数式処理 *J.JSSAC (2002)*
Vol. 9, No. 1, pp. 26 - 31

特集「いろいろな数式処理システム」

PARI

平野 照比古

神奈川工科大学*

1 システムの特徴

PARI はそのマニュアルの冒頭で次のように主張している。

> The PARI system is a package which is capable of doing formal computations on recursive types at high speed; it is primarily aimed at number theorists, but can be used by anybody whose primary need is speed.
>
> Although quite an amount of symbolic manipulation is possible in PARI, this system does very badly compared to much more sophisticated systems like Axiom, Macsyma, Maple, Mathematica or Reduce on such manipulations (e.g. multivariate polynomials, formal integration, etc...). On the other hand, the three main advantages of the system are its speed (which can be between 5 and 100 times better on many computations), the possibility of using directly data types which are familiar to mathematicians, and its extensive algebraic number theory module which has no equivalent in the above-mentioned systems.

このシステムの特徴はここに尽きているといっても過言ではない。上記でも触れられているが有利な点としては次の事柄を特に挙げておく。

- 速度が速い。
- 数論関係の関数が充実している。
- 冪級数環や p 進数体がサポートされている。
- 対話型として利用する (pari-gp) ほか、C のプログラムの関数として利用すること (pari libraly mode) ができる。

*hilano@ic.kanagawa-it.ac.jp

この一方で、不利な点として次の事柄が挙げられる。

- 数式処理であるべき基本的な関数がない。たとえば、2 変数以上の多項式の因数分解、Gröbner 基底の計算、形式積分がサポートされていない。
- pari libraly mode ではガーベッジコレクションがユーザーに任されている。
- pari-gp におけるプログラミングが見にくい。(次のセクションの例を参照)

2 システムを用いた事例

2.1 pari-gp によるプログラミングの例

ここでは整数係数の多項式の根の範囲を求める関数を紹介しよう。この関数をファイルにしておけば \r <file name>で読み込むことができる。なお、この関数では与えられた式が一変数多項式であるかどうかはチェックしていない。

なお、計算の根拠となる定理は次のものである。

定理 1
$f(x) = a_0x^n + a_1x^{n-1} + \cdots + a_n$ を一変数の多項式とする。このときすべての根は

$$2 \max_{k=1,2,\ldots n} \left| \frac{a_k}{a_0} \right|^{1/k}$$

の範囲にある。

右辺の値を正確に求めても余り意味がないので 10 進の桁数でこの値を近似することにする。

```
1   getrootlimit(f)=
2   {
3     local( d, leadingsize,i, i0, k ,m);
4     d = poldegree(f);
5     leadingsize = sizedigit(polcoeff(f,d));
6     d--; m=0;
7     k=1;
8     forstep( i = d , 0, -1,
9       t = (sizedigit(polcoeff(f,i))-leadingsize)\ k+1;
10      if( t > m , m = t);
11      k++
12    );
13    2*10^m
14  }
```

- 1 行目は関数の宣言である。宣言は <関数名>(<引数リスト>)=で始める。

- 引数リストではデフォルト値を与えることで省略可能な引数も利用できる。
- 3 行目はこの関数で使用するローカル変数の宣言である。
- 4 行目は与えられた多項式の次数を求めている。(関数 poldegree を利用)
- 5 行目は最高次の係数の 10 進法による桁数を求めている。(関数 sizedigit を利用)
- 定理の最大値を求めるために仮の最大値を変数 m に 0 としている。また、その他の初期化を行う。(6 行目と 7 行目)
- 8 行目は forstep を用いたループの開始である。次のところで<初期化変数>= <初期値>, <終了値>,<変化量>, の順に書いてある。(この例と同じことを C のプログラムで書けば for(i = d; i >= 0; i += -1) となる。)
- <変化量>の後にコンマ (,) があることに注意して欲しい。この後にループ内の処理を書く。複数の処理が必要であればセミコロン (;) で区切る。
- $\frac{a_k}{a_0}$ の桁数を求めるためにそれぞれの桁数の差を計算し、k 乗根を求めるためにその桁数を k で割っている。\ は除法を整数の範囲でするための演算子である。ここでは上限を求めるので切り上げのために 1 加えてある。(9 行目)
- 仮の最大値と比べ大きければ置き換える。(10 行目) このために if を用いる。if(<条件式>,<成立したとき実行する式>,<成立しなかったとき実行する式>) と記述する。ここでは成立しなかった場合の式が空なのでその前のコンマも省略されている。
- 求めたのは 10 進法における桁数であるから定理の形にするために 2×10^m を計算する。(11 行目)
- pari では最後に計算された値が関数の戻り値になる。

この例でも分かるように pari-gp におけるプログラミングは形式的には制御構造までもが関数の並びで記述されるので一定のスタイルで書く習慣をつけないと非常に読みにくいコーディングになってしまう。

2.2 pari ライブラリーモードにおける注意点

pari ライブラリーモードは C から呼び出し可能な関数の集まりとなっている。したがって、一定の約束で C のプログラムとして記述すればよい。具体的なプログラミング例はマニュアルを参照のこと。

一番の問題となるのは途中で計算されたデータが不要になりメモリー上に残る。これを再利用するためのメモリーの回収 (ガーベッジコレクション) をユーザが自分でする必要があるということである。

- pari は pari スタックと呼ばれる領域に計算の途中で現れるもの (オブジェクト) をおく。この大きさはすべての pari の関数を呼び出す前に実行しなければならない parinit で指定する。
- 使用可能な位置は大域変数 avma で示されている。
- ある時点での avma (AVailable Memory Address) の値を保持し、後にこの値に戻すことによりその時点以降に作成されたオブジェクトを廃棄できる。
- この間において必要なものを指定してそれ以外のものを廃棄する関数が用意されている。(gerepileupto gerepilemany など)

pari で扱うオブジェクトは GEN の型になる。筆者が錯覚したガーベッジコレクションの例は極端な書き方をすれば次のような形であった。

```
GEN a,b,c;
ulong ltop;
.......
ltop = avma;
a = gadd(b,c);
a = gerepileupto(ltop, a);
```

b と c を加え、結果を a にしまい、以前 a が使っていた領域を開放したつもりであるが、これは全く開放されていない。 ltop に保存した時点以降に使われた領域しかガーベッジコレクションの対象になっていないからである。

マニュアルの細かい例や注意をよく読んでなぜそうしなければいけないかを理解して欲しい。

3 入手先

http://www.parigp-home.de/ がこのプロジェクトのホームページである。このページの下の方にある Download Area からダウンロードできる。Windows と Mac に対しては Precompiled executables がある。また、ユーザーのためのメーリングリストの紹介もここでなされている。

最新版のソースコードはオープンソースの開発では良く用いられている CVS を用いて入手が可能である。http://www.gn-50uma.de/ftp/pari-2.2/manuals/CVS.txt にその説明がある。(上記のホームページから CVS のリンクをたどるとこのページに飛ぶ。)

4 インストール方法

• Unix の場合

上記サイトにある Source distributions を用いる。解凍したディレクトリに Configure スク

リプトがあるので

```
./Configure
make gp
make install
```

で必要なバイナリーとマニュアルが作成されてインストールされる。作成された対話型のプログラム名は gp である。

• Windows の場合

Precompiled executables があるのでそれを用いるのが一番簡単である。最新版を使用したければ Source distributions からコンパイルすることになるが[1] このためには Windows 上で Unix 互換環境を実現するプロジェクトである Cygwin(http://www.cygwin.com) 上で行うことを薦める。この環境では CVS を用いることもできる。これを用いた時のインストールの方法は Unix の場合とまったく同じである。

なお、対話型で利用するための gp を利用するときはコマンド入力でヒストリの編集ができるように GNU の readline ライブラリもリンクするのが良い。しかし、Cygwin に付属する GNU readline は改造されているので上記の方法ではリンクに失敗する。オリジナルの GNU readline ライブラリをソースから (Cygwin システムから見た) /usr/local/lib にインストールしておくと良い。現在の Configure スクリプトはこちらにインストールしたものを優先する設定になっている。また、Configure スクリプトの 408 行目に cygtop=/ なる記述があるが、この部分を

```
cygtop=
GP_READLINE=/usr/local/include
```

とする必要がある。こうしないと readline ライブラリのヘッダファイルが見つけられず、readline ライブラリが組み込めない。正しく組み込まれれば起動時に次のように表示される。

```
GP/PARI CALCULATOR Version 2.2.3 (development)
running cygwin (C portable kernel) 32-bit version
(readline v4.2 enabled, extended help available)

Copyright (C) 2002 The PARI Group

PARI/GP is free software, covered by the GNU General Public License,
and comes WITHOUT ANY WARRANTY WHATSOEVER.

Type ? for help, \q to quit.
Type ?12 for how to get moral (and possibly technical) support.
```

[1] 3 月 30 日現在 Source distributions における Stable Release のバージョンは 2.1.3 であるが Windows 版の Precompiled executables のバージョンは一つ前の 2.0.17 であり、 readline ライブラリが組み込まれていない。また、cygwin 版も 2.0.20 である。

```
   realprecision = 28 significant digits
   seriesprecision = 16 significant terms
   format = g0.28

parisize = 4000000, primelimit = 500000
?
```

また、マニュアルを作成するために (日本語の) TEX のシステムが必要である。Cygwin には TeTEX のシステムがあるように見えるがこれだけではインストールが不十分である。Windows 用の日本語の TEX システムのサイトとして次のところを紹介しておく。

`http://www.fsci.fuk.kindai.ac.jp/~kakuto/win32-ptex/`

5 システムを使用する上で知っておくと便利な知識

5.1 リストについて

pari においてリスト構造はサポートされているが実態は不定の長さを持ったベクトルである。リストの対する演算はリストの接合とリストの成分の置き換えぐらいしかない。リストの先頭の要素を取り出す関数 (Lisp でいう car 関数) や残りの部分を取り出す関数 (Lisp でいう cdr 関数) すらない。car に相当する関数は list[length[list]] で実現できるが cdr に相当する関数は自分で書く必要がある。

ライブラリモードでは使える関数がもっと減る。たとえば、リストを初期化するために pari-gp では x=[1,2,3] とすることは可能であるが、ライブラリモードではこのような関数は用意されていない。

5.2 pari のライブラリーモードのオブジェクトの型のチェックをしない関数

ここで利用する演算の関数はオブジェクトの型に関係なくしたければ gadd (加法) のように接頭辞 g で始まる関数群が用意されている。型がすでに限定されている場合には型のチェックをしない関数を呼ぶことも可能になっている。たとえば他倍長の整数同士の加法の関数としては addii がある。これらの関数を利用したときに起こる責任はすべてユーザにあることは言うまでもない。

6 その他

pari のシステムは Risa/Asir も利用している。引数の数が一つの関数は Risa/Asir から pari(<関数名>,<引数>) の形で呼び出すことができる。具体的に呼び出せる関数のリストは Risa/Asir のマニュアルに記載されている。