

ランチョス法その後

筑波大 電子情報工学系 名取 亮
(Makoto Natori)

慶應義塾大 数理科学科 野寺 隆
(Takashi Nodera)

1. はじめに

高橋・名取 [12] の論文のものは、1970年11月に数理解析研究所の研究集会「数値計算のアルゴリズムの研究」で発表したもので、講究録は、翌1971年 4月に発行された。この研究は、1969年に東京大学大型センターで主催した「大次元行列の計算に関するシンポジウム」において、法政大学の瀬部先生がランチョス法による三重対角化を途中でやめても、最大・最小の固有値からいくつか（外側の固有値）は精度よく求まるという講演をされ、高橋先生がそれに興味を持たれたのがきっかけであった。この論文では、(1) 再直交化なしでも外側の固有値は正確に求まること、(2) 直交性の崩れがどう伝播するかを調べ、直交性の崩れと固有値の収束が対応することを述べた。1971年、Paige がPh.D論文 [5] でランチョス法の誤差解析を行い、その結果の一部が1972年と1976年に雑誌 (Paige [6, 7]) に発表された。その後、Parlett & Scott [8] は、選択的直交化を提案し、Parlett [15] で、直交性の崩れをどうして検出するかを述べた。一方、Cullum & Willoughby [4] は、中間の固有値および固有ベクトルを求める時にも再直交化をせずに三重対角化を続行する方法を提案した。その場合、“偽の固有値”をどう見分けるかが、重要なポイントとなる。この方法に対して、高橋 [14] は再直交化をおこなわないランチョス3重対角行列をどこで終了したらよいかという重要な問題点に対して、物理的な直感をもとにした重要なコメントを与えている。前者については、1980年のParlett [2] の本に、後者については、Cullum & Willoughby [3] の本に詳しい。近年、Parlett と彼のファミリーは、直交性の崩れに関する高橋・名取 [12] の論文を拡張した様々な理論の展開を発表している。特に、Simon [10, 11] は、高橋、名取 [12] の直交性の崩れに関する漸化式をモニターすることで、再直交化の時期を判断し、部分的に直交性が崩れていると思われるランチョス・ベクトルのみについて再直交化する部分的な再直交化を提案している。また、ランチョス法とCG法の直交性に関するデータフローを用いた物理的な解説

が高橋・野寺 [13] にある。

2. ランチョス法

固有値問題：

$$A y = \lambda y, \quad (2.1)$$

に対するランチョス法は、 $n \times n$ の対称行列 A を直交変換により

$$A V = V T, \quad (2.2)$$

$$\begin{array}{c} \leftarrow n \rightarrow \\ \uparrow n \downarrow \end{array} \begin{array}{|c|} \hline A \\ \hline \end{array} \begin{array}{|c|} \hline V \\ \hline \end{array} = \begin{array}{|c|} \hline V \\ \hline \end{array} \begin{array}{|c|} \hline T \\ \hline \end{array}$$

のように変換して、3重対角行列 T を構成する方法である。即ち、適当な v_1 から出発し

$$\left\{ \begin{array}{l} \alpha_j = v_j^T A v_j, \\ A v_j = \beta_{j-1} v_{j-1} + \alpha_j v_j + \beta_j v_{j+1}, \\ \|v_j\| = 1, V^T V = I, \end{array} \right. \quad (2.3)$$

を満たすように、 α_j 、 β_j を決定する方法である。

現在、最もよく用いられている対称な $n \times n$ 行列 A に対する固有値問題のランチョスの算法は、Paige [5.6] の論文の中で述べられているものである。

〔ランチョスの算法 (Paige (1972))〕

(1) 任意のベクトル v_1 ($\|v_1\| = 1$)を選び $u_1 = A v_1$ を計算する。

(2) 次の操作を繰り返し、 α_j と β_j を計算する。(j=1,2,3,...)

$$\alpha_j = u_j^T v_j, \quad (2.4a)$$

$$r_j = u_j - \alpha_j v_j, \quad (2.4b)$$

$$\beta_j = \|r_j\|, \quad (2.4c)$$

$$v_{j+1} = r_j / \beta_j, \quad (2.4d)$$

$$u_{j+1} = A v_{j+1} - \beta_j v_j. \quad (2.4e)$$

この(2)の操作をランチョス・ステップと呼び、これを1つの式で表すと

$$\beta_j v_{j+1} = A v_j - \alpha_j v_j - \beta_{j-1} v_{j-1} \quad (2.5)$$

となり、次のような行列形式で書くことができる。

$$\begin{array}{|c|} \hline A \\ \hline \end{array} \begin{array}{|c|} \hline v_j \\ \hline \end{array} = \begin{array}{|c|} \hline v \\ \hline \end{array} \begin{array}{|c|} \hline T_j \\ \hline \end{array} \leftarrow \beta_j$$

$$= \begin{bmatrix} V_j \end{bmatrix} \begin{bmatrix} T_j \end{bmatrix} + \begin{bmatrix} 0 & * \end{bmatrix} \leftarrow \beta_j v_{j+1}$$

$$AV_j - V_j T_j = \beta_j v_{j+1} e_j^T, \quad (2.6)$$

ただし, $V_j = (v_1, v_2, \dots, v_j)$, $e_j^T = (0, 0, \dots, 1)$, そして

$$T_j = \begin{bmatrix} \alpha_1 & \beta_1 & 0 & \cdot & \cdot \\ \beta_1 & \alpha_2 & \beta_2 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \beta_{j-2} & \alpha_{j-1} & \beta_{j-1} \\ \cdot & \cdot & 0 & \beta_{j-1} & \alpha_j \end{bmatrix}. \quad (2.7)$$

ベクトル v_j は, 正規直交ベクトルであるので,

$$V_j^T V_j = I_j \quad (2.8)$$

を満足する。ただし, I_j は $j \times j$ の単位行列である。

この算法は, 丸め誤差の影響を受けなければ, $\beta_j = 0$ ($\exists j \leq n$) で終了することになる。三重対角行列 T_j の固有値を $\{\theta_i\}_{i=1}^j$ とし, その固有ベクトルを $\{s_i\}_{i=1}^j$ としよう。 T_j の固有値は, Ritz値とも呼ばれ, ベクトル $v_1, v_2, \dots, v_j$ で張られた部分空間における A の固有値の Rayleigh-Ritz 近似である。さらに, $y_i = V_j s_i$ ($i = 1, 2, \dots, j$) を Ritzベクトルと言う。

現在の電子計算機のように有限桁の計算でランチョス法を実行する場合, 当然, 丸め誤差のことを考えなければならないので, (2.6) と (2.8) 式は次のように修正される。

$$AV_j - V_j T_j = \beta_j v_{j+1} e_j^T + F_j \quad (2.9)$$

$$V_j^T V_j = I_j - H_j \quad (2.10)$$

ただし, F_j と H_j は, 丸め誤差の行列である。ここで, (2.6) 式がほぼ満足されているにもかかわらず ($\|F_j\|$ が小さい), ランチョス・ベクトル間の直交性が完全に崩れる場合があることも事実である ($\|H_j\| \geq 1$)。

Paige [6] は, 実際に Ritzベクトルを計算せずに, 残差ノルム $\|Ay_i - y_i \theta_i\|$ を計算することで, j ステップにおけるランチョス法の収束性をモニターすることを提案した。それは, 次の事実に基づいている。

(2.6) 式に右から s_i を掛けると次の式が得られる。

$$AV_j s_i - V_j T_j s_i = \beta_j v_{j+1} e_j^T s_i \quad (2.11)$$

となり, Ritzベクトルの定義より, $y_1 = V_1 s_1$ を代入し, 整理すると次の式が得られる。

$$\begin{aligned} A y_1 - V_1 s_1 \theta_1 &= \beta_{j+1} \sigma_{j+1} v_{j+1} \\ A y_1 - y_1 \theta_1 &= \beta_{j+1} \sigma_{j+1} v_{j+1} \end{aligned} \quad (2.12)$$

ただし, $\sigma_{j+1} (= e_j^T s_1)$ は, Ritz値 θ_1 に対する固有ベクトル s_1 の第 j 成分である。ここで, (2.12) 式のノルムをとれば, 次のようになる。

$$\|A y_1 - y_1 \theta_1\| = \beta_{j+1} |\sigma_{j+1}| \equiv \boxed{\beta_{j+1}} \leftarrow \text{小なら } \theta_1 \text{ は収束} \quad (2.13)$$

ランチョス法は, 通常, (2.9) 式を満足するが, 幸いなことに, 実際の計算では, $\|F_j\|$ は常に小さい場合が多い。よって, β_{j+1} は小さな3重対角行列を解くことによって計算することが可能で, その計算量も少なく, Ritzベクトルの残差ノルムを正確に予測できる。しかし, ランチョス法が生成するランチョス・ベクトル間の線形独立性の崩れが, $\|y\| \ll 1$ となるRitzベクトルを生成する。こんな場合に, 固有値の正確な予測に β_{j+1} を利用できない恐れがある。Paige [7] の解析によれば, $\|y\| \ll 1$ は, 行列 A の同じ固有値を近似するすべてのRitz値が集積している場合にのみ起こる。しかし, その集積したRitz値のなかで, 少なくとも1つ $\|y\| \approx 1$ となるRitzベクトルが存在するならば, そのようなRitz値の集積は, 常に計算機の演算精度まで収束するのである。即ち, Paige [6] の最も重要な事柄は, 次のような収束性と直交性の崩れの関係を見つけたことであった。

$$\left[\begin{array}{c} \text{Ritz値} \\ \text{Ritzベクトル} \end{array} \right] \text{の収束} \iff \left[\begin{array}{c} \text{ランチョス・ベクトルの} \\ \text{直交性の崩れ} \end{array} \right]$$

ここで, 実際の計算で零となる $V_j^T v_{j+1}$ をモニターすることは, 興味のあることである。 v_{j+1} に関してRitzベクトルとの内積を考えると次のようになる。

$$S_j^T V_j^T v_{j+1} = Y_j^T v_{j+1} \quad (2.14)$$

Paige [5,7] によれば,

$$|y_1^T \beta_{j+1} v_{j+1}| \cdot |\sigma_{j+1}| = r_{j+1} \quad (2.15)$$

ただし, $r_{j+1} \approx \varepsilon \|A\|$ である。また, (2.15) 式は, 次式のように書き直すことができる。

$$y_1^T v_{j+1} = r_{j+1} / \beta_{j+1} \quad (2.16)$$

ただし, β_{j+1} は演算の精度 ε に対する y_1 の残差ノルムであることは明らかである。よ

って、直交性は、Ritzベクトルの収束する方向でのみ崩れることになる。即ち、 $y_1^T v_{j+1}$ の値が大のとき、 v_{j+1} と $v_1, v_2, \dots, v_j$ の直交性が崩れることになる。また、Ritzベクトルの線形独立性が崩れるのは、第2のRitzベクトルが最初のものに平行に現れた場合にのみ起こることになる。即ち、第2のRitzベクトルは、 (β_{j+1}) をほぼ $\sqrt{\varepsilon} \|A\|$ 程度に) 最初のRitzベクトルを固有ベクトルから摂動させるが、その後、この2つのベクトルは第3のRitzベクトルのコピーが出現するまでともに収束することになる ($\beta_{j+1} \approx \varepsilon \|A\|$)。よって、ランチョス法は、望ましい全ての固有値が収束するまで、即ち線形独立性の崩れを過ぎるまで反復を行うのが望ましいことになる。しかし、この方法で十分精度の良い固有値を計算することができるが、残念なことに偽の固有値が現れることになり、これを見分ける必要が生まれてくる。上記の結果をまとめると次のようになる。

- (i) 外側の固有値は、(経験的に言って) 小さな j ($\sim 2\sqrt{n}$) で収束する。
- (ii) 更に続けると、ランチョス・ベクトル v_j 間の直交性が崩れるため、収束した固有値が重複して現れる。
- (iii) 偽の固有値が現れる。

3. 直交性の崩れ

高橋、名取 [12] は、ランチョス法が生成するランチョス・ベクトルの直交性の崩れの伝播に関して、重要な漸化式を与えたが、近年、Simon [10] はそれについて次のような詳しい解析をおこなった。

$\omega_{1k} = v_1^T v_k$ とおけば、 ω_{1k} は、次の漸化式を満足する。

$$\begin{aligned} \omega_{kk} &= 1, \quad k=1, \dots, j \\ \omega_{kk-1} &= \phi_k, \quad k=2, \dots, j \end{aligned} \quad (3.1)$$

$$\omega_{j+1k} = 1/\beta_j \cdot [\beta_k \omega_{jk+1} + (\alpha_k - \alpha_j) \omega_{jk} + \beta_{k+1} \omega_{j,k-1} - \beta_{j-1} \omega_{j-1,k}] + \phi_{jk}, \quad 1 \leq k \leq j$$

ただし、 $\omega_{jk+1} = \omega_{k+1j}$ 、 $\omega_{k0} = 0$ 。また、 ϕ_k, ϕ_{jk} は、ある適当な乱数である。Simon [10] の統計的な解析によれば、 ϕ_k, ϕ_{jk} は次式のようなになる。

$$\phi_k = \varepsilon n \frac{\beta_1}{\beta_j} \Psi, \quad \Psi \in N(0, 0.6) \quad (3.2)$$

$$\phi_{jk} = \varepsilon (\beta_k + \beta_j) \Phi, \quad \Phi \in N(0, 0.3)$$

ただし、 N は正規乱数である。

また、再直交化を行った後では、 $v_{j+1}^T v_j$ は、零にリセットされるので、これら上記の値は丸め誤差のレベルになると推測され、

$$\omega_{j+1} \leq \epsilon \in N(0, 1.5) \epsilon$$

となる。

4. 再直交化

ランチョス法の算法を続けていくと、当然のことながらランチョス・ベクトル間の直交性が崩れてゆくことになる。そこで、以前から行われてきたのは、計算した全てのランチョス・ベクトルをグラム・シュミット法によって、再直交化し、各々のベクトルの間の直交性を保持することであった。しかし、この操作には、多大なる計算時間を必要とし、また、過去に計算したランチョス・ベクトルを全て記憶しなければならず、あまり良い方法とは言えない。近年の研究によれば、この操作を全てのベクトルについて行う必要はないことも明らかにされてきた。4.1 節では、再直交化法の中で、準直交化 (semiorthogonalization) といわれるものについて述べることにする (Simon [11])。

また一方では、ランチョス・ベクトルの間の直交性が崩れても、再直交化などを全然考えず、3重対角化をどんどん続行する方法もある。この操作で、行列Aの全固有値を計算することも可能であるが、3重対角行列 T_j の次数が大きくなればなる程、端の固有値が何重にも重複して現れたり、偽の固有値が多数現れることになる。よって、得られた固有値の中から、正しい固有値を見つけ出す操作を必要とする。この手法について4.2 節で述べることにする。

4.1 再直交化あり

再直交化法の目的は、ランチョス・ベクトルの間の直交関係を保つことになる。即ち、各々のベクトルの間の直交性の崩れをある程度までに抑えることにある。j ステップでのランチョス・ベクトルの間の直交性のレベルを次のように定義する。

$$\kappa_j = \max_{1 \leq k \leq j-1} |v_j^T v_k|$$

もし生成されたランチョス・ベクトルに対して、 $\kappa_j \approx \epsilon$ (演算の精度) を保つためには完全再直交化を行う以外に方法はない。

(1) 完全再直交化: FOR (Full Reorthogonalization)

$$r_j \equiv \beta_j v_{j+1} = A v_j - \alpha_j v_j - \beta_{j-1} v_{j-1} - f_j \quad (4.2a)$$

$$r_j \leftarrow r_j - \sum_{k=1}^j (r_j^T v_k) v_k$$

これは、各ランチョス・ステップにおいて以前の v_j 全てに対して新しい v_{j+1} を直交化させる方法である。明らかに、この方法によれば、ランチョス・ベクトル間の直交性の崩れを丸め誤差のレベルに抑えることができる。しかし、完全再直交化法の計算量はステップが進むに従って、膨大なものになるのであまりおすすめできる手法ではない。

Parlett and Scott [8] は、各々のランチョス・ベクトルに $\kappa_j \approx \varepsilon$ が成立していなくても、第2のRitzベクトルのコピーが現れる前に $\kappa_j = \sqrt{\varepsilon}$ の準直交性が成立していれば、正しい固有値が得られることを報告している。よって、次にあげる2つの算法が得られることになる。

(2) 部分的直交化: PRO (Partial Reorthogonalization — Simon (1984))

この方法は、Simon [10] により提案された方法で、言うなれば、(1) の手法のように再直交化を全てのランチョス・ベクトルについて行うのではなく、直交性が崩れていると思われるランチョス・ベクトルを(3.1) 式の漸化式で $\omega_{j+1, k}$ をモニターすることで見つけ、それに対して次の手順で部分的に再直交化するものである。

(i) ランチョスの算法を実行する。

$$r_j \equiv \beta_j v_{j+1} = A v_j - \alpha_j v_j - \beta_{j-1} v_{j-1} - f_j \quad (4.3)$$

(ii) $v_{j+1}^T v_k$ ($k=1, 2, \dots, j$) に対して、 $\omega_{j+1, k}$ を(3.1) の漸化式を利用して計算する。

(iii) $\omega_{j+1, k}$ からの情報に基づいて、 $L(j) = \{k \mid 1 \leq k \leq j, \omega_{j+1, k} > \sqrt{\varepsilon}\}$ を決定し、再直交化する。

$$r_j \leftarrow r_j - \sum_{k \in L(j)} (r_j^T v_k) v_k \quad (4.4)$$

(3) 選択的直交化: SO (Selective Orthogonalization — Parlett & Scott (1979))

(i) ランチョスの算法を実行する。

$$r_j \equiv \beta_j v_{j+1} = A v_j - \alpha_j v_j - \beta_{j-1} v_{j-1} - f_j \quad (4.5)$$

(ii) $L(j) = \{k \mid 1 \leq k \leq j, \beta_{j, k} < \sqrt{\varepsilon}\}$ を決定する。

(iii) $L(j)$ に対して、 $y_k^{(j)} = V_j s_k$ を計算し、次のランチョス・ベクトルを計算する。ただし、 $L(j) = \emptyset$ のときは、この計算は行わない。

$$r_j \leftarrow r_j - \sum_{k \in L(j)} (r_j^T y_k^{(j)}) y_k^{(j)} \quad (4.6)$$

この手法は、(2.16) 式の情報を利用することなく、繰り返し現れるRitzベクトルのコピーを抑制する手法として、収束したRitzベクトルとのみ直交化する方法である。(2.16)

式が、この方法の必要とする考え方の理論的な正当性を与えている。

4.2 再直交化なし

再直交化を行わないで、求めたい固有値が全て求まれば最良なことなのであるが、ランチョス・ベクトルの直交性の崩れから、両端の固有値が重複したり、偽の固有値が現れたりしてなかなかうまく求まらないこともよく知られている。しかし、Cullum & Willoughby [4] は、再直交化を行わなくても全ての固有値と固有ベクトルが求まることをいくつかの実例で示した。その方法は、3重対角化を行列の次元よりもさらに続行し、より大きな T_j ($j \gg n$) を作り、QR法やQL法を用いて全部の固有値を計算し、正しい固有値を判別するというものである。しかし、この方法には、下記の問題点があることを高橋 [14] は指摘している。

- (i) j を十分に大きくすれば、必ず全ての固有値が求まるという保証がない。
 - (ii) j をどの程度にとればよいか事前に知ることができない。
 - (iii) 3重対角化した行列 T_j の固有値には、多くの偽の固有値が含まれているのでそれらを見分けて取り除く必要がある。
 - (iv) 見掛け上の重根と真の近接根との区別が難しい。
 - (v) 異なる固有値が全部で何個あるかを事前に知ることができないから、実際に3重対角化によって得られた行列 T_j の固有値がもとの行列 A の全ての固有値であるという保証が得られない。
- (iii) と (iv) の問題点に関して、Cullum & Willoughby [4] は、高精度計算を用いて繰り返し現れる固有値や、偽の固有値を見分ける方法を提案した。しかし、この方法の最大の難点は、ランチョス3重対角化をどこで終了させるかという点にある。経験的に、 $j = cn$ ($c = 2 \sim 5$) と言われているが、固有値が縮重しているような場合には、 $j \geq 10n$ という場合もあり、 n が十分大きい場合には、この時の3重対角行列の固有値の計算には、膨大な計算時間を必要とする。この点に関して、高橋 [14] は、適切な j を決定する算法と物理的な考察を与えている。また、Parlett & Reid [9] は、Tracking法 (T_j の次数を増やししながら、Ritz値の収束していく状況を追跡し、収束したRitz値を取り出す。) を採用し、このような問題解決の指針を与えている。また、高橋、名取 [12] も同様な計算方法を採用して、固有値および固有ベクトルを計算している。

4.2.1 偽の固有値の判別 (Cullum & Willoughby (1981))

偽の固有値の判別は、次のようにする。

(1) T_j : $j \times j$ の 3 重対角行列,

及び,

$\overline{T}_2$: T_j から第 1 行, 第 1 列を除いた $(j-1) \times (j-1)$ の行列,

を作り, 高精度計算によって, 各々の固有値を計算する。

(2) 下記の表に従って, T_j と $\overline{T}_2$ を比較して, 偽の固有値を取り除き正しい固有値を見つける。

表3.1 偽の固有値の判定法

Case	判定	T_j の固有値の重複	$\overline{T}_2$ の固有値
(1)	Accept	Yes	Yes
(2)	<Reject>	No	Yes
(3)	Accept	No	No

(1) T_j の重複固有値は, 「本物」, $\overline{T}_2$ の固有値にもなっている。

(2) T_j の単純固有値で, $\overline{T}_2$ の固有値でもあるものは「偽物」。

(3) T_j の単純固有値で, $\overline{T}_2$ の固有値でないものは「本物」。

実際の数値例でこの操作を示すと下記のようになる。下線部の固有値が偽固有値として取り除かれることになる。

	判定	T_j	$\overline{T}_2$
k=65	<ACCEPT>	1.070513330620176	1.070513330620176
k=66	@REPEAT@	1.070513330620177	1.070513330620178
k=67	@REPEAT@	1.070513330620180	1.075799503043244
k=68	<ACCEPT>	1.081014052771011	1.081014052771011
k=69	@REPEAT@	1.081014052771012	1.081014052771015
k=70	@REPEAT@	1.081014052771015	1.085871332159758
k=71	<ACCEPT>	1.191508321545976	1.191508321545976
k=72	@REPEAT@	1.191508321545976	1.191508325318810
k=73	*REJECT*	1.191508325318809	1.192025261968861
k=74	<ACCEPT>	1.224174788449782	1.224174788449783
k=75	@REPEAT@	1.224174788449784	1.224174887505310
k=76	*REJECT*	1.224174887505309	1.232450808242045
k=77	<ACCEPT>	1.258024362423952	1.258024362423951
k=78	@REPEAT@	1.258024362423954	1.258026924223992
k=79	*REJECT*	1.258026924223993	1.260866993725309

5. 数値例

ランチョス法の有効性を示す為に、

- (1) 「再直交化を行うランチョス法」では、選択的直交化法、
- (2) 「再直交化を行わないランチョス法」については、Cullum & Willoughby [4] の偽の固有値を見分ける判定法、

を用いた数値例について述べる。

(例1) ランチョス法に選択的直交化法を利用した場合の数値例

次のような6次元 ($n=6$) の対角行列 A を考える。

$$A = \text{diag} (0, 0.0025, 0.0005, 0.00075, 0.001, 10) ,$$

$$v_1 = 6^{-1/2} (1, 1, 1, 1, 1, 1) : \text{初期値},$$

$$\varepsilon = 10E-16 \text{ (演算の精度)} .$$

表5.1 は、ランチョス法により3重対角化した行列 T_7 の各成分と $\|I - V_j^T V_j\|$ を計算したものである。また、表5.2 は各反復における T_j の固有値 (Ritz値) と、再直交化するための判定条件となる β_{j+1} およびRitzベクトルのノルムを示したものである。この表から、 $\beta_{4,4}$ が $\beta_{4,4} < \sqrt{\varepsilon}$ の条件を満足するので、第4の固有値 θ_4 ($= 9.999999999999965$, この場合には T_j の最大固有値) に対するRitzベクトルを計算し、これと r_j を直交化し、次のランチョス・ステップを続行する。即ち、 $j=4$ で選択的な直交化を行い、更に2回のランチョス・ステップを続行し、 T_6 (Ritz値) を計算すると、次の結果が得られることになる。

表5.3 選択的直交化を $j=4$ で行った T_6 の固有値

j	θ	固有値
6	1	0.1978295315785178E-14
	2	0.25000000000019150E-03
	3	0.50000000000019653E-03
	4	0.75000000000018928E-03
	5	0.1000000000001820E-02
	6	9.999999999999965

このように選択的直交化を行った場合と、そうでない場合のランチョス・ベクトルの間の直交性をみると、表5.4 と表5.5 のようになる。これらの事柄から、ランチョス・ベクトルの直交性の崩れは、固有値10に対する第2のRitzベクトルのコピーが現れたことによることが分る。また、 $j=4$ で選択的な直交化を行ない、ランチョス3重対角化を行ったものは、全てのランチョス・ベクトルに対してある程度の直交性が保持されているこ

表 5.1 3重対角行列 T_j の要素

$A = \text{diag}(0, 0.00025, 0.0005, 0.00075, 0.001, 10.)$ $v_1 = 6^{-1/2} (1, 1, 1, 1, 1, 1)$			
j	α_j	β_j	$\ I - V_j^T V_j\ $
1	1.667083333333333	3.72659363747748	0.
2	8.333416579162292	0.8660253965848979E-03	0.69E-17
3	0.5000700034993815E-03	0.2958039902115344E-03	0.68E-13
4	0.5000046430893902E-03	0.2535462769028216E-03	0.20E-08 ←
5	0.5000398050625197E-03	0.6136791911709209E-03	0.77E-04
6	9.051718071527039	2.929693876937098	1.26
7	0.9487819015256840	0.2263664586082772E-15	0.41

表5.2 Ritzベクトルからの情報

j	i	θ_{ji}	β_{ji}	$\ y_{ji}\ $
3	1	0.1464378593296902E-03	0.1792801E-03	1.000000000000147
	2	0.8535446398046466E-03	0.1792883E-03	0.999999999998148
	3	9.999999999999965	0.2004557E-07	1.000000000000002
4	1	0.3902013007893531E-04	0.1150416E-03	0.999999999996549
	2	0.4999924366006440E-03	0.1944611E-03	1.000000000000060
	3	0.9609745755441913E-03	0.1150478E-03	1.000000000000133
	4	9.999999999999965	0.5929862E-12 ←	1.000000000000002
5	1	0.6211135460530229E-05	0.798397	0.999999999991327
	2	0.3184727833117499E-03	1.911407	1.000000000000676
	3	0.6815501604137788E-03	1.911585	0.999999999996056
	4	0.9937928681004200E-03	0.798565	1.000000000000716
	5	9.999999999999965	0.173483E-12	1.000000000000002
6	1	0.6208045170576507E-05	0.418277E-20	0.999999999990801
	2	0.3184550714314634E-03	0.100137E-20	1.0000000000000626
	3	0.6815324460819470E-03	0.100141E-20	0.999999999995643
	4	0.9937897767053343E-03	0.418339E-20	1.000000000000693
	5	9.051718113134897	0.226366E-15	0.999999999999603
	6	9.999999999999965	0.302652E-20	1.000000000000002
7	1	0.1875214351600081E-14	0.328829	0.9999999999985274
	2	0.2500000000001565E-03	1.327053	1.000000000000837
	3	0.4999999999980795E-03	1.990683	0.999999999999537
	4	0.7500000000001318E-03	1.327188	0.9999999999994851
	5	0.1000000000001723E-02	0.331813	1.000000000001152
	6	9.999999999999965	0.898493	0.999999999999372
	7	9.999999999999965	0.446167	1.000000000000002

表5.4 $V_s^T V_s$ (再直交化なし)

1.0	-0.6E-17	0.1E-13	-0.6E-09	-0.2E-04	-0.3
-0.6E-17	1.0	-0.5E-13	-0.1E-08	-0.5E-04	-0.9
0.1E-13	-0.5E-13	1.0	0.7E-14	-0.4E-08	-0.7E-04
-0.6E-09	-0.1E-08	0.7E-14	1.0	0.5E-14	-0.2E-08
-0.2E-04	0.5E-04	-0.4E-08	0.5E-14	1.0	-0.2E-14
-0.3	-0.9	-0.7E-04	-0.2E-08	-0.2E-14	1.0

表5.5 $V_s^T V_s$ (選択的直交化 $j=4$)

1.0	-0.6E-17	0.1E-13	-0.6E-09	-0.2E-12	0.8E-09
-0.6E-17	1.0	-0.5E-13	-0.1E-08	0.9E-13	0.2E-08
0.1E-13	-0.5E-13	1.0	0.7E-14	-0.3E-13	0.5E-12
-0.6E-09	-0.1E-08	0.7E-14	1.0	0.5E-13	-0.6E-13
-0.2E-12	0.9E-13	-0.3E-13	0.5E-13	1.0	-0.5E-14
-0.8E-09	0.2E-08	0.5E-12	-0.6E-13	-0.5E-14	1.0

とがわかる。

(例2) 長方形領域上での2次元ラプラス方程式の境界値問題: $\nabla^2 u = 0$ を5点差分近似した行列の固有値問題を考える。ただし、離散化行列Aは、次のようになる。

$$A = \begin{bmatrix} B & -I & & & \\ -I & B & -I & & \\ & -I & B & -I & \\ & & -I & B & -I \\ & & & -I & B \end{bmatrix} \quad \begin{matrix} \uparrow \\ \text{nb} \\ \downarrow \end{matrix} \quad (5.1)$$

← nb →

$$B = \begin{bmatrix} 4 & -1 & & & \\ -1 & 4 & -1 & & \\ & -1 & 4 & -1 & \\ & & -1 & 4 & -1 \\ & & & -1 & 4 \end{bmatrix} \quad \begin{array}{c} \uparrow \\ b \\ \downarrow \end{array} \quad (5.2)$$

|← b →|

この行列の正しい固有値は、次のようになる。

$$\lambda_{ij} = 4 \left(\sin^2 \pi i / 2(nb+1) + \sin^2 \pi j / 2(b+1) \right) \quad (5.3)$$

$i = 1, 2, \dots, nb, \quad j = 1, 2, \dots, b$

今、 $n=200$ ($nb=10$, $b=20$) の場合を考えることにする。3重対角化を $j=200$ で打ち切って T_j ($j=200$) の固有値を計算すると、その結果は次のようになる。

表5.6 T_j ($j=200$) の固有値

固有値の精度	個数
5桁～9桁	14個
10桁～15桁	75個

ここで、さらに3重対角化を続行して、 $j=2n$ ($=400$)、 $3n$ ($=600$) とランチョス法の反復を行い、前節の表3.1に従って、真の固有値（重複固有値も含む）と偽の固有値を判定すると、表5.7 及び表5.8 のような結果が得られることになる。また、表5.9 は再直交化を全然行わず、3重対角化をマトリックスの次元の2倍行い、Cullum&Willoughby [4] の判定法により全ての固有値を計算したものである。重複固有値や偽の固有値がたくさん現れるが、真の固有値は全て含まれており、この方法の有効性をしめしている。

固有値問題に対するランチョス法について、歴史的な背景を踏まえて、最近の話題を取り上げた。現在までの解析で、ランチョス法の全ての問題点が解決されたわけではない。例えば、再直交化の時期、Ritzベクトルの計算の時期、即ちモニタリングの問題点、ランチョス・ベクトルの記憶容量など、いろいろ存在するのである。また、再直交化を全然行わないランチョス法でも、どこまで3重対角化すると求めたい固有値が本当に現れるという理論的な保証がないし、3重対角行列を大きくすればするほど、その固有値を計算する計算時間が膨大なものになってしまうのである。ようやく問題解決の緒の一手手前にたど

表5.7 $n=200$, T_j ($j=2n$) 固有値の数

6	6	5	5	4	0	4	0	4	4	4	3	0	3	0
3	3	3	3	3	3	3	2	0	2	0	2	0	2	2
0	2	0	2	0	2	2	2	2	2	2	2	2	2	2
2	2	1	0	1	0	1	0	1	0	1	0	1	0	1
0	1	0	1	1	0	1	0	1	0	1	0	1	1	0
1	0	1	0	1	1	1	0	1	1	0	1	0	1	1
1	1	1	1	0	1	0	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	0	1
0	1	1	1	1	1	1	0	1	0	1	0	1	1	1
1	0	1	0	1	0	1	0	1	0	1	1	0	1	0
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0
1	0	1	2	2	2	2	2	2	2	2	2	2	2	2
0	2	0	2	2	0	2	0	2	0	2	0	2	0	2
3	3	3	3	3	3	0	3	0	3	4	4	4	0	4
0	4	5	5	6	6									

表5.8 $n=200$, T_j ($j=3n$) の固有値の数

9	0	9	8	7	6	0	6	0	6	0	6	5	0	5
0	5	5	5	5	4	0	4	4	4	4	4	0	3	0
3	0	3	4	0	3	0	3	0	3	3	3	3	3	3
0	2	0	2	0	2	3	3	0	2	0	2	0	2	0
2	0	2	2	0	2	0	2	0	2	2	2	2	0	2
0	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2	2	2	2	0	1	1	0	1	0	1	0	1	0	1
0	1	2	0	1	1	0	1	0	1	1	0	1	1	0
1	0	1	0	1	0	1	0	1	0	1	0	1	1	1
1	1	1	0	1	1	1	0	1	1	1	0	1	1	1
0	1	1	1	1	1	1	0	1	1	0	1	0	1	0
1	0	1	0	1	0	1	0	1	0	1	1	1	0	1
0	1	0	1	0	1	0	1	0	1	0	1	1	0	1
0	1	0	1	2	2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	0	2	0	2	0	2	2	2	2	0
2	0	2	2	0	2	0	2	0	2	0	2	0	2	3
3	0	2	0	2	3	3	3	3	3	3	3	0	3	0
3	0	3	0	3	0	3	4	4	4	4	4	0	4	4
0	5	5	5	5	5	0	5	0	6	6	0	6	0	6
0	7	0	8	8	0	9								

表の中の数字は、固有値の重複を表しており、0 は偽の固有値の存在をしめしている。

りついたと言えるのかもしれない。しかし、ランチョス法は、現実の大型行列計算で重要な役割をはたしていることも事実である。

(参考文献)

- [1] C.Lanczos, An iteration method for the solution of the eigenvalue problem of linear differential and integral operators, J. Res. Nat. Bur. Standards 45: 255-282 (1950) .
- [2] B.N.Parlett, The Symmetric Eigenvalue Problem, Prentice-Hall (1980) .
- [3] J.K.Cullum&R.A.Willoughby, Lanczos Algorithm for Large Symmetric Eigenvalue Computations, I, II, Birkhauser (1985) .
- [4] J.K.Cullum&R.A.Willoughby, Computing Eigenvalues of Very Large Symmetric Matrices-An Implementation of a Lanczos Algorithm with No Reorthogonalization, J.Comp. Phys. 44, pp.329-358 (1981) .
- [5] C.C.Paige, The computation of eigenvalues and eigenvectors of very large sparse matrices, Ph.D. Thesis, Univ. of London (1971) .
- [6] C.C.Paige, Computational Variants of the Lanczos Method for the Eigenproblem, J. Inst. Math. Appl. 10, p.373-381 (1972) .
- [7] C.C.Paige, Error analysis of the Lanczos method for eigenproblem, J. Inst. Maths. Applics. 18, pp.341-349 (1976) .
- [8] B.N.Parlett &D.S.Scott, The Lanczos Algorithm with Selective Orthogonalization, Math. Comp. 33, pp.217-238 (1979) .
- [9] B.N.Parlett &J.K.Reid, Tracking the Progress of the Lanczos Algorithm for Large Symmetric Eigenproblems, IMA J. Num. Anal. 1, pp.135-155 (1981) .
- [10] H.D.Simon, The Lanczos Algorithm with Partial Reorthogonalization, Math. Comp. 42, pp.115-142 (1984) .
- [11] H.D.Simon, Analysis of the Symmetric Lanczos Algorithm with Reorthogonalization Methods, Linear Alg. and its Applications 61, pp.101-131 (1984) .
- [12] H.Takahasi&M.Natori, Eigenvalue Problem of Large Sparse Matrices, Rep. Comp. Center, University of Tokyo, 4, pp.129-148 (1972) .
- [13] 高橋, 野寺, 行列問題に対するLanczos 法と共役勾配法, 京都大学数理解析研究所講究録 373, pp.117-132 (1979) .
- [14] 高橋, 固有値問題のLanczos 法について, 京都大学数理解析研究所講究録422, pp.119-139 (1981) .
- [15] B.N.Parlett, Two Monitoring Schemes for the Lanczos Algorithm, Computing Methods in Applied Sciences and Engineering V, North-Holland, pp.27-34 (1982) .

表5.9 (その1) $n = 200$, T_j ($j = 2, n$) の全ての固有値の真偽の判定

K=	1	<ACCEPT>	0.1033524003207532	0.1033524003207539	1	K=	51	<REPEAT>	0.7791329205371569	0.7810110804020544	11
K=	2	<REPEAT>	0.1033524003207552	0.1033524003207553	1	K=	52	*REJECT*	0.8301916504791453	0.8301916504791448	0
K=	3	<REPEAT>	0.1033524003207557	0.1033524003207557	1	K=	53	<ACCEPT>	0.8340344490535444	0.8340344490535427	12
K=	4	<REPEAT>	0.1033524003207566	0.1033524003207564	1	K=	54	<REPEAT>	0.8340344490535448	0.8340344490535474	12
K=	5	<REPEAT>	0.1033524003207578	0.1033524003207588	1	K=	55	<REPEAT>	0.8340344490535478	0.8340344490535478	12
K=	6	<REPEAT>	0.1033524003207592	0.1034485470822909	1	K=	56	<ACCEPT>	0.8513891906779903	0.8513891906779903	13
K=	7	<ACCEPT>	0.1698684411987306	0.1698684411987306	1	K=	57	<REPEAT>	0.8513891906779943	0.8513891906779943	13
K=	8	<REPEAT>	0.1698684411987307	0.1698684411987307	1	K=	58	<REPEAT>	0.8513891906779944	0.8513891906779944	13
K=	9	<REPEAT>	0.1698684411987316	0.1698684411987314	1	K=	59	<ACCEPT>	0.8513891906779944	0.8513891906779944	13
K=	10	<REPEAT>	0.1698684411987321	0.1698684411987321	1	K=	60	<REPEAT>	0.8513891906779944	0.8513891906779944	13
K=	11	<REPEAT>	0.1698684411987325	0.1698684411987325	1	K=	61	<REPEAT>	0.8513891906779944	0.8513891906779944	13
K=	12	<REPEAT>	0.1698684411987330	0.1698684411987330	1	K=	62	<REPEAT>	0.8513891906779944	0.8513891906779944	13
K=	13	<ACCEPT>	0.1698684411987335	0.1698684411987335	1	K=	63	<REPEAT>	0.8513891906779944	0.8513891906779944	13
K=	14	<REPEAT>	0.1698684411987335	0.1698684411987335	1	K=	64	<REPEAT>	0.8513891906779944	0.8513891906779944	13
K=	15	<REPEAT>	0.1698684411987335	0.1698684411987335	1	K=	65	<ACCEPT>	0.8513891906779944	0.8513891906779944	13
K=	16	<REPEAT>	0.1698684411987335	0.1698684411987335	1	K=	66	<REPEAT>	0.8513891906779944	0.8513891906779944	13
K=	17	<REPEAT>	0.1698684411987335	0.1698684411987335	1	K=	67	<REPEAT>	0.8513891906779944	0.8513891906779944	13
K=	18	<ACCEPT>	0.1698684411987335	0.1698684411987335	1	K=	68	<ACCEPT>	0.8513891906779944	0.8513891906779944	13
K=	19	<REPEAT>	0.1698684411987335	0.1698684411987335	1	K=	69	<REPEAT>	0.8513891906779944	0.8513891906779944	13
K=	20	<REPEAT>	0.1698684411987335	0.1698684411987335	1	K=	70	<REPEAT>	0.8513891906779944	0.8513891906779944	13
K=	21	<REPEAT>	0.1698684411987335	0.1698684411987335	1	K=	71	<ACCEPT>	0.8513891906779944	0.8513891906779944	13
K=	22	<REPEAT>	0.1698684411987335	0.1698684411987335	1	K=	72	<REPEAT>	0.8513891906779944	0.8513891906779944	13
K=	23	<ACCEPT>	0.1698684411987335	0.1698684411987335	1	K=	73	<REPEAT>	0.8513891906779944	0.8513891906779944	13
K=	24	<REPEAT>	0.1698684411987335	0.1698684411987335	1	K=	74	<ACCEPT>	0.8513891906779944	0.8513891906779944	13
K=	25	<REPEAT>	0.1698684411987335	0.1698684411987335	1	K=	75	<REPEAT>	0.8513891906779944	0.8513891906779944	13
K=	26	<REPEAT>	0.1698684411987335	0.1698684411987335	1	K=	76	*REJECT*	0.8513891906779944	0.8513891906779944	13
K=	27	*REJECT*	0.4063473227653630	0.4063473227653630	1	K=	77	<ACCEPT>	0.8513891906779944	0.8513891906779944	13
K=	28	<ACCEPT>	0.4063473227653630	0.4063473227653630	1	K=	78	<REPEAT>	0.8513891906779944	0.8513891906779944	13
K=	29	<REPEAT>	0.4063473227653630	0.4063473227653630	1	K=	79	*REJECT*	0.8513891906779944	0.8513891906779944	13
K=	30	<REPEAT>	0.4063473227653630	0.4063473227653630	1	K=	80	<ACCEPT>	0.8513891906779944	0.8513891906779944	13
K=	31	<REPEAT>	0.4063473227653630	0.4063473227653630	1	K=	81	<REPEAT>	0.8513891906779944	0.8513891906779944	13
K=	32	*REJECT*	0.4063473227653630	0.4063473227653630	1	K=	82	*REJECT*	0.8513891906779944	0.8513891906779944	13
K=	33	<ACCEPT>	0.4063473227653630	0.4063473227653630	1	K=	83	<ACCEPT>	0.8513891906779944	0.8513891906779944	13
K=	34	<REPEAT>	0.4063473227653630	0.4063473227653630	1	K=	84	<REPEAT>	0.8513891906779944	0.8513891906779944	13
K=	35	<REPEAT>	0.4063473227653630	0.4063473227653630	1	K=	85	<ACCEPT>	0.8513891906779944	0.8513891906779944	13
K=	36	<REPEAT>	0.4063473227653630	0.4063473227653630	1	K=	86	<REPEAT>	0.8513891906779944	0.8513891906779944	13
K=	37	<ACCEPT>	0.4063473227653630	0.4063473227653630	1	K=	87	*REJECT*	0.8513891906779944	0.8513891906779944	13
K=	38	<REPEAT>	0.4063473227653630	0.4063473227653630	1	K=	88	<ACCEPT>	0.8513891906779944	0.8513891906779944	13
K=	39	<REPEAT>	0.4063473227653630	0.4063473227653630	1	K=	89	*REPEAT*	0.8513891906779944	0.8513891906779944	13
K=	40	<REPEAT>	0.4063473227653630	0.4063473227653630	1	K=	90	*REPEAT*	0.8513891906779944	0.8513891906779944	13
K=	41	<ACCEPT>	0.4063473227653630	0.4063473227653630	1	K=	91	<ACCEPT>	0.8513891906779944	0.8513891906779944	13
K=	42	<REPEAT>	0.4063473227653630	0.4063473227653630	1	K=	92	<REPEAT>	0.8513891906779944	0.8513891906779944	13
K=	43	<REPEAT>	0.4063473227653630	0.4063473227653630	1	K=	93	*REJECT*	0.8513891906779944	0.8513891906779944	13
K=	44	<REPEAT>	0.4063473227653630	0.4063473227653630	1	K=	94	<ACCEPT>	0.8513891906779944	0.8513891906779944	13
K=	45	<ACCEPT>	0.4063473227653630	0.4063473227653630	1	K=	95	<REPEAT>	0.8513891906779944	0.8513891906779944	13
K=	46	<REPEAT>	0.4063473227653630	0.4063473227653630	1	K=	96	<ACCEPT>	0.8513891906779944	0.8513891906779944	13
K=	47	<REPEAT>	0.4063473227653630	0.4063473227653630	1	K=	97	<REPEAT>	0.8513891906779944	0.8513891906779944	13
K=	48	*REJECT*	0.4063473227653630	0.4063473227653630	1	K=	98	<ACCEPT>	0.8513891906779944	0.8513891906779944	13
K=	49	<ACCEPT>	0.4063473227653630	0.4063473227653630	1	K=	99	<REPEAT>	0.8513891906779944	0.8513891906779944	13
K=	50	<REPEAT>	0.4063473227653630	0.4063473227653630	1	K=	100	<ACCEPT>	0.8513891906779944	0.8513891906779944	13

表5.9 (その2) $n = 200$, T_j ($j = 2, n$) の全ての固有値の真偽の判定

K= 101	REPEAT	1.703066330336584	1.723444602070535	29	N= 151	<ACCEPT>	2.818525932886421	2.8249225709538876	54
K= 102	<ACCEPT>	1.737708671003177	1.737708671003179	30	K= 152	<ACCEPT>	2.8397338715282281	2.842538482938221	57
K= 103	REPEAT	1.737708671003182	1.739393681939005	30	K= 153	*REJECT*	2.842538482938221	2.852667200957388	0
K= 104	<ACCEPT>	1.804224711881151	1.804224711881155	31	K= 154	<ACCEPT>	2.853168373553521	2.871639370442197	58
K= 105	REPEAT	1.804224711881155	1.806833991574625	31	K= 155	<ACCEPT>	2.919484414445195	2.919745618940958	59
K= 106	<ACCEPT>	1.872451066425012	1.872451066425016	32	K= 156	*REJECT*	2.919745618940959	2.921812549810270	0
K= 107	REPEAT	1.872451066425016	1.878106398006413	32	K= 157	<ACCEPT>	2.9844688274720642	2.987477063232054	60
K= 108	<ACCEPT>	1.913432587648595	1.913432587648607	33	K= 158	*REJECT*	2.987477063232054	2.995807212574642	0
K= 109	REPEAT	1.913432587648606	1.915155487229506	33	K= 159	<ACCEPT>	3.028892290198941	3.031709407678833	61
K= 110	<ACCEPT>	1.922190370278763	1.922190370278835	34	K= 160	<ACCEPT>	3.028892290198941	3.031709407678833	62
K= 111	REPEAT	1.922190370278835	1.929544545184507	34	K= 161	<ACCEPT>	3.0373750072829108	3.045654961753522	63
K= 112	<ACCEPT>	1.9315384598161	1.931538459816150	35	K= 162	<ACCEPT>	3.048174983070433	3.076979910812303	64
K= 113	REPEAT	1.9315384598161	1.938548545120591	35	K= 163	<ACCEPT>	3.081014053771010	3.093227135003115	65
K= 114	<ACCEPT>	1.959596483376643	1.959596483376647	36	K= 164	<ACCEPT>	3.135220400022061	3.137449454413345	66
K= 115	REPEAT	1.959596483376648	1.9836339747602635	36	K= 165	*REJECT*	3.137449454413345	3.154210098195927	67
K= 116	<ACCEPT>	2.042892774821442	2.042892774821448	37	K= 166	<ACCEPT>	3.178415407473560	3.178415407473560	68
K= 117	REPEAT	2.042892774821448	2.072731479454945	37	K= 167	*REJECT*	3.178415407473560	3.1967449834320	0
K= 118	<ACCEPT>	2.16803274764793	2.168032747901346	38	K= 168	<ACCEPT>	3.270228455548804	3.270228455548804	69
K= 119	*REJECT*	2.168032747901344	2.168384410404985	0	K= 169	<ACCEPT>	3.270228455548804	3.299832138489934	70
K= 120	<ACCEPT>	2.169169973996230	2.169169974873137	39	K= 170	<ACCEPT>	3.317733950911694	3.317733950911694	71
K= 121	*REJECT*	2.169169974873136	2.189441891914578	0	K= 171	<ACCEPT>	3.31830161169079	3.327561948703280	72
K= 122	<ACCEPT>	2.230474239943857	2.230474268185344	40	K= 172	<ACCEPT>	3.327993256488479	3.331911821842729	73
K= 123	*REJECT*	2.230474268185343	2.231461129921267	0	K= 173	<ACCEPT>	3.332059815440320	3.351424533360526	74
K= 124	<ACCEPT>	2.245236664196804	2.245236664196804	41	K= 174	<ACCEPT>	3.354726282344124	3.354726282344124	75
K= 125	*REJECT*	2.245236664196804	2.247370925456172	0	K= 175	<ACCEPT>	3.398575856318294	3.398575856318294	76
K= 126	<ACCEPT>	2.247370925456172	2.249273913106153	42	K= 176	<ACCEPT>	3.401427970774620	3.401427970774620	77
K= 127	*REJECT*	2.249273913106152	2.283731919134739	0	K= 177	<ACCEPT>	3.401427970774620	3.401427970774620	78
K= 128	<ACCEPT>	2.249273913106152	2.283731919134739	43	K= 178	<ACCEPT>	3.401427970774620	3.401427970774620	79
K= 129	*REJECT*	2.304968024096317	2.304968024096317	0	K= 179	<ACCEPT>	3.401427970774620	3.401427970774620	80
K= 130	<ACCEPT>	2.304968024096317	2.308352749499378	44	K= 180	<ACCEPT>	3.401427970774620	3.401427970774620	81
K= 131	*REJECT*	2.37348406673525	2.37348406673525	45	K= 181	<ACCEPT>	3.401427970774620	3.401427970774620	82
K= 132	<ACCEPT>	2.439487925234440	2.439487925234440	46	K= 182	<ACCEPT>	3.401427970774620	3.401427970774620	83
K= 133	*REJECT*	2.439487925234440	2.439487925234440	47	K= 183	<ACCEPT>	3.401427970774620	3.401427970774620	84
K= 134	<ACCEPT>	2.468953121510491	2.468953121510491	48	K= 184	<ACCEPT>	3.401427970774620	3.401427970774620	85
K= 135	*REJECT*	2.468953121510491	2.468953121510491	49	K= 185	<ACCEPT>	3.401427970774620	3.401427970774620	86
K= 136	<ACCEPT>	2.468953121510491	2.468953121510491	50	K= 186	<ACCEPT>	3.401427970774620	3.401427970774620	87
K= 137	*REJECT*	2.468953121510491	2.468953121510491	51	K= 187	<ACCEPT>	3.401427970774620	3.401427970774620	88
K= 138	<ACCEPT>	2.468953121510491	2.468953121510491	52	K= 188	<ACCEPT>	3.401427970774620	3.401427970774620	89
K= 139	*REJECT*	2.468953121510491	2.468953121510491	53	K= 189	<ACCEPT>	3.401427970774620	3.401427970774620	90
K= 140	<ACCEPT>	2.468953121510491	2.468953121510491	54	K= 190	<ACCEPT>	3.401427970774620	3.401427970774620	91
K= 141	*REJECT*	2.468953121510491	2.468953121510491	55	K= 191	<ACCEPT>	3.401427970774620	3.401427970774620	92
K= 142	<ACCEPT>	2.468953121510491	2.468953121510491	56	K= 192	<ACCEPT>	3.401427970774620	3.401427970774620	93
K= 143	*REJECT*	2.468953121510491	2.468953121510491	57	K= 193	<ACCEPT>	3.401427970774620	3.401427970774620	94
K= 144	<ACCEPT>	2.468953121510491	2.468953121510491	58	K= 194	<ACCEPT>	3.401427970774620	3.401427970774620	95
K= 145	*REJECT*	2.468953121510491	2.468953121510491	59	K= 195	<ACCEPT>	3.401427970774620	3.401427970774620	96
K= 146	<ACCEPT>	2.468953121510491	2.468953121510491	60	K= 196	<ACCEPT>	3.401427970774620	3.401427970774620	97
K= 147	*REJECT*	2.468953121510491	2.468953121510491	61	K= 197	<ACCEPT>	3.401427970774620	3.401427970774620	98
K= 148	<ACCEPT>	2.468953121510491	2.468953121510491	62	K= 198	<ACCEPT>	3.401427970774620	3.401427970774620	99
K= 149	*REJECT*	2.468953121510491	2.468953121510491	63	K= 199	<ACCEPT>	3.401427970774620	3.401427970774620	100
K= 150	<ACCEPT>	2.468953121510491	2.468953121510491	64	K= 200	<ACCEPT>	3.401427970774620	3.401427970774620	100

表5.9 (その3) $n = 200$, T_j ($j = 2, n$) の全ての固有値の真偽の判定

K= 201	<ACCEPT>	4.007840335656720	101	K= 251	<ACCEPT>	5.18830389846207	146
K= 202	<ACCEPT>	4.030029517030377	102	K= 252	<REJECT>	5.206457474158803	0
K= 203	<ACCEPT>	4.058675705212170	103	K= 253	<ACCEPT>	5.237465197749734	147
K= 204	<ACCEPT>	4.062741864173107	104	K= 254	<REJECT>	5.238178175449488	0
K= 205	<ACCEPT>	4.062741864173107	105	K= 255	<ACCEPT>	5.275871893916403	148
K= 206	<ACCEPT>	4.1107048211424164	106	K= 256	<REJECT>	5.278423948886548	0
K= 207	<ACCEPT>	4.119430670142482	107	K= 257	<ACCEPT>	5.284029676540572	149
K= 208	<ACCEPT>	4.13516948973755	108	K= 258	<REJECT>	5.367847861408313	0
K= 209	<ACCEPT>	4.156362275769088	109	K= 259	<ACCEPT>	5.367847861408313	150
K= 210	<ACCEPT>	4.160412191366063	110	K= 260	<REJECT>	5.459180567474556	0
K= 211	<ACCEPT>	4.169169973996232	111	K= 261	<ACCEPT>	5.459180567474556	151
K= 212	<ACCEPT>	4.21640322002713	112	K= 262	<ACCEPT>	5.47394407931369	152
K= 213	<ACCEPT>	4.228638545909924	113	K= 263	<REJECT>	5.47394407931369	0
K= 214	<ACCEPT>	4.266508398597010	114	K= 264	<ACCEPT>	5.53140928024039	153
K= 215	<ACCEPT>	4.295154586787900	115	K= 265	<REJECT>	5.53140928024039	0
K= 216	<ACCEPT>	4.309721467890573	116	K= 266	<ACCEPT>	5.53140928024039	154
K= 217	<ACCEPT>	4.342756080741423	117	K= 267	<ACCEPT>	5.53140928024039	155
K= 218	<ACCEPT>	4.385788158091147	118	K= 268	<REJECT>	5.53140928024039	0
K= 219	<ACCEPT>	4.416149577713698	119	K= 269	<ACCEPT>	5.561512074735664	156
K= 220	<ACCEPT>	4.434089863719422	120	K= 270	<REJECT>	5.626515924342679	0
K= 221	<ACCEPT>	4.435527461944902	121	K= 271	<ACCEPT>	5.626515924342679	157
K= 222	<ACCEPT>	4.446053371866232	122	K= 272	<REJECT>	5.693031975518129	0
K= 223	<ACCEPT>	4.452882203569347	123	K= 273	<ACCEPT>	5.693031975518129	158
K= 224	<ACCEPT>	4.472216267914271	124	K= 274	<REJECT>	5.750730776563395	0
K= 225	<ACCEPT>	4.479039416157783	125	K= 275	<ACCEPT>	5.750730776563395	159
K= 226	<ACCEPT>	4.601424143681715	126	K= 276	<REJECT>	5.754760732387539	0
K= 227	<ACCEPT>	4.635273717655883	127	K= 277	<ACCEPT>	5.754760732387539	160
K= 228	<ACCEPT>	4.667940184559691	128	K= 278	<REJECT>	5.769525705244525	0
K= 229	<ACCEPT>	4.672006343511532	129	K= 279	<ACCEPT>	5.769525705244525	161
K= 230	<ACCEPT>	4.681349833830927	130	K= 280	<REJECT>	5.830830023437294	0
K= 231	<ACCEPT>	4.68250706862366	131	K= 281	<ACCEPT>	5.830830023437294	162
K= 232	<ACCEPT>	4.715370323453433	132	K= 282	<REJECT>	5.831967211730810	0
K= 233	<ACCEPT>	4.729671544459201	133	K= 283	<ACCEPT>	5.831967211730810	163
K= 234	<REJECT>	4.8192344713215426	0	K= 284	<ACCEPT>	5.937107225178560	164
K= 235	<ACCEPT>	4.82164752628219	134	K= 285	<REJECT>	5.937107225178560	0
K= 236	<REJECT>	4.85622792266288	0	K= 286	<ACCEPT>	6.040403516623362	165
K= 237	<ACCEPT>	4.864679599779943	135	K= 287	<REJECT>	6.040403516623362	0
K= 238	<ACCEPT>	4.9189859473229998	136	K= 288	<ACCEPT>	6.068446134401841	166
K= 239	<ACCEPT>	4.951835016795575	137	K= 289	<REJECT>	6.068446134401841	0
K= 240	<ACCEPT>	4.962349927170899	138	K= 290	<ACCEPT>	6.077809629721239	167
K= 241	<ACCEPT>	4.971107709801068	139	K= 291	<REJECT>	6.077809629721239	0
K= 242	<ACCEPT>	4.980290213176624	140	K= 292	<ACCEPT>	6.08567412351407	168
K= 243	<REJECT>	5.007712290472531	0	K= 293	<REJECT>	6.08567412351407	169
K= 244	<ACCEPT>	5.015311732799363	141	K= 294	<ACCEPT>	6.127548933574990	170
K= 245	<REJECT>	5.074799807463591	0	K= 295	<REJECT>	6.127548933574990	171
K= 246	<ACCEPT>	5.080315585568511	142	K= 296	<ACCEPT>	6.195775288118951	172
K= 247	<REJECT>	5.107704735481298	0	K= 297	<REJECT>	6.195775288118951	0
K= 248	<ACCEPT>	5.146831626446486	143	K= 298	<ACCEPT>	6.262291328996826	173
K= 249	<ACCEPT>	5.16026128071723	144	K= 299	<REJECT>	6.262291328996826	0
K= 250	<ACCEPT>	5.181474067113084	145	K= 300	<ACCEPT>	6.296933769663424	174

表5.9 (その4) $n = 200$, T_j ($j = 2, n$) の全ての固有値の真偽の判定

N = 301	<ACCEPT>	6.29693769643426	6.304312122756477	172	N = 351	<REPEATE>	7.220867079462851	7.220867079462851	190
N = 302	<ACCEPT>	6.309721467890572	6.309721467890572	173	N = 352	<REPEATE>	7.220867079462852	7.220867079462852	190
N = 303	<REPEATE>	6.309721467890571	6.361882105172581	173	N = 353	<REJECT>	7.287383119840311	7.287383119840310	0
N = 304	<ACCEPT>	6.364027815141622	6.364027815141626	174	N = 354	<ACCEPT>	7.287383120340824	7.287383120340823	191
N = 305	<REPEATE>	6.364027815141624	6.402397026879762	174	N = 355	<REPEATE>	7.287383120340825	7.287383120340825	191
N = 306	<ACCEPT>	6.413189114395150	6.413189114395155	175	N = 356	<REPEATE>	7.287383120340826	7.287383120340826	191
N = 307	<REPEATE>	6.413189114395153	6.474455224835222	175	N = 357	<ACCEPT>	7.334984614294332	7.334984614294332	192
N = 308	<REJECT>	6.475604533761817	6.475604533761822	0	N = 358	<REPEATE>	7.334984614294349	7.334984614294346	192
N = 309	<REPEATE>	6.483307574635759	6.483307574635764	176	N = 359	<REPEATE>	7.334984614294350	7.334984614294350	192
N = 310	<REPEATE>	6.483307574635763	6.545149769370354	176	N = 360	<REPEATE>	7.334984614294352	7.334984614294352	192
N = 311	<REJECT>	6.556696712682805	6.556696712682808	0	N = 361	<ACCEPT>	7.334984614294355	7.334984614294355	193
N = 312	<ACCEPT>	6.556701071608033	6.556701071608037	177	N = 362	<REPEATE>	7.335089690888645	7.335089690888642	193
N = 313	<REPEATE>	6.556701071608025	6.622656776291432	177	N = 363	<REPEATE>	7.335089690888645	7.335089690888644	193
N = 314	<ACCEPT>	6.63276761808606	6.632767761808612	178	N = 364	<REPEATE>	7.335089690888648	7.335089690888645	193
N = 315	<REPEATE>	6.63276761808610	6.6386255184736983	178	N = 365	<ACCEPT>	7.484444801467197	7.484444801467197	194
N = 316	<REJECT>	6.6386255184736982	6.647624645273958	179	N = 366	<REPEATE>	7.484444801467198	7.484444801467197	194
N = 317	<ACCEPT>	6.649667995961782	6.649667995961785	179	N = 367	<REPEATE>	7.484444801467199	7.484444801467200	194
N = 318	<REPEATE>	6.649667995961784	6.67585235349940	179	N = 368	<REPEATE>	7.484444801467201	7.484444801467200	194
N = 319	<REJECT>	6.682090402758393	6.682090402758395	0	N = 369	<REJECT>	7.570300459079656	7.570300459079655	0
N = 320	<ACCEPT>	6.682507045662361	6.682507045662362	180	N = 370	<ACCEPT>	7.571463495860982	7.571463495860979	195
N = 321	<REPEATE>	6.682507045662361	6.739133004274491	180	N = 371	<REPEATE>	7.571463495860982	7.571463495860982	195
N = 322	<REJECT>	6.741973073601514	6.741973073601515	0	N = 372	<REPEATE>	7.571463495860984	7.571463495860984	195
N = 323	<ACCEPT>	6.741973637576051	6.741973637576054	181	N = 373	<REPEATE>	7.571463495860985	7.571463495860983	195
N = 324	<REPEATE>	6.741973637576054	6.767549191579793	181	N = 374	<REJECT>	7.592693592722018	7.592693592722019	0
N = 325	<REJECT>	6.77580592757058	6.77580592757059	0	N = 375	<ACCEPT>	7.5936522677234641	7.5936522677234639	196
N = 326	<ACCEPT>	6.775822111550219	6.775822111550222	182	N = 376	<REPEATE>	7.5936522677234641	7.5936522677234640	196
N = 327	<REPEATE>	6.775822111550222	6.808974738031140	182	N = 377	<REPEATE>	7.5936522677234642	7.5936522677234642	196
N = 328	<REJECT>	6.808491608481261	6.808491608481262	0	N = 378	<REPEATE>	7.5936522677234642	7.5936522677234642	196
N = 329	<ACCEPT>	6.808491678454026	6.808491678454029	183	N = 379	<ACCEPT>	7.660168718112616	7.660168718112616	197
N = 330	<REPEATE>	6.808491678454029	6.914128667840421	183	N = 380	<REPEATE>	7.660168718112617	7.660168718112615	197
N = 331	<ACCEPT>	6.918985947227594	6.918985947227595	184	N = 381	<REPEATE>	7.660168718112618	7.660168718112618	197
N = 332	<REPEATE>	6.918985947228992	6.918985947228993	184	N = 382	<REPEATE>	7.660168718112619	7.660168718112618	197
N = 333	<REPEATE>	6.918985947228993	6.924200494956754	184	N = 383	<REPEATE>	7.660168718112619	7.660168718112619	197
N = 334	<ACCEPT>	6.92486669379802	6.92486669379804	185	N = 384	<ACCEPT>	7.7209233683033829	7.7209233683033827	198
N = 335	<REPEATE>	6.92486669379827	6.92486669379829	185	N = 385	<REPEATE>	7.7209233683033830	7.7209233683033829	198
N = 336	<REPEATE>	6.92486669379829	6.956897231892486	185	N = 386	<REPEATE>	7.7209233683033830	7.7209233683033829	198
N = 337	<ACCEPT>	6.962199016522558	6.962199016522558	186	N = 387	<REPEATE>	7.7209233683033831	7.7209233683033832	198
N = 338	<REPEATE>	6.962199016522560	6.962199016522562	186	N = 388	<REPEATE>	7.7209233683033832	7.7209233683033832	198
N = 339	<REPEATE>	6.962199016522560	6.962199016522560	186	N = 389	<ACCEPT>	7.830131558801272	7.830131558801270	199
N = 340	<ACCEPT>	7.111659203695405	7.111659203695404	187	N = 390	<REPEATE>	7.830131558801273	7.830131558801271	199
N = 341	<REPEATE>	7.111659203695407	7.111659203695407	187	N = 391	<REPEATE>	7.830131558801273	7.830131558801271	199
N = 342	<REPEATE>	7.111659203695409	7.143306167119405	187	N = 392	<REPEATE>	7.830131558801274	7.830131558801272	199
N = 343	<ACCEPT>	7.148610809322012	7.148610809322012	188	N = 393	<REPEATE>	7.830131558801275	7.830131558801273	199
N = 344	<REPEATE>	7.148610809322015	7.148610809322014	188	N = 394	<REPEATE>	7.830131558801275	7.830131558801275	199
N = 345	<REPEATE>	7.148610809322015	7.164532887837289	188	N = 395	<ACCEPT>	7.896647599679248	7.896647599679246	200
N = 346	<ACCEPT>	7.165965550946458	7.165965550946457	189	N = 396	<REPEATE>	7.896647599679249	7.896647599679246	200
N = 347	<REPEATE>	7.165965550946462	7.165965550946461	189	N = 397	<REPEATE>	7.896647599679249	7.896647599679246	200
N = 348	<REPEATE>	7.165965550946462	7.218988915979742	189	N = 398	<REPEATE>	7.896647599679250	7.896647599679249	200
N = 349	<REJECT>	7.220767800285423	7.220767800285423	0	N = 399	<REPEATE>	7.896647599679250	7.896647599679250	200
N = 350	<ACCEPT>	7.220867079462850	7.220867079462847	190	N = 400	<REPEATE>	7.896647599679250	0.000000000000000	200