

# 両側ヤコビ法の実装について

荒木 翔<sup>1,a)</sup> 高田 雅美<sup>2,b)</sup> 木村 欣司<sup>3,c)</sup> 中村 佳正<sup>1,d)</sup>

**概要**：特異値分解のためのヤコビ法は、すべての特異値と特異ベクトルを高い精度で計算することができる。ヤコビ法には、片側および両側ヤコビ法が提案されている。片側ヤコビ法は、LAPACK に既に実装されている。両面ヤコビ法の実装には、改善できる部分がいまだたくさん残っている。そこで、本稿では、両側ヤコビ法を改良する。

## On an Implementation of Two-Sided Jacobi Method

### 1. はじめに

櫻井-杉浦法は、対称行列と正定対称行列からなる一般化固有値問題の一部の固有値と固有ベクトルを計算可能な解法であり、行列の列空間を利用する。列空間を計算するには、長方形行列を特異値分解する必要がある。一般的に、与えられた行列は、前処理法としてハウスホルダー変換を利用して二重対角行列に変換される。[2]には、二重対角行列の場合に採用できる列空間の計算法が提案されている。特異値を計算するDQDS法と特異ベクトルを計算するOQDS法を組み合わせた方法である。

計算コストを削減し精度を向上させるために、櫻井-杉浦法は、今倉らにより改良された[10]。改良された櫻井-杉浦法は、左右両方の特異ベクトルを必要とする。列空間のみが必要な場合、OQDS法は高い精度を達成できる。しかし、OQDS法によって下二重対角行列の左特異ベクトルを計算する場合、与えられた行列サイズの2倍の大きさの行列を利用して計算する必要がある。その結果、下二重対角行列の計算された左特異ベクトルが高い直交性を持つことは、保証されない。したがって、改良された櫻井-杉浦法を高精度に実装するためには、特異値と左右の特異ベクトルのす

べてを高精度に計算可能な特異値分解のための方法を確立する必要がある。

### 2. 両側ヤコビ法による特異値分解

#### 2.1 計算の対象となる行列

固有値分解のための両側ヤコビ法は、実対称行列の固有値と固有ベクトルを計算できる。より正確には、対象となる行列をエルミート行列に拡張することも可能である。さらに、特異値分解のための両側ヤコビ法は、任意のサイズの複素行列に対して計算を実行するように設計することも可能である。しかし、本稿では、実数の上三角行列の場合のみを検討する。長方形行列の場合、QR分解およびLQ分解を使用して前処理を行うと、長方形行列の特異値分解の問題は、上三角行列の特異値分解の問題に単純化できる。さらに、我々の方法を、複素上三角行列に拡張することも容易に可能であるため、この論文では、実上三角行列のみを扱う。

#### 2.2 概略

$J^{(i)}$ ,  $K^{(i)}$ ,  $N^{(i)}$ ,  $M^{(i)}$  は、回転行列の積とする。 $R^{(i)}$  は、上三角行列、 $L^{(i)}$  は、下三角行列である。両側ヤコビ法による特異値分解は、式(1)、(2)により、計算される。

$$K^{(i)}R^{(i)}J^{(i)} = L^{(i)}, \quad (1)$$

$$N^{(i)}L^{(i)}M^{(i)} = R^{(i+1)}, \quad i = 0, 1, \dots \quad (2)$$

反復を繰り返すと、 $R^{(i)}$  と  $L^{(i)}$  は、対角行列に収束する。収束の段階において、左特異ベクトル  $U$  と右特異ベクトル  $V$  は、次のように計算することができる。

<sup>1</sup> 京都大学  
Kyoto University, Kyoto, Kyoto 606-8501, JAPAN  
<sup>2</sup> 奈良女子大学  
Nara Women's University, Nara, Nara 630-8506, JAPAN  
<sup>3</sup> 福井大学  
University of Fukui, Fukui, Fukui 910-8507, JAPAN  
a) araki@amp.i.kyoto-u.ac.jp  
b) takata@ics.nara-wu.ac.jp  
c) kkimur@u-fukui.ac.jp  
d) ynaka@i.kyoto-u.ac.jp

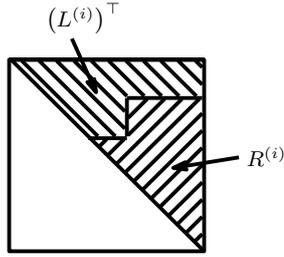


図 1 Space sharing of the upper triangular matrix

$$U = \left( K^{(0)} \right)^{\top} \left( N^{(0)} \right)^{\top} \left( K^{(1)} \right)^{\top} \left( N^{(1)} \right)^{\top} \dots \left( K^{(m-1)} \right)^{\top} \left( N^{(m-1)} \right)^{\top}, \quad (3)$$

$$V = J^{(0)} M^{(0)} J^{(1)} M^{(1)} \dots J^{(m-1)} M^{(m-1)}, \quad (4)$$

$m$  は、収束段階における反復の回数である。式 (3) と式 (4) における行列の乗算は、ギブンス回転によって実装される。図 1 は、 $R^{(i)}$  と  $L^{(i)}$  が、上三角行列の中に保持されることを表している。ここで、 $R^{(i)}$  と  $L^{(i)}$  は、同じメモリ空間において計算可能である。

式 (7) が示すように、回転行列  $P$  と  $Q$ ,

$$P = \begin{pmatrix} 1 & 0 & \cdots & \cdots & 0 \\ 0 & c_1 & \cdots & s_1 & \vdots \\ \vdots & \vdots & 1 & \vdots & \vdots \\ \vdots & -s_1 & \cdots & c_1 & 0 \\ 0 & \cdots & \cdots & 0 & 1 \end{pmatrix}, \quad (5)$$

$$Q = \begin{pmatrix} 1 & 0 & \cdots & \cdots & 0 \\ 0 & c_2 & \cdots & -s_2 & \vdots \\ \vdots & \vdots & 1 & \vdots & \vdots \\ \vdots & s_2 & \cdots & c_2 & 0 \\ 0 & \cdots & \cdots & 0 & 1 \end{pmatrix}, \quad (6)$$

を利用して、 $R_{j,k}$  を 0 に変換することができる、

$$P \times \begin{pmatrix} \ddots & \cdots & \cdots & \cdots & \cdots \\ \vdots & R_{j,j} & \cdots & R_{j,k} & \vdots \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \vdots & 0 & \cdots & R_{k,k} & \vdots \\ \cdots & \cdots & \cdots & \cdots & \ddots \end{pmatrix} \times Q = \begin{pmatrix} \ddots & \cdots & \cdots & \cdots & \cdots \\ \vdots & \hat{R}_{j,j} & \cdots & 0 & \vdots \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \vdots & 0 & \cdots & \hat{R}_{k,k} & \vdots \\ \cdots & \cdots & \cdots & \cdots & \ddots \end{pmatrix}. \quad (7)$$

式 (7) を繰り返すことで、 $R^{(i)}$  は、 $L^{(i)}$  に変換される。しかし、式 (7) は、 $R^{(i)}$  から  $L^{(i)}$  を計算する方法を示しているのではないことを注意する。よって、式 (7) に、 $R_{j,j}^{(i)}$ 、 $R_{j,k}^{(i)}$ 、 $R_{k,k}^{(i)}$ 、 $L_{j,j}^{(i)}$ 、 $L_{k,k}^{(i)}$  は、現れない。 $P$  と  $Q$  は、回転行列であるため、 $c_1 = \cos \theta_1$ 、 $s_1 = \sin \theta_1$ 、 $c_2 = \cos \theta_2$ 、 $s_2 = \sin \theta_2$  を満たす  $\theta_1$  と  $\theta_2$  が存在する。以下、値が変化する要素部分のみについて述べる。

$$\begin{pmatrix} c_1 & s_1 \\ -s_1 & c_1 \end{pmatrix} \begin{pmatrix} R_{j,j} & R_{j,k} \\ 0 & R_{k,k} \end{pmatrix} \begin{pmatrix} c_2 & -s_2 \\ s_2 & c_2 \end{pmatrix} = \begin{pmatrix} \hat{R}_{j,j} & 0 \\ 0 & \hat{R}_{k,k} \end{pmatrix}. \quad (8)$$

$R^{(i)}$  から  $L^{(i)}$  を計算するために、式 (8) の操作は、何度も行われる。この反復において、非対角成分を 0 にするオーダーリングと収束の判定条件が必要となる。詳細については、2.3 節で述べる。 $L^{(i)}$  から  $R^{(i+1)}$  を求める場合も同様である。 $R_{j,j}$ 、 $R_{j,k}$ 、 $R_{k,k}$  から  $c_1$ 、 $s_1$ 、 $c_2$ 、 $s_2$ 、 $\hat{R}_{j,j}$ 、 $\hat{R}_{k,k}$  を求める方法については、2.4 節、2.5 節、2.6 節で説明する。

### 2.3 オーダーリングと収束の判定条件

上三角行列  $R^{(i)}$  の非対角成分を 0 にすると、下三角行列  $L^{(i)}$  に 0 でない成分が現れる。0 にする場合の詳細は、次の通りである： $|R_{1,1}^{(0)}| \geq |R_{n,n}^{(0)}|$  ならば、 $R^{(i)}$  から  $L^{(i)}$  を得るために、次のオーダーリングを用いる。(1,2)(1,3), ..., (1,n), (2,3), (2,4), ..., (n-2, n-1), (n-2, n), (n-1, n). そして、 $L^{(i)}$  から  $R^{(i+1)}$  を得るために、次のオーダーリングを用いる。(2,1)(3,1), ..., (n,1), (3,2), (4,2), ..., (n-1, n-2), (n, n-2), (n, n-1).  $|R_{1,1}^{(0)}| < |R_{n,n}^{(0)}|$  ならば、 $R^{(i)}$  から  $L^{(i)}$  を得るために、次のオーダーリングを用いる。(n-1, n)(n-2, n), ..., (1, n), (n-2, n-1), (n-3, n-1), ..., (1,3), (1,2). そして、 $L^{(i)}$  から  $R^{(i+1)}$  を得るために、次のオーダーリングを用いる。(n, n-1)(n, n-2), ..., (n,1), (n-1, n-2), (n-1, n-3), ..., (3,1), (2,1). 両側ヤコビ法により上記の計算を続けると、すべての非対角成分は、0 に収束する。数値計算の観点では、非対角成分が完全に 0 になる前に 0 と判定しなければ、丸め誤差が蓄積する。その判定基準として、式 (9) を利用する、

$$|R_{j,k}| \leq \varepsilon \sqrt{|R_{j,j}|} \times \sqrt{|R_{k,k}|}. \quad (9)$$

すべての非対角成分を 0 と見なしてよくなると、反復計算を終了する。

### 2.4 逆正接関数を用いた実装

特異値分解のための両側ヤコビ法は、特異値分解のための片側ヤコビ法に比べ、 $c_1, s_1, c_2, s_2$  の値を決定する際に、多数の演算が必要になるという欠点がある。数値計算にお

いて、多数の演算が必要になることは、多量の誤差の混入を招く。この方法では、 $\tan^{-1}, \theta_1, \theta_2$  を利用することで、少ない演算で  $c_1, s_1, c_2, s_2$  の値を決定できるため、多量の誤差の混入を招く欠点を克服できる。

$$\alpha = \tan^{-1} \left( \frac{R_{j,k}}{R_{j,j} - R_{k,k}} \right), \quad \beta = \tan^{-1} \left( \frac{-R_{j,k}}{R_{j,j} + R_{k,k}} \right), \quad (10)$$

$$\theta_1 = \frac{1}{2}(\alpha + \beta), \quad \theta_2 = \frac{1}{2}(\alpha - \beta), \quad (11)$$

$$c_1 = \cos(\theta_1), \quad s_1 = \sin(\theta_1), \quad c_2 = \cos(\theta_2), \quad s_2 = \sin(\theta_2). \quad (12)$$

ここで、 $-\frac{\pi}{2} \leq \theta_1 \leq \frac{\pi}{2}$  と  $-\frac{\pi}{2} \leq \theta_2 \leq \frac{\pi}{2}$  が成立する。さらに、

$$u = c_1 + c_2, \quad (13)$$

$$\hat{R}_{j,j} = \frac{R_{j,j} + \frac{s_2}{u} \times R_{j,k}}{u}, \quad \hat{R}_{k,k} = \frac{R_{k,k} - \frac{s_1}{u} \times R_{j,k}}{u}. \quad (14)$$

を得る。二重下線部は、融合積和演算により実装する。 $S_q = \sum_{i=0}^q x_i$  に対して、 $|x_0|$  が  $|x_i| (i = 1, \dots, q)$  よりも、極めて大きい場合、 $T_q = \sum_{i=1}^q x_i$ ,  $S_q = x_0 + T_q$  と計算する。この工夫を、式 (14) の計算において活用する。

## 2.5 Rutishauser の実装法

固有値分解を計算するためのヤコビ法において、Rutishauser の実装法 [15] が提案されている。Rutishauser の実装法を両側ヤコビ法の実装に応用すると、計算誤差を減らす目的で、融合積和演算を活用できる。

$$h_1 = \frac{R_{j,j} - R_{k,k}}{R_{j,k}}, \quad f_1 = \sqrt{1 + (h_1)^2}, \quad t_1 = \frac{1}{h_1 \pm f_1}, \quad (15)$$

$$h_2 = -\frac{R_{j,j} + R_{k,k}}{R_{j,k}}, \quad f_2 = \sqrt{1 + (h_2)^2}, \quad t_2 = \frac{1}{h_2 \pm f_2}. \quad (16)$$

ここで、 $f_1$  と  $f_2$  の符号は、 $h_1$  と  $h_2$  の符号と合わせる。

$$v_1 = \underline{1 - t_1 \times t_2}, \quad w_1 = t_1 + t_2, \quad (17)$$

$$u_1 = \max(|v_1|, |w_1|), \quad (18)$$

$$c_1 = \frac{\frac{v_1}{u_1}}{\sqrt{\left(\frac{v_1}{u_1}\right)^2 + \left(\frac{w_1}{u_1}\right)^2}}, \quad s_1 = \frac{\frac{w_1}{u_1}}{\sqrt{\left(\frac{v_1}{u_1}\right)^2 + \left(\frac{w_1}{u_1}\right)^2}}, \quad (19)$$

## Algorithm 1 ギブンス回転を用いた実装法

---

```

1:  $h_1 \leftarrow R_{j,j} - R_{k,k}$ 
2:  $g_1 \leftarrow |R_{j,k}|$ 
3:  $f_1 \leftarrow |h_1| + \sqrt{h_1^2 + R_{j,k}^2}$  The Givens rotation is adopted in
   underlined part
4:  $g_1 \leftarrow \text{SIGN}(R_{j,k}, R_{j,k}/h_1)$ 
5:  $h_2 \leftarrow R_{j,j} + R_{k,k}$ 
6:  $f_2 \leftarrow |h_2| + \sqrt{h_2^2 + R_{j,k}^2}$  The Givens rotation is adopted in
   underlined part
7:  $g_2 \leftarrow \text{SIGN}(R_{j,k}, -R_{j,k}/h_2)$ 
8: if  $f_1 \geq f_2$  then
9:    $t_1 \leftarrow g_1/f_1$ 
10:   $\hat{c}_1 \leftarrow \underline{-t_1 \times g_2 + f_2}$ 
11:   $\hat{s}_1 \leftarrow \underline{t_1 \times f_2 + g_2}$ 
12:  Compute  $c_1$  and  $s_1$  using the Givens rotation for  $x \leftarrow \hat{c}_1$ 
   and  $y \leftarrow \hat{s}_1$ 
13:   $\hat{c}_2 \leftarrow \underline{t_1 \times g_2 + f_2}$ 
14:   $\hat{s}_2 \leftarrow \underline{t_1 \times f_2 - g_2}$ 
15:  Compute  $c_2$  and  $s_2$  using the Givens rotation for  $x \leftarrow \hat{c}_2$ 
   and  $y \leftarrow \hat{s}_2$ 
16: else
17:   $t_2 \leftarrow g_2/f_2$ 
18:   $\hat{c}_1 \leftarrow \underline{-g_1 \times t_2 + f_1}$ 
19:   $\hat{s}_1 \leftarrow \underline{f_1 \times t_2 + g_1}$ 
20:  Compute  $c_1$  and  $s_1$  using the Givens rotation for  $x \leftarrow \hat{c}_1$ 
   and  $y \leftarrow \hat{s}_1$ 
21:   $\hat{c}_2 \leftarrow \underline{g_1 \times t_2 + f_1}$ 
22:   $\hat{s}_2 \leftarrow \underline{-f_1 \times t_2 + g_1}$ 
23:  Compute  $c_2$  and  $s_2$  using the Givens rotation for  $x \leftarrow \hat{c}_2$ 
   and  $y \leftarrow \hat{s}_2$ 
24: end if

```

---

$$v_2 = \underline{1 + t_1 \times t_2}, \quad w_2 = t_1 - t_2, \quad (20)$$

$$u_2 = \max(|v_2|, |w_2|), \quad (21)$$

$$c_2 = \frac{\frac{v_2}{u_2}}{\sqrt{\left(\frac{v_2}{u_2}\right)^2 + \left(\frac{w_2}{u_2}\right)^2}}, \quad s_2 = \frac{\frac{w_2}{u_2}}{\sqrt{\left(\frac{v_2}{u_2}\right)^2 + \left(\frac{w_2}{u_2}\right)^2}}, \quad (22)$$

$$c = c_1 + c_2, \quad (23)$$

$$\hat{R}_{j,j} = \frac{R_{j,j} + \frac{s_2}{c} \times R_{j,k}}{c}, \quad \hat{R}_{k,k} = \frac{R_{k,k} - \frac{s_1}{c} \times R_{j,k}}{c}, \quad (24)$$

を得ることができる。二重下線部は、融合積和演算により実装する。

## 2.6 ギブンス回転を用いた実装法

$c_1, s_1, c_2, s_2$  を計算するために、Algorithm 1 を適用する。Algorithm 1 で必要となる ギブンス回転の実装は、[1] を適用する。関数  $\text{SIGN}(A, B)$  は、 $A$  の値の絶対値に、 $B$  の符号を付加した値を返す。式 (25) と (26) は、 $c_1, s_1, c_2, s_2$  を用いて

$$u \leftarrow c_1 + c_2, \quad (25)$$

$$\hat{R}_{j,j} \leftarrow \underline{\underline{R_{j,j} + \frac{s_2}{u} \times R_{j,k}}}, \quad \hat{R}_{k,k} \leftarrow \underline{\underline{R_{k,k} - \frac{s_1}{u} \times R_{j,k}}}. \quad (26)$$

のように計算される。

## 2.7 両側ヤコビ法へのソーティング機能の付加

$|R_{1,1}^{(0)}| \geq |R_{n,n}^{(0)}|$  の場合,  $|\hat{R}_{j,j}| < |\hat{R}_{k,k}|$  かつ  $s_1 > 0$  ならば,  $\theta_1 \leftarrow \theta_1 - \frac{\pi}{2}$  の意味で  $c_1 \leftarrow s_1, s_1 \leftarrow -c_1, |\hat{R}_{j,j}| < |\hat{R}_{k,k}|$  かつ  $s_1 \leq 0$  ならば,  $\theta_1 \leftarrow \theta_1 + \frac{\pi}{2}$  の意味で  $c_1 \leftarrow -s_1, s_1 \leftarrow c_1$  を行う. さらに,  $|\hat{R}_{j,j}| < |\hat{R}_{k,k}|$  かつ  $s_2 > 0$  ならば,  $\theta_2 \leftarrow \theta_2 - \frac{\pi}{2}$  の意味で  $c_2 \leftarrow s_2, s_2 \leftarrow -c_2, |\hat{R}_{j,j}| < |\hat{R}_{k,k}|$  かつ  $s_2 \leq 0$  ならば,  $\theta_2 \leftarrow \theta_2 + \frac{\pi}{2}$  の意味で  $c_2 \leftarrow -s_2, s_2 \leftarrow c_2$  を行う. これらの選択の結果,  $\theta_1$  と  $\theta_2$  がどちらも  $\frac{\pi}{2}$  を減算もしくは加算している場合,  $\hat{R}_{j,j} \leftarrow \hat{R}_{k,k}, \hat{R}_{k,k} \leftarrow \hat{R}_{j,j}$ , そのほかの場合,  $\hat{R}_{j,j} \leftarrow -\hat{R}_{k,k}, \hat{R}_{k,k} \leftarrow -\hat{R}_{j,j}$  とする.  $|R_{1,1}^{(0)}| < |R_{n,n}^{(0)}|$  の場合,  $|\hat{R}_{j,j}| \geq |\hat{R}_{k,k}|$  かつ  $s_1 > 0$  ならば,  $\theta_1 \leftarrow \theta_1 - \frac{\pi}{2}$  の意味で  $c_1 \leftarrow s_1, s_1 \leftarrow -c_1, |\hat{R}_{j,j}| \geq |\hat{R}_{k,k}|$  かつ  $s_1 \leq 0$  ならば,  $\theta_1 \leftarrow \theta_1 + \frac{\pi}{2}$  の意味で  $c_1 \leftarrow -s_1, s_1 \leftarrow c_1$  を行う. さらに,  $|\hat{R}_{j,j}| \geq |\hat{R}_{k,k}|$  かつ  $s_2 > 0$  ならば,  $\theta_2 \leftarrow \theta_2 - \frac{\pi}{2}$  の意味で  $c_2 \leftarrow s_2, s_2 \leftarrow -c_2, |\hat{R}_{j,j}| \geq |\hat{R}_{k,k}|$  かつ  $s_2 \leq 0$  ならば,  $\theta_2 \leftarrow \theta_2 + \frac{\pi}{2}$  の意味で  $c_2 \leftarrow -s_2, s_2 \leftarrow c_2$  を行う. これらの選択の結果,  $\theta_1$  と  $\theta_2$  がどちらも  $\frac{\pi}{2}$  を減算もしくは加算している場合,  $\hat{R}_{j,j} \leftarrow \hat{R}_{k,k}, \hat{R}_{k,k} \leftarrow \hat{R}_{j,j}$ , そのほかの場合,  $\hat{R}_{j,j} \leftarrow -\hat{R}_{k,k}, \hat{R}_{k,k} \leftarrow -\hat{R}_{j,j}$  とする. さらに, 上記の操作を適用することにより, 両側ヤコビ法では, より大きな特異値から小さな特異値へと並べ替える機能を有する. また, これらの操作によって,  $c_1$  と  $c_2$  は負の値にはならない.

## 3. $c_1, s_1, c_2, s_2$ の修正

Rutishauser の実装 [15] を利用して  $c_1, s_1, c_2, s_2$  を修正する.  $c_1$  と  $c_2$  の値を  $c = \cos(\theta)$  で代表させ,  $s_1$  と  $s_2$  の値を  $s = \sin(\theta)$  で代表させる. 修正後の値を,  $\hat{c}$  と  $\hat{s}$  とする. 理論的には,  $c^2 + s^2 = 1$  が満たされるが, 数値計算においては, 丸め誤差の影響により, 満たされない場合もある. そこで,  $c$  と  $s$  を割線法により修正する. すると,  $-\frac{\pi}{4} \leq \theta \leq \frac{\pi}{4}$  の場合, すなわち,  $|c| \geq s$  の場合には,  $x_0 = 1, x_1 = c$  とし,  $f(x) = x^2 + s^2 - 1$  に対する割線法,

$$x_{n+1} = \frac{x_{n-1}f(x_n) - x_n f(x_{n-1})}{f(x_n) - f(x_{n-1})}, \quad (27)$$

により,  $x_2 = \hat{c} = 1 - s \times (s/(1+c))$  を得る. すると, ベクトル  $\mathbf{x}$  と  $\mathbf{y}$  に対するギブンス回転  $\mathbf{x} \leftarrow c\mathbf{x} + s\mathbf{y}, \mathbf{y} \leftarrow -s\mathbf{x} + c\mathbf{y}$  は,  $z_1 = s/(1+c)$  とすると,

$$\mathbf{x} \leftarrow s \left( \underline{\underline{-z_1 \mathbf{x} + \mathbf{y}}} \right) + \mathbf{x}, \quad \mathbf{y} \leftarrow -s \left( \underline{\underline{z_1 \mathbf{y} + \mathbf{x}}} \right) + \mathbf{y}, \quad (28)$$

となる. 二重下線部は, 融合積和演算により実装する. 同様の操作を行うための式を,  $-\frac{\pi}{2} \leq \theta < -\frac{\pi}{4}, \frac{\pi}{4} < \theta \leq \frac{\pi}{2}$  の場合にも設計する.

謝辞 本研究は JSPS 科研費 JP17H02858 と JP17H00167 の助成を受けたものです.

## 参考文献

- [1] M. Aoki, M. Takata, K. Kimura, and Y. Nakamura, *Improvement of the Thick-Restart Lanczos Method in Single Precision Floating Point Arithmetic using Givens rotations*, Proc. of PDPTA 2019, to appear.
- [2] S. Araki, H. Tanaka, M. Takata, K. Kimura, and Y. Nakamura, *Fast Computation Method of Column Space by using the DQDS Method and the OQDS Method*, Proc. of PDPTA 2018, pp. 333–339, 2018.
- [3] R. P. Brent, F. T. Luk, and C. van Loan, *Computation of the singular value decomposition using mesh-connected processors*, Journal of VLSI and computer systems, Vol. 1, pp. 242–270, 1985.
- [4] J. Demmel, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, 1997.
- [5] J. Demmel and K. Veselic, *Jacobi's Method is More Accurate than QR*, SIAM J. Matrix Anal. Appl., Vol. 13, No. 4, pp. 1204–1245, 1992.
- [6] Z. Drmac and K. Veselic, *New fast and accurate Jacobi SVD algorithm: I.*, SIAM J. Matrix Anal. Appl., Vol. 29, pp. 1322–1342, 2008.
- [7] Z. Drmac and K. Veselic, *New fast and accurate Jacobi SVD algorithm: II.*, SIAM J. Matrix Anal. Appl., Vol. 29, pp. 1343–1362, 2008.
- [8] K. V. Fernando and B. N. Parlett, *Accurate singular values and differential qd algorithms*, Numer. Math., Vol. 67, pp. 191–229, 1994.
- [9] G. E. Forsythe and P. Henrici, *The cyclic Jacobi method for computing the principal values of a complex matrix*, Transactions of the American Mathematical Society, Vol. 94, pp. 1–23, 1960.
- [10] A. Imakura and T. Sakurai, *Block Krylov-type complex moment-based eigensolvers for solving generalized eigenvalue problems*, Numer. Alg., Vol. 75, pp. 413–433, 2017.
- [11] E. Kogbetliantz, *Solution of linear equations by diagonalization of coefficients matrix*, Quarterly of Applied Mathematics, Vol. 13, pp. 123–132, 1955.
- [12] Linear Algebra PACKage, <<http://www.netlib.org/lapack/>>.
- [13] U. von Matt, *The orthogonal qd-algorithm*, SIAM J. Sci. Comput., Vol. 18, pp. 1163–1186, 1997.
- [14] B. N. Parlett and O. A. Marques, *An Implementation of the dqds Algorithm (Positive Case)*, Lin. Alg. Appl., Vol. 309, No. 1–3, pp. 217–259, 2000.
- [15] H. Rutishauser, *The Jacobi Method for Real Symmetric Matrices*, Numerische Mathematik, Vol. 9, No. 1, pp. 1–10, 1966.
- [16] T. Sakurai and H. Tadano, *CIRR: a Rayleigh-Ritz type method with counter integral for generalized eigenvalue problems*, Hokkaido Math. J., Vol. 36, pp. 745–757, 2007.