第 2 章

Maxima 入門 2

■既に習ったこと

- 四則演算：a+b;, a-b;, a*b;, a/b;, a^2;
- 代入：ev(a^2, a:b+1);
- 微分：diff(3*a^2,a);
- 連立方程式：solve([2*a+3*b=21, 4*a-2*b=2],[a,b]);

2.1 式の整理

Maxima を用いると計算を容易に行うことができるが，出力される式が必ずしも見やすい形で整理されているわけではない。ここでは，ここでは，式を整理する場合に用いるいくつかのコマンドに慣れてみよう。

■**展開** 例えば，$(a+b+c)^2$ を展開すると，$(a+b+c)^2 = a^2+b^2+c^2+2ab+2bc+2ac$ となる。このような操作を行う場合，関数「expand」を用いれば良い。

```
expand((a+b+c)^2);
```

出力された結果をある変数について整理したい場合は，関数「factorout」を用いれば良い。例えば，$(a+b+c)^2$ を展開することで得られた $a^2+b^2+c^2+2ab+2bc+2ac$ を a について整理したいのであれば，次のようなコマンドを入力する。

```
factorout(a^2 +b^2 +c^2 +2*a*b +2*b*c +2*a*c, b, c);
```

整理したい式の右側に入力されている部分は，整理しなくて良い変数を表している。ここでは，a について整理したいので，a 以外の変数である b と c を入力することになる。

■**因数分解** 因数分解を行いたい場合は，関数「factor」を用いれば良い。例えば，a^2-b^2 を因数分解し，$a^2-b^2=(a+b)(a-c)$ としたい場合，次のようなコマンドを入力する。

```
factor(a^2 -b^2);
```

■簡単化　因数分解はできないが，出来るだけ式を簡単化したい場合もあるだろう。そのような場合，関数「ratsimp」を用いれば良い。例えば，$(a^2+2ab+b^2)+(a^2+2ab+b^2)$ を簡単にしたければ，以下の様に入力すれば良い。

```
ratsimp(a^2+2*a*b+b^2+a^2+2*a*b+b^2);
```

■実数化　計算された値が分数で表示されているため，その大きさが分かりにくいということがある。例えば，$13a/137$ が $0.095a$ より大きいか知りたいことがあるかもしれない。その場合には，関数「float」を用いれば良く，入力するプログラムは以下のようになる。

```
float(13*a/137);
```

■出力の省略　計算結果の出力を省略する場合，プログラムの最後に「$」を入力すれば良い。プログラムの例は，次の「式の参照」で確認されたい。

■式の参照　実際にモデルの均衡を計算する場合，出力された結果を用いて，次の計算を行うことがよくある。そのため，コピーとペーストを何度も行う必要があるかもしれない。Maxima において，直前の結果を用いる場合，それを指示する特別な記号が容易されている。それは「%」である。例えば，$(a+b+c)^2$ を展開した後で，それを a について整理したい場合，展開と整理という 2 つの手順が必要となる。そのためには次のように，2 行のプログラムを入力し，最後に [Shift]+[Enter] を押せば良い。

```
expand((a+b+c)^2)$
factorout(%, b, c);
```

ここでは，1 行目の最後に「$」が入力されているため，1 行目の結果は出力されない。この行の結果も出力したい場合，「`expand((a+b+c)^2);`」と書き換えれば良い。また，2 行目の「%」は直前に計算された出力結果を参照している。そのため，このプログラムを次のように入力しても同様の結果を得る。

```
factorout( expand((a+b+c)^2), b, c);
```

■複数行のプログラム　式の参照が出来るようになると，複数の操作をまとめて行うことが可能になる。例えば，2 企業の数量競争における均衡生産量を求めるプログラムを書いてみよう。

企業 1 の生産量を x_1，企業 2 の生産量を x_2 とし，各企業は費用ゼロで生産しているとしよう。また市場の逆需要関数は $p=a-x_1-x_2$ で与えられるとする。このとき，各企業の利潤は，$\pi_1=(a-x_1-x_2)x_1$, $\pi_2=(a-x_1-x_2)x_2$ となる。企業の対称性を使って，均衡生産量を求めるプログラムを書くと以下のようになる。

```
diff( (a -x1 -x2)*x1, x1)$
ev(%, x1:x)$
ev(%, x2:x)$
solve( [%=0], [x] )$
ratsimp(%);
```

まず，1 行目は企業 1 の利潤を生産量 x_1 で微分し，利潤最大化条件を求めている。2 行目と 3 行目は，対称性から均衡生産量を $x = x_1 = x_2$ とし，これを代入している。4 行目は利潤最大化条件を均衡生産量 x について解いており，5 行目は計算結果を簡単に整理している。

このように，複数行の処理を一括で行うと，プログラムの塊に意味ができるので，後で読みやすいプログラムになる。また，各プログラムにメモを残したい場合，「`/*  */`」で囲めば良い。例えば，先ほどのプログラムに「クールノー競争」という名前を付けたいのであれば，以下のように入力すれば良い。

```
/* クールノー競争 */
diff( (a -x1 -x2)*x1, x1)$
ev(%, x1:x)$
ev(%, x2:x)$
solve( [%=0], [x] )$
ratsimp(%);
```

メモとしている部分は，プログラムが実行される際に無視される。実際に，これを確かめるために，以下のような例を考えてみれば良い。

```
1+a$
/* ev(%, a:2) */
%^2;
```

メモとして挿入している 2 行目の処理が行われていないことが分かる。

■保存と一括評価　Maxima の計算結果を保存したい場合，左上のメニューの [ファイル(F)] から保存すれば良い。

保存したファイルを閉じて，再び開いた場合，全てのプログラムが実行されていない状態になっている。全てのプログラムを一括で実行したい場合，[Ctrl]+[r] で実行できる。

■ギリシャ文字　経済学ではギリシャ文字を使うことがよくある。例えば利潤は π で表し，製品差別化の程度は γ で表すなどである。以下のプログラムでは，よく使うギリシャ文字として $\alpha, \beta, \gamma, \theta, \phi, \pi$ を挙げておく。

```
%alpha;
%beta;
```

```
%gamma;
%theta;
%phi;
%pi;
```

ギリシャ文字を表示したい場合，まず「%」を入力し，その直後にギリシャ文字のアルファベット表記を入力すれば良い。

■記号の定義 記号を定義すると，プログラムの見通しが良くなることがある。記号の定義は「:」で行う。例えば，次のような独占企業の均衡生産量と均衡独占利潤を求めるプログラムを考えてみよう。

独占企業は生産量 q を限界費用 c で生産することができ，逆需要関数 $p = a - q$ で表される市場で販売している。このとき，独占企業の利潤は $\pi_M = pq - cq = (a - q - c)q$ となる。逆需要関数を定義し，独占企業の利潤を定義した後で，利潤最大化問題のプログラムを書くと以下のようになる。

```
p:a-q$
piM:p*q-c*q$
diff(piM, q)$
solve( [%=0],[q]);
```

その後，得られた均衡生産量を企業の利潤に代入すると，均衡利潤を得る。

```
ev(piM,q:(a-c)/2)$
factor(%);
```

■記号の定義の解除 一度定義された記号は，自分で解除するか Maxima を再起動しない限り，その定義が有効であり続ける。例えば，前の例で利潤 `piM` を $\pi_M = (a - q - c)q$ と定義したのであれば，計算中はそれが常に有効となる。しかし，間違って定義をした場合や，その定義を解除したい場合もあるだろう。その際には，関数「kill」を用いれば良い。

例えば，「`f:a+2`」と定義し，$a = 1$ を代入してみよう。その後，記号 `f` の定義を解除し，再び $a = 1$ を代入してみると良いだろう。

```
f:a+2$
ev(f, a:1);
kill(f)$
ev(f, a:1);
```

多くの記号を定義している場合，1 つずつ定義を解除するのは面倒に感じることがあるかもしれない。そのような場合には，全ての定期を解除する「`kill(all)`」を使えば良

い。例えば,「記号 g」と「h」を定義し，代入を行った後で，全ての定義を解除し，再び代入して「kill(all)」の効果を確かめてみよう。

```
g:x+3$
h:2*x$
ev(g+h, x:1);
kill(all)$
ev(g+h, x:1);
```