Ｊ言語　と　固有値問題　（その６）

直接法の　**Frame**法　を　主に

中野嘉弘　（８４才・札幌市）　山下紀幸　（８２才・横浜市）

FAX 専　011-588-3354　**TEL/ FAX　045-851-3721**
yoshihiro@river.ocn.ne.jp

今まで見逃されて来た直接法の最古参、 **Frame** 法を主に、前回までの 我ら日本勢の **Naigen**法、仏露の大家 **Leverrier-Faddeev** 法の追加記事を述べる。　予定して居た固有ベクトル計算法等の更なる改良の件は次回以降に廻す。

０．　は　し　が　き

土佐礼子選手が熱闘した大阪の世界陸上、また谷亮子選手が金メダルで、頑張ってくれたブラジルの国際柔道大会の頃、固有値問題の（その５）「ル・ヴェリエ　＆　ファデーエフ法」を書いた（文献１）。　これは、固有値問題の直接法解の白眉である。　その送稿後に、それと全く同様のアルゴリズムが、かなり昔から、既に報告されている事を発見した。　「**Frame** 法」である。
古屋　茂先生著「行列と行列式」（文献２）の　**pp.100-102** に解説があり、一般に **n** 次行列 **A** の余因子行列　**adj A**、行列式、逆行列、固有多項式を、いちどきに計算する方法である。　以前から、多少知ってはいたが、関心が薄かったのは、口惜しい。
他の成書にも、殆ど見えぬ。　岩波数学辞典では、新しい第４版・第３版に記述は無くなんと、却って、４０年以上も昔の第２版に説明があった。 **James Sutherland Frame （1907-97**、米国）が　**1949**年に発表した方法であると（文献３）。
インターネットで検索しても、詳しい事は判らない。　中国語の「法拉米（**frame)**」が延々と多数、並ぶだけである。　中国恐るべし！

この　**J.S. Frame**　は、**1945**年には代数方程式を解く機械を発明して居る（文献４）。
アメリカ数学会の重鎮で、女流数学者の応援で知られる（文献５）。
また、数学ジャーナル「パイ・ミュー・イプシロン」社の総裁として、数学の普及にも貢献した（文献６）。
世界の大数学者１０１人（ユークリッド、フェルマー、オイラー、ガウス、アーベル、ガロア、等々・・・、日本人では高木貞治、谷山豊の名前もある）リストの中で、ＡＢＣ順でフーリエの次に並ぶ大学者でもある（文献７）。
固有値問題の「直接法」では、例の仏露連合軍、ルベリエ・ファデーエフ法の証明を与えたのは、ロシア女性のファデーエバ **Vera Nikolaevna Faddeeva**であるとされている（文献８）。　その為か？ファデーエフの名前が残って、アメリカ人フレイムの方が無視されたのは、合点が行かぬ。　固有値問題で有名なヤコービの名は、先の１０１人中に残っているのだから、なおさらである。
このフレイムの伝記は、さらに、調べると面白いようだ。
大半の話は、（平成１９年）９月１８日までに作文して居たが、その後、山下 から面白い情報があった。
それによると、ファジェーエフ・ルベリ法は、すでに大昔、１９８５年には、カシオのポケコン数値計算ライブラリーに紹介されており、山下もＡＰＬ言語で、固有値問題の

トライを経験して居たと云う（文献９、９－ａ）。
どうも、古来の「フレイム」法と近頃話題の「ファデーエフ（バ？）」法とは同じものだと云える。
　さらに、我らの **iterative** な直接法、**Naigen** 法関係へのコメントもあるので、追加する。

１．　Ｊｏｈｎ　Ｒａｄａｌｌ　先生のこと

　今回の固有値問題・直接法・ファデーエヴ（もし女性の方ならば？　ヴァ）のアルゴリズム騒ぎの張本人 **Randall** と同名が、**9**月 **23**日の日本物理学会年会（札幌）で話題として登場した（文献１０）。　それは、ハーヴァード大の女流教授 **Lisa Randall** で、彼女は、「５次元空間重力理論」で、最近、被引用回数１万回以上の記録を作った。
その著書： **Unraveling the Mysteries of the Uni-verse's Hidden Dimensions** は
ベストセラーとして有名である。
面白い偶然と思い感激したが、固有値問題のそれは **John Randall** で男性である。
　それがきっかけで、**9**月 **26**日の志村メールは、米国 **NJ** 州の数学の男の先生を報じて来た（写真は見ない方がとも・・）。　それにつられて、中野は、写真を拝見、素性を知った。　ニューヨークの隣、**NJ** 州のニューアーク市の **Rutgers Uni.** の数学と情報科学の准教授となっているから、専門家ですね。　実は、別人ですが、英国には同名のジョン・ランドール卿（**1905-1984**）が居て、これは物理屋さん。　先の第２次世界大戦当時、電波兵器のマグネトロンの陽極の冷却装置の発明で、可能出力を６００倍に高め連合軍の勝利に貢献した。　平和利用では「電子レンジ」の開祖に当たる人です。
　これが　**John Randall**、 世の中とにかく、面白いものです！

２．　Ｆｒａｍｅ法　の　あらまし

　本法の特長は、固有値問題の厳密解を目指すもので、行列の特性（**characteristic**）方程式を扱う。　そのアルゴリズムが、高知大学の、数値解析の講義のシラバスの中に紹介されて居るのを発見した（文献９－ｂ）。
　Ｃ言語流に書かれているが、大変、簡単なものだ。

```
    データの原行列を A とし、
      X := E;    単位行列
  for  k := 1 to n  do begin
     X := AX;
    c(k) := - ( X の対角成分和）／ k;
    X := X + c(k)E;
  end;
```

　　c(k) は、特性方程式の各次数の項の係数である。
　　c(1) は原行列の対角和、**c**（最高次）は原行列の **det**（行列式）の値である。
　我らの　Ｊ　言語でも、殆ど、上とそっくりにプログラム出来る。

　　ただし、カシオ・ポケコン数値計算ライブラリー（文献９－ａ）の該当説明には、重大な記述ミスがある。　上式の　（対角成分和）／**k** の、 **k** での割り算が脱落しているのだ。　他の、プログラム例や流れ図は大丈夫のようだが？

　その辺は注意して、例題は：　　　**A** が　　

```
1 1 -1
3 4  3
1 0 -3
```

ならば
　特性方程式は　**f(x) = -4 -13x -2x^2 + x^3**　である。

固有値は、この（３次）代数方程式の根である。

固有ベクトルは、固有値が異なる場合には、対応する原行列 Ａ の連立方程式を解けば簡単に判る。

実は、この **Frame** 法の原理の証明は、手ごろな演習問題として、例えば、０．節の「はしがき」、古屋先生の教科書（文献２ **p.102** 演習問題 Ｖ の ５．）に出ている。
海王星の予言者・ルベリエと素粒子論のファデーエフ教授、あるいはフィールズ賞級の女流数学者ファデーエヴァが証明したとか、または最近、米国の某大学、准教授 **John Randall** 博士が云々のレベルほどの難問では無いと思われる。 大学学部生の演習問題レベルの話であった。 実は、すでに或範囲の人々には公知だったのだ。
既に ０．節に述べたが、岩波数学辞典第２版（１９５４年、文献３）には、次のような記述がある：

【 **Frame** 法は原理的には、評価出来る。 しかし、この方法は計算時の「桁落ちがひどい」、特に「原行列Ａの行列式の値が小さい時」には、「精度が悪い」ことが多い。

それに、固有方程式の各次の係数が求められた後に、高次代数方程式の求根をせねばならぬ。 この為の計算量が大変だ。】

散々な評価である。 実用性が乏しいとして、岩波数学辞典でも第３版以降では、話題から消えて仕舞った。 まあ、「神々しい方法」ではないからな！
今回、それが、Ｊ言語の特長を生かせる数学グループによって、復活したのだ。
方法が良くても、道具が悪ければ、残念ながら、通用しない場合があると云うこと！

Ｊ言語による **Frame** 法、同じ事だが、 **Faddeev** 法の関数例は、我らの報告（文献１）の **p.4** の **Feigenp** （特性方程式の求根まで含む）等で与えてある。

３． 逆Ｆｒａｍｅ法 の提案

前節に述べた難点「原行列Ａの行列式の値が小さい時には精度が悪い」への対策を考える。
固有方程式で、固有値ラムダの最高次からの降べき順での係数は、**1,** **trace(A)**・・・・・・・・ 最後が **det(A)** と、この３つは必須であるが、これらは、Ｊ言語派にとっては、計算と云えないほど簡単な事だ。
そこで、真っ先に「行列式を計算」し、それが、「０ または 小さい」時には諦めて別な方法でやる判定をする。 「可」ならば、Ｆｒａｍｅ法を「最終段階の行列式」から逆行して実行し、最後に簡単に **trace(A)** 値 と **1** をアペンドすれば終わる。
計算らしい計算は、これら簡単な ３項値 以外のみをすれば良い。
この方法を **Inverted Feigen** 法、略して **Ifeigen** 法と名付けよう。
その Ｊ **Script** を **Feigen** と並べて下掲する（Ｊ６．０１版）。
（改良する余地も若干あろうが、それは、ただ今の焦点では無い。）

```
Ifeigen =: 3 : 0              Feigen =: 3 : 0
   t=.6!:1''                     t=.6!:1''
 In=. unitm n =. # A =. y      In=. unitm n =. # A =. y
 Ai=. %. A
 Bd=. det y                    Bi =. A
 sig=.(_1)^n                   pi =. trace Bi
 nans=. p =. sig* Bd              nans=. 1, (-pi)
    i=. 1                      i =. 0
```

```
 Bf=. Bd*In
    while. i < n-1  do.            while. i < n  do.
                                                    i =. i + 1
    isig=.i* sig =.(_1)^i
    Bfi =. (Bf indot Ai)           Bi =. (A indot ( Bi - pi * In))
    p=. (trace Bfi) % i            pi =. (trace Bi) % i
    nans =. nans, p                nans =. nans, -pi
    Bf =. Bfi + (p*_1) * In
    i=.i+1
end.                         end.
wr (":(6!1")-t), 'sec'             wr (":(6!1")-t), 'sec'
nans=.nans,((trace y)*_1), 1        nans
)                            )
```

諸関数 unitm : to make unit matrix
det : to make determinant
indot : inner product between matrices
trace : to get diagonal sum

演算結果は、特性方程式の各次の係数が、当然ながら、両者、互いに逆順に並ぶ。**det** 値が先頭になる。 結果値全体並びに演算時間は変わらないので例示は省く。
上記には省いたが、プログラムの実際には、**det** 値が ０ の時には、演算を諦めるような分岐を挿入する。

４． 旧プログラム： 山下の改良案

我らの「Ｊ言語と固有値問題」シリーズの最初の２報（文献１－ａ、－ｂ）で、中野は、甚だ愚直的に長大なるＪプログラムを用いた。 そこで、山下は、その短縮化を努力した。 勿論中野も、余り長大では、印刷原稿の作成や紙代がかさむので、短縮化の努力はした。 その内に、このシリーズの（その３）、（その４）で述べた如く、極めて簡潔、かつじ次数に依らない万能型の **iterative** なムログラム **Naigen** 法に気付き英国コンピュータ学会の **VECTOR** 誌に投稿する事になった（文献１－ｃ）。

山下は、その間の、プログラム短縮化の努力を、記念碑的に残したい希望を持った。現在もどんどん進行中なので、今は、その旧版の改良案の一部のみに言及して置く。
eigen9y の如く、末尾に **y** の字を付けたものが、山下のものである。
中野にも別に、同様な改良的プログラムがある。 それは先頭に、**neigen9** の如くに、文字 **n** を付けてある（長大につき、今はプログラム・リストは掲載しない）。
参考の為に、両者を演算時間で比較して見よう。
山下流（短縮形）と中野流（長大のまま）のプログラムを、マシンは共通に、中野の **Windows XP** マシンで実行して、演算時間の比較テストをした。
データ行列は エル７、即ち **l7=. latin 7** の類である。

７次の場合：
山下流 **eigen7y l7** からの固有多項式と計算時間（秒）と 時間比は
1 _28 _98 2744 2401 _67228 _16807 470596 0.561sec 28
中野流 **neigen7 l7** からは、逆順で
470596 _16807 _67228 2401 2744 _98 _28 1 0.02 sec 1

８次の場合：
山 **eigen8y l8** から

1 _32 _304 5120 29184 _196608 _950272 2.09715e6 9.4318e6 0.841sec 14
中 neigen8 l8 から（、多倍長演算をしている）
9437184 2097152 _950272 _196608 29184 5120 _304 _32 1 0.06sec 1

９次の場合：
山 eigen9y l9 から
1 _45 _270 12150 19683 _885735 _531441 2.39148e7 4.78297e6 _2.15234e8 7.57 84
中 neigen9 l9 から（、多倍長演算をしている）
_215233605 4782969 23914845 _531441 _885735 19683 12150 _270 _45 1 0.09 1

演算の時間の比較では、中野流の方が、数倍から百倍近くも速い。　特に、与行列が高次の場合に大差である。
山下は若干の改良で、その演算時間を６割程度の短縮に成功したが、大勢は変わらぬ。
この時間差の理由は､未だ､よく判らぬが､ただ云える事は､プログラムを短縮して､｢長さ｣で得をすれば、「演算時間」で損をするらしい事だ。　大昔、アセンブラーとフォートランが演算時間の損得を競った時代もありましたと、思い出しました。
この件は、目下も進行中である。　いずれ、報告出来よう。

５． Naigen 法　の改良

我らの目に新しく登場した **Faddeev** や **Frame** 法を除けば、中野が先に提案した **Naigen** 法（文献１－ｃ）は、先ず、最も短い（直接法）固有値求解プログラムである。「再帰法」であるから行列の次元によらない。　つまり次元毎にプログラムを作る必要も無い。
欠点としては、次元の増加と共に、演算時間が急速に増加する事である。
その為、実用上、１０次以上では、１時間以上で、待ちくたびれて仕舞う。
１０次が「限界」か？

既に、固有値－その４（文献１－ｄ）の **p.3** に述べたが、行列の次数と特性方程式の展開項数との関係は

次数	n = 2	3	4	5	6	7	8	9	10
項数		5	16	65	326	1957	13700	109601	986410	9854101

の程度であり、１０次では、ほぼ、１千万項を処理せねばならぬ。
数学ソフト **Maple** で、上の各項を印刷した例では、サイズ４版の紙で、４次では１ペイジだが、６次ではなんと、３３ペイジも要した。　これらの演算プログラムを誤り無く、書き下すとしたら、労力その他、どんな事になろうか？
最近 再び **Maple** で、**Frame** 法の計算の各項を追跡させたら、なんと４次の場合で、はや、能力を超えたとて、下掲の如く、途中打ち切りになって仕舞った。

system error, memory limit 4194304 bytes

直接法の計算とは、この程度にギリギリの限界作業なのである。　従って、 **Naigen**法の旧版では、１０次の計算が限界であった。
最近、この壁を突破して１３次まで処理出来たので、追加報告する。

Naigen 法は、処理の対象行列の次数を逐次、ステップダウンして行き、最後に、２次の行列の集合体として、纏めあげるものであった。
ところが、その後の努力によって、９次の固有値求解関数、例えば、中野の **neigen9** 、山下の **eigen9y** 等が出来たので、上記、ステップダウン（逓降）の終点を、２次では無くて、より手前の９次にすれば、逓降による項数の増加は大幅に避けられる。

その様に、改良した関数を、 **nnaigen** 、さらに時間計時をも組み込んだ関数を **Tnnaigen** とする。　そのテスト結果は

10 Tnnaigen latin 10　演算時間　　0.841 sec （昔の１時間超が１秒以下に短縮）
11　Tnnaigen l11　　演算時間　　8.883 sec
12　Tnnaigen l12　　演算時間　　107.224 sec
13　Tnnaigen l13　　演算時間　　1399.76 sec　（約　２４分）

大成功であった。　データ処理は纏めてやるべきだ！

　次の次数　１４　では、恐らく　５時間以上を要するであろうから、テストはせぬ。しかし、アイデアの有効さは立証された。　今後の改良の指針となろう。

６．　む　す　び

　固有値問題の解の直接法は、「高次代数方程式の求根関数　**p.**」を有する　Ｊ言語が特に有効な分野である。　今後の発展が、更に期待される。
　それにしても、岩波数学辞典の版毎に、**Frame** 法が、出たり引っ込んだりしたのは何故であろうか？　学問には「判り易さ」よりも「神々しさ」の方が大切なのかな？旨く両立させたいものだ。　ルベリエ・ファデーエフ法は、それに成功したか！

文　　献

１）中野嘉弘・山下紀幸：「Ｊ言語と固有値問題（その５）　ファデーエバ法」
JAPLA報告　2007 Sep22 pp.8

－ａ）中野嘉弘「Ｊ言語と高等数学　（固有値問題直接法）」JAPLA報告
2007.4.2 pp.9

－ｂ）中野嘉弘「Ｊ言語と固有値問題（その２）直接法の発展」JAPLA報告
2007.5.26 pp.13

－ｃ）中野嘉弘「Ｊ言語と固有値問題（その３）VECTOR誌投稿原稿」JAPLA報告
2007.6.23 pp.3

－ｄ）中野嘉弘「Ｊ言語と固有値問題（その４）チュートリアル等　LAPACKまで」
JAPLA報告　2007.6.23 pp.9

２）古屋　茂「行列と行列式」培風館、1957 初版、1959 増補版、1998 増補第65刷

３）「岩波数学辞典　第２版」　1954.4 第１刷、1960.12 増訂第１刷、
1968.6 第２版第１刷　p.421

４）http://thamous.univ-rennes1.fr/sites/graph/algeq/

５）http://www.agnesscot.edu/lriddlw/women/prizes.htm

６）http://cache.yahoofs.jp/search/cache?p=James+Sutherland+Frame&ei

７）http://www.lipn.univ-paris13.fr/~banderier/Recipro/node53.html

８）http://math.fullerton.edu/mathews/n2003/FaddeevLeverrierMod.html
Vera Nikolaevna Faddeeva :"Computational Methods of Linear Algebra"
(Translated from The Sussian by Curtis D.Benster) Dover Publications
Inc. N.Y. 1959 ISBN 0486604241

９）山下 FAX （'07. 10.）カシオ・ポケコン数値計算ライブラリー

－ａ）　内田昭宏　他：「カシオ　ポケコン数値計算ライブラリー」
誠文堂新光社　 1985 pp.42-48
「１－６　ファジェーエフ・ルベリ法（固有値）」

－ｂ）http://lupus.is,kochi-u.ac.jp/~shiota/la98/la98.html
同上内　v23.dvi　　数値解析Ｃ　（塩田）　2007年 1月 17日
固有値・固有ベクトル（実対称行列のJacobi 法、一般行列の Frame 直接法）

１０）中野電子メール志村氏宛（'07.Sept.25）：日本物理学会年会（於いて北大）
Sept.23 午後、素粒子論会場講演「５次元重力に於ける対称性」内の話題、
Randall- Sundrum Background。　Lisa Randall (Harvard大女流教授）。
JAPLA報告　2007.6.23 pp.9