

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 (1)

4-1 論理関係 (論理演算子)

論理演算では、数値を論理値に変換します。

ゼロでない値は1に、ゼロは0になります。

```
a=0.1;  
b=0.0;  
c=logical(a);  
c;
```

a は0でないので、logical(a)=1 です。

a & b %(and) 命題a と命題bについて「a かつ b」が真か偽か

```
ans = logical  
0
```

a | b %(or) 命題a と命題bについて「a か b」の真, 偽

```
ans = logical  
1
```

~a %(not) 命題Not(a) の真偽

```
ans = logical  
0
```

a==b %同値関係. a=b が恒等的に成り立つか否か.

```
ans = logical  
0
```

a==b はよく使うが注意が必要.

```
a=0.33333;  
b=1/3;  
a==b
```

```
ans = logical  
0
```

答は偽 (0) .

```
a-b
```

```
ans = -3.3333e-06
```

```
format long  
a-b
```

```
ans =
```

```
format  
a=pi
```

```
a = 3.1416
```

```
b=vpa(pi,3)
```

```
b = 3.14
```

4-2 if文, for文

If A

if A

B

else

C

end

「もしAが真ならばBを実行せよ, 偽であればCを実行せよ」という条件実行文.

`randi(M)` は1~Mの整数一様乱数を与える.

`randi(M,n)` は1~Mの整数一様乱数を $n \times n$ 行列の形で返す.

```
a = randi(100)
```

```
a = 29
```

```
b = randi(100)
```

```
b = 76
```

```
B= randi(100,5)
```

```
B = 5x5
```

```
76  54  47  32  66  
39  78   2  53  69  
57  94  34  17  75  
 8  13  17  61  46  
 6  57  80  27   9
```

```
if a < 50  
    disp('small')  
else  
    disp('large')  
end
```

```
small
```

```
C=B<50 ;  
C
```

```
C = 5x5 # logical ##  
  0  0  1  1  0  
  1  0  1  0  0  
  0  0  1  1  0  
  1  1  1  0  1  
  1  0  0  1  1
```

for 文

条件が真である間だけ、与えられた実行文を繰り返すための文（loop文）です。

```
for n=0.1:2:5.0  
    a=n  
end
```

```
a = 0.1000  
a = 2.1000  
a = 4.1000
```

for文による計算とベクトル演算の比較

次のコードは三角関数表を作るものです。

for文：

```
clearvars  
x=0;  
tic;  
for k=1:1000000  
    x=x+0.000001;  
    log(x);  
end  
toc
```

経過時間は 0.364424 秒です。

ベクトル演算

```
tic  
x=0.000001:0.000001:1;  
log(x);  
toc
```

経過時間は 0.063385 秒です。

計算に要する時間が計算方法によって簡単に数倍になってしまうことが分かる。

課題

上と同じように、for 文とベクトルを用いた演算の経過時間を確かめよ。