

帯行列の一般化固有値問題向け分割統治法

廣田 悠輔^{1,2,a)} 今村 俊幸^{1,2}

受付日 2015年4月21日, 採録日 2015年6月24日

概要: 本稿では, 実対称正定値帯行列向けの一般化固有値解法を提案する. 提案法は, Elsner らによって提案された三重対角行列の一般化固有値問題の分割統治法の拡張であり, 三重対角行列向け解法の統治フェーズを繰り返し適用することで一般の帯幅の帯行列の固有値問題を解く. 近年のマルチコア CPU の普及と性能向上により, マルチコア計算機に適した数値解法の重要性はますます高くなっているが, 問題を標準固有値問題に変換して解く従来法はデータ再利用性の低い演算を多く含むため, マルチコア計算機上で高い性能を実現することが難しい. 一方, 提案法では演算のほとんどが行列積として実行され, 従来法に比べて高い実行性能が実現できる. Intel Xeon E5-2660 2 ソケットを備えるマルチコア計算機における性能評価では, 次数 10240 の五重対角行列の一般化固有値問題を解くとき, 提案法は, 従来法の 3.18 倍高速であり, 219 GFLOPS (ピーク性能比 77.6%) の高い性能を示すことが確認された.

キーワード: 一般化固有値問題, 三重対角行列, 帯行列, 分割統治法, マルチコア

Divide-and-conquer Method for Banded Generalized Eigenvalue Problems

YUSUKE HIROTA^{1,2,a)} TOSHIYUKI IMAMURA^{1,2}

Received: April 21, 2015, Accepted: June 24, 2015

Abstract: In this paper, we present a new solution method for symmetric-positive definite generalized eigenvalue problems of banded matrices. The proposed method is an extension of the divide and conquer method proposed by Elsner et al., which repeats the conquer phase of the divide and conquer method for a problem of tridiagonal matrices. Recently, numerical solution methods are required to work efficiently on modern multicore processors. However, the conventional methods show poor performance on such environment since they contain many cache inefficient operations. On the other hand, the proposed method is dominated by matrix products thus it shows higher performance than the conventional methods. The proposed method is 3.23 times faster than the conventional method, achieving 219 GFLOPS (77.6% of the peak performance) on a multicore environment (two octa-core Intel Xeon E5-2660 CPUs).

Keywords: generalized eigenvalue problem, tridiagonal matrix, banded matrix, divide and conquer method, multicore

1. はじめに

近年, 多くの計算機においてマルチコア CPU が利用されている. マルチコア CPU を備える計算機 (マルチコア

計算機) では, CPU は高いピーク演算性能を持つ一方, ピーク性能で動作する CPU に対してデータを供給し続けるだけのメモリ帯域を持たないことが一般的である. したがって, マルチコア計算機において高い性能で演算を実行するためには, 1 度メモリから読み込んだデータを CPU のキャッシュメモリに蓄えて再利用し, メモリからのデータのロード回数をできるだけ削減する必要がある.

行列やベクトルの計算のデータ再利用性について考えると, ベクトルの内積や加算などのベクトルどうしの演算 (Level-1 演算) や, 行列ベクトル積, 行列のランク 1 更新

¹ 理化学研究所計算科学研究機構
RIKEN Advanced Institute for Computational Science,
Kobe, Hyogo 650-0047, Japan

² 科学技術推進機構戦略的創造研究推進事業
Japan Science and Technology Agency CREST, Chiyoda,
Tokyo 102-0076, Japan

^{a)} yusuke.hirota@riken.jp

などの行列とベクトルの演算 (Level-2 演算) は、演算回数に対して必要となるデータ量が多く、マルチコア計算機において高い実行性能を実現することが難しい。一方、行列積などの行列どうしの演算 (Level-3 演算) は、演算回数に対して必要となるデータ量が少なく、適切にキャッシュメモリを利用すれば、高い実行性能が実現できる。したがって、数値計算アルゴリズムを基本行列演算の組合せとして構築する場合、できるだけ Level-3 演算が中心的となるようにアルゴリズムを構築することが、マルチコア計算機で高い性能を実現するために必須となる。

本稿では、半帯幅 k が小さな値の実対称帯行列 $A \in \mathbb{R}^{n \times n}$ および、同じ半帯幅の実対称正定値帯行列 $B \in \mathbb{R}^{n \times n}$ の一般化固有値問題

$$Ax = \lambda Bx \tag{1}$$

の固有値 λ , 固有ベクトル x をすべて求める数値解法について考える。式 (1) は n 組の固有値, 固有ベクトル (固有対) を持つ。したがって、式 (1) の全固有対を求めることは、

$$X^T(A - \lambda B)X = \Lambda - \lambda I$$

を満たす対角行列 $\Lambda \in \mathbb{R}^{n \times n}$, B -直交行列 $X \in \mathbb{R}^{n \times n}$ を求めることに等しく、 Λ の対角項, X の各列ベクトルがそれぞれ式 (1) の固有値, 固有ベクトルとなる。このような問題に対する解法は、帯化前処理と組み合わせた密行列向け解法の部品 [1] として応用可能であるほか、電子状態計算に利用できる。

問題 (1) に対する解法は、図 1 に示されるように、様々なアプローチが考えられる。従来法では、赤や緑の線で示されるように、与えられた一般化固有値問題を標準固有値問題に変換し、標準固有値問題を解いた結果を、一般化固有値問題の固有ベクトルに逆変換するという手順がとられる。しかしながら、これらの解法は Level-1, Level-2 演算を多く含み、マルチコア計算機で十分な性能を引き出すことが難しい。そこで、青の線で示される、中間形を経ずに直接に式 (1) の一般化固有値, 固有ベクトルを求めることを考える。このような方法は、 $k = 1$ (すなわち三重対角行列) の場合には、Elsner らによって解法が提案されており [2], そのアルゴリズムは Level-3 演算が支配的となる。本研究では、Elsner らの解法を $k \geq 2$ の場合に拡張することで、Level-3 演算が支配的となる数値解法を新たに提案する。なお、Elsner らの方法とは別に、文献 [3], [4] で $k = 1$ の場合の解法が提案されているが、固有ベクトルの高精度化手段 [5] の適用手段が確立されておらず、また、その解法の性質上 $k \geq 2$ への拡張が困難であるなどの理由により、本稿ではこれらについては取り扱わない。

本稿の構成は以下のとおりである。2 章では、標準固有値問題を經由する従来法について述べ、その演算量, 演算の種

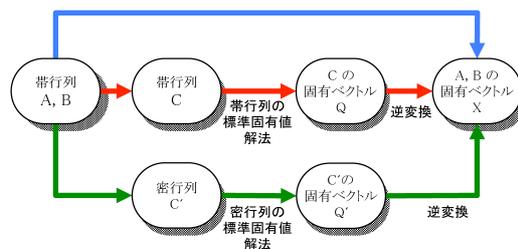


図 1 帯行列一般化固有値問題に対する解法アプローチ
 Fig. 1 Solution approaches for banded generalized eigenvalue problems.

類について分析する。3 章では、一般化固有値問題向けの分割統治法について述べる。その中で、帯行列向け分割統治法を提案する。また、演算量, 演算の種類について分析し、従来法との比較を行う。4 章では、従来法および提案法の精度ならびに性能についてマルチコア計算機上で評価し、評価結果をもとに議論を行う。5 章で本稿についてまとめる。

2. 標準固有値問題を經由する解法

一般化固有値問題 (1) は、 A, B が帯行列か否かに関係なく、両辺に左から S^{-1} をかけることで、

$$(S^{-1}AS^{-T})y = \lambda y, y = S^T x$$

という標準固有値問題に変換することができる。ただし、 S は $B = SS^T$ を満たす任意の行列である。したがって、一般化固有値問題 (1) は、コレスキー分解などにより $B \rightarrow SS^T$ と分解し、 $C \leftarrow S^{-1}AS^{-T}$ を構成し、 C の標準固有値, 固有ベクトルを求め、固有ベクトルを逆変換することで解くことができる。この原理に基づく解法は、数値計算ライブラリ LAPACK [6] に採用され、DSBGV, DSBGVD, DSYGVD ルーチンとして実装されている。

本章では、式 (1) を標準固有値問題を經由して解く 2 つの解法について述べ、その演算量および演算の種類について分析する。

2.1 行列の帯構造を利用する解法

本節では、上述の原理に基づく解法のうち、行列 A, B の帯構造を利用する解法について述べる。このような帯構造を利用する解法は、LAPACK では DSBGV, DSBGVD などとして実装されている。DSBGVD の解法は以下のようになる。

- (i) 帯行列 B を $B = SS^T$ と split Cholesky 分解する (DPBSTF ルーチン)。
- (ii) A を $C \leftarrow Z^T A Z$ と合同変換する。ただし、 $Z = S^{-1}P$ であり、 P は C の帯幅が A に等しくなるように fill-in を消去するような直交行列である。また、同時に $Z \leftarrow S^{-1}P$ を計算する (DSBGST ルーチン)。
- (iii) 帯行列 C を直交行列 Q_1 によって $T \leftarrow Q_1^T C Q_1$ と三重対角行列に変換し、同時に $X' \leftarrow Z Q_1$ を計算する

表 1 行列の帯構造を利用する解法の演算量内訳および演算の種類
Table 1 The number of FLOPs and the computational pattern in the conventional method which exploits the band structure of the matrix.

	演算量	種類
(i)	$(k+1)^2n$	Level-2
(ii) A の変換	$6kn^2$	Level-1 および Level-2
Z の計算	$(3/2)n^3$	Level-1 および Level-2
(iii) 三重対角化	$6kn^2$	Level-1
X' の計算	$2n^3$	Level-1
(iv) 分割統治法	$(4/3)n^3$	Level-3
行列積	$2n^3$	Level-3

(DSBTRD ルーチン).

(iv) $T \rightarrow Q_2 D Q_2^T$ と分割統治法により標準固有値問題を解き (DSTEDC ルーチン), その後, 行列積ルーチン (DGEMM ルーチン) により $X \leftarrow X' Q_2$ を計算することで式 (1) の固有ベクトルに変換する.

ただし, $k = 1$ (すなわち A, B が三重対角行列) である場合にはステップ (iii) はスキップされる. (i) から (iv) の演算量および支配的となる演算の種類を表 1 に示す. 総演算量は, $k = 1$ の場合には $(29/6)n^3 + O(kn^2)$, $k \geq 2$ の場合には $(41/6)n^3 + O(kn^2)$ となる. そのうち, Level-3 演算は $(10/3)n^3$ であり, いずれの半帯幅 k でも多くの Level-1, Level-2 演算が含まれる. このため, マルチコア計算機上において高い性能 (FLOPS 値) が得られない可能性がある. なお, DSBGV は, DSBGVD と同様に帯構造を利用するが, DSTEDC および DGEMM ルーチンの代わりに QR 法ルーチン (DSTEQR) を用いて (iv) のステップを実行している. したがって, DSBGVD と同様に (ii), (iii) において Level-1, Level-2 演算が必要となり, これらの部分は同様の性能を示すと考えられる.

2.2 行列を密行列として扱う解法

問題 (1) の A, B が帯行列であっても, その疎構造を無視して密行列として問題を解くことも可能である. そのような方法は, LAPACK では DSYGVD などとして実装されており, 以下の手順で式 (1) の固有対を求める.

- (i) 実対称正定値行列 B を密行列として $B \rightarrow LL^T$ コレスキー分解する (DPOTRF ルーチン).
- (ii) B の分解結果を A に作用させて $C \leftarrow L^{-1} A L^{-T}$ を計算する (DSYGST ルーチン).
- (iii) $C \rightarrow Q D Q^T$ と標準固有値問題を解く. DSYGVD で用いられる解法 (DSYEVD ルーチン) は, C をブロック化されたハウスホルダ変換によって三重対角化し, 三重対角行列を標準固有値問題向け分割統治法によって固有値分解し, 逆変換によって C の固有値分解に戻すという手順がとられる.
- (iv) $X = L^{-T} Q$ を計算することで式 (1) の固有ベクトル

表 2 密行列として扱う解法の演算量内訳および演算の種類
Table 2 The number of FLOPs and the computational pattern in the conventional method which does not exploit the band structure of the matrix.

	演算量	種類
(i)	$2/3n^3$	Level-3
(ii)	n^3	Level-3
(iii) 三重対角化	$4/3n^3$	Level-2, Level-3 が半分ずつ
分割統治法	$4/3n^3$	Level-3
逆変換	$2n^3$	Level-3
(iv)	n^3	Level-3

に変換する (DTRSMLR ルーチン).

(i) から (iv) の演算量および支配的となる演算の種類を表 2 に示す. 行列の帯構造を利用する解法と比べると演算量が増大するが, 標準固有値解法の三重対角化以外が Level-3 演算となるためマルチコア計算機などでより高い性能が得られると予想され, 結果的に帯構造を利用する解法より高速に問題を解ける可能性がある.

3. 一般化固有値問題向け分割統治法

本章では, まず, Elsner らによって提案された $k = 1$ の帯行列 (三重対角行列) 向けの分割統治法アルゴリズムについて説明する. 続いて, その拡張である半帯幅が $k \geq 2$ の帯行列向けの分割統治法アルゴリズムを提案し, 提案法および従来法のアルゴリズムの性質について高性能計算の観点から議論する.

3.1 三重対角行列向け分割統治法

本節では, Elsner らの分割統治法について述べる.

3.1.1 原理

三重対角行列 A, B は, 任意の分割点 m を定めて, $b_{m,m+1} \neq 0$ であれば,

$$A - \lambda B = (A_1 \oplus A_2 - \rho \mathbf{v} \mathbf{v}^T) - \lambda (B_1 \oplus B_2 - \mathbf{v} \mathbf{v}^T) \quad (2)$$

とブロック対角行列と, 共通のベクトル \mathbf{v} によって表現されるランク 1 行列に分解できる. ただし, $A_1, B_1 \in \mathbb{R}^{m \times m}$, $A_2, B_2 \in \mathbb{R}^{(n-m) \times (n-m)}$ であり, $\rho = a_{m+1,m} / b_{m+1,m}$,

$$\mathbf{v} = \sqrt{|b_{m+1,m}|} \mathbf{e}_m - \text{sign}(b_{m+1,m}) \sqrt{|b_{m+1,m}|} \mathbf{e}_{m+1}$$

である. このような分解によって得られる B_1, B_2 は正定値行列であり, A_1, A_2, B_1, B_2 はそれぞれ対称三重対角行列となる.

A_1, A_2, B_1, B_2 が実対称かつ B_1, B_2 は正定値行列であるので,

$$Y_i^T (A_i - \lambda B_i) Y_i = D_i - \lambda I \quad (i = 1, 2) \quad (3)$$

を満たす B_1 -直交行列 Y_1, B_2 -直交行列 Y_2 が存在する. ただし, D_1, D_2 は対角行列である. したがって, 式 (2)

の右辺は、 $Y = Y_1 \oplus Y_2$ の合同変換によって

$$Y^T[(A_1 \oplus A_2 - \rho vv^T) - \lambda(B_1 \oplus B_2 - vv^T)]Y = (D - \rho ww^T) - \lambda(I - ww^T) \quad (4)$$

と対角行列とランク 1 摂動の和に変換できる。ただし、 $D = D_1 \oplus D_2$, $w = Y^T v$ である。

式 (4) の右辺を

$$W^T[(D - \rho ww^T) - \lambda(I - ww^T)]W = D' - \lambda I \quad (5)$$

と対角化すれば、式 (2), (4), (5) より

$$(YW)^T(A - \lambda B)(YW) = D' - \lambda I$$

が成り立ち、式 (1) の解が $X = YW$, $\Lambda = D'$ として表される。なお、一般化固有値問題 (5) は、必要に応じて減次 (デフレーション) を行った後、一変数非線型方程式 (secular 方程式) を解いて固有値を求め、その後に対応する固有ベクトルを計算することで解くことができる。具体的な方法については文献 [2] を参照されたい。

3.1.2 アルゴリズム

以上の原理に基づく Elsner らの分割統治法は以下の手順にまとめられる。

- (i) 行列 A, B を式 (2) の形に分解する。
- (ii) もとの行列よりも次数の小さな固有値問題 (3) を Elsner らの解法によって再帰的に解く。
- (iii) 小さな固有値問題を解いた結果をもちいて $w = Y^T v$ を計算する。
- (iv) secular 方程式を解くことにより、式 (5) を満たす W, D' を求める。
- (v) 式 (1) の固有ベクトル $X = YW$ を計算する。

(i), (iii), (iv) の演算量はいずれも $O(n^2)$ であり、(v) の演算量は Y のブロック対角性を考慮する場合には $2nm^2 + 2n(n-m)^2$ である。したがって、つねに $m \simeq n/2$ として行列を中心付近で分割して再帰的に問題を解く場合、総演算量は $(4/3)n^3 + O(n^2)$ となる。なお、(iv) でデフレーションが行われる場合、 W を陽に計算せずに低次元の密行列と特殊な構造を持つ疎行列の積として陰的に求めることで、(v) の行列積の演算量の削減が可能であるが、本稿では W を陽に計算する場合について考えている。

3.2 帯行列向け分割統治法

本節では、一般の半帯幅 k の帯行列に適用可能な分割統治法を提案する。最初の項で提案法の原理について述べ、次の項でアルゴリズムについて述べる。

3.2.1 原理

帯行列 A, B に対して、 $k \leq m \leq n - k$ を満たす分割点 m を定めたとき、

$$A - \lambda B = (A_1 \oplus A_2 - VEV^T) - \lambda(B_1 \oplus B_2 - VV^T) \quad (6)$$

を満たす、正整数 p 、半帯幅 k の実対称帯行列 $A_1, B_1 \in \mathbb{R}^{m \times m}$, $A_2, B_2 \in \mathbb{R}^{(n-m) \times (n-m)}$ 、行列 $V \in \mathbb{R}^{n \times p}$ 、対角行列 $E \in \mathbb{R}^{p \times p}$ が存在する。具体的な $p, A_1, A_2, B_1, B_2, V, E$ の構成法については後述する。

分解 (6) を満たす B_1, B_2 はいずれも正定値行列であることを示す。ブロック対角行列 $B_1 \oplus B_2 = B + VV^T$ は、正定値行列 B と半正定値行列 VV^T の和であるので正定値行列である。行列 B_1 が正定値でないと仮定すると、 $z_1^T B_1 z_1 \leq 0$ を満たす $z_1 \in \mathbb{R}^m$ が存在する。このとき、 $z := [z_1^T, \mathbf{0}^T]^T \in \mathbb{R}^n$ に対して、 $z^T (B_1 \oplus B_2) z = z_1^T B_1 z_1 \leq 0$ となり、 $B_1 \oplus B_2$ が正定値であることに矛盾する。行列 B_2 が正定値でないと仮定した場合も同様である。したがって、 B_1, B_2 はいずれも正定値行列である。

A_1, A_2, B_1, B_2 が実対称かつ B_1, B_2 は正定値行列であるので、

$$(X_i^{(0)})^T (A_i - \lambda B_i) X_i^{(0)} = D_i^{(0)} - \lambda I \quad (i = 1, 2) \quad (7)$$

を満たす B_1 -直交行列 $X_1^{(0)}$, B_2 -直交行列 $X_2^{(0)}$ が存在する。ただし、 $D_1^{(0)}, D_2^{(0)}$ は対角行列である。したがって、式 (6) の右辺は、 $X^{(0)} := X_1^{(0)} \oplus X_2^{(0)}$ による合同変換で

$$\begin{aligned} & (X^{(0)})^T [(A_1 \oplus A_2 - VEV^T) - \lambda(B_1 \oplus B_2 - VV^T)] X^{(0)} \\ &= [D^{(0)} - \sum_{i=1}^p e_{i,i} u_i^{(0)} (u_i^{(0)})^T] - \lambda [I - \sum_{i=1}^p u_i^{(0)} (u_i^{(0)})^T] \end{aligned} \quad (8)$$

と対角行列と p 個のランク 1 行列の和に変換できる。ただし、 $U^{(0)} = (X^{(0)})^T V$, $u_l^{(0)}$ は $U^{(0)}$ の第 l 列ベクトル、 $D^{(0)} = D_1^{(0)} \oplus D_2^{(0)}$ である。ここで、

$$(W^{(1)})^T \{ [D^{(0)} - e_{1,1} u_1^{(0)} (u_1^{(0)})^T] - \lambda [I - u_1^{(0)} (u_1^{(0)})^T] \} W^{(1)} = D^{(1)} - \lambda I \quad (9)$$

と $[D^{(0)} - e_{1,1} u_1^{(0)} (u_1^{(0)})^T] - \lambda [I - u_1^{(0)} (u_1^{(0)})^T]$ を対角化する $W^{(1)}$ により、(8) の右辺を合同変換すると、

$$\begin{aligned} & (W^{(1)})^T \{ [D^{(0)} - \sum_{i=1}^p e_{i,i} u_i^{(0)} (u_i^{(0)})^T] - \lambda [I - \sum_{i=1}^p u_i^{(0)} (u_i^{(0)})^T] \} W^{(1)} \\ &= [D^{(1)} - \sum_{i=2}^p e_{i,i} u_i^{(1)} (u_i^{(1)})^T] - \lambda [I - \sum_{i=2}^p u_i^{(1)} (u_i^{(1)})^T] \end{aligned}$$

と、ランク 1 行列が 1 つ少ない式に変換できる。同様の交換

$$(W^{(j)})^T \{ [D^{(j-1)} - e_{j,j} u_j^{(j-1)} (u_j^{(j-1)})^T] - \lambda [I - u_j^{(j-1)} (u_j^{(j-1)})^T] \} W^{(j-1)} = D^{(j)} - \lambda I, \quad (10)$$

$$\begin{aligned} & (W^{(j)})^\top \left\{ [D^{(j-1)} - \sum_{i=j}^p e_{i,i} \mathbf{u}_i^{(j-1)} (\mathbf{u}_i^{(j-1)})^\top] \right. \\ & \quad \left. - \lambda [I - \sum_{i=j}^p \mathbf{u}_i^{(j-1)} (\mathbf{u}_i^{(j-1)})^\top] \right\} W^{(j)} \\ & = [D^{(j)} - \sum_{i=j+1}^p e_{i,i} \mathbf{u}_i^{(j)} (\mathbf{u}_i^{(j)})^\top] \\ & \quad - \lambda [I - \sum_{i=j+1}^p \mathbf{u}_i^{(j)} (\mathbf{u}_i^{(j)})^\top] \end{aligned}$$

を $j = 2, 3, \dots, p$ と繰り返すことで、最終的に

$$\begin{aligned} & (X^{(0)} W^{(1)} W^{(2)} \dots W^{(p)})^\top [(A_1 \oplus A_2 - V E V^\top) \\ & \quad - \lambda (B_1 \oplus B_2 - V V^\top)] (X^{(0)} W^{(1)} W^{(2)} \dots W^{(p)}) \\ & = D^{(p)} - \lambda I \end{aligned}$$

という関係が得られ、式 (1) の固有値、固有ベクトルは

$$\Lambda = D^{(p)}, X = X^{(0)} W^{(1)} W^{(2)} \dots W^{(p)} \quad (11)$$

と表されることが分かる。

3.2.2 2つの帯行列の同時分割法

式 (6) を満たす $p, A_1, A_2, B_1, B_2, V, E$ の構成法について述べる。

3.2.2.1 $p = 2k$ の分解

実対称帯行列 A に対して、つねに

$$\begin{aligned} A & = A_1 \oplus A_2 - V_A V_A^\top, \quad (12) \\ V_A & = \begin{bmatrix} O_{(m-k) \times k} \\ V_A^{(1)} \\ V_A^{(2)} \\ O_{(n-m-k) \times k} \end{bmatrix}, V_A^{(1)}, V_A^{(2)} \in \mathbb{R}^{k \times k} \end{aligned}$$

を満たす分解が可能である。ただし、 $O_{i \times j}$ は $i \times j$ のゼロ行列を意味する。このような分解は帯行列の標準固有値問題の分割統治法で用いられており、文献 [7], [8], [9] などで言及されている。 $B + V_A V_A^\top$ は、 A, B と同じ半帯幅 k の帯行列となるので、

$$B + V_A V_A^\top = B_1 \oplus B_2 - V_B V_B^\top, V_B \in \mathbb{R}^{n \times k}$$

を満たす分解が式 (12) の自然な拡張として得られる。このとき、 $V = [V_A, V_B], E = I_{k \times k} \oplus O_{k \times k}$ とおけば、 $p = 2k, A_1, A_2, B_1, B_2, V, E$ は式 (6) を満たす。ただし、 $I_{i \times i}$ は $i \times i$ の単位行列を意味する。したがって、上記の方法で順に $A, B + V_A V_A^\top$ の分解を行うことで、 $p = 2k$ の分解 (6) を構成できることが分かる。

3.2.2.2 $p = k$ の分解

分解 (12) は一意ではなく、式 (6) を満たす分解も一意ではない。また、式 (11) に示されるとおり固有ベクトル X は $p+1$ 個の行列の積として表現されるため、実際に固有ベクトルを計算する際の演算量を減らすことを考えると、

p ができるだけ小さな値となる分解が望まれる。そこで、 $p = k$ である分解 (11) の構成法について考える。以下では、 V の部分行列を

$$V = \begin{bmatrix} V_0 \\ V_1 \\ V_2 \\ V_3 \end{bmatrix}, V_0 \in \mathbb{R}^{(m-k) \times k}, V_1, V_2 \in \mathbb{R}^{k \times k}, V_3 \in \mathbb{R}^{(n-m-k) \times k}$$

とおく。

ある A_1, A_2, B_1, B_2, V, E に対して、 $p = k$ である分解 (11) が成り立つと仮定する。このとき、式 (6) の両辺の行列の第 $m+1$ 行から第 n 行、第 1 列から第 m 列の部分行列 (すなわち左下非対角ブロック) について見てみると、

$$\begin{aligned} & \begin{bmatrix} O_{k \times (m-k)} & C_A \\ O_{(n-m-k) \times (m-k)} & O_{(n-m-k) \times k} \end{bmatrix} \\ & - \lambda \begin{bmatrix} O_{k \times (m-k)} & C_B \\ O_{(n-m-k) \times (m-k)} & O_{(n-m-k) \times k} \end{bmatrix} \\ & = \begin{bmatrix} -V_2 E V_0^\top & -V_2 E V_1^\top \\ -V_3 E V_0^\top & -V_3 E V_1^\top \end{bmatrix} \\ & - \lambda \begin{bmatrix} -V_2 V_0^\top & -V_2 V_1^\top \\ -V_3 V_0^\top & -V_3 V_1^\top \end{bmatrix} \end{aligned}$$

が成り立つことが分かる。ただし、 C_A, C_B は、それぞれ実上三角行列 $A(m+1:m+k, m+1-k:m), B(m+1:m+k, m+1-k:m)$ である。 $M(i:j, k:l)$ は行列 M の第 i 行から第 j 行、第 k 列から第 l 列を取り出した $(j-i+1) \times (l-k+1)$ 部分行列を意味する。

C_B が正則であると仮定する。 $C_B = -V_2 V_1^\top$ より、 V_1, V_2 は正則であり、

$$V_2 = -C_B V_1^{-\top} \quad (13)$$

が成り立つ。式 (13) を $C_A = -V_2 E V_1^\top$ に代入して、両辺に左から C_B^{-1} を作用させると

$$C_B^{-1} C_A = (V_1^{-\top}) E (V_1^{-\top})^{-1}$$

の関係が得られる。したがって、分解 (6) が存在するとき、 $C_B^{-1} C_A$ は実行列による対角化が可能であり、 $V_1^{-\top}$ は対角化する行列、 E は対角化結果となる。また、正則である V_1, V_2 に対して、 $O_{k \times (m-k)} = V_2 V_0^\top, O_{(n-m-k) \times k} = V_3 V_1^\top$ が成り立つことから、 $V_0 = O_{(m-k) \times k}, V_3 = O_{(n-m-k) \times k}$ が導かれる。

したがって、 C_B が正則かつ $C_B^{-1} C_A$ が実行列によって対角化可能な場合、 $C_B^{-1} C_A$ を適当な正則行列 $\tilde{X} \in \mathbb{R}^{k \times k}$ と対角行列 $\tilde{D} \in \mathbb{R}^{k \times k}$ によって

$$C_B^{-1} C_A = \tilde{X} \tilde{D} \tilde{X}^{-1},$$

と対角化し、

$$E = \tilde{D}, V_0 = O_{(m-k) \times k}, V_1 = Z_1 S,$$

$$V_2 = Z_2 S^{-1}, V_3 = O_{(n-m-k) \times k},$$

$$Z_1 = \tilde{X}^{-T}, Z_2 = -C_B \tilde{X}$$

とおき, A_1, A_2 をそれぞれ $A + VEV^T$ の, B_1, B_2 をそれぞれ $B + VV^T$ の対角ブロックとおけば, $p = k$ の分解 (6) が構成できる. ただし, S は任意の正則な実対角行列である. なお, $C_B^{-1}C_A$ は非対称行列だが実三角行列であるので, 対角要素に重複がない場合には必ず実行列による対角化が可能である. 一方, C_B が正則だが $C_B^{-1}C_A$ が対角化不可能な場合には, $p = k$ の分解 (6) は存在しない.

行列 S を選択する基準の1つとして, A, B のブロック対角要素の修正量

$$f(S) := \|A(1:m, 1:m) - A_1\|_F$$

$$+ \|A(m+1:n, m+1:n) - A_2\|_F$$

$$+ \|B(1:m, 1:m) - B_1\|_F$$

$$+ \|B(m+1:n, m+1:n) - B_2\|_F$$

$$= \|Z_1 S^2 E Z_1^T\|_F + \|Z_2 S^{-2} E Z_2^T\|_F$$

$$+ \|Z_1 S^2 Z_1^T\|_F + \|Z_2 S^{-2} Z_2^T\|_F \quad (14)$$

をできるだけ小さくすることが考えられる. 帯行列の標準固有値問題の分割統治法では, 帯行列のブロック対角行列と摂動行列への分解において, ブロック対角要素への修正量が大きくなる場合に解の精度が悪化することが経験的に知られている. 我々は予備実験で, 帯行列の一般化固有値問題の分割統治法においても式 (14) が増大するほど解の残差ノルム

$$\|AX - BXA\|_F$$

が増大する傾向を確認しており, 一般化固有値問題においても修正量 (14) を小さく抑えることが解の精度悪化を防ぐために有効であると考えている. そこで, 式 (14) をヒューリスティックに最小化する S の決定方法の1つについて述べる. Z_1, Z_2 の i 番目の列ベクトルをそれぞれ $z_i^{(1)}, z_i^{(2)}$ とおくと, $f(S)$ について不等式

$$f(S)$$

$$\leq \left(\sum_{i=1}^k \|e_{i,i} s_{i,i}^2 z_i^{(1)} (z_i^{(1)})^T\|_F + \sum_{i=1}^k \|e_{i,i} s_{i,i}^{-2} z_i^{(2)} (z_i^{(2)})^T\|_F \right)$$

$$+ \left(\sum_{i=1}^k \|s_{i,i}^2 z_i^{(1)} (z_i^{(1)})^T\|_F + \sum_{i=1}^k \|s_{i,i}^{-2} z_i^{(2)} (z_i^{(2)})^T\|_F \right)$$

$$= \sum_{i=1}^k (|e_{i,i}| + 1)$$

$$\left(s_{i,i}^2 \|z_i^{(1)} (z_i^{(1)})^T\|_F + s_{i,i}^{-2} \|z_i^{(2)} (z_i^{(2)})^T\|_F \right)$$

$$= \sum_{i=1}^k (|e_{i,i}| + 1) \left(s_{i,i}^2 \|z_i^{(1)}\|_2^2 + s_{i,i}^{-2} \|z_i^{(2)}\|_2^2 \right)$$

が成り立つ. ここで, 右辺が最小化が最小化されるように S を決定すれば, 左辺の修正量 $f(S)$ は右辺の最小値以下に抑えられる. 右辺の最小化は, 最右辺の総和の各項を個別に最小化するように $s_{i,i}$ を決定すれば達成できる. したがって,

$$s_{i,i} = \sqrt{\|z_i^{(2)}\|_2 / \|z_i^{(1)}\|_2} \quad (i = 1, 2, \dots, k)$$

と S の対角要素を選べば, 修正量 (14) はヒューリスティックに最小化される.

C_B が特異な場合 (対角にゼロ要素を持つ場合) や $C_B^{-1}C_A$ が対角化不可能な場合, ゼロ要素の個数や固有値の重複度に応じて, $p > k$ の分解が可能である. 今, C_B の対角要素のうち, 第 i 対角要素のみがゼロであるとする. このとき, B と同じ帯構造を持つ $B' := B + \alpha(e_{m-k+i} + e_{m+i})(e_{m-k+i} + e_{m+i})^T$ を考えると, $C'_B := B'(m+1-k:m, m+1:m+k) = C_B + \alpha e_i e_i^T$ となり, C_B の第 i 対角要素に α を加えたものになっている. したがって, α が非ゼロかつ $C_B^{-1}C_A$ が重複する対角要素を持たない値にとられていれば,

$$A = A_1 \oplus A_2 - V'E'(V')^T, B' = B_1 \oplus B_2 - V'(V')^T$$

を満たす行列 $V' \in \mathbb{R}^{n \times k}$, 対角行列 $E' \in \mathbb{R}^{k \times k}$ が存在する. したがって, $V = [V', \sqrt{|\alpha|}(e_{m-k+i} + \text{sign}(\alpha)e_{m+i})]$, $E = E' \oplus 0$ とおけば, $p = k+1$ を満たす式 (6) が構成できる. C_B が対角に k' 数のゼロ要素を持つ場合には, k' 個だけ同様のランク 1 行列を加えることで, $p = k+k'$ を満たす分解が可能である. 次に, $C_B^{-1}C_A$ が対角化不可能な場合について考える. $C_B^{-1}C_A$ が対角化不可能な場合, $C_B^{-1}C_A$ は重複する対角要素を持つ. したがって, 第 i, j 対角要素のみが重複する場合, C_B の第 i 対角要素がゼロである場合と同様の摂動を加えることで $C_B^{-1}C_A$ は対角化可能な行列となり, $p = k+1$ の分解を得ることができる. 複数の重複が存在する場合にはその重複度に応じた個数のランク 1 行列を加えることで, 同様の分解が実現できる.

3.2.3 アルゴリズム

3.2.1 項で述べた原理に基づく, 提案法の手順を以下に示す.

- (i) 3.2.2.2 で述べた方法に基づき, $C_B^{-1}C_A$ の計算を計算し, $C_B^{-1}C_A$ の固有値問題を解いて E, V を求め, 式 (6) の A_1, A_2, B_1, B_2 を計算する.
- (ii) もとの行列より小さな固有値問題 (7) を解き, $X^{(0)}, D^{(0)}$ を計算する.
- (iii) 得られた $X^{(0)}$ により, (8) 右辺の $u_i^{(0)} = (X^{(0)})^T v_j$ ($i = 1, \dots, p$) を計算する.
- (iv) 以下の手順を $j = 1, 2, \dots, p$ について繰り返すことにより, 順番に $W^{(1)}, \dots, W^{(p)}$ を求め, 式 (10) の変換を繰り返して, 式 (11) に示される $p+1$ 個の行列の積の計算を進める.

Algorithm 1 Divide-and-conquer algorithm for banded generalized eigenvalue problems

```

1:  $A \rightarrow A_1 \oplus A_2 - VEV^T, B \rightarrow B_1 \oplus B_2 - VV^T$ 
2: Solve  $(X_i^{(0)})^T(A_i - \lambda B_i)X_i^{(0)} = D_i^{(0)} - \lambda I \quad (i = 1, 2)$ 
3:  $X^{(0)} := X_1^{(0)} \oplus X_2^{(0)}, D^{(0)} := D_1^{(0)} \oplus D_2^{(0)}, E^{(0)} := E$ 
4:  $\mathbf{u}_i^{(0)} \leftarrow (X^{(0)})^T \mathbf{v}_i \quad (i = 1, 2, \dots, p)$ 
5: for  $j = 1, 2, \dots, p$  do
6:   Solve  $(W^{(j)})^T \{ [D^{(j-1)} - e_{j,j} \mathbf{u}_j^{(j-1)} (\mathbf{u}_j^{(j-1)})^T] - \lambda [I - \mathbf{u}_j^{(j-1)} (\mathbf{u}_j^{(j-1)})^T] \} W^{(j)} = D^{(j)} - \lambda I$ 
7:    $X^{(j)} \leftarrow X^{(j-1)} W^{(j)}$ 
8:    $\mathbf{u}_i^{(j)} \leftarrow (W^{(j)})^T \mathbf{u}_i^{(j-1)} \quad (i = j + 1, j + 2, \dots, p)$ 
9: end for
10:  $X := X^{(p)}, \Lambda := D^{(p)}$ 

```

- (a) 一般化固有値問題 (9), (10) を, 文献 [2] に示された反復法により解き, $D^{(j)}, W^{(j)}$ を求める.
- (b) 行列積 $X^{(j)} \leftarrow X^{(j-1)} W^{(j)}$ を計算する.
- (c) $\mathbf{u}_i^{(j)} \leftarrow (W^{(j)})^T \mathbf{u}_i^{(j-1)} \quad (i = j + 1, j + 2, \dots, p)$ を計算する.

上記の手順により, 最終的に式 (1) の固有値, 固有ベクトル $\Lambda = D^{(p)}, X = X^{(p)}$ が計算できる. 以上をまとめたものを, **Algorithm 1** に示す. なお, $k = 1$ の場合 (三重対角行列の場合) に, 提案法は Elsner らの分割統治法と一致し, その意味で提案法は Elsner らの分割統治法の拡張となっている.

次に, アルゴリズムの演算量, 演算の種類について考える. 1 行目は, $C_B^{-1} C_A$ の計算, $C_B^{-1} C_A$ の固有値問題の求解, A_1, A_2, B_1, B_2 の計算から構成され, いずれも Level-3 型の演算によって求められ, 演算量は $O(k^3)$ となる. 4, 8 行目は行列積として実行され, 演算量はそれぞれ $O(kn^2)$ となる. 6 行目では, 反復法による secular 方程式の求解が支配的な演算となり, 文献 [2] で述べられる反復法が少ない反復回数で収束すれば, 演算量は $O(pn^2)$ となる. 7 行目の行列積は, $j = 1$ のとき $X^{(0)}$ のブロック対角性を考慮すれば演算量は $2nm^2 + 2n(n - m)^2$ で計算でき, $j \geq 2$ のときは密行列どうしの積となり演算量は $2n^3$ となる.

つねに $m \simeq n/2$ として行列を中心付近で分割して, 2 行目の次数の低い固有値問題を提案法によって再帰的に問題を解く場合の各ステップの演算量および支配的となる演算の種類をまとめたものを表 3 に示す. このときの総演算量は $(4/3)(2p - 1)n^3 + O(p^2n^2)$ となる. $p = k$ となるように 1 行目の分解を行う場合には, 総演算量は $(4/3)(2k - 1)n^3 + O(k^2n^2)$ となる. $k \ll n$ である場合, 総演算量の第 2 項は無視できるため, 演算のほとんどすべてが 7 行目の行列積として実行されることになる.

3.3 アルゴリズムの比較

表 1, 表 2, 表 3 にまとめられた, 3 つのアルゴリズムの演算量と演算の種類について比較する. まず, 提案法で $p = k$ を満たす分解が可能であると仮定すると, 帯構

表 3 提案法の演算量内訳および演算の種類

Table 3 The number of FLOPs and the computational pattern in the proposed method.

	演算量	種類
1 行目	$C_B^{-1} C_A$ の計算	$O(k^3)$ Level-3
	$C_B^{-1} C_A$ の固有値問題	$O(k^3)$ Level-3
	A_1, A_2, B_1, B_2 の計算	$O(k^3)$ Level-3
4 行目		$O(pn^2)$ Level-3
6 行目		$O(pn^2)$ 反復法
7 行目		$(4/3)(2p - 1)n^3$ Level-3
8 行目		$O(p^2k^2)$ Level-3

造を用いる従来法, 密行列として扱う従来法, 提案法の演算量はそれぞれ, $(41/6)n^3 + O(kn^2)$ ($k = 1$ の場合のみ $(29/6)n^3 + O(kn^2)$), $(22/3)n^3$, $(4/3)(2k - 1)n^3 + O(k^2n^2)$ となる. 帯構造を用いる従来法の演算量は, 最高次の項は, 三重対角化ステップが不要な $k = 1$ の場合を例外として, k に対して一定であり, 低次の項のみが k にともなって増大する. また, 密行列として扱う従来法の演算量は k の影響を受けない. これに対して, 提案法の演算量は最高次の項が k に対して線形に増大しており, 従来法と比べて半帯幅 k に対して強い依存性がある. $k \ll n$ を仮定して演算量の最高次の項のみを比較すると, $k \leq 3$ では提案法の演算量が最小になり, $k \geq 4$ では最大となる. また, それぞれ Level-3 演算として実行される演算量は $(10/3)n^3$, $(20/3)n^3$, $(4/3)(2k - 1)n^3$ であり, 提案法のみ演算量のほとんどすべてが Level-3 型演算として実行されることが分かる.

半帯幅 $k \leq 3$ では, 提案法が最も演算量が少なく, 性能面でも有利であるため, 提案法の実行時間は 2 つの従来法よりも短くなると考えられる. 一方, $k \geq 4$ では, 提案法は, 問題を密行列として問題を扱う従来法と比較して, Level-2 演算の演算量が $(2/3)n^3$ 少なく, Level-3 演算が $(8/3)kn^3 - 8n^3$ だけ多い. したがって, ある値以上の半帯幅の行列に対しては, 従来法の実行時間がより短くなると予想される.

4. 数値実験

マルチコア CPU 上で従来法および提案法 ($k = 1$ の場合には Elsner らの手法に一致) の精度および性能を評価する. 標準固有値問題を經由する従来法の実装には, それぞれ, Intel による LAPACK 実装 Intel MKL [10] に含まれる DSBGVD, DSYGVD ルーチンを使用した. また, 実行時間の内訳を評価するため, 上記の実装とは別に, Netlib による LAPACK 実装の DSBGVD および DSYGVD 各ルーチンのソースコードを修正して時間計測機能を付加した実装を作成した. ただし, 時間計測付き実装で内部的に呼び出される LAPACK ルーチン (DPBSTF や DSBGST など) は Intel による実装 (Intel MKL) を使用している. ま

表 4 計算機環境

Table 4 Computational environment.

CPU	Intel Xeon E5-2660 (8 コア, 140.8 GFLOPS, Hyper-Threading 無効) × 2 ソケット
メモリ	64 GB
コンパイラ	Intel Fortran Compiler 14.0.0
LAPACK	Intel Math Kernel Library 11.1 および Netlib LAPACK 3.5.0
BLAS	Intel Math Kernel Library 11.1

た、精度評価を行う際には、問題を帯行列の標準固有値問題に変換し標準固有値問題を QR 法によって解く解法の、Intel MKL 実装 (DSBGV) も使用した。提案法は、行列積などの基本行列計算については BLAS ライブラリを用いて実装し、Algorithm 1 は Fortran および OpenMP によって独自に実装した。ただし、Algorithm 1 の 6 行目は、 $W^{(j)}$ が陽に計算されるように実装した。また、分割点 m は $m = n/2$ と問題を 2 等分するようを選び、Algorithm 1 の 2 行目の小問題については、提案法を再帰的に適用して解いた。ただし、行列の次数が 200 未満になった問題については、LAPACK の DSBGVD を適用して解いた。

テスト行列は、 A が半帯幅 k の実対称行列、 B が半帯幅 k の実対称正定値行列を満たすようにするため、以下のよう

$$a_{i,j} = \begin{cases} [0,1) \text{ 乱数} & (|i-j| \leq k) \\ 0 & (\text{otherwise}) \end{cases},$$

$$b_{i,j} = \begin{cases} 2k & (i=j) \\ [0,1) \text{ 乱数} & (1 \leq |i-j| \leq k) \\ 0 & (\text{otherwise}) \end{cases}.$$

このように生成される問題は、固有値分布がクラスタを持ちにくく、絶対値のきわめて小さな固有値を持つ確率が低い。分割統治法を適用する際に精度面で有利に働く可能性がある。しかしながら、任意の固有値分布を持つ帯構造、正定値性を備えたテスト行列 A, B を生成する方法が確立されていないため、本実験では上記の方法で生成されたテスト行列を使用する。

実験で使用する計算機環境は表 4 のとおりである。

4.1 精度評価

本節では Intel MKL のみを使用した従来法の実装 (DSBGV, DSBGVD, DSYGVD) および提案法の各実装について精度評価を行う。 $n = 10240, k = 1$ および $k = 2$ としてテスト行列を作成し、求めた近似固有対の精度を、以下に定義される相対残差、近似固有ベクトルの B -直交性、QR 法を利用する従来法 (DSBGV) を基準にした解法間の近似固有値の最大相対誤差

$$\frac{\|AX - BX\Lambda\|_F}{\|A\|_F}, \frac{\|X^T BX - I\|_F}{\sqrt{n}},$$

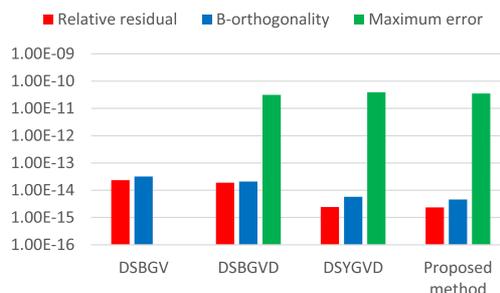


図 2 固有値, 固有ベクトルの精度 ($n = 10240, k = 1$)

Fig. 2 The accuracy of the computed eigenvalues and eigenvectors ($n = 10240, k = 1$).

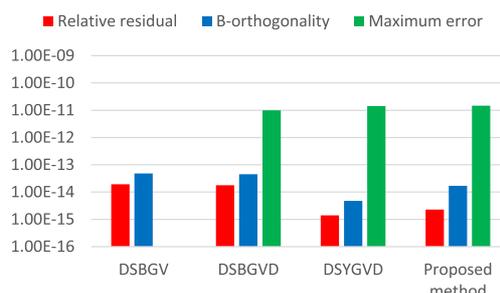


図 3 固有値, 固有ベクトルの精度 ($n = 10240, k = 2$)

Fig. 3 The accuracy of the computed eigenvalues and eigenvectors ($n = 10240, k = 2$).

$$\max_j \frac{|\lambda_j^{(\text{DSBGV})} - \lambda_j|}{|\lambda_j^{(\text{DSBGV})}|}$$

によって評価する。ここで、 $\lambda_j^{(\text{DSBGV})}$ ($j = 1, 2, \dots, n$) は DSBGV によって求められた近似固有値である。

評価結果を図 2, 図 3 に示す。いずれの解法も同程度の相対残差, B -直交性, 固有値の最大相対誤差を示しており、提案法は、三重対角行列, 五重対角行列のいずれに対しても実用的な精度の解が得られていることが確認できる。

4.2 性能評価

半帯幅を $k = 1, 2$, 行列の次数 $n = 10240$ として、各実装を 1, 2, 4, 8, 16 スレッドで実行し、実行時間およびその内訳を調べる。 $k = 1$ の場合の各実装の評価結果を図 4, 図 5, 図 6 に、 $k = 2$ の場合の結果を図 7, 図 8, 図 9 に示す。

半帯幅の異なる 2 つのテスト問題の実行結果を比較すると、DSBGVD (図 4, 図 7) では $k = 1$ の場合に比べて $k = 2$ での実行時間が増大しているが、これは $k = 1$ の場合のみスキップされる三重対角化ステップ (DSBTRD ルーチン) の実行時間が $k = 2$ では加わったことが大きく影響している。また、DSYGVD (図 5, 図 8) の実行時間はほとんど半帯幅 k の影響を受けないことが確認できる。提案法は、 $k = 2$ のときの実行時間 (図 6) が、 $k = 1$ の Elsner

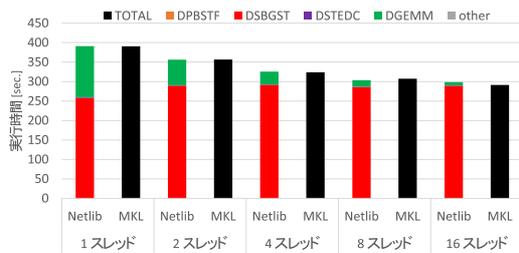


図 4 DSBGVD の実行時間 ($n = 10240, k = 1$)

Fig. 4 The execution time of DSBGVD ($n = 10240, k = 1$).

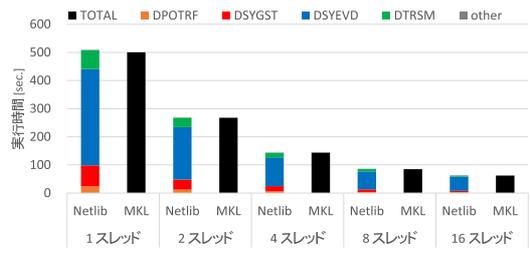


図 8 DSYGVD の実行時間 ($n = 10240, k = 2$)

Fig. 8 The execution time of DSYGVD ($n = 10240, k = 2$).

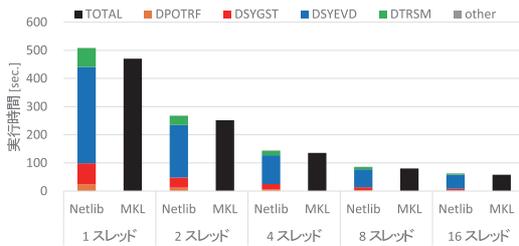


図 5 DSYGVD の実行時間 ($n = 10240, k = 1$)

Fig. 5 The execution time of DSYGVD ($n = 10240, k = 1$).

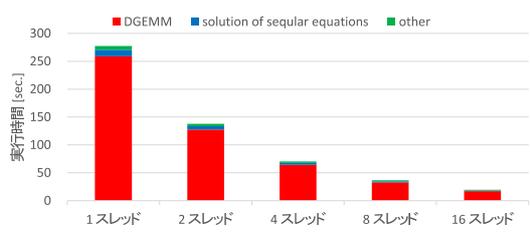


図 9 提案法の実行時間 ($n = 10240, k = 2$)

Fig. 9 The execution time of the proposed method ($n = 10240, k = 2$).

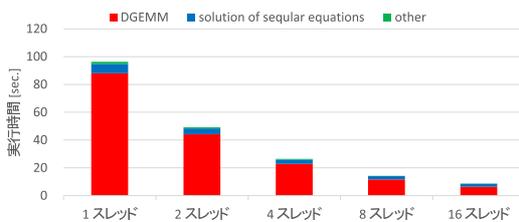


図 6 提案法の実行時間 ($n = 10240, k = 1$)

Fig. 6 The execution time of the proposed method ($n = 10240, k = 1$).

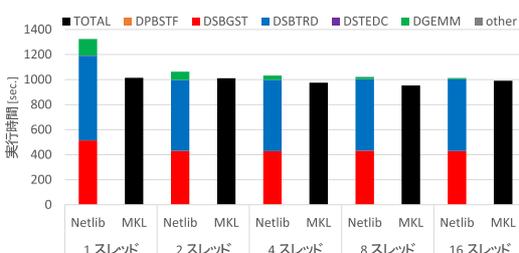


図 7 DSBGVD の実行時間 ($n = 10240, k = 2$)

Fig. 7 The execution time of DSBGVD ($n = 10240, k = 2$).

の手法に一致する場合の実行時間 (図 9) に比べて大きく増大しており、実行時間の強い k 依存性が確認できる。

次に、スレッド数の増大による加速についてみると、帯構造を用いる従来法の実装はほとんど加速されないことが確認できる。また、DSYGVD では、スレッド数の増加によって性能は向上しているものの、Level-2 演算を含む DSYEVD が性能上のボトルネックになり、16 スレッド実行時の逐次実行時に対する加速率は $k = 2$ の場合に 9.0 倍となっている。一方、提案法は順調にスケールし、16 スレッド実行時の加速率は $k = 2$ の場合に 14.0 倍となっている。従来法に対する加速率は、並列実行時に逐次実行時

よりも高く、提案法のマルチコア計算機における優位性を確認できる。

また、いずれのスレッド数で比較した場合でも、提案法は $k = 1, 2$ の両問題で実行時間が従来法より短く、 $k \leq 3$ では従来法より高速であるという 3.3 節の予想に合致する結果となった。

5. おわりに

三重対角行列の一般化固有値問題向け分割統治法をもとに、帯行列の一般化固有値問題向け分割統治法を提案した。提案法の演算量は問題の半帯幅 k に比例して増大するが、 $k \leq 3$ では、帯行列の標準固有値問題を經由して解く従来法と比べても演算量が少ない。また、演算のほとんどが行列積として実行される。次数 10240 の三重対角行列、五重対角行列の一般化固有値問題をマルチコア計算機上で解いて性能を評価し、提案法は従来法に対して三重対角行列では 6.6 倍、五重対角行列では約 3.2 倍高速であり、その優位性が確認できた。

今後の課題としては、任意の固有値分布のテスト行列生成手法の確立後、固有値が密集する問題や、絶対値の小さな固有値が存在する問題に対する提案法の精度評価があげられる。

謝辞 本稿に対して数々の貴重なコメントをいただいた匿名の査読者に深い感謝の意を表す。本稿執筆にあたり多くの有用な意見をいただいた深谷猛氏 (北海道大学) に深く感謝する。本稿の図の作成の一部を支援していただいた椋木大地氏 (理化学研究所計算科学研究機構) にお礼申し上げる。

本研究は、科学技術振興機構戦略的創造研究推進事業研

究領域「ポストベタスケール高性能計算に資するシステムソフトウェア技術の創出」における研究課題「ポストベタスケールに対応した階層モデルによる超並列固有値解析エンジンの開発」の援助を受けている。

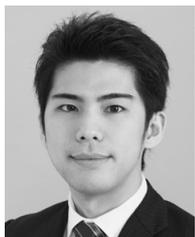
参考文献

- [1] Du, L. and Imakura, A.: Reducing Two Symmetric Matrices to Band Form by Congruence Transformations, 日本応用数学会 2013 年度年会 予稿集, pp.66-67 (2013).
- [2] Elsner, L., Fasse, A. and Langmann, E.: A divide-and-conquer method for the tridiagonal generalized eigenvalue problem, *Journal of computational and applied mathematics*, Vol.86, No.1, pp.141-148 (1997).
- [3] Beattie, C., Ribbens, C.J., Dongarra, J., Kennedy, K., Mesina, P., Sorensen, D. and Voight, R.: Parallel solution of a generalized symmetric matrix eigenvalue problem, *Proc. 5th SIAM Conference on Parallel Processing for Scientific Computing*, pp.16-21, Society for Industrial and Applied Mathematics (1991).
- [4] Borges, C.F. and Gragg, W.B.: A parallel divide and conquer algorithm for the generalized real symmetric definite tridiagonal eigenproblem, Technical report, DTIC Document (1992).
- [5] Gu, M. and Eisenstat, S.C.: A stable and efficient algorithm for the rank-one modification of the symmetric eigenproblem, *SIAM Journal on Matrix Analysis and Applications*, Vol.15, No.4, pp.1266-1276 (1994).
- [6] Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J., Dongarra, J., Du Croz, J., Greenbaum, A., Hammarling, S., McKenney, A. and Sorensen, D.: *LAPACK Users' Guide, 3rd edition*, Society for Industrial and Applied Mathematics, Philadelphia, PA (1999).
- [7] Arbenz, P.: Divide and conquer algorithms for the bandsymmetric eigenvalue problem, *Parallel computing*, Vol.18, No.10, pp.1105-1128 (1992).
- [8] Gansterer, W.N., Schneid, J. and Ueberhuber, C.W.: A Divide-and-Conquer Method for Symmetric Banded Eigenproblems-Part I: Theoretical Results (1999).
- [9] Pham, H.P., Imamura, T., Yamada, S. and Machida, M.: Novel approach in a divide and conquer algorithm for eigenvalue problems of real symmetric band matrices, *Proc. Joint Int. Conf. Supecomputing in Nuclear Applications + Monte Carlo 2010 (SNA+MC2010)*, pp.17-21 (2010).
- [10] Intel Math Kernel Library (online): available from (<https://software.intel.com/en-us/intel-mkl>) (accessed 2015-01-15).



今村 俊幸 (正会員)

1969 年生。1996 年京都大学大学院工学研究科応用システム科学専攻博士後期課程単位認定退学。同年日本原子力研究所入所。計算科学技術推進センターにて途切れのない思考を支援する並列処理基本システム STA の開発に従事。2001 年から 2002 年までシュツットガルト大学 HLRS 招聘研究員。2003 年より電気通信大学電気通信学部講師。2006 年助教授, 2007 年准教授。2010 年より同大学情報理工学研究科准教授。2012 年より理化学研究所計算科学研究機構大規模並列数値計算技術研究チームチームリーダー。HPC とその周辺ソフトウェア, 性能自動チューニング, 高性能固有値計算ライブラリの研究開発に従事。博士 (工学)。平成 18 年度山下記念研究賞受賞, 2011 年度日本応用数学会業績賞受賞。日本応用数学会, SIAM 各会員。



廣田 悠輔 (正会員)

1985 年生。2010 年名古屋大学大学院工学研究科計算理工学専攻 (博士前期課程) 修了。2013 年神戸大学大学院システム情報学研究科計算科学専攻 (博士後期課程) 修了, 博士 (計算科学)。2014 年 4 月より理化学研究所計算科学研究機構特別研究員。並列計算機向けの行列計算アルゴリズムの研究開発に従事。