

# プログラミングで理解する数学（ベクトル） - 日経ソフトウェア 2020/09 号

SHIMURA Masato  
jcd02773@nifty.ne.jp

2020年9月10日

## 目次

1	VectorAdd	1
2	VectorProd	2
3	主成分分析	3
4	連立方程式	5

## はじめに

日経ソフトウェア 2020年9月号から「プログラミングで理解する数学」という特集が始まった。今回はベクトルのようだ。

順序は逆のようだが、私も表題のように理解したので異論はない。

原著では python と proceccing で描いているが,J(901) 一本で描いてみよう。

## 1 VectorAdd

2本のベクトルを (a,b) と (x,y) と書く。

$$(a, b) + (x, y) = (a + x, b + y)$$

```
100 50 + 200 50
300 100
vecadd 100 50 ; 200 50
300 100
```

計算は `vecadd` で行い、描画は `draw_vecadd` を用いる。

J はベクトルを変数に持つが、描画はやりにくい。

```
require 'plot numeric trig'
```

```
vecadd=: 3 : 0
```

```
NB. usage:vecadd a b;x y//100 50;200 50
```

```
'x1 x2'=: y
```

```
y1=: +/ x1 ,: x2
```

```
)
```

```
draw_vecadd=: 3 : 0
```

```
NB. draw_vecadd y
```

```
NB. calc
```

```
'x1 x2'=: y
```

```
y1=: vecadd y
```

```
NB. draw
```

```
pd 'new'
```

```
pd ( 0 ,({. x1)); 0 ,({: x1)
```

```
pd ( 0 ,({. x2)); 0 ,({: x2)
```

```
pd (({. x1),{. y1); ({: x1),{: y1
```

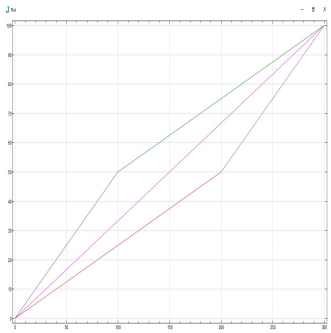
```
pd (({. x2),{. y1); ({: x2),{: y1
```

```
pd (0, { . y1);0 ,{: y1
```

```
pd 'show'
```

```
)
```

```
)
```



入力と描画

```
draw_vecadd 100 50;200 50
```

## 2 VectorProd

3 点のベクトル

$x_1 y_1 = 100, 50$  NB. 2 本のベクトルの共通の始点

$x_2 y_2 = 300, 100$  NB. ベクトル **b** の終点

$x_3 y_3 = 200, 200$  NB. ベクトル **c** の終点

$$(a, b) \cdot (x, y) = ax + by$$

$$(a, b) \cdot (x, y) = |(a, b)| |(x, y)| \cos\theta$$

```
100 50 +/ . * 300 100
```

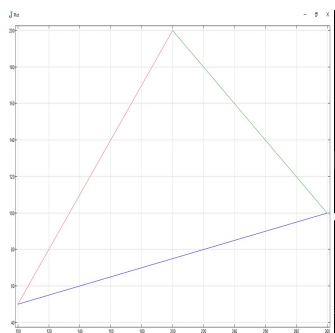
35000

100 50 +/ . \* 200 200

30000

ベクトルの計算はJではお手の物である  
2本のベクトルとその内積

mp=: +/ . \* NB. 内積  
-~ NB. n -~ m -> m - n



(100 50) -~ 300 100  
 200 50 NB. b  
 (100 50) -~ 200 200  
 100 150 NB. c  
  
 200 50 mp 100 150  
 27500 NB. b • c

### 3 主成分分析

応用として主成分分析と連立方程式が上がっている

この例は鈴木義一郎 JAPLA 2009/01に掲載されている

第2回から25回までのミスユニバース日本代表の身長、体重、バスト、ウエスト、ヒップ。(毎日新聞『数学雑学事典』)

STYLE=: ,:165 53 86 56 92  
 STYLE=:STYLE,160 47 84 52 92  
 STYLE=:STYLE,166 55 86 64 89  
 STYLE=:STYLE,164 56 90 60 95  
 STYLE=:STYLE,168 55 87 56 87  
 STYLE=:STYLE,164 54 87 57 92  
 STYLE=:STYLE,168 54 94 58 97  
 STYLE=:STYLE,169 55 88 57 92  
 STYLE=:STYLE,169 53 86 58 93  
 STYLE=:STYLE,166 56 84 57 90  
 STYLE=:STYLE,165 53 85 55 90  
 STYLE=:STYLE,163 49 84 59 90

```

STYLE=:STYLE,164 52 87 58 90
STYLE=:STYLE,167 53 86 59 88
STYLE=:STYLE,169 58 89 60 90
STYLE=:STYLE,169 51 84 60 90
STYLE=:STYLE,166 50 86 59 87
STYLE=:STYLE,168 53 88 60 88
STYLE=:STYLE,165 54 88 62 90
STYLE=:STYLE,167 50 88 58 89
STYLE=:STYLE,170 55 88 60 90
STYLE=:STYLE,168 57 84 62 92
STYLE=:STYLE,168 56 85 62 94
STYLE=:STYLE,163 52 83 60 88

```

```
stand=:3 :0
```

NB. stadadyzation of data

```
s=.y-(k=. $y)$(+/%#)y
```

```
s%k$%:((<0 1)|:(|:s)+/ .*s)%{.k
)
```

```
corm=:3 :'((|:s)+/ .*s)%#s=.stand y'
```

NB. 相関行列を出力する関数

```
pca=:4 :0
```

NB. Principal Component Analysis

```
s=.y-(k=. $y)$(+/%#)y
```

```
s=.s%k$%:((<0 1)|:(|:s)+/ .*s)%{.k
```

```
ev=.x evs((|:s)+/ .*s)%{.k
```

```
t=. 'e-value & vector :',7j3":ev
```

```
t, 'pc-values :',6j2":s+/ .*}.|:ev
```

```
)
```

相関行列

```

corm STYLE
      1 0.542494 0.326565 0.372425 0.0115011
0.542494      1 0.3131 0.449056 0.241926
0.326565 0.3131      1 0.0605705 0.424285
0.372425 0.449056 0.0605705      1 _0.0649716
0.0115011 0.241926 0.424285 _0.0649716      1

```

固有値と固有ベクトル。2つの成分で70%の説明ができる  
それに見合う各データの第一、第2主成分。7番目が児島明子

```
2 pca STYLE
e-value & vector :
  2.149  0.521  0.571  0.420  0.400  0.258
  1.307  _0.237  _0.104  0.483  _0.488  0.679
pc-values :
_0.74  0.94
_4.12  2.03
 0.86  _1.62
 1.37  1.79
 0.00  _0.63
_0.40  1.00
 2.39  3.24
 1.08  0.68
 0.54  0.44
_0.27  _0.44
_1.28  0.37
_2.15  _0.25
_0.89  0.33
_0.26  _0.95
 2.18  _0.38
_0.26  _1.11
_1.24  _1.01
 0.47  _0.84
 0.57  _0.41
_0.62  0.05
 1.55  _0.56
 1.38  _1.08
 1.55  _0.28
_1.71  _1.32
```

#### 4 連立方程式

連立一次方程式は雉兔算として漢の時代に中国で生まれた。

$$\begin{cases} x + y = 40 \\ 2x + 4y = 86 \end{cases}$$

これを拡大クラメル法で解いてみよう

```
cr=: %. }:"1
] d=: 1 1 40,:2 4 86
1 1 40
2 4 86

cr d
1 0 37
0 1 3
```

$$\frac{\begin{pmatrix} 1 & 1 & 40 \\ 2 & 4 & 86 \end{pmatrix}}{\begin{pmatrix} 1 & 1 \\ 2 & 4 \end{pmatrix}}$$

```
(1 1 40,: 2 4 86)%:1 1,: 2 4
```

`cr=: %. }:"1` の前処理で行列の後ろの列を落として、行列除算を行っている