

## 最初の一葉(FirstLeaf):文法とヘルプとプロット

Copyright ©2010 by Shigeto R. Nishitani

### 解説

#### 入力領域(>)

Mapleを起動すると赤いプロンプトがともっている。ここに命令(コマンド)を打ち込んでMapleの計算部にこちらの意志を伝えて動作させる。例えば、

```
> plot(sin(x), x);
```

と入力し、enterを入力してみよ。sin関数がプロットされる。

#### 入力の注意点(ShiftEnter)

1. 赤い領域のどこにカーソルがあってもenterを入れれば、そのブロックごとMapleに命令として渡される。
2. テキストの修正は普通のワープロソフトと同じ。
3. 命令の入力ではなく、改行だけをいれたいときはshift+enterを入れる。
4. 命令は、enterを入れた順に解釈されるのであって、テキストの上下とは関係ない。

#### コマンドの基本形(command();)

コマンドは全て次のような構文を取る。

```
command(引数1,引数2,...);
```

あるいは

```
command(引数1,オプション1,オプション2,...);
```

となる。()の中の引数やオプションの間はコンマで区切る。最後の; (セミコロ)は次のコマンドとの区切り記号。セミコロ(;)をコロン(:)に替えるとMapleからの返答が出力されなくなるが、Mapleへの入力は行われている。C言語などの手続き型プログラミング言語の標準的なフォーマットと同じである。

#### ヘルプ(?)

ヘルプは少し違った構文で、例えば先程のコマンドplotのヘルプを参照するときには、

```
> ?plot;
```

である。

ヘルプ画面は、左側に操作アイコン、検索ウィンドウ、関連リストが表示され、右側にヘルプの本文がある。本文は、簡単な意味と使い方、説明、例、参照で構成される。ほとんどが日本語に訳されているが、古いテキストやあまり使わないコマンドは英語のまま。英語が分からなくても例を参考にすればだいたい予測できる。と言うより、日本語訳を読んでも初めはチンプンカンプン。Mapleコマンドのコンセプトに慣れるまでは使用例をまねるのが一番の早道でしょう。

#### 間違い(Error)

いくつかの典型的な間違い。

まずは、左右の括弧の数が合っていないとき。

```
> plot(sin(x), x=-Pi..Pi);
```

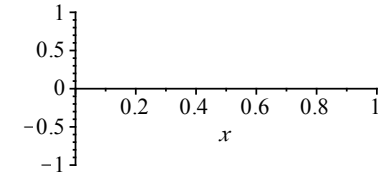
```
Error, ; unexpected
```

正しくは、

```
> plot(sin(x), x=-Pi..Pi);
```

関数の中に変数が残ったままplotしようとしたとき。

```
> plot(sin(a*x), x=0..1);
```



何も表示されない。xに0を代入しても、sin(a\*x)からは数値ではなく記号で答えが返って来ている。plotは関数が数値を返したときしか表示できない。以下のように、変数aのかわりに数値を入れる。

```
> plot(sin(2*x), x=0..1);
```

### 課題

#### 1. plotに関する以下の作業を行え。

- i) plotに関するhelpを開けよ。
- ii) 例をいくつかcopyして実行せよ。円をパラメータプロットする方法を確認せよ。
- iii) 2つの関数、sin(x)とcos(x)、をx=-Pi..Piで同時にプロットせよ。
- iv) 上記の2つの関数の表示で、オプションにcolor=[red,blue]およびstyle=[point, line]を加えて、結果を観察せよ。plot[options]に関するhelpを開け、そのほかのオプションを試してみよ。

#### 2. plot3d

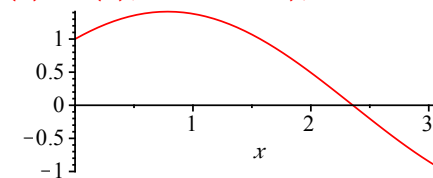
- i) x,yの2次元平面を定義域とする関数のプロットには、3次元で表示するコマンドplot3dが必要となる。2変数関数sin(x)\*cos(y)をx=-Pi..Pi,y=-Pi..Piでプロットせよ。
- ii) プロットをつまんでメニューバーにある表面あるいは軸のアイコンを操作して、等高線にしたり、軸を変更してみよ。
- iii) プロットをつまんで回してみよ。

### 解答例

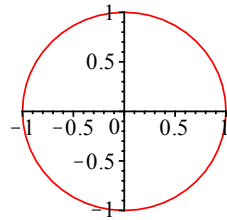
#### 1.

```
i) > ?plot;
```

```
ii) > plot(cos(x)+sin(x), x = 0 .. Pi);
```

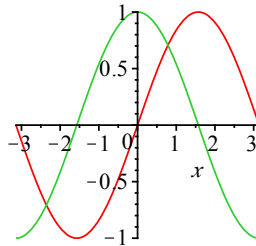


```
> plot([sin(t), cos(t), t = -Pi .. Pi]);
```



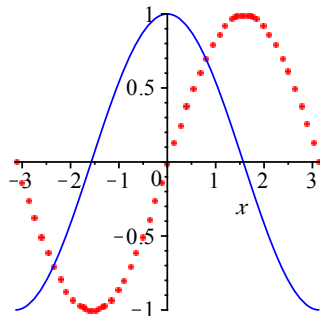
iii)

```
> plot([sin(x),cos(x)],x=-Pi..Pi);
```



iv)

```
> plot([sin(x),cos(x)],x=-Pi..Pi,color=[red,blue],style=[point,line]);
```



2.

i)

```
> plot3d(sin(x)*cos(y),x=-Pi..Pi,y=-Pi..Pi);
```



ii)

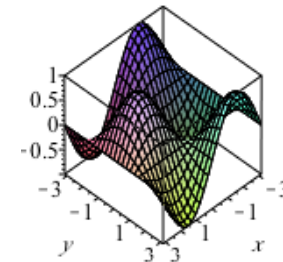
等高線は

```
> plot3d(sin(x)*cos(y),x=-Pi..Pi,y=-Pi..Pi,style=contour);
```



軸の変更は

```
> plot3d(sin(x)*cos(y),x=-Pi..Pi,y=-Pi..Pi,axes=boxed);
```

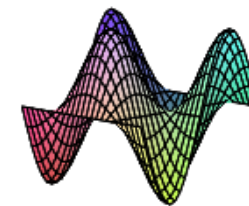


詳しくは?plot3d[option];を参照.

iii)

デフォルトの角度をplot3dのoptionで変更することが可能です.

```
> plot3d(sin(x)*cos(y),x=-Pi..Pi,y=-Pi..Pi,orientation=[45,80]);
```



## 初等関数とそのほかの関数(Elementary Function)

### 解説

#### 四則演算と evalf

四則演算は"+.\*^/"、割り切れない割り算は分数のまま表示される。

```
> 3/4;
```

強制的に数値(浮動小数点数)で出力するにはevalfを用いる。

```
> evalf(3/4);
```

#### 多項式関数(polynom)

かけ算も省略せずに打ち込む必要がある。またべき乗は"^"である。

```
> 3*x^2-4*x+3;
```

#### 平方根(sqrt)

平方根はsquare rootを略したsqrtを使う。

```
> sqrt(4);
```

#### 三角関数(trigonal)

sin, cosなどの三角関数はラジアンで入力する。ただし、 $\sin^2 x$ などは

```
> sin^2 x;
```

```
Error, missing operator or `;`
```

ではだめで、

```
> sin(x)^2
```

と省略せずに打ち込まねばならない。三角関数でよく使う定数 $\pi$ はPiと入力する。Mapleは大文字と小文字を区別するので注意。

ラジアン(radian)に度(degree)から変換するには以下のようにする。

```
> convert(180*degrees, radians);
```

#### その他の関数(inifnc)

その他の初等関数の表記やよく使われる超越関数でMapleの起動状態で用意されている関数のリストは、

```
> ?inifnc;
```

#### ユーザー定義関数(unapply)

初等関数やその他の関数を組み合わせてユーザー定義関数を作ることができる。関数 $f(x) = 2x - 3$ とおくとする場合、Mapleでは、

```
> f:=x->2*x-3;
```

と、矢印で書く。これが関数としてちゃんと定義されているかどうかを確認めるには、いくつかの数値や変数を $f(x)$ に代入して確認できる。

```
> f(3);
```

```
> f(a);
```

```
> plot(f(x), x=-2..2);
```

もう一つ関数定義のコマンドとして次のunapplyも同じ意味である。

```
> f:=unapply(2*x-3, x);
```

#### packageの呼び出し(with)

Mapleが提供する膨大な数の関数から、目的とするものを捜し出すにはhelpを使う。

普段は使わない関数は、使う前に明示的に呼び出す必要がある。例えば、線形代数によく使われる関数群は、

```
> with(LinearAlgebra);
```

としておく必要がある。この他にもいくつかの有益な関数パッケージが用意されている。

```
> ?index[package];
```

で用意されているすべてのpackageが表示される。

### 課題

▼ 1. evalfのヘルプを参照して、 $\pi$ を1000桁まで表示せよ。

▼ 2. 正接関数(tan)とその逆関数 $\tan^{-1}x$ を $x=-\text{Pi}/2..\text{Pi}/2, y=-\text{Pi}..\text{Pi}, \text{scaling}=\text{constrained}$ で同時にプロットせよ。

▼ 3. 対数関数はln(x)で与えられる。ヘルプを参照しながら次の値を求めよ。

```
[log10 1000 (常用対数), log2 1/16, log√5 125]
```

▼ 4. 次の関数は $y=2^x$ とどのような位置関係にあるか $x=-5..5, y=-5..5$ で同時にプロットして観察せよ。

```
[y=-2^x, y=(1/2)^x, y=-(1/2)^x]
```

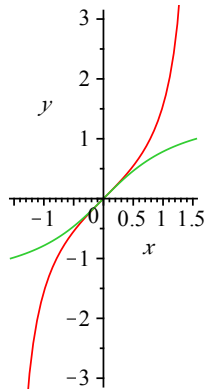
▼ 5. 指数関数はexpで与えられる。 $e^x$ とlog x関数および $y=x$ を同時に $x=-5..5, y=-5..5$ でplotせよ。またそれらの関数の位置関係を述べよ。

▼ 6. 階乗関数factorialに3を代入して何を求める関数か予測せよ。ヘルプを参照し、よりなじみの深い表記を試してみよ。

解答例

```
1. > ?evalf;  
> evalf[1000](Pi);  
省略
```

```
2. > plot([tan(x), arctan(x)], x=-Pi/2..Pi/2, y=-Pi..Pi, scaling=  
constrained);
```



```
3. > ?ln;  
> log10(1000); 3 (2.3.3.1)  
> log[2](1/16); -4 (2.3.3.2)  
> log[sqrt(5)](125); 6 (2.3.3.3)
```

```
4. > plot([2^x, -2^x, (1/2)^x, -(1/2)^x], x=-5..5, y=-5..5);  
注目している関数以外を消せばはっきりするが, i) x軸に対称, ii) y軸に対称, iii)  
原点に対称.
```

```
5. > plot([exp(x), log(x), x], x=-5..5, y=-5..5);
```

```
6. > factorial(3);  
> ?factorial;  
> 3!;  
6  
6
```

(2.3.6.1)

## 等号(ColonEqual)

### 解説

等号は、数学でいろいろな意味を持つことを中学校で学ぶ。等号それぞれの状況による操作の違いを人間は適当に判断できるが、プログラムであるMapleではそれぞれ違った記号や操作として用意され、人間がMapleに指示する必要がある。

#### 変数への代入:= (colonequal)

変数に値を代入する時には:= (colonequal)を使う。例えば、 $a=3, b=2$ のとき、 $a+b$ はいくらか？  
という問題を、Mapleで解かす時には、 $a$ に3,  $b$ に2を代入したとき、 $a+b$ はいくらか？

と読み直し、

```
> a:=3; b:=2; a+b;
```

式の定義も同様。以下は $a*x...=c*x^2...$ という式をeq1として定義している。

```
> eq1:=a*x+b=c*x^2+d*x+e;
```

#### 変数の初期化(restart)

一度何かを代入した変数を何も入れていない状態に戻す操作を変数の初期化という。すべての変数を一度に初期化するには、

```
> restart;
```

とする。ひとつの変数だけを初期化するには、シングルクォート'でくくる。

```
> a:='a';
```

#### 方程式の解(solve)

$3x=2$ を満たす $x$ をもとめよ、

という問題は、

```
> solve(3*x=2, x);
```

$$\frac{2}{3}$$

(3.1.3.1)

連立方程式は以下のとおり。

```
> solve({x+y=1, x-y=2}, {x, y});
```

$$\left\{ x = \frac{3}{2}, y = -\frac{1}{2} \right\}$$

(3.1.3.2)

ただし、solveだけでは、 $x, y$ に値は代入されない。

```
> soll:=solve({x+y=1, x-y=2}, {x, y});
```

```
> assign(soll);
```

$$\text{soll} := \left\{ x = \frac{3}{2}, y = -\frac{1}{2} \right\}$$

(3.1.3.3)

とする必要がある。確認してみると

```
> x, y;
```

$$\frac{3}{2}, -\frac{1}{2}$$

(3.1.3.4)

となり、値が代入されていることがわかる。

#### 方程式の数値解(fsolve)

解析的に解けない場合は、数値的に解を求めるfsolveを使う。

```
> x:='x';
```

```
> fsolve(log(x)-exp(-x), x);
```

$x := x$

1.309799586

(3.1.4.1)

### 式の変形

式の変形にも等号が使われる。例えば、

$$(x-2)^2 = x^2 - 4x + 4$$

というのが等号で結ばれている。式の変形とは、変数 $x$ がどんな値であっても成り立つ恒等的な変形である。

この式変形も、問題としては、

$$(x-2)^2 \text{を展開(expand)せよ}$$

と与えられるので、そのままMapleコマンドに読み替えて

```
> expand((x-2)^2);
```

$$x^2 - 4x + 4$$

(3.1.5.1)

とすればよい。因数分解(factor)や微分(diff)・積分(int)も同様に等号で結ばれるが、Mapleには操作を指示する必要がある。詳しくは他の単元で。

### 課題

1.  $a=3, b=4$ として $a, b$ の四則演算をおこなえ。また、べき乗 $a^b$ を求めよ。

2.  $eq1=3*x+4=2*x-2, a=2$ とした場合の $eq1/a, eq1+a$ を試し、両辺を観察せよ。

3. 3点 $(1,2), (-3,4), (-1,1)$ を通る2次方程式を求めよ。

4. 方程式 $\sin(x+1)-x^2=0$ の2つの解をfsolveのヘルプを参照して求めよ。

5. 関数 $y=\exp(x)$ の逆関数を求めよ。2つの関数および $y=x$ を同時に $x=-5..5, y=-5..5$ でplotし、 $y=x$ に対して2つの関数が対称であることを確認せよ。

## 解答例

1.

```
> a:=3;
> b:=4;
> a+b;a-b;a*b;a/b;a^b;
省略
```

2.

```
> eq1:=3*x+4=2*x-2;
> a:=2;
> eq1/a;
> eq1+2;
```

$$\begin{aligned} eq1 &:= 3x + 4 = 2x - 2 \\ a &:= 2 \\ \frac{3}{2}x + 2 &= x - 1 \\ 3x + 6 &= 2x \end{aligned}$$

(3.3.2.1)

3.

まず2次関数を定義する.

```
> restart;
> f:=x->a*x^2+b*x+c;
```

$$f:=x \rightarrow ax^2 + bx + c$$

(3.3.3.1)

(1,2)と通ることから,  $f(1)=2$ が成立. これをeq1として保存.

```
> eq1:=f(1)=2;
```

$$eq1 := a + b + c = 2$$

(3.3.3.2)

他の点も同様に

```
> eq2:=f(-3)=4;
> eq3:=f(-1)=1;
```

$$\begin{aligned} eq2 &:= 9a - 3b + c = 4 \\ eq3 &:= a - b + c = 1 \end{aligned}$$

(3.3.3.3)

この3個の連立方程式から, a,b,cを求めれば解となる.

```
> solve({eq1,eq2,eq3},{a,b,c});
```

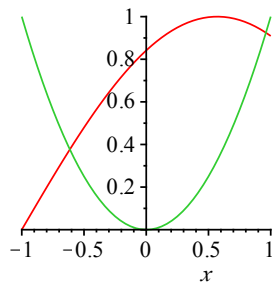
$$\left\{ a = \frac{1}{2}, b = \frac{1}{2}, c = 1 \right\}$$

(3.3.3.4)

4.

まず, 2つの関数とみなしてプロット.

```
> plot([sin(x+1),x^2],x=-1..1);
```



解が2つあることに注意. 与えられた関数値が0となる方程式として定義し, これをsolveでとく.

```
> eq1:=sin(x+1)-x^2=0;
> solve(eq1,x);
```

$$\begin{aligned} eq1 &:= \sin(x+1) - x^2 = 0 \\ -1 + \text{RootOf}(-\sin(\_Z) + 1 - 2\_Z + \_Z^2) \end{aligned} \quad (3.3.4.1)$$

これでは解を求めてくれないので, fsolveで数値解を求める.

```
> fsolve(eq1,x);
```

$$0.9615690350 \quad (3.3.4.2)$$

これではxの負にあるもう一つの解がでない. これを解決するには, fsolveでxに初期値を入れて実行する.

```
> fsolve(eq1,x=-1..0);
```

$$-0.6137631294 \quad (3.3.4.3)$$

5.

まず, 与関数をf(x)として定義しておく.

```
> f:=x->exp(x);
```

$$f:=x \rightarrow e^x \quad (3.3.5.1)$$

y=f(x)として, xについて解いてみる.

```
> eq1:=y=f(x);
> solve(eq1,x);
```

$$\begin{aligned} eq1 &:= y = e^x \\ \ln(y) \end{aligned} \quad (3.3.5.2)$$

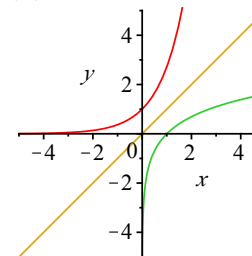
うまくyの関数として解けている. これをea2と定義し直して, 逆関数として定義する.

```
> eq2:=solve(eq1,x);
> invf:=unapply(eq2,y);
```

$$\begin{aligned} eq2 &:= \ln(y) \\ \text{invf} &:= y \rightarrow \ln(y) \end{aligned} \quad (3.3.5.3)$$

同時にプロットしてみる.

```
> plot([f(x),invf(x),x],x=-5..5,y=-5..5);
```



y=xを軸として, 2つの関数が対称であることが確認できる.