

# MATLABプログラミング 練習問題

担当：高安 亮紀

# 目的

- 数値シミュレーションにおけるプログラミングにおいて、MATLABでの行列操作（行列の生成，要素の取り出し、演算など）の習得は必須。
- 練習問題を解いて、MATLABの操作方法、関数の作成の仕方を確実に習得しましょう。

※) わからない場合は、第1回、[MATLABの基本的な使い方](#)を参照しながら行うこと。問題のヒントとしてページ番号も示しています。

# 目次

## 1、基本

(1) 1次元配列(ベクトル)の生成・要素の操作  
(代入・絶対値・最大値・最小値)

(2) 2次元配列(行列)の生成・要素の操作

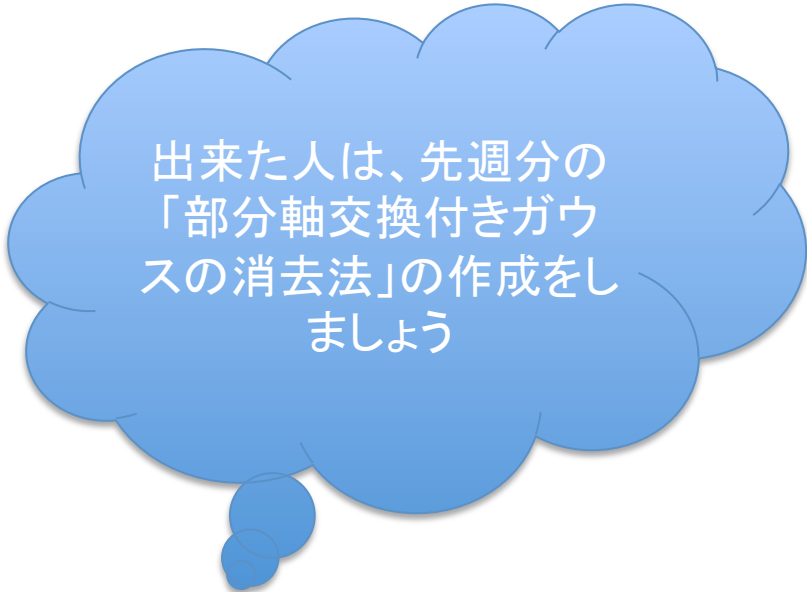
## 2、if文を利用した操作

## 3、for文を利用した操作

(1) ベクトルの総和計算

(2) ベクトルの内積計算

(3) 行列と行列の積



出来た人は、先週分の  
「部分軸交換付きガウ  
スの消去法」の作成をし  
ましょう

# 1次元配列(ベクトル)の生成・要素の操作

1. ベクトル $v = (1, -2, 3, 4, -5)$ を作成しなさい(ベクトルを作成する際に使うカッコに注意すること)。
2. ベクトル $v$ の3番目の要素に0を代入しなさい。
3. ベクトル $v$ の5番目の要素に-2を掛けなさい。
4. ベクトル $v$ の絶対値最大要素、及び絶対値最小要素をそれぞれ求めなさい。

※ヒント1: 要素の指定の仕方は、p.7

※ヒント2: ベクトル $v$ の要素ごとに絶対値を取ったベクトルの求め方は、`abs(v)`

※ヒント3: 最大値、最小値の求め方はp.22

# 演習問題(1)

- 以下のようなベクトルを作成し、それらを用いた計算をする。
  - (1) 自然数 $n$ について、 $x = (1, 2, 3, \dots, n)$ 。ただし、 $n$ は事前に適当な値 ( $n > 5$ ) を与える。
  - (2)  $y = (1, 2, 3, \dots, n)^T$ 。(縦ベクトル)
  - (3)  $z = (1, 0, 0, \dots, 0)$ 。ただし、 $z$ の要素数は $n$ 個。
  - (4)  $x$ の2番目から5番目の要素が並んだベクトル。
  - (5)  $x$ の $n-3$ 番目から $n$ 番目の要素が並んだベクトル。
  - (6)  $x$ の総和。
  - (7)  $x$ と $z$ の和・差・要素ごとの積・内積。
  - (8)  $x * y, z * y, y * x, y * z$

# 2次元配列(行列)の生成・要素の操作

1. 関数randnを利用して、要素が乱数の3×3行列Aを作成しなさい。
2. 行列Aの2行目と3行目を交換しなさい。
3. 行列Aの1列目と3列目を交換しなさい
4. 行列Aの1列目に[1; 2; 3]を代入しなさい。
5. 行列Aの各列の絶対値最大要素、及び絶対値最小要素をそれぞれ求めなさい。

※ヒント1: 要素の指定の仕方は、p.7

※ヒント2: 行列Aの要素ごとに絶対値を取った行列の求め方は、`abs(A)`

※ヒント3: 最大値、最小値の求め方は、p.22

## 演習問題(2)

- 以下のような行列を作成し、それらを用いた計算をする。
  - (1)  $n = 4$  として、 $A = \text{magic}(n)$ のように打ち込む。
  - (2)  $A$ の要素を列ごとにすべて足す。(横ベクトルが作成される)
  - (3)  $A$ の要素を行ごとにすべて足す。(縦ベクトルが作成される)
  - (4)  $m \times m$ 行列 $B$ を作成。ただし、 $m$ は事前に適当に与える。

$$B = \begin{pmatrix} 1 & 1 & & & 1 & 1 \\ & 1 & & & & 1 \\ & & (中はすべて 0) & & & \\ 1 & & & & & 1 \\ 1 & 1 & & & 1 & 1 \end{pmatrix}$$

# 関数の作成(if文)

- テストの点数(0点～100点)に応じて成績を表示するスクリプトscore1.mを作成しなさい。
  - 引数: テストの点数(score)
  - 戻り値: 成績(S,A,B,C,F)
  - 成績の条件: S(90点以上)、A(80点以上90点未満)、B(70点以上80点未満)、C(60点以上70点未満)、F(60点未満)
- コマンドウィンドウでの実行例

```
>> score1(30)
F
>> score1(100)
S
```

※ヒント1: if文については、p.30を参照

# 関数の作成(for文)(1)

- ベクトルを入力すると、その要素の総和を表示する関数sum1.mをfor文を用いて作成しなさい。

- コマンドウィンドウでの実行例

```
>> v=[1 2 3]
```

```
>> sum1(v)
```

```
6
```

※ヒント1: for文の説明はp.26を参照

※ヒント2: ベクトルのサイズは`length(v)`でわかります。(for文の繰り返す回数に使用します。)

※ヒント3: 例えば、`v=[1 2 3]`という3要素のベクトルの場合、

`s = 0, s = s + v(1), s = s + v(2), s = s + v(3)` と3回足し算を繰り返せば総和が求められることがわかります。同じような操作の繰り返しなので、これをfor文を使って表現します。

# 関数の作成(for文)(2)

- 2つのベクトルを入力すると、ベクトルの内積を表示する関数dot1.mをfor文を用いて作成しなさい。

- コマンドウィンドウでの実行例

```
>> a=[1; 2; 3],b=[4; 5; 6]
```

```
>> dot1(a,b)
```

```
32
```

※ヒント1: for文の説明はp.26を参照

※ヒント2: ベクトルのサイズは`length(v)`でわかります。

(for文の繰り返す回数に使用します。)

※ヒント3: 例えば、`a=[1; 2; 3],b=[4; 5; 6]`という3要素のベクトルの場合、

$s = a(1)*b(1) + a(2)*b(2) + a(3)*b(3)$ となります。

※for文を用いた考え方: 例えば  $k$  番目のとき、 $a(k)*b(k)$ を計算し、その計算結果を前( $k-1$ 番目)までの結果に足すということを繰り返します。

# 関数の作成(for文)(3)

- (3)で作成した関数を参考にし、行列とベクトルをそれぞれ入力すると、行列とベクトルの積を表示する関数mtimes1.mをfor文を用いて作成しなさい。
- コマンドウィンドウでの実行例

```
>> A=[1 2 3;4 5 6;7 8 10], b=[4 5 6]
```

```
>> mtimes1(A,b)
```

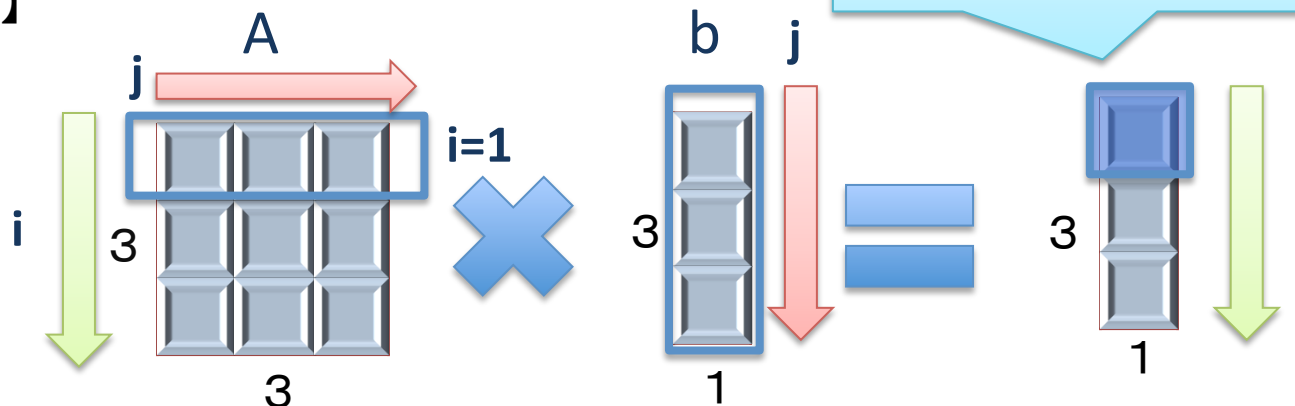
```
ans =
```

```
32
```

```
77
```

```
128
```

【考え方】



1,2,3と変化する箇所  
を変数jに置き換え、  
for文を作ってみよう

$$A(1,1)*b(1)+A(1,2)*b(2)+A(1,3)*b(3)$$

# 関数の作成(for文)(4)

- (3)で作成した関数を参考にし、2つの行列をそれぞれ入力すると、行列の積を表示する関数mtimes2.mをfor文を用いて作成しなさい。

- コマンドウィンドウでの実行例

```
>> A=[1 2 3; 4 5 6; 7 8 10];
```

```
>> B=magic(3);
```

```
>> mtimes2(A,B)
```

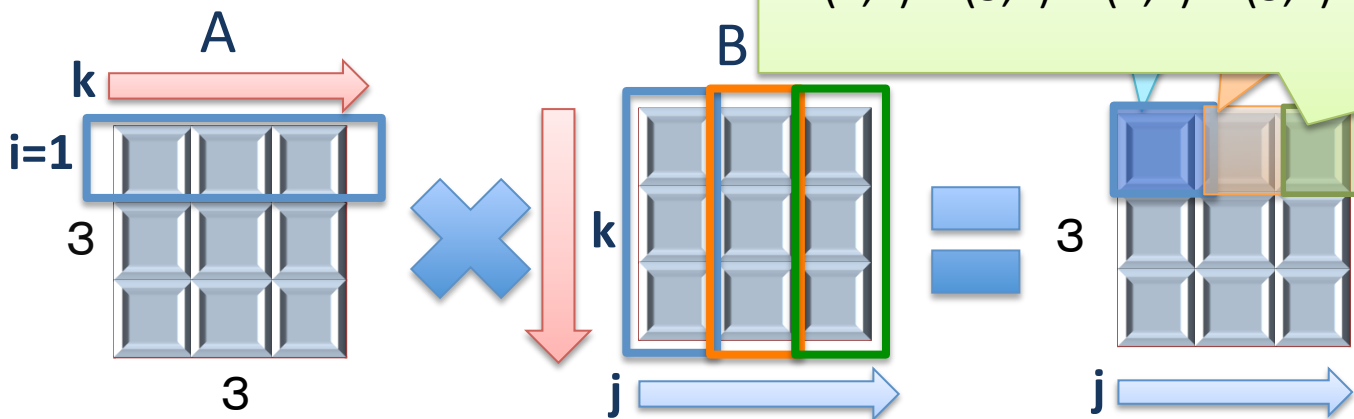
```
ans =
```

```
26 38 26
```

```
71 83 71
```

```
120 137 118
```

【考え方】



(3)の関数をベースに、B  
の列を1,2,3と動かすため  
のfor文を追加しよう

$A(1,1)*B(1,1)+A(1,2)*B(2,1)+A(1,3)*B(3,1)$

$A(1,1)*B(2,1)+A(1,2)*B(2,2)+A(1,3)*B(2,3)$

$A(1,1)*B(3,1)+A(1,2)*B(3,2)+A(1,3)*B(3,3)$