マトリクスの数学と数値計算 (2)
行列の変換

SHIMURA Masato
jcd02773@nifty.ne.jp

2010 年 9 月 17 日

目次

第 1 章

分散共分散行列と相関行列

1.1　分散共分散行列と標準化

1.1.1　標準化

　標準化とは、データを平均０、分散１に変換することであり、単位が異なるデータの回帰や図示に用いる。

$$\frac{x-\overline{x}}{\sqrt{\frac{1}{n}\sum(x-\overline{x})^2}}$$

KX　ある果実の直径 *cm*
KY 水分含有率 (%) のデータ。
（データの出典:金谷「これならわかる応用数学教室」）

```
KX  KY
5.6 30
5.8 26
  6 33
6.2 31
6.4 33
6.4 35
6.4 37
6.6 36
6.8 33
```

```
'key KX KY ' plot |:  stand KD
  pd 'eps /temp/kanaya_02.eps'
```

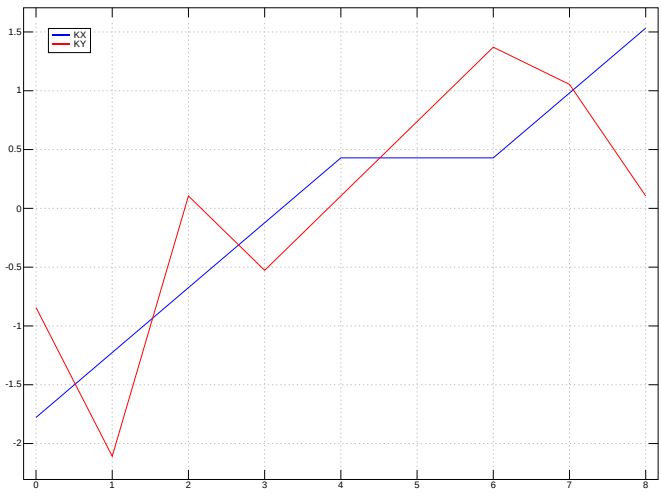


図 1.1　果実の直径 (直線) と水分含有率

6

標準化	$\frac{x-\bar{x}}{\sqrt{\frac{1}{n}\sum(x-\bar{x})^2}}$	偏差（平均を引く）
		$x-\bar{x}$
		`mean=: +/%#`
		`dev=:- mean`
		`dev2=:-"1 mean  NB. rank 1`
		`var=:# %~([:+/[: *: dev)`
		`NB.` **分散**
		`sd=. %:@var NB.` **標準偏差**
		`stand=: dev % "1 sd`

データを平均０、分散１に変換する	`stand KD=. KX,.KY`
	`  _1.7781 _0.843274`
	`_1.22628  _2.10819`
	`_0.674453  0.105409`
	`_0.122628 _0.527046`
	`0.429198  0.105409`
	`0.429198  0.737865`
	`0.429198   1.37032`
	`0.981023   1.05409`
	`1.53285  0.105409`
		単位が異なるものはうまく一枚のグラフには表せないことが多いが標準化すると単位が無くなり、同一のグラフに表すことができる。
		*1

1.1.2 分散共分散行列

数式に書くと厳めしいがスクリプトは単純である。データを予め標準化しておけば直ちに相関行列が求められる。

$$C = \frac{{}^tXX}{N} = \frac{1}{N}\begin{pmatrix} x_1-\bar{x} & x_2-\bar{x} & \dots & x_n-\bar{x} \\ y_1-\bar{y} & y_2-\bar{y} & \dots & y_n-\bar{y} \end{pmatrix}\begin{pmatrix} x_1-\bar{x} & y_1-\bar{y} \\ x_2-\bar{x} & y_2-\bar{y} \\ x_3-\bar{x} & y_3-\bar{y} \\ x_n-\bar{x} & y_n-\bar{y} \end{pmatrix}$$

$$\frac{1}{N}\begin{pmatrix} (x_n-\bar{x})(x_n-\bar{x}) & (x_n-\bar{x})(y_n-\bar{y}) \\ (y_n-\bar{y})(x_n-\bar{x}) & (y_n-\bar{y})(y_n-\bar{y}) \end{pmatrix} = \begin{pmatrix} x\text{の分散} & \text{共分散} \\ \text{共分散} & \text{y の分散} \end{pmatrix}$$

項目/数式	Example
分散共分散行列	(|: dev2) +/ . * dev2 vartable KX,.KY 0.131358 0.792593 0.792593 10
相関行列	cortable KX,.KY 1 0.691547 0.691547 1

■Script

```
dev2=. -"1(+/%#)
vartable=:]# %~ |:@dev2 +/ .* dev2
cortable=: #@] %~ (|:@stand@] +/ . * stand@])
```

■J Grammar

`|:`	*Transpose*	転置
`*:`	*square*	2 乗 (`^2` と同じ)
`%:`	*square_root*	2 乗根 ($\sqrt{}$)
`-"1(+/%#)`	$x - \bar{x}$	残差
`[:`	*Cap*	関数を記述順に実行するように指定
`+/ . *`	*innerproducts*	内積
`]`	*right*	右の値

第 2 章

三角行列への変換

2.1　LU 分解

A が正則ならば

$A = LU$

$A = LDU$

の形に分解できる。

■Example　（例題の出典　木村 P30）

A=:D210A

$$A = \begin{bmatrix} 2 & 1 & 1 \\ -4 & 1 & 3 \\ -2 & 2 & 1r2 \end{bmatrix}$$

単位行列の第 1 列に $-a_{21}/a_{11}$と $-a_{31}/a_{11}$ を入れる

$$T_1 = \begin{bmatrix} 1 & 0 & 0 \\ \mathbf{-2} & 1 & 0 \\ \mathbf{-1} & 0 & 1 \end{bmatrix}$$

内積計算 (T_1 を左からかける)

$$A^{(1)} = T_1 A = \begin{bmatrix} -2 & 1 & 1 \\ 0 & -1 & 1 \\ 0 & 1 & -1r2 \end{bmatrix}$$

単位行列に $A^{(1)}$の $-a_{32}^{(1)}/a_{22}^{(1)}$

$$T_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & \mathbf{1} & 1 \end{bmatrix}$$

内積計算

$$A^{(2)} = T_2 A^{(1)} = \begin{bmatrix} -2 & 1 & 1 \\ 0 & -1 & 1 \\ 0 & 0 & -1r2 \end{bmatrix} = U$$

T_2, T_1 の内積

$$T = T_2 T_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -3 & 1 & 1 \end{bmatrix}$$

T の逆行列

$$L = T^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & -1 & 1 \end{bmatrix} = L$$

$A = LU$

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & -1 & 1 \end{bmatrix} \begin{bmatrix} -2 & 1 & 1 \\ 0 & -1 & 1 \\ 0 & 0 & -1r2 \end{bmatrix}$$

- 数値計算では絶対値最大のピボットをはじめに持ってくるように列を交換する。
- 0 のピボットの除算を避けるように列交換を行う。

2.1.1 J の Library

```
require '~system/packages/math/linear.ijs'
require '~system/packages/math/matfacto.ijs'
```

```
   (<D210), a=.lud D210
+--------+--------+----------+-----+
| 2 1   1|   1 0 0|_4   1    3|0 1 0|
|_4 1   3|_1r2 1 0| 0 3r2  5r2|1 0 0|
|_2 2 1r2| 1r2 1 1| 0   0 _7r2|0 0 1|
+--------+--------+----------+-----+
  D210       L         U       pivot
```

pivot 変換を行っている。

```
D210;   (>{.a) +/ . * >1{a
+--------+--------+
| 2 1   1|_4 1   3|
```

```
|_4 1   3| 2 1   1|
|_2 2 1r2|_2 2 1r2|
+--------+--------+
```

完全に戻すには pivot をかけなければならない

2.2 LDU 分解

先の U を DU に分解する。D は U の対角行列。新しい U は U を D で割ったもの。

LD 分解

$A = LU$

$$\begin{bmatrix}1 & 0 & 0\\ 2 & 1 & 0\\ 1 & -1 & 1\end{bmatrix}\begin{bmatrix}-2 & 1 & 1\\ 0 & -1 & 1\\ 0 & 0 & -1r2\end{bmatrix}$$

U を DU' に分解

$$\begin{bmatrix}-2 & 1 & 1\\ 0 & -1 & 1\\ 0 & 0 & 1r2\end{bmatrix} = \begin{bmatrix}-2 & 0 & 0\\ 0 & -1 & 0\\ 0 & 0 & 1r2\end{bmatrix}\begin{bmatrix}1 & -1r2 & -1r2\\ 0 & 1 & -1\\ 0 & 0 & 1\end{bmatrix}$$

$A = LDU$

$$A = \begin{bmatrix}1 & 0 & 0\\ 2 & 1 & 0\\ 1 & -1 & 1\end{bmatrix}\begin{bmatrix}-2 & 0 & 0\\ 0 & -1 & 0\\ 0 & 0 & 1r2\end{bmatrix}\begin{bmatrix}1 & -1r2 & -1r2\\ 0 & 1 & -1\\ 0 & 0 & 1\end{bmatrix} = LDU$$

```
   u % diag u
1 _1r2 _1r2
0    1   _1 NB. new U
0    0    1
```

■J Grammar

`diag=:(<0 1)&|:`	対角行列を得る	*idiom*
%	÷	*divide*
1r2	*rational*	有理数の表記

2.3 コレスキー分解

And’re Lous Cholesky(1875-1918) ウクライナ系のフランス人北ボルドー生まれ、エコール・ポリテクニーク出身の軍の測量技師でアルジェリアとチュニジアの鉄道建設のための測量に従事。第 1 次大戦中フランス砲兵隊やルーマニア軍の地理業務支援に従事。北フランスでの戦闘で死亡。

12

A が対称かつ正定置ならば下三角行列 L が存在し

$A = LL^t$

$A = LDL^t$

の形に分解できる。さらに分解は対角要素を正になるように定めることで一意に定まる。

対称性に着目し、ガウスージョルダン法のように LU 分解に右からもう 1 ステップ T^t の内積を加える。

■Example

$$A = \begin{bmatrix} 4 & -2 & -6 \\ -2 & 10 & 9 \\ -6 & 9 & 14 \end{bmatrix}$$

単位行列の第 1 列に $-a_{21}/a_{11}, -a_{31}/a_{11}$ を入れる

$$T_1 = \begin{bmatrix} 1 & 0 & 0 \\ \mathbf{1r2} & 1 & 0 \\ \mathbf{3r2} & 0 & 1 \end{bmatrix}$$

2 重の内積計算

$$A^{(1)} = T_1 A T_1^t = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 9 & 6 \\ 0 & 6 & 5 \end{bmatrix}$$

単位行列に $-a_{32}^{(1)}/a_{22}^{(1)}$

$$T_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & \mathbf{-2r3} & 1 \end{bmatrix}$$

2 重の内積計算

$$A^{(2)} = T_2 A^{(1)} T_2^t = \Sigma = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 9 & 0 \\ 0 & 0 & 1 \end{bmatrix} = U^2$$

$A = T^{-1}UU(T^{-1})^t$

$V = T^{-1}U$ とおくと

$A = VV^t$

$$T = T_2 T_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & \mathbf{-2r3} & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ \mathbf{1r2} & 1 & 0 \\ \mathbf{3r2} & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1r2 & 1 & 0 \\ 7r6 & -2r3 & 1 \end{bmatrix}$$

$$T^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ -1r2 & 1 & 0 \\ 3r2 & 2r3 & 1 \end{bmatrix}$$

$$(T_2T_1)^{-1}U = \begin{bmatrix} 1 & 0 & 0 \\ -1r2 & 1 & 0 \\ 3r2 & 2r3 & 1 \end{bmatrix} \begin{bmatrix} \sqrt{4} & 0 & 0 \\ 0 & \sqrt{9} & 0 \\ 0 & 0 & \sqrt{1} \end{bmatrix} = \begin{bmatrix} 2 & 0 & 0 \\ -1 & 3 & 0 \\ -3 & 2 & 1 \end{bmatrix}$$

$$= V(choleski - decomposition)$$

$A = VV^t$

V とその転置行列の組み合わせ

$$\begin{bmatrix} 4 & -2 & -6 \\ -2 & 10 & 9 \\ -6 & 9 & 14 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 0 \\ -1 & 3 & 0 \\ -3 & 2 & 1 \end{bmatrix} \begin{bmatrix} 2 & -1 & -3 \\ 0 & 3 & 2 \\ 0 & 0 & 1 \end{bmatrix}$$

$Ly = b$ を解く

2.4 修正コレスキー分解

LDL 分解のコレスキー版

D220 4 _2 _6 _2 10 9 _6 9 14	v % 2 3 1 1 0 0 _1r3 1 0 _3 2 1 2 3 1=diag V	d0 2 0 0 0 3 0 0 0 1 diag V * I(単位行列)
v 2 0 0 _1 3 0 _3 2 1	c=.|:(|: v % 2 3 1) +/ . * d0 2 0 0 _1 3 0 _3 2 1 choleski D220	c +/ . * |: c 4 _2 _6 _2 10 9 _6 9 14 c=v D220

$DL^{t}x = y$ を解く

2.4.1 J のパッケージ

```
  require '~system/packages/math/matfacto.ijs'

   choleski D220  NB.  A
 2 0 0
_1 3 0
_3 2 1
```

2.5 QR 分解

(*R*) を求めるには元のデータを Gram-shumit のノルム化した直交行列 (*Q*) で割ればよい。1961 年に *Francis* と *Kublanovskaya* が独立にに考案した。[*1]

シュミットの直交化は QR 分解で用いられている。Q,R の左がシュミットの直交化でもある。

J の外部接続詞 128 ！：0 は QR 分解で、左に Q=グラム・シュミットの直交化、右に R=上三角行列が現れる。

■Example .

```
 A=.  D150
1 1  2
1 3 _1
0 1  1


    128!:0 D150
+--------------------------+------------------------+
|0.707107 _0.57735  0.408248|1.41421 2.82843 0.707107|
|0.707107  0.57735 _0.408248|      0 1.73205 _0.57735|
|       0  0.57735  0.816497|      0       0  2.85774|
+--------------------------+------------------------+
  NB.   Q (shumit orthogonal)      R
```

[*1] householder matrix を用いる方法もある。

$Q \times R$

```
'a0 a1'=.   128!:0 D150  NB. Q and R
```

```
 a0 +/ . * a1              NB. Q +/ . * R
1 1  2
1 3 _1
0 1  2
```

R を求める。$\frac{A}{Q} = R$

$$\begin{bmatrix} e_1 & e_2 & e_3 \end{bmatrix} = \begin{bmatrix} v_1 & v_2 & v_3 \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ 0 & b_{22} & b_{23} \\ 0 & 0 & b_{33} \end{bmatrix}$$

```
clean  D150 %. a0   NB. A %. Q = R
    1.41421 2.82843 0.707107
          0 1.73205 _0.57735
          0       0  2.85774
```

2.5.1 QR 法により固有値を求める

QR 分解の左右の内積を何回か反復すると固有値が求められる。（反復解法）

Script は次のように僅か 1,2 行と簡単である。

```
pwr=: ^:100  NB. repeat 100 times (change you like)
eval_t=: (>&{: +/ . * >&{.)&(128 !:0)
```

■Example .

```
                          eval_t pwr D180
    D180
 2 _2  3                          3     0.324443 _0.749269
 1  1  1                          0           _2  _4.04145
 1  3 _1                          0            0         1
```

LF 法で照合

```
   char_lf D180
+-+------+---------+
|1|3 _2 1|6 _5 _2 1|
+-+------+---------+
```

$$f = 6 - 5x - 2x^2 + x^3$$

2.5.2 Grammar

power　関数型定義のループで、慣れると便利である。`^: 100` は 100 回リピート。`^:_` は収束するまでループ。（筋の悪い行列では無限反復もあるので注意）

128!:0　QR 分解の補助関数

clean　numeric.ijs にある塵とり関数

2.6 SVD 変換（特異値分解）

singular value decomposition

J の package/math に *svd.ijs* が入っている。[*2]

ネットで拾った幾つかの声

- 学校では Jordan を教えるが、実務では圧倒的に SVD 法を用いる。
- SVD 法は面長や幅の広い非正方行列に適用できる。
- 金谷の本が唯一わかりやすい。買っておいて良かった。

早速「金谷」を取り出してみた。画像処理の著書を多く書いている関係か SVD は画像処理の項に入っていた。[*3]

$P = U \sum V^T$

$$P = U \begin{pmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_r \end{pmatrix} V^r, \quad \sigma_i = \sqrt{\lambda_i}, \quad i = 1, \cdots, r$$

[*2] LAPACK にも入っている

[*3] $m^2 \times m^2, m = 512$ で 512^4 のサイズでも SVD 分解だと計算が実用の範囲に入るとのことである。

P	$M = PP^T$	$N = P^TP$
D260 3 1 2 3 2 1	M=: D260 +/ . * |: D260 14 13 13 14	N=: (|: D260) +/ . * D260 18 9 9 9 5 4 9 4 5
M と N の 0 でない固有値は一致する。	char_lf M +-+----+--------+ |1|27 1|27 _28 1| +-+----+--------+	char_lf N +-+------+----------+ |1|27 1 0|0 27 _28 1| +-+------+----------+
%: 27 1 5.19615 1	pick_evec M 0.707107 0.707107 0.707107 _0.707107	pick_evec N 0.816497 0 _0.57735 0.408248 _0.707107 0.57735 0.408248 0.707107 0.57735

svd D260

0.707107 _0.707107 0.707107 0.707107	5.19615 1	0.816497 4.44089e_16 0.408248 0.707107 0.408248 _0.707107

U	P の出力の基底となる正規直交ベクトル
$\sum$	$\sum$ は特異値を対角成分に持つ。特異値は増幅率を表し、入力成分がそれぞれ何倍されて出力されるかを表す
V^T	P の入力の基底となる正規直交ベクトル

第 3 章

マトリクスの相似変換

3.1 マトリクスの対角化とジョルダン標準型

オードリー・ヘップバーンのドレスは若き日のジバンシーが担当した。ヘップバーンとジバンシーのドレスの双方を楽しむことができる。

行列 A に左右から正則な行列とその逆行列をかけることを A の相似変換という。

$A' = TAT^{-1}$

相似変換によっても、行列式、トレース、固有値、固有多項式は変わらない。

*1

3.1.1 対角化

各固有値に対応する固有ベクトルを 1 本ずつとる。固有ベクトルは大きさは無視してよい。

Example　　D180

```
2 _2  3
1  1  1
1  3 _1
```

特性方程式と固有値　　char_lf D180

```
+-+------+---------+
|1|3 _2 1|6 _5 _2 1|
+-+------+---------+
```

$$f = 6 - 5\lambda - 2\lambda^2 + \lambda^3$$

固有ベクトル　av=.　pick_evec D180

*1 ローマの休日は担当していない

```
0.57735  0.616849 _0.57735
0.57735 0.0560772  0.57735
0.57735 _0.785081  0.57735
```

対角化 clean (%.av)+/ . * D180 +/ . * av

```
3  0 0
0 _2 0
0  0 1
```

ベクトルの逆行列は左からかける

相似変換と行列式 相似変換前後の行列式の値は同じ

```
   -/ . * D180
_6

 av=.  pick_evec D180
 -/ . * (%. av) +/ . * D180 +/ . * av
_6
```

3.1.2 ジョルダン標準型

線形数学の高峰ジョルダン標準型 (*Jordan canonical form*) の登山口にたどり着けた。

1. 行列の特性方程式が X 軸に見事に接する場合に固有値に重根がでる場合があること。
2. 重根の場合、固有ベクトルが重根と同じ個数取れる場合と、取れない場合があること。

この高峰の登頂に成功したのがジョルダンである。こちらはカミーユ・ジョルダン（Camille Jordan、1838 年-1922）フランスの数学者。リヨン生まれ、エコール・ポリテクニークで学び教える。母の弟は象徴主義の画家 *Chavannes*。[*2]ガロアの群論を出版し発展させた。

父と旅をした若き日のモーツアルトはローマのシスティーナ礼拝堂で秘曲 *9* 声のグレゴリオ聖歌を聴き、宿で書き写し次の日再度聞いて数カ所のチェックで再現したと伝えられる。

相似変換で対角化したマトリクスは対角行列に固有値が並び見事なポリフォニーとなるが、重根を持つマトリクスでは些か不協和音が出る。これを見事なバランスで纏めるのがジョルダン標準形である。

■一斉集合 貴種の行列のチェックにツール類は一斉集合。

[*2] 絵がオルセー美術館の入場券になった

Example　3×3

```
   D261
_1  2  0 _1
 1  2 _1 _2
_1 _1  0  1
 2  6 _2 _5
```

（例題の出典:笠原）

ルベリエ · ファディーエフ法　*4* 重根である

*3

```
+-+-----------+---------+
|1|_1 _1 _1 _1|1 4 6 4 1|
+-+-----------+---------+
```

$$f = 1 + 4x + 6x^2 + 4x^3 + x^4$$

行列式　*0* ではない

```
      -/ . * D261
1
```

J の逆行列　　逆行列は取れる

```
   %. D261
_1 _2  0  1
_1 _4  1  2
 1  1 _2 _1
_2 _6  2  3
```

gausss_elimination/**ガウスの掃き出し**　　*J* のライブラリの *gauss_elimination*

一応通る

```
   gauss_elimination D261
+-----------------+-------+---+
|2 6   _2        _5|1 3 2 0|0 3|
|0 5   _1      _3.5|       |1 3|
|0 0 _0.6      _0.1|       |   |
```

*3 極上の貴富ワイン

```
|0 0    0 _0.166667|       |   |
+-----------------+-------+---+
```

LAPACK　　*LAPACK* も難渋している。

```
require '~addons/math/lapack/lapack.ijs'
require '~addons/math/lapack/dgeev.ijs'

   ,. clean L:0 }. dgeev_jlapack_ D261
+-------------------------------------------------------------+
|_1 _1 _1j4.38896e_8 _1j_4.38896e_8                           |
+-------------------------------------------------------------+
|        0         0 0.227886j_4.34212e_10 0.227886j4.34212e_10|
|_0.408248 _0.408248   0.426309j5.00091e_9 0.426309j_5.00091e_9|
| 0.408248  0.408248 _0.198423j_4.14202e_9 _0.198423j4.14202e_9|
|_0.816497 _0.816497              0.852618             0.852618|
+-------------------------------------------------------------+
```

固有値が違う。(複素数の塵は *clean* では取れない)

対角化　出来ていない。

```
a=.>{: dgeev_jlapack_ D261
   (%. a) +/ . * D261 +/ . *  a
|domain error
|   (    %.a)+/ .*D261+/ .*a
```

多重根　*2* 重根なら運が良ければベクトルが *2* 本取れるが *3* 重根なら絶望と思い知る。

3.1.3　Jordan

4 重根　*EXample* は*-1* の *4* 重根であった。次の何れかになる。

$$\left(\begin{array}{cc|cc} \lambda_1 & 1 & & \\ & \lambda_1 & & \\ \hline & & \lambda_1 & 1 \\ & & & \lambda_1 \end{array}\right), \left(\begin{array}{ccc|c} \lambda_1 & 1 & & \\ & \lambda_1 & 1 & \\ & & \lambda_1 & \\ \hline & & & \lambda_1 \end{array}\right), \left(\begin{array}{cc|c|c} \lambda_1 & 1 & & \\ & \lambda_1 & & \\ \hline & & \lambda_1 & \\ \hline & & & \lambda_1 \end{array}\right)$$

$\lambda = -1$ **を入れる**　$A-(-1)I, (A-(-1)I)^2$ を求める

$A-(-1)I$=	$(A-(-1)I)^2$=
a1=. D261 - _1 * =/˜i.4 0 2 0 _1 1 3 _1 _2 _1 _1 1 1 2 6 _2 _4 a1=._1 small_pol D261	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 a1 +/ . * a1 漂数は *2*

ランク　小行列式からランクを *2* と推測する

```
  small_matrix a1=. _1 small_pol D261
+-----------+--------+-----+
| 0  2  0 _1| 3 _1 _2| 1  1|
| 1  3 _1 _2|_1  1  1|_2 _4|
|_1 _1  1  1| 6 _2 _4|     |
| 2  6 _2 _4|        |     |
+-----------+--------+-----+
|0          |0       |_2   |
+-----------+--------+-----+
```

*4

マジシャン　ランクを見ながら *2* 本の固有ベクトルをつかみ取る。$A-(-1)I$ にかける

```
   a1 +/ . * 0 1 1 1
1 0 1 0
   a1 +/ . * 0 1 0 1
1 1 0 2
```

固有ベクトル　*P*

---

*4 グラム・シュミットの直交化も効かない

```
   128!:0 a
|domain error
|       128!:0 a
```

```
   ev=. 1 0 1 0 ,. 0 1 1 1,. 1 1 0 2 ,. 0 1 0 1
    ev
1 0 1 0
0 1 1 1
1 1 0 0
0 1 2 1
```

対角化 $P^{-1}AP$

```
   (%. ev) +/ . * D261 +/ . * ev
_1  1  0  0
 0 _1  0  0
 0  0 _1  1
 0  0  0 _1
```

しかしながら任意の固有ベクトルをつかみ取る方法はなかなか難しい。P を巧く取らないと重複固有値は対角に来るが *1* 以外の数字が散乱することになる。

3.2 スペクトル分解

行列を射影の一次結合で表すことをスペクトル分解という。
マトリクスの対角化と同様で

- 単根の場合
- 重根の場合
  対角化の場合は重根があっても重根の固有ベクトルが重複相当分取れれば対角化でき、そうでない場合はジョルダン標準形になった。
  スペクトル分解でも同様で次のように区分できる
  - 重根があってもスペクトル分解できる場合
    1 の重根に関係するスペクトルは一個である
  - , 部分分数展開による場合

行列 A がスペクトル分解可能のとき、その射影行列は $P_1, \cdots .P_r$ は次により求められる。

$$P_k = \frac{(A-\lambda_1 I)\cdots\underbrace{\quad k \quad}\cdots(A-\lambda_r I)}{(\lambda_k-\lambda_1)\cdots\underbrace{\quad k \quad}\cdots(\lambda_k-\lambda_r)}$$

$$P_1 = \frac{(A-\lambda_2 I)(A-\lambda_3 I)}{(\lambda_1-\lambda_2)(\lambda_1-\lambda_3)}$$

$$P_2 = \frac{(A-\lambda_1 I)(A-\lambda_3 I)}{(\lambda_2-\lambda_1)(\lambda_2-\lambda_3)}$$

$$P_3 = \frac{(A-\lambda_1 I)(A-\lambda_2 I)}{(\lambda_3-\lambda_1)(\lambda_3-\lambda_2)}$$

3.2.1 単根の場合

■Example

```
a=.2 _2 3,1 1 1,:1 3 _1
2 _2  3
1  1  1
1  3 _1
```

```
   char_lf a
+-+------+---------+
|1|3 _2 1|6 _5 _2 1|
+-+------+---------+
```

$fx = 6 - 5\lambda - 2\lambda^2 + \lambda^3$

Eigenvalue: 3 -2 1

$$P_1 = \frac{(A-\lambda_2 I)(A-\lambda_3 I)}{(\lambda_1-\lambda_2)(\lambda_1-\lambda_3)} = \frac{1}{(3-(-2))(3-1)}\begin{bmatrix}4 & -2 & 3\\ 1 & 3 & 1\\ 1 & 3 & 1\end{bmatrix}\begin{bmatrix}1 & -2 & 3\\ 1 & 0 & 1\\ 1 & 3 & -2\end{bmatrix} = \frac{1}{10}\begin{bmatrix}5 & 1 & 4\\ 5 & 1 & 4\\ 5 & 1 & 4\end{bmatrix}$$

$$P_2 = \frac{(A-\lambda_1 I)(A-\lambda_3 I)}{(\lambda_2-\lambda_1)(\lambda_2-\lambda_3)} = \frac{1}{(-2-3)(-2-1)}\begin{bmatrix}-1 & -2 & 3\\ 1 & -2 & 1\\ 1 & 3 & -4\end{bmatrix}\begin{bmatrix}1 & -2 & 3\\ 1 & 0 & 1\\ 1 & 3 & -2\end{bmatrix} = \frac{1}{-1}\begin{bmatrix}0 & 11 & -11\\ 0 & 1 & -1\\ 0 & -14 & 14\end{bmatrix}$$

$$P_3 = \frac{(A-\lambda_1 I)(A-\lambda_2 I)}{(\lambda_3-\lambda_1)(\lambda_3-\lambda_2)} = \frac{1}{(1-3)(1-(-2))}\begin{bmatrix}-1 & -2 & 3\\ 1 & -2 & 1\\ 1 & 3 & -4\end{bmatrix}\begin{bmatrix}4 & -2 & 3\\ 1 & 3 & 1\\ 1 & 3 & 1\end{bmatrix} = \frac{1}{-6}\begin{bmatrix}-3 & 5 & -2\\ 3 & -5 & 2\\ 3 & -5 & 2\end{bmatrix}$$

- $A - \lambda I$ の組み合わせ

```
   MAT
+--------+------+-------+
|_1 _2  3|4 _2 3|1 _2  3|
| 1 _2  1|1  3 1|1  0  1|
| 1  3 _4|1  3 1|1  3 _2|
+--------+------+-------+
```

- λ の組み合わせ

```
   LAMBDA2
+------+------+------+
|3 _2 1|_2 3 1|1 3 _2|
+------+------+------+
```

- spectol a

```
+-----------+--------------+-------------+
|1r2 1r10 2r5|0  11r15 _11r15| 1r2 _5r6  1r3|
|1r2 1r10 2r5|0   1r15  _1r15|_1r2  5r6 _1r3|
|1r2 1r10 2r5|0 _14r15  14r15|_1r2  5r6 _1r3|
+-----------+--------------+-------------+
```

（基準化は行っていない。）

各固有値毎に異なる固有ベクトルを一本選べば対角化できる。

スペクトル分解は

$A = \lambda_1 P_1 + \lambda_2 P_2 + \lambda_3 P_{31}$

となる。

LF 法で求めた固有ベクトルと比較すると *LF* 法は分子の部分と同一である。

```
  char_evec a
+-----+---------+--------+
|3    |_2       |1       |
+-----+---------+--------+
|5 1 4|0  11 _11|_3  5 _2|
|5 1 4|0   1  _1| 3 _5  2|
|5 1 4|0 _14  14| 3 _5  2|
+-----+---------+--------+
```

3.2.2 重根でもスペクトル分解できる場合

■Worked Example

$$A = \begin{matrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{matrix}$$

出典　笠原対話 *p68 (D8)*

```
  char_lf D8
+-+-------+---------+
|1|2 _1 _1|_2 _3 0 1|
+-+-------+---------+
```

$$f = -2 - 3x + x^3$$

固有値 *2 -1 -1*

$$P_1 = f(\lambda) = \frac{\lambda + 1}{2 + 1} = \frac{1}{3}(\lambda + 1)$$

$$P_1 = \frac{1}{3}(A + I) = \frac{1}{3}\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

$$P_2 = f(\lambda) = \frac{\lambda - 2}{-1 - 2} = \frac{1}{3}(\lambda - 2)$$

$$P_2 = -\frac{1}{3}(A - 2I) = -\frac{1}{3}\begin{pmatrix} -2 & 1 & 1 \\ 1 & -2 & 1 \\ 1 & 1 & -2 \end{pmatrix}$$

```
   spectol D8
+-----------+--------------+
|1r3 1r3 1r3| 2r3 _1r3 _1r3|
|1r3 1r3 1r3|_1r3  2r3 _1r3|
|1r3 1r3 1r3|_1r3 _1r3  2r3|
+-----------+--------------+
```

$$e^{Mt} = \frac{1}{3}\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} e^{2t} \frac{1}{3}\begin{pmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{pmatrix} e^{-t}$$

3.2.3 部分分数展開を用いたスペクトル分解

重根がある場合は部分分数展開を行って後、スペクトル分解のジョルダン標準型版と言うべき形に持ち込む。

$$\frac{1}{\Phi(\lambda)} = \frac{g_1(\lambda)}{(\lambda - \lambda_1)^{m1}} + \cdots + \frac{g_r(\lambda)}{(\lambda - \lambda_r)^{mr}}$$

■Example *issue:* 笠原

```
   D330
 2  0 1
_1  3 1
 1 _1 2
```

28

1. 固有値を求める

```
   char_lf D330
+-+-----+-----------+
|1|3 2 2|_12 16 _7 1|
+-+-----+-----------+
```

$$f = -12 + 16x - 7x^2 + x^3$$

固有値　3 2(重根）

2. 重根は部分分数展開が必要

```
 mp_insert  Lamda_I
+-------+
|1 _1 _1|
|1 _1 _1|
|0  0  0|
+-------+
```

$(A-3I)(A-2I)$ が 0 行列と成らない

3. 部分分数展開

$$\frac{1}{(\lambda-2)^2(\lambda-3)}$$

$$\frac{1}{(\lambda-2)^2(\lambda-3)} = \frac{A}{\lambda-2} + \frac{B}{(\lambda-2)^2} + \frac{C}{\lambda-3}$$

- 分母を払う

$$1 = A(\lambda-2)(\lambda-3) + B(\lambda-3) + C(\lambda-2)^2$$

- 連立方程式に組み替える

$$1 = (A+C)\lambda^2 + (-5A+B-4C)\lambda + 6A - 3B + 4C$$

$$\begin{cases} A & & +C & = 0 \\ -5A & +B & -4C & = 0 \\ 6A & -3B & +4C & = 1 \end{cases}$$

- 拡大係数行列

*5

$$\begin{bmatrix} 1 & 0 & 1 & 0 \\ -5 & 1 & -4 & 0 \\ 6 & -3 & 4 & 1 \end{bmatrix}$$

- 連立方程式を解く

```
   a=.1 0 1 0 ,_5 1 _4 0,: 6 _3 4 1
   cr=: %.}:"1    NB.Cramer Method
   cr a
1 0 0 _1
0 1 0 _1
0 0 1  1
```

- 部分分数に戻して通分する

$$\frac{1}{(\lambda-2)^2(\lambda-3)} = \frac{-1}{\lambda-2} + \frac{-1}{(\lambda-2)^2} + \frac{1}{\lambda-3} = \frac{-\lambda+1}{(\lambda-2)^2} + \frac{1}{\lambda-3}$$

4. スペクトルを求める

（*a*）部分分数展開から

$P_1 = (-A + I)(A - 3I)$

$P_2 = (A - 2I)^2$

$$P_1 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ -1 & 1 & 1 \end{bmatrix}$$

$$P_2 = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 0 & 0 \\ 1 & -1 & 0 \end{bmatrix}$$

（*b*）$S = 2P_1 + 3P_2$

$$2P_1 + 3P_2 = \begin{bmatrix} 3 & -1 & 0 \\ 0 & 2 & 0 \\ 1 & -1 & 2 \end{bmatrix}$$

（*c*）$N = (A - 2I)P_1$

$$(A - 2I)P_1 = \begin{bmatrix} -1 & 1 & 1 \\ -1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

（*d*）$A = S + N$

*5 移項ではないので右の項の符号は変更しない

$$\begin{bmatrix} 2 & 0 & 1 \\ -1 & 3 & 1 \\ 1 & -1 & 2 \end{bmatrix} = \begin{bmatrix} 3 & -1 & 0 \\ 0 & 2 & 0 \\ 1 & -1 & 2 \end{bmatrix} + \begin{bmatrix} -1 & 1 & 1 \\ -1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

3.3　2 次形式

$$A = \begin{bmatrix} a & b \\ b & c \end{bmatrix}, X = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

■Example　$x_1^2 + x_2^2 + x_3^2 + 4x_1x_2 + 4x_2x_3 + 4x_1x_3$

$$A = \begin{bmatrix} 1 & 2 & 2 \\ 2 & 1 & 2 \\ 2 & 2 & 1 \end{bmatrix}$$

```
   b1
1 2 2
2 1 2
2 2 1
```

固有値が良くないので *LAPACK* で固有ベクトルを求める

```
require '~addons/math/lapack/lapack.ijs'
require '~addons/math/lapack/dgeev.ijs'

 ] ev=. ;{: dgeev_jlapack_ b1
 0.816497 _0.57735         0
_0.408248 _0.57735 _0.707107
_0.408248 _0.57735  0.707107
```

```
   char_lf b1
+-+-------+----------+
|1|5 _1 _1|_5 _9 _3 1|
+-+-------+----------+
```

$f(x) = -5 - 9\lambda - 3\lambda^2 + \lambda^3 = 0$

固有値　*5, -1(*重根）

対角化

```
   (|: ev) +/ . * b1 +/ . * ev
        _1 8.88178e_16 1.66533e_16
5.55112e_16           5 1.66533e_16
1.11022e_16 4.44089e_16          _1
```

$$\begin{bmatrix} -1 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

次の標準形が得られる。

$$x^t Ax = -y_1^2 + 5y_2^2 - y_3^2$$

3.4 シューア変換

実対称行列を上三角行列に変換する方法にシューア変換がある。例題の 3×3 行列は *3* 重根で、ノルム化した後に残るベクトルは *1* 本のみであって、まさに「包丁一本晒しに巻いて」である。

I.S chur(1875 – 1941)

■Examples

```
   A2
6 4 _3
1 4 _1
4 5 _1

   char_lf A2
+-+-----+-----------+
|1|3 3 3|_27 27 _9 1|
+-+-----+-----------+
```

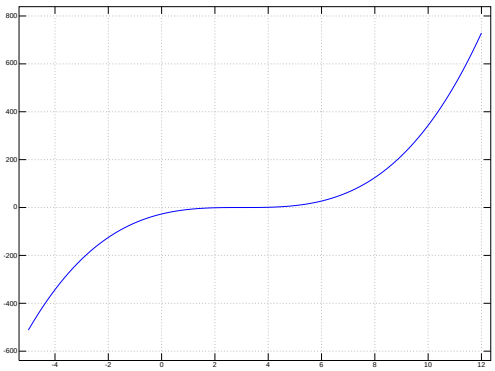
特性方程式

$$f = \_27 + 27\lambda + \_9\lambda^2 + \lambda^3$$

有用な固有ベクトルは得られない

```
   char_evec0 A2
+------+------+------+
|1 1 _1|1 1 _1|1 1 _1|
|0 0  0|0 0  0|0 0  0|
|1 1 _1|1 1 _1|1 1 _1|
+------+------+------+

  plot _5 12;'_27 27 _9 1&p.'
```



*6

3.4.1 部品の作成

ノルム化

縦ベクトル ベクトルは縦ベクトルを基本とする。関数によっては縦積み（横ベクトル）で働くものもあるので、トランスポーズ (|:) を行う。

clean ~. で重複が排除できるが *0* にゴミが残っていると正確さが損なわれる。*numeric.ijs* に入っている *clean* で掃除する。

$\| u \| = \sqrt{\sum 6^2, 1^2, 4^2}$	`%: +/ ^&2 ] 6 1 4` `7.28011`

$\frac{u_1}{\| u_1 \|} = \frac{6\ 1\ 4}{\sqrt{\sum 6^2, 1^2, 4^2}}$	`6 1 4 % (%: +/ ^&2 ] 6 1 4)`
	`0.824163 0.137361 0.549442`
	
	`norm0 6 1 4`
	`0.824163 0.137361 0.549442`

3.4.2 グラム · シュミットの直交化

*6 3 重根は実データでは先ず出てこない。先達が苦労して見つけたもの

$$v_1 = x_1$$
$$v_2 = x_2 - \frac{x_2 v_1}{v_1 v_1} v_1$$
$$v_3 = x_3 - \frac{x_3 v_1}{v_1 v_1} v_1 - \frac{x_3 v_2}{v_2 v_2} v_2$$
$$\cdots$$
$$v_p = x_p - \frac{x_p v_1}{v_1 v_1} v_1 - \frac{x_p v_2}{v_2 v_2} v_2 - \cdots - \frac{x_p v_{p-1}}{v_{p-1} v_{p-1}} v_{p-1}$$

グラム・シュミットの直交化は *J* の 128!:0 にノルム化されたものが入っている。ここではノルム化しないものを作成した。多少ゴチャゴチャしている。

```
   A3
1 1  2
1 0 _1
0 1  1
   norm2 gs0 A3
0.707107  0.408248  0.57735
0.707107 _0.408248 _0.57735
       0  0.816497 _0.57735
```

0 ベクトルの取り除き

ノルム化すると *0* ベクトルが出ることがあるので除く。

```
removeZero=: 3 : '(-.NR = +/"1 (clean y) e. (NR=.{:$y)# 0)# y'
NB. remove  0 0 0
```

直交ベクトル

マトリクスのサイズ分の直交ベクトルが無いときに乱数から作成する。重複や *0* ベクトルを慎重に除く。

Example A2　は　固有値は *3* の *3* 重根でこ固有ベクトルは *1* 本のみである。

$$A2 =: \begin{matrix} 6 & 4 & -3 \\ 1 & 4 & -1 \\ 4 & 5 & -1 \end{matrix}$$

```
A2;  ~. open2 char_evec0 A2
+------+------+
|6 4 _3|1 1 _1|
|1 4 _1|0 0  0|
|4 5 _1|      |
```

```
+------+------+
 A2       evec
```

乱数から任意のベクトルを作成し、直交化とノルム化を行う。

```
  fo A2
 0.57735 0.293294  0.762001
 0.57735 0.513265 _0.635001
_0.57735 0.806559     0.127
```

3.4.3 シューア変換

固有値が異なる場合はマトリクスの各固有値から取った固有値の次数分の直交ベクトルがあれば一気に対角化ができる。
シューア変換は *1* 個の固有値にかかる *1* 本のノルム化された固有ベクトル (X_1) とこれに直交するノルム化された $n-1$ 本のベクトル (W_2, W_3) を用いる。ジョルダン標準形の様に大げさでなく、シューア変換は *Example A2* の様に単一の重根の場合に重宝である。

$T_1 = [X_1, W_2, W_3]$

$$T_1^t A T_1 = \left[\begin{array}{c|cc} \lambda_1 & b_{12} & b_{13} \\ \hline 0 & b_{22} & b_{23} \\ 0 & b_{32} & b_{33} \end{array}\right] = \left[\begin{array}{c|c} \lambda_1 & b^t \\ \hline 0_2 & A_2 \end{array}\right]$$

A_2 以下に同じ作用を 2×2 まで行う。T_2 の首座に *1*、他を *0* を付加しサイズを戻す。

$$T_2 = \left[\begin{array}{c|c} 1 & 0_2^t \\ \hline 0_2 & S_2 \end{array}\right]$$

変換マトリクスを作成する。
$U = T_1 T_2$ *NB.* ユニタリマトリクス

シューア変換
$U^t A U$
*7

```
  shur0 A2
```

*7 対角化のフォームは $U^{-1}AU$ である。直交行列であるので逆行列と転置は同一

```
+-------------------------------+-----------------+
| 0.57735 _0.408248     0.707107|3 2.12132 9.38971|
| 0.57735  0.816497 _2.08167e_16|0       3  1.1547|
|_0.57735  0.408248     0.707107|0       0       3|
+-------------------------------+-----------------+
```

Shur **変換で用いたベクトル**　　Shur **変換後の上三角行列**
固有値が対角行列に並ぶ

shur 変換後の上三角行列は元の行列 *(A2)* と特性方程式、固有値は同じ

```
    char_lf A2                      char_lf ;{: shur0 A2
+-+-----+-----------+             +-+-----+-----------+
|1|3 3 3|_27 27 _9 1|             |1|3 3 3|_27 27 _9 1|
+-+-----+-----------+             +-+-----+-----------+
```

第 4 章

アフィン変換

Affin transform　アフィン変換
平行移動、スケーリング（拡大 · 縮小), 回転、シェアリングを同時に行う。

図 4.1　fern

4.0.4　バーンスレーの羊歯/2 次元

次の行列 (アフィン変換行列）と並進ベクトルで表現できる。

$$w(x) = w\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix} = Ax + t$$

θ_n は反時計回りの極座標での回転

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} r_1 cos\theta_1 & -r_2 sin\theta_2 \\ r_1 sin\theta_1 & r_2 cos\theta_2 \end{pmatrix}$$

Burnslay のパラメータ

バーンスレーの *IFS* のパラメータは次のような表に記載されている。

Burnslay' s fern

parameter of fern by Burnsray

w	*a*	*b*	*c*	*d*	*e*	*f*	*p*
1	0	0	0	0.16	0	0	0.01
2	0.85	0.04	−0.04	0.85	0	1.6	0.85
3	0.2	−0.26	0.23	0.22	0	1.6	0.07
4	−0.15	0.28	0.26	0.24	0	0.44	0.075

p は点を打つ確率（出現頻度）

parameter

```
  FERN
   0     0     0 0.16 0    0 0.01
0.85  0.04 _0.04 0.85 0  1.6 0.85
 0.2 _0.26  0.23 0.22 0  1.6 0.07
_0.15  0.28  0.26 0.24 0 0.44 0.07
```

calc

```
   10 calc_ifs0 FERN
        0     1.6
   _0.416   1.952
 _0.27552 3.27584
_0.103158 4.39548
0.0881348 5.34029
 0.288526 6.13572
 0.490676 6.80382
 0.689227 7.36362
 0.880388 7.83151
  1.06159 8.22157
   2.1428 2.68919
```

50000 の点で描画する。

```
view_pixel 50000 calc_ifs0 FERN
```

4.0.5 Script

```
NB. -----section 3.4------------
   calc_ifs0=:4 : 0
NB. Usage: 100 calc_ifs0 FERN
X0=. x
Y0=. (\:({:"1 y)){ y NB. sort by prob
AD=. 2 2 $ L:0 { 4{."1 Y0 NB. a b c d/box
EF=. {4 5{"1 Y0 NB. e,f/box
RND=. ( ? 100)% 100 NB. rand
P0=.0.01 - ~ +/ \{:"1 Y0 NB. probability/0.01 adjustment origin
ANS=.< TMP=.(;{.EF) +(>{. AD) +/ . *"(1) 0 0 NB. first
for_ctr. i. X0 do.
IND=.{.(((?100)%100) < P0) # i. # y
TMP=. (;IND{EF)+  (>IND{AD) +/ . * "1 TMP
ANS=. ANS,<TMP
end.
;("1),. ANS
)

   view_pixel=:3 : 0
gopen ''
gclear''
glpixel adj_pixel_sub y
)
   adj_pixel_sub=:3 : 0
TMP=. y
MINMAX=. ;("1),.(<./y); >./y
TMP=. TMP +  -<./ {. MINMAX
MINMAX=. ;("1),.(<./TMP); >./TMP
TMP * TIMES=. <.<./1000 % {: MINMAX
)
```

References

Michael F. Barnsley [Fractals Everywhere 2nd Edition] 1993 Morgan-Kaufmann
笠原 皓司「行列の構造」日本評論社 *1994*
笠原 皓司「新微分方程式対話（新版）」日本評論社 *1995*
木村英紀「線形代数」東京大学出版会　*2003*

Miscellance

J(602) is download available
`http://www.jsoftrare.com`
WIN32/64 Linux32/64 MAC
Document and Script is also DL available
`http://japla.sakura.ne.jp`
`http://homepage3.nifty.com/asagaya_avenue`