

# 群論計算システム GAP

加古 富志雄

奈良女子大学理学部情報科学科\*

## Abstract

GAP (Groups, Algorithms, and Programming) は整数論や群論における計算を支援するための数式処理システムである。ここでは、GAP の入手方法および UNIX (FreeBSD4.4) と Windows98 にインストールする方法について紹介し、簡単なシステムの使用例を示す。

## 1 はじめに

オープンソース運動の広まりの中で、良質の OS や開発環境、応用ソフトが無料で自由に使えるようになってきている。数式処理ソフトについても、数多くのシステムがフリーソフトあるいはオープンソースで利用できる。これらのシステムは、汎用の計算を行なえるものから、ある特定の分野での研究の計算を行なうためのものまで、機能や計算速度において多種多様なものがある。

オープンソースでのソフトを利用する利点としては、まず無料であるという点が大きいですが、それ以外にも、システムによらず同じソフトを利用できるという点が上げられる。

フリーの数式処理関係のソフトウェアには、数式処理システムや、数式処理用のライブラリ等がある。

フリーあるいはオープンソースの数式処理関係のソフトに関する情報については、SAL (Symbolic Applications on Linux) のホームページ [1] から Computer Algebra Systems に関するページ <http://sal.kachinateck.com/A/1>、が詳しい。また、SymbolicNet (Information Center for Symbolic and Algebraic Computation)[2] や Mathematical Resources by Bruno Kevius: Computer Algebra, Symbolic mathematics [3] にも情報がある。

これらのサイトから、主な (著者の独断と偏見で選んだ) 数式処理システムを次にリストする。

- CoCoA (Computations in Commutative Algebra) は、可換代数に関する計算を行なうシステムで、例えば Gröbner 基底の計算、syzygies 計算、イデアル、多項式の因数分

---

\*kako@ics.nara-wu.ac.jp

解等、多項式環での計算が行なえる。SUN Solaris, HP, SGI IRIX, DEC Alpha, Linux, DOS, Windows, MacOS あるいは LinuxPPC の上で動作する。

CoCoA は研究および教育目的での使用のためであれば、無料で利用できる。最新のバージョンは CoCoA 4.1 (May 2001) で、次のサイトから入手できる。

<http://cocoa.dima.unige.it>

<http://www.physik.uni-regensburg.de/~krm03530/cocoa>

- FELIX は、Buchberger の算法およびその非可換環への拡張に基づく可換および非可換な環における計算を行なうシステムで、Linux, Windows, Solaris, NetBSD(intel) の上で動作する。次のサイトから入手できる。

<http://felix.hgb-leipzig.de/>

<http://www.ibiblio.org/pub/Linux/apps/math/>

- Fermat は、Mac と Windows で動く数式処理システムで、多項式や行列計算およびグラフィックスの機能を持っている。最新のバージョンは 2.6.4 (2002 年 1 月) で次のサイトから入手できる。

<http://www.bway.net/~lewis/>

- GAP (Groups, Algorithms, and Programming) は整数論や群論における計算を支援するための数式処理システムです。最新のバージョンは 4.3(2002 年 5 月) で、商用以外の用途には無料で使用できる。UNIX 用のソースコードおよび、Windows および MacOS 用のバイナリが次のサイトから入手できる。

<http://www.gap-system.org/~gap/>

<ftp://ftp-gap.dcs.st-and.ac.uk/pub/gap/gap4>

- HartMath は Java で書かれた数式処理のプログラムで、スタンドアローンアプリケーションとしてもしくは jEdit のプラグインソフトとして使用できる。リスト処理、多項式、ベクトル、行列の計算やグラフをプロットする機能を持っている。GPL に従ってソースコードも含めて公開されている。

<http://www.hartmath.org/>

- Kalamaris は関数の定義と計算、関数のプロット、行列、数値計算によって微分方程式を解くこと、3次元でのデータの可視化等の機能を持つ、数式処理システムである。GNU の GPL2 に基づいて、ソースコードが公開されている。コンパイルするには KDE, Mesa, Qt が必要である。

<http://perso.wanadoo.es/antlarr/kalamaris.html>

- Macaulay 2 は、代数幾何および可換代数における計算を行なうシステムである。最新のバージョン 0.9.3 です。Solaris, Windows, Linux でのバイナリーが提供されている。また、ホームページからソースコードも入手可能です。

<http://www.math.uiuc.edu/Macaulay2/>

- MAXIMA は、MIT で作成された汎用の数式処理システム Macsyma を元にした数式処理ソフトで、GPL に基づいて公開されている。現在最新のバージョンは MAXIMA 5.6 で、以下のサイトから入手できる。Common Lisp でかかれていますので、Lisp の処理系が必要であるが、GCL(Gnu Common Lisp) あるいは CLISP, CMU Common Lisp の上で動作する。

Windows にインストールするには、maxima55l-setup.exe をダウンロードして、実行するとイントラが起動するので、質問(インストールするディレクトリ名を入力するだけ)に答えるとインストールされる。

<http://www.ma.utexas.edu/maxima.html>

<http://sourceforge.net/projects/maxima/>

<http://www.gnu.org/software/maxima/>

- Singular 多項式に関する計算を行なうための数式処理システムで、特に、可換代数や代数幾何、特異点理論での応用を目的とする。現在最新のバージョンは 2.0.3 で、Linux, HPUX, SunOS, IRIX, AIX, FreeBSD 4.2, MacOS X, MacOS 9, および Windows 95/98/NT/2K でのバイナリが公開されている。ソースコードも公開されているが、2.0.2 のものしか見つけられなかった。

<http://www.singular.uni-kl.de/>

これらのシステムをインストールするには比較的簡単です。といっても Windows 版ではコンパイルした結果のバイナリも含めてあるのでほとんどファイルを解凍するだけで済むが、UNIX では設定とコンパイルを行う必要があるものも含まれている。次の章では、群論計算用の数式処理システムである GAP のインストールの方法について説明する。

## 2 GAP

GAP(Group, Algorithms, and Programming) は整数論や群論における計算を支援するための数式処理システムです。GAP は 1985 年から西ドイツアーヘン大学の J. Neubüser 教授を中心とするグループにより開発が行なわれ、1988 年に正式バージョンとして GAP2.4 が公開された。1997 年からは、スコットランドのセントアンドリュース大学で開発と保守が行なわれている。Pascal 言語に似たプログラミング言語で、多くの関数や有限群のデータを持っている。一番新しいバージョンは 4.3(2002 年 5 月) で、商用以外の用途には無料で使用できる。ソースコードも提供されており、Windows 9x と NT(cygwin を使ってコンパイルする)、

UNIX、MacOS で使用可能である。Windows 版と MacOS 版については、バイナリ形式でのファイルが用意されている。UNIX 版については、自分でコンパイルする必要がある。

GAP を実行するには、64MB 以上 (通常の使用には 128MB あれば十分) の主記憶を必要とする。

GAP の配布されているパッケージをすべてインストールするには、約 320MB のディスク容量が必要で、marks のテーブルをインストールするには更に 80MB が必要です。また、CARAT パッケージをインストールすると、約 180MB のデータファイルを作成するので、注意する必要があります。

## 2.1 GAP の入手

<http://www.gap-system.org>

が GAP のホームページでここから、GAP を入手することが出来る。また、GAP のマニュアルや GAP についての情報を見ることが出来る。

<http://www.math.rwth-aachen.de/~GAP>

のサイトでミラーリングをしているので、ここから入手することも出来る。

(<http://www.ccs.neu.edu/mirrors/GAP> も GAP の最新版をミラーリングしていると書かれているが、5月にこのサイトを覗いてみたところでは、古い版の GAP(GAP4.2) しか見当たらなかった。)

ファイルは、zoo, zip, あるいは tar.gz の三つの形式で圧縮したものが用意されている。UNIX で使うには、tar.gz 形式あるいは zoo 形式のファイルを、Windows では、zip 形式あるいは zoo 形式のファイルを、MacOS で使うためには zoo 形式のファイルを選べば良い。

ファイル	サイズ	内容
basic4r3	5Mb	最小構成でのインストール用ファイル
gap4r3	45Mb	標準のインストールファイル (UNIX 用)
gappc4r3	48Mb	標準のインストールファイル (Windows 用)
gapmac4r3	49Mb	標準のインストールファイル (MacOS 用) bin/mac のディレクトリ中にある実行ファイルを別途ダウンロードする必要がある
accpkg4r3	12Mb	シェアパッケージで動作が確認されたもの
deppkg4r3	4Mb	シェアパッケージで未確認のもの
xtom4r3	16Mb	群のデータ (オプション)

表 1: 配布ファイルのリスト

バグの修正による訂正があれば、bugfixes のディレクトリに fix4r3nX.zoo の名前でパッチ

ファイルが用意される。カーネルが修正された場合には、Windows 用の実行ファイルを作り直したものが、wbinXXX.zoo の名前で bugfixes のディレクトリに用意される。

新しいパッチファイルには、それ以前の修正が含まれているので、最新のパッチファイルを適用すれば良い。(これは、gap4r2 とは方針が変わっているみたいで、gap4r2 では全てのパッチを順に当てる必要があった。4r3 では、最新のパッチに全ての修正を含めるようです。最も 4r3 がリリースされたばかりですので、今のところ (5 月の末) ではパッチは出ていません。)

## 2.2 UNIX へのインストール

GAP4.3 は UNIX やその互換 OS であればほとんど問題無く動作するはずである。次のシステムと、C コンパイラで動作することが確認されている。

```
DECalpha under OSF 3.2 with GNU cc 2 or cc,
HP9000 under HP-UX with GNU cc 2,
IBM PC under Linux with GNU cc 2,
IBM RS/6000 under AIX with GNU cc 2,
SGI Mips under IRIX 5/6 with gcc2 or cc,
Sun Sparc under Solaris with GCC 2 or cc.
```

UNIX で GAP を完全にインストールするには

```
gap4r3.tar.gz
accpkg4r3.tar.gz
deppkg4r3.tar.gz
xtom4r3.tar.gz
```

のファイルが必要です。

FreeBSD4.4 でインストールするには、

1. 例えば、/usr/tmp のディレクトリに、gap4r3.tar.gz と accpkg4r3.tar.gz (もし必要なら、deppkg4r3.tar.gz と xtom4r3.tar.gz も) をダウンロードする。(バイナリモードでロードすることに注意。)
2. インストールするディレクトリ (例えば /usr/local/ あるいは自分のホームディレクトリ) に移動して、gap4r3.tar.gz を展開する。

```
tar xvzf /usr/tmp/gap4r3.tar.gz
```

を実行すると、gap4r3 のディレクトリが作成され、その下にファイルが展開される。同様にパッケージを展開する。パッケージの展開には、gap4r3/pkg のディレクトリに移動して、



```
Information at: http://www.gap-system.org
Try '?help' for help. See also '?copyright' and '?authors'
```

```
Loading the library. Please be patient, this may take a while.
GAP4, Version: 4r3 of 06-May-2002, i386-unknown-freebsd4.4-gcc
Components: small, small2, small3, small4, small5, small6, small7,
            small8, id2, id3, id4, id5, id6, trans, prim installed.
Packages:   tomlib, autpgrp, crisp, cryst, polycyclic, crystcat,
            ctllib, EDIM, GAPDoc, factint, lag installed.

gap>
```

と表示される。(ライブラリをロードするのにちょっと時間がかかる。)ここで、gap> は GAP の入力プロンプトである。例えば、

```
gap> 2 * 3 + 4;
10
gap> Factorial( 30 );
265252859812191058636308480000000
gap> Size( SymmetricGroup( 10 ) );
3628800
gap> Factors( 10^42 + 1 );
[ 29, 101, 281, 9901, 226549, 121499449, 4458192223320340849 ]
gap> m11 := Group((1,2,3,4,5,6,7,8,9,10,11),(3,7,11,8)(4,10,5,6));;
gap> Size( m11 );
7920
gap> Factors( 7920 );
[ 2, 2, 2, 2, 3, 3, 5, 11 ]
gap> Length( ConjugacyClasses( m11 ) );
10
```

5. ルートの権限を持っているのであれば、bin のディレクトリ中のシェルスクリプトファイル gap.sh を /usr/local/bin/gap にリンクするかあるいはコピーする。

以上が gap4r3 のインストールの手順である。多分、FreeBSD 以外でも同様に行なえるでしょう。ルート権限を持っていなければ、自分のホームディレクトリにインストールすることも可能である。

最後に、GAP のテストを行なうファイルが tst ディレクトリにある。

```
gap> Read("tst/testall.g");
```

と入力すると、全てのテストを実行する。このテストの実行には主記憶が 100MB 以上必要ということです。athlon 800MHz のマシン (主記憶 512MB) で実行した結果では、

	GAP4stones	time(msec)
total	75714	2044512

となって、約 34 分かかった。また Sharp PC-PJ2 (300MHz,128MB) で同じテストを行なったところ、計算時間は約 1 時間 15 分かかった。

### 2.3 XGAP のインストール

シェアパッケージは pkg ディレクトリに入っている。パッケージには、GAP のみでかかれたパッケージと C 言語でかかれたプログラムと GAP のライブラリによるパッケージがある。C 言語でかかれたプログラムを必要とするパッケージについては、それぞれコンパイルして実行モジュールを作成する必要がある。configure というスクリプトファイルが用意されていれば、/configure と make を実行すれば良い。そうでなければ、Makefile を修正して make を実行する必要がある。

UNIX 版では、X-Window によるグラフィカルユーザインターフェースを備えた、XGAP をインストールすることができる。XGAP は、accpkg4r3 に入っているので、これを展開しておく。

1. gap4r3/pkg/xgap のディレクトリに移動して configure と make を実行する。

```
cd gap4r3/pkg/xgap
./configure
make
```

2. make が正常に終了したら、bin のディレクトリ中に xgap.sh というシェルスクリプトファイルが作成される。これを実行すると XGAP が起動する。

```
bin/xgap.sh
```

3. bin のディレクトリ中のシェルスクリプトファイル xgap.sh を /usr/local/bin/xgap にリンクするかあるいはコピーする。

examples のディレクトリにサンプルファイルが用意されている。例えば、XGAP を起動して、

```
gap> Read("examples/puzzle.g");
gap> Puzzle(5,5);
```

と入力すると、GAP でかかれたパズルゲームが動く。

### 2.4 WINDOWS へのインストール

GAP を Windows9x あるいは、WindowsNT,Windows XP で動作する。

Windows にインストールするには、

1. 例えば、c:\tmp のディレクトリに、gappc4r3.zip と accpkg4r3.zip (もし必要なら、deppkg4r3.zip と xtom4r3.zip も) をダウンロードする。(バイナリモードでロードすることに注意。)
2. winzip を起動して、gappc4r3.zip を解凍する。解凍先として、C:\を設定する。(他のディレクトリを選ぶことも可能ですが、この場合ちょっとバッチファイルを修正する必要がある。)

同様にパッケージを解凍する。パッケージの解凍先として、C:\gap4r3\pkg を指定する。

gappc4r3.zip と accpkg4r3.zip 、deppkg4r3.zip をインストールした場合でディスク領域が 240MB 必要でした。

3. C:\gap4r3\bin のディレクトリに GAP のバッチファイル gap.bat と、ショートカットファイル gap.pif があるので、gap.pif をデスクトップにコピーする。
4. もし、インストールするディレクトリを C:\gap4r3 以外に下場合には、gap.pif を修正する必要がある。デスクトップにコピーした gap.pif のアイコンを右クリックして、プロパティを表示させる。

プログラムタブをクリックして

コマンドライン	C:\gap4r3\bin\gapw95.exe
作業ディレクトリ	

となっている部分で、コマンドラインの C:\gap4r3 を GAP をインストールしたディレクトリに変更する。例えば、C:\mygap\gap4r3 にインストールした場合コマンドラインを

```
C:\mygap\gap4r3\bin\gapw95.exe -l C:\mygap\gap4r3\lib
```

と修正し、[OK] をクリックする。

5. テストプログラムを実行するには、メモリとして 100MB 以上を指定する必要がある。上と同様にして gap.pif のプロパティを開き、コマンドラインを

```
c:\gap4r3\bin\gap95.exe -N -A -x 80 -r -m 100m
```

と書き直して、[OK] をクリックし、GAP を起動する。

```
gap> Read("C:/gap4r3/tst/testgall.g");
```

と入力するとフルテストが始まる。Sharp の PC-PJ2 (celeron 300Mhz, 128Mb) で実行した結果では、GAP4stone 値が 36957 で、4656 秒の計算時間がかかった。

UNIX で GAP を実行した場合には、実行中にメモリが不足すると自動的に必要なメモリを確保して実行を続けるが、Windows の場合には、どうもそうになっていないようです。フルテストを実行する場合には、オプションとして「-m 100m」を指定して 100MB 以上のメモリをあらかじめ確保しておく必要がある。

また、シェアパッケージについては、C のソースをコンパイルする必要があるパッケージが含まれているが、これらのバイナリは用意されていないようですので、別途コンパイルする必要があります。

### 3 GAP の使用例

ここでの例は、奈良女子大学人間文化研究科の落合豊行教授によるものである。

#### 3.1 組紐群の計算

群  $G = \{a, b : abbABBB = baaBAAA = e\}$  を調べる。ここで、 $A = a^{-1}$  で、 $B = b^{-1}$  である。また、 $e$  は単位元を表す。

```
gap> f:=FreeGroup("a","b");
<free group on the generators [ a, b ]>
gap> g:=f/[f.1*f.2^2*f.1^-1*f.2^-3,f.2*f.1^2*f.2^-1*f.1^-3];
<fp group on the generators [ a, b ]>
gap> a:=g.1;
a
gap> b:=g.2;
b
gap> e:=Identity(g);
<identity ...>
gap> g;
<fp group on the generators [ a, b ]>
gap> e;
<identity ...>
gap> Size(g);
1
gap>
```

群  $g$  のサイズが 1 であることから、この群は自明な群である。

#### 3.2 群の同型対応

正 20 面体群と 5 次の交代群が同型であることを調べる。

```

GAP4, Version: 4r2 of 29-Feb-2000, i386-unknown-freebsd4.4-gcc
Components:  small, small2, small3, small4, small5, small6, small7,
              small8, id2, id3, id4, id5, id6, trans, prim, tbl,
              tom installed.

gap> g:=FreeGroup("a","b");          #自由群を作る。
<free group on the generators [ a, b ]>
gap> ico:=g/[g.1^5,(g.1*g.2)^2,g.2^3]; #20 面体群
<fp group on the generators [ a, b ]>
gap> Size(ico);
60
gap> s5:=Group((1,2),(1,2,3,4,5));
Group([ (1,2), (1,2,3,4,5) ])        #5 次の対称群
gap> Size(s5);
120
gap> a5:=CommutatorSubgroup(s5,s5); #5 次の交代群は交換子群に等しい
Group([ (1,3,2), (2,4,3), (2,3)(4,5) ]) #生成元は三つ
gap> ig:=Subgroup(s5,[(1,2,3,4,5),(1,4,2)]);
Group([ (1,2,3,4,5), (1,4,2) ])
gap> Size(a5);
60
gap> Size(ig);
60
gap> hom:=GroupHomomorphismByImages(ico,ig,GeneratorsOfGroup(ico),
> GeneratorsOfGroup(ig)); # 群 ico と ig との同型対応を調べる。
[ a, b ] -> [ (1,2,3,4,5), (1,4,2) ]
gap> Kernel(hom);
Group(<fp, no generators known>)
gap> Size(Kernel(hom)); # 同型である。
1
gap> a1:=PreImagesRepresentative(hom,(1,3,2));
# 同型対応で (1,3,2) に対応する元を求める。
 $b^{-1}a^{-3}b^{-2}a^{-3}b^{-2}a^{-1}b^{-1}a^{-3}b^{-2}a^{-4}$ 
gap> a2:=PreImagesRepresentative(hom,(2,4,3));
# 同型対応で (2,4,3) に対応する元を求める。
 $b^{-1}a^{-3}b^{-2}a^{-1}b^{-1}a^{-3}b^{-2}a^{-1}$ 
gap> a3:=PreImagesRepresentative(hom,(2,3)(4,5));
# 同型対応で (2,3)(4,5) に対応する元を求める。
 $b^{-1}a^{-1}$ 
gap> ico_ag:=Subgroup(ico,[a1,a2,a3]);

```

```

# 群 ico で生成元 [a1,a2,a3] による部分群を求める。
# これは、ico 自身と一致する。
Group([ b^-1*a^-3*b^-2*a^-3*b^-2*a^-1*b^-1*a^-3*b^-2*a^-4,
        b^-1*a^-3*b^-2*a^-1*b^-1*a^-3*b^-2*a^-1, b^-1*a^-1 ])
gap> hom1:=GroupHomomorphismByImages(ico_ag,a5,
> GeneratorsOfGroup(ico_ag), GeneratorsOfGroup(a5));
# 群 ico で生成元を [a1,a2,a3] とした群と 5 次の交代群
# との同型対応を調べる。
[ b^-1*a^-3*b^-2*a^-3*b^-2*a^-1*b^-1*a^-3*b^-2*a^-4,
  b^-1*a^-3*b^-2*a^-1*b^-1*a^-3*b^-2*a^-1, b^-1*a^-1 ] ->
[ (1,3,2), (2,4,3), (2,3)(4,5) ]
gap> Kernel(hom1);
Group([ ])
gap> Size(Kernel(hom1));
1
gap>

```

実は、この計算は gap4r2 を使って計算した。gap4r3 で同じ計算をすると、正常に求められない。

```

Loading the library. Please be patient, this may take a while.
GAP4, Version: 4r3 of 06-May-2002, i386-unknown-freebsd4.4-gcc
Components:  small, small2, small3, small4, small5, small6, small7,
              small8, id2, id3, id4, id5, id6, trans, prim installed.
Packages:    tomlib, autpgrp, crisp, cryst, polycyclic, crystcat,
              ctllib, EDIM, GAPDoc, factint, lag installed.
gap> g:=FreeGroup("a","b");
.
.
.
gap> a1:=PreImagesRepresentative(hom,(1,3,2));
a^-1*b*a^3*b
gap> a2:=PreImagesRepresentative(hom,(2,4,3));
b^-1*a^-1*b*a^-2*b^-2
gap> a3:=PreImagesRepresentative(hom,(2,3)(4,5));
b^-1*a^-1
.
.
gap> hom1:= ...
gap> Kernel(hom1);

```

```

Error, cannot test whether <C> contains the family of its elements
called from IsWholeFamily( f ) called from
KernelOfMultiplicativeGeneralMapping( map )
called from <function>( <arguments> )
called from read-eval-loop Entering break read-eval-print loop ...
you can 'quit;' to quit to outer loop, or you can 'return;'
to continue
brk>

```

と、最後に *hom1* の kernel を求める段階でエラーを起こす。また、 $a_1, a_2, a_3$  の表現が *gap4r2* の結果とは異なっている。

## 4 おわりに

フリーソフトもしくはオープンソースの数式処理関係のソフトウェアをインターネットで検索すると非常にたくさんのソフトウェアが見つかる。これらは、機能やその目的とするところもさまざまで、グラフ表示ができる関数電卓のようなものから、汎用の数式処理システムや数学や物理の分野での研究に使用することを目的としたものまである。特に、ある分野での研究に必要な計算を行なうシステムでは、その得意とする分野での機能も豊富で計算速度も速い。

正 20 面体群の計算で、初めは *gap4r2* で計算を行っていたが、5 月に *gap4r3* がリリースされたのでそれをインストールして同じ計算を行なったところ正しく計算ができなくなってしまった。*gap* を使用したのはこれが初めてなのでどこかに間違いをしているかもしれない。(実はあまり詳しくマニュアルを見ていない。なにしろリファレンスだけで 900 ページ近くある。)このような時に、自分で解決しなければならないというのは、フリーなソフトを使う時の欠点だけれども、それは商用のソフトでもあまり事情は変わらない。むしろソースが公開されていることから、自分で調べる、あるいは誰か詳しい人に聞いて解決できる(可能性がある?)。また、UNIX 互換の OS であれば利用できるというのも、メジャーでない OS を使っている私からすると嬉しい。

最後に、ここでは FreeBSD へインストールする時に、*configure* と *make* を行なって作成したが、*port* コレクションに数式処理ソフトとして *asir* (*asir2000*), *gap*, *jacal*, *yacas* 等のソフトが含まれている。簡単にインストールするには、コンパイルされたパッケージをダウンロードして *pkg\_add* コマンドでインストールできる。これは、Linux や他の BSD 系の UNIX ででも同じであろう。

## 参 考 文 献

- [1] SAL (Symbolic Applications on Linux): <http://sal.kachinateck.com/>
- [2] SymbolicNet (Information Center for Symblic and Algebraic Computation):

<http://www.SymbolicNet.org/>

- [3] Mathematical Resources by Bruno Kevius: Computer Algebra, Symbolic mathematics  
<http://www.abc.se/~m9847/matre/comal.html>
- [4] 澤田秀樹: C 言語と GAP による暗号と代数プログラミング、海文堂、2000 年 4 月
- [5] G.-M. Greuel and G. Pfister: *A SINGULAR Introduction to Commutative Algebra*, Springer-verlag, 2002 年 5 月
- [6] David Eisenbud, Daniel R. Grayson, Michael E. Stillman, and Bernd Sturmfels: *Computations in algebraic geometry with Macaulay 2*, Springer-Verlag, 2001 年 9 月
- [7] M. Kreuzer and L. RObbiano: *Computational Commutative Algebra 1*, Springer.
- [8] CoCoA – Computations in Commutative Algebra:  
<http://cocoa.dima.unige.it>  
<http://www.physik.uni-regensburg.de/~krm03530/cocoa>
- [9] FELIX – computer algebra system for commutative and non-commutative algebra:  
<http://felix.hgb-leipzig.de/>  
<http://www.ibiblio.org/pub/Linux/apps/math/>
- [10] Fermat – computer algebra system for MacOS and Windows:  
<http://www.bway.net/~lewis/>
- [11] GAP – Groups, Algorithms, and Programming:  
<http://www.gap-system.org/~gap/>  
<ftp://ftp-gap.dcs.st-and.ac.uk/pub/gap/gap4>
- [12] GiNaC – open framework for symbolic computation within C++:  
<http://www.ginac.de/>  
<ftp://ftpthep.physik.uni-mainz.de/pub/GiNaC/>
- [13] HartMath – a computer algebra system written in Java:  
<http://www.hartmath.org/>
- [14] Kalamaris – mathematical package similar to Mathematica:  
<http://perso.wanadoo.es/antlarr/kalamaris.html>
- [15] JACAL – symbolic simplification and manipulation of equations:  
<http://www-swiss.ai.mit.edu/~jaffer/JACAL.html>
- [16] Macaulay 2 — algebraic geometry and commutative algebra:  
<http://www.math.uiuc.edu/Macaulay2/>
- [17] Mathomatic – artificially intelligent algebraic manipulator program:  
<http://www.lightlink.com/computer/math/>

- [18] MAXIMA – computer algebra system for symbolic and numerical computations:  
<http://www.ma.utexas.edu/maxima.html>  
<http://sourceforge.net/projects/maxima/>  
<http://www.gnu.org/software/maxima/>
- [19] Pari/GP – formal computations on recursive types at high speed (CA system in number theory):  
<http://www.parigp-home.de/>
- [20] Risa/Asir :日本で作成された数式処理システム  
<http://risa.cs.ehime-u.ac.jp/>  
<http://www.math.kobe-u.ac.jp/Asir/asir.html>
- [21] SACLIB – library for C programs for computer algebra:  
<http://www.ics.udel.edu/~saclib/>
- [22] SAML – a simple algebraic math library:  
<http://topo.math.u-psud.fr/~bousch/saml-eng.html>  
<ftp://topo.math.u-psud.fr/pub/bousch/>
- [23] SIMATH –computer algebra system for number theory:  
<http://diana.math.uni-sb.de/~simath/>
- [24] SimLab :  
<http://www.cs.cornell.edu/Info/Projects/SimLab/>
- [25] Singular – a computer algebra system for polynomial computations:  
<http://www.singular.uni-kl.de/>
- [26] Yacas – a small and highly flexible computer algebra language:  
<http://www.xs4all.nl/~apinkus/yacas.html>