数式処理 *J.JSSAC (2002)*
Vol. 9, No. 1, pp. 3 - 10

特集「いろいろな数式処理システム」

Risa/Asir

野呂 正行

神戸大学理学部*

1 Risa/Asir とは

Risa/Asir は, 数学に関連するさまざまな計算のためのツールを目指して開発が続けられているソフトウェアである. Risa/Asir の開発は, (株) 富士通研究所で 1989 年に始まった. その後, 1994 年よりバイナリが公開, 配布され, 2000 年からはソースコードが公開されている. その特徴は以下の通りである.

1. 多項式に関するさまざまな演算を大規模かつ高速に実行できる環境

   Risa/Asir は, 内部形式で表現されたさまざまな数学オブジェクトに対する演算を行う Risa engine および, engine を呼び出しながら, C 言語風のユーザ言語で書かれたプログラムを解釈実行する Asir からなる. Risa/Asir の主な機能は, 再帰表現による多項式演算を基礎とする多項式因数分解, GCD および, 分散表現による多項式演算を基礎とするグレブナ基底関連演算である. 有理数体, 代数体, 有限体など種々の基礎体上の演算が可能である.

2. 分散並列計算

   筆者らは, 同種/異種の数学ソフトウェアのクライアント-サーバ型の結合のためのプロトコルとして OpenXM (Open eXchange protocol for Mathematical objects) を提案している [3]. Risa/Asir では OpenXM に基づく API を実装することにより,

   - 複数の Asir サーバにより台数効果を目指す分散並列計算
   - Risa/Asir が OpenXM クライアントとなり, 他の OpenXM サーバを呼び出す計算

   が可能になっている. 逆に, Risa/Asir OpenXM サーバの他の OpenXM クライアントからの呼び出し, あるいは Risa/Asir サブルーチンライブラリのリンクにより, 他の数

*noro@math.kobe-u.ac.jp

© 2002 Japan Society for Symbolic and Algebraic Computation

学ソフトウェアが Risa/Asir の機能を利用することもできる. 種々の数学ソフトウェアを OpenXM クライアント, サーバ化し, このような相互呼び出しを行えるようにしたものを OpenXM パッケージとして `http://www.openxm.org/` から配布中である.

3. Open source

   ソースが完全に公開されているので, 必要に応じて使用アルゴリズムおよびその正当性などを第三者が検証できる. また, C で書かれたコードを追加することや, 既存のサブルーチンライブラリをユーザ言語から呼び出せるようにすることも困難ではない. 内部構造に関するマニュアルは, 日本語マニュアルのみではあるが, OpenXM パッケージの一部として公開されている.

本稿では, Risa/Asir の構成および機能, 入手方法について解説する.

2 実現されている機能

2.1 多項式因数分解

開発初期においては, 多項式因数分解の効率よい実装を目指して, 整数, 多項式の基本演算から因数分解アルゴリズムそのものに至るまでの各レベルでさまざまな改良が加えられた. 結果として, 現在においても Maple, Mathematica などの汎用システムと比較して遜色ない効率を実現している. 一変数多項式に関しては, Risa/Asir は古典的な Berlekamp-Hensel アルゴリズムを実装しており, 有限体上での extraneous factor を多く持つ例に対しては分解が困難になるが, 組み込みの PARI ライブラリ中の `factor` は knapsack factorization が実装されており, これを呼び出すことにより, extraneous factor を多くもつ例の高速に因数分解を実行できる.

```
[1] fctr((x-1)^200+1); [[1,1],
[x^8-8*x^7+28*x^6-56*x^5+70*x^4-56*x^3+28*x^2-8*x+2,1],...
98.16sec + gc : 16.66sec(115.9sec)
[2] pari(allocatemem,1000000)$
[3] pari(factor,(x-1)^200+1);
[ x^8-8*x^7+28*x^6-56*x^5+70*x^4-56*x^3+28*x^2-8*x+2 1 ]...
1.152sec + gc : 0.01554sec(1.19sec)
```

多変数の場合は, Risa/Asir では古典的な EZ 法が実装されているが, 主係数に整数を代入した値の間での整除関係から, 因子の主係数の上限を決める方法により, 主係数問題の軽減を図っている.

他に, 逐次拡大で表現された代数体上の一変数多項式の因数分解および最小分解体計算も行うことができる. 次の例は 6 次式の最小分解体の計算である. 結果のリストの二番目の要素は逐次拡大を表し, 一番目の要素はその拡大体上での入力多項式の因子を表す.

```
[0] load("sp")$
[151] sp(x^6+6*x^4+2*x^3+9*x^2+6*x-4);
[[x+(-#13),x+(#13+#11),20*x+(((-2*#11^5-8*#11^3-2*#11^2-6*#11-2)*#12^2
+(#11^5+4*#11^3+6*#11^2+3*#11+6)*#12-4*#11^5-16*#11^3-4*#11^2-12*#11-4)*
+#13
+(2*#11^4+#11^3+6*#11^2+5*#11-4)*#12^2+(-#11^4+2*#11^3-3*#11^2+12)*#12
+4*#11^4+2*#11^3+12*#11^2+10*#11-8),20*x+(((2*#11^5+8*#11^3+2*#11^2+6*#11+2)
*#12^2+(-#11^5-4*#11^3-6*#11^2-3*#11-6)*#12+4*#11^5+16*#11^3+4*#11^2+12*#11
+4)*#13+(-2*#11^4-#11^3-6*#11^2-5*#11+4)*#12^2+(#11^4-2*#11^3+3*#11^2+8)
+*#12
-4*#11^4-2*#11^3-12*#11^2-10*#11+8),x+(-#12),x+(-#11)],
[[(#13),t#13^2+t#11*t#13+t#11^2+3],
[(#12),t#12^3+3*t#12+t#11^3+3*t#11+2],
[(#11),t#11^6+6*t#11^4+2*t#11^3+9*t#11^2+6*t#11-4]]]
```

2.2 Gröbner 基底, イデアルの分解

1993 年ごろから始まった Gröbner 基底計算機能の実装は, 各時点における最新の研究成果を積極的に採り入れている. 特に, Buchberger アルゴリズムおよび, ある項順序から他の項順序への change of ordering において有限体上の演算を利用する modular アルゴリズムを利用することで, 効率よい Gröbner 基底計算が可能になっている. また, 最近 Faugère により提案された F_4 アルゴリズムも実装されており, 特に有限体上のグレブナ基底計算に威力を発揮している. 詳細は [1], [2] を参照されたい.

Buchberger アルゴリズムの計算は, 最終結果を得るまでどの程度中間基底が生成されるか予測できない場合が多い. このような場合にもメモリ不足を起こさずに計算が続行できるように中間基底をディスク上におく方法も提供している.

```
% mkdir nfdir
% asir
...
[0] dp_gr_flags(["Demand","./nfdir/","Print",1])$
[1] F = [...]$
[2] V = [...]$
[3] G=dp_gr_main(F,V,0,1,0)$
... /* n 番目の基底まで計算 */
interrupt ?(q/t/c/d/u/w/?) q
Abort this session? (y or n) y
% asir
...
```

```
[0] dp_gr_flags(["Demand","./nfdir/","Print",1])$
[1] F = [...]$ /* 中断前と同じ入力 */
[2] V = [...]$ /* 中断前と同じ変数リスト */
[3] G=dp_gr_main(F,V,0,1,0)$
... /* n 番目まではディスクから読み込み, n+1 番目の基底から計算 */
```

最初の計算で nfdir には生成された中間基底が保存されていく. 再開後, 基底の計算の前に nfdir を見て, 対応する基底が存在する場合には, 改めて正規形計算をせずその基底を読み込む. この方法により, 長時間におよび, かつ大量の中間基底を生成するような計算も遂行できる場合があり, また, システムダウン後も, 計算を再開できるなどの利点もある. この方法は modular アルゴリズムと併用する, すなわち, dp_gr_main() の 4 番目の引数を 1 として用いるのがよい.

Gröbner 基底計算の応用として, イデアルの準素分解, およびイデアルの根基の素イデアル分解が実装されている. これらは, 代数方程式系の解を, 既約成分に分解するために用いることができる. 次の例は, 根基の素イデアル分解の例である.

```
[0] load("primdec")$
[176] F = [(2*t*y-2)*x+z*y^2-z,-z*x^3+(4*t*y+4)*x^2+(4*z*y^2+4*z)*x+2*t*y^3
-10*y^2-10*t*y+2,(t^2-1)*x+2*t*z*y-2*z,(-z^3+(4*t^2+4)*z)*x+(4*t*z^2+2*t^3
-10*t)*y+4*z^2-10*t^2+2]$
[177] V = [x,y,z,t]$
[178] primedec(F,V);
[[t+1,z,y-1,x],[t+1,y+1,x-z,z^4-16*z^2+16],[t+1,z-2,y+1,x+2],
[t+1,y+1,x+z,z^2+4],[t+1,z+2,y+1,x-2],[t-1,z,y+1,x],
[t-1,y-1,x-z,z^4-16*z^2+16],[t-1,z-2,y-1,x+2],[t-1,y-1,x+z,z^2+4],
[t-1,z+2,y-1,x-2],[3*t^2+1,z+2*t,y+t,x-2*t],[3*t^2+1,z-2*t,y+t,x+2*t],
[t^2+3,z+2,y+t,x+2],[t^2+3,z-2,y+t,x-2],[t^4-10*t^2+1,z,y-t,x],
[t^2+1,z,y+t,x]]
```

2.3 OpenXM と分散計算

Risa/Asir から OpenXM により他の数学ソフトウェアを呼び出して計算が実行できる. OpenXM においては, 数学ソフトウェアがサーバスタックマシンとして存在し, Risa/Asir はクライアントとしてサーバに計算を依頼し, 結果を受け取る, という形をとる. サーバに計算を依頼する際には, 入出力である数学オブジェクトは CMO (Common Mathematical Object format) と呼ばれる形式により表現される. OpenXM ではスタックマシンに対するコマンド, あるいはサーバの起動, 終了, 実行中断, 復帰に至るまで仕様化されている. また, スタックマシンコマンドとして, 各数学ソフト固有のユーザ言語での呼び出し, 文字列による入出力も用意されているため, さまざまな数学ソフトをとりあえずサーバ化することは比較的容易である. Risa/Asir においてグラフ描画を行う ox_plot も OpenXM サーバとして実現されてい

	OpenXM	Mathlink
プロトコル	公開	非公開
汎用性	任意のプログラム間で通信できる	一方が Mathematica の場合のみ
データ型	多倍長整数, 多項式 etc.	原始的なもののみ (マシン整数など)
byte order	気にする必要なし	プログラム側に責任あり
実行中断, 再開	あり	なし

表 1: OpenXM と Mathlink の比較

4 る. 次の例では, `ox_plot` を立ち上げ, 陰関数の描画を依頼している.

```
[0] ox_launch(0,"ox_plot");
0
[1] ifplot(0,x^2+x*y^2+y^4+x*y^3-1)$
```

表 1 は同様の機能を提供する Mathlink との比較を示す. 特に, OpenXM では通信中, 計算中といったあらゆる状態において, クライアントからただちにサーバをリセットできる機能が仕様化されている. Asir はこれを完全に実装しており, サーバをインタラクティブに使用している場合に特に有用である.

2.4 asir OpenXM contrib

これまで述べてきた機能は, いわば Risa/Asir の kernel に関するものであった. これらの機能および, 外部の OpenXM サーバの提供する機能を Risa/Asir のユーザ言語から快適に使えるようにするため, OpenXM プロジェクトにおいて, asir OpenXM contrib (以下 asir-contrib) を開発中である. 現在までに, 基礎 (集合, 組合せ), 行列, 数式の表示, 多項式環 (グレブナ基底, 消去, ヒルベルト関数他), OpenMath CD 準拠関数, gnuplot, Mathematica PHC, sm1, TiGERS 各サーバの呼び出しのための関数が用意されている. sm1 が提供する機能としては, 多項式環関係の機能の他に, 偏微分方程式系の多項式解, 級数解計算などがある.

3 インストール方法

現在, Risa/Asir は UNIX (FreeBSD, Linux, Solaris など) および Windows (95/98/Me, NT/2000/XP) 上で動作する. それぞれについてインストール方法を解説する. 以下で, プロンプトが `#` の場合, root で作業する必要があることを示している.

3.1 UNIX

いくつかの 機種/OS 用のバイナリが http://risa.cs.ehime-u.ac.jp より入手可能である. まず, インストール先のマシンに対応する asir.tgz を入手する. これらは全て gzip で圧縮してあるので, 入手後 gzip で展開する. まず, インストールするディレクトリを決める. デフォルトでは/usr/local/lib に asir というディレクトリとしてインストールされることを仮定している. 以下このディレクトリをライブラリディレクトリと呼ぶ.

```
# gzip -dc asir.tgz | ( cd /usr/local/lib; tar xf - )
```

個人的に使用する場合には, $HOME などに置いてもよい.

```
% gzip -dc asir.tgz | ( cd $HOME; tar xf - )
```

この場合, ライブラリディレクトリの名前を環境変数 ASIR_LIBDIR に設定する必要がある.

```
% setenv ASIR_LIBDIR $HOME/asir
```

Asir 本体は, ライブラリディレクトリの asir である. /usr/local/bin あるいはユーザの実行ファイルサーチパスのどこかにシンボリックリンクを作ると便利である.

```
# ln -s /usr/local/lib/asir/asir /usr/local/bin/asir
```

これで asir が起動できる.

```
% /usr/local/bin/asir
This is Risa/Asir, Version 20010322 (Kobe Distribution).
Copyright (C) 1994-2000, all rights reserved, FUJITSU LABORATORIES LIMITED.
Copyright 2000,2001, Risa/Asir committers, http://www.openxm.org/.
GC 5.3, copyright 1999, H-J. Boehm, A. J. Demers, Xerox, SGI, HP. PARI
 2.0.17(beta), copyright (C) 1989-1999,
C. Batut, K. Belabas, D. Bernardi, H. Cohen and M. Olivier.
```

おいてあるバイナリは必ずしも最新とは言えず, また shared library などとの関係で動作しない場合もあるので, ソースコードを入手してインストール先のマシン上で make するのが最良である. この場合, PARI library など, 必要なソフトウェアを自動的にインストールするためには OpenXM パッケージのソースを anonymous cvs により入手するのがよい.

- パスワード登録 (一回目のみ)

  ```
  % setenv CVSROOT :pserver:anoncvs@kerberos.math.kobe-u.ac.jp:/home/cvs
  % cvs login
  CVS password: <anoncvs と入力>
  ```

- ソースコードの入手

```
% setenv CVSROOT :pserver:anoncvs@kerberos.math.kobe-u.ac.jp:/home/cvs
% cvs checkout OpenXM OpenXM_contrib OpenXM_contrib2
```

- make およびインストール

```
% cd OpenXM/src; make all; make install
```

実行ファイルはすべて OpenXM/bin に作られる. OpenXM パッケージを使用する場合には, 更に次を行う.

```
% cd OpenXM/rc; make
```

これで, シェルスクリプト OpenXM/rc/openxm にパスが正しく設定されるので, openxm <application name> で種々の OpenXM クライアントが起動できる. openxm はどこにおいても構わない.

```
% openxm asir
This is Risa/Asir, Version 20020301 (Kobe Distribution).
...
Asir-Contrib xm version 20010310. Copyright 2000-2001, OpenXM
Committers. ... [843] get_rootdir();
/private/noro/OpenXM/lib/asir
```

- ソースコードの更新

```
% setenv CVSROOT :pserver:anoncvs@kerberos.math.kobe-u.ac.jp:/home/cvs
% cvs update OpenXM OpenXM_contrib OpenXM_contrib2
```

3.2 Windows

必要なファイルは asirwin.tgz である. 他に, gzip.exe, tar.exe が必要だが, asirwin.tgz と同じディレクトリに用意してある. これら 3 つのファイルを同一ディレクトリにおき, DOS プロンプトから

```
C:\...> tar xzf asirwin.tgz
```

を実行すれば, Asir というディレクトリ (Asir ルートディレクトリ) ができる. このディレクトリの bin/asirgui.exe を起動すればよい. なお, Windows 版もソースから make できるが, Visual C++ が必要なので, ここでは詳しく述べない.

4 マニュアルその他

Risa/Asir のマニュアルは `http://risa.cs.ehime-u.ac.jp` から PostScript および HTML 形式で入手できる他,

`http://www.math.kobe-u.ac.jp/OpenXM/1.1.3/doc/asir2000/html-jp/man\_toc.html`

でも見ることができる. また, チュートリアルとして使えるものに, Risa/Asir ドリル

`http://www.math.kobe-u.ac.jp/~noro/main/index.html`

がある.

参 考 文 献

[1] Noro, M, Yokoyama, K.: A Modular Method to Compute the Rational Univariate Representation of Zero-Dimensional Ideals, *J. Symb. Comp.*, **28**,1, 1999, 243–263.

[2] Faugère, J.C.: A new efficient algorithm for computing Gröbner bases (F_4), *Journal of Pure and Applied Algebra*, **139** 1-3, 1999, 61-88.

[3] Maekawa, M., Noro, M., Ohara, K., Okutani, Y., Takayama, N., Tamura, Y.: The Design and Implementation of OpenXM-RFC 100 and 101, *Proceedings of ASCM2001*, 2001, 102-111.