**数式処理** *J.JSSAC (2002)*
*Vol. 9, No. 2, pp. 42 - 46*

**解説**

# 数式処理システムSIMATHの紹介

小林 大輔* 中村 憲†

東京都立大学理学研究科 東京都立大学理学研究科

## 1 SIMATHとは

代数計算システムSIMATHとはドイツのザールランド大学のSIMATH groupがSiemens AGの支援によって開発した, 大学と科学研究機関向けには純粋に科学的な目的に使用されている限り無償で使用できるソフトウェアである. SIMATHには以下のような特徴がある. [1)]

## 2 SIMATHの特徴

- $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$, $\mathbb{C}$, $\mathbb{Z}/m\mathbb{Z}$, $\mathbb{Q}_p$, 有限体や大域体, すなわち代数体や関数体上の計算が出来る.
- 代数体や関数体上の多変数の多項式の計算が出来る.
- 代数体や関数体, 多項式環上の行列, ベクトル計算が出来る.
- 有理数体, 素体, 標数2の有限体, 代数体上の楕円曲線に関する種々の計算が出来る.
- システムは基本的に自動のガーベージコレクタと直接的なメモリ管理を用いるリストシステムと, 入出力の制限によって出来ている.
- C言語で記述されており, C言語のライブラリを含んでいるのでC言語の関数を簡単に呼び出せる.
- インタラクティブSIMATHカルキュレーターsimcalcがある.
- 以下のようなアルゴリズムを持っている.

*daisukek@comp.metro-u.ac.jp

†nakamula@tnt.math.metro-u.ac.jp

1) SIMATHを他のシステムに組み込むときは, simath@math.uni-sb.deとPascale.Serf@mchp.siemens.deに報告すること. なお, SIMATHの再配布は禁じられている.

– 整数基底, 付値の拡張, 数体と代数関数体の分解法則.
– 2 次の代数関数体の単数規準, 単数群, 因子やイデアルの類数, イデアル類群と 0 類群の同型の型と生成元.
– 導手, 最小モデル, 有理数体上の楕円曲線のランクと基底を求めるアルゴリズム.
– 素体や標数 2 の有限体上の楕円曲線上の有理点の個数を数える Schoof-Shanks アルゴリズム.
– 与えられた素体上で点の個数が与えられたときに, その条件をみたす楕円曲線を構成するアルゴリズム.
– LLL アルゴリズム.

## 3 入手方法

anonymous ftp で simath.math.metro-u.ac.jp (133.86.76.12) か ftp.math.orst.edu, 又は http://simath.info/ から手に入る.

## 4 インストールについて

1. 動作確認環境
   - HP 9000 series 700. OS は HP-UX 9.0x, HP-UX 10.x.
   - SGI マシン. OS は IRIX 5.3.
   - Sun SPARC ステーション. OS は SunOS 4.1.1.
   - Intel の PC. OS は Linux 1.x, 2.x.
2. 必要なもの
   (a) GNU C コンパイラ “gcc”.
   (b) GNU “make”.
   (c) 50 MB ディスク領域.
3. 手順
   (a) “./configure” と入力する.
   (b) ./smconfig.h 内の “# MAX_BLOCK_NUMBER 1024” の数をインストールするシステムの最大メモリに書き換える.
   (c) “make” と入力する.
   (d) “make SIMATH_install” と入力する.

(e) “make links” と入力する.

これでインストールが完了する. 詳しくは http://www.simath.info/INSTALLATION 又は SIMATH パッケージ内のファイル INSTALLATION を参照.

# 5 事例

UNIX のコマンドプロンプトに対して **simcalc** と入力することでインタラクティブカルキュレータ **simcalc** が起動する. [2)]

**% simcalc**

```
           *****                                          ****
           *****                                          ****
                                                          ****
*********** ***** ***************** *********** ***********  ****  ***********
*********** ***** ***************** *********** ***********  ****  ***********
***          ****  ****  ****  **** ****   ****         ***  ****  ****   ****
***********  ****  ****  ****  **** ****        ***********  ****  ****
***********  ****  ****  ****  **** ****        ***********  ****  ****
        ***  ****  ****  ****  **** ****   **** ***     ***  ****  ****   ****
*********** ***** ***** ***** ***** ********** ************ ***** ***********
*********** ***** ***** ***** ***** ********** ************ ***** ***********

                                                      version 4.5, 15 Mar 2000
Type "?help" for more information.
Type "?helpfunc" for more information about functions in simcalc.
Type "?NEW" for information about new functions in simcalc.
```

**simcalc** に計算をさせるのに難しい表記はいらない. 例えば $(1+2)*(3+4)-5$ を計算させたい時はそのまま入力し, enter を押せば良い.

```
> (1+2)*(3+4)-5
        @ = 16
```

変数も使用できる. 変数名はシステム上の特殊な文字 (列) を除いた 20 文字以下の任意の文字列が使用できる.

```
> X1234567890123456789 = 1024
        X1234567890123456789 = 1024
> X1234567890123456789 + 2^10
        @ = 2048
```

多変数関数の標記も簡単である. $f(x,y)=(x+3)^2+(y+2)^2-63$ として $f(4,6)$ を求めてみる.

```
> f=(x+3)^2 + (y + 2) ^ 2-63
```

[2)] % を UNIX のプロンプト記号として用いる.

```
        f = y^2 + 4*y + x^2 + 6*x - 50
> f(x=4,y=6)
        @ = 50
```

様々な関数を用意してある．楕円曲線に関して言えば, $X = EC(a_1, a_2, a_3, a_4, a_6)$ と入力[3)]することで楕円曲線 $y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$ を操作出来る．関数 **cond** で導手を求めてみる．

```
> f=EC(5,4,3,2,1)
        f = EC(5, 4, 3, 2, 1)
> cond(f)
  40575 = 3 * 5^2 * 541
        @ = 40575
```

ベクトル, 行列計算は次のとおり．

```
> A={{5,2,4}{4,36,7}{7,14,34}}
        A = { { 5   2   4 }
              { 4   36   7 }
              { 7   14   34 } }
> B={{7,2,1}{75,5,9}{0,7,8}}
        B = { { 7   2   1 }
              { 75   5   9 }
              { 0   7   8 } }
> A*B
        @ = { { 185   48   55 }
              { 2728   237   384 }
              { 1099   322   405 } }
> det(A)
        @ = 4672
> A^(-1)
        @ = { { 563/2336   -3/1168   -65/2336 }
              { -87/4672   71/2336   -19/4672 }
              { -49/1168   -7/584   43/1168 } }
```

?help と打つことで help を呼び出せる．詳しくは SIMATH のマニュアルを参照.

UNIX コマンドプロンプトに対して **SM** と打つことでシステム SIMATH を呼び出せる． システム SIMATH はプログラムのライブラリ操作が簡単に行えるように設計されている． C 言語の知識があれば手助けになるだろう． 詳しくは help を参照のこと． また, http://simath.info/examples/ 又は SIMATH パッケージ内の ./examples/ にも多くの例がある．

[3)] 各引数 $a_1, ..., a_6$ は実際の数字でなければならない.

## 6 注意点

楕円曲線の rank と basis の計算に bug があることが知られている. ヘルプやエラー文, メッセージ文がドイツ語で書かれているところがある. 詳しくは http://www.simath.info/known_bugs.txt 等を参照.

## 7 システム SIMATH の東京都立大学移行, SIMATH の今後

この度 SIMATH の運用, 開発の拠点を東京都立大学に移すことになった. 現在は移行期間中で 2002 年夏以前に移行完了予定である. これまで SIMATH は限られた開発者の手により設計, 修正されて来た. それにより, その開発者以外の人が状態を把握出来ず, 手を出せない状況になっていた. 更新状況もはっきりせず, おそらく 1998 年以降バージョンアップしていない. 東京都立大への移行に際してオープンソースでの運用, 開発をするべく準備している.

今後の展開としては

- これまでに知られている楕円曲線の rank と basis の計算についての bug を修正する
- CVS を用いてのシステムのオープンソースでの保守, 開発環境の構築
- マニュアルの詳細化

を目標に行動する. また, 新たな試みとして

- SIMATH の 64 ビット機への対応
- 数体の類数の円単数 · 楕円単数による計算法 (Cyclo-Elliptic-Method) の SIMATH への実装

を挙げる.

!! システムの運用 · 開発には多くの人材が必要です. SIMATH の運用 · 開発にネットワークを通じて協力して下さる方々を広く募集しています. 上記の目標以外にも, マニュアルの日本語訳作成やホームページの作成, 資料収集など様々な仕事があります. ご協力頂ける方, ご質問等は下記連絡先まで御一報下さい.

## 8 連絡先

nakamula@tnt.math.metro-u.ac.jp 又は daisukek@comp.metro-u.ac.jp まで.