

実対称三重対角固有値問題に対する 多分割の分割統治法の改良

桑島 豊* 重原 孝臣*

* 埼玉大学理工学研究科

An Improvement of Multiple Division Divide-and-Conquer for Real Symmetric Tridiagonal Eigenproblem

Yutaka Kuwajima* Takaomi Shigehara*

* Graduate School of Science and Engineering, Saitama University

Abstract. We improve divide-and-conquer with multiple divisions for real symmetric tridiagonal eigenproblem proposed in [13]. The main improvements are the following two. The first is that we succeed in developing an algorithm for keeping the orthogonality among eigenvectors in double-precision floating-point number processing without a substantial increase of numerical cost. As a result, all the calculations are done without quadruple-precision floating-point number processing, which is required in [13]. The second is that we implement a deflation effect which substantially decreases numerical cost. As a result of these two improvements, we succeed in developing a program for real symmetric tridiagonal eigenproblem which is even faster than a LAPACK routine DSTEVD, while keeping a numerical accuracy comparable to DSTEVD.

1. 序論

実対称固有値問題のための分割統治法は、Multiple Relatively Robust Representations 法 [7] などと並んで、現在最も高速な解法のひとつである。

2 分割の分割統治法は、1981 年に発表された [5]。当時は、核となっている、対角行列と階数 1 の摂動の和の固有ベクトルを高精度に求めることができず、実用的ではなかった [4]。その後、Löwner の定理 [15] により、摂動の成分を再定義することで固有ベクトルの直交性を保証する方法が提案され [11], [12]、実用に耐えうるアルゴリズムとなった。現在は、数値線形代数のためのライブラリ LAPACK [1], [18] に採用され、広く利用されている。

昨年我々は、分割統治法を従来の 2 分割から k 分割 ($k > 2$) に拡張したアルゴリズムを提案した [13]。このアルゴリズムは、分割数 k が行列の次元より十分小さければ、2 分割と比較して、演算量が $3k/(2(k^2 - 1))$ に削減される。

あわせて、 k 分割の分割統治法の実装法を提案した。実装面では、精度に直結するため、対角行列と階数 $k - 1$ の摂動の和の固有値問題の解法アルゴリズムが最も重要である。固

有値は、対角行列と階数 1 の摂動の和の固有値問題を逐次的に $k-1$ 回解くことで求めることができる。また、逐次的に解く固有値問題の順番を変更することで演算量を削減できることを示した。固有ベクトルは、特性方程式を変形することで構成した特異行列の、固有値 0 に対応する固有ベクトルを用いることで、効率よく求める方法 [2] を採用した。しかし、その方法では倍精度演算のみでは固有ベクトルの直交性を得ることができないという問題点があった。その解決法として、4 倍精度演算を用いることで高精度に計算できること、またその適切な使用箇所を示した。

本論文では、これまでの実装の改良として、拡張精度演算を用いずに、倍精度演算のみで k 分割の分割統治法を実装する方法を提案する。提案手法では、固有ベクトル計算後に、直交していない固有ベクトルを再直交化することで精度を得る方法をとる。そのために、固有ベクトル同士の直交性を、事後に少ない演算量で求める方法も示す。また、再直交化前の固有ベクトルの計算の方法を改善し、精度を高める方法も提案する。これにより、再直交化すべき固有ベクトルを少なくすることができる。

速度面においても、我々の実装方法は問題があった。2 分割の分割統治法では、自明な固有値・固有ベクトルの計算を省く deflation という手法が用いられる [4], [8]。この手法により、演算量の最も多い部分である行列行列積から自明な演算を省くことができ、平均で $O(n^{2.3})$ に演算量を削減することができる [6], [20]。しかし、今まで我々はこの効果を k 分割の分割統治法に導入していなかった。本論文では、deflation を多分割に拡張し、演算量減少の効果を取り入れる。これにより、2 分割と同等以上の速度を実現することができる。

本論文は、以下のように構成される。2 節では、従来の 2 分割の分割統治法を概説する。3 節では、2 分割の分割統治法を自然に拡張した、我々の k 分割の分割統治法について説明する。4 節、5 節が本論文の主張で、今まで拡張精度演算を用いて精度を確保していた固有ベクトルの計算を倍精度のみで行う方法を提案する。4 節では、固有ベクトル計算後の再直交化の方法について、5 節では、再直交化前の固有ベクトルの精度向上の方法について述べる。6 節では、deflation の拡張をする。7 節では、 k 分割の分割統治法のアルゴリズムのまとめをする。8 節では、数値実験結果により k 分割の分割統治法の有効性を検証する。9 節はまとめである。

2. 2 分割の分割統治法の概説

この節では、 n 次実対称三重対角行列 T の固有値問題 $T = Q\Lambda Q^T$ を解くための、Cuppen の 2 分割の分割統治法について概説する。

T は対角成分が a_i ($i = 1, 2, \dots, n$)、副対角成分が b_i ($i = 1, 2, \dots, n-1$) であるとする。ただし、ある b_i が 0 のとき T は可約のため、一般性を失うことなく $b_i \neq 0$ を仮定する。

まず、 n 次実対称三重対角行列 T を、 n_1 次実対称三重対角行列 T_1 と n_2 次実対称三重

対角行列 T_2 ($n_1 + n_2 = n$), 非零実数 $c = |b_{n_1}|$ を用いて,

$$(2.1) \quad T = \begin{pmatrix} T_1 & \\ & T_2 \end{pmatrix} + c \mathbf{v} \mathbf{v}^T, \quad \mathbf{v} = \begin{pmatrix} \mathbf{e}_{n_1}^{(n_1)} \\ s \mathbf{e}_1^{(n_2)} \end{pmatrix}, \quad s = \begin{cases} 1 & (b_{n_1} > 0) \\ -1 & (b_{n_1} < 0) \end{cases}$$

と分割する. ただし, $\mathbf{e}_m^{(r)}$ は r 次元単位行列 I_r の第 m 列ベクトルとする. 次に, T_1, T_2 の固有値問題 $T_1 = Q_1 \Lambda_1 Q_1^T, T_2 = Q_2 \Lambda_2 Q_2^T$ を解くことで, T を

$$(2.2) \quad T = \begin{pmatrix} Q_1 & \\ & Q_2 \end{pmatrix} [D + c \mathbf{u} \mathbf{u}^T] \begin{pmatrix} Q_1 & \\ & Q_2 \end{pmatrix}^T, \quad D = \begin{pmatrix} \Lambda_1 & \\ & \Lambda_2 \end{pmatrix}$$

のように相似変換する. ここで, D は n 次実対角行列である. ただし,

$$(2.3) \quad \mathbf{u} = \begin{pmatrix} Q_1 & \\ & Q_2 \end{pmatrix}^T \mathbf{v} = \begin{pmatrix} Q_1^T \text{ の最終列} \\ s \times (Q_2^T \text{ の第 1 列}) \end{pmatrix}$$

次に, n 次実対角行列と階数 1 の摂動の和 $\tilde{D} = D + c \mathbf{u} \mathbf{u}^T$ の固有値問題 $\tilde{D} = \tilde{Q} \Lambda \tilde{Q}^T$ を解く. 最後に Q_1 と Q_2 の直和と \tilde{Q} の積を計算することで T の固有ベクトルが求められる. T_i は実対称三重対角行列であるので, これらの固有値問題に, 再帰的にこのアルゴリズムを適用することで, T の固有値問題を解くことができる.

実対角行列と階数 1 の摂動の和 \tilde{D} の固有値を求めるためには, 特性方程式 $\det(\lambda I_n - \tilde{D}) = 0$ の解を求めればよいが, $\lambda I_n - D$ が正則ならば, \tilde{D} の特性多項式は次のように変形できる [4], [9].

$$\begin{aligned} \det(\lambda I_n - \tilde{D}) &= \det(\lambda I_n - D) \det(I_n - c(\lambda I_n - D)^{-1} \mathbf{u} \mathbf{u}^T) \\ &= \det(\lambda I_n - D) (1 - c \mathbf{u}^T (\lambda I_n - D)^{-1} \mathbf{u}) \end{aligned}$$

よって, \tilde{D} の $\det(\lambda I_n - D) \neq 0$ を満たす固有値 λ は $f(\lambda) \equiv 1 - c \mathbf{u}^T (\lambda I_n - D)^{-1} \mathbf{u} = 0$ を解くことで求められる. 非線形方程式 $f(\lambda) = 0$ の解は, 変形した Newton 法 [10], [14], [17] を用いることで求めることができる.

一方, \tilde{D} の固有ベクトル \mathbf{q} は, 対応する固有値 λ が既知のとき,

$$(2.4) \quad \mathbf{q} = (\lambda I_n - D)^{-1} \mathbf{u}$$

で求められる.

しかし, この方法を陽に用いたのでは固有ベクトル同士の直交性が悪い. そのため, 求めた全固有値が「正確に」 $D + \tilde{\mathbf{u}} \tilde{\mathbf{u}}^T$ の固有値となるように, Löwner の定理 [15] を用いて, $\tilde{\mathbf{u}} = (\tilde{u}_1, \dots, \tilde{u}_n)^T$ を

$$(2.5) \quad |\tilde{u}_j| = \sqrt{(\lambda_j - d_j) \prod_{\substack{k=1 \\ k \neq j}}^n \frac{\lambda_k - d_j}{d_k - d_j}}, \quad (j = 1, 2, \dots, n)$$

と再定義し, それを改めて \mathbf{u} とし, (2.4) に代入することで固有ベクトルを求める.

$\mathbf{u} = (u_1, u_2, \dots, u_n)^T$ が要素に 0 を持つときや, $D = \text{diag}(d_1, d_2, \dots, d_n)$ の対角要素に重複があるときは, 自明な固有値や固有ベクトルを持つ. \mathbf{u} の第 j 要素 u_j が 0 のときは, $D + c\mathbf{u}\mathbf{u}^T$ は固有値 d_j を持ち, 対応する固有ベクトルは $\mathbf{e}_j^{(n)}$ である. また, $d_j = d_{j+1}$ のときは, \mathbf{u} に左からかけることで, 第 j 要素を 0, 第 $j+1$ 要素を $\sqrt{u_j^2 + u_{j+1}^2}$ に写す n 次 Givens 回転により, $u_j = 0$ の場合に帰着することができる.

これらの操作を deflation という. 対角行列と階数 1 の摂動の和の固有値問題の次元を下げ, さらに自明な演算を省くことで, 分割統治法で最後に行う行列積の演算量を少なくすることができる.

n 次実対角行列と階数 1 の摂動の和の固有値問題は平均で $38n^2$ の演算量で解くことができ [13], $n_1 = \lfloor n/2 \rfloor$ としたとき, 2 分割の分割統治法の演算量は, deflation が発生しなければ,

$$(2.6) \quad t_2(n) = 2t_2(n/2) + n^3 + 38n^2 = \left(1 + \frac{1}{4} + \frac{1}{4^2} + \dots\right) n^3 + \left(1 + \frac{1}{2} + \frac{1}{2^2} + \dots\right) 38n^2 \\ \simeq \frac{4}{3}n^3 + 2 \cdot 38n^2$$

となる.

3. k 分割の分割統治法の概説

この節では, これまでに我々が提案した, n 次実対称三重対角行列 T の固有値問題を解くための, k 分割の分割統治法について概説する [13].

まず, n 次実対称三重対角行列 T を, n_j 次実対称三重対角行列 T_j ($j = 1, 2, \dots, k; n_1 + n_2 + \dots + n_k = n$), 非零実数 $c_j = |b_{n_j}|$ ($j = 1, 2, \dots, k-1$) を用いて,

$$(3.1) \quad T = T_1 \oplus \dots \oplus T_k + \sum_{l=1}^{k-1} c_l \mathbf{v}_l \mathbf{v}_l^T, \quad \mathbf{v}_j = \mathbf{e}_{m_j}^{(n)} + s_j \mathbf{e}_{m_j+1}^{(n)}, \quad m_j = \sum_{l=1}^j n_l, \quad s_j = \frac{b_{n_j}}{|b_{n_j}|}$$

と分割する. ただし, $\mathbf{e}_m^{(n)}$ は n 次単位行列 I_n の第 m 列ベクトルとする. 次に, T_j の固有値問題 $T_j = Q_j \Lambda_j Q_j^T$ ($j = 1, 2, \dots, k$) を解くことで, T を

$$(3.2) \quad T = P \left[D + \sum_{l=1}^{k-1} c_l \mathbf{u}_l \mathbf{u}_l^T \right] P^T, \quad D = \Lambda_1 \oplus \dots \oplus \Lambda_k, \quad P = Q_1 \oplus \dots \oplus Q_k$$

のように相似変換する. ここで, D は n 次実対角行列である. ただし,

$$(3.3) \quad \mathbf{u}_j = P^T \mathbf{v}_j = \begin{pmatrix} \mathbf{0} \\ Q_j^T \text{ の最終列} \\ s_j \times (Q_{j+1}^T \text{ の第 1 列}) \\ \mathbf{0} \end{pmatrix}$$

である. 次に, n 次実対角行列と階数 $k-1$ の摂動の和 $\tilde{D} = D + c_1 \mathbf{u}_1 \mathbf{u}_1^T + \cdots + c_{k-1} \mathbf{u}_{k-1} \mathbf{u}_{k-1}^T$ の固有値問題 $\tilde{D} = \tilde{Q} \Lambda \tilde{Q}^T$ を解く. 最後に P と \tilde{Q} の積を計算することで T の固有ベクトルが求められる. T_i は実対称三重対角行列であるので, 再帰的にこのアルゴリズムを適用することで, T の固有値問題を解くことができる.

ここで, 表記を簡潔にするために, \mathbf{u}_i を並べた $n \times (k-1)$ 実行列を $U \equiv (\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_{k-1})$, c_i を対角要素に持つ $k-1$ 次実対角行列を $C \equiv \text{diag}(c_1, c_2, \dots, c_{k-1})$ と定義する. 以降はこれらの表記を適宜使用する.

実装においては, n_1, n_2, \dots, n_k をできるだけ等しく分割する. n を k 等分できない場合でも, 最大の値と最小の値との差を 1 にすることができる. そのため, 等分割が可能であると仮定し, 以下では $n_1 = n_2 = \cdots = n_k = n/k$ とする. この仮定においても, 以降のいかなる演算量評価も優位な差は生じない.

n 次実対角行列と階数 $k-1$ の摂動の和 $\tilde{D} = D + c_1 \mathbf{u}_1 \mathbf{u}_1^T + \cdots + c_{k-1} \mathbf{u}_{k-1} \mathbf{u}_{k-1}^T$ の固有値問題は, 逐次的に対角行列と階数 1 の摂動の和の固有値問題を $k-1$ 回解くことで求められる. ただし, \mathbf{u}_j の $(1-2/k)n$ 個の要素が 0 であることを利用し, 添え字の順に逐次的に解くのではなく, 和を取る順番を変えることで, 2 分割の分割統治法の固有値計算部分と同じ手順, 同じ演算量で固有値を求めることができる. 詳しくは, 前論文 [13] を参照のこと.

一方, 固有ベクトルは次の定理を用いて求めることができる.

定理 1. D を n 次実対角行列, U を $n \times k$ 次行列, C を k 次正則実対角行列とする. n 次実対称行列 $\tilde{D} = D + UCU^T$ の固有値が λ であるとき, $\lambda I_n - D$ が正則ならば, k 次実対称行列 $F(\lambda) = C^{-1} - U^T(\lambda I_n - D)^{-1}U$ は固有値 0 を持つ. またこのとき, F の固有値 0 に対応する固有ベクトル \mathbf{x} を用いて, λ に対応する \tilde{D} の固有ベクトルは $(\lambda I_n - D)^{-1}U\mathbf{x}$ と表せる.

証明 前論文 [13] の定理 3 を参照のこと. ■

n 次行列 T に対して k 分割の分割統治法を適用したときの演算量 $t_k(n)$ は以下のようになる. まず, k 回の再帰呼出しで $kt_k(n/k)$ の演算量である. 次に, 対角行列と階数 $k-1$ の摂動の和の固有値問題の演算量は, $O(k^2n^2 + k^3n)$ である. これは, k の依存性を最大限に見積もったものであり, 後述する実装法ではより小さい. 最後に T の全ての固有ベクトルを求めるための行列行列積は, 被乗行列の要素の $1/k$ が非零のため, その演算量は $(2/k)n^3$ である. したがって, $t_k(n) = kt_k(n/k) + O(k^2n^2 + k^3n) + (2/k)n^3$ となる. ここで, $k \ll n$ を仮定し, 演算量が $O(k^2n^2 + k^3n)$ の項を無視すると,

$$(3.4) \quad t_k(n) = kt_k(n/k) + \frac{2}{k}n^3 = \frac{2}{k} \left(1 + \frac{1}{k^2} + \frac{1}{k^4} + \cdots \right) n^3 \simeq \frac{2k}{k^2 - 1} n^3$$

を得る. これは, 2 分割の分割統治法の $3k/(2(k^2 - 1))$ の演算量である. より詳しくは, 7 節の最後で評価する.

4. 倍精度演算のみによる再直交化法の提案

これまで我々は、対角行列と低階数摂動の和の固有ベクトルの計算には拡張精度演算を用いてきた。それは、倍精度演算のみで計算した場合、固有ベクトルの精度が不十分で、直交誤差、残差ともに要求する閾値を下回らなかったからである。この問題に対する解決法として、本節では固有ベクトル計算後に再直交化を行うことで、直交性を確保する方法を提案する。ここで、2本の正規固有ベクトルが直交しているとは、それらの自然内積の絶対値が閾値（我々の実装では 10^{-14} とした）以下であることとする。

4.1 再直交化のアルゴリズム

まず、効率的な再直交化のために、求められた固有ベクトルを以下の2条件の下で最も細かいグループに分割する：

1. 異なるグループに属するベクトルは直交する。
2. 2本以上のベクトルを持つグループに属するベクトル v は、 v と直交しない（かつ、 v とは異なる）ベクトルを同一グループ内に持つ。

ただし、1グループ1ベクトルも許すこととする。すなわち、再直交化はそれぞれ1ベクトルのみからなる n グループに分割できるようにすることと言える。

このグループ分けは、キューを用いた次のアルゴリズムで達成できる（ただし、キューを用いることは本質ではない）。

アルゴリズム 1 `proc group_vectors(v_1, \dots, v_m)` m 本のベクトル v_1, \dots, v_m を入力として、上記の2条件を満たすように、それぞれのベクトルが属するグループを出力する

キューを空にする。

for $i := 1$ to m

if v_i がグループに属していない

v_i だけのグループをつくる

キューに v_i を入れる。

while キューが空でない

キューから取り出してそれを x とする。

for $j := i + 1$ to m

if v_j がグループに属していない

if v_j と x が直交していない

v_j を v_i と同じグループに入れる

```

        キューに  $v_j$  を入れる.
    end if
end if
end for
end while
end if
end for

```

アルゴリズム 1 では、相異なるベクトルの内積をほとんど全て計算することとなり、入力されたベクトルが m 本のとき、最大 $m(m-1)/2$ 回の内積計算が行われる。ただし、必要な内積計算の全てを通常の方法で行う必要はなく、多くの場合は、後述する手法 (4.3) に従って、内積 1 組を $O(1)$ の演算量で過大評価をすることで、直交していると判定できる。実際に通常の方法で内積計算を行うのは、過大評価で直交していないと判定された場合のみで、その割合は少ない。

次に、このアルゴリズムで分けられたそれぞれのグループ内で再直交化を行う。以下では、あるグループ G を固定して考える。グループの構成法から、 G に属するそれぞれのベクトルは他のグループに属するベクトルとは直交している。したがって、 G に属する l 本全てのベクトルで張られる空間が、 $D + UCU^T$ の l 次元不変部分空間に一致することが期待される。再直交化の方法は後述のアルゴリズム 2 で示す。

グループの構成法から G に属する l 本全てのベクトルで張られる空間が、 $D + UCU^T$ の不変部分空間に一致することが期待されると述べたが、数学的には正確ではない。そのため、事後に評価することを考える。 G に属する全てのベクトルで張られる空間の正規直交基底を作り、それらを列に並べた列直交行列を $M = (\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_l)$ とする。このとき、 $\text{span}\{\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_l\}$ が $D + UCU^T$ の不変部分空間であれば、任意の $j = 1, 2, \dots, l$ に対して

$$1 - \frac{\|M^T(D + UCU^T)\mathbf{m}_j\|_2^2}{\|(D + UCU^T)\mathbf{m}_j\|_2^2}$$

が 0 であることを利用して、これらの値が閾値（実装では 10^{-14} ）以下なら、 G に属する全てのベクトルで張られる空間が $D + UCU^T$ の不変部分空間であると、数値的に判定する。この計算内でノルム計算をするベクトルは、後述のアルゴリズム 2 で必要となる $M^T(D + UCU^T)M$ の値とその計算過程で得られるため、実質的に演算量の増加は殆どない。不変部分空間ではないと判定されたグループが存在する場合は、それらのグループに属するベクトルを全てまとめ、1 グループに集約する。このグループに属するベクトルが張る空間は、 $D + UCU^T$ の不変部分空間である。なお、 G が張る空間が不変部分空間ではないと判定されるのは非常に稀であり、著者が行った数値実験では発生しなかった。その数値実験の際には、成分が乱数の実対称三重対角行列や、[16] から無作為に選んだ 30 種の実対称行列を三重対角化した行列を与えた。

以上の方法により，再直交化すべきグループが確定する．それぞれのグループに属するベクトルの再直交化には，以下のアルゴリズムを用いる．

アルゴリズム 2 `proc reorthogonalize(G, D, U, C)` グループの 1 つ G を入力とし， G に属する全ベクトルを，それらが張る空間で $D + UCU^T$ の固有値問題を解くことで再直交化する．

G に属する全てのベクトルで張られる空間の正規直交基底を作り，それらを列に並べた列直交行列を M とする．

$A := M^T(D + UCU^T)M$ を構成する．

A の固有値問題 $A = Q_A \Lambda_A Q_A^T$ を解く．

MQ_A を計算し，その列ベクトルで G に属するベクトルを置き換える．

アルゴリズム 2 の演算量は以下の通り． G に属するベクトルの本数を l とする．また， U の各列ベクトル \mathbf{u}_i は，(3.3) の形式とする．修正 Gram-Schmidt の直交化法を用いると， M は $2l^2n$ の演算量で求められる． D, C が実対角行列であることと U の形式を利用すると， A は $(n+k)l(l+1) + 4ln$ の演算量で求められる． l 次実対称行列 A の固有値問題は，QR 法で $t_{QR}(l)$ の演算量で解けるとする．行列積 MQ_A は $2l^2n$ の演算量で求められる．よって，アルゴリズム 2 の演算量はこれらの和で，

$$(4.1) \quad 5l(l+1)n + kl(l+1) + t_{QR}(l)$$

となる．

$D + UCU^T$ の固有ベクトルを求めた後に，アルゴリズム 1 によってそれらを適切なグループに分割し，それぞれのグループに対してアルゴリズム 2 を適用することで $D + UCU^T$ の固有ベクトルの再直交化が達成できる．

4.2 直交性の事後評価

アルゴリズム 1 で固有ベクトルをグループに分割する際， n 本の固有ベクトル全ての直交性の情報が必要になる． n 本の n 次ベクトルの互いの内積を得るためには，通常 $O(n^3)$ 回の演算が必要になる．しかし，次の定理を用いると，対角行列と階数 k の摂動の和の n 次固有ベクトル同士の内積の計算が $O(n^2k)$ の演算量で完了する．

定理 2. D を n 次実対角行列， U を $n \times k$ 行列， C を k 次正則行列とする． λ_1, λ_2 は相異なる実数で， $\det(\lambda_i I_n - D) \neq 0$ を満たすとする． \mathbf{x}_i を k 次実対称行列 $F_i = C^{-1} - U^T(\lambda_i I_n - D)^{-1}U$ の固有ベクトルとし，対応する固有値を μ_i とする．

$$\mathbf{q}_i = \alpha_i (\lambda_i I_n - D)^{-1} U \mathbf{x}_i, \quad \alpha_i = \|(\lambda_i I_n - D)^{-1} U \mathbf{x}_i\|_2^{-1}$$

とするとき,

$$(4.2) \quad \mathbf{q}_1^T \mathbf{q}_2 = \alpha_1 \alpha_2 \frac{\mu_1 - \mu_2}{\lambda_1 - \lambda_2} \mathbf{x}_1^T \mathbf{x}_2$$

が成り立つ.

証明 付録参照のこと. ■

この定理の \mathbf{q}_i は, 定理 1 に従って, 誤差を含んだ可能性のある固有値 λ_i を用いて計算した $D + UCU^T$ の正規固有ベクトルと等しい. また, λ_i, α_i は今までに計算済みの値であり, μ_i は \mathbf{x}_i を求めるときに同時に取得可能であるため, これらの値は既知としてよい. よって定理 2 を用いると, 定理 1 を用いた $D + UCU^T$ の固有ベクトル計算後に, それらの内積は 1 組が $O(k)$ の演算量で, 固有ベクトル n 本全ての互いの内積は $O(n^2 k)$ で計算できる.

なお, \mathbf{x}_i が正規化されている場合, $|\mathbf{x}_1^T \mathbf{x}_2| \leq 1$ の関係から, (4.2) の代わりに

$$(4.3) \quad |\mathbf{q}_1^T \mathbf{q}_2| \leq \alpha_1 \alpha_2 \left| \frac{\mu_1 - \mu_2}{\lambda_1 - \lambda_2} \right|$$

を用いて直交性を判定することができる. この場合, 内積 1 組を $O(1)$ の演算量で, 固有ベクトル n 本全ての互いの内積を $O(n^2)$ の演算量で判定できるが, 厳密な判定ではなく, 直交している組を直交していないと判定する可能性がある (ただし, 逆はない). そのため, 直交していないと判定された場合, 以降の処理に 2 つの選択肢がある. 一つは, 厳密に直交性を判定しなおす方法, もう一つは, 直交していないものとみなす方法である. このどちらでも再直交化は正しく達成されるが, 我々は数値実験から各々のグループの大きさをできるだけ小さくしたほうが高速であると判断し, 前者で実装した.

5. 対角行列と低階数摂動の和の固有ベクトル計算法の改善

前節のアルゴリズム 1 と 2 を用いれば, 固有ベクトルの直交性を確保することができる. しかし, 再直交化前の固有ベクトルの直交性が悪い場合, アルゴリズム 1 で分割するグループが大きくなる. それにより, アルゴリズム 2 で解くべき固有値問題の次元が大きくなり, 必要な演算量が増大してしまう.

この節では, 得られた固有値から固有ベクトルを計算する方法を改良し, 再直交化前の固有ベクトルの精度を改善する方法について述べる.

5.1 固有ベクトル計算法の改良

対角行列と階数 k の摂動の和 $D + UCU^T$ の固有ベクトルは, 定理 1 を用いて求めることができる. この定理は, 無限精度演算の下では正確な固有ベクトルを出力するが, 有限精度演算では, 精度面で以下の 3 つの問題点がある.

第一に、汎用の固有値計算アルゴリズムで $F(\lambda)$ の固有値問題を解く際、固有値、固有ベクトルの精度は絶対評価（後述の (5.3)）でのみ保障されているため、 $F(\lambda)$ の条件数が大きい場合、絶対値最小の固有値 μ と対応する固有ベクトル \mathbf{x} は高精度に計算できない点。 \mathbf{x} の精度が不十分のとき、この \mathbf{x} を用いた $(\lambda I_n - D)^{-1}U\mathbf{x}$ は、 $D + UCU^T$ の固有ベクトルとしては精度が得られない。第二に、 $D + UCU^T$ の固有値 λ の誤差が、本来 0 となるべき $F(\lambda)$ の絶対値最小の固有値 μ に拡大されて伝わる可能性がある点。 $|\mu|$ が大きいとき、定理 2 により、固有ベクトルの精度が劣化する。 λ に誤差が含まれている限り、以降を無限精度で計算しても、この問題は解決しない。第三に、 $D + UCU^T$ が近接固有値を持つとき、 $F(\lambda)$ の計算で重大な桁落ち誤差が発生する可能性がある点。近接固有値を持つとき、これらの固有値の間に D の対角成分が存在することが多い。このとき $F(\lambda)$ の対角成分の計算には、異符号で、絶対値が大きく同程度の 2 数の和が必要で、この部分に桁落ちの可能性がある。

なお、以上の 3 つ以外の問題点としては、(4.2), (4.3) の評価は、同式の $\mathbf{x}_1, \mathbf{x}_2$ の誤差を考慮していないため、正確な直交性の指標を与えない可能性が考えられる。この場合、直交していない 2 ベクトルを直交していると誤判定することで、本来属すべきグループから外れるベクトルが発生し、そのベクトルの再直交化が正しく達成されない可能性がある。しかし、一般に、直交していないベクトルを直交していると誤判定することは極めて稀であり、対策の必要はないと考えられる。実際、著者が行った数値実験では、再直交化後の固有ベクトルは直交性や残差において良好で、直交性の誤判定は確認されなかった。その数値実験の際には、成分が乱数の実対称三重対角行列や、[16] から無作為に選んだ 30 種の実対称行列を三重対角化した行列を与えた。

上記の 3 つの問題点の影響を緩和するため、定理 1 に代えて、次の定理で固有ベクトルの計算を行う。

定理 3 では、 $F(\lambda)$ より次元が大きく、行列式が 0 になる λ が $F(\lambda)$ と全て等しい行列 $\tilde{F}(\lambda)$ を定義し、それに基づいて $D + UCU^T$ の固有ベクトルを求める。 $F(\lambda)$ を $\tilde{F}(\lambda)$ に拡大することで、行列式が 0 付近において、 λ に対する固有値の変動を小さくすることができる。この観点から見た $F(\lambda)$ と $\tilde{F}(\lambda)$ の関係は、後述の定理 4 で示す。

定理 3. D を n 次実対角行列、 U を $n \times k$ 行列、 C を k 次正則実対角行列とする。 λ を $D + UCU^T$ の多重度 1 の固有値とし、 $\det(\lambda I_n - D) \neq 0$ を満たすとする。 1 以上 n 以下の相異なる m 個の整数を i_1, i_2, \dots, i_m とし、 $W = (\mathbf{e}_{i_1}^{(n)}, \mathbf{e}_{i_2}^{(n)}, \dots, \mathbf{e}_{i_m}^{(n)})$ とする。 $k + m$ 次実対称行列 $\tilde{F}(\lambda)$ を

$$(5.1) \quad \tilde{F}(\lambda) = \begin{pmatrix} C^{-1} - U^T(I_n - WW^T)(\lambda I_n - D)^{-1}U & U^TW \\ WTU & W^T(\lambda I_n - D)W \end{pmatrix}$$

とおくと、これは特異行列である。 $\tilde{F}(\lambda)$ の固有値 0 に対応する固有ベクトルを $(\mathbf{y}^T \mathbf{z}^T)^T$ とするとき、 λ に対応する $D + UCU^T$ の固有ベクトルは、

$$\mathbf{q} = (\lambda I_n - D)^{-1}U\mathbf{y}$$

$$= (\lambda I_n - D)^{-1}(I_n - WW^T)Uy - Wz$$

と表せる. ただし, $e_j^{(n)}$ は n 次単位行列 I_n の第 j 列ベクトルとする.

証明 付録参照のこと. ■

ここで, i_1, i_2, \dots, i_m は $F(\lambda)$ と比較して, $\tilde{F}(\lambda)$ の左上 k 次ブロックの計算から取り除かれるため, 以下では「除去インデックス」と呼ぶ.

定理 3 を用いて, n 次元の $D + UCU^T$ の固有ベクトルを求めるための演算量を評価する. U の各列ベクトル u_i は, (3.3) の形式を仮定する. 固有ベクトル 1 本を求めるためには, 以下の演算が必要である. まず, 次節で述べる方法による除去インデックスの選択 (後述の (5.4) の計算) に $7n$ の演算量が必要である. これには, U の形式を利用した. 次に $\tilde{F}(\lambda_i)$ を求める. U の形式より, $\tilde{F}(\lambda_i)$ の左上 k 次ブロックは実三重対角行列となり, $\tilde{F}(\lambda_i)$ を求めるためには, $4(3 - 1/k)n$ の演算量が必要である. 以下ではこれを $12n$ とする. 次に, $\tilde{F}(\lambda_i)$ の絶対値最小の固有値に対応する固有ベクトル x を求める. $\tilde{F}(\lambda_i)$ の次元は $m + k$ である. 後述するように数値実験では m は平均して k 程度であるから, n 次元実対称行列の絶対値最小の固有値に対応する固有ベクトルを逆反復法で求めるために必要な演算量を $t_{II}(n)$ とすると, x は, $t_{II}(2k)$ の演算量で求められる. 最後に, $q = (\lambda I_n - D)^{-1}(I_n - WW^T)Uy - Wz$ を求める. U の形式より, Uy の計算には最大 $3n$ の演算量が必要. $(\lambda I_n - D)^{-1}(I_n - WW^T)$ は対角行列であるから, q の計算には $5n$ 程度の演算量が必要である. 従って, $D + UCU^T$ の固有ベクトル 1 本の演算量は, $7n + 12n + t_{II}(2k) + 5n = 24n + t_{II}(2k)$ である. 従って, $D + UCU^T$ の全固有ベクトルの演算量は,

$$(5.2) \quad 24n^2 + nt_{II}(2k)$$

である.

除去インデックスは, 1 以上 n 以下の相異なる整数である限り任意だが, 固有ベクトルの直交性を改善できる選び方は後述する.

5.2 除去インデックスの選び方

前小節で述べた 3 つの問題点それぞれに対して, 効果的な除去インデックスを考える.

第一の問題点は, $D + UCU^T$ の固有値 λ が D の第 j 対角成分 d_j に近接している場合に起こる. 以下で, どの程度の間隔で問題が発生するかを考察する.

$D + UCU^T$ の固有ベクトル同士の内積を c (我々の実装では 10^{-14}) 程度にするためには, 定理 2 より, 同定理の $|\mu_i|$ が $c|\lambda_i - \lambda_j|$ 程度でなければならない. すなわち, $D + UCU^T$ の固有値の最小の間隔が 1 のオーダーであっても, $|\mu_i|$ は c のオーダーが要求される.

一方, 多くの固有値計算アルゴリズムでは, A の固有値 μ と対応する正規固有ベクトル x の精度は,

$$(5.3) \quad \frac{\|Ax - \mu x\|_2}{\|A\|_2} = O(\epsilon)$$

の評価（絶対評価）のみが保証されている。ただし、 ϵ はマシンイプシロン。したがって、 $\|A\|_2$ が c/ϵ の数倍以上の場合、 μ は c のオーダーの精度では計算できない。

$\|C^{-1} - U^T(\lambda I_n - D)^{-1}U\|_2$ は $\|(\lambda I_n - D)^{-1}\|_2 = \max_j \{|\lambda - d_j|^{-1}\}$ で近似できるため、以上により、固有値 λ_i に対応する固有ベクトルを求める際は、

$$|\lambda_i - d_j| < \frac{\epsilon}{c}$$

をみたす j を除去インデックスに選ぶことで、第一の問題点を改善できる。

第二の問題点も、 λ が d_j に近接している場合に起こるが、それは以下の理由による。

議論の準備として、 $D + UCU^T$ の近似固有値 λ_i の真の値 $\bar{\lambda}_i$ からの誤差に対して、 $\tilde{F}(\lambda_i)$ の絶対値最小の固有値 $\tilde{\mu}_i$ がどの程度の誤差を含むかを見る。

定理 4. $\mu(t)$ を定理 1 の $F(t)$ の絶対値最小固有値、対応する正規固有ベクトルを $\mathbf{y}(t)$ 、 $\tilde{\mu}(t)$ を定理 3 の $\tilde{F}(t)$ の絶対値最小固有値とすると、それぞれの t に関する微分係数 $\mu'(t) = d\mu(t)/dt$ 、 $\tilde{\mu}'(t) = d\tilde{\mu}(t)/dt$ の $D + UCU^T$ の多重度 1 の固有値 $\bar{\lambda}$ での値は、

$$\begin{aligned}\tilde{\mu}'(\bar{\lambda}) &= \mu'(\bar{\lambda})(1 + \|W^T(\bar{\lambda}I_n - D)^{-1}U\mathbf{y}(\bar{\lambda})\|_2^2)^{-1} \\ &= \mu'(\bar{\lambda}) \left(1 + \sum_{j=i_1, \dots, i_m} \frac{(e_j^{(n)T}U\mathbf{y}(\bar{\lambda}))^2}{(\bar{\lambda} - d_j)^2}\right)^{-1}\end{aligned}$$

の関係にある。

証明 付録参照のこと。 ■

λ_i の変化に対して $\tilde{\mu}_i$ が頑健であるためには、 $\tilde{\mu}'(\bar{\lambda}_i)$ が小さければよく、この定理から、

$$(5.4) \quad \frac{\|U^T e_j^{(n)}\|_2^2}{(\lambda_i - d_j)^2} \quad (\simeq \frac{(e_j^{(n)T}U\mathbf{y})^2}{(\bar{\lambda}_i - d_j)^2})$$

がある定数 η （我々の実装では 2）以上の j を除去インデックスに選ばばよい。

なお、定理 4 より、除去インデックスが 0 個から 1 個に増加したときの $\tilde{\mu}'(\lambda_i)$ の変化に対して、1 個からそれ以上に増えたときの変化は緩慢であるため、判定条件を分けることでより効果的に除去インデックスを選択することができる。

実装には、以上の 2 つの判定条件を使う。第三の問題点に対してもこの判定条件は有効で、副次的な効果として、 $\lambda_i - d_j$ が小さいときに、 j が除去インデックスに選ばれることで、桁落ち誤差を軽減することができる。

以上が除去インデックスの決定法であるが、この決定法は絶対評価に基づいており、 $D + UCU^T$ が定数倍された場合に対応できない。それは、 \tilde{F} が $D + UCU^T$ のスケーリングに対して変化するからである。そのため、 $D + UCU^T$ を何らかの意味で正規化しなければならない。我々は、分割前の実三重対角行列 T を正規化することを選択し、 T の全成分の絶対値最大の値を 1 にスケーリングした。

このように選んだ除去インデックスの数 m は、数値実験によると、平均して k 程度になる。その数値実験の際には、成分が乱数の実対称三重対角行列や、[16] から無作為に選んだ 30 種の実対称行列を三重対角化した行列を与えた。

5.3 定理 3 を用いたときの直交性の事後評価

この節の定理 3 を用い、効果的な除去インデックスを選ぶことで、固有ベクトルの精度を向上させることができるが、依然として直交性が不足する可能性がある。その場合は、アルゴリズム 1, 2 を用いて再直交化することになる。この小節では、定理 1 に代えて定理 3 を用いたときの、アルゴリズム 1 における直交性の効率的な事後評価について述べる。

定理 3 を用いて $D + UCU^T$ の固有ベクトルを計算したとき、固有ベクトル同士の内積は、次の定理で評価できる。

定理 5. λ_1, λ_2 は相異なる実数で、 $\det(\lambda_i I_n - D) \neq 0$ を満たすとする。 $\mathbf{x}_i = (\mathbf{y}_i^T \mathbf{z}_i^T)^T$ を定理 3 の $\tilde{F}_i \equiv \tilde{F}(\lambda_i)$ の固有ベクトルとし、対応する固有値を μ_i とする。

$$\mathbf{q}'_i = (\lambda_i I_n - D)^{-1} (I_n - W_i W_i^T) U \mathbf{y}_i - W_i \mathbf{z}_i, \quad \mathbf{q}_i = \alpha_i \mathbf{q}'_i, \quad \alpha_i = \|\mathbf{q}'_i\|_2^{-1}$$

とするとき、

$$(5.5) \quad \mathbf{q}_1^T \mathbf{q}_2 = \frac{\alpha_1 \alpha_2}{\lambda_1 - \lambda_2} \{ (\mu_1 - \mu_2) \mathbf{y}_1^T \mathbf{y}_2 - \mu_1 \mathbf{z}_1^T W_1^T \mathbf{q}'_2 + \mu_2 \mathbf{z}_2^T W_2^T \mathbf{q}'_1 \}$$

が成り立つ。特に、 $W_1 = W_2$ なら、

$$(5.6) \quad \mathbf{q}_1^T \mathbf{q}_2 = \alpha_1 \alpha_2 \frac{\mu_1 - \mu_2}{\lambda_1 - \lambda_2} \mathbf{x}_1^T \mathbf{x}_2$$

証明 付録参照のこと。 ■

この定理によると、 $D + UCU^T$ の正規固有ベクトル同士の内積の絶対値 $|\mathbf{q}_i^T \mathbf{q}_j|$ は、

$$(5.7) \quad |\mathbf{q}_i^T \mathbf{q}_j| \leq \alpha_i \alpha_j \frac{1}{|\lambda_i - \lambda_j|} (|\mu_i - \mu_j| + \alpha_j |\mu_i| + \alpha_i |\mu_j|)$$

$$(5.8) \quad \leq \alpha_i \alpha_j \frac{(1 + \alpha_j) |\mu_i| + (1 + \alpha_i) |\mu_j|}{|\lambda_i - \lambda_j|}$$

で、上から押さえられる。よって、アルゴリズム 1 のグループ分けに用いる直交性の判定では、全ての内積を計算する代わりに、(5.8) が閾値より小さければ直交していると判定し、そうでない場合に実際に通常の方法で内積を計算して判定することで演算量を削減できる。

6. deflation

deflation とは、対角行列と階数 1 の摂動の和の固有値問題から、自明な固有値と固有ベクトルを取り除くことで、演算量を削減する手法である。この節では、対角行列と階数 k の摂動の和の固有値問題に適用できるように、deflation を拡張する。

6.1 deflation の拡張

一般の n 次対角行列 $D = \text{diag}(d_1, d_2, \dots, d_n)$ と階数 k の摂動の和 $D + UCU^T$ の固有値問題にも $k = 1$ の場合と同様に, 自明な固有値と固有ベクトルが存在する. それらを取り除くことで, 減次をすることができる. この操作を deflation と称する. 摂動の階数が 2 以上の場合の deflation は前論文では述べておらず, 本論文が初出の手法である. U の (i, j) 成分を u_{ij} とするとき, deflation には, 以下の二つの場合がある.

1. $u_{l1} = u_{l2} = \dots = u_{lk} = 0$
2. $d_l = d_{l+1} = \dots = d_{l+m-1}$ かつ, $W = (e_l^{(n)}, e_{l+1}^{(n)}, \dots, e_{l+m-1}^{(n)})$ を用いて,
 $\text{rank}(W^T U) \equiv s < m$

1 の場合は, 固有値が d_l で対応する固有ベクトルが $e_l^{(n)}$ である. 2 の場合は, 重複度 $m - s$ の固有値 d_l が存在し, 対応する固有空間は $\text{Ker}(U^T W W^T)$ である.

有限精度演算下では, 条件 1 は次の条件に緩和する必要がある. ただし, ϵ_0 はマシンイプシロンの定数倍とする (我々の実装では 8 倍).

$$1'. \quad \epsilon_0 \|D + UCU^T\|_2 \geq \|UCU^T e_l^{(n)}\|_2 \quad (= \|d_l e_l^{(n)} - (D + UCU^T) e_l^{(n)}\|_2)$$

ただし, $\|D + UCU^T\|_2$ は事前に知ることはできないので, 代わりに $\|D\|_2$ で近似する.

以上は, 一般的な対角行列と階数 k の摂動の和の固有値問題に対する deflation の議論だが, 以降は $k + 1$ 分割の分割統治法から生じる固有値問題に対して議論する. すなわち, 摂動の成分 $U = (u_1, u_2, \dots, u_k)$ は (3.3) の形式を仮定する. このとき, 条件 1' は, $(u_i, u_j) = 2\delta_{ij}$ を用いて

$$1''. \quad \sqrt{2 \sum_{j=1}^k c_j^2 u_{lj}^2} \leq \epsilon_0 \max_j |d_j|$$

と簡単化できる. 1'' による判定は, 1 要素 $O(k)$ の演算量で, 全ての対角要素に対して判定すれば $O(nk)$ の演算量である.

分割統治法に由来する対角行列と階数 k の摂動の和の固有値問題は, 1'' の条件は頻繁に満たす. 例えば, 2000 次の $(-1, 1)$ の範囲の一樣乱数の要素を持つ三重対角行列に対して, 4 分割の分割統治法を適用すると, 平均して 1700 の要素が 1 の条件を満たし, 約 300 次元の問題に減次される. それに対し, 2 の条件を満たすことは稀なため, 1 の場合だけ実装した.

1'' の条件により得られた自明な固有値, 固有ベクトルが $n - r$ 個あるとき, $D + UCU^T$ は r 次対角行列 D_1 , 自明な固有値を対角要素にもつ $n - r$ 次対角行列 D_2 , $r \times k$ 行列 U_1 , 適当な n 次順列行列 P を用いて,

$$(6.1) \quad P(D + UCU^T)P^T = \left(\begin{array}{c|c} D_1 + U_1 C U_1^T & \\ \hline & D_2 \end{array} \right)$$

と変形でき, $D + UCU^T$ の固有値問題は, r 次固有値問題 $D_1 + U_1 C U_1^T = Q \Lambda Q^T$ に減次できる.

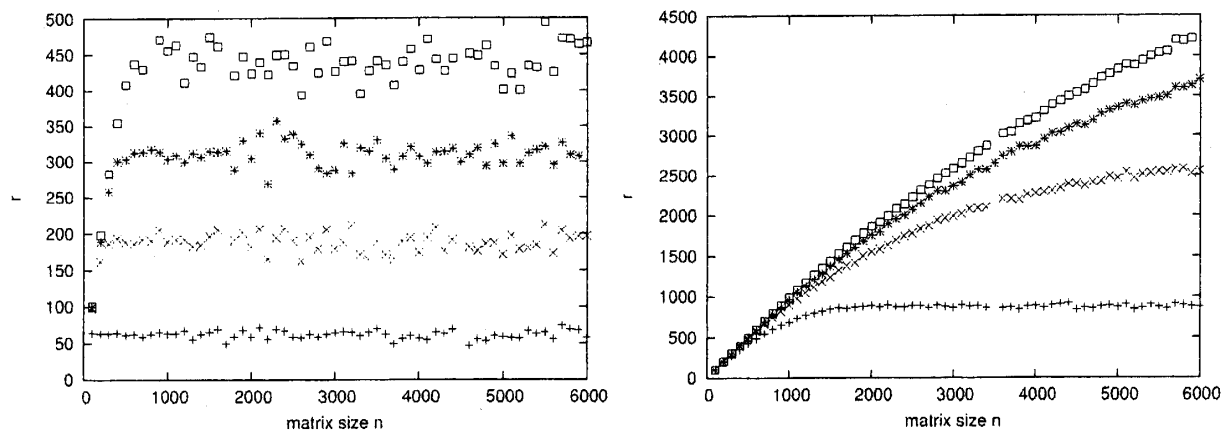


Fig. 1: Number of occurrences of deflation. Left figure is for a symmetric tridiagonal matrix such that the diagonal elements are uniform random numbers between -2 and 2 , while the subdiagonal elements are uniform random numbers between -1 and 1 . Right figure is for a symmetric tridiagonal matrix such that the diagonal elements are uniform random numbers between 2 and 4 , while the subdiagonal elements are uniform random numbers between 1 and 2 .

(6.1) を満たし、かつ、

$$\begin{pmatrix} Q_1 & & \\ & \ddots & \\ & & Q_k \end{pmatrix} P^T = \begin{pmatrix} Q_1^{(1)} & & Q_1^{(2)} \\ & \ddots & \\ & & Q_k^{(1)} \end{pmatrix} \begin{pmatrix} Q_1^{(2)} & & \\ & \ddots & \\ & & Q_k^{(2)} \end{pmatrix}$$

の形式にできる P が存在するため、 $D_1 + U_1 C U_1^T$ の固有値問題を解いた後、分割統治法の最後の行列行列積は、

$$\begin{pmatrix} Q_1 & & \\ & \ddots & \\ & & Q_k \end{pmatrix} P^T \left(\begin{array}{c|c} Q & \\ \hline I_{n-r} \end{array} \right) = \left(\begin{pmatrix} Q_1^{(1)} & & \\ & \ddots & \\ & & Q_k^{(1)} \end{pmatrix} Q \mid \begin{array}{c} Q_1^{(2)} \\ \vdots \\ Q_k^{(2)} \end{array} \right)$$

と変形して、レベル 3 の BLAS を用いる。ここで $Q_j^{(1)}$ と $Q_j^{(2)}$ の列ベクトルを合わせ、列を適当に並べ替えると、 Q_j が再現される。

deflation 発生率を $\delta = 1 - r/n$ とするとき、この行列積は、 $(1 - \delta)^2 (2/k) n^3$ の演算量になる。

6.2 deflation 発生率

対象となる行列によって、deflation の発生率には大きな差異がある。Fig. 1 はその差異を示したものである。左図は、対角成分が区間 $(-2, 2)$ 、副対角成分が区間 $(-1, 1)$ の一様乱数である実対称三重対角行列の集合 T_A による図、右図は、対角成分が $(2, 4)$ 、副対角成

分が $(1, 2)$ の区間の一様乱数である実対称三重対角行列の集合 T_B による図である。それぞれ、横軸は行列の次元 n を、縦軸は前小節の r 、すなわち n 次固有値問題を k 分割の分割統治法で、1 階層だけ計算したときに生じる n 次元の対角行列と階数 $k-1$ の摂動の和 $D + UCU^T$ の固有値問題の deflation 後に生じる自明でない固有値問題の次元である。言い換えれば r は、 n 次三重対角行列 T を (3.1) に従って、 k 個の n/k 次三重対角行列 T_j に分割したとき、 T の固有値とは異なる T_1, T_2, \dots, T_k の固有値の総数である。deflation 発生率は $\delta = 1 - r/n$ 。マークはそれぞれ、+ が 2 分割、 \times が 4 分割、* が 6 分割、 \square が 8 分割を表す。

これらの図より、 T_A は deflation が非常に発生する性質を持ち、それと比較して T_B は deflation が発生しにくい性質を持つことが分かる。また、両者ともに分割数の増加に伴って deflation 発生率が低下する。

その結果として、この階層における行列行列積の演算量が $(2/k)(1-\delta)^2 n^3 = (2/k)r^2 n$ であることに注意すると、分割数 k の増加による演算量の推移に対して、次の知見が得られる。 T_A は、分割数の増加による演算量の減少と、deflation 発生率低下による演算量増加の効果が相殺されて、分割数の増加によるメリットはあまり期待できない。一方 T_B は、分割数の増加による演算量の減少より、deflation 発生率低下による演算量増加が小さく、分割数の増加による演算量の減少が期待できる。従って、最適な分割数を決めるには、対象となる行列の性質を知らなければならない。

7. アルゴリズムのまとめ

これまでに述べたアルゴリズムをまとめると、我々の提案する k 分割の分割統治法は以下ようになる。

アルゴリズム 3 分割統治法 (k 分割) を用いて、 n 次実対称三重対角行列 T の固有値・固有ベクトルを求める:

proc dc_eig_k(T, Q, Λ) 入力 T から $T = Q\Lambda Q^T$ を満たす直交行列 Q , 対角行列 Λ を出力する

除去インデックスを決定するためのパラメータ c, η を決める。

$\rho := \max\{\max_j\{|d_j|\}, \max_j\{|e_j|\}\}$ を決め、 $d_j := d_j/\rho$, $e_j := e_j/\rho$ とスケールする。

$T = T_1 \oplus \dots \oplus T_k + c_1 v_1 v_1 + \dots + c_{k-1} v_{k-1} v_{k-1}$ と分割する。

固有値問題 $T_j = Q_j \Lambda_j Q_j^T$ ($j = 1, \dots, k$) を (再帰的に) 解く。

Λ_j, Q_j を用いて T を $D + UCU^T$ の形式にする。

6 節の deflation を用いて、適当な順列行列 P で $D + UCU^T$ を r 次対称行列 $(D_1 + U_1 C U_1^T) \oplus D_2$ の形式にする。

$D_1 + U_1 C U_1^T$ の固有値 λ_j ($j = 1, \dots, r$) を

前論文のアルゴリズム 3' [13, pp. 102-103] を用いて求める.

(ただし, 4 倍精度演算を全て倍精度演算に変え, \tilde{U} は出力しない. それに伴って, 2 つ目の j に関する **do** ループ内の最後の 6 行 [13, p. 103, ll. 1-6] は実行しない)

do $i := 1$ **to** r

do $j := 1$ **to** r

if $(|\lambda_i - d_j| < \epsilon/c)$ **or** $(\|U_1^T \mathbf{e}_j^{(n)}\|_2^2 / (\lambda_i - d_j)^2 \geq \eta)$

j を除去インデックスに選ぶ.

end if

end do

 選ばれた除去インデックスによる定理 3 で \mathbf{q}'_i を求める.

 (このとき, U_1 の構成から, $\tilde{F}(\lambda_i)$ の

 左上 $k-1$ 次ブロックは実三重対角行列になることに注意する)

end do

アルゴリズム 1 を用いて, \mathbf{q}'_i をグループ分けする.

(直交性の判定には定理 5 を用いる)

グループごとに不変部分空間の判定を行い, 必要ならグループを集約する.

foreach G **in** 全てのグループ

if G に属するベクトルの本数 ≥ 2

 アルゴリズム 2 で G に属するベクトルを再直交化する.

end if

end foreach

$Q := (Q_1 \oplus \cdots \oplus Q_k) P^T (Q' \oplus I_{n-r})$ を求める.

$\lambda_j := \rho \lambda_j$ とスケーリングを戻す.

実装の際には, $c = 10^{-14}$, $\eta = 2$ とした.

続いて, アルゴリズム 3 の演算量を評価する.

まず, 行列に依存するパラメータを準備する. n 次元の実対角行列と階数 $k-1$ の摂動の和の固有値問題が, deflation により r_n 次元に減るとする. すなわち, U の行のうち一つでも非零の要素がある行の数が r_n である. また, 平均 g_n 個の再直交化すべきグループがあり, 各々のグループに平均して v_n 本のベクトルが含まれると仮定する.

総演算量を $t_k(n)$ とすると, 以下の無視できない 9 つの演算部分がある. (1) スケーリングに $2n$ の演算量. (2) 分解した小問題を解くために $kt_k(n/k)$ の演算量. (3) deflation が発生するか否かの判定 (6.1 節の条件 1" の判定) に $7n$ の演算量. (4) r_n 次元の $D_1 + U_1 C U_1^T$ の全ての固有値を求めるために $(2k/(k-1))(c_{R1} + 2/3)r_n^2$ の演算量 [13, 式 (3.13)]. 以降の計算の簡略化のため, この演算量を $2(c_{R1} + 2/3)r_n^2$ とする. ただし, m 次元の対角行列と階数 1 の摂動の和の全ての固有値は $c_{R1} m^2$ の演算量で求められるとし, アルゴリズム中で

前論文のアルゴリズム 3' から取り除いた 6 行に相当する演算量を省いた. (5) $D_1 + U_1 C U_1^T$ の全ての固有ベクトルを求めるために, (5.2) より, $24r_n^2 + r_n t_{II}(2(k-1))$ の演算量. $t_{II}(x)$ は (5.2) の 5 行上で定義した. (6) 定理 5 を用いた直交性の判定に $(9/2)r_n^2$ の演算量. (7) 実際の再直交化に, (4.1) より, 約 $g_n\{5v_n^2 r_n + kv_n^2 + t_{QR}(v_n)\}$ の演算量. $t_{QR}(x)$ は (4.1) の 3 行上で定義した. ただし, 以降の計算の簡略化のために $v_n(v_n + 1)$ を v_n^2 とした. (8) Q を求めるための行列行列積は, 6.1 節の最後の記述より, $(2/k)r_n^2 n$ の演算量. (9) 最後のスケーリングは n の演算量.

よって,

$$\begin{aligned} t_k(n) &= 2n + kt_k(n/k) + 7n + 2(c_{R1} + 2/3)r_n^2 + 24r_n^2 + t_{II}(2(k-1))r_n \\ &\quad + (9/2)r_n^2 + g_n\{5v_n^2 r_n + kv_n^2 + t_{QR}(v_n)\} + (2/k)r_n^2 n + n \\ &= kt_k(n/k) + 10n + (2(c_{R1} + 2/3) + 24 + 9/2)r_n^2 + t_{II}(2(k-1))r_n \\ &\quad + g_n\{5v_n^2 r_n + kv_n^2 + t_{QR}(v_n)\} + (2/k)r_n^2 n \end{aligned}$$

となる.

ここで, $r_n = \gamma n$, $g_n = \sigma n^\alpha$, $v_n = \tau n^\beta$, ($0 \leq \gamma, \alpha, \beta \leq 1$, $\alpha + \beta = 1$) とする. すなわち, $1 - \gamma$ は deflation 発生率 (6.1 節の δ) であり, $\sigma\tau$ は再直交化発生率である. $t_{QR}(m) = c_{QR}m^3$, $t_{II}(m) = c_{II}m^3$ とする. さらに, 簡単化のために, $\alpha = \beta = 1/2$ とする. これらを代入すると,

$$\begin{aligned} t_k(n) &= kt_k(n/k) + 10n + (2(c_{R1} + 2/3) + 24 + 9/2)(\gamma n)^2 + c_{II}(2(k-1))^3 \gamma n \\ &\quad + \sigma\sqrt{n}\{5(\tau\sqrt{n})^2 \gamma n + k(\tau\sqrt{n})^2 + c_{QR}(\tau\sqrt{n})^3\} + (2/k)(\gamma n)^2 n \\ &= kt_k(n/k) + (2/k)\gamma^2 n^3 + 5\sigma\tau^2 \gamma n^{5/2} \\ &\quad + \{(2c_{R1} + 29 + 5/6)\gamma^2 + c_{QR}\sigma\tau^3\} n^2 + k\sigma\tau^2 n^{3/2} + O(n) \end{aligned}$$

となる. 一般に,

$$t(n) = kt(n/k) + \sum_{i=1}^N c_i n^{e_i} + O(n), \quad t(1) = O(1), \quad (e_i > 1)$$

のとき,

$$t(n) = \sum_{i=1}^N c_i \frac{k^{e_i-1}}{k^{e_i-1} - 1} n^{e_i} + O(n)$$

であることを用いると,

$$\begin{aligned} t_k(n) &= \gamma^2 \frac{2k}{k^2 - 1} n^3 + 5\sigma\tau^2 \gamma \frac{k^{3/2}}{k^{3/2} - 1} n^{5/2} \\ &\quad + \{(2c_{R1} + 29 + 5/6)\gamma^2 + c_{QR}\sigma\tau^3\} \frac{k}{k - 1} n^2 + \sigma\tau^2 \frac{k^{3/2}}{k^{1/2} - 1} n^{3/2} + O(n) \end{aligned}$$

$k \ll n$ のときは, $t_k(n) \simeq \gamma^2((2k)/(k^2 - 1))n^3$ となり, deflation が発生しなければ ($\gamma = 1$), (3.4) の評価と一致する.

なお, 我々の実装でのパラメータの値は, $c_{R1} = 30$ [13], $c_{QR} = 26/3$ [3] である. γ, σ, τ は行列によって異なる.

8. 数値実験

この節では, n 次実対称三重対角行列 T の固有値問題 $T = Q\Lambda Q^T$ を, アルゴリズム 3 に従った k 分割の分割統治法で求めたときの, 精度と速度についての数値実験を行った結果を示す. ただし, $Q = (q_1, q_2, \dots, q_n)$ は直交行列, $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$ は対角行列である.

全ての実験において, T の全固有値 $\lambda_1, \lambda_2, \dots, \lambda_n$ と, 対応する全固有ベクトル q_1, q_2, \dots, q_n を求めた.

精度については, 直交誤差 ϵ_O と残差 ϵ_R

$$\epsilon_O = \max_{1 \leq i \leq n} \|Q^T q_i - e_i^{(n)}\|_2, \quad \epsilon_R = \max_{1 \leq i \leq n} \frac{\|T q_i - \lambda_i q_i\|_2}{\|T\|_2}$$

を用いて評価する.

8.1 計算機環境

この節の実験は, 以下の環境で行った.

CPU	Intel Pentium 4 HT 3.2 GHz
OS	Windows XP SP 2
メインメモリ	1 GB
使用コンパイラ	g77 2.95.3
コンパイラオプション	-O3
LAPACK	Version 3.0 (update; May 31, 2000)
BLAS	LAPACK 付属のコード

コードは Fortran 77 で作成した. コード自体のチューニングはしておらず, ほぼアルゴリズムに忠実なコードとなっている.

我々が提案する分割統治法では補助解法として, QR 法 (LAPACK ルーティン DSTEQR), 実対称三重対角化 + QR 法 (LAPACK ルーティン DSYEV), 二分法 + 逆反復法 (LAPACK ルーティン DSYEVX) を用いた. DSTEQR は, 分割が進行して小行列の次元が $\max\{k, 25\}$ 以下 (k は分割数) になったときに, 再帰を打ち切って小行列の固有値・固有ベクトルを求めるときに用いた. なお, 2 分割の分割統治法を実装した LAPACK ルーティン DSTEVD でも, $n = 25$ で分割統治法と QR 法を切り替える. DSYEV は, アルゴリズム 2 の A の固有

ベクトルを求めるときに用いた. DSYEVX は, 定理 3 の \tilde{F}_i の絶対値最小固有値とそれに対応する固有ベクトルを求めるときに用いた.

実験は全て, 我々の分割統治法と, 2 分割の分割統治法を実装した LAPACK ルーティン DSTEVD とを比較した.

また, 提案手法と比較対象の DSTEVD は, とともに BLAS を用いているが, 今回の実験では計算機環境専用に最適化された BLAS を用いてはいない. このため, 最適化された BLAS を用いて同様の実験を行った場合, 提案手法と DSTEVD との実行時間の差が縮まる可能性があることを注意しておく.

8.2 結果

精度に対する実験結果から示す.

Table 1 は, 対角要素が区間 $(2, 4)$, 副対角要素が区間 $(1, 2)$ の一様な乱数である n 次実対称三重対角行列の固有値問題を, Table 2 は, 対角要素が区間 $(-2, 2)$, 副対角要素が区間 $(-1, 1)$ の一様な乱数である n 次実対称三重対角行列の固有値問題を, Table 3 は,

$$(8.1) \quad \begin{pmatrix} W_{21} & \beta & & \\ \beta & W_{21} & & \\ & & \ddots & \beta \\ & & \beta & W_{21} \end{pmatrix}$$

の固有値問題 [19] を, それぞれ k 分割の分割統治法で解いたときと, LAPACK ルーティン DSTEVD で解いたときの精度の比較を表す. ここで, W_{21} は 21 次三重対角行列で, 対角成分が $10, 9, \dots, 1, 0, 1, \dots, 9, 10$, 副対角成分は 1 である. β は実数である. (8.1) は, 近接した固有値の組を複数もつために固有ベクトルの直交性確保が困難で, 分割統治法が苦手としている行列である. しかし, DSTEVD は Löwner の定理に基づく直交化手法によって十分な精度で計算可能であり, 数値結果でも精度に問題はない.

これらの表から, 分割統治法が不得手とする問題であっても, k 分割の分割統治法は, 分割数 k に関係なく, DSTEVD と比較しうる精度を有することがわかる. 後述するが, 高精度に固有値問題を解くための再直交化は, 速度の負荷にもなっていないことをあらかじめ述べておく.

続いて, 計算速度の実験結果を示す. 精度に関する実験と異なる行列に対して実験している理由は, 以降で述べる 2 種の行列が, 速度に関して両極端の性質を示すからである. これらの行列に対しても, 精度が劣化してはいないことを断っておく.

ここで, 実行時間が最短である分割数を, 最適な分割数と定義する. この値は, 行列の次元や deflation 発生率に大きく依存する.

Table 4 は, 対角要素が区間 $(-2, 2)$, 副対角要素が区間 $(-1, 1)$ の一様な乱数である n 次実対称三重対角行列の固有値問題を, Table 5 は, 第 j 対角要素が $j \times 10^{-6}$ で, 副対角

Table 1: Numerical precision for a symmetric tridiagonal matrix such that the diagonal elements are uniform random numbers between 2 and 4, while the subdiagonal elements are uniform random numbers between 1 and 2.

n	分割統治法 ($k = 8$)		分割統治法 ($k = 16$)		DSTEVD	
	ϵ_R	ϵ_O	ϵ_R	ϵ_O	ϵ_R	ϵ_O
1000	2.35×10^{-15}	7.14×10^{-15}	2.28×10^{-15}	5.23×10^{-15}	2.59×10^{-15}	2.08×10^{-15}
3000	4.43×10^{-15}	1.54×10^{-14}	4.33×10^{-15}	2.51×10^{-14}	7.16×10^{-15}	3.51×10^{-15}

Table 2: Numerical precision for a symmetric tridiagonal matrix such that the diagonal elements are uniform random numbers between -2 and 2 , while the subdiagonal elements are uniform random numbers between -1 and 1 .

n	分割統治法 ($k = 8$)		分割統治法 ($k = 16$)		DSTEVD	
	ϵ_R	ϵ_O	ϵ_R	ϵ_O	ϵ_R	ϵ_O
1000	8.08×10^{-15}	7.27×10^{-14}	2.51×10^{-15}	4.42×10^{-14}	6.19×10^{-15}	2.75×10^{-15}
3000	2.24×10^{-15}	5.39×10^{-14}	2.28×10^{-15}	1.40×10^{-14}	5.40×10^{-15}	3.38×10^{-15}

Table 3: Numerical precision for a matrix in (8.1).

		ϵ_O	ϵ_R
DSTEVD	$n = 105, \beta = 10^{-2}$	9.8×10^{-16}	8.6×10^{-16}
	$n = 105, \beta = 10^{-4}$	9.5×10^{-16}	9.3×10^{-16}
分割統治法 ($k = 4$)	$n = 105, \beta = 10^{-2}$	9.8×10^{-16}	1.8×10^{-15}
	$n = 105, \beta = 10^{-4}$	9.6×10^{-16}	3.2×10^{-15}

要素が全て 1 の n 次実対称三重対角行列 T_g の固有値問題 [12] を, それぞれ k 分割の分割統治法で解いたとき, LAPACK ルーティン DSTEVD で解いたときの実行時間の比較を表す. 単位は秒である. T_g は, 2 分割の分割統治法では deflation が全く起こらない行列である. 実際, 今回実験した分割数での多分割の分割統治法でも deflation は全く起こらなかった.

Table 4 の行列は, 6 節にあるように deflation が非常に多く発生し, 分割数の増加に対し演算量の減少が見込めない行列である. 実験結果から, この行列に対する最適な分割数は, $n = 1000$ のとき 2, 3, 4 であり, 2000 のとき 3, 3000 のとき 5, 4000 のとき 4, 5000 のとき 6, $n = 6000$ では 4 である. 行列の次元に対して最適な分割数はあまり増加せず, 最大の減少率を示す $n = 6000$ で最適な分割数 ($k = 4$) に設定しても, 我々の実装での 2 分割の分割統治法より, 18 パーセントに実行時間の減少がとどまっている. しかし, DSTEVD

Table 4: Execution time in second for a symmetric tridiagonal matrix such that the diagonal elements are uniform random numbers between -2 and 2 , while the subdiagonal elements are uniform random numbers between -1 and 1 .

n	k 分割の分割統治法									DSTEVD
	2	3	4	5	6	7	8	9	10	
1000	0.16	0.16	0.16	0.18	0.20	0.18	0.19	0.22	0.24	0.11
2000	0.43	0.39	0.44	0.43	0.48	0.51	0.53	0.48	0.50	0.38
3000	0.81	0.78	0.76	0.73	0.79	0.87	0.92	0.99	1.02	0.82
4000	1.36	1.19	1.16	1.25	1.17	1.26	1.38	1.51	1.53	1.45
5000	1.98	1.70	1.64	1.74	1.63	1.70	1.85	2.09	2.14	2.22
6000	2.72	2.28	2.22	2.33	2.37	2.23	2.36	2.67	2.84	2.91

Table 5: Execution time in second for a symmetric tridiagonal matrix such that j -th diagonal element is $j \times 10^{-6}$, while all the subdiagonal elements are 1.

n	k 分割の分割統治法								
	2	3	4	5	6	7	8	9	10
1000	2.84	1.92	1.09	1.13	1.05	1.00	0.95	1.13	1.43
2000	20.60	13.13	10.50	9.62	7.91	6.01	4.80	5.42	5.23
3000	66.23	41.48	31.42	27.92	25.36	23.61	22.06	21.14	17.14
4000	150.83	94.34	69.81	61.48	54.22	49.75	47.28	47.20	45.81
5000	289.13	179.55	134.11	114.27	99.70	92.14	84.72	85.38	82.86

n	k 分割の分割統治法								DSTEVD
	11	12	13	14	15	16	17	18	
1000	1.14	1.40	1.18	1.01	1.05	1.29	1.39	1.49	2.14
2000	5.22	4.95	5.15	7.85	6.45	8.29	9.43	15.32	17.14
3000	14.36	14.33	14.25	14.41	15.75	13.78	18.36	27.30	56.97
4000	43.88	40.00	41.03	38.58	33.75	29.36	44.33	46.44	133.34
5000	85.72	83.27	74.86	73.75	86.58	80.81	79.59	87.69	258.05

と比較しても、 $n = 6000$ で最適な分割数ならば 25 パーセントの実行時間が減少している。また、我々の実装で $k = 2$ の場合と DSTEVD を比較しても、実行速度に遜色はなく、 n が 3000 以上では我々が勝っていることから、我々が用いている再直交化は、DSTEVD が用いている Löwner の定理に基づく直交化手法と比較しても、実行時間で不利になることはないと言える。

Table 5 は、deflation が発生しない行列であるので、分割数の増加に従って、実行時間が大幅に減少する。この行列に対する最適な分割数は、 $n = 1000, 2000$ のとき 8, $n = 3000, 4000$ のとき 16, $n = 5000$ では 13 である。最適な分割数での我々の実装と DSTEVD とを比較すると、我々の実装が DSTEVD より少なくとも 2 倍、最大で $n = 5000$ のとき 3.5 倍に速度が向上している。

9. 結論

本論文では、昨年我々が提案した、実対称三重対角固有値問題の k 分割の分割統治法の実装を改良した。

従来の実装において倍精度演算のみでは、分割統治法の精度面での核となる、実対角行列と階数 $k-1$ の摂動の和の固有値問題を高精度に解くことができず、固有ベクトルの直交性が不足する問題があった。その問題を解決するために、改良前には、適切な箇所に拡張精度演算を用いることで直交性を確保した。それに対し本論文では、拡張精度演算を用いずに、倍精度演算のみで固有ベクトルの精度を確保する方法を提案した。

対角行列と階数 $k-1$ の摂動の和の固有ベクトルの直交性が不足している問題は、全ての固有ベクトルの直交性が悪いわけではなく、一部の固有ベクトルの精度のみが不足していることに着目し、必要な固有ベクトルのみを再直交化することで、直交性を確保することができた。そのためには、直交しているか否かの判定が必要であるが、固有ベクトルの計算手法とそれまでに既知の値を利用することで、少ない演算量で評価が完了する方法もあわせて提案した。数値実験では、この再直交化手法が、速度低下を招いていないことを確認した。

また、対角行列と階数 $k-1$ の摂動の和の固有ベクトルの直交性不足の理由を明らかにし、その原因である固有ベクトルの計算法の改善方法も述べた。この改善法で全ての固有ベクトルの直交性が保証されるわけではないものの、再直交化すべきベクトルの本数を減少することに成功した。

速度の面でも、成果をあげた。2 分割の分割統治法では、演算量を削減する意味で必須といえる deflation という手法を、 k 分割の分割統治法に適用できるように拡張した。これにより、最大の演算コストがかかる行列行列積の演算を減らすことができ、2 分割の分割統治法と同等以上の実行速度を実現した。また、deflation の効果が高い行列と低い行列があることを示し、deflation の効果が低い行列に対して、分割数の増加による演算量減少の効果が高いという知見を得た。

今後の課題としては, deflation の演算量削減の効果に対しての理論的, 数値的な側面からの考察, 最適分割数の自動的な決定法の確立がある.

A. 定理の証明

定理 2. D を n 次実対角行列, U を $n \times k$ 行列, C を k 次正則行列とする. λ_1, λ_2 は相異なる実数で, $\det(\lambda_i I_n - D) \neq 0$ を満たすとする. \mathbf{x}_i を k 次実対称行列 $F_i = C^{-1} - U^T(\lambda_i I_n - D)^{-1}U$ の固有ベクトルとし, 対応する固有値を μ_i とする. $\mathbf{q}_i = \alpha_i(\lambda_i I_n - D)^{-1}U\mathbf{x}_i$, $\alpha_i = \|(\lambda_i I_n - D)^{-1}U\mathbf{x}_i\|_2^{-1}$ とするとき,

$$\mathbf{q}_1^T \mathbf{q}_2 = \alpha_1 \alpha_2 \frac{\mu_1 - \mu_2}{\lambda_1 - \lambda_2} \mathbf{x}_1^T \mathbf{x}_2$$

が成り立つ.

証明 $\lambda_i \neq \lambda_j$ より,

$$(\lambda_i I_n - D)^{-1}(\lambda_j I_n - D)^{-1} = -\frac{1}{\lambda_i - \lambda_j} \{(\lambda_i I_n - D)^{-1} - (\lambda_j I_n - D)^{-1}\}$$

である. よって,

$$\mathbf{q}_1^T \mathbf{q}_2 = \alpha_1 \alpha_2 \frac{1}{\lambda_1 - \lambda_2} \mathbf{x}_1^T (F_1 - F_2) \mathbf{x}_2$$

より, 題意が従う. ■

定理 3. D を n 次実対角行列, U を $n \times k$ 行列, C を k 次正則実対角行列とする. λ を $D + UCU^T$ の多重度 1 の固有値とし, $\det(\lambda I_n - D) \neq 0$ を満たすとする. 1 以上 n 以下の相異なる m 個の整数を i_1, i_2, \dots, i_m とし, $W = (\mathbf{e}_{i_1}^{(n)}, \mathbf{e}_{i_2}^{(n)}, \dots, \mathbf{e}_{i_m}^{(n)})$ とする. $k + m$ 次実対称行列 $\tilde{F}(\lambda)$ を

$$(A.1) \quad \tilde{F}(\lambda) = \begin{pmatrix} C^{-1} - U^T(I_n - WW^T)(\lambda I_n - D)^{-1}U & U^T W \\ W^T U & W^T(\lambda I_n - D)W \end{pmatrix}$$

とおくと, これは特異行列である. $\tilde{F}(\lambda)$ の固有値 0 に対応する固有ベクトルを $(\mathbf{y}^T \mathbf{z}^T)^T$ とするとき, λ に対応する $D + UCU^T$ の固有ベクトルは,

$$\begin{aligned} \mathbf{q} &= (\lambda I_n - D)^{-1}U\mathbf{y} \\ &= (\lambda I_n - D)^{-1}(I_n - WW^T)U\mathbf{y} - W\mathbf{z} \end{aligned}$$

と表せる. ただし, $\mathbf{e}_j^{(n)}$ は n 次単位行列 I_n の第 j 列ベクトルとする.

証明 定理 1 より, 同定理の $F(\lambda)$ は固有値 0 を持つ. $F(\lambda)$ の固有値 0 に対応する固有ベクトルを \mathbf{x} とすると,

$$\tilde{F}(\lambda) \begin{pmatrix} \mathbf{x} \\ -W^T(\lambda I_n - D)^{-1}U\mathbf{x} \end{pmatrix} = \mathbf{o}$$

となる. よって, $\tilde{F}(\lambda)$ は特異行列. 非零実数 κ を用いて, $\mathbf{x} = \kappa \mathbf{y}$ となるから, $\mathbf{q} = (\lambda I_n - D)^{-1} U \mathbf{y}$ が言える. また,

$$\begin{aligned}\mathbf{q} &= (\lambda I_n - D)^{-1} (I_n - WW^T) U \mathbf{y} + WW^T (\lambda I_n - D)^{-1} U \mathbf{y} \\ &= (\lambda I_n - D)^{-1} (I_n - WW^T) U \mathbf{y} - W \mathbf{z}\end{aligned}$$

である. ■

定理 4. $\mu(t)$ を定理 1 の $F(t)$ の絶対値最小固有値, 対応する正規固有ベクトルを $\mathbf{y}(t)$, $\tilde{\mu}(t)$ を定理 3 の $\tilde{F}(t)$ の絶対値最小固有値とすると, それぞれの t に関する微分係数 $\mu'(t) = d\mu(t)/dt$, $\tilde{\mu}'(t) = d\tilde{\mu}(t)/dt$ の $D + UCU^T$ の多重度 1 の固有値 $\bar{\lambda}$ での値は,

$$\begin{aligned}\tilde{\mu}'(\bar{\lambda}) &= \mu'(\bar{\lambda}) (1 + \|W^T(\bar{\lambda} I_n - D)^{-1} U \mathbf{y}(\bar{\lambda})\|_2^2)^{-1} \\ &= \mu'(\bar{\lambda}) \left(1 + \sum_{j=i_1, \dots, i_m} \frac{(e_j^{(n)T} U \mathbf{y}(\bar{\lambda}))^2}{(\bar{\lambda} - d_j)^2} \right)^{-1}\end{aligned}$$

の関係にある.

証明 Rayleigh 商より, $\mu(t) = \mathbf{y}(t)^T F(t) \mathbf{y}(t)$, $\tilde{\mu}(t) = \tilde{\mathbf{x}}(t)^T \tilde{F}(t) \tilde{\mathbf{x}}(t)$ である. ただし, $\tilde{\mathbf{x}}(t)$ は $\tilde{F}(t)$ の $\tilde{\mu}(t)$ に対応する正規固有ベクトルである. よって,

$$\begin{aligned}\mu'(t) &= \mathbf{y}(t)^T F'(t) \mathbf{y}(t) + 2\mathbf{y}(t)^T F(t) \mathbf{y}'(t) \\ &= \mathbf{y}(t)^T F'(t) \mathbf{y}(t) + 2\mu(t) \mathbf{y}(t)^T \mathbf{y}'(t) \\ &= \mathbf{y}(t)^T F'(t) \mathbf{y}(t) \\ &= \|(tI_n - D)^{-1} U \mathbf{y}(t)\|_2^2\end{aligned}$$

となる. 3 番目の等号には, $2\mathbf{y}(t)^T \mathbf{y}'(t) = d\|\mathbf{y}(t)\|_2^2/dt = 0$ を用いた. 同様に, $\tilde{\mu}'(t) = \mathbf{x}(t)^T \tilde{F}'(t) \mathbf{x}(t)$ である.

ここで, $t = \bar{\lambda}$ とすると, $\mu'(\bar{\lambda}) = \|(\bar{\lambda} I_n - D)^{-1} U \mathbf{y}(\bar{\lambda})\|_2^2$ であり,

$$\mathbf{x}(\bar{\lambda}) = \frac{1}{\sqrt{1 + \|W^T(\bar{\lambda} I_n - D)^{-1} U \mathbf{y}(\bar{\lambda})\|_2^2}} \begin{pmatrix} \mathbf{y}(\bar{\lambda}) \\ -W^T(\bar{\lambda} I_n - D)^{-1} U \mathbf{y}(\bar{\lambda}) \end{pmatrix}$$

であるから,

$$\begin{aligned}\tilde{\mu}'(\bar{\lambda}) &= \frac{\|(\bar{\lambda} I_n - D)^{-1} (I_n - WW^T) U \mathbf{y}(\bar{\lambda})\|_2^2 + \|W^T(\bar{\lambda} I_n - D)^{-1} U \mathbf{y}(\bar{\lambda})\|_2^2}{1 + \|W^T(\bar{\lambda} I_n - D)^{-1} U \mathbf{y}(\bar{\lambda})\|_2^2} \\ &= \frac{\|(\bar{\lambda} I_n - D)^{-1} U \mathbf{y}(\bar{\lambda})\|_2^2}{1 + \|W^T(\bar{\lambda} I_n - D)^{-1} U \mathbf{y}(\bar{\lambda})\|_2^2}\end{aligned}$$

より, 証明された. ■

定理 5. λ_1, λ_2 は相異なる実数で, $\det(\lambda_i I_n - D) \neq 0$ を満たすとする. $\mathbf{x}_i = (\mathbf{y}_i^T \mathbf{z}_i^T)^T$ を定理 3 の $\tilde{F}_i \equiv \tilde{F}(\lambda_i)$ の固有ベクトルとし, 対応する固有値を μ_i とする.

$$\mathbf{q}'_i = (\lambda_i I_n - D)^{-1} (I_n - W_i W_i^T) U \mathbf{y}_i - W_i \mathbf{z}_i, \quad \mathbf{q}_i = \alpha_i \mathbf{q}'_i, \quad \alpha_i = \|\mathbf{q}'_i\|_2^{-1}$$

とするとき,

$$\mathbf{q}_1^T \mathbf{q}_2 = \frac{\alpha_1 \alpha_2}{\lambda_1 - \lambda_2} \left\{ (\mu_1 - \mu_2) \mathbf{y}_1^T \mathbf{y}_2 - \mu_1 \mathbf{z}_1^T W_1^T \mathbf{q}'_2 + \mu_2 \mathbf{z}_2^T W_2^T \mathbf{q}'_1 \right\}$$

が成り立つ. 特に, $W_1 = W_2$ なら,

$$(A.2) \quad \mathbf{q}_1^T \mathbf{q}_2 = \alpha_1 \alpha_2 \frac{\mu_1 - \mu_2}{\lambda_1 - \lambda_2} \mathbf{x}_1^T \mathbf{x}_2$$

証明 $(\mathbf{y}_i^T \mathbf{z}_i^T)^T$ が $\tilde{F}(\lambda_i)$ の固有値 μ_i に対応する固有ベクトルであることから, \mathbf{q}_i は以下のように変形できる.

$$\begin{aligned} \mathbf{q}'_i &= (\lambda_i I_n - D)^{-1} (I_n - W_i W_i^T) U \mathbf{y}_i - W_i \mathbf{z}_i \\ &= (\lambda_i I_n - D)^{-1} U \mathbf{y}_i - (\lambda_i I_n - D)^{-1} W_i W_i^T U \mathbf{y}_i - W_i \mathbf{z}_i \\ &= (\lambda_i I_n - D)^{-1} U \mathbf{y}_i - (\lambda_i I_n - D)^{-1} W_i \left\{ \mu_i \mathbf{z}_i - W_i^T (\lambda_i I_n - D) W_i \mathbf{z}_i \right\} - W_i \mathbf{z}_i \\ &= (\lambda_i I_n - D)^{-1} (U \mathbf{y}_i - \mu_i W_i \mathbf{z}_i) \end{aligned}$$

よって,

$$\mathbf{q}_i^T \mathbf{q}_j = \frac{\alpha_i \alpha_j}{\lambda_i - \lambda_j} \left\{ (U \mathbf{y}_i - \mu_i W_i \mathbf{z}_i)^T \mathbf{q}'_j - \mathbf{q}_i^T (U \mathbf{y}_j - \mu_j W_j \mathbf{z}_j) \right\}$$

である. 括弧の中の第 1 項は,

$$\begin{aligned} & (U \mathbf{y}_i - \mu_i W_i \mathbf{z}_i)^T \mathbf{q}'_j \\ &= \mathbf{y}_i^T U^T ((\lambda_j I_n - D)^{-1} (I_n - W_j W_j^T) U \mathbf{y}_j - W_j \mathbf{z}_j) - \mu_i \mathbf{z}_i^T W_i^T \mathbf{q}'_j \\ &= \mathbf{y}_i^T C^{-1} \mathbf{y}_j - \mathbf{y}_i^T (\mu_j \mathbf{y}_j - U^T W_j \mathbf{z}_j) - \mathbf{y}_i^T U^T W_j \mathbf{z}_j - \mu_i \mathbf{z}_i^T W_i^T \mathbf{q}'_j \\ &= \mathbf{y}_i^T C^{-1} \mathbf{y}_j - \mu_j \mathbf{y}_i^T \mathbf{y}_j - \mu_i \mathbf{z}_i^T W_i^T \mathbf{q}'_j \end{aligned}$$

となるから, 結局,

$$\mathbf{q}_i^T \mathbf{q}_j = \frac{\alpha_i \alpha_j}{\lambda_i - \lambda_j} \left\{ (\mu_i - \mu_j) \mathbf{y}_i^T \mathbf{y}_j - \mu_i \mathbf{z}_i^T W_i^T \mathbf{q}'_j + \mu_j \mathbf{z}_j^T W_j^T \mathbf{q}'_i \right\}$$

である. また, $W_1 = W_2$ を仮定すれば, $\mathbf{z}_1^T W_1^T \mathbf{q}'_2 = \mathbf{z}_2^T W_2^T \mathbf{q}'_1 = -\mathbf{z}_1^T \mathbf{z}_2$ となるから, 直ちに (A.2) が示される. ■

参考文献

- [1] Anderson, E. et al., LAPACK Users' Guide Third Edition, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1999.
- [2] Arbenz, P. and Golub, G. H., On the spectral decomposition of hermitian matrices modified by low rank perturbations with applications, SIAM J. Matrix Anal. Appl., 9 (1998), 40–58.
- [3] Bai, Z. et al., Templates for the Solution of Algebraic Eigenvalue Problems, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2000.
- [4] Bunch, J. R., Nielsen, C. P. and Sorensen, D. C., Rank-One Modification of the Symmetric Eigenproblem, Numerische Mathematik, 31 (1978), 31–48.
- [5] Cuppen, J. J. M., A divide and conquer method for the symmetric tridiagonal eigenproblem, Numerische Mathematik, 36 (1981), 177–195.
- [6] Demmel, J. W., Applied Numerical Linear Algebra, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1997.
- [7] Dhillon, I. S. and Parlett, B. N., Multiple representations to compute orthogonal eigenvectors of symmetric tridiagonal matrices, Linear Algebra and its Applications, 387 (2004), 1–28.
- [8] Dongarra, J. J. and Sorensen, D. C., A fully parallel algorithm for the symmetric eigenvalue problem, SIAM J. Sci. Stat. Comput., 8 (1987), 139–154.
- [9] Gates, K. and Arbenz, P., Parallel divide and conquer algorithms for the symmetric tridiagonal eigenproblem, Technical report, Institute for Scientific Computing, ETH Zurich, 1994.
- [10] Golub, G. H., Some modified matrix eigenvalue problems, SIAM Rev., 15 (1973), 318–334.
- [11] Gu, M. and Eisenstat, S. C., A stable and efficient algorithm for the rank-one modification of the symmetric eigenproblem, SIAM J. Matrix Anal. Appl., 15 (1994), 1266–1276.
- [12] Gu, M. and Eisenstat, S. C., A divide-and-conquer algorithm for the symmetric tridiagonal eigenproblem, SIAM J. Matrix Anal. Appl., 16 (1995), 172–191.

- [13] 桑島 豊, 重原 孝臣, 実対称三重対角固有値問題の分割統治法の拡張, 日本応用数学会論文誌, 15 (2005), 89–115.
- [14] Li, R.-C., Solving Secular Equations Stably and Efficiently, CS-94-260, 1994.
- [15] Löwner, K., Über monotone Matrixfunktionen, Math. Z., 38 (1934), 177–216.
- [16] Matrix Market, (<http://math.nist.gov/MatrixMarket/>).
- [17] Melman, A., A numerical comparison of methods for solving secular equations, J. Comp. Appl. Math., 86 (1997), 237–249.
- [18] Rutter, J. D., A Serial Implementation of Cuppen's Divide and Conquer Algorithm for the Symmetric Eigenvalue Problem, Computer Science Division Report, University of California, Berkeley.
- [19] Sorensen, D. C. and Tang, P. T. P., On the orthogonality of eigenvectors computed by divide-and-conquer techniques, SIAM J. Numer. Anal., 28 (1991), 1752–1775.
- [20] Tisseur, F. and Dongarra, J., Parallelizing the divide and conquer algorithm for the symmetric tridiagonal eigenvalue problem on distributed memory architectures, SIAM Journal on Scientific Computing, 20 (1999), 2223–2236.

謝辞 丁寧に原稿を読んでいただき, 適切なお指摘をしていただいた査読者に感謝いたします。本研究の一部は科学研究費補助金(課題番号: 16656033)のもとで行われました。

桑島豊(学生会員) 〒338-8570 さいたま市桜区下大久保 255

平成14年埼玉大学工学部情報システム工学科卒業。平成16年より同大学理工学研究科博士後期課程情報数理学専攻に在籍。専門分野は数値線形代数, ハイパフォーマンスコンピューティング。

重原孝臣(正会員) 〒338-8570 さいたま市桜区下大久保 255

昭和63年東京大学大学院理学系研究科物理学専攻修了。平成16年より埼玉大学工学部情報システム工学科教授。理学博士。専門分野は数値線形代数, ハイパフォーマンスコンピューティング, 量子カオス。電子情報通信学会, 情報処理学会, 日本物理学会各会員。

(2006年4月3日 受付)

(2006年11月6日 最終稿受付)