

# シミュレーション基礎(4)

第2章 すべては行列から #2

# 2.4 行列の処理

## 2.4.1 行列の部分を取り出す

番号を指定して取り出す  
ベクトルの場合

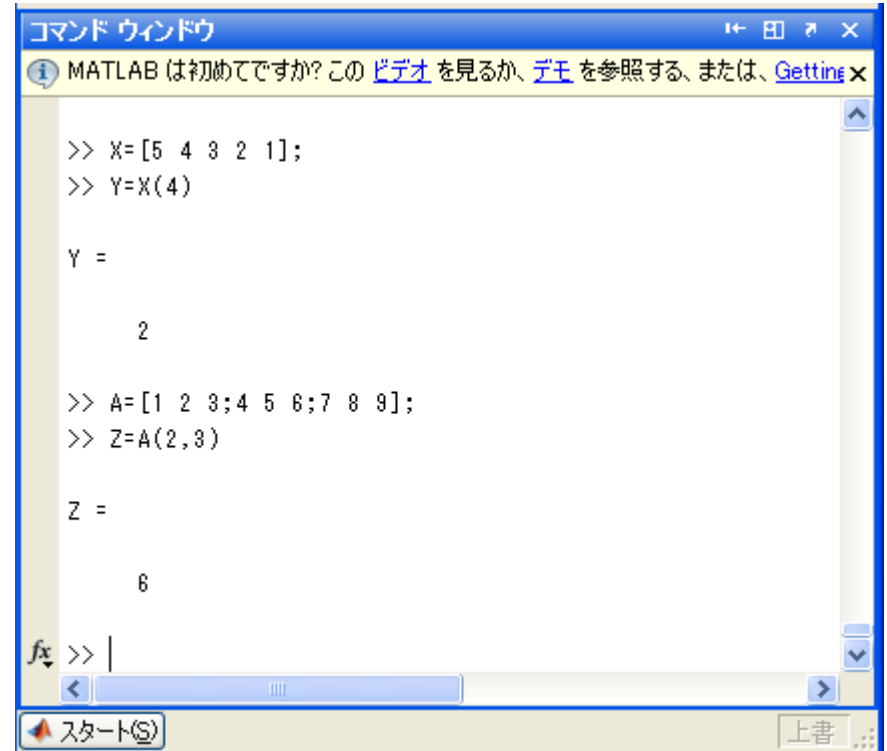
```
X=[5 4 3 2 1];
```

```
Y=X(4)
```

行列の場合

```
A=[1 2 3;4 5 6;7 8 9];
```

```
Z=A(2,3)
```



The screenshot shows the MATLAB Command Window with the following text:

```
コマンド ウィンドウ
MATLAB は初めてですか? この ビデオ を見るか、デモ を参照する、または、Getting started
>> X=[5 4 3 2 1];
>> Y=X(4)

Y =

     2

>> A=[1 2 3;4 5 6;7 8 9];
>> Z=A(2,3)

Z =

     6
```

※ C言語行番号や列番号は0から始まるのに対し、  
MATLABでは1から始まる。

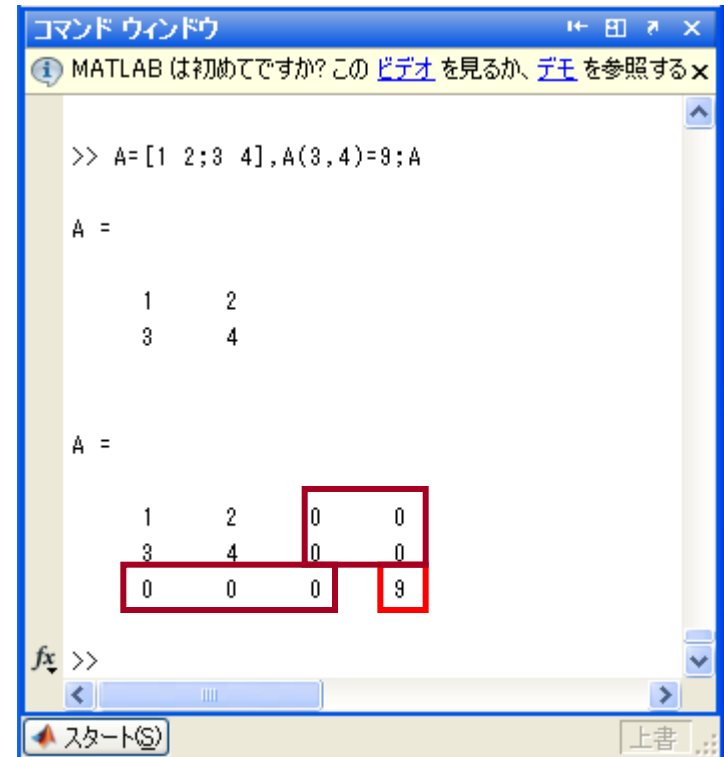
## 2.4 行列の処理

### 2.4.2 成分を追加する

ベクトルや行列に新たな成分を追加するには  
追加する成分をセットするだけでOK

$A=[1 \ 2;3 \ 4];A(3,4)=9;A$

※ $A(3,4)$ をセットすることにより、  
行列Aは2行2列の行列から  
3行4列の行列に変更される  
(指定されていない成分は0)



```
コマンド ウィンドウ
MATLAB は初めてですか? この ビデオ を見るか、 デモ を参照する x

>> A=[1 2;3 4],A(3,4)=9;A

A =

     1     2
     3     4

A =

     1     2     0     0
     3     4     0     0
     0     0     0     9
```

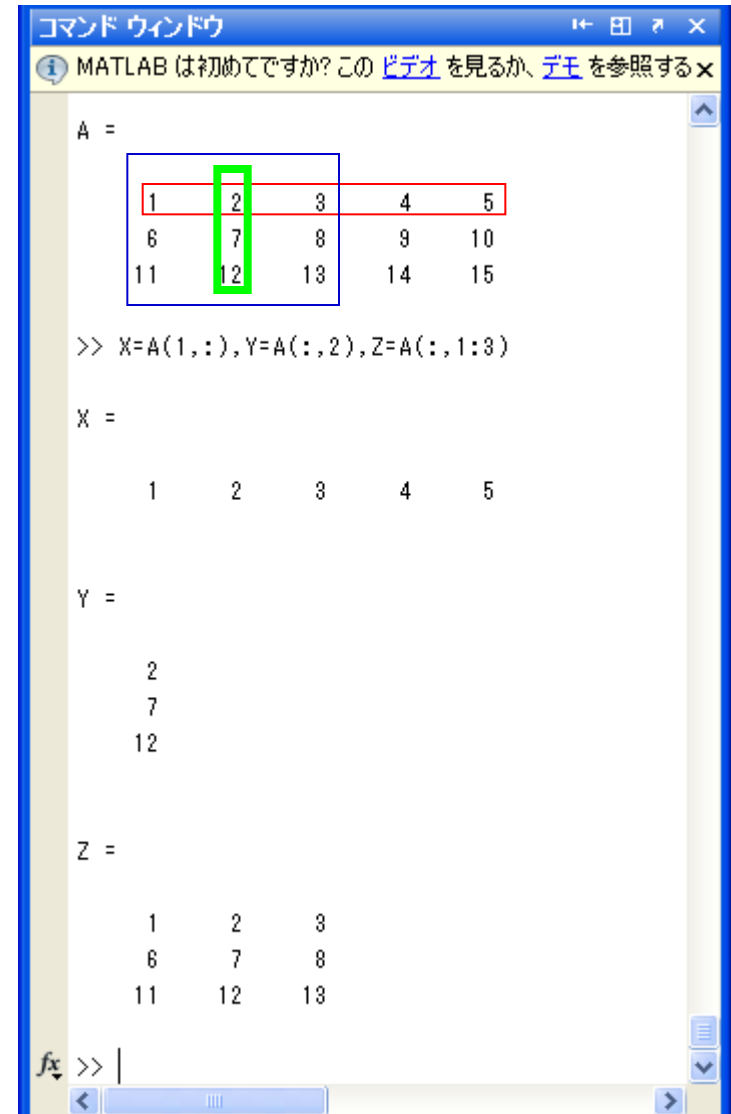
# 2.4 行列の処理

## 2.4.3 行列から行や列を取り出す

取り出したい行や列の  
部分に「:」をおいて取り出す

$A=[1:5;6:10;11:15]$

$X=A(1,:), Y=A(:,2), Z=A(:,1:3)$



The screenshot shows the MATLAB Command Window with the following content:

```
コマンド ウィンドウ
MATLAB は初めてですか? この ビデオ を見るか、デモ を参照する x

A =
     1     2     3     4     5
     6     7     8     9    10
    11    12    13    14    15

>> X=A(1,:),Y=A(:,2),Z=A(:,1:3)

X =
     1     2     3     4     5

Y =
     2
     7
    12

Z =
     1     2     3
     6     7     8
    11    12    13

fx >> |
```

The matrix A is displayed with a red box around the first row and a green box around the second column. The output shows X as the first row, Y as the second column, and Z as the first three columns of A.

# 2.4 行列の処理

## 2.4.3 行列から行や列を取り出す #2

- (1)行列の任意の列や行の番号を成分とするベクトルを用意
- (2)そのベクトルを用いて行や列を指定して成分を取り出す

$A=[1:3;4:6;7:9;10:12]$

$L=[1\ 4];B=A(L,:)$

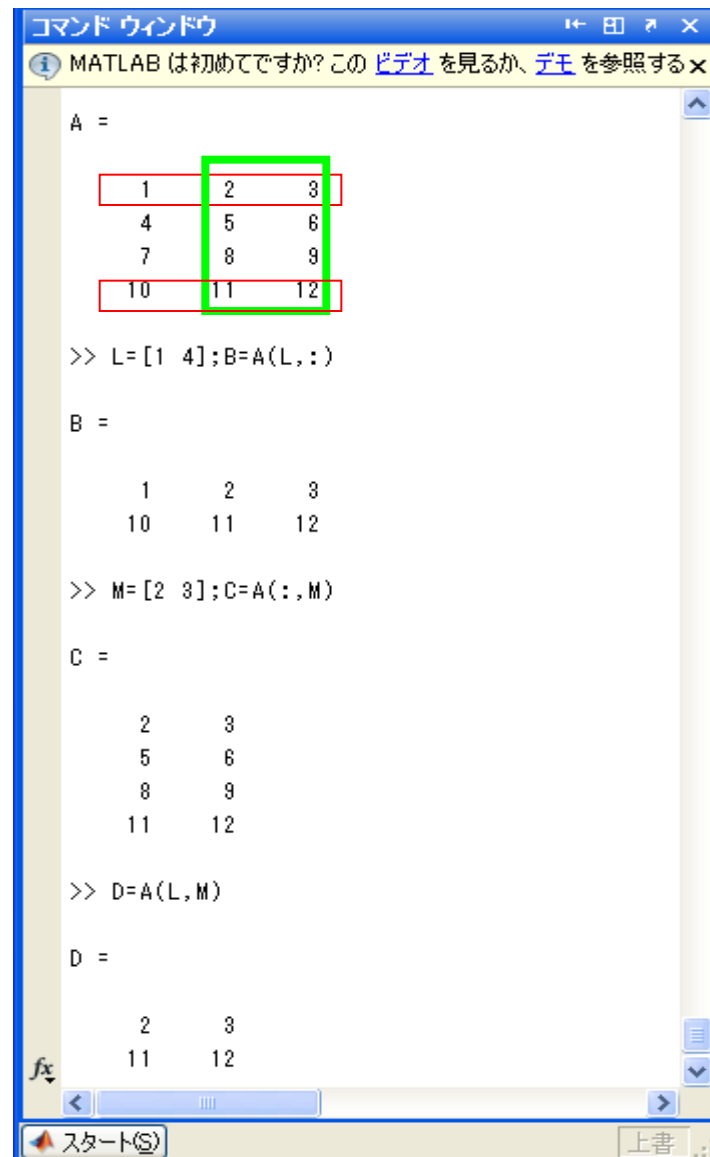
$M=[2\ 3];C=A(:,M)$

$D=A(L,M)$

Aの1行目と4行目  
を取り出す

Aの2列目と3列目  
を取り出す

Aの1行目4行目  
の2列目と3列目を  
取り出す



# 2.4 行列の処理

## 2.4.4 行や列を追加する

追加したい行を行列と一緒に  
行方向に並べる

$A=[1\ 2\ 3;4\ 5\ 6;7\ 8\ 9]$

$B=[10\ 11\ 12]$

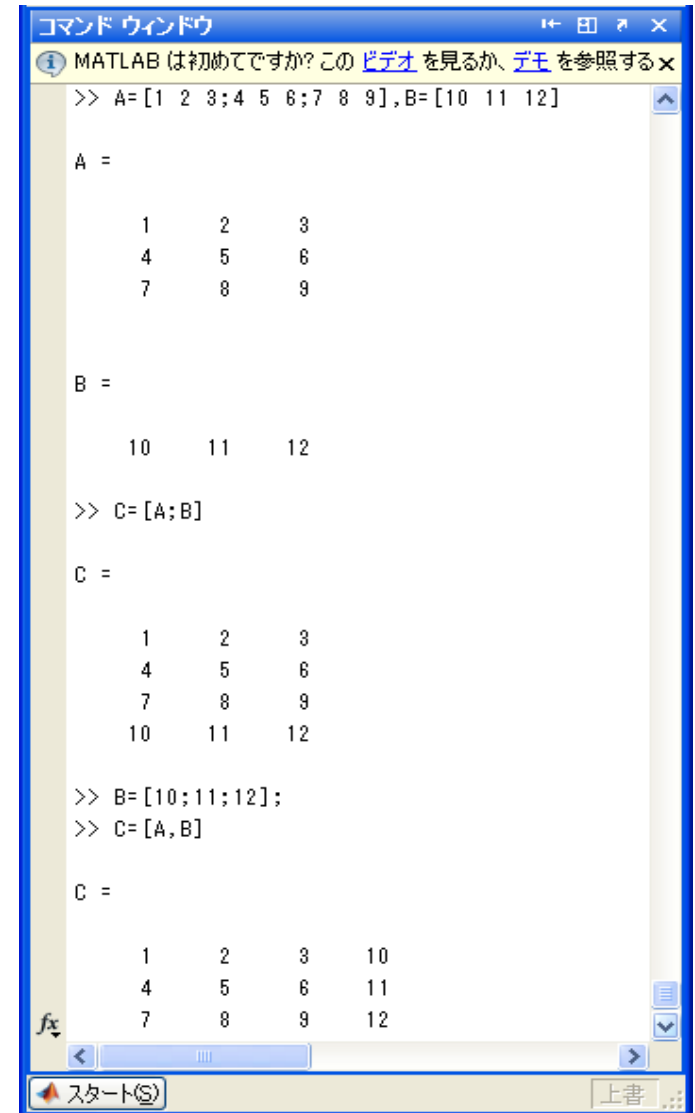
$C=[A;B]$

列も同じ

$A=[1\ 2\ 3;4\ 5\ 6;7\ 8\ 9];$

$B=[10;11;12];$

$C=[A,B]$



コマンド ウィンドウ

MATLAB は初めてですか? この [ビデオ](#) を見るか、[デモ](#) を参照する ×

```
>> A=[1 2 3;4 5 6;7 8 9],B=[10 11 12]
```

A =

1	2	3
4	5	6
7	8	9

B =

10	11	12
----	----	----

```
>> C=[A;B]
```

C =

1	2	3
4	5	6
7	8	9
10	11	12

```
>> B=[10;11;12];
```

```
>> C=[A,B]
```

C =

1	2	3	10
4	5	6	11
7	8	9	12

fx

スタート(S) 上書

# 2.4 行列の処理

## 2.4.5 行または列を取り除く

削除したい行や列に空の行や列をセットする

例

`A=[1:3;4:6;7:9;10:12];`

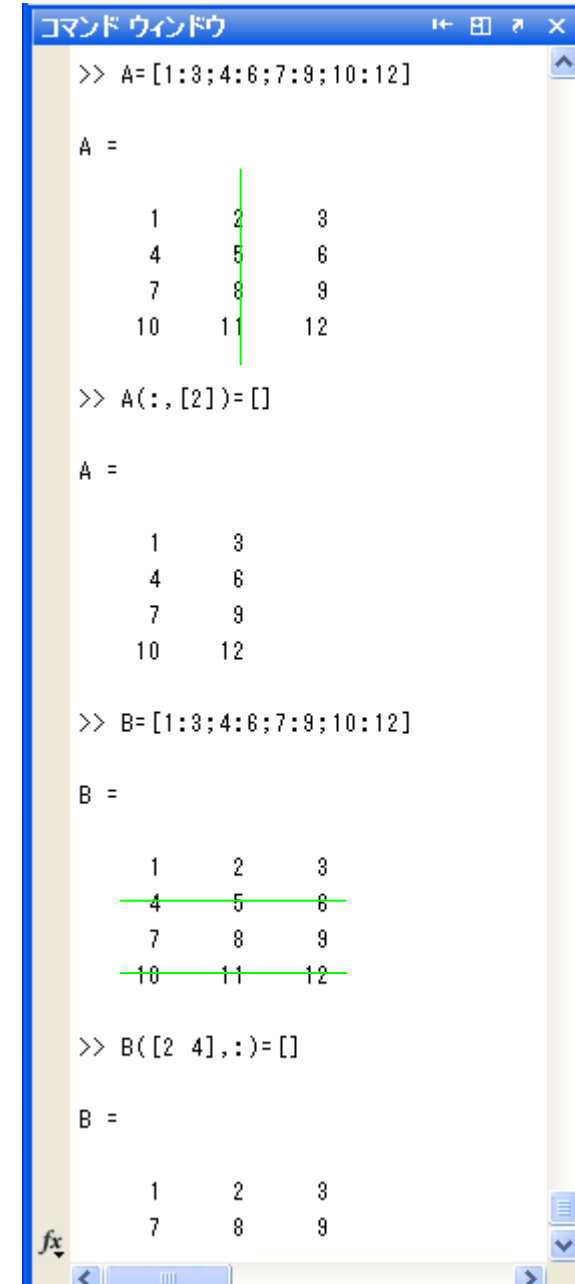
`A(:,[2])=[]`

2列目を抜く

`B=[1:3;4:6;7:9;10:12];`

`B([2 4],:)=[]`

2,4行目を抜く



```
コマンドウィンドウ
>> A=[1:3;4:6;7:9;10:12]

A =

     1     2     3
     4     5     6
     7     8     9
    10    11    12

>> A(:,[2])=[]

A =

     1     3
     4     6
     7     9
    10    12

>> B=[1:3;4:6;7:9;10:12]

B =

     1     2     3
    4     5     6
     7     8     9
    10    11    12

>> B([2 4],:)=[]

B =

     1     2     3
     7     8     9
```

## 2.4 行列の処理

### 2.4.6 行や列を逆順に並べ替える

行や列に逆順に並べた番号をセットする

例

$A = [1 \ 2 \ 3; 4 \ 5 \ 6; 7 \ 8 \ 9; 10 \ 11 \ 12];$

$B = A(4:-1 : 1, :), C = A(:, 3:-1:1)$



## 2.4 行列の処理

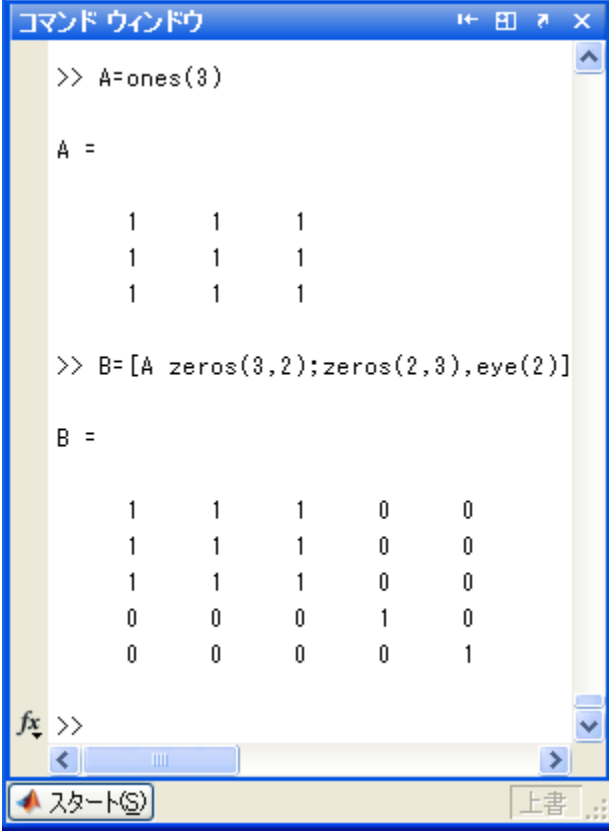
### 2.4.7 小さな行列から大きな行列を作る

小さな行列を大きな行列の成分  
であるかのように配置すればよい

例

```
A=ones(3);
```

```
B=[A zeros(3,2);zeros(2,3) eye(2)]
```



```
コマンドウィンドウ
>> A=ones(3)

A =

     1     1     1
     1     1     1
     1     1     1

>> B=[A zeros(3,2);zeros(2,3) eye(2)]

B =

     1     1     1     0     0
     1     1     1     0     0
     1     1     1     0     0
     0     0     0     1     0
     0     0     0     0     1

fx >>
```

# 2.4 行列の処理

## 2.4.8 行列の成分に施す関数

行列の成分毎に計算

**abs()** : 絶対値

**angle()** : 位相角

**sqrt()** : 平方根

**real()** : 実部

**imag()** : 虚部

**exp()** : 指数関数

**log ()** : 対数関数

**log10()** : 常用対数関数

三角関数 : **sin(),cos(),tan(),asin(),acos(),atan(),atan2(),**など

# 2.4 行列の処理

## 2.4.8 行列の成分に施す関数

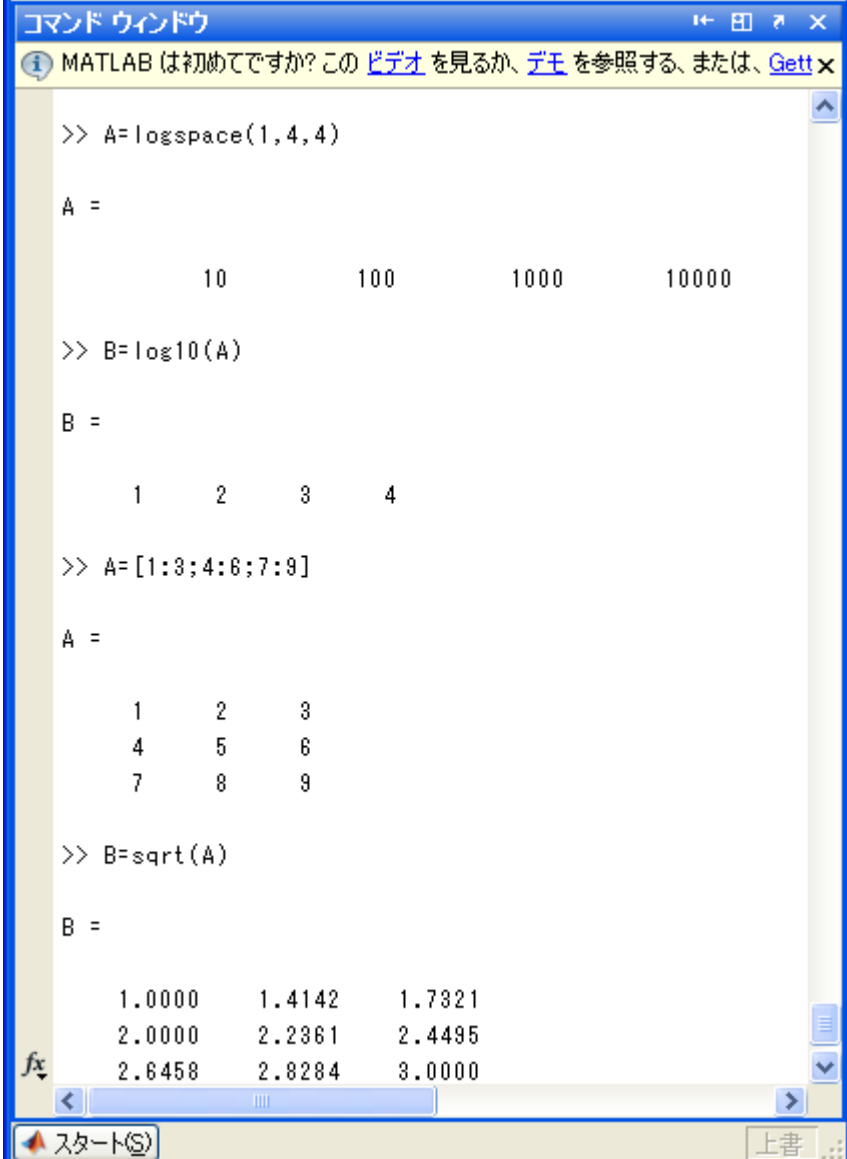
$A = \text{logspace}(1,4,4)$

$y = \text{logspace}(a,b,n)$  は、 $10^a$  と  $10^b$  の間で  $n$  点作る

$B = \text{log}_{10}(A)$

$A = [1:3; 4:6; 7:9]$

$B = \text{sqrt}(A)$



```
コマンド ウィンドウ
MATLAB は初めてですか? この ビデオ を見るか、デモ を参照する、または、Getit ×

>> A=logspace(1,4,4)

A =

    10    100   1000  10000

>> B=log10(A)

B =

    1    2    3    4

>> A=[1:3;4:6;7:9]

A =

    1    2    3
    4    5    6
    7    8    9

>> B=sqrt(A)

B =

    1.0000    1.4142    1.7321
    2.0000    2.2361    2.4495
    2.6458    2.8284    3.0000

fx
スタート(S) 上書
```

## 練習（固有値, 固有ベクトルを求める）

行列[σ]が次式で定義されるものとする

$$\begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{12} & \sigma_{22} & \sigma_{23} \\ \sigma_{13} & \sigma_{23} & \sigma_{33} \end{bmatrix} = \begin{bmatrix} 10 & 5 & 1 \\ 5 & 5 & 7 \\ 1 & 7 & 40 \end{bmatrix}$$

この行列の固有値と固有ベクトルを求めなさい

プログラム例

```
a=[10 5 1; 5 5 7; 1 7 40]
[Q,S]=eig(a)
```

実行すると次の結果を得る.

$$Q = \begin{bmatrix} -0.4730 & 0.8789 & 0.0623 \\ 0.8692 & 0.4538 & 0.1964 \\ -0.1443 & -0.1470 & 0.9785 \end{bmatrix}$$

$$S = \begin{bmatrix} 1.1169 & 0 & 0 \\ 0 & 12.4147 & 0 \\ 0 & 0 & 41.4684 \end{bmatrix}$$

Qの各列が固有ベクトルを, また対応するSの対角項が固有値を表す.

(固有値を**主応力**と呼ぶ  
固有ベクトルはその向きを表す)

## 練習

対角項の和(trace)

行列[A]=[1 2 3; 4 5 6; 7 8 9]  
の対角項の和を取りなさい  
(この演算はトレースと呼ばれる)

$\text{trace}(A)$

または

$b = \text{sum}(\text{diag}(A))$

## 練習

行列[B]=[4 4 6; 1 2 3; 5 8 5]のとき  
左で定義したAを用いて以下の計算を行いなさい

(1)  $\text{trace}(A^T B)$

(2)  $\text{trace}(A B^T)$

(3)  $\text{trace}(B A^T)$

(4)  $\text{trace}(B^T A)$

なお、 $\text{trace}(A B^T)$ は  $A:B$ と書く

## 練習

前ページの[σ]のtraceを計算しなさい。

## プログラム例

$a = [10 \ 5 \ 1; \ 5 \ 5 \ 7; \ 1 \ 7 \ 40]$

$s = \text{trace}(a)$

(備考:  $s/3$  は静水圧と呼ばれ応力の  
普遍量のひとつである)

# 2.5 文字列を扱う

## 2.5.1 文字列をつくる

- 文字列は「`'`」またはダブルクォーテーション「`"`」で阻む。
- 行ベクトルとして扱われる。
- 1文字に1配列

```
X='MATLAB ver.7.0 '
```

したがって`X(1)`は `'M'` である。

- 文字列を二つ以上作成するには行列に格納する。

```
X=[ 'MATLAB ver.5.0 '; 'MATLAB ver.7.0.1']
```

列数が揃うよう空白を入れる必要がある。 . . 不便

そこで. . .

# 2.5 文字列を扱う

## 2.5.1 文字列をつくる

- 列数を自動的に揃えてくれる関数 `char( )`を使う  
`X=char('MATLAB ver.5.0', 'MATLAB ver.7.0.1')`  
行列の大きさを調べる `size( )`を使う  
`size(X)`  
Xは2行16列なので 2 16 と出力される
- `char( )` は行列に行を追加することもできる  
`X=char('MATLAB ver.5.0', 'MATLAB ver.7.0.1')`  
`Y=char(X, 'Third line. ')`
- Yは3行16列になる (`size(Y)`で確認しよう)

# 2.5 文字列を扱う

## 2.5.2 文字列を取り出す

- 文字列の行列から1つの文字列を取り出し、文字列のうしろに付けられた不要なスペースを取り除く組込み関数

`deblank()`

```
X=char('MATLAB ver.5.0', 'MATLAB ver.7.0.1');  
Y=X(1, :), length(Y)  
Z=deblank(X(1, :)), length(Z)
```

`length()`は配列の大きさ（文字数を与える）

Yは末尾の余分な空白まで文字数に入る

Zは末尾の空白がとられた文字数となる



# 2.5 文字列を扱う

## 2.5.3 文字列とASCIIコード

- 文字列をASCIIコード(番号)に変換する関数: `double()`

`X='MATLAB ver.7.0', Y=double(X)`

`Y= 77 65 84 76 65 66 32 118 101 114 46 55 46 48`

- ASCIIコードの数値が格納されている行列を文字列の行列に変換する関数: `char( )`

`Z=char(Y)`

`Z='MATLAB ver.7.0'`

# 2.5 文字列を扱う

## 2.5.4 文字列を比較する

- 比較演算子 `==` を使う.
- 結果は行ベクトルとして得る.

```
S='fate'; T='cake'; C=(S==T)
```

```
C=0101 (2,4文字目が一致する)
```

## 2.5.6 文字を分類する

- アルファベットか判定する関数： `isletter()`

```
X='MATLAB ver.7.0', Y=isletter(X)
```

```
Y=11111101110000
```

- スペースか判定する関数： `isspace()`

```
z=isspace(X)
```

```
z=00000010000000
```

# 2.5 文字列を扱う

## 2.5.7 文字列を検索する

- 文字列の中から指定した文字列を見つける関数： `findstr( )`  
`X='MATLAB and Mathematica'`  
`Y=findstr(X, 'M'), Z=findstr(X, 'Maple')`  
`Y = 1 12`  
`Z = [ ] (なし)`

## 2.5.8 文字列を置換する

- 文字列の中から指定した文字列を見つける関数： `findstr( )`  
`X='MATLAB and Mathematica'`  
`Y=findstr(X, 'M'), Z=findstr(X, 'Maple')`  
`Y = 1 12`  
`Z = [ ] (なし)`

# 2.5 文字列を扱う

## 2.5.9 数値を文字列に変換する

小数を四捨五入し整数として文字列にする関数 : `int2str()`

指定された桁数文字列にする関数 : `num2str(, )`

`X=100.*rand(2, 3), Y=int2str(X), Z=num2str(X, 5)`

`X=`

```
81.4724    12.6987    63.2359
90.5792    91.337     69.7540
```

`Y=`

```
81 13 63
91 91 10
```

`Z=`

```
81.472 12.699 63.236
90.579 91.338 9.754
```

## 2.5 文字列を扱う

### 2.5.10 文字列を数値に変換する

- 文字行列を数値行列に変える関数: `str2num( )`

```
X=char('10.1 10.5 10.8', '20.1 20.5 20.8')
```

```
Y=str2num(X)
```

```
X=
```

```
10.1 10.5 10.8
```

```
20.1 20.5 20.8
```

```
Y=
```

```
10.1000 10.5000 10.8000
```

```
20.1000 20.5000 20.8000
```

# 2.5 文字列を扱う

## 2.5.11 基底変換

- 10進数を16進数や2進数に変換したり, その逆をして表示する関数
    - dec2hex( ) : 10進数を16進数に
    - dec2bin( ) : 10進数を2進数に
    - hex2dec( ) : 16進数を10進数に
    - bin2dec( ) : 2進数を10進数に
- 10進数以外 (16進数, 2進数) は文字列である

X=0: 15


Y=dec2hex(X); Y', X1=hex2dec(Y); X1'

Z=dec2bin(X); Z', X2=bin2dec(Z); X2'

X = 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 (16個の10進数)

Y' = 0 1 2 3 4 5 6 7 8 9 A B C D E F (16個の16進数)

X1 = 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 (16個の10進数)

Z' =  0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 (16個の4桁の2進数(縦方向に見ること) )  
0 0 0 0 1 1 1 1 1 0 0 0 0 1 1 1 1  
0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1  
0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1

X2 = 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 (16個の10進数)

# 2.5 文字列を扱う

## 2.5.12 文字列を実行する

- 数字を文字列に変換する関数: `mat2str()`
- 文字列を数字に変換する関数: `eval()`
- `eval()`はasciiファイル（文字列ファイル）から、文字列として
- 読みだした数字を数値に戻すときなどに重要（詳しくは第6章で）