固有ベクトル計算法

志村氏論文の理解の為に

中野 嘉弘 (札幌市、後3年で91歳)

FAX **専** 011-588-3354

E-mail: yoshihiro@river.ocn.ne.jp

はしがき

Yahoo 知恵袋・数学カテゴリーは、年寄りのホビーとして最適である。

固有値問題、特に、固有ベクトルの質問が多い。 世の中に、良い参考書の 少ない証拠であろう。 私は $_{
m J}$ 言語の御蔭で、若者のお役にたつ機会が多い。

しかし、」言語そのものの「普及は今一つ」の恨みがある。

この8月のJAPLAの恒例・夏季合宿の成果として、有能なる世話人・志村正人 氏の報告(文献1)を拝見した。大変、有益であった。 関係して、気が付いた事を述べて置きたい。

1. 固有ベクトルの計算は難物?

●質問(ikmh212211 さん・文献2):

2x2 行列 1 3 4 2 の固有値は 5 と _2 と判りました。 後者の 固有ベクトルは (_1, 1) とも判りましたが、固有値 5 の方は判りません。 答が (0 0) になって困ります。 お願いします!

○ 私の回答(文献 2): 東南アジアで好評なカナダ産の J 言語なら簡単!
 与行列 ikm2=.>1 3; 4 2 と定義する。 ルヴェリエ・ファデーエフ法
 の関数 LF0 で、 LF0 ikm2 から、解の固有値は 5, _2 です。
 固有ベクトルの計算は、関数(中野) eigvec を用い、

5 2 eigvec ikm2 から(44)、即ち4 で割って、(11) と

(34) である。

[検算] は、ip を「行列とベクトル」の内積関数として、例えば、

ikm2 ip (3 4) -> (15 20)、更に固有値 5 で割って、

(15 20) % 5 -> (3 4)、確かに 固有ベクトルに戻った。

●質問 (ppink0917 さん・文献 3):

3x3 対称行列 ppi3 即ち 2 0 1 0 2 1 1 1 3 の固有値 1 2 4 は判りました。 しかし、固有値 2 の時の固有ベクトルが判りません。途中で Z = 0 となってしまい、そのあとの続け方が判りません。 宜しく!

- ◎ 回答は、後に例題として述べる。
- ★ 志村報告(文献 1) 中にも、しきりと、同じ悩みを云々して居る。
 例えば、p.28、先頭行に「ランク落ちして逆行列もとれない厄介者が多い」と告白して居る。
 - 2. 志村報告流の処方

有効な LF 法による諸関数の Scripts を挙げて置く。

p.23: $tr=: (<0 \ 1) \&|: NB. diag$

志村 char_lf (TR_SUM 等が stack errorを起こすらしいので、 中野はその代わりに、自前の関数 nchar_lf を用いた。)

p.29: 志村 char lf evec sub

p.33: 志村 char_evec (stack errorを起こすらしい上記の関数 char lf を含む。)

私は多少のエラーらしきものを避けるため、下記の如き自前の関数を 使っている。しかし、大筋に於いて中野流は、志村流と大差は無い。

ip=: idot=: +/ . * NB. inner product

NB. Leverrier-Faddeev method by Nakano

LF=: >@{:@p.@charn (途中経過の印刷あり)

LF0=: >@{:@p.@charn0 (結果のみ印刷)

```
charn0 =: 3 : 0
In=. =@i.n=. # y
X=.In
i=.0
p=.1
for_k. >: i.n do.
X = . y + / . * X
trX = . +/(<0 1)|:X
pk=.-k%~trX
p=.p,pk
 X=.X+ pk * In
i = .i + 1
end.
 l.p
NB. by Nakano 2010 Sep.19
nchar lf =: 3 : 0
1; (LF0 y); charn0 y
nchar evec=: 3 : 0 NB. by Nakano
NB. char evec=: 3 : 0 NB. Shimura
NB. EIGEN=: \{0>; 1\{ char\_lf y NB. Shimura \}
EIGEN=: {@> ; 1{ nchar_lf y NB. by Nakano
EIGEN2=: {@> L:0 EIGEN ^/L:0 |.i.# EIGEN
ADJMAT=: char_lf_evec_sub y
ANS=.<''
for_LF. i. # y do.
TMP=. +/> (> LF{ EIGEN2) * L:0 ADJMAT
ANS=. ANS, <TMP
end.
 EIGEN,: }.ANS
)
例題: 上記の行列 ppi3 にて、固有値 4 2 1。
  ppi3
2 0 1
0 2 1
1 1 3
 nchar evec ppi3
4
      2
              1
                    1112 1 1 0 1 1 1 1
1 1 2 1 1 0 1 1 1 1
2 2 4 0 0 0 0 1 1 1
```

```
ppi3 ip (1 1 2)
4 4 8
4 4 8 % 4
1 1 2

ppi3 ip (_1 1 0)
_2 2 0
_2 2 0 %2
_1 1 0

ppi3 ip (1 1 _1)
1 1 _1
```

確かに、予想通りに戻った。

3. 随伴マトリックス の話題

志村報告(文献 1) の pp.28-29 に、丁寧な説明があるが、余因子行列の個数が多い時には、数値計算が大変である(志村例では 3 次)。

幸い、4 次の場合の丁寧な解説を掲載する本に遭遇した(文献 4 pp.100-103)。 このやり方を、J 言語でプログラム化して見た。 関数名 adjmat。

(wr は作業中の便宜で入れたものが多い。 適当に省略可。)

```
adjmat=: 3 : 0
k=.1
wr A=.y
wr uA=.un n=.#A
k=.1
wr ' k = ', ":k
c1=. +/ diag A
wr 'c1 = '
wr c1
wr A1=.(|: A)- c1*uA
 wr 'ans'
 Α1
 wr A=. y
 wr A1=. x
 k=.1
label k2.
k = .k + 1
wr ' k = ', ": k
ck=. (+/ +/"1 A*A1)%k
wr 'ck = ', ": ck
```

```
R=.0
 i=.0
 while. i < n = . # y do.
 wr 'i= ', ":i
 wr R
 j=.0
 while. j < (\# A) do.
 wr 'j =', ":j
 wr ja=.j{A
 R=.R , (+/(j\{A) * i\{A1)
  j=.j+1
 end.
 i=.i+1
  end.
wr Rn=. (n,n) $ }. R
wr 'ans = '
wr Rnn=. Rn-ck*uA=.un #A
label kn.
wr 'next ck '
wr dag=.(+/"1 A*Rnn)
wr ck=. (+/dag)%(k+1)
A1=.Rnn
wr ' k = ', ": (k+1)
if. k < (\#A) do. goto_k2. end.
 1 1
 )
```

未だ作業中のものであるが、名人達の改良を待ちたいので、敢えて 書いて置く。

古屋先生の原著の p.100 の例の 4 次行列 A を FA4 と命名してトライした初段の計算結果を、以下に示す。

```
adjmat FA4

1 3 _2 5
2 1 4 0
_1 _6 3 7
5 _4 _1 _3
1 0 0 0
0 1 0 0
0 0 1 0
0 0 0 1
k = 1
c1 =
2
_1 2 _1 5
```

この結果 ans を、左データとして(右データは不変)、同様の計算 adjmat

を続ければ、最後まで進行して、停止する。

その各段階の出力 ck から、固有多項式の各次の係数が判る。

```
]A10=.> _1 2 _1 5; 3 _1 _6 _4;_2 4 1 _1; 5 0 7 _5 _1 2 _1 5
5 0 7 5
A10 adjmat FA4
1 3 _2 5
2 1 4 0
_1 _6 3 7
\begin{bmatrix} 5 & -4 & -1 & -3 \\ -1 & 2 & -1 & 5 \end{bmatrix}
5 0 7 _5
k = 2
ck = 10
i= 0
0
j =0
1 3 2 5
j = \overline{1}
2 1 4 0
j =2
_1 _6 3 7
\frac{1}{3} = 3
5 - 4 - 1 - 3
i = 1
0 32 _4 21 _27
j =0
1 3 2 5
j = \overline{1}
2 1 4 0
j =2
\frac{1}{j} = \frac{6}{3} = 3
5 - 4 - 1 - 3
i = 2
1 3 _2 5
j = \overline{1}
2 1 4 0
```

```
j =2
_1 _6 3 7
j =3
5 _4 _1 _3
i= 3
1 3 _2 5
j = 1
2 1 4 0
j =2
\frac{1}{j} = \frac{6}{3} = 3
5 _4 _1 _3
32 _4 21 _27
 _8 _19 _43 _37
3 4 26 24
34 38 19 33
ans =
22 _4 21 _27
 _8 _29 _43 37
3 4 36 24
34 38 19 23
_167 _217 _303 _372
_353
k = 2
k = 3
ck = _353
i= 0
0
j =0
1 3 _2 5
j =1
2 1 4 0
j =2
\frac{1}{j} = \frac{6}{3} = 3
5 - 4 - 1 - 3
i = 1
0 167 124 124 186
j =0
1 3 2 5
j = \overline{1}
2 1 4 0
j =2
\frac{1}{j} = \frac{6}{3} = 3
5 - 4 - 1 - 3
i = 2
0 _167 124 _124 186 176 _217 312 8
   =0
j
1 3 _2 5
j =1
2 1 4 0
j =2
_1 _6 3 7
\bar{j} =3 5 - 4 - 1 - 3 i = 3
0 \ \_167 \ 124 \ \_124 \ 186 \ 176 \ \_217 \ 312 \ 8 \ \_33 \ \_134 \ \_303 \ 107
```

```
j =0
1 3 2 5
j = \overline{1}
2 1 4 0
j =2
_1 _6 3 7
<del>j</del> =3
176 _217 312
33 134 303 107
233 _106 _90 _372
ans =
next ck
1736 1736 1736 1736
1736
k = 4
```

結局、係数 c1=2、c2=10、 c3= -353、 c4=1736 で、 固有多項式

 $f(\lambda) = \lambda^4 - c1 \cdot \lambda^3 - c2 \cdot \lambda^2 - c3 \cdot \lambda - c4$ と成る。

文献 4 の古屋先生の原著 (p.102 第1行)と比較されたい。

(勿論、今では ルベリエ・ファデーエフ法の関数 LFO を用いて、一挙に

算出出来るものではあるが・・・・・)

4. 古屋行列 FA4 の 固有ベクトルの解

与行列 FA4

1 3 _2 5 2 1 4 0 _1 _6 3 7 5 4 1 3

演算 nchar evec FA4 (結果は膨大で、分割せねば印刷出来ない。)

固有値 4ケ(内訳は 実数2ケ、複素共役1組)

_8.04131 4.73208 2.65462j6.21085 2.65462j 6.21085

固有ベクトル (縦列に読め):

固有値1 (実数)に対し、

```
_575.544 434.319 _186.449 829.718
285.491 _215.438 92.4855 _411.57
_357.53 269.8 _115.823 515.424
726.429 548.179 235.329 1047.24
```

固有値2 (実数)に対し、

```
373.677 205.321 _63.5889 184.072
149.857 82.3406 _25.5013 73.8191
_47.0189 _25.8351 8.00125 _23.1614
170.197 93.5166 28.9626 83.8385
```

固有値3(複素数) に対し、

固有値4(複素数) に対し、

(いやはや、印刷を省略したいなー!)

「検算]

(1) 固有値1 (実数) lm1=. _8.04131 に対して: その 固有ベクトル の例として (第1縦列)v11 を選ぶ。

v11=._575.544 285.491 _357.53 726.429

計算

FA4 ip v11 4628.13 _2295.72 2875.01 _5841.44 4628.13 _2295.72 2875.01 _5841.44 % lm1 (固有値で割る) 575.544 285.491 357.53 726.429 ・・・・確かに v11 に戻った。

(4) 固有値 4 (複素数) 1m4=. 2.65462j 6.21085 に対して:

その 固有ベクトル の例として (第1縦列)√41 を選ぶ。

_12.5662j4.61412 50.3262j_41.1062 _36.7254j_97.453 _43.3128j2.81878 計算

FA4 ip v41

4.7008j90.2954 121.708j 421.69 702.757j 30.6045 97.472j276.492

固有値で割る

```
_4.7008j90.2954 _121.708j_421.69 _702.757j_30.6045 _97.472j276.492 % lm4
```

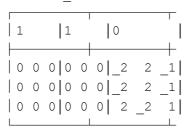
_12.5662j4.61412 50.3262j_41.1063 _36.7254j_97.4529 _43.3128j2.81878 確かに v41 に戻った。

と云う具合で、固有ベクトルの計算は成功している。

5. 固有値 に 重複がある場合

円滑な結果にならない。 下記の如く、重複しない固有値では良いが、重複ある 時は、ゼロ・ベクトルになる等々、その結果は何かおかしい。 演算の実例を示す。

nchar_evec nka3



0 0 0

0 0 0

$$\begin{smallmatrix} nka3 & ip & _1 & _1 & 1 \\ 0 & 0 & 0 \end{smallmatrix}$$

0 0 0

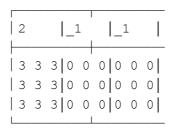
0 0 0

0 1 1

1 0 1

1 1 0

nchar_evec nkb3

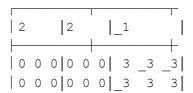


6 6 6

3 3 3 ・・固有ベクトルの戻った。

$$\begin{array}{ccccc} 1 & 1 & -1 \\ 1 & -1 & 1 \end{array}$$

nchar_evec nkd3



今後の研究課題であろう。

6. むすび

誰かが申された。 ちょっと以前までは、逆行列の計算だけに、数日を要し、 それだけで一仕事した気分になったとか?

今は、あっと云う間に、事は済んで仕舞う。 J 言語の威力も大いに預って居ると思われる。 我々は、幸せと感謝すべきだ。 その一里塚の勉強の例を報告した次第です。 その、きっかけを提供された志村さまに感謝したい!しかし、大きな難問題が残る。 それは、志村さまも言及すみだが「固有値に重複値のある場合」の処理である!

文 献

- 1) 志村正人:「マトリックスの数学と数値計算 (1) 行列式と行列の固有値」pp. 47、JAPLA 夏の合宿 2010/8/3
- 2) http://detail.chiebukuro.yahoo.co.jp/qa/question_detail.php?qid= 1144612321 時に2010/08/02、yahoo 知恵袋、数学カテゴリー。
- 3) http://detail.chiebukuro.yahoo.co.jp/qa/question_detail/q1346371703 yahoo 知恵袋、数学カテゴリー。
- 4) 古屋 茂著:「行列と行列式」 培風館、1959.5.20 増補版発行 p.100~