第 1 章

Maxima の基礎

1.1　理論分析に必要な計算

Maxima は計算の途中仮定を助けてくれる非常に優れたソフトウェアである。Maxima をより効果的に使うためには，分析の各時点でどのような計算が必要なのかを理解することが重要となる。このことを理解するために，簡単な例としてクールノー競争（数量競争）を考えてみよう。

■クールノー競争　市場には 2 企業存在している状況を考える。各企業の限界費用は c とし，各企業の生産量は x_1, x_2 であるとする。市場の逆需要関数は $p = 1 - x_1 - x_2$ で与えられるとしよう。

この時，各企業の利潤は以下となる。

$$\pi_1 = px_1 - cx_1 = (1 - x_1 - x_2)x_1 - cx_1,$$
$$\pi_2 = px_2 - cx_2 = (1 - x_1 - x_2)x_2 - cx_2.$$

すると，各企業の利潤最大化条件は，以下となる。

$$\frac{\partial \pi_1}{\partial x_1} = 1 - 2x_1 - x_2 - c = 0, \tag{1.1}$$
$$\frac{\partial \pi_2}{\partial x_2} = 1 - x_1 - 2x_2 - c = 0. \tag{1.2}$$

この 2 式を連立すると，以下の均衡生産量を得る。

$$x_1 = x_2 = \frac{1-c}{3}.$$

■クールノー競争における計算手続き　クールノー競争の均衡導出における計算内容は以下の通りである。

1. 定義 : 例えば企業 1 については，$\pi_1 = px_1 - cx_1$ という部分で定義がなされている。
2. 代入 : 例えば企業 1 について，定義した利潤式に逆需要関数を代入している。それは，$px_1 - cx_1 = (1 - x_1 - x_2)x_1 - cx_1$ の部分で確認できる。

3. 微分：利潤最大化条件を作るために利潤を生産量で微分している。
4. 連立方程式：均衡生産量を得るために連立方程式を解いている。

この他にも四則演算がなされている。

このように見えると，分析をするために必要な個々の作業は非常に単純であり，それを組み合わせるだけで分析ができることが分かる。以下では，Maxima でクールノー競争に必要な計算をさせるためのプログラムを見てみよう。

1.2　Maxima でのプログラム

■プログラムの実行　実行するためには，[Shift]+[Enter] キーを押せばよい。

■四則演算　以下のプログラムにおける最初の注意点は，「(* *)」と表示されている部分は，プログラムにおけるメモであり，この部分を省略してもプログラムは正常に動くということである。長いプログラムを書くと，各プログラムの意味が分からなくなったりすることもあるので，必要に応じてメモを挿入すると良いだろう。

さて，最初のプログラムは四則演算に関わるものである。

```
a+a;         (* 和 *)
5-2;         (* 差 *)
2*b*b;       (* 積 *)
6*a/(2*a); (* 商 *)
(2*c)^2;     (* 乗 *)
```

各行の最後に入力されているセミコロン「;」は結果を出力するための印である。入力したプログラムの結果を出力したくない場合は，セミコロンの代わりにドルマーク「$」を入力すれば良い。例えば，1 行目の「a+a」の結果を出力したくなければ「a+a$」と入力する。

四則演算について特に注意点は無いが，あえて言うなら掛け算（積）の場合は必ず「*」の記号を挿入する必要があることである。また，どの順番で計算するかを指定するために，丸かっこ「( )」を正しく使う必要がある。その他のかっこ（例えば「 [ ] 」など）は別の意味になるので，四則演算に使用しないこと。

■代入　代入は関数「ev」を使う（evaluate の略）。代入するものは実数などでも良いし，式でも良い。ここでは，例として，x に 5 を代入する場合と x^2 に $x = 3y + 1$ を代入する場合を示しておく。

```
ev(x, x:5);
ev(x^2, x:3*y+1);
```

では，クールノー競争の例に戻ると，例えば企業 1 の利潤 $px_1 - cx_2$ に $p = 1 - x_1 - x_2$ を代入すれば良いので，そのコマンドは以下となる。

```
ev(p*x1-c*x1, p:1-x1-x2);
```

■微分　微分は関数「diff」を使う。例として，関数 x^2 を x で微分する場合と関数 $(3x+5y)^2$ を y で微分する場合を示しておく。

```
diff(x^2,x);
diff((3*x+5*y)^2, y);
```

　クールノー競争の例では，先ほど代入して得られた利潤を生産量で微分すれば良いので，企業 1 の場合，$x_1(-x_2-x_1+1)-cx_1$ を x_1 で微分すると，そのコマンドは以下となる。

```
diff(x1*(-x2-x1+1)-c*x1, x1);
```

同様に，企業 2 の利潤を企業 2 の生産量で微分する場合は，以下となる。

```
diff((-x2-x1+1)*x2-c*x2, x2);
```

■連立方程式　連立方程式を解くためのコマンドは「solve」である。例えば，1 本の（連立）方程式である $2x-1=0$ を x について解きたい場合，次のようなコマンドを入力すれば良い。

```
solve([2*x-1=0],[x]);
```

　また，次のような連立方程式を考える。

$$\begin{aligned}2x+3y&=21,\\4x-2y&=2.\end{aligned}$$

この方程式の解は $(x,y)=(3,5)$ であるが，これを求めるために入力するプログラムは以下となる。

```
solve([2*x+3*y=21, 4*x-2*y=2],[x,y]);
```

　さて，クールノー競争のケースの戻る。これまでの計算により，次の連立方程式を解けば良いことが分かっている。

$$\begin{aligned}-x_2-2x_1-c+1&=0,\\-2x_2-x_1-c+1&=0.\end{aligned}$$

これを解くために入力するプログラムは，次のようになる。

```
solve([-x2-2*x1-c+1=0,-2*x2-x1-c+1=0],[x1,x2]);
```

　以上より，均衡生産量である $x_1=-(c-1)/3$, $x_2=-(c-1)/3$ を得る。