第 4 章

Maxima 入門 4

■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]);`
- 展開：`expand((a+b+c)^2);`
- ある変数でまとめる:`factorout(a^2 +b^2 +c^2 +2*a*b +2*b*c +2*a*c, b, c);`
- 因数分解：`factor(a^2 -b^2);`
- 簡単化：`ratsimp(a^2+2*a*b+b^2+a^2+2*a*b+b^2);`
- 実数化：`float(13*a/137);`
- 出力の省略：`1+2$`
- 式の参照：`expand((a+b+c)^2)$  factorout(%, b, c);`
- メモ：`/* クールノー競争 */`
- ギリシャ文字：`%alpha;`
  （Maxima のバージョンが新しい場合，[Esc] a [Esc] の順に入力しても良い。）
- 文字の定義：`p:a-q;`
- 定義の解除：`kill(p);, kill(all);`
- 2 次元の図：`plot2d(x^2,[x,-1,1]);`
- 3 次元の図：`plot3d(x*y,[x,0,1],[y,0,2]);`

4.1 リスト

4.1.1 リストの定義

Maxima にはリストと呼ばれる概念がある。リストはおおむねベクトルのような概念であり，リストの各要素に様々なものを格納できる。例えば，2 から 10 までの偶数が順に

並んでいるリストは [2,4,6,8,10] という感じで定義される。このリストは，5 つの要素から構成されており，その各要素は左から順に 2, 4, 6, 8, 10 となっている。これはおおむねベクトル $(2,4,6,8,10)$ を同じ概念であることが分かる。

リストの要素には単に実数だけでなく，別のリストを要素とすることも可能である。先ほどのリスト [2,4,6,8,10] の 2 番目の要素をリスト [0,1] で置き換えたリスト [2,[0,1],6,8,10] もまたリストである。

4.1.2 リストを用いた計算

リストを用いると，同じような計算を一括して行うことが可能となる。例えば，$x+1$ と x^2 に $x=5$ を代入したい場合，リストを用いた処理を知らなければ以下の様にプログラムすることとなる。

```
ev(x+1, x:5);
ev(x^2, x:5);
```

一方，リストを用いて，この計算を行うと以下のようになる。

```
ev( [x+1, x^2], x:5);
```

計算したい内容がこのように簡単であれば，リストという概念の必要性を感じないかもしれない。しかしながら，産業組織論のモデルを解いた後で，均衡生産量，均衡価格，均衡利潤などを求めたい場合，代入を何度も繰り返すのは面倒である。そのような場合に，リストを用いると便利であろう。

4.1.3 リストを用いた定義

リストを用いて定義を行うこともできる。例えば，$(x_1, x_2) = (2,5)$ といった定義を行ってみよう。この場合，プログラムは次のようになる。

```
[x1, x2]:[2, 5];
```

これを実行した後で，x1 を実行すると，x_1 に 2 が代入されていることが確認できる。このような操作は，差別化された財の逆需要関数を定義する場合などに用いられる。例えば，$p_1 = 1 - x_1 - bx_2$, $p_2 = 1 - x_2 - bx_1$ を定義したければ，次のようにすれば良い。

```
[p1, p2]:[1-x1-b*x2, 1-x2-b*x1];
```

4.1.4 リストの要素の選択

リストの特定の要素を取り出したい場合，リストの後ろにどの要素かを示すプログラムを追記すれば，その要素を取り出すことができる。例えば，リスト [2,4,6,8] を listA

と定義し，3 番目の要素である 6 を取り出す場合，次にように入力すれば良い。

```
listA:[2,4,6,8]$
listA[3];
```

listA の後ろに書かれた [3] はリストの中の 3 番目の要素ということを表している。そのため，この部分を [2] と書き換えれば，4 が取り出されることとなる。

リストの要素にリストを含んでいる場合は次のような操作を行えば良い。例えば，リスト [2,[a,b],6,8,10] を listB と定義し，その 2 番目の要素である [a,b] を取り出す場合，次のようなプログラムを書くことになる。

```
listB:[2,[a,b],6,8,10]$
listB[2];
```

では，2 番目の要素の中の最初の要素である a のみを取り出すにはどうしたら良いであろうか。既に listB[2] によって [a,b] が取り出されており，[a,b][1] と入力すれば a が取り出されることが分かっているので，結局，次のようなプログラムによって a を取り出すことができる。

```
listB:[2,[a,b],6,8,10]$
listB[2][1];
```

より複雑な問題として，[1，[2，3，[4，5，6]，7]，8] から 5 を取り出したければ，このリストを listC と定義し，次のようなプログラムを書けば良い。

```
listC:[1，[2，3，[4，5，6]，7]，8]$
listC[2][3][2];
```

このような複雑なリストの操作を行うことはそれほど多くないが，連立方程式の解に何らかの名前を付けたい場合に，リストの操作を覚えておくと便利であろう。例えば，次のような連立方程式の解のうち，x の値が正となる解を取り出したい場合に便利である。

$$x + y = 0, \tag{4.1}$$

$$x^2 + y^2 = 1. \tag{4.2}$$

この解は $(x, y) = (-1/\sqrt{2}, 1/\sqrt{2}), (1/\sqrt{2}, -1/\sqrt{2})$ となるが，x の値が正となる解を取り出すプログラムは次のようになる。

```
solve( [ x+y=0, x^2+y^2=1],[x,y]);
%[2];
```

4.2　2 ステージゲーム

前章の資料における自由参入のモデルの様に，企業が競争（例えば数量競争）を行う前に，何らかの行動を選択するモデルは数多く存在する。このようなモデルは 2 ステージゲームと呼ばれ，多くの分野に応用されている。このゲームの基本的な構造は以下の通りである。

ステージ 1： 企業が何らかの行動を選択する。例えば，研究開発の程度，経営目的，製品差別化の程度，抱き合わせなどの販売戦略など。

ステージ 2： 企業が価格競争や数量競争を行う。市場に製造業者と小売着業者が存在し，製造業者が卸売価格を決定し，小売業者が小売価格を決定する場合，このステージがさらに 2 つのステージに分かれることもある。

2 ステージゲームを定義するためには，以下の要素を決めてやればよい。

- 誰が市場にいるのか。（企業数など）
- 各ステージで企業は何ができるのか。（価格を決定，数量を決定，投資量を決定など）
- 各ステージで行動を行う場合，企業は何を知っているのか。（価格を決定する際に，前のステージでの投資量を知っているなど）
- 各企業が行動した結果，どのような利得を得るのか。（需要関数や費用関数の形状を決め，利潤を定める）

異なるステージを持つゲームを解く場合，基本的に後ろ向きに解くことになる[*1]。以下では，様々な 2 ステージゲームの例を見てみよう。

4.3　2 ステージゲームの例

4.3.1　研究開発

企業は研究開発を行うことで，生産プロセスを効率化し，より安く生産できるようになる。このような問題を考えるためには，数量競争などの基礎的なモデルに研究開発の要素を加えれば良い。以下では，2 ステージゲームの定義に従い，どのような問題を考えているかを明らかにしてみよう。

- 市場には，企業 1 と企業 2 が存在する。
- 各企業は，最初のステージ（投資ステージ）で投資量 c_i $(i = 1, 2)$ を決定する。また，次のステージ（競争ステージ）では，生産量 x_i $(i = 1, 2)$ を決定する。

[*1] このような解き方をする理由については，ゲーム論の教科書を参照すると良いだろう。その際に，部分ゲーム完全均衡やバックワードインダクション（後ろ向き帰納法）と書かれた部分を読むと理解が深まる。

- 投資ステージでは，各企業は同時に投資量 c_i を決定する。次の競争ステージでは，自分とライバルの投資量を観察した後で，同時に生産量 x_i を決定する。
- 各企業の直面している逆需要関数は $p = 1 - x_1 - x_2$，各企業の限界費用は $c - c_i$ とする。c は投資量がゼロである場合の限界費用を表す。また，投資量が c_i であった場合，それに必要な費用は kc_i^2 であるとする。

上記の設定の下，各企業の利潤は以下の様になる。

$$\pi_1 = [p - (c - c_1)]x_1 - kc_1^2 = [1 - x_1 - x_2 - (c - c_1)]x_1 - kc_1^2, \quad (4.3)$$
$$\pi_2 = [p - (c - c_2)]x_2 - kc_2^2 = [1 - x_1 - x_2 - (c - c_2)]x_2 - kc_2^2. \quad (4.4)$$

では，このモデルを解くためのプログラムを書いてみよう。まず，逆需要関数と利潤の定義を行ってみよう。

```
kill(all)$
p:1-x1-x2$
pi1:(p-(c-c1))*x1-k*c1^2$
pi2:(p-(c-c2))*x2-k*c2^2$
```

まず，競争ステージを解いてみよう。利潤最大化条件は，各企業の利潤（π_1 と pi_2）を各企業の生産量（x_1 と x_2）で微分しゼロと置いたものである。これを連立して，各企業の生産量を求めれば良いので，そのプログラムは次のようになる。

```
solve( [ diff(pi1,x1)=0, diff(pi2,x2)=0 ], [x1, x2] )[1]$
OutcomeStage2:%;
```

次に，投資ステージを解いてみよう。そのためには，企業の利潤に先ほどの競争ステージの生産量を代入しなければならない。この生産量を代入された各企業の利潤を「pi1Stage1」および「pi2Stage1」と名付けておこう。

```
ev(pi1, OutcomeStage2)$
ratsimp(%)$
pi1Stage1:%;

ev(pi2, OutcomeStage2)$
ratsimp(%)$
pi2Stage1:%;
```

投資ステージの利潤最大化条件は，この式を各企業の投資量（c_1 と c_2）でそれぞれ微分しゼロとおいたものである。これらを連立して解いたものが，均衡投資量となる。それを求めるためのプログラムは以下の通りである。

```
solve( [ diff(pi1Stage1,c1)=0, diff(pi2Stage1,c2)=0 ], [c1, c2] )[1]$
```

```
ratsimp(%)$
OutcomeStage1:%;
```

以上から，均衡生産量，均衡投資量，均衡利潤を求めると以下のようになる。

```
[x1, x2, c1, c2, pi1, pi2]$
ev(%, OutcomeStage2)$
ev(%, OutcomeStage1)$
ratsimp(%)$
Outcome:%;
```

4.3.2　所有と経営の分離

企業の所有者と経営者が異なる場合，その人たちの目的が異なる場合がある。例えば，企業の所有者が経営者に対して利潤最大化だけではなく，一定程度の売り上げの大きさを目標とさせることもあるだろう。このような場合，企業 i の利潤を π_i，企業 i の売り上げを $r_i = px_i$ とすると，経営者に提示される目的 u_i は次式の様になる。

$$u_i = a\pi_i + (1-a)r_i.$$

この式において，$a = 1$ の場合，経営者は純粋に利潤最大化を行うことになり，$a = 0$ の場合，経営者は売上最大化を行うこととなる。a が中間的な値をとる場合，利潤と売り上げのバランスをとりながら経営を行うこととなる。

ここでは，所有と経営の分離を分析する 2 ステージゲームの定義を以下の様に行ってみよう。

- 企業 1 について，所有者 1 と経営者 1 が存在し，企業 2 について所有者 2 と経営者 2 が存在する。
- 最初のステージ（経営目的ステージ）では，各所有者が経営目的 a_i $(i = 1, 2)$ を決定する。また，次のステージ（競争ステージ）では，各経営者が与えられた目的に従い，生産量 x_i を決定する。
- 経営目的ステージでは，各所有者は同時に経営目的 a_i を決定し，競争ステージでは，経営者が全ての企業の経営目的を確認した後で，同時に自分の生産量 x_i を決定する。
- 各経営者の直面する逆需要関数は $p = 1 - x_1 - x_2$ とし，限界費用は c とする。

以上の設定の下，各経営者の目的は以下のようになる。

$$u_1 = a_1\pi_1 + (1-a_1)r_1 = a_1(p-c)x_1 + (1-a_1)px_1,$$
$$u_2 = a_2\pi_2 + (1-a_2)r_2 = a_2(p-c)x_2 + (1-a_2)px_2.$$

企業の所有者の目的は企業価値（利潤）の最大化であるとしよう。すると，所有者の目

的は以下の様になる。

$$\pi_1 = (p - c)x_1, \tag{4.5}$$
$$\pi_2 = (p - c)x_2. \tag{4.6}$$

このモデルを解くために，まず，各種の定義を行おう。

```
kill(all)$

p:1-x1-x2$

r1:p*x1$
r2:p*x2$

pi1:(p-c)*x1$
pi2:(p-c)*x2$

u1:a1*pi1+(1-a1)*r1$
u2:a2*pi2+(1-a2)*r2$
```

まず，競争ステージで選択される生産量を求めよう。そのためには，各経営者の目的 u_i を生産量 x_i で微分しゼロと置いた連立方程式を解けば良い。その解を OutcomeStage2 と定義しよう。そのためのプログラムは以下となる。

```
solve( [ diff(u1, x1)=0, diff(u2, x2)=0], [x1,x2])[1]$
ratsimp(%)$
OutcomeStage2:%;
```

次に，経営目的ステージを解いてみよう。そのためには，競争ステージでの生産量を各所有者の利得である π_i に代入しなければならない。代入後の π_i を「pi1Stage1」および「pi2Stage1」と名付け，その後，これを経営目的 a_i で微分しゼロと置いた連立方程式を解いてみよう。さらに，得られた結果を「OutcomeStage1」と名付けよう。そのためのプログラムは以下となる。

```
ev(pi1, OutcomeStage2)$
ratsimp(%)$
pi1Stage1:%$

ev(pi2, OutcomeStage2)$
ratsimp(%)$
pi2Stage1:%$
```

```
solve( [ diff(pi1Stage1, a1)=0, diff(pi2Stage1, a2)=0 ], [a1,a2] )[1]$
ratsimp(%)$
OutcomeStage1:%;
```

以上より，均衡生産量，均衡経営目的，均衡価格，均衡利潤は次のプログラムによって求めることができる。

```
[x1,x2,a1,a2,p,pi1,pi2]$
ev(%, OutcomeStage2)$
ev(%, OutcomeStage1)$
ratsimp(%);
```