

クレジット:

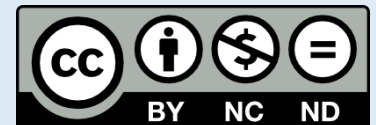
Mathematics and Informatics Center 文科系のための線形代数・解析 I
2020 藤堂 眞治・松尾 泰・藤原 毅夫

ライセンス:

利用者は、本講義資料を、教育的な目的に限ってページ単位で利用することができます。特に記載のない限り、本講義資料はページ単位でクリエイティブ・コモンズ 表示-非営利-改変禁止 ライセンスの下に提供されています。

<http://creativecommons.org/licenses/by-nc-nd/4.0/>

本講義資料内には、東京大学が第三者より許諾を得て利用している画像等や、各種ライセンスによって提供されている画像等が含まれています。個々の画像等を本講義資料から切り離して利用することはできません。個々の画像等の利用については、それぞれの権利者の定めるところに従ってください。



第4講 (2)

4-3 数式処理

MATLABはシンボリック演算（数式処理, symbolic computation, computer algebra）の機能も備えている。数式処理の分野は現在ではより広く、計算機数学（computer mathematics）ともよばれる。

シンボリック演算をおこなうには、最初に変数や行列、関数が記号であることを宣言しなくてはならない。

```
syms x % 変数xがシンボリック変数であることを宣言.  
expand((x+1)^3)
```

```
ans = x3 + 3x2 + 3x + 1
```

```
simplify(x^3+3*x^2+3*x+1)
```

```
ans = (x + 1)3
```

```
simplify(x^2+2+2*x^2+7)
```

```
ans = 3x2 + 9
```

シンボリックな求解 1

```
syms x  
eqn=sin(x)-1==0;  
sol=solve(eqn)
```

```
sol =
```

```
 $\frac{\pi}{2}$ 
```

シンボリックな求解 2

年5%の利息で、何年経つと2倍になるか？

```
clearvars  
syms n  
eqn=n*log(1.05)==log(2) %n=log(2)/log(1.05)
```

```
eqn =
```

```

$$439462730345599 n = \frac{6243314768165359}{9007199254740992}$$

```

```
sol=solve(eqn)
```

```
sol =
```

6243314768165359
439462730345599

```
a=double(sol) % 倍精度
```

```
a = 14.2067
```

```
format long  
a
```

```
a =  
14.206699082890461
```

```
a=single(sol) % 単精度
```

```
a = single  
14.2066994
```

```
format  
a
```

```
a = single  
14.2067
```

シンボリックな求解：変数が複数ある場合

2次式の根の公式

```
clearvars  
syms a b c x sol1 sol2  
eqn=a*x^2+b*x+c
```

```
eqn = a x2 + b x + c
```

```
sol1=solve(eqn,x)
```

```
sol1 =  

$$\left( \begin{array}{c} -\frac{b + \sqrt{b^2 - 4ac}}{2a} \\ -\frac{b - \sqrt{b^2 - 4ac}}{2a} \end{array} \right)$$

```

これをaで解いたら

```
sol2=solve(eqn,b)
```

```
sol2 =  

$$-\frac{ax^2 + c}{x}$$

```

ベクトル，行列もシンボリックとして定義.

```
syms A [3 4]
```

```
A
```

```
A =
```

$$\begin{pmatrix} A_{1,1} & A_{1,2} & A_{1,3} & A_{1,4} \\ A_{2,1} & A_{2,2} & A_{2,3} & A_{2,4} \\ A_{3,1} & A_{3,2} & A_{3,3} & A_{3,4} \end{pmatrix}$$

```
A(1,3)
```

```
ans = A_{1,3}
```

```
A'
```

```
ans =
```

$$\begin{pmatrix} \overline{A_{1,1}} & \overline{A_{2,1}} & \overline{A_{3,1}} \\ \overline{A_{1,2}} & \overline{A_{2,2}} & \overline{A_{3,2}} \\ \overline{A_{1,3}} & \overline{A_{2,3}} & \overline{A_{3,3}} \\ \overline{A_{1,4}} & \overline{A_{2,4}} & \overline{A_{3,4}} \end{pmatrix}$$

```
A.'
```

```
ans =
```

$$\begin{pmatrix} A_{1,1} & A_{2,1} & A_{3,1} \\ A_{1,2} & A_{2,2} & A_{3,2} \\ A_{1,3} & A_{2,3} & A_{3,3} \\ A_{1,4} & A_{2,4} & A_{3,4} \end{pmatrix}$$

```
B=sym('B',[1 3])
```

```
B = (B_1 B_2 B_3)
```

```
syms C [1 3] % C と [1 3]の間には空白が必要.
```

```
C
```

```
C = (C_1 C_2 C_3)
```

```
C(2)
```

```
ans = C_2
```

```
D=C'
```

```
D =
```

$$\begin{pmatrix} \overline{C_1} \\ \overline{C_2} \\ \overline{C_3} \end{pmatrix}$$

ベクトルの掛け算

C

$$c = (C_1 \ C_2 \ C_3)$$

C'

ans =

$$\begin{pmatrix} \overline{C_1} \\ \overline{C_2} \\ \overline{C_3} \end{pmatrix}$$

C*C'

$$\text{ans} = C_1 \overline{C_1} + C_2 \overline{C_2} + C_3 \overline{C_3}$$

C.*C'

ans =

$$\begin{pmatrix} C_1 \overline{C_1} & C_2 \overline{C_1} & C_3 \overline{C_1} \\ C_1 \overline{C_2} & C_2 \overline{C_2} & C_3 \overline{C_2} \\ C_1 \overline{C_3} & C_2 \overline{C_3} & C_3 \overline{C_3} \end{pmatrix}$$

abs(C)

$$\text{ans} = (|C_1| \ |C_2| \ |C_3|)$$

norm(C)

$$\text{ans} = \sqrt{|C_1|^2 + |C_2|^2 + |C_3|^2}$$

行列の掛け算

syms A [2 3]; syms B [3 2]

A

A =

$$\begin{pmatrix} A_{1,1} & A_{1,2} & A_{1,3} \\ A_{2,1} & A_{2,2} & A_{2,3} \end{pmatrix}$$

B

B =

$$\begin{pmatrix} B_{1,1} & B_{1,2} \\ B_{2,1} & B_{2,2} \\ B_{3,1} & B_{3,2} \end{pmatrix}$$

A*B

ans =

$$\begin{pmatrix} A_{1,1}B_{1,1} + A_{1,2}B_{2,1} + A_{1,3}B_{3,1} & A_{1,1}B_{1,2} + A_{1,2}B_{2,2} + A_{1,3}B_{3,2} \\ A_{2,1}B_{1,1} + A_{2,2}B_{2,1} + A_{2,3}B_{3,1} & A_{2,1}B_{1,2} + A_{2,2}B_{2,2} + A_{2,3}B_{3,2} \end{pmatrix}$$

微分

```
clearvars
syms x n
f(x,n)=x^n
```

$$f(x, n) = x^n$$

% x,y がシンボリック変数だから, fはシンボリック関数
diff(f,x)

$$\text{ans}(x, n) = n x^{n-1}$$

diff(f,n) %f=x^n=exp(n*log(x)) 合成関数

$$\text{ans}(x, n) = x^n \log(x)$$

4-4 関数

4-4-1 例1

多くの初等関数が組み込まれている.

sqrt(x),

sin(x), cos(x) など

exp(x)

log10(x)

log(x)

log2(x)

など

4-4-2 例2

上で既に出てきたが

```
clearvars
syms x n
f(x,n)=x.^n
```

$$f(x, n) = x^n$$

```
2*f
```

$$\text{ans}(x, n) = 2 x^n$$

```
g(x,n)=2*f(x,n)^2+f(x,1)+f(1,2)
```

$$g(x, n) = x + 2 x^{2n} + 1$$

```
g(2,n)
```

$$\text{ans} = 2 \cdot 2^{2n} + 3$$

```
g(2,3)
```

$$\text{ans} = 131$$

また次のようにも使える

4-4-3 例3

ここでは例2で定義されてclearvars

```
x
```

$$x = x$$

```
g(x)=f(x,2)
```

$$g(x) = x^2$$

```
a=g(1)
```

$$a = 1$$

```
b=f(1,2)
```

$$b = 1$$

という使い方ができる。これをクリアするには

```
clearvars x g
%x
%g
f
```

$$f(x, n) = x^n$$

```
f(1,1)
```

```
ans = 1
```

また次のようにも使える

4-4-4 例4

無名関数

```
clearvars  
f=@(x,y) x.^2+x.*y-y.^2-1
```

```
f = 値をもつ function_handle:  
@(x,y)x^2+x*y-y^2-1
```

```
f(1,2)
```

```
ans = -2
```

シンボリック関数

それでは

```
syms x y  
g(x,y) = x.^2+x.*y-y.^2-1
```

$$g(x, y) = x^2 + xy - y^2 - 1$$

```
x=1
```

```
x = 1
```

```
y=2
```

```
y = 2
```

```
g(x,y)
```

```
ans = -2
```

```
g(1,2)
```

```
ans = -2
```

最初に

```
syms x y
```

を定義したらどうなるか.


```
syms x y
g(x,y) = x.^2+x.*y-y.^2-1
```

$$g(x, y) = x^2 + xy - y^2 - 1$$

```
x=1
```

$$x = 1$$

```
y=2
```

$$y = 2$$

```
g(x,y)
```

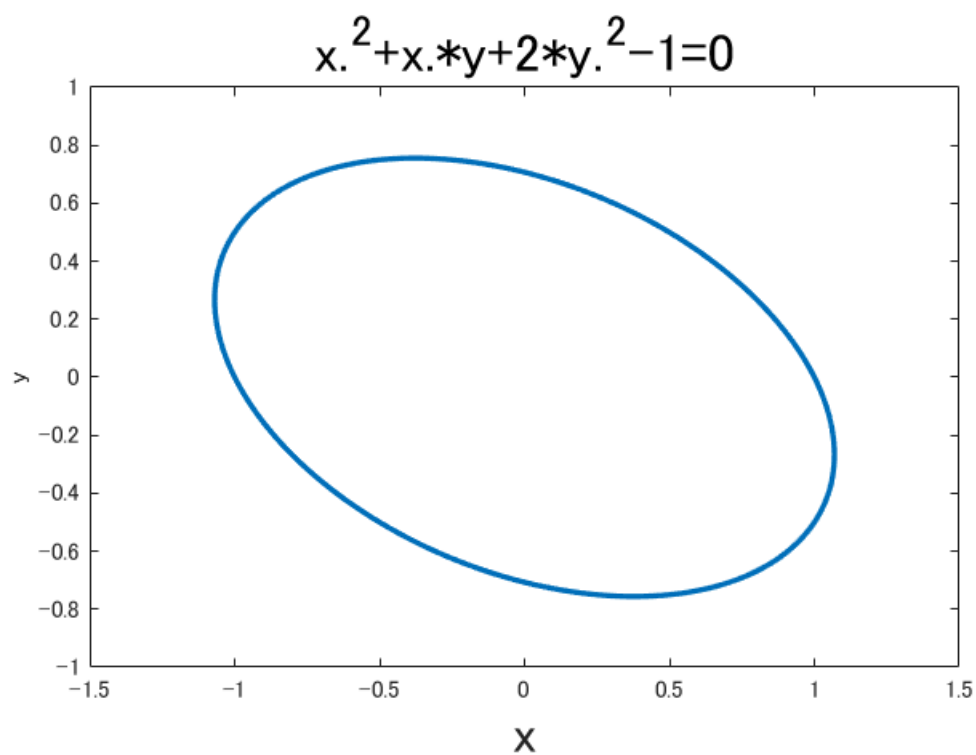
$$\text{ans} = -2$$

今度は、 x , y がシンボリック変数として定義されたので、 $g(x, y)$ はシンボリック関数となり、許された。

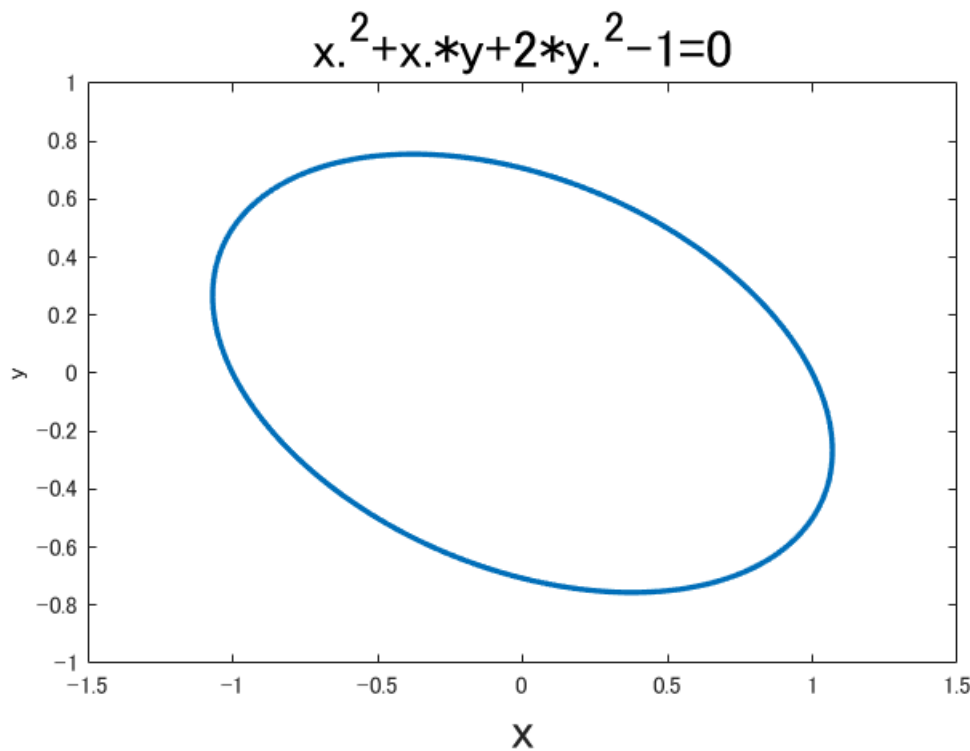
4-4-5 関数の使い方 例

陰関数を定義，関数をplotするルーチンを用いる場合．

```
clearvars
f=@(x,y) x.^2+x.*y+2*y.^2-1; %陰関数 F(x,y)=0
fimplicit(f,[-1.5 1.5 -1 1], 'LineWidth',2)
daspect([1 1 1])
xlabel('x', 'FontSize',20);ylabel('y')
title('x.^2+x.*y+2*y.^2-1=0', 'FontSize',20)
```



```
clearvars f
%f
%x
syms x y
f(x,y)=x.^2+x.*y+2*y.^2-1;
fimplicit(f,[-1.5 1.5 -1 1], 'LineWidth',2)
daspect([1 1 1])
xlabel('x', 'FontSize',20);ylabel('y')
title('x.^2+x.*y+2*y.^2-1=0', 'FontSize',20)
```



4-5 ローカル関数の定義と受け渡し

複数のスクリプト間で関数を受け渡す方法. 関数が複雑になると便利.

メインスクリプト以外のスクリプト内で定義された関数はローカル関数と呼ばれる.

```
x = 5;
y = 2;
z1 = comb(x,y)    % x!/(x-y)!*y!
```

```
z1 = 10
```

```
z2 = comb(x,x-y)
```

```
z2 = 10
```

```
function c=comb(x,y)
    c = fact(x)/perm(x,y);
end

function p = perm(n,m)
    p = fact(n-m)*fact(m);    % n!*(n-r)!
end

function f = fact(n)
    f = prod(1:n);    % 階乗 n!    (組み込み関数)
end
```

課題

角度を，度分秒で入力し、ラジアンで答えるプログラムを，function文を用いて作成せよ．