

行列の固有値固有ベクトル計算ライブラリの 問題点

京大大型計算機センター

星野 聡

Ⅰ 緒言

行列の固有値・固有ベクトルの計算プログラムでは固有値には等しいものがある場合や固有値の絶対値が非常に大きいものと非常に小さいものがある場合など、特殊なケースについて精度上問題のない様に配慮されている必要がある。

以下においてはこのような種々の問題を取り上げ、これらをチェックするためのテスト行列について考え実例について述べる。テスト行列としては精度に影響がある問題点を良くテストできる、なるべく簡単な行列を選ぶ。テスト行列には精度に影響を与えるパラメタが含まれるものとし、これを変化させて精度の変化を調べるのである。

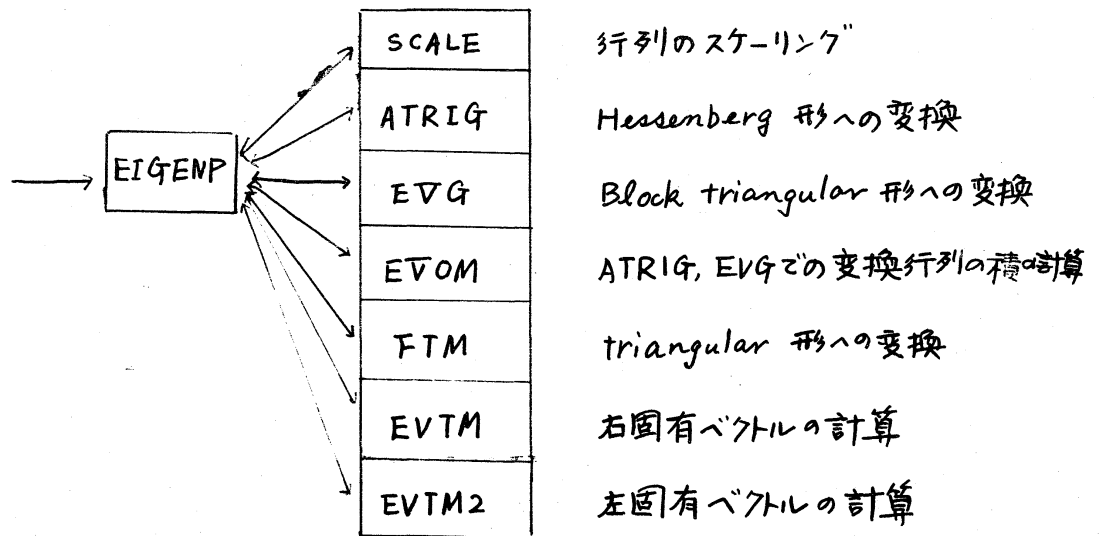
精度を調べるための criterion として種々のものが考えうるが、EISPACK⁽³⁾では $(n \times n)$ の行列 A に対し

$$\mu = \max_{1 \leq i \leq n} \frac{\|Az_i - \lambda_i z_i\|}{10n\epsilon \|A\| \cdot \|z_i\|},$$

を用いている。ここで ϵ は machine precision, λ_i, z_i は i 番目の固有値・固有ベクトルである。また 10 は便宜上とられた factor である。以下の各節ではこの μ を用いる。

その他の criteria としては、かつて京大大型計算機センターの南苑計画の下で南苑を行った (1970年) ライブラリ——実対称行列の固有値・固有ベクトル EVVIS および一般実行列の固有値・固有ベクトル EIGENP —— では次の諸点について調べたことがある。⁽¹⁾

- 計算所要時間, ◦ 残差 $(A - \lambda I)X, Y(A - \lambda I)$
 - 量 $\alpha_i = y_i^T x_i$, ($i=1, \dots, n$), ◦ trace の変化
 - 固有ベクトルの直交性 $y_i^T x_j$, ◦ scaling の影響
- ここで X, Y は右および左固有ベクトル x_i, y_i より成る行列である。また α_i は固有ベクトルの行列要素の変化に対する sensitivity に関する量で、小さいほど sensible である。[文献 (2) p. 68] プログラムの構成は EIGENP を例にとるとオ1図に示す様にいくつかのサブプログラムに分割し各部分のアルゴリズムの改良テストが容易である様にしていた。



EIGENP 内には λ_i の計算が含まれる。また、固有値のみ求めるか右固有ベクトルも求めるか左固有ベクトルも求めるかが選択できる。このようにいくつかのサブルーチンに分割したので flexibility は増したが、サブルーチン間の call に要するオーバーヘッドが、特に小さい行列では無視できない様に思われる。

なお、ここでは実行列のみを扱うことにする。計算は京大大型計算機センターの FACOM 230-75 で単精度により行った。

- [2] 実対称行列における近似的に多重な固有値を持つ場合は、等しい固有値を持つ行列に対して互いに直交する固有ベクトルが計算されるかをテストするため、

テスト行列 I :

$$A = \begin{bmatrix} d & e & & 0 \\ e & d\beta & e\beta & \\ & e\beta & d\beta^2 & \ddots \\ 0 & & \ddots & \ddots \end{bmatrix},$$

を用いた。

$d = \beta = 1$ とし, e を変化させると表 2 をえた。プログラムは SYMQR⁽⁴⁾ を用いた。ただし $EPS = 10^{-6}$, また

$$\nu_1 = \| Z^T Z - I \|,$$

$$\nu_2 = \| Z^T A Z - \Lambda \|,$$

である。Z は eigenvector から成る行列であり $\Lambda = \text{diag}(\lambda_i)$ であり λ_i は固有値である。

e	μ	ν_1	ν_2	t (ms)
0.1	0.09	1.28×10^{-6}	4.35×10^{-8}	183
5×10^{-6}	0.01	1.61×10^{-7}	3.42×10^{-8}	124
10^{-6}	0.01	2.83×10^{-7}	0	58

表 2

EPS を 10^{-6} に選んだ結果として e が 10^{-6} より小さいと off-diagonal の影響は無視されて diagonal の値そのものが固有値となる。固有値が互いに接近していないと off-diagonal

の存在はそのほゞ2乗の形で *diagonal* に影響するが、固有値が接近している場合には *off-diagonal* の要素の大きさと同程度の影響が *diagonal* に加わることに注意しておく。

$$(例) \quad \begin{pmatrix} 1 & 10^{-5} \\ 10^{-5} & 1 \end{pmatrix} \rightarrow \lambda = 1 \pm 10^{-5}$$

$$\begin{pmatrix} 1 & 10^{-5} \\ 10^{-5} & 2 \end{pmatrix} \rightarrow \lambda = 1 + 10^{-10}, 2 - 10^{-10}$$

固有ベクトルの直交性については SYMQR の様に変換行列の積として固有ベクトルを作るのであれば良いが、*inverse iteration* による方式では工夫が必要である。固有値が接近している場合、固有値を少しずらして固有ベクトルを計算することが行なわれるが、これだけでは不十分であり EISPACK に含まれる TINVIT では次の様な工夫がされている。すなわち、接近した固有値に対する固有ベクトルは、すでに求めた分に対して直交するようベクトルを修正する。固有値をずらせる大きさや接近した固有値の定義などは経験的な値がとられている様であり、より数学的な検討を要すると思われる。上のテスト行列で ϵ の大きさを小さくして行くと TINVIT で計算される固有ベクトルは直交していることがわかる。 $(|v_2| < 3.3 \times 10^{-7})$ なお *inverse iteration* においては *overflow* が生ずる可能性があり配慮が必要である。

3 実対称行列の固有値の間で絶対値に大きい開きがある場合

固有値の大きさに非常に大きいものと小さいものがあるとき QR 変換で大きな origin shift が行われると小さい固有値の精度が低下する可能性がある。この影響⁽³⁾は explicit shift と implicit shift で異なるし (文献 6 の p.230) また実際のアルゴリズムにもよると思われる。⁽⁴⁾ テストに用いた SYMQR と HQR2 は implicit shift を用いている。

テスト行列には前節のテスト行列 1 で $|\beta| \ll 1$ に選ぶとよく、左上から右下に進むにしたがって要素は小さくなり、固有値は小さいものから求められる。これに対し逆に左上から右下へ進むにしたがって要素が大きくなる場合

$$A = \begin{bmatrix} \cdot & \cdot & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & d_{\beta^2} & e_{\beta} & \cdot \\ 0 & e_{\beta} & d_{\beta} & e & \cdot \\ \cdot & \cdot & \cdot & \cdot & d \end{bmatrix},$$

もテストできる様にし、前者を $INDIC=0$ 、後者を $INDIC=1$ として選ぶ様にした。 $d=1$, $e=0.3125$, $\beta=0.5$ での SYMQR での結果では $INDIC=1$ の方が長い計算時間を要している。($n=24$ で 42%, $n=12$ で 28% 増) 精度については同々の場合で異なっていた。

EISPACK の TRED1-BISECT-TINVIT-TRBAK1 ルーチンでは三項行列の固有値の計算に *strum sequence* を用いており計算所要時間は $INDIC=0, 1$ にかかわらず同じであった。ただし μ, ν_1, ν_2 は両者で多少異っている。SYMQR と TRED1-BISECT-TINVIT-TRBAK1 による所要時間を下に示す。

n	d	e	β	INDIC	t (ms)	
					TRED1-BISECT-TINVIT-TRBAK1	SYMQR
24	1	0.3125	0.5	0	159	130
"	"	"	"	1	159	185
12	"	"	0.25	0	36	25
"	"	"	"	1	36	32
24	"	0.1	1	0	198	183
"	"	10^{-4}	"	0	158	152
"	"	10^{-7}	"	0	206	59*

* 固有値はすべて 1 が求められた。

表 3

4 行列の *balancing* の影響

固有値・固有ベクトルの誤差を減少させるため行列 A のノルムを小さく出来れば良い。このため A から

$$B = D^{-1}AD,$$

を作り D をうまく選んで B のノルムを減少させ、 B の固有値固有ベクトルから A のそれらを計算する。ここで $D = \text{diag}(d_i)$ 。この工夫は行列の *scaling* により大きい要素と小さい要素が混在しているとき有効なことがある。テストには

テスト行列 2 :

$$A = \begin{bmatrix} n & & 0 & \alpha \\ & n-1 & & \alpha \\ & & \ddots & \vdots \\ 0 & & & 2, \alpha \\ \beta, \beta, \dots, \beta, 1 \end{bmatrix},$$

を用いた。EISPACK の一般実行列の固有値固有ベクトルを計算するルーチン ELMHES - ELTRAN - HQR2 によるテスト結果を才 4 表に示す。表中の BALANC は行列の *balancing* を行うルーチンで ELMHES の前に、また BALBAK は HQR2 の後につけて BALANC による固有ベクトルの補正を行うサブルーチンである。この例の場合には *balancing* により μ を非常に小さくできることがわかる。前述の EVVLS⁽¹⁾ では行列の *balancing* を採用している。

[5] 固有ベクトルが *defective* に近い行列の場合

テスト行列として

$n=5$		BALANC, BALBAKを 使用しない場合		BALANC, BALBAKを 用いる場合	
α	β	λ	t, μ	λ	t, μ
1.0	1.0	-0.19817508 5.3472065 3.2862560 2.2771435 4.2875696	$t=7$ $\mu=0.13$	-0.19817507 5.3472065 3.2862560 2.2771435 4.2875696	$t=8$ $\mu=0.13$
10^{-7}	10^{-7}	5.3319806 -0.18912362 3.3222396 2.2885836 4.2463199	$t=4$ $\mu=1.73$	-0.19817497 5.3472055 3.2862552 2.2771435 4.2875695	$t=9$ $\mu=5.2 \times 10^{-8}$
10^7	10^{-7}	-540.05025 543.99091 5.7081451 1.0494493 4.2903094	$t=7$ $\mu=6766$	-0.19817511 5.3472056 3.2862553 2.2771435 4.2875700	$t=8$ $\mu=3.1 \times 10^{-8}$

表4

テスト行列3:

$$A = \begin{bmatrix} 1 & 1 & & & 0 \\ & 1 & 1 & & \\ & & \ddots & \ddots & \\ 0 & & & \ddots & 1 \\ \alpha & & & & 1 \end{bmatrix},$$

について考えると、固有値入が1からずれる距離は $\alpha^{1/n}$ である。 α の大きさが小さくて α が無視されてしまう場合にも

固有値の誤差が大きくなることになる。したがって高精度計算による必要があるが balancing によって精度が上げられることがあることも経験した。(下表)

$n=5$	BALANC, BALBAK を 使用しない場合		BALANC, BALBAK を 用いる場合	
	λ	ϵ	λ	ϵ
10^{-5}	0.91910170 $\pm 0.058775123i$ 1.0308987 $\pm 0.095103206i$ 1.0999992	22	0.91909832 $\pm 0.058778524i$ 1.0309017 $\pm 0.095105655i$ 1.1	20
10^{-7}	1.0 1.0 1.0 1.0 1.0	2	0.96779245 $\pm 0.023400151i$ 1.0123022 $\pm 0.037862240i$ 1.0398107	22

表5 表4表と同様 ELMHES-ELTRAN-HQR21による
計算結果 (ただしこの表は MACHEP = 10^{-7} とした場合)

$\alpha = 0$ のときは固有値はすべて 1 であり固有ベクトルはた
び一般となる。 $\alpha \simeq 0$ では固有ベクトルはほぼ等しい。一
対称実行列では互いに直交する固有ベクトルが存在する。こ
れを考えると一般行列の固有ベクトルの計算は途中の変換行
列の積を作った行く方法によるのが良いであろう。

6 その他の問題点

(i) 実対称三項行列での QR 変換における origin shift は行列 A の右下の (2×2) 行列の固有値のうち a_{nn} に近いものを採用すると globally に収束する。そして k 回目の QR 変換による a_{ij} を $a_{ij}^{(k)}$ とするとき $|a_{n,n-1}^{(k+1)}| \leq |a_{n,n-1}^{(k)}|$, $|a_{n-1,n-2}^{(k+1)} a_{n,n-1}^{(k+1)}| \leq |a_{n-1,n-2}^{(k)} a_{n,n-1}^{(k)}|$ が成立することも証明されている。⁽⁵⁾ 非対称行列については origin shift をどう選べば良いかは、きりしない様である。QR 変換の所要回数は一般に少ないが、stopping condition の与え方によってループになる恐れがあるので、変換回数がある回数 (EISPACK では 30 回) をこえると return をさせる様にしている。

一般にこの様な constant はなるべく避けられる様にあるべきである。

(ii) 固有ベクトルの normalization

ライブラリで一般に異なるが ユークリッド・ノルムを 1 にするのが良いと思われる。

(iii) 固有値固有ベクトルの計算誤差

近似的な固有値固有ベクトルの値を用いてそれらの誤差を求める機能をライブラリに持たせてはどうかと思われる。

文献(2) p.637 の方法で実際の手続きは文献(7)によるのが一方法で Gerschgorin の定理を利用するものである。

参考文献および資料

(1) 京都大学大型計算機センター - 用途資料

EVVIS, EIGENP (作成者 星野聰 1970年)

(2) Wilkinson, J. H. (1965), The Algebraic Eigenvalue problem, Clarendon Press.

(3) Smith, B. T., J. M. Boyle, B. S. Garbow, Y. Ikebe, V. C. Klema, C. B. Moler, (1974). Matrix Eigensystem Routines - EISPACK guide, Springer (Lecture Notes in Computer Science Vol. 6)

(4) Stewart, G. W. (1970) Incorporating Origin Shifts into QR Algorithm for Symmetric Tridiagonal Matrices, Comm. ACM, 13, 365-367.

同じく

Stewart, G. W (1970), Algorithm 384, Eigenvalues and Eigenvectors of a Real Symmetric Matrix, Comm. ACM, 13, 369-371

(T=Vレ プロ グラム内の shift の計算に用いられる変数について REAL K1, K2 が抜けているので修正必要)

- (5) Wilkinson, J.H. (1968), Global Convergence of Tridiagonal QR Algorithm with Origin Shifts, Linear Algebra and its Applications, 1, 409-420.
- (6) Wilkinson, J. H and Reinsch, C. (1971), Handbook for Automatic Computation, vol. II, Linear Algebra, Part II, Springer.
- (7) Varah, J.M. (1968), Rigorous machine bounds for the Eigensystem of a General Complex Matrix, Math. Comp. 22, 793-801.