固有ヴェクトル　(Eigen_Vector)　計算の話題

Ｊ　と　固有値問題（その１０）

志村報告の理解を増す為に

中野嘉弘　（札幌市南区・８５才）

FAX 専　011-588-3354　　yoshihiro@river.ocn.ne.jp

固有値計算シリーズに続いて、固有ヴェクトルの計算法の話題を提供する。
志村氏の最近の報告「非対称行列の固有値を求める」の理解と推進を目的とする。

０．　は　し　が　き

JAPLA １月例会に志村報告「非対称行列の固有値を求める」を拝見した。
これは、昨年末（12.27）の志村メールがドキュメント化を約束していたものだ。
その中に、固有ベクトルを求める計算例があった。（文献１）
昨年の中野らロートルの固有値問題シリーズ９編中の第５報（文献２）は、副題として「ダニレフスキー法からファデーエヴァ法、そして固有ベクトルへ」の如く、同じ話題を取り上げた。　所謂付き「固有値ファン」の一員として、志村氏には大変お世話になって来たが、機は熟したと思うので、さらに実戦的、追加情報を提供したい。

１．　固有値問題分類学は要るか？

固有値問題は、古くて新しい大問題らしく、多くの解説があり、「神々しい術語」が氾濫し、その為か、分類学までが幅を利かす状態にある。　事柄は簡単な方が判り易い。
その観点からすれば

１）対称行列、非対称行列の区別を無くして、どちらにも有効な議論をしたい。
　　実数だ、複素数だと区別立てするのは、ナウイ話では無かろう。

２）しかし、大規模行列の場合には、特別の工夫が必要だろう。
　　その境界は、ありふれたパソコンで計算する限り、固有方程式の展開は今までは１３次付近であった。　志村氏が昨年夏ころ紹介下さったＬＦ法（２．節）は、遥かに高性能であるが、それでも３１次付近が限界であるらしい。
　　これらの事は、中野（文献２－ａ）が既に報じている。
　　それは、Ｊ言語の **root finder** 関数、かの有能なる **p.** 関数 がハング して仕舞う為である。　これらの改良を、昔、中野・山下・西川や森脇氏あたりが論じて居た。（例えば、文献２－ｂ）
　　それより、大規模行列の議論は、我々には無理か？あるいは必要なかろう？

３）　線形代数の教科書、参考書あたりの例題や演習問題が、高々、３次程度なのは全く物足りぬ！
　　せめて、**JAPLA**　会友、鈴木氏著「Ｊ言語による統計分析」（文献３）中の例の如く、５次行列位は取り上げたいものだ。　これ位が、キチンと処理出来るなら実用性はある。

　或る Ｗｅｂ情報で、対称大規模、疎行列のランチョス (Lanczos) 法の例題でさえ、なんと、５次行列に過ぎなかったのには、全く呆れて仕舞った。
実践者と解説者の「ずれ」であろうか？

４）固有ベクトル計算例の大半が、固有値が全て異なる場合なのも、残念だ。
固有値に等しいものがある例を、もっと議論すべきだ。　つまりジョルダン細胞の話題が欲しい。　これが無いと、行列の対角化の議論が深まら無い。

　等々が、考えられる。　分類学優先は、この辺に欲しいものだ。
以下、その観点から稿を進める。　そう云っても、説明は簡単な例からになるから最近の志村資料の理解から始めよう。

２．　ＬＦ法の理解

小生が　ＬＦ（**Leverrir-Faddeeva**）法を知ったのは、**JAPLA** の志村氏のＥメールによってであった（文献４）。　文字 Ｆ はロシア女性 ファデーヴア　**Vera Nikolaevna Faddeeva** の事で、彼女の原著の英訳　**Computational Methods of Linear Algebra** が **Dover** 版 になったのは、１９５９年であった。　これは行列の特性方程式や逆行列の構築法として、数学ソフト**Mathematica** のサブルーチン　にもなっている方法である。
その辺の話題も中野報告（文献２－ａ）にある。
同好のものは Ｆｒａｍｅ法として、昔から「知る人ぞ知る」ものであった。（文献５）
中野・山下・西川らが昨年来、苦心していた Ｎｅｉｇｅｎ法も、今では、余り器用とは云えないが、同好のものであった。　（文献５など多数）
　その後の Ｆｅｉｇｅｎ法は、そのＦｒａｍｅ法を用いたものである。（文献６）
　或意味で、一番すっきりしているのがＬＦ法であろうか？
以下、黙って志村報告と云えば、「文献１」のものを指すとする。

例１）志村報告(p.3)　**1.2.1** 数値例

A =. 3 3 $ 2 _1 1 _1 2 1 1 _1 2

　高々３次の簡単な「実対称行列」を例にするのは「鶏に牛刀」のそしりを受けそうだが、最初は止むを得ない。

　ＬＦ法の演算（エルエフ）　**lf A**　から　固有値　**λ**は　**3 2 1**　。
（関数 **lf** は、文献４の志村メール内の名前、文献１の志村資料内では　**char_lf** 、この手の解法は、暗算・筆算を含め、他にも多々ある。　小規模行列ならば、どれでも可であろう。　逆行列の算出については、Ｆｒａｍｅ法が判り易い。）

　固有ベクトルの算出：　次節の Ｃｒａｍｅｒ法 の方が、はるかに簡便であるが、志村報告の理解の為に、ここで解説しよう。　しかし、そこでの志村例は、不幸にも？行列 **A** でない別行列（町田・駒崎・松浦氏 **1990**のもの）であったので、今は 元通りの **A** のままで計算を続行しよう。

先ず、随伴補助行列 **adj**（Ｂ）　は

char_vec A　から　（志村資料（文献１）では　関数名　**dhar_lf_vec**　だが）

```
|1 0 0|_4 _1 1| 5 1 _3|
|0 1 0|_1 _4 1| 3 3 _3|
|0 0 1| 1 1 _4|_1 1 3|
```

または

```
          | 1 0 0 |    | _4 _1  1 |   | 5 1 _3 |    ベクトル成分  x 値 用
λ^2 * | 0 1 0 | + λ* | _1 _4  1 | + | 3 3 _3 |   ベクトル成分  y 値 用
          | 0 0 1 |    |  1  1 _4 |   | _1 1  3 |   ベクトル成分  z 値 用
```

これから固有ベクトルを計算する方法は、各 λ値 について、上式の行列成分を計算すればよい。

λ＝1 の時：

第１行は　1*1 + 1*_4 + 5 = 2, 1*0 + 1*_1 + 1 = 0, 1*0 + 1*1 + _3 = _2

第２行は　1*0 + 1*_1 + 3 = 2, 1*1 + 1*_4 + 3 = 0, 1*0 + 1*1 + _3 = _2

第３行は　1*0 + 1*1 + _1 = 0, 1*0 + 1*1 + 1 = 2, 1*1 + 1*_4 + 3 = 0

行列表示では

```
| 2  0  _2 |  ・・・  x  成分
| 2  0  _2 |  ・・・  y  成分
| 0  2   0 |  ・・・  z  成分
```

```
同様にして、固有値 λ＝2 では  | 1 _1 _1 |、  λ＝3 では | 2 _2  0 |
                              | 1 _1 _1 |            | 0  0  0 |
                              | 1 _1 _1 |            | 2 _2  0 |。
```

これらを纏めれば、下記の　char_evec A からの表と一致する。

（志村資料（文献１）では　関数名　dhar_lf_evec　A となっているが）

```
| λ＝3   | λ＝2    | λ＝1    |
|  2 _2 0 | 1 _1 _1 | 2 0 _2 |  ・・・  ベクトル成分  x 値
|  0  0 0 | 1 _1 _1 | 2 0 _2 |  ・・・  ベクトル成分  y 値
|  2 _2 0 | 1 _1 _1 | 0 0  0 |  ・・・  ベクトル成分  z 値
```

これらが固有ベクトルの計算法である。

３．　固有ベクトル計算表の意味付け

上の結果表の意味は、下記の如く考えるのが良かろう。　（直解は困難かも？）

１）　固有値 λ= 3 の時、特性方程式は　A3 =.

```
(2 - 3)   _1      1
  _1    (2 - 3)   1
   1      _1    (2 - 3)
```

成分で書けば　-x -y + z = 0　(1)

　　-x -y + z = 0　(2)　式 (1) と (2) は同じ、　一方のみで可。

　　x -y - z = 0　(3)

先ず　(1) + (3) より　y = 0　従って　x - z = 0　より、　x = z

即ち　x = 0　なら　z = 0　で　x = y = z = 0 、

　　x = 2　なら　z = 2　で、且つ　y = 0、

　　x = - 2　なら　z = - 2　で、且つ　y = 0。

上記の結果の表の左端３列（縦に見よ）は、その事を表している。

２）　固有値 λ= 2 の時、特性方程式は　A2 =.

```
(2 - 2)   _1      1
  _1    (2 - 2)   1
   1      _1    (2 - 2)
```

成分で書けば　-y + z = 0　(1)　即ち　y = z

-x　 + z = 0 (2) 即ち x = z
x -y 　= 0 (3) 即ち x = y = z

３者等しいので、固有ベクトル成分は
X = y = z = 1 1 1 , _1 _1 _1。

上記の結果の表の中央の３列（縦に見よ）は、その事を表している。

しかし、解 (0 0 0) が省かれ、解 (_1 _1 _1) が重複するのは、理解出来かねる。
引用された John Radall 准教授の当該 Script に改良の余地があるかも？

３） 固有値 λ= 1 の時、特性方程式は A1 =. (2 - 1) _1 1
_1 (2 - 1) 1
1 _1 (2 - 1)

成分で書けば x -y + z = 0 (1)、 (1) + (2) より z = 0
-x +y + z = 0 (2)、 (1) - (2) より x = -y
x -y + z = 0 (3) これは (1) と同じ故、用いない。
かくて z = 0, x = 2, y= 2 は 解。 x = -2, y= -2 でも良い。
x = 0, y = 0 も可故、 X = y = z = 0 0 0 は trivial な解だが入っている。

上記の結果の表の右単端の３列（縦に見よ）は、その事を表している。

志村報告の関数 char_lf_evec では、対応する固有値も表示に加えてある。
以上が、ＬＦ法の結果の説明である。

なお、志村報告 p.6 最下段行や p. 7 中央付近に char_lf_evec_sub とあるのは
上記の説明を計算で表示する為の、中間データである。
元来のＬＦ法の記事（文献４）では、その意味の説明も無く利用されて居た関数
char_vec の事である。
今回志村氏は、その説明を、「町田・駒崎・松浦」氏の著書により、追加されたのだ。
上記の中野の初等的説明で理解出来るなら、この関数の利用に拘わる必要は無い。
それに「拘らない」別なより簡単な方法を 次節４. で述べる。

４. Ｃｒａｍｅｒ法の利用

与行列の固有値とは、同次連立方程式が non-trivial な解を持つ為の条件でもある。 それは、特性方程式を解けば、すぐ判る。 その方法は今や簡単である。
そして、その時の non-trivial (non-zero) な解が固有ベクトルである。
その連立方程式の解法に、周知の Cramer 法を、先ずトライする。
その Script " crame0 " は、稿末に付録として添付してある。

例： 与行列 A =. 3 3 $ 2 _1 1 _1 2 1 1 _1 2
固有値は lf A より 3 2 1
固有値 3 に対して A3 =, A - 3 * I3 （３次単位行列）
2 A2 =. A - 2 * I3
1 A1 =. A - 1 * I3

cram0 A3 よりの 解 (x y z) = 1 0 1
cram0 A2 よりの 解 (x y z) = 1 1 1
cram0 A1 よりは 解無し det = 0, please another way !

「解無しの場合」　解くべき連立方程式は　x - y + z = 0　故
この解は　1)　(x y z) = ０ ０ ０
　　　　　2)　(x y z) = １ １ ０　または　２ ２ ０
　　　　　3)　(x y z) = _1 _1 ０　または　_2 _2 ０
これらは、前節２．の結果と同じ事を云うのだが、理解は、はるかに容易であろう。

５．　固有値・固有ベクトルの同時的求め方

志村報告（文献１）　pp.5-7 の記事への異見である。
p.5 の　１．２．２節「固有ベクトルを求める」の冒頭に、Randall 教授のスクリプトは固有値のみで（固有ベクトルには及ばずと）あった。
しかし、中野は、志村メールで教えられた直後から、すでに char_vec, char_evec など、関係する固有ベクトル計算プログラムを利用して来ている。　それらは Randall 教授のスクリプトだったと思うが？　そうでないとすると、何か記憶違いかな？
また、志村報告 p.5 では、それ故、与行列の特性行列の随伴行列から固有ベクトルを求める町田・駒崎・松浦の方法（文献４－ｃ）を紹介し、それに依って必要スクリプトを解説するとあった。
この方法では、３次行列の場合でも、９ケの随伴行列を作る手間が要る。
しかし、実際には、わざわざ作らずとも、Randall の固有値プログラム内で、すでに（自動的に？）作られているので、その必要は無い（この事は、志村氏も、当該報告の p.6　下段にて、実は述べている）。　それ故必要なのは、原理の解説のみであろう。

そこで中野は、志村報告（文献１）　２．節　Ｆｒａｍｅ法の紹介スクリプト内で、同じ事を実現してお目に掛け、解説の役目を果たす事にしよう。

```
Feigen =: 3 : 0   NB. J6.01,  by  Yoshihiro NAKANO '07. Oct. 12
 wr  y
 In =. unitm n =. # A =. y
    i =. 1
 Bi =. A
    pi =. trace  Bi
 Ci =. Bi - pi * In
 nans =. 1, (-pi)
 vc =. (<In), < Ci
     i =. 2
    while. i < n do.
 Bi =. A indot Ci
    pi=. (trace Bi) % i
 Ci =. Bi - pi*In
 nans =. nans, -pi
 vc =. vc, < Ci
    i =. i+1
    end.
        wr   ,>}. p. |. nans
          vc
)
```

６．　対角化問題

これは難問の始まりである。　今回の志村報告（文献１）p.8 に　１．３　対角化

なる解説がある。 例題の行列　a0 は p.6 のもので
a0 =. 3 3 $ 2 _2 3 1 1 1 1 3 _1、固有値は　**lf a0** から　**3 _2 1** である。
対角化の為の変換行列 **P** を先ず作る。　　**P = | 1 11 _1 |**
| 1 1 1 |
| 1 _14 1 |　を作る。

これは **p.7** 末尾の各固有値に対応する固有ベクトル表（今は各３列づつある）から、
適当に一本づつ採ったものである（ただし、ベクトル成分の比率のみが問題である）。
演算の実際は　**p =. 1 11 _1 ,. 1 1 1 ,. 1 _14 1** とでもする。
また、 **p** の逆行列　を　**pm =. %. p** とする。　行列のかけ算を **mp =: +/ . *** と
定義すると、対角化の演算は　**pm mp a0 mp p**　である。
その演算の首尾良き例が志村報告に掲載されている。

ここで、問題が若干ある。

１）志村例の如く、目視・手計算では大変なので、プログラム化したい。
その為の、中野の　**diaglize** 関数の **Script** を下掲しておく。

```
   diaglize =: 3 : 0
 n=. # Y =. y
 Z =. char_evec  Y
      i =. 0
   while. i < n  do.
 pi =. i { Z
   m =. 0
 label_m.  pi0 =. m {"1 > pi
   if. pi0 = n # 0  do. m =. m + 1
                          goto_m.  end.
   if. i = 0  do. P =. pi0
                else.  P =. P, pi0  end.
      i =. i + 1
   end.
     P =. |: (n,n) $ P
   Pm =. %. P
  dm =. Pm  mp  Y  mp  P
  )
```

この関数で、大抵の場合は（本来、対角化が可能な場合では）、旨く行く。

２）固有ベクトル表から、各一本づつ「適当に！」採る事は、いつでも可能か？
実は、固有ベクトルがゼロ（ヌル）ベクトルは採れないので、この関数には、その考慮がされている。
３）いろいろ工夫しても、対角化出来ない例が、多々ある。　大抵は、固有値に等根がある場合である。　東海大学出版の参考書（文献７）から例示しよう。
ｉ）対角化不能の行列の例：
a) 文献７、**p.165**、例**9.6**
Atoka =. 3 3 $ 10 22 15 _1 _3 _3 _4 _8 _4
演算　**diaglize Atoka**　の結果は　**domain error　Pm =. %. P**　（逆行列不可）
今は、　固有値は　　**lf Afno**　から　**2 2 _1**　（等根あり）

「対策」志村報告（文献１）　**p.9-10**、　３　ＬＡＰＡＣＫ の利用
Ata1 =. dgeev_jlapack_　Atoka

(%. >{: Ata1) mp Atoka mp >{: Ata1 より、結果はおよそ、
_1 0 0
0.0937 2.1444 _0.1444
0.0937 0.1444 1.8556
演算は遂行されたが、結果は、不審？

b) 文献７、p.165、例9.7
Atokb =. 3 3 $ 7 18 12 _3 _8 _6 0 0 1
演算 diaglize Atob の結果 index error pi0 =. m {"1 > pi （探索不能）
今は、 固有値は lf Afno から _2 1 1 （等根あり）

詳しくは、ＬＦ法の演算 char_evec Atokb から
lambda _2 1 1
eigen | _18 _54 _36 | 0 0 0 | 0 0 0 |
vec- | 9 27 18 | 0 0 0 | 0 0 0 |
tors | 0 0 0 | 0 0 0 | 0 0 0 |
固有値 1 に対応する固有ベクトルは、共に ヌル・ベクトルである。

「対策」志村報告（文献１） p.9-10、 ３ ＬＡＰＡＣＫ の利用
Atb1 =. dgeev_jlapack_ Atokb
(%. >{: Atb1) mp Atokb mp >{: Atb1 より、結果はおよそ、
1 0 0
0 _2 0
0 0 1
今は、演算は遂行され、結果は、首尾良く対角化出来た。

どうも、これで見ると、最近話題の ＬＦ法 は、固有値計算では、判り易く、有効であるが、固有ベクトル計算では、さらに今まで、充分磨かれて来た従来の方法、例えば ＬＡＰＡＣＫ法 の助けが要るらしい。

７． Ｊｏｒｄａｎ 分解

行列の対角化問題は、固有値に等しいもの（等根、重根）がある場合には、俄然、難儀となる。 そして、**Jordan Block** が登場するのが普通のシナリオである。
その代わりか（？）、志村報告（文献１）の４．節にスペクトル分解の記事がある。
そして、固有値が「重根の場合」には、部分分数展開が必要との記述を見た。
これは、有力な知識になれるかも？！
ところで、スペクトルとは何か？ よく判らぬので、数学ソフト **Mathematica** の研究陣の書いた数学百科事典（文献８）を調べた。 行列のスペクトルとは、行列 Ａ の固有値（全体）の事を指し、λ（Ａ）と表示する。 そして、全固有値の積が 行列式値に当たると。
これだけでは、志村報告を理解出来なかった。 ただ、 これは、大切な事だ。
しかし、その計算例には、重根の場合は示されては無かった。
そこで、志村氏御推奨の参考書、金谷健一氏「これなら分かる応用数学教室」（文献９）を、私も購入して眺めた。
p.178 に曰く「対称行列のスペクトル分解（または固有値分解）」とある。
p.165 に曰く「対称行列でない時は、固有値が複素数になったり、さらに、重解の時にはジョルダン標準形と云う非常に（！）、ややこしい話が生じます。
しかし、実際の必要問題では、ほとんどが対称行列の場合です。」

「我々の欲しいのは、未知数 λ を含む固有方程式の計算ではなくて、固有値と固有ベクトルだから、直接、それを計算すればよいのです。」
こりゃ、駄目だ！　「直接」の目的が違う！
我等の「直接法」とは、固有方程式を「直接」解こうと云うのであるが、金谷先生のは固有ベクトルを反復解法で、近似的にでも、「直接」求めようと云うものらしい！
さらに確認すれば、 **p.229** の　デイスカッション　の項で、「混同してはいけません」、スペクトル分解は対称行列に対する表現で、ＳＶＤ（特異値分解）は「任意の長方行列」に対する表現です！！！　とある。

そこで、普通のシナリオに戻って、 **Jordan** 分解を取り上げよう。
また、志村報告の末尾（文献１、 **p.16**）に、７．ＳＶＤ　特異値分解の記事があり、学校では **Jordan** を教えるが、実務では圧倒的に **SVD**法を用いるともあった。
その理由は、**Jordan** は小型正方行列向きで、**SVD** は大型または非正方行列に適用出来るのだそうだ！　（脚注に、サイズ **512^4** でも実用計算可能とか！）
やはり、先ず、教育向きの **JORDAN** を先ずトライしよう。　なお、例の **LAPACK** 中には、 **SVD** の名前はあったが、**JORDAN** の名前は、ロートルの目には！　見えなかった。

JORDAN 分解の教科書的解説は成書に任せて、計算の実例のみを示そう。
例は、一松信氏著「代数学入門第二課（固有値問題など）」**p.187** 単因子論の直前にある６ **x** ６次行列である。 （文献１０）
普通の教科書では、志村氏の申される如く、高々、４次までしか扱っていないので、６次とは、珍しい例である。　勿論、非対称である。　さて、どうすべきか？

与行列

```
        |5 _1  1  1  0  0|
        |1  3 _1 _1  0  0|
    A = |0  0  4  0  1  1|
        |0  0  0  4 _1 _1|
        |0  0  0  0  3  1|
        |0  0  0  0  1  3|
```

固有値は、関数　**lf A**　より　**2 4 4 4 4 4**　で　５重根が含まれる。
固有ベクトルは
１）　固有値　λ＝　**2** に対して、固有ベクトルを求める。
　　固有方程式　**A2I =. A - 2 * I6**　**NB. I6**　＝　６次単位行列　として
　　　　A2I = 0　を解く。
　クラメール関数　**cram0 A2I** の結果、解は　**v0 =　0 0 0 0 _1 1**
２）　固有値　λ＝　**4** に対して、固有ベクトルを求める。
　　固有方程式　**A4I =. A - 4 * I6**　**NB. I6**　＝　６次単位行列　として
　　　　A4I = 0　を解く。
　重根の為、クラメール法　**cram0 A4I** からの固有ベクトルは、同一であり、この先、旨く進展しない。　別法を採る。

```
          |1 _1  1  1  0  0|
          |1 _1 _1 _1  0  0|
    A4I = |0  0  0  0  1  1|
          |0  0  0  0 _1 _1|
          |0  0  0  0 _1  1|
          |0  0  0  0  1 _1|
```

６成分（ **x y z u v w** ）の　ベクトル方程式　の解を探す。

A4I * （ x y z u v w） = （0 0 0 0 0 0）

解 **v1 = （ 1 1 0 0 0 0）**
v2 = （ 0 0 1 _1 0 0） が、直ぐ見付けられる。

次に
A4I * （ x y z u v w） = v1 を解く。
解は **v3 =. （ 0.5 _0.5 0 0 0 0）**

さらに
A4I * （ x y z u v w） = v2 を解く。
解は **v4 =. （ 0 0 0 0 0.5 0.5）**
最後に
A4I * （ x y z u v w） = v3 を解く。
解は **v5 =. （ 0 0 0.5 0.5 0 0）** を得る。
かくて変換行列は
P = （ v0, v1, v2, v3, v4, v5 ） になろうか？ 具体的には
ベクトル **v0 v1 v2 v3 v4 v5** を並列して

```
        | 0  1  0   0.5  0    0 |
        | 0  1  0  _0.5  0    0 |
    P = | 0  0  1   0    0    0.5|
        | 0  0 _1   0    0    0.5|
        |_1  0  0   0    0.5  0 |
        | 1  0  0   0    0.5  0 |
```

P の逆行列を **Pm =. %. P**、 **mp** を行列の積演算として、**JORDAN** 分解のつもりは

```
                      |2 0 0 0 0 0|
                      |0 4 0 1 0 0|
Pm  mp A  mp  P  =    |0 0 4 0 1 0|
                      |0 0 0 4 0 2|
                      |0 0 0 0 4 0|
                      |0 0 0 0 0 4|
```

固有値が対角線上に並ぶのは予定通りであるが、 **1** が **4** の直上または直右より離れているのは、気に入らない。

そこで変換行列 **P** を若干、手直しする。 それを、固有ベクトルの並列で示すと
P1 =. （v0 v1 v3 v5 v2 v4） とする。 その時は

```
                        |2 0 0 0 0 0|
                        |0 4 1 0 0 0|
P1m  mp A  mp  P1  =    |0 0 4 2 0 0|
                        |0 0 0 4 0 0|
                        |0 0 0 0 4 1|
                        |0 0 0 0 0 4|
```

の如く、 数 **1** の位置は直近となる。
しかし、結果行列の第３行目、中央の数 **2** が気に入らぬ。 本来は **1** の筈である。

そこで、変換行列の若干部分で、数値の大きさを半分にする。 それを **P2** とする。

ベクトル **v0 v1 v3 v5 v2 v4** を並列して

```
                | 0  1  0.5  0    0    0  |
                | 0  1 _0.5  0    0    0  |
        P2 =  | 0  0  0    0.25 1    0  |
                | 0  0  0    0.25 _1   0  |
                | _1  0  0    0    0   0.5 |
                | 1  0  0    0    0   0.5 |
```

結果は

```
                                  | 2  0  0  0  0  0 |
                                  | 0  4  1  0  0  0 |
  P2m   mp A  mp  P2  =  | 0  0   4  1  0  0 |
                                  | 0  0  0  4  0  0 |
                                  | 0  0  0  0  4  1 |
                                  | 0  0  0  0  0  4 |
```

これで、どうやら期待の **JORDAN** 分解が得られたらしい。
この **Jordan** 分解結果は、数学ソフト **Maple V (R3)** での結果とも一致している。
しかし、上記のような方法は廻りくどく、複雑さを伴う。
実際、参考にした（文献１０、一松先生）にも、具体的には便利で無いと書いている。
数式ソフト **Maple V** でも、結果が判っているので、形式上、それになるような作業手順しか説明してはいないように感ずる。　そして、検算法が例示されている。
Mathematica の方ではどうかな？　実は、小生の **Mathematica V.2.0** は少々旧版なのか旨く処理出来なかった。　少しは新しい **Ver. 2.2** に対応をうたう解説ハンドブック（文献１１）の説明記事でも、似たり寄ったりと見えた。　難しいのだな！

む　す　び

新しい　ＬＦ法　といえども、固有ベクトル計算では、今一つな事が判った。
その先、スペクトル分解や特異値分解について、詳しく論ずるのは次回以降とする。
特異値分解は統計（多変量データ）の「分散共分散分析」でよく利用され、そのソフトを販売する会社もある。　我がＪの会友・鳥邊さんあたりも、お詳しいのではないか？
また、「数学工房」にも「特異値分解」コースの予定がある（**4**月**26, 27**日　文献**12**）。
しかし、志村報告は、固有値問題のロートルファンにとって、面白くかつ大変有意義であった。　謝意を表し、会友からの、今後の議論の発展を期待する。

文　　献

１）志村正人：「非対称行列の固有値を求める」JAPLA報告　**2008.Jan.24, pp. 18**
２）**Faddeeva, V.N.: " Computational Methods of Linear Algebra" New York, Dover, 1958,** その **p.50** には　ジョルダン分解の話題もある。
　－ａ）中野嘉弘・山下紀幸：「Ｊと固有値（その５）ファデーエヴァ法」JAPLA報告 **2007.Sep.22, pp.8**
　－ｂ）中野・山下・西川：**" Technical Correspondence" VECTOR 2005 Spring Vol.21 No.2 pp.116-118**
３）鈴木義一郎：「Ｊ言語による統計分析」森北出版、**1996.10 p.132**
４）志村Ｅメール： **<JAPLA@aplsoft.co.jp>, <jcd02773@nifty.ne.jp>**

－ａ）「固有値ファンへ　John Randall の Scripts」'07 Aug 27 11:37
「韓国 Korean J Group」'07 Aug 27 16:29
「固有値の話題、世界を飛ぶ　Leverrier-Faddeev Algorithm」
'07 Aug 28 9:59　<JAPLA@aplsoft.co.jp>,
<jcd02773@nifty.ne.jp>　etc.
－ｂ）「(SHIMURA)固有値ファンへ　Scripts by John Ranall 」2007.12.12. 12:56
char, char_vec, char_evec (using adj matrix)
「(SHIMURA)中野固有vec再」　char_evec , Gerschgorin theorem
2007.12.17 9:51
－ｃ）町田・駒崎・松浦：「マトリックスの固有値と対角化」東海大学出版部 1990
中野は残念ながら、この文献を実際には見てはいない（孫引きのみ）。
－ｄ）「Re:中野（SHIM）またまた固有値で」2007.12.27. 22:35
char_lf, char_lf_evec, char_lf_evec_sub, char_invmat,
char_vec, char_evec by M.Shimura
５）中野嘉弘：
－ａ）「Ｊ言語と高等数学　固有値問題（直接法）」JAPLA 2007 Apr 28 pp. 9
６次まで　Eigen.doc
－ｂ）「　Ｊ言語　と　固有値問題（その２）　」JAPLA '07 May 26　pp. 13
直接 Neigen法　９次まで　Neigen2.doc
－ｃ）「 Ｊ言語 と 固有値問題（その３）　」JAPLA '07 Jun 23　pp. 3
VECTOR誌投稿原稿　Eigen3.doc
－ｄ）「 Ｊ言語 と 固有値問題（その４）　」JAPLA '07 Jun 23　pp. 9
再帰 Naigen法　１０次(10 min)、LAPACK 一般１６次まで　Eigen4.doc
－ｄ）中野・山下紀幸：「 Ｊ言語 と 固有値問題（その５）」JAPLA '07 Sep 22
Naigen法　１３次(24 min)、ＬＦ法　３０次（0.01 sec）、　Ｃｒａｍｅｒ法、
固有ベクトル（鈴木５次）、 固有値・ベクトル同時解法　Feigen4.doc pp.8
－ｅ）「 Ｊ言語 と 固有値問題（その６）」　JAPLA '07 Oct 27 pp. 7
直接法　Ｆｒａｍｅ法、 James Sutherland Frame (1949) Feigen2.doc
逆 Frame法、Naigen法改良（１０次 0.8sec、１３次、24 min まで）
－ｆ）「行列式の計算法異考、　固有値問題（その７）」JAPLA '07 Nov 24　pp.10
Recursive（反復法）と Elimination（消去法）、　APL と J
－ｇ）中野「ＡＰＬからＪへ　再帰法・行列式演算　固有値シリーズ（その８）」
Dyalog APL101　から　J601　へ　Proc JAPLA 2007 Dec 8
DET8.doc　pp. 7
－ｈ）中野嘉弘「ＡＰＬと固有値問題　シリーズ（その９）」Proc JAPLA 2007 Dec 8
Root Finder、　Dyalog APL101 版で　APL9.doc pp. 6
－ｉ）山下紀幸：「行列の減次法アラカルト」 Proc JAPLA 2007 Dec 8　p.4
－ｊ）山下紀幸：「もう一つの行列の減次法（Sylvesterの定理）」2008 Jan 24　p.1
注：インターネット・志村ページの表（Proc JAPLA 2007）はやや、不正確か？

６）古屋茂：「行列と行列式」培風館　1957.2.10 初版、1998.3.30 増補第65刷、p.100
－ａ）中野嘉弘・山下紀幸：「Ｊと固有値（その６）直接法　Frame法を主に」
JAPLA '07 Oct 27 pp. 7

７）東海大学基礎数学研究会（板井・志摩・土屋・原・前田）編「新版基礎線形代数」
東海大学出版部、 2007.3.20、　pp.267

８）E.W. Weisstein:" CRC Concise Encyclopedia of MATHEMATICS 2nd ed." 2003,
Chapman & Hall/CRC pp.3242 (p.2766R)
９）金谷健一：「これなら分かる応用数学教室」共立出版、2003.6.15 初版１刷
pp.271、　2005.12.5 初版８刷

１０）一松　信：「代数学入門第二課」近代数学社　1992 pp.260
１１）M.L.Abell & J.P.Braselton　共著、川瀬・五島・佐藤・田沢　共訳
「Ｍａｔｈｅｍａｔｉｃａ　ハンドブック　Mathematica バージョン 2.2 対応版」
pp.797、東京電機大学出版局 1994.12.10 第１版１刷、 p.245 JordanForm　参照
１２）http://www.sugakukobo.com/

※　付録　「 J Scripts 類 」

```
        cram0=: 3 : 0
   wr Y0=.y
  n1=.<:  n=. # y
   Y =. }: y
    jy=.0
   goto_1.
 label_chg.
   if. jy >0   do.  ' d0 (det) = 0,then  please other way !'
                                              goto_ed. end.
               Y=.Y0
           wr    Y =.  Y E1r (1, 2)
               Y =. }: Y    NB. old y -> new  Y  '08.2.29
               jy=.jy+1
 label_1.   wr ' jy, Y = '
    wr   jy
    wr    Y

    Ym=. Y E2c (n1, _1)
  wr   ' d0 =  '
  wr   d0=. det }: "1 Ym
   if. d0 = 0 do. NB. wr  }: "1 Ym
                      wr ' Ym = '
                      wr  Ymp=. +/ Ym
                  wr   ' x + y + ...  *  (". Ymp) =  0 '
                      goto_chg. end.
     i=.0
    dx0=. Ym E1c (i, n1)
    dx=. det }: "1 dx0
   X=.  x=. dx % d0
    i=.i+1
    while.  i < n1 do.
     di0=. Ym E1c (i, n1)
     di=. det }: "1 di0
     xi=. di % d0
     X=.X, xi
     i=.i+1
```

```
 end.
   X=. X,1
label_ed.
 )
```

未完、作業中（未熟品）！