

## 実対称三重対角固有値問題の分割統治法の拡張

桑島 豊\* 重原 孝臣\*\*

\* 埼玉大学理工学研究科情報数理科学専攻

\*\* 埼玉大学工学部情報システム工学科

An Extension of Divide-and-Conquer for Real Symmetric Tridiagonal Eigenproblem

Yutaka Kuwajima\* Takaomi Shigehara\*\*

\* Graduate School of Science and Engineering, Saitama University

\*\* Faculty of Engineering, Saitama University

*Abstract.* Divide-and-conquer (DC) is one of the fastest algorithms for eigenproblem of a large-size symmetric tridiagonal matrix (STM). In the original DC, a STM is supposed to be divided in half. In this paper, we propose an extended DC (EDC) where a STM is divided into  $k$  parts ( $k > 2$ ). Compared to DC, EDC requires only  $3k/(2(k^2 - 1))$  floating operation counts if  $k$  is much smaller than the matrix size. In implementation of EDC, the orthogonality among eigenvectors with nearly multiple eigenvalues is ensured by an appropriate usage of quadruple-precision floating-point number processing. We give a formula for the floating operation counts of the present implementation, whose validity is confirmed by numerical experiment.

## 1 序論

現在、実対称固有値問題の数値解法の中で、Multiple Relatively Robust Representations 法 [6] などと並んで、分割統治法は演算量最小の解法の一つである。固有値問題の分割統治法は Cuppen が発表した解法 [4] で、実対角行列と階数 1 の摂動の和の固有値問題解法 [3], [8] に基づいている。この固有値問題の解法は、それ以前の Golub の研究 [9] に着想を得ている。初期の簡単な実装による分割統治法は、近接固有値に対応する固有ベクトルの直交性が保証されず、数値的に不安定であるという問題があった。この問題の解決策として拡張精度の浮動小数点演算を用いるという方式 [16] も提案された。しかし、現在は拡張精度を用いずに、計算した固有値の近似値を解とする固有値問題を精密に再構成することにより、固有ベクトルの直交性を保証する Gu らの方法 [10], [11] が主流である。これらの研究によって、分割統治法は実用的なアルゴリズムとなり、行列計算ライブラリ LAPACK に実装されている [1], [12], [15]。

本稿では、分割統治法を従来の 2 分割から  $k$  分割へ拡張した解法を提案する。 $k$  分割の分割統治法は、以下の 4 ステップからなる。まず、 $n$  次実対称三重対角行列を、 $k$  個の実対称三重対角行列の直和と階数  $k - 1$  の摂動の和に分割する。次に、分割した  $k$  個の実対

称三重対角行列の固有値問題を解くことで、 $n$  次実対称三重対角行列を、実対角行列と階数  $k-1$  の摂動の和に相似変換する。第三に、実対角行列と階数  $k-1$  の摂動の和の固有値問題を解く。最後に、相似変換前の基底で表すために、固有ベクトルを行列行列積により変換する。以上により、 $n$  次実対称三重対角行列の固有値問題を解くことができる。このアルゴリズムは  $k=2$  に従来法を含み、自然な拡張となっている。

$k$  分割の分割統治法では、実対称三重対角行列の固有値問題を解くために対角行列と階数  $k-1$  の摂動の和の固有値問題を解かなければならない。対角行列と階数  $k-1$  の摂動の和の固有値問題は、 $k-1$  次実対称行列値関数の問題に帰着できることが知られている [2]。固有値は、行列値関数の行列式の零点と等しく、符号数を用いた二分法で求める方法が提案されている [2]。我々はその方法を用いず、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方法をとる。この数値解法は、分割統治法においては安定かつ高速である。2 分割の分割統治法における固有値を求めるアルゴリズムと本質的に等価であるため、精度、演算量ともに分割数に依存せず、有効である。一方、固有ベクトルは、 $k-1$  次実対称行列値関数に、対応する固有値を代入して得られる行列の固有値 0 に対応する固有ベクトルを用いて、高速に求めることができる。しかし、階数 1 の場合と同様に近接固有値に対応する固有ベクトルの直交性が損なわれる可能性がある。そのため、Gu らの方法と 4 倍精度演算を併用することで直交性を確保する。

$k$  分割の分割統治法の最大の長所は、分割数が増えることにより、計算の主要部にあたる固有ベクトルを計算するための行列行列積の演算量が  $2/k$  に減少し、その結果として全体の演算量が約  $3k/(2(k^2-1))$  に減少することである。しかし分割数の増加に伴い、分割した問題の解を統合する際に解くべき問題、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増加する。そのため、適切な分割数についても論じる。

本稿の構成は以下の通り。2 節では、 $k$  分割の分割統治法を提案する準備として、従来の 2 分割の分割統治法の概要と実装方法を述べる。 $k$  分割の分割統治法は、従来の分割統治法の自然な拡張になっているため、対応をつけて述べる。また、対角行列と階数 1 の摂動の和の固有値問題の数値解法も述べる。この解法は、提案法で、対角行列と階数  $k-1$  の摂動の和の固有値問題を解く際にも用いる。3 節では、2 分割を拡張して  $k$  分割の分割統治法を提案する。4 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。5 節はまとめである。

## 2 2 分割の分割統治法

$k$  分割の分割統治法を提案する準備として、従来の 2 分割の分割統治法を述べる。分割統治法の概要を述べた後、アルゴリズム面で中核をなす対角行列と階数 1 の摂動の和の固有値問題の数値解法を述べる。

## 2.1 2 分割の分割統治法の概要

Cuppen は 2 分割の分割統治法を用いて,  $n$  次実対称三重対角行列  $T$  の全ての固有値と固有ベクトルを求める方法を発表した [4]. 以下にその概要を示す.

$T$  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. いま  $T$  を,

$$(2.1) \quad T = \left( \begin{array}{ccc|ccc} a_1 & b_1 & & & & \\ b_1 & \ddots & & & & \\ & \ddots & a_{m-1} & b_{m-1} & & \\ & & b_{m-1} & a_m & & \\ \hline & & & b_m & a_{m+1} & b_{m+1} \\ & & & b_m & b_{m+1} & \ddots \\ & & & & \ddots & \ddots \\ & & & & & a_{n-1} & b_{n-1} \\ & & & & & b_{n-1} & a_n \end{array} \right)$$

$$= \left( \begin{array}{ccc|ccc} a_1 & b_1 & & & & \\ b_1 & \ddots & & & & \\ & \ddots & a_{m-1} & b_{m-1} & & \\ & & b_{m-1} & a_m - |b_m| & & \\ \hline & & & a_{m+1} - |b_m| & b_{m+1} & \\ & & & b_{m+1} & \ddots & \ddots \\ & & & & \ddots & \ddots \\ & & & & & a_{n-1} & b_{n-1} \\ & & & & & b_{n-1} & a_n \end{array} \right) + \left( \begin{array}{ccc|ccc} & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ \hline & & & |b_m| & b_m & \\ & & & b_m & |b_m| & \\ & & & & & \\ & & & & & \\ & & & & & \end{array} \right)$$

と分割すれば,

$$(2.2) \quad T = \left( \begin{array}{c|c} T_1 & \\ \hline & T_2 \end{array} \right) + |b_m| \mathbf{v} \mathbf{v}^T, \quad \mathbf{v} = \begin{cases} (0 \cdots 0 & 1 & | & 1 & 0 \cdots 0)^T, & (b_m > 0) \\ (0 \cdots 0 & 1 & | & -1 & 0 \cdots 0)^T, & (b_m < 0) \end{cases}$$

と書ける. ここで,  $T_1, T_2$  の固有値問題  $T_i = Q_i D_i Q_i^T$  の解 (実対角行列  $D_1, D_2$ , 直交行列  $Q_1, Q_2$ ) が得られているとすると,

$$(2.3) \quad \begin{aligned} T &= \left( \begin{array}{cc} Q_1 D_1 Q_1^T & \\ & Q_2 D_2 Q_2^T \end{array} \right) + |b_m| \mathbf{v} \mathbf{v}^T \\ &= \left( \begin{array}{cc} Q_1 & \\ & Q_2 \end{array} \right) \left\{ \left( \begin{array}{cc} D_1 & \\ & D_2 \end{array} \right) + |b_m| \mathbf{u} \mathbf{u}^T \right\} \left( \begin{array}{cc} Q_1^T & \\ & Q_2^T \end{array} \right) \end{aligned}$$

と書ける. ただし,

$$(2.4) \quad \mathbf{u} = \left( \begin{array}{cc} Q_1^T & \\ & Q_2^T \end{array} \right) \mathbf{v} = \left( \begin{array}{c} Q_1^T \text{の最終列} \\ s Q_2^T \text{の第1列} \end{array} \right), \quad \begin{cases} b_m > 0 \text{ ならば } s = 1 \\ b_m < 0 \text{ ならば } s = -1 \end{cases}$$

である. すなわち,  $T$  の固有値は  $D + |b_m|uu^T$  の固有値と等しく,  $T$  の固有値  $\lambda$  に対応する固有ベクトルは,  $D + |b_m|uu^T$  の  $\lambda$  に対応する固有ベクトル  $q'$  を用いて,  $Qq'$  で与えられる. ただし,  $D = D_1 \oplus D_2$  は実対角行列,  $Q = Q_1 \oplus Q_2$  は直交行列である.

$T_i$  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.

## 2.2 対角行列と階数 1 の摂動の和の固有値問題

分割統治法では, 実対称固有値問題は実対角行列と階数 1 の摂動の和  $D + cuu^T$  に帰着される. ただし,  $D = \text{diag}(d_1, d_2, \dots, d_n)$  は  $n$  次実対角行列,  $d_1 \leq d_2 \leq \dots \leq d_n$ ,  $u = (u_1, u_2, \dots, u_n)^T$  は  $n$  次実ベクトル,  $c$  は正の実数である.

この小節では,  $d_i$  はことごとく異なり, かつ  $u_i \neq 0$  ( $i = 1, 2, \dots, n$ ) を仮定する. この仮定の下では,  $D + cuu^T$  の固有値は  $d_i$  とはならない. この仮定が満たされない場合は, 次小節の deflation という操作により, 仮定を満たすようにできる.

以下では,  $D + cuu^T$  の固有値問題を解く方法を述べる.  $\lambda I_n - D$  が正則であれば,  $D + cuu^T$  の特性多項式は

$$(2.5) \quad \det(\lambda I_n - (D + cuu^T)) = \det(\lambda I_n - D) \det(I_n - c(\lambda I_n - D)^{-1}uu^T)$$

と変形される. ここで,  $\lambda I_n - D$  が正則であることより,  $D + cuu^T$  の固有値  $\lambda$  は,

$$(2.6) \quad \det(I_n - c(\lambda I_n - D)^{-1}uu^T) = 0$$

を満たす. 次の補題を用いると, 左辺の行列式は簡単に計算することができる.

補題 1.  $n \times k$  実行列  $X, Y$  に対して,  $\det(I_n + XY^T) = \det(I_k + Y^T X)$ .

証明 付録参照のこと. ■

この補題により,

$$(2.7) \quad \det(I_n - c(\lambda I_n - D)^{-1}uu^T) = 1 - cu^T(\lambda I_n - D)^{-1}u$$

である. したがって,  $D + cuu^T$  の固有値  $\lambda$  は,

$$(2.8) \quad f(\lambda) \equiv \frac{1}{c} - u^T(\lambda I_n - D)^{-1}u = \frac{1}{c} - \sum_{i=1}^n \frac{u_i^2}{\lambda - d_i} = 0$$

の解である. 区間  $(d_i, d_{i+1})$  において,  $f(\lambda)$  は  $\lambda \rightarrow d_i + 0$  で負に,  $\lambda \rightarrow d_{i+1} - 0$  で正に発散し, 連続かつ単調増加であるので, 必ず唯一つの解を持つ. さらに,  $\lambda > d_n + c\|u\|_2^2$  で  $f(\lambda) > 0$  であるので,  $D + cuu^T$  の固有値  $\lambda_i$  は, インタレース性

$$(2.9) \quad d_1 < \lambda_1 < d_2 < \lambda_2 < d_3 < \dots < d_{n-1} < \lambda_{n-1} < d_n < \lambda_n \leq d_n + c\|u\|_2^2$$

を持つ.

$f(\lambda) = 0$  の一つの解は、後述の変形 Newton 法を用いて  $O(n)$  で、したがって  $D + cuu^T$  の全ての固有値は  $O(n^2)$  の演算量で求めることができる。

一方、 $D + cuu^T$  の固有ベクトルは、次の定理を用いることで簡単に求められる。

定理 1.  $A$  を  $n$  次実対称行列、 $u$  を  $n$  次実ベクトル、 $c$  を非零実数とする。  $n$  次実対称行列  $\hat{A} = A + cuu^T$  の固有値が  $\lambda$  であるとき、 $\lambda I_n - A$  が正則ならば、 $\lambda$  に対応する  $\hat{A}$  の固有ベクトルは  $(\lambda I_n - A)^{-1}u$  である。

証明 付録参照のこと。 ■

この定理で  $A = D$  の場合の主張より、 $D + cuu^T$  の固有ベクトルは、 $(\lambda I_n - D)^{-1}u$  である。  $\lambda I_n - D$  が対角行列であることから逆行列が簡単に計算でき、 $D + cuu^T$  の一つの固有ベクトルは  $O(n)$  の演算量で、全ての固有ベクトルは  $O(n^2)$  の演算量で求められる。

## 2.3 deflation

対角行列と階数 1 の摂動との和  $D + cuu^T$  の固有値問題は、摂動によって変化しない固有値を解に持つとき、その固有値を取り除くことで、より小さい次元の問題に変換することができる [7]。

$u$  の第  $i$  要素が 0 ならば、 $D + cuu^T$  は固有値  $d_i$  を持ち、対応する固有ベクトルは  $e_i$  である。 また、 $D$  が重複する対角要素  $d_i = d_{i+1} = \dots = d_{i+m-1}$  を持つならば、適当な直交行列で相似変換することで、 $D$  を変化させずに  $u$  の第  $i+1$  成分から第  $i+m-1$  成分を 0 にすることができる。 このとき、 $D + cuu^T$  は重複度  $m-1$  の固有値  $d_i$  を持つ。

以上の変換を繰り返すことで、

$$(2.10) \quad PG(D + cuu^T)G^T P^T = PDP^T + c \begin{pmatrix} \tilde{u} \\ 0 \end{pmatrix} (\tilde{u}^T 0^T) = \begin{pmatrix} \tilde{D}_1 + c\tilde{u}\tilde{u}^T & 0 \\ 0 & \tilde{D}_2 \end{pmatrix}$$

とできる。 ただし、 $G$  は相似変換の直交行列の積、 $P$  は左から乗ずることで  $Gu$  の 0 の要素を末尾に移す順列行列、 $\tilde{D}_1$  は重複する対角要素を持たない実対角行列、 $\tilde{D}_2$  は実対角行列で  $\tilde{D}_1 \oplus \tilde{D}_2 = PDP^T$  を満たし、 $\tilde{u}$  は要素に 0 を持たない実ベクトルである。 このとき、 $\tilde{D}_1 + c\tilde{u}\tilde{u}^T$  は前小節の仮定を満たす。

deflation には 2 つの効果がある。 一つは、deflation により得られた固有値を取り除くことで、固有値問題の次元を下げるができるという効果である。 これを、本稿では deflation の効果 A と称し、アルゴリズムを擬似言語で記述する際には、この効果が及ぶ箇所に下線を引く。 我々の実装では、この効果は取り入れる。

もう一つの効果は、任意の行列と  $D + cuu^T$  の固有ベクトルとの積の計算で生じる。 deflation により自明な固有ベクトルが得られたとき、その積も自明となり、演算量が削減できるという効果である。 これを、本稿では deflation の効果 B と称し、アルゴリズムを擬似言語で記述する際には、この効果が及ぶ箇所に下波線を引く。 我々の今回の実装では、この効果は取り入れない。 これについては、結論で改めて議論する。

## 2.4 アルゴリズムとその演算量

この小節では、擬似言語を用いて実対称固有値問題の分割統治法 (2 分割) のアルゴリズムを示し、その演算量を評価する。分割統治法アルゴリズムは、再帰構造を用いると明快である。

アルゴリズム 1 分割統治法 (2 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\_2( $T, Q, \Lambda$ ) ..... 入力  $T$  から  $T = Q\Lambda Q^T$  を満たす直交行列  $Q$ , 対角行列  $\Lambda$  を出力する

if  $T$  が 1 次

**return**  $Q = 1, \Lambda = T$

else

$T = \begin{pmatrix} T_1 & 0 \\ 0 & T_2 \end{pmatrix} + cvv^T$  と分割する.

**call** dc\_eig\_2( $T_1, Q_1, \Lambda_1$ )

**call** dc\_eig\_2( $T_2, Q_2, \Lambda_2$ )

$\Lambda_1, \Lambda_2, Q_1, Q_2$  を用いて  $D + cuu^T$  の形式にする.

deflation をし、 $f(\lambda) = 0$  を解くことで、

$D + cuu^T$  の全固有値  $\lambda_i$  ( $i = 1, 2, \dots, n$ ) を求める.

    定理 1 により、 $D + cuu^T$  の全ての固有ベクトル  $q'_i$  ( $i = 1, 2, \dots, n$ ) を求める.

$Q = \begin{pmatrix} Q_1 & 0 \\ 0 & Q_2 \end{pmatrix} (q'_1 \ q'_2 \ \dots \ q'_n)$  を計算する.

~~~~~  
    **return**  $Q, \Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$

**endif**

このアルゴリズムで、下線は deflation の効果 A が及ぶ箇所、下波線は deflation の効果 B が及ぶ箇所である。

$n$  次行列に対してこのアルゴリズムを適用したときの演算量  $t_2(n)$  は以下になる。まず、2 回の再帰呼出しで  $2t_2(n/2)$  の演算量である。次の、対角行列と階数 1 の摂動の和の固有値問題の演算量は、 $O(n^2)$  である。最後の直交行列  $Q$  を求めるための行列行列積は、被乗行列の要素の半分の 0 のため、その演算量は  $n^3$  である。したがって、 $t_2(n) = 2t_2(n/2) + O(n^2) + n^3$  となる。ここで、 $O(n^2)$  の項を無視すると、

$$(2.11) \quad t_2(n) = 2t_2(n/2) + n^3 = \left(1 + \frac{1}{4} + \frac{1}{4^2} + \dots\right) n^3 \simeq \frac{4}{3} n^3$$

を得る。

## 2.5 $f(\lambda) = 0$ の数値解法

2.2 節では,  $n$  次実対角行列  $D = \text{diag}(d_1, d_2, \dots, d_n)$ ,  $d_1 < d_2 < \dots < d_n$ ,  $n$  次実ベクトル  $\mathbf{u} = (u_1, u_2, \dots, u_n)^T$ , 実数  $c > 0$  が与えられたとき,  $D + c\mathbf{u}\mathbf{u}^T$  の固有値は, 非線型方程式 (2.8) の解であることを述べた. (2.8) の解  $\lambda \in (d_i, d_{i+1})$  は, Newton 法を変形した方法を用いることで, 反復計算により 2 次収束で求めることができる [3]. 概要を以下に述べる.

$\lambda$  の第  $j$  近似を  $\lambda^{(j)}$  とする. 変形 Newton 法では,  $f(\lambda)$  を直線の代わりに,  $\lambda^{(j)}$  で  $f(\lambda)$  と共通の接線を持つ有理関数

$$(2.12) \quad \hat{f}(\lambda) = p - \frac{q}{\lambda - d_i} - \frac{r}{\lambda - d_{i+1}}$$

で近似する. そして, 次の近似値を  $\hat{f}(\lambda) = 0$  の解  $\lambda^{(j+1)} \in (d_i, d_{i+1})$  とする. パラメータの決定にはいくつか方法があるが [14], 我々は, パラメータ  $p = p_1 + p_2, q, r$  を,

$$(2.13) \quad p_1 - \frac{q}{\lambda^{(j)} - d_i} = \psi(\lambda^{(j)}), \quad \frac{q}{(\lambda^{(j)} - d_i)^2} = \psi'(\lambda^{(j)})$$

$$(2.14) \quad p_2 - \frac{r}{\lambda^{(j)} - d_i} = \phi(\lambda^{(j)}), \quad \frac{r}{(\lambda^{(j)} - d_i)^2} = \phi'(\lambda^{(j)})$$

を満たすように決定する方法を採用する. ただし,  $\psi(\lambda)$  と  $\phi(\lambda)$  はそれぞれ  $f(\lambda)$  の負の項の和と正の項の和で

$$(2.15) \quad \psi(\lambda) \equiv -\sum_{l=1}^i \frac{u_l^2}{\lambda - d_l} < 0, \quad \phi(\lambda) \equiv \frac{1}{c} - \sum_{l=i+1}^n \frac{u_l^2}{\lambda - d_l} > 0$$

である. それぞれ, 区間  $(d_i, d_{i+1})$  において適切な初期値  $\lambda^{(0)}$  を決め, このアルゴリズムを繰り返すことで,  $D + c\mathbf{u}\mathbf{u}^T$  の固有値  $\lambda \in (d_i, d_{i+1})$  を 2 次収束で求めることができる.

ただし,  $D + c\mathbf{u}\mathbf{u}^T$  の最大固有値  $\lambda_n \in (d_n, d_n + c\|\mathbf{u}\|_2^2]$  は例外として扱う. この場合は  $f(\lambda)$  を

$$(2.16) \quad \hat{f}(\lambda) = p - \frac{q}{\lambda - d_{n-1}} - \frac{u_n^2}{\lambda - d_n}$$

で近似し,  $\hat{f}(\lambda) = 0$  の解  $\lambda^{(j+1)} \in (d_n, d_n + c\|\mathbf{u}\|_2^2]$  を次の近似解とする. ただし,

$$(2.17) \quad p_1 - \frac{q}{\lambda^{(j)} - d_{n-1}} = \psi(\lambda^{(j)}), \quad \frac{q}{(\lambda^{(j)} - d_{n-1})^2} = \psi'(\lambda^{(j)}), \quad \psi(\lambda) \equiv -\sum_{l=1}^{n-1} \frac{u_l^2}{\lambda - d_l}$$

を満たすようにパラメータ  $p = c^{-1} + p_1, q$  を決定する. このアルゴリズムでは, 初期値  $\lambda^{(0)}$  は真の値より大きくなければならず, そのとき反復により, 右から 2 次収束する.

$u_l/(\lambda - d_l)$  を先に計算しておくことで,  $f(\lambda), f'(\lambda)$  を  $6n$  の演算量で求められる. いずれの固有値に対しても, 以上の変形 Newton 法は平均で 5 回程度で収束するため,  $D + c\mathbf{u}\mathbf{u}^T$  の固有値 1 つにつき平均で  $30n$ , 全ての固有値を平均で  $30n^2$  の演算量, すなわち  $O(n^2)$  で求めることができる.

## 2.6 固有ベクトルの直交性保証手法

定理 1 により対角行列と階数 1 の摂動の和  $D + \mathbf{u}\mathbf{u}^T$  の固有値が  $\lambda$  であるとき、対応する固有ベクトルは  $(\lambda I - D)^{-1}\mathbf{u}$  で与えられる。しかし、この公式では近接固有値同士の固有ベクトルの数値的直交性が保証できない。この問題を解決するために、次の定理を用いる。

定理 2. (Löwner) [13]  $n$  次実対角行列  $D = \text{diag}(d_1, d_2, \dots, d_n)$  と、実数  $\lambda_j$  ( $j = 1, 2, \dots, n$ ) が与えられて、インタレース性  $d_1 < \lambda_1 < d_2 < \lambda_2 < \dots < \lambda_{n-1} < d_n < \lambda_n$  を満たすとする。  $D + \tilde{\mathbf{u}}\tilde{\mathbf{u}}^T$  の固有値が  $\lambda_i$  であるとき、  $\tilde{\mathbf{u}}$  の第  $j$  要素  $\tilde{u}_j$  は

$$(2.18) \quad |\tilde{u}_j| = \sqrt{(\lambda_j - d_j) \prod_{\substack{k=1 \\ k \neq j}}^n \frac{\lambda_k - d_j}{d_k - d_j}}, \quad (j = 1, 2, \dots, n)$$

で与えられる。

$D + \mathbf{u}\mathbf{u}^T$  の固有値の近似値を  $\lambda_i$  ( $i = 1, 2, \dots, n$ ) としてこの定理に適用すると、  $\lambda_i$  全てを高精度な固有値の近似値とする行列  $D + \tilde{\mathbf{u}}\tilde{\mathbf{u}}^T$  を構成できる。ただし、通常  $\tilde{u}_j$  の符号は  $u_j$  と同符号にとる。  $\lambda_k$  が高精度な固有値のため、  $\tilde{\mathbf{u}}$  を替わりに定理 1 に適用した

$$(2.19) \quad \frac{(\lambda_k I_n - D)^{-1}\tilde{\mathbf{u}}}{\|(\lambda_k I_n - D)^{-1}\tilde{\mathbf{u}}\|_2}$$

は、  $\lambda_k$  に対応する高精度の固有ベクトルであり、数値的直交性が保証される [10]。

Löwner の定理では、  $4n^2$  の演算量で  $\tilde{\mathbf{u}}$  を計算できる。(2.19) では、Löwner の定理で  $\tilde{\mathbf{u}}$  を再定義する際に  $\lambda_i - d_j$  を保存しておけば、1 つの正規固有ベクトルは  $4n$  の演算量で計算できる。したがって、  $D + \mathbf{c}\mathbf{u}\mathbf{u}^T$  の全ての固有ベクトルは  $8n^2$  の演算量で計算できる。

したがって、2.5 節と 2.6 節で述べたアルゴリズムを用いると、  $n$  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、2 分割の分割統治法の演算量は、

$$(2.20) \quad \begin{aligned} t_2(n) &= 2t_2(n/2) + n^3 + 38n^2 = \left(1 + \frac{1}{4} + \frac{1}{4^2} + \dots\right)n^3 + \left(1 + \frac{1}{2} + \frac{1}{2^2} + \dots\right)38n^2 \\ &\simeq \frac{4}{3}n^3 + 2 \cdot 38n^2 \end{aligned}$$

となる。

## 3 $k$ 分割の分割統治法の提案

この節では、従来の分割統治法を  $k$  分割 ( $k > 2$ ) に拡張したアルゴリズムを提案する。このアルゴリズムは  $k = 2$  に従来の分割統治法を含んでおり、自然な拡張となっている。

以下では当分の間、固有値問題の次元  $n$  は  $k$  と比較して十分大きいと仮定する。この節の最後で最適な分割数について論じ、実際に  $k \ll n$  であることを述べる。



### 3.1 $k$ 分割の分割統治法の概要

いま 実対称三重対角行列  $T$  を, 2 分割の分割統治法と同様の手法で

$$(3.1) \quad T = \left( \begin{array}{c|c|c|c} T_1 & & & \\ \hline & T_2 & & \\ \hline & & \ddots & \\ \hline & & & T_k \end{array} \right) + |b'_1| \mathbf{v}_1 \mathbf{v}_1^T + \cdots + |b'_{k-1}| \mathbf{v}_{k-1} \mathbf{v}_{k-1}^T$$

と,  $k$  個の実対称三重対角行列  $T_i$  の直和と階数  $k-1$  の半正定値対称な摂動の和に分割する. ただし,  $\mathbf{v}_1 = (0 \cdots 0 \ 1 \mid s_1 \ 0 \cdots 0 \mid 0 \cdots 0 \mid \cdots)^T$ ,  $\mathbf{v}_2 = (0 \cdots 0 \mid 0 \cdots 0 \ 1 \mid s_2 \cdots 0 \mid \cdots)^T$ ,  $\cdots$ ,  $\mathbf{v}_{k-1} = (\cdots \mid 0 \cdots 0 \mid 0 \cdots 0 \ 1 \mid s_{k-1} \ 0 \cdots 0)^T$ . ここで,  $b'_i > 0$  ならば  $s_i = 1$ ,  $b'_i < 0$  ならば  $s_i = -1$  である.

$T_i$  をそれぞれ適当な直交行列  $Q_i$  を用いて,

$$(3.2) \quad Q_i^T T_i Q_i = D_i, \quad (i = 1, 2, \cdots, k-1)$$

と対角化する. これにより  $T$  は,

$$(3.3) \quad \begin{aligned} T &= \left( \begin{array}{ccc} Q_1 D_1 Q_1^T & & \\ & \ddots & \\ & & Q_k D_k Q_k^T \end{array} \right) + |b'_1| \mathbf{v}_1 \mathbf{v}_1^T + \cdots + |b'_{k-1}| \mathbf{v}_{k-1} \mathbf{v}_{k-1}^T \\ &= Q(D + |b'_1| \mathbf{u}_1 \mathbf{u}_1^T + \cdots + |b'_{k-1}| \mathbf{u}_{k-1} \mathbf{u}_{k-1}^T) Q^T \\ &\equiv Q(D + UCU^T) Q^T \end{aligned}$$

と書ける. ただし,  $D = (D_1 \oplus \cdots \oplus D_k)$  は実対角行列,  $Q = (Q_1 \oplus \cdots \oplus Q_k)$  は直交行列,  $C = \text{diag}(|b'_1|, \cdots, |b'_{k-1}|)$  は対角行列,  $U = (\mathbf{u}_1 \cdots \mathbf{u}_{k-1})$  は  $n \times (k-1)$  行列,

$$(3.4) \quad \mathbf{u}_i = Q^T \mathbf{v}_i = \begin{pmatrix} 0 \\ Q_i^T \text{の最終列} \\ s_i Q_{i+1}^T \text{の第1列} \\ 0 \end{pmatrix}, \quad \begin{cases} b'_i > 0 \text{ ならば } s_i = 1 \\ b'_i < 0 \text{ ならば } s_i = -1 \end{cases}$$

である.

すなわち,  $T$  の固有値は  $D + UCU^T$  の固有値と等しく,  $T$  の固有値  $\lambda$  に対応する固有ベクトルは,  $D + UCU^T$  の  $\lambda$  に対応する固有ベクトル  $\mathbf{q}'$  を用いて,  $Q\mathbf{q}'$  で与えられる.

$T_i$  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.

### 3.2 対角行列と階数 $k-1$ の摂動の和の固有値問題

$k$  分割に拡張した分割統治法では, 実対称行列の固有値問題は対角行列と階数  $k-1$  の摂動の和  $D + UCU^T$  の固有値問題に帰着される.

以下では,  $D + UCU^T$  の固有値問題を解く方法を述べる [2].

$\lambda I_n - D$  が正則であれば,  $D + UCU^T$  の特性多項式は

$$(3.5) \quad \det(\lambda I_n - (D + UCU^T)) = \det(\lambda I_n - D) \det(I_n - (\lambda I_n - D)^{-1} UCU^T)$$

と変形される. ここで,  $\lambda I_n - D$  が正則であることより,  $D + UCU^T$  の固有値  $\lambda$  は,

$$(3.6) \quad \det(I_n - (\lambda I_n - D)^{-1} UCU^T) = 0$$

を満たす. 補題 1 より左辺は,

$$(3.7) \quad \det(I_n - (\lambda I_n - D)^{-1} UCU^T) = \det(I_{k-1} - CU^T(\lambda I_n - D)^{-1} U)$$

と変形できる. したがって,  $k-1$  次実対称行列  $F(\lambda)$  を

$$(3.8) \quad F(\lambda) \equiv C^{-1} - U^T(\lambda I_n - D)^{-1} U$$

と定義すれば,  $D + UCU^T$  の固有値  $\lambda$  は,

$$(3.9) \quad \det(F(\lambda)) = 0$$

の解である.

一方,  $D + UCU^T$  の固有ベクトルは, 定理 1 を拡張した次の定理を用いることで簡単に求められる.

定理 3. [2]  $A$  を  $n$  次実対称行列,  $U$  を  $n \times k$  次行列,  $C$  を  $k$  次正則実対称行列とする.  $n$  次実対称行列  $\hat{A} = A + UCU^T$  の固有値が  $\lambda$  であるとき,  $\lambda I_n - A$  が正則ならば,  $k$  次実対称行列  $F = C^{-1} - U^T(\lambda I_n - A)^{-1} U$  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  $q$  を用いて,  $\lambda$  に対応する  $\hat{A}$  の固有ベクトルは  $(\lambda I_n - A)^{-1} Uq$  と表せる.

証明 付録参照のこと. ■

この定理で  $A = D$  の場合の主張より,  $D + UCU^T$  の固有ベクトルは,  $F(\lambda)$  の固有値 0 に対応する固有ベクトル  $x$  を用いて  $(\lambda I_n - D)^{-1} Ux$  である.  $\lambda I_n - D$  が対角行列であることから逆行列が簡単に計算でき,  $D + UCU^T$  の一つの固有ベクトルは  $O(n)$  の演算量で, 全ての固有ベクトルは  $O(n^2)$  の演算量で求められる.

### 3.3 提案アルゴリズムとその演算量

2 分割の分割統治法と同様に, 擬似言語を用いて  $k$  分割の分割統治法を記述し, その演算量を評価する.

アルゴリズム 2 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( $T, Q, \Lambda$ ) ..... 入力  $T$  から  $T = Q\Lambda Q^T$  を満たす直交行列  $Q$ , 対角行列  $\Lambda$  を出力する

if  $T$  が  $k-1$  次以下

他のアルゴリズムで  $T$  の固有値問題  $T = Q\Lambda Q^T$  を解く.

return  $Q, \Lambda$

else

$T = (T_1 \oplus \cdots \oplus T_k) + c_1 v_1 v_1^T + \cdots + c_{k-1} v_{k-1} v_{k-1}^T$  と分割する.

call dc\_eig\_k( $T_i, Q_i, \Lambda_i$ ) ( $i = 1, 2, \dots, k$ )

$\Lambda_i, Q_i$  を用いて  $D + c_1 u_1 u_1^T + \cdots + c_{k-1} u_{k-1} u_{k-1}^T$  の形式にする.

固有値問題  $D + c_1 u_1 u_1^T + \cdots + c_{k-1} u_{k-1} u_{k-1}^T = Q' \Lambda Q'^T$  を解く.

$Q = (Q_1 \oplus \cdots \oplus Q_k) Q'$  を計算する.

return  $Q, \Lambda$

endif

このアルゴリズムで, 下線は deflation の効果 A が及ぶ箇所, 下波線は deflation の効果 B が及ぶ箇所である.

$n$  次行列に対してこのアルゴリズムを適用したときの演算量  $t_k(n)$  は以下になる. まず,  $k$  回の再帰呼出しで  $kt_k(n/k)$  の演算量である. 次の, 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, 後述のように  $O(n^2)$  である. 最後の直交行列  $Q$  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,  $t_k(n) = kt_k(n/k) + O(n^2) + (2/k)n^3$  となる. ここで, 演算量が  $O(n^2)$  の項を無視すると,

$$(3.10) \quad t_k(n) = kt_k(n/k) + \frac{2}{k}n^3 = \frac{2}{k} \left( 1 + \frac{1}{k^2} + \frac{1}{k^4} + \cdots \right) n^3 \simeq \frac{2k}{k^2 - 1} n^3$$

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.

### 3.4 分割統治法での対角行列と階数 $k-1$ の摂動の和の固有値問題の数値解法

3.2 節では, (3.9) を解くことで, 対角行列と階数  $k-1$  の摂動の和  $B = D + c_1 u_1 u_1^T + \cdots + c_{k-1} u_{k-1} u_{k-1}^T$  の固有値を求める手法を述べた. 別の手法として, 対角行列と階数 1 の摂動の和の固有値問題を逐次的に  $k-1$  回解く方法が考えられる. 擬似言語で記述すると, 以下の逐次アルゴリズムとなる.

$D_1 = D$  とおく.

do  $i := 1$  to  $k-1$

$z_i := Q_i^T \cdots Q_3^T Q_2^T u_i$  を計算する.

固有値問題  $D_i + c_i z_i z_i^T = Q_{i+1} D_{i+1} Q_{i+1}^T$  を解く.

end do

このアルゴリズムで,  $D_k$  の対角成分が  $B$  の固有値である.

この手法には, 3 個の利点がある. 第一に, 摂動を加えても変化しない固有値が存在しても, 例外処理が必要でない点である. 3.2 節の方法で全固有値を求められる問題は, 摂動が加わったときに全ての固有値が変化する問題のみである. 第二に, 従来の対角行列と階数 1 の摂動の和の固有値問題解法が活用できる点である. この問題は, 2 分割の分割統治法の核となるアルゴリズムでもあるので, 2.5 節で述べたような安定で高速な解法が知られている. 第三の利点は, 分割統治法で解くべき問題の特性を効率よく扱える点である. 以下では, 最後の点について詳しく述べる. この小節では簡単のために,  $n$  は  $k$  の倍数であり,  $m = n/k$  とする. ただし,  $n$  が  $k$  で割り切れない場合でも同様の議論は成り立つ.

$n$  次固有値問題の  $k$  分割の分割統治法に現れる階数  $k-1$  の摂動は

$$(3.11) \quad u_1 = \begin{pmatrix} l_1 \\ f_1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}, u_2 = \begin{pmatrix} 0 \\ l_2 \\ f_2 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}, u_3 = \begin{pmatrix} 0 \\ 0 \\ l_3 \\ f_3 \\ \vdots \\ 0 \\ 0 \end{pmatrix}, u_4 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ l_4 \\ f_4 \\ \vdots \\ 0 \\ 0 \end{pmatrix}, \dots, u_{k-1} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ l_{k-1} \\ f_{k-1} \end{pmatrix}$$

を用いて,  $\sum_{i=1}^{k-1} c_i u_i u_i^T$  である. ただし,  $c_i$  は実数,  $l_i, f_i$  ( $i = 1, 2, \dots, k-1$ ) は  $m$  次ベクトル,  $0$  は  $m$  次元の零ベクトルである.

摂動の成分が (3.11) の構造を有している場合には, 逐次アルゴリズムの直接的な適用ではなく, 摂動項を加える順番を変えることで演算量を削減することができる. 添え字の番号順に加えた場合は, 上記のアルゴリズムで  $z_i$  の非零要素は  $(i+1)m$  個に増加し, (3.11) の特殊な構造を有効に活用することはできない. しかし, 対角行列に奇数番目のみの摂動を加えた行列  $D' = D + u_1 u_1^T + u_3 u_3^T + \cdots + u_p u_p^T$  は, 対角行列と階数 1 の摂動の和の小行列  $(p+1)/2$  個の直和となる. ただし,  $p$  は  $k-1$  以下の最大奇数. すなわち, これらの小行列の固有値問題は独立であり, かつ非零要素の増加もないため, 添え字の番号順に加えた場合と比較して, 演算量が削減される. この方法を, 効率よく行うための方法を以下に述べる.

一般に、ある番号  $l$  に対応する摂動を除いた

$$(3.12) \quad \hat{D} = D + \sum_{i=1, i \neq l}^{k-1} c_i \mathbf{u}_i \mathbf{u}_i^T$$

を考える. この行列は 2 つの行列の直和であり, この行列の固有値問題は 2 つの独立な固有値問題に分割できる. この分割した固有値問題が解けると,  $\hat{D}$  の対角化  $\hat{Q}^T \hat{D} \hat{Q} = \hat{\Lambda}$  が得られる. すると,  $D + \sum_{i=1}^{k-1} c_i \mathbf{u}_i \mathbf{u}_i^T$  の固有値問題は,  $\hat{\Lambda} + c_l \mathbf{w} \mathbf{w}^T$  の固有値問題に帰着される. ただし,  $\mathbf{w} = \hat{Q}^T \mathbf{u}_l$  である.

実際は,  $\hat{D}$  を分割したときの 2 つの固有値問題は, (3.11) で示される特別な構造を有しており, このアルゴリズムの再帰呼び出しにより解くことができる. このアルゴリズムの原理をまず以下に示す. その後でこのアルゴリズムを繰り返しによる解法に書き直した, 実装のためのアルゴリズムを示す.

**アルゴリズム 3** 再帰構造を用いて, (3.11) の構造を持つ階数  $k-1$  の摂動と対角行列との和  $D + UCU^T$  の固有値を求める:

**proc** eig\_lowrank\_rec( $k, D, U, C, Q, \Lambda, \tilde{U}$ ) .....  $k$  分割の分割統治法から生じた  $D, U, C$  を入力とし,  $D + UCU^T = Q\Lambda Q^T$  を満たす 4 倍精度の直交行列  $Q$ , 対角行列  $\Lambda$ , 各ステップにおいて Löwner の定理で再定義した 4 倍精度の摂動  $\tilde{U}$  を出力する.

**if**  $k = 2$  のとき

固有値問題  $D + UCU^T = Q\Lambda Q^T$  を解く.

その際に Löwner の定理で  $\tilde{U}$  を再定義する. (4 倍精度演算)

**return**  $Q, \Lambda, \tilde{U}$

**else**

$l = \lfloor \frac{k}{2} \rfloor$  とする.

$D + UCU^T = \begin{pmatrix} D_1 + U_1 C_1 U_1^T & 0 \\ 0 & D_2 + U_2 C_2 U_2^T \end{pmatrix} + c_l \mathbf{u}_l \mathbf{u}_l^T$  と分割する.

**call** eig\_lowrank\_rec( $l, D_1, U_1, C_1, Q_1, \Lambda_1, \tilde{U}_1$ )

**call** eig\_lowrank\_rec( $k-l, D_2, U_2, C_2, Q_2, \Lambda_2, \tilde{U}_2$ )

$\hat{Q} = Q_1 \oplus Q_2$  とする. (4 倍精度)

$\mathbf{w} = \hat{Q}^T \mathbf{u}_l$  を計算する. (4 倍精度演算)

固有値問題  $(\Lambda_1 \oplus \Lambda_2) + c_l \mathbf{w} \mathbf{w}^T = Q' \Lambda Q'^T$  を解く.

その際に Löwner の定理で  $\tilde{\mathbf{w}}$  を再定義する. (4 倍精度演算)

$\tilde{\mathbf{u}}_l = \hat{Q} \tilde{\mathbf{w}}$  を計算する. (4 倍精度演算)

$\tilde{U} = \begin{pmatrix} \tilde{U}_1 & 0 \\ 0 & \tilde{\mathbf{u}}_l & \tilde{U}_2 \end{pmatrix}$  とする.

$Q = (Q_1 \oplus Q_2) Q'$  を計算する. (4 倍精度演算)

**return**  $Q, \Lambda, \tilde{U}$

**endif**

このアルゴリズムで 4 倍精度演算を行っているのは、後述の固有ベクトル計算で 4 倍精度の  $\tilde{U}$  を用いるためである。固有値のみが必要な場合は 4 倍精度演算を用いる必要はない。なお、 $\tilde{U}$  を 4 倍精度で求めるためには、 $\tilde{w}$ ,  $\hat{Q}$ ,  $\tilde{u}_i$  の順に 4 倍精度で求める必要がある。

アルゴリズム 3 で、出力  $Q$  は必要ない。また、アルゴリズム中、 $Q$  はベクトルとの積としてのみ現れるので、行列行列積  $Q = (Q_1 \oplus Q_2)Q'$  の演算は無駄である。そのため、この行列行列積を明示的に省くことのできるように、アルゴリズム 3' に書き換える。繰り返し構造に基づいているため、演算量が厳密に評価できる。

以下では簡単のために、 $k$  は 2 のべきであり、 $l = \log_2 k$ , かつ  $n$  は  $k$  の倍数で、 $m = n/k$  とする。これらの仮定を外れても、同様のアルゴリズムを適用でき、同様の議論が可能である。

アルゴリズム 3' 繰り返し構造を用いて、対角行列と (3.11) の構造を持つ階数  $k-1$  の摂動の和  $D + UCU^T$  の固有値を求める:

**proc** eval\_lowrank( $D, U, C, \Lambda, \tilde{U}$ ) ..... 入力  $D, U, C$  から  $D + UCU^T$  の固有値  $\lambda_i$  を対角成分に持つ対角行列  $\Lambda = \text{diag}(\lambda_i)$ ,  $\lambda_i$  が「正確に」 $D + \tilde{U}\tilde{U}^T$  の固有値となる 4 倍精度の  $\tilde{U}$  を出力する。

```

do  $j := 1$  to  $k-1$ 
   $D_j^{(0)} := D_j$ ,  $l_j^{(0)} := l_j$ ,  $f_j^{(0)} := f_j$ 
end do
do  $t := 1$  to  $l$ 
  do  $j := 1$  to  $2^{l-t}$ 
     $p := 2^{t-1}(2j-1)$ 
    do  $i := 1$  to  $t-1$ 
       $l_p^{(i)} := Q_{p/2^i}^{(i)T} \begin{pmatrix} 0 \\ l_p^{(i-1)} \end{pmatrix}$ ,  $f_p^{(i)} := Q_{p/2^i+1}^{(i)T} \begin{pmatrix} f_p^{(i-1)} \\ 0 \end{pmatrix}$  (4 倍精度演算)
    end do
     $\Delta_j^{(t)} = \begin{pmatrix} D_{2j-1}^{(t-1)} & 0 \\ 0 & D_{2j}^{(t-1)} \end{pmatrix}$ ,  $z = \begin{pmatrix} l_p^{(t-1)} \\ f_p^{(t-1)} \end{pmatrix}$  とおく. (倍精度)
    2 節で述べた解法で、対角行列と階数 1 の摂動の和
     $\Delta_j^{(t)} + c_p z z^T$  の固有値  $\mu_i$  ( $i = 1, 2, \dots, 2^t m$ ) を求める. (倍精度演算)
     $D_j^{(t)} = \text{diag}(\mu_1, \mu_2, \dots, \mu_{2^t m})$  とおく.
     $\Delta_j^{(t)} + \tilde{z} \tilde{z}^T$  の固有値が「正確に」 $\mu_i$  となるように,
    Löwner の定理で  $\tilde{z}$  を再定義する. (4 倍精度演算)
     $q'_i := \frac{\mu_i I - \Delta_j^{(t)-1} \tilde{z}}{\|(\mu_i I - \Delta_j^{(t)})^{-1} \tilde{z}\|_2}$  ( $i = 1, 2, \dots, 2^t m$ ) (4 倍精度演算)
     $Q_j^{(t)} := (q'_1 \ q'_2 \ \dots \ q'_{2^t m})$ 
  end do
end do

```

```

 $\tilde{\mathbf{u}}_{p,t}^{(t)} := \tilde{\mathbf{z}}$ 
do  $i := t - 1$  downto 1
   $r := 2^{t-i}$ 
   $\tilde{\mathbf{u}}_{p,i}^{(t)} := \left( Q_{r(j-1)+1}^{(i)} \oplus \cdots \oplus Q_{rj}^{(i)} \right) \tilde{\mathbf{u}}_{p,i+1}^{(t)}$  (4 倍精度演算)
end do
 $\tilde{\mathbf{u}}_p := \begin{pmatrix} 0^T & \tilde{\mathbf{u}}_{p,1}^{(t)T} & 0^T \end{pmatrix}^T$ 
end do
end do
return  $\Lambda := D_1^{(l)}, \tilde{U} := (\tilde{\mathbf{u}}_1 \ \tilde{\mathbf{u}}_2 \ \cdots \ \tilde{\mathbf{u}}_{k-1})$ 

```

アルゴリズム 3' で、行列やベクトルの上付きの添え字は、計算の進行度を示すだけでなく次元も表している。すなわち、 $D_j^{(i)}$  は  $2^i m$  次行列、 $l_j^{(i)}$  は  $2^i m$  次ベクトルなどである。

ここで、アルゴリズム 3' の総演算量を評価する。無視できない演算量がかかる計算は以下の 4 つがある。

1.  $l_p^{(i)}, f_p^{(i)}$  の計算

$l_p^{(i)}$  と  $f_p^{(i)}$  の演算量は等しいため、 $l_p^{(i)}$  のみ評価する。 $l_p^{(i)}$  1 回の演算量は、行列に掛けるベクトルの上半分が 0 のため、 $2 \cdot 2^i m \cdot 2^{i-1} m = 2^{2i} m^2$  である。よってアルゴリズム 3' で求める  $l_p^{(i)}$  の演算量の総和は、

$$\begin{aligned}
 m^2 \sum_{t=2}^l \sum_{j=1}^{2^{l-t}} \sum_{i=1}^{t-1} 2^{2i} &= m^2 \sum_{t=2}^l \sum_{j=1}^{2^{l-t}} \frac{4}{3} (2^{2t-2} - 1) \\
 &= \frac{4}{3} m^2 \sum_{t=2}^l 2^{l-t} (2^{2t-2} - 1) \\
 &= \frac{4}{3} m^2 \left[ 2^l (2^0 + 2^1 + \cdots + 2^{l-2}) - (2^{l-2} + 2^{l-1} + \cdots + 2^0) \right] \\
 &= \frac{4}{3} m^2 (2^l - 1)(2^{l-1} - 1) \\
 &= \frac{2}{3} \frac{(k-1)(k-2)}{k^2} n^2
 \end{aligned}$$

となる。ただし、 $t=1$  のときは計算しないため、省いている。

2.  $\Delta_j^{(t)} + c_p z z^T$  の全固有値の計算

ある  $t$  での  $2^t m$  次行列  $\Delta_j^{(t)} + c_p z z^T$  の全固有値を求めるための演算量は、 $p$  次対角行列と階数 1 の摂動の和の固有値を求めるための演算量を  $cp^2$  とすると、 $c(2^t m)^2 = cm^2 2^{2t}$  である。ここで、2 節で述べたように、我々の実装では  $c = 30$  程度である。したがって、アルゴリズム 3' での  $\Delta + c_p z z^T$  の全固有値を求めるための演算量の総和は、

$$cm^2 \sum_{t=1}^l \sum_{j=1}^{2^{l-t}} 2^{2t} = cm^2 \sum_{t=1}^l 2^{l-t} 2^{2t} = cm^2 2^{l+1} (2^l - 1) = 2c \frac{k-1}{k} n^2$$

である。

3. Löwner の定理による  $z$  の再定義と, 定理 1 による固有ベクトル  $q'_i$  の計算

これらの計算は, 2.6 節での評価により,  $8cm^22^{2t}$  の演算量がかかる. したがって, アルゴリズム 3' で求める  $z$  と  $q'_i$  のための演算量の総和は,

$$8m^2 \sum_{t=1}^l \sum_{j=1}^{2^{l-t}} 2^{2t} = 8m^2 \sum_{t=1}^l 2^{l-t} 2^{2t} = 8m^2 2^{l+1} (2^l - 1) = 16 \frac{k-1}{k} n^2$$

になる.

4.  $\tilde{u}_{p,i}^{(t)}$  の計算

$\tilde{u}_{p,i}^{(t)}$  は,  $2^{l-i}$  個の  $2^i m$  次直交行列の直和と  $2^t m$  次ベクトルとの積からなるので, その演算量は  $2(2^i m)^2 2^{t-i} = 2^{i+t+1} m^2$  である. よって, 演算量の総和は

$$\begin{aligned} m^2 \sum_{t=2}^l \sum_{j=1}^{2^{l-t}} \sum_{i=1}^{t-1} 2^{i+t+1} &= m^2 \sum_{t=2}^l \sum_{j=1}^{2^{l-t}} 2^{t+2} (2^{t-1} - 1) \\ &= m^2 \sum_{t=2}^l 2^{l-t} 2^{t+2} (2^{t-1} - 1) \\ &= 2^{l+2} (2^l - l - 1) m^2 \\ &= 4 \left( 1 - \frac{\log_2 k + 1}{k} \right) n^2 \end{aligned}$$

となる. ただし,  $t=1$  のときは計算しないため, 省いている.

以上の 4 種の計算のうち, 1, 3, 4 は 4 倍精度演算を行う. ここで, 4 倍精度演算は倍精度演算の  $d$  倍の演算時間がかかるとすると, アルゴリズム 3' の総演算量は,

$$\begin{aligned} &\frac{4}{3} d \frac{(k-1)(k-2)}{k^2} n^2 + 2c \frac{k-1}{k} n^2 + 16d \frac{k-1}{k} n^2 + 4d \left( 1 - \frac{\log_2 k + 1}{k} \right) n^2 \\ &= \frac{2(k-1)}{k} \left( c + 2d \left( \frac{1}{3} \frac{k-2}{k} + 4 + \frac{k - (\log_2 k + 1)}{k-1} \right) \right) n^2 \\ (3.13) \quad &= \frac{2(k-1)}{k} \left( c + 2d \left( \frac{16}{3} - \frac{2}{3k} - \frac{\log_2 k}{k-1} \right) \right) n^2 \end{aligned}$$

であり,  $n$  に関して  $O(n^2)$  である.

$k$  分割の分割統治法の一部としてアルゴリズム 3' を用いた場合, 固有値を求める部分は 2 分割の分割統治法とほとんど同じである. 異なるのは, 下の階層の答えを結合する順番のみである. 演算量の削減のためには, 摂動を加える順序が重要であり, 常に中央の摂動から加えると最も演算量が削減できる. なお,  $k$  が 2 のべき乗であるときは, 両者はアルゴリズムとして完全に等価である.

一方,  $D + \tilde{U}\tilde{U}^T$  の固有ベクトルは, 定理 3 で求められる. しかし, 単に適用しただけでは, 直交性が得られない. そのため, Löwner の定理と適切な箇所で 4 倍精度で演算することで直交性を得る.



アルゴリズム 4 対角行列と階数  $k-1$  の摂動の和  $D + \tilde{U}\tilde{U}^T$  の固有ベクトルを求める:

**proc** `vec_lowrank`( $D, \tilde{U}, \Lambda, Q$ )  $\cdots \cdots$  入力  $D, \tilde{U}, \Lambda$  から  $D + \tilde{U}\tilde{U}^T = Q\Lambda Q^T$  を満たす直交行列  $Q$  を出力する.

**do**  $i = 1$  **to**  $n$

$F_i := I_{k-1} - \tilde{U}^T(\lambda_i I_n - D)^{-1}\tilde{U}$  を計算する. (4 倍精度演算)

$F_i$  の絶対値最小の固有値に対応する固有ベクトル  $p_i$  を計算する. (4 倍精度演算)

$q_i := (\lambda_i I_n - D)^{-1}\tilde{U}p_i / \|(\lambda_i I_n - D)^{-1}\tilde{U}p_i\|_2$  を計算する. (倍精度演算)

**end do**

**return**  $Q = (q_1 \ q_2 \ \cdots \ q_n)$

$F(\lambda)$  は  $U$  の構造から三重対角行列になるが,  $\tilde{F}(\lambda)$  は Löwner の定理で再定義した摂動  $\tilde{U}$  を用いるので, 三重対角行列にはならない.  $k-1$  次実対称行列  $F_i$  は, 一つの要素が  $4n$  の演算量で求められ,  $k-1$  次対称であるから  $(k-1)(k-2)/2$  個の要素を計算すればよいので,  $2(k-1)(k-2)n$  の演算量で計算できる.  $k-1$  次実ベクトル  $p_i$  は  $ek^3$  の演算量で求められる. ここで,  $e$  は逆反復法の反復回数により決まり, 平均して 5 程度である. 最後に  $q_i$  は,  $2n(k-1) + 5n = (2k+3)n$  の演算量で計算できる.

$F_i$  と  $p_i$  は 4 倍精度で計算するので,  $D + UCU^T$  の固有ベクトル 1 つにつき  $2d(k-1)(k-2)n + dek^3 + (2k+3)n$  の演算量, したがって, 全ての固有ベクトルは

$$(3.14) \quad \begin{aligned} & (2d(k-1)(k-2)n + dek^3 + (2k+3)n)n \\ &= (2d(k-1)(k-2) + (2k+3))n^2 + dek^3n \end{aligned}$$

の演算量である. これは  $O(n^2)$  である.

アルゴリズム 3 (3') と 4 で 4 倍精度演算を用いるのは, 固有ベクトル  $q_i$  を倍精度で求めるためである.  $q_i$  を倍精度で求めるためには,  $p_i$  を倍精度で求める必要があり,  $p_i$  は  $F_i$  から計算される. 特に近接固有値を持つ場合,  $F_i$  の対角成分の絶対値が 1 と比べて非常に小さくなる可能性があり, 1 とある実数との差で決まる対角成分に情報落ちが発生する. そのため,  $F_i$  の構成要素である  $\tilde{U}$  を 4 倍精度で求める必要がある. これにより, 固有ベクトルの直交性を得ることができる. 2 分割の場合には,  $p_i$  がスカラのため, この問題は生じない.

以上で述べたアルゴリズム 3' と アルゴリズム 4 を用いると, 提案アルゴリズム (アルゴリズム 2) は以下のアルゴリズムに書き換えられる.

アルゴリズム 2' 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`( $T, Q, \Lambda$ )  $\cdots \cdots$  入力  $T$  から  $T = Q\Lambda Q^T$  を満たす直交行列  $Q$ , 対角行列  $\Lambda$  を出力する

```

if  $T$  が  $k-1$  次以下
  他のアルゴリズムで  $T$  の固有値問題  $T = Q\Lambda Q^T$  を解く.
  return  $Q, \Lambda$ 
else
   $T = (T_1 \oplus \cdots \oplus T_k) + c_1 v_1 v_1^T + \cdots + c_{k-1} v_{k-1} v_{k-1}^T$  と分割する.
  call dc_eig( $T_i, Q_i, \Lambda_i$ ) ( $i = 1, 2, \dots, k$ )
   $\Lambda_i, Q_i$  を用いて  $D + UCU^T$  の形式にする.
  call eval_lowrank( $D, U, C, \Lambda, \tilde{U}$ )
  call evec_lowrank( $D, \tilde{U}, \Lambda, Q'$ )
   $Q = (Q_1 \oplus \cdots \oplus Q_k)Q'$  を計算する.
  return  $Q, \Lambda$ 
endif

```

このアルゴリズムの演算量は次の小節で評価する.

### 3.5 適切な分割数

分割数の増加に伴い, 2 分割では最も演算量の多い固有ベクトルを求めるための行列行列積の演算量は減少する. しかし, その他の演算量  $O(n^2)$  の部分は  $k$  が増加すると演算量が増加するため,  $O(n^2)$  の演算量は無視できない. そのためこの小節では, 我々の実装において, 行列の次元に応じた演算量評価を理論的にを行い, それに基づいて適切な分割数について議論する.

$k$  分割の分割統治法の演算量を  $t(n, k)$  とすると, (3.13) と (3.14) より

$$\begin{aligned}
 t(n, k) &= kt(n/k, k) && \text{再帰的に } k \text{ 個の三重対角行列の固有値問題を解く} \\
 &+ \frac{2(k-1)}{k} \left( c + 2d \left( \frac{16}{3} - \frac{2}{3k} - \frac{\log_2 k}{k-1} \right) \right) n^2 && \text{対角行列と階数 } k-1 \text{ の和の固有値を求める} \\
 &+ (2d(k-1)(k-2) + (2k+3))n^2 + dek^3n && \text{対角行列と階数 } k-1 \text{ の和の固有ベクトルを求める} \\
 &+ (2/k)n^3 && \text{固有ベクトルを計算するための行列行列積}
 \end{aligned}$$

となる. ただし, 対角行列と階数 1 の和の固有値解法の演算量を  $cn^2$ , 4 倍精度演算は倍精度演算の  $d$  倍の演算時間がかかり,  $k$  次実対称行列の固有値 0 に対応する固有ベクトルの演算量は  $ek^3$  であるとする.

ここで,  $n$  は  $k$  のべき乗と仮定し,  $t(n, k) = kt(n/k, k) + \alpha n^3 + \beta n^2 + \gamma n$  において  $t(k, k)$  になるまで展開すると,

$$t(n, k) = \alpha \left( 1 + \frac{1}{k^2} + \frac{1}{k^4} + \cdots + \frac{1}{(n/k)^2} \right) n^3 + \beta \left( 1 + \frac{1}{k} + \frac{1}{k^2} + \cdots + \frac{1}{n/k} \right) n^2$$

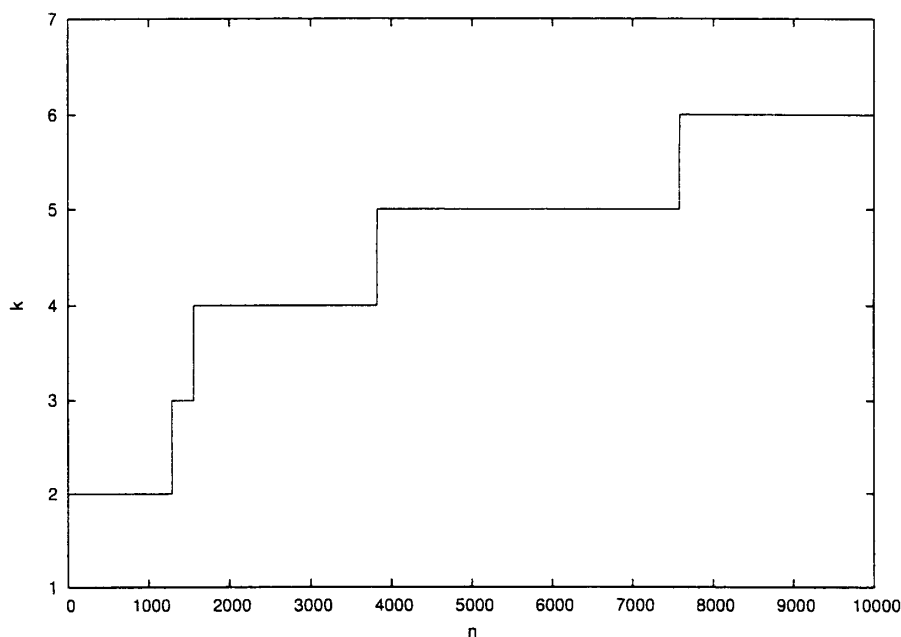


Fig. 1: Optimal division number  $k$  as a function of matrix size  $n$ .  $c = 30, d = 30, e = 5$  are taken in  $t(n, k)$  in the text.

$$\begin{aligned}
 & +\gamma(1+1+\cdots+1)n + \frac{n}{k}t(k, k) \\
 = & \alpha \frac{k^2(n^2-1)}{n^2(k^2-1)}n^3 + \beta \frac{k(n-1)}{n(k-1)}n^2 + \gamma(\log_k n)n + \frac{n}{k}t(k, k) \\
 = & \alpha \frac{k^2}{k^2-1}n(n^2-1) + \beta \frac{k}{k-1}n(n-1) + \gamma(\log_k n)n + \frac{n}{k}t(k, k)
 \end{aligned}$$

となる.  $\alpha, \beta, \gamma$  にそれぞれ代入すると,

$$\begin{aligned}
 t(n, k) = & \frac{2k}{k^2-1}n(n^2-1) + 2\left(c + 2d\left(\frac{16}{3} - \frac{2}{3k} - \frac{\log_2 k}{k-1}\right)\right)n(n-1) \\
 & + (2d(k-1)(k-2) + (2k+3))\frac{k}{k-1}n(n-1) + dek^3(\log_k n)n + \frac{n}{k}t(k, k)
 \end{aligned}$$

となる. ただし, 再帰を行列サイズが  $k$  になるまで繰り返したときの評価である.  $k \ll n$  のときは,  $t(n, k) \simeq (2k/(k^2-1))n^3$  となり, 3.3 節の評価と一致する.

Figure 1 は, 倍精度に換算した理論的な演算回数  $t(n, k)$  を用いて求めた, 行列の次数に対する最適な分割数を示すグラフである. 横軸が行列の次数で, 縦軸が最適な分割数である. ただし,  $k=2$  は従来の分割統治法で, その演算量は (2.20) より  $(4/3)n^3 + 2(c+8)n^2$  とした. 第 2 項は, 対角行列と階数 1 の摂動の和の固有値問題を解くための演算量である. ここで,  $t(n, k)$  のパラメータは, アルゴリズム 2' を用いると  $c=30, e=5$  であり,  $d$  は 4 節に示す数値実験の計算機環境における値を用い  $d=30$  と選んだ.

計算機環境によって  $d$  は変動するものの、このように  $d$  に応じて  $f(n, k)$  から最適な分割数を決定することができる。

## 4 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  $k$  分割の分割統治法で求めたときの演算回数と精度についての数値実験を行った結果を示す。ただし、 $Q = (q_1 \ q_2 \ \cdots \ q_n)$  は直交行列、 $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$  は対角行列である。精度については、直交誤差  $\epsilon_O$  と残差  $\epsilon_R$

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

を用いて評価する。

### 4.1 計算機環境

この節の実験は、以下の環境で行った。

- CPU: Intel Pentium III 600 MHz
- OS: Windows Me
- メモリ: 320 MB
- 使用コンパイラ: g77, gcc, g++ 2.95.3 (コンパイラオプション -O2)

倍精度演算は CPU の浮動小数点演算装置により行い、4 倍精度演算は自作のプログラム (言語: C, C++) により行った。4 倍精度演算の仕様と性能は、以下の通り。ビット長は 128 ビット (仮数部 112 ビット, 指数部 15 ビット, 符号部 1 ビット, これは SPARC の 4 倍精度のフォーマットと同じ) である。マシンイプシロンは  $2^{-112} \simeq 1.9 \times 10^{-33}$  である。倍精度演算との速度比は、平均して  $d = 30$  倍である。

提案手法として、アルゴリズム 2' に基づいて実装している。分割統治法以外の固有値問題解法として、行列の次元  $n$  が  $k$  以下の固有値問題には QR 法を、アルゴリズム 4 で求める  $F_i$  の絶対値最小の固有値に対応する固有ベクトルには、逆反復法を用いている。

### 4.2 結果

Table 1 は、区間  $(-1, 1)$  の一様な乱数を要素に持つ  $n$  次実対称三重対角行列の固有値問題を、従来の分割統治法と提案した  $k$  分割の分割統治法で解いたときの演算回数と演算時間の比較を表す。Table 1 から、まず総演算量が理論とほぼ同じで  $3k/(2(k^2 - 1))n^3$  程度

になっていることがわかる. 次に,  $n = 500$  での倍精度演算の回数の理論値は,  $k = 2, 3, 4$  でそれぞれ約  $167 \times 10^6$ ,  $94 \times 10^6$ ,  $67 \times 10^6$  で, 10 パーセント以内の誤差であり, その他の倍精度演算の回数と 4 倍精度演算の回数を 3.5 節と比べても大きな差異はなくほぼ理論どおりである. また, 実行時間では  $n = 2000$  以上ならば提案法が速い.  $n = 3000$  のときは, 分割数  $k = 4$  の実行時間は従来法に比べ, 62.3 % に短縮されている. 現在の実装の 4 倍精度演算は  $d = 30$  で速いとは言えず,  $d$  が小さくなれば, 提案法が高速になる次数はより小さくなると予想される.

Table 2 は, 区間  $(-1, 1)$  の一様な乱数を要素に持つ 1000 次実対称三重対角行列の固有値問題を, 分割数が  $k = 4$  で, 直交化手法なし, Löwner の定理のみ使用, 4 倍精度のみ使用, Löwner の定理と 4 倍精度の併用 (提案手法), 従来法 ( $k = 2$ ) のそれぞれで計算したときの  $\epsilon_O$ ,  $\epsilon_R$  の表である. Table 2 より, 提案手法は従来法と同等な精度が出ていることがわかる. それ以外の手法では, 十分な精度をだすことができなかった. 直交化手法なしの場合と 4 倍精度を用いた場合で精度の改善が見られないのに対し, Löwner の定理は効果があった. このことから, 分割統治法の誤差は, 4 倍精度演算を行っていない, 対角行列と階数  $k - 1$  の固有値の誤差が大きく反映されているものと考えられる. Löwner の定理のみでは固有値の誤差を補いきれず, 4 倍精度演算も用いることで直交性が得られたことから, 倍精度演算のみで精度を保証するためには, 高精度の固有値を求めることのできる方法か, Löwner の定理の拡張が必要であると考えられる.

Table 3 は,  $n$  次 Frank 行列 ( $a_{ij} = n + 1 - \min\{i, j\}$ ) を Householder 法で三重対角行列に変換した行列の固有値問題を,  $n = 1000$  の場合で解いたときの  $\epsilon_O$ ,  $\epsilon_R$  の表である. Table 2 と同様に, 分割数が  $k = 4$  で, 直交化手法なし, Löwner の定理のみ使用, 4 倍精度のみ使用, Löwner の定理と 4 倍精度の併用 (提案手法), 従来法 ( $k = 2$ ) のそれぞれで計算した. Table 3 より, Frank 行列は近接固有値を持つものの, 従来法と同等の精度が得られた. その他は, Table 2 と似た結果が得られ, 同様の現象が起こっているものと考えられる.

また, Poisson 方程式に対する差分法の行列を  $n = 900$  に対して従来法と提案法 ( $k = 4$ ) を用いて固有値問題を解いた. その結果は, 従来法が,  $\epsilon_O = 1.7 \times 10^{-15}$ ,  $\epsilon_R = 2.9 \times 10^{-15}$ , 提案法が,  $\epsilon_O = 1.6 \times 10^{-15}$ ,  $\epsilon_R = 3.4 \times 10^{-15}$  である. この行列に対しても, 提案法は従来法と同等の精度が得られた.

Table 4 は,

$$(4.1) \quad \begin{pmatrix} W_{21} & \beta & & \\ \beta & W_{21} & & \\ & & \ddots & \beta \\ & & \beta & W_{21} \end{pmatrix}$$

の固有値問題 [16] を, 従来の分割統治法と, 4 分割の分割統治法で解いたときの  $\epsilon_O$ ,  $\epsilon_R$  の表である. 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.  $\beta$  は実数である. (4.1) は, 分割統治法が苦手としている行列である. 提案手法では, 従来法に及ばないものの, 直交誤差, 残差ともに優秀な結果を得ている. Löwner の定理のみを用いた場合は, まともな解は得られていない. この結果からも, Löwner の定理と 4 倍精度演算を併用することは有効であるといえる.

Table 1: Comparison of floating operation counts in units of  $M = 10^6$  between the original ( $k = 2$ ) and proposed ( $k = 3, 4$ ) algorithms. A symmetric tridiagonal matrix with uniform random numbers between  $-1$  and  $1$  as the matrix elements is used.  $n$  is the matrix size. For each  $n$ , the upper (middle) line shows the operation counts in double- (quadruple-) precision floating-point number processing, while the lower line shows the total execution time in seconds for reference.

|            |          | $k = 2$   | $k = 3$   | $k = 4$   |
|------------|----------|-----------|-----------|-----------|
| $n = 500$  | 倍精度      | 180.9 M   | 96.0 M    | 67.6 M    |
|            | 4 倍精度    | -         | 5.3 M     | 11.2 M    |
|            | 実行時間 [秒] | 1.6       | 2.5       | 4.1       |
| $n = 1000$ | 倍精度      | 1350.9 M  | 755.3 M   | 549.8 M   |
|            | 4 倍精度    | -         | 26.7 M    | 55.9 M    |
|            | 実行時間 [秒] | 11.9      | 14.8      | 21.4      |
| $n = 2000$ | 倍精度      | 10822.2 M | 6024.4 M  | 4312.7 M  |
|            | 4 倍精度    | -         | 103.0 M   | 166.5 M   |
|            | 実行時間 [秒] | 92.1      | 82.8      | 85.8      |
| $n = 3000$ | 倍精度      | 36351.7 M | 20449.4 M | 14544.7 M |
|            | 4 倍精度    | -         | 248.6 M   | 345.1 M   |
|            | 実行時間 [秒] | 465.1     | 336.1     | 289.6     |

Table 2: Numerical precision with/without techniques to keep orthogonality among the eigenvectors in the proposed algorithm ( $k = 4$ ). The matrix ( $n = 1000$ ) is the same as in Table 1. For reference, the results for the original divide-and-conquer (with Löwner's theorem) are shown in the second line.

|                              | $\epsilon_O$          | $\epsilon_R$          |
|------------------------------|-----------------------|-----------------------|
| 従来の分割統治法 ( $k = 2$ )         | $1.2 \times 10^{-15}$ | $2.4 \times 10^{-15}$ |
| 直交化手法なし                      | $1.8 \times 10^{-8}$  | $7.6 \times 10^{-7}$  |
| Löwner の定理を使用                | $8.2 \times 10^{-11}$ | $5.2 \times 10^{-10}$ |
| 4 倍精度演算の使用                   | $9.7 \times 10^{-9}$  | $2.9 \times 10^{-7}$  |
| Löwner の定理と 4 倍精度演算併用 (提案手法) | $4.3 \times 10^{-16}$ | $3.3 \times 10^{-16}$ |

Table 3: The same as in Table 2 except that the matrix is Frank matrix of size  $n = 1000$ .

|                              | $\epsilon_O$          | $\epsilon_R$          |
|------------------------------|-----------------------|-----------------------|
| 従来の分割統治法 ( $k = 2$ )         | $9.5 \times 10^{-16}$ | $2.6 \times 10^{-15}$ |
| 直交化手法なし                      | $3.7 \times 10^{-6}$  | $1.0 \times 10^{-5}$  |
| Löwner の定理を使用                | $1.6 \times 10^{-10}$ | $3.4 \times 10^{-8}$  |
| 4 倍精度演算の使用                   | $7.3 \times 10^{-7}$  | $2.8 \times 10^{-7}$  |
| Löwner の定理と 4 倍精度演算併用 (提案手法) | $1.7 \times 10^{-16}$ | $2.9 \times 10^{-16}$ |

Table 4: Comparison of numerical precision between double- and quadruple-precision floating-point number processings for the matrix (4.1) in the text. The division number is  $k = 4$ . The fourth and fifth (sixth and seventh) lines show  $\epsilon_O$  and  $\epsilon_R$  in double- (quadruple-) precision floating-point number processing. In both precisions, Löwner's theorem is taken into account. For reference, the results for the original divide-and-conquer (with Löwner's theorem) are shown in the second and third lines.

|                      |                            | $\epsilon_O$          | $\epsilon_R$          |
|----------------------|----------------------------|-----------------------|-----------------------|
| 従来の分割統治法 ( $k = 2$ ) | $n = 105, \beta = 10^{-4}$ | $9.7 \times 10^{-15}$ | $1.5 \times 10^{-14}$ |
|                      | $n = 105, \beta = 10^{-6}$ | $4.1 \times 10^{-13}$ | $5.8 \times 10^{-12}$ |
| Löwner の定理のみ         | $n = 105, \beta = 10^{-4}$ | $4.1 \times 10^{-1}$  | $8.5 \times 10^{-1}$  |
|                      | $n = 105, \beta = 10^{-6}$ | $1.0 \times 10^0$     | $1.4 \times 10^0$     |
| 提案手法                 | $n = 105, \beta = 10^{-4}$ | $1.5 \times 10^{-14}$ | $3.6 \times 10^{-13}$ |
|                      | $n = 105, \beta = 10^{-6}$ | $7.2 \times 10^{-12}$ | $4.4 \times 10^{-10}$ |

## 5 結論

本稿では、実対称三重対角行列の固有値問題の分割統治法を  $k$  分割に拡張したアルゴリズムを提案した。行列の次元が  $k$  より十分大きければ、 $k$  分割の分割統治法は、従来の 2 分割の分割統治法と比較して  $3k/(2(k^2 - 1))$  に演算量が削減されることを理論的に示した。

提案アルゴリズムを実装するために、実対称三重対角行列の固有値問題から帰着される、実対角行列と階数  $k - 1$  の摂動との和の実対称固有値問題に対する数値解法アルゴリズムを構成した。その数値解法アルゴリズムは、実対角行列と階数 1 の摂動との和の固有値問題を逐次的に  $k - 1$  回解くことにより、固有値を求める方法である。実対角行列と階数 1 の摂動との和の固有値問題は、従来の分割統治法の中核をなす問題で、既に知られている方法で解くことができる。逐次的に解く方法は、分割統治法に由来する、実対角行列と階数  $k - 1$  の摂動の和の固有値問題に対して特に有効である。この固有値問題は、加える摂動が疎で、かつ規則的な構造を有しており、我々の解法は、この構造を活用することができる。階数 1 の摂動を加える順序を変えることで、並列性を高めることができ、かつ演算量を削減することができる。さらに、摂動項の順序を並び替えたことで、固有値の求め方が従来の分割統治法と本質的に等価であることが明らかになった。これは、固有値に関しては、従来の分割統治法と同程度の精度が得られることを意味している。

一方、実対角行列と階数  $k - 1$  の摂動の和の固有ベクトルは、特性方程式から導かれる  $k - 1$  次の実対称行列 (3.8) の固有値 0 に対応する固有ベクトルを用いた公式を単に利用するだけでは数値的直交性が得られない。直交性を確保するために、2 つの手法を併用した。一つは、実対角行列と階数 1 の摂動の和の固有値問題で、直交性を保証するために用いる Löwner の定理を用いる手法である。Löwner の定理を用いると、計算した固有値の近似値が「正確な」固有値をもつように、摂動を再定義することができる。しかし、それだけでは十分ではなく、4 倍精度の演算を適切な箇所で行うことで直交性を得た。

数値実験により、演算回数は理論値とほぼ同様の結果が得られ、理論の正当性が検証された。精度の点では、Löwner の定理と 4 倍精度演算のどちらか一方のみでは有効な手段とはいえず、組み合わせることにより効果的な手法となることが明らかになった。また、数値実験の結果よりアルゴリズム面で  $k$  分割の分割統治法の中核をなす、対角行列と階数  $k - 1$  の摂動の和の固有値が、分割統治法自体の精度に大きく関わっており、精度の高い固有値、固有ベクトルを得るためには、精度の高い固有値を求めるか、固有値の近似値が「正確な」固有値であるようにできる、Löwner の定理の拡張が必要であるという知見を得た。

今後の課題として次の 3 つが挙げられる。

一つは、対角行列と階数 2 以上の摂動の和の固有値問題においてアルゴリズムの下波線の部分に、deflation をとり入れることである。2 分割の分割統治法において、Tisseur は、deflation の効果 B によって、平均で  $O(n^{2.3})$  の演算量で済み、特別な行列に対しては、その演算量は  $O(n^2)$  にまで抑えられると述べている [5], [17]。また Cuppen は、対角優位の行列に対して  $O(n \log n)$  の演算量であるという数値実験結果を得ている [4]。これらの結果



は, deflation の効果 B による演算量の削減であり, 3 分割以上でも deflation を用いることで同程度の演算量の削減が期待できる.

第二には, 専用のハードウェアまたはソフトウェアが必要となる 4 倍精度演算を用いない実装方法を設計することである. そのためには, 現在のアルゴリズムより高精度なアルゴリズムや Löwner の定理の拡張が必要である.

第三に, 実用的な実装のためには, 行列の次元が小さい場合に他の解法へ切り替えることによる高速化を取り入れる必要がある. また, 行列分割の再帰的過程において, 同一の分割数ではなく, 行列の次元に応じて分割数を可変にすることで, より演算量の削減が期待できる.

謝辞 本研究の一部は科学研究費補助金(課題番号: 16656033)のもとで行われた.

## 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] Bunch, J. R., Nielsen, C. P. and Sorensen, D. C., Rank-One Modification of the Symmetric Eigenproblem, Numerische Mathematik, 31 (1978), 31–48.
- [4] Cuppen, J. J. M., A divide and conquer method for the symmetric tridiagonal eigenproblem, Numerische Mathematik, 36 (1981), 177–195.
- [5] Demmel, J. W., Applied Numerical Linear Algebra, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1997.
- [6] 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.
- [7] 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.
- [8] Gates, K. and Arbenz, P., Parallel divide and conquer algorithms for the symmetric tridiagonal eigenproblem, Technical report, Institute for Scientific Computing, ETH Zurich, 1994.

- [9] Golub, G. H., Some modified matrix eigenvalue problems, SIAM Rev., 15 (1973), 318–334.
- [10] 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.
- [11] 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.
- [12] Li, R.-C., Solving Secular Equations Stably and Efficiently, CS-94-260, 1994.
- [13] Löwner, K., Über monotone Matrixfunktionen, Math. Z., 38 (1934), 177–216.
- [14] Melman, A., A numerical comparison of methods for solving secular equations, J. Comp. Appl. Math., 86 (1997), 237–249.
- [15] 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.
- [16] 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.
- [17] 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.

## A 定理, 補題の証明

補題. 1  $n \times k$  実行列  $X, Y$  に対して,  $\det(I_n + XY^T) = \det(I_k + Y^T X)$

証明  $n$  次正則行列  $A$ ,  $n \times k$  次行列  $B, C$ ,  $k$  次正則行列  $D$  が与えられたとき

$$(A.1) \quad \det \begin{pmatrix} A & B \\ C^T & D \end{pmatrix} = \det(A) \cdot \det(D - C^T A^{-1} B)$$

である. 一方, 左右から適当な順列行列を掛けることにより

$$(A.2) \quad \det \begin{pmatrix} A & B \\ C^T & D \end{pmatrix} = \det \begin{pmatrix} D & C^T \\ B & A \end{pmatrix}$$

となる. よって,

$$(A.3) \quad \det(D) \cdot \det(A - BD^{-1}C^T) = \det(A) \cdot \det(D - C^T A^{-1} B)$$

より,  $A = I_k, B = X, C = Y, D = I_n$  とおくと, 題意が示される. ■

定理 1.  $A$  を  $n$  次実対称行列,  $u$  を  $n$  次実ベクトル,  $c$  を非零実数とする.  $n$  次実対称行列  $\hat{A} = A + cuu^T$  の固有値が  $\lambda$  であるとき,  $\lambda I_n - A$  が正則ならば,  $\lambda$  に対応する  $\hat{A}$  の固有ベクトルは  $(\lambda I_n - A)^{-1}u$  である.

証明  $\det(\lambda I_n - A) \neq 0$  と補題 1 より,

$$(A.4) \quad \frac{\det(\lambda I_n - \hat{A})}{\det(\lambda I_n - A)} = \det(I_n - c(\lambda I_n - A)^{-1}uu^T) = 1 - cu^T(\lambda I_n - A)^{-1}u = 0.$$

また,

$$\begin{aligned} (\lambda I_n - (A + cuu^T))(\lambda I_n - A)^{-1}u &= u - cuu^T(\lambda I_n - A)^{-1}u \\ &= (1 - cu^T(\lambda I_n - A)^{-1}u)u \\ &= 0. \end{aligned}$$

よって題意を得る. ■

定理 3.  $A$  を  $n$  次実対称行列,  $U$  を  $n \times k$  次行列,  $C$  を  $k$  次正則実対称行列とする.  $n$  次実対称行列  $\hat{A} = A + UCU^T$  の固有値が  $\lambda$  であるとき,  $\lambda I_n - A$  が正則ならば,  $k$  次実対称行列  $F = C^{-1} - U^T(\lambda I_n - A)^{-1}U$  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  $q$  を用いて,  $\lambda$  に対応する  $\hat{A}$  の固有ベクトルは  $(\lambda I_n - A)^{-1}Uq$  と表せる.

証明  $\det(\lambda I_n - A) \neq 0$  と補題 1 より,

$$\frac{\det(\lambda I_n - \hat{A})}{\det(\lambda I_n - A)} = \det(I_n - (\lambda I_n - A)^{-1}UCU^T) = \det(C) \det(C^{-1} - U^T(\lambda I_n - A)^{-1}U) = 0$$

であり,  $\det(C) \neq 0$  より,  $C^{-1} - U^T(\lambda I_n - A)^{-1}U$  は固有値 0 を持つ. また,

$$\begin{aligned} (\lambda I_n - (A + UCU^T))(\lambda I_n - A)^{-1}Uq &= Uq - UCU^T(\lambda I_n - A)^{-1}Uq \\ &= UC(C^{-1} - U^T(\lambda I_n - A)^{-1}U)q \\ &= 0. \end{aligned}$$

よって題意を得る. ■

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科学専攻に在籍. 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 (非会員) 〒 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.

(2005 年 3 月 14 日 受付)

(2005 年 5 月 16 日 最終稿 受付)