/86) .

The latest version 3.9 of SIMATH may be obtained by **anonymous ftp** from ftp.math.uni-sb.de (134.96.32.23) in /pub/simath.
The installation of SIMATH is performed by makefiles and shellscripts.

If you have any problems or suggestions, please contact us by mail:

SIMATH-Gruppe
Lehrstuhl Prof. Dr. H.G. Zimmer
FB 9 Mathematik
Universitaet des Saarlandes
Postfach 151150
D–66041 Saarbruecken

or by e-mail: **simath@math.uni-sb.de**
or by phone: 0681/302-2206.
www: **http://emmy.math.uni-sb.de/~simath/**

The main parts of SIMATH

- SIMATH libraries containing the SIMATH procedures, i.e. C functions performing the memory administration or solving problems in algebraic number theory
- the SIMATH shell **SM**, a link between the operating system and SIMATH (In **SM** it is easy to edit, to compile and to administer programs and library archives.)
- the **keyword index** and the **online documentation**
- the **interactive calculator simcalc**, which makes most of the existing algorithms available in the course of the dialogue
- include files and C functions which are used by the **SM** (e.g. the SIMATH preprocessor)
- the **memory administration** which saves time and space because of the **automatic garbage collector** and which is compatible with C functions like malloc and free and with various computer architectures.

simcalc

The calculator **simcalc** is a user interface for solving problems to your specific need. It enables you to perform calculations in an extensive range and allows you the use of standard mathematical notation in a fully interactive environment.

simcalc handles calculations in

- $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$ (with arbitrary precision), $\mathbb{C}$, $\mathbb{Z}/m\mathbb{Z}$, algebraic number fields $\mathbb{Q}(\alpha)$ and Galois-fields $\mathbb{F}_{p^n}$,
- $\mathbb{Z}[x_1,\ldots,x_n]$, $\mathbb{Q}[x_1,\ldots,x_n]$, $\mathbb{Z}/m\mathbb{Z}[x_1,\ldots,x_n]$, $\mathbb{Q}(\alpha)[x_1,\ldots,x_n]$, $\mathbb{F}_{p^n}[x_1,\ldots,x_n]$ and $\mathbb{Q}(x_1,\ldots,x_n)$, $\mathbb{R}[x_1,\ldots,x_n]$ and $\mathbb{C}[x_1,\ldots,x_n]$,
- matrices and vectors over all these structures,
- elliptic curves (and their points) over $\mathbb{Q}$, $\mathbb{F}_p$, $\mathbb{Q}(\alpha)$ and $\mathbb{F}_{2^n}$.

simcalc is easy to use because of its built-in system facilities, e.g.

- user-defined functions
- loop constructions and if-statements
- substitution of variables in polynomial structures
- extensive and comprehensive on-line help
- complete set of on-line documentation
- input errors are intercepted by self-explanatory error messages
- you can edit your input line and use the history with the usual keys of emacs
- you can use arrays as variable names
- variable store with the possibility to list it entirely or partly and to delete in it
- overwrite protection that can be switched on and off
- predefinitions by .simcalcrc
- data input from files
- data output on files
- statistical functions
- you can interrupt an output or a computation
- you are allowed to enter shell-commands and to branch into a subshell.

The algorithms in SIMATH

There are modified input/output functions for mathematical objects, e.g. rationals, polynomials, matrices.

Arithmetic

- multiple precision arithmetic over $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$, $\mathbb{C}$, $\mathbb{Z}/m\mathbb{Z}$, $\mathbb{F}_{p^n}$, $\mathbb{F}_{2^n}$ (bitwise implementation), $\mathbb{Q}(\alpha)$, $\mathbb{Q}_p$, $\mathbb{Q}(x_1, \ldots, x_r)$ and $\mathbb{F}_p(x)$
- primality testing over $\mathbb{Z}$ (Goldwasser-Kilian-Atkin), construction of prime numbers
- computation of the order of elements in the multiplicative group of $\mathbb{Z}/m\mathbb{Z}$
- discriminant of number fields $\mathbb{Q}(\alpha)$
- decomposition law for primes in $\mathbb{Q}(\alpha)$
- extensions of p–adic valuations of $\mathbb{Q}(\alpha)$
- LLL-reduction for lattices in $\mathbb{Q}^n$
- determining the minimal polynomials for elements of a number field or an algebraic congruence function field
- ROUND 4 algorithm (Ford/Zassenhaus) for determining integral bases in number fields or in algebraic congruence function fields
- divisor and ideal class number in quadratic congruence function fields
- optimized continued fractions algorithm for determining fundamental units and regulators in real quadratic congruence function fields
- optimized baby step - giant step - algorithm for determing regulators in real quadratic congruence function fields
- relative class numbers of abelian number fields with odd prime power conductor.

Elliptic Curves

- arithmetic for elliptic curves over $\mathbb{Q}$, $\mathbb{Q}(\alpha)$, $\mathbb{F}_p$ and $\mathbb{F}_{2^n}$
- special concept for elliptic curves (An elliptic curve is a list of 4 lists (L_1, L_2, L_3, L_4), where L_1 contains the datas of the actual model, L_2 of the minimal model, L_3 of the short Weierstraß normal form and L_4 the invariants of the elliptic curve. All datas once computed are stored so that if they are again required they are read out of the lists.)

- structure and generators of the torsion group of an elliptic curve over $\mathbb{Q}$
- rank and basis of the Mordell-Weil group of an elliptic curve over $\mathbb{Q}$
- regulator and order of the Tate-Shafarevich group of an elliptic curve over $\mathbb{Q}$
- L-series and its derivations at $s = 1$ of an elliptic curve over $\mathbb{Q}$
- determination of all integral points of an elliptic curve over $\mathbb{Q}$
- construction of elliptic curves with a group of rational points with given isomorphism type
- global minimal model of elliptic curves over $\mathbb{Q}$ due to Laska
- conductor, reduction type and local minimal model of elliptic curves over $\mathbb{Q}$ and $\mathbb{Q}(\sqrt{d})$ due to Tate
- Néron-Tate height for elliptic curves over $\mathbb{Q}$
- determining the number of rational points of elliptic curves over $\mathbb{F}_p$ or $\mathbb{F}_{2^n}$ (combined Schoof/Shanks method and Pollards $\lambda-$method).

Polynomials

- arithmetic for multivariate polynomials over $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{Q}(\alpha)$, $\mathbb{Q}(x)$, $\mathbb{Q}_p$, $\mathbb{Z}/m\mathbb{Z}$, $\mathbb{F}_{p^n}$ and $\mathbb{F}_{2^n}$
- Buchberger algorithm for determining Gröbner bases for polynomials over $\mathbb{Z}$, $\mathbb{F}_p$, $\mathbb{F}_{p^n}$, $\mathbb{Q}$, $\mathbb{Q}(\alpha)$, $\mathbb{Q}(x)$ and $\mathbb{Z}[x_1, \ldots, x_n]$
- factorization of univariate polynomials over $\mathbb{F}_p$ and $\mathbb{F}_{p^n}$ due to Berlekamp, Cantor/Zassenhaus, Ben Or and Niederreiter
- factorization of multivariate polynomials over $\mathbb{Z}$ and $\mathbb{Q}(\alpha)$.

Matrices and vectors

- arithmetic for matrices and vectors, inverse, transpose, trace, determinant, rank, characteristic polynomial for matrices over $\mathbb{Z}$, $\mathbb{Z}/m\mathbb{Z}$, $\mathbb{F}_{p^n}$, $\mathbb{Q}$, $\mathbb{Q}(\alpha)$, the polynomial rings over these structures, $\mathbb{F}_p(x)$ and $\mathbb{Q}(x_1, \ldots, x_r)$
- Hermite normal form for matrices over $\mathbb{Z}$
- elementary divisor form for matrices over $\mathbb{Z}$, $\mathbb{F}_p[x]$ and $\mathbb{Q}[x]$.