Performance of the parallel block Jacobi method with dynamic ordering for the symmetric eigenvalue problem

Shuhei Kudo[1], Kousuke Yasuda[1] and Yusaku Yamamoto[1]

[1] The University of Electro-Communications, 1-5-1 Chofugaoka, Chofu, Tokyo 182-8585, Japan

E-mail *shuhei-kudo@outlook.jp*

Received March 7, 2018, Accepted May 1, 2018

Abstract

We investigate the performance of the parallel block Jacobi method for the symmetric eigenvalue problem with dynamic ordering both theoretically and experimentally. First, we present an improved global convergence theorem of the method that takes into account the effect of annihilating multiple blocks at once. Next, we compare the dynamic ordering with two representative parallel cyclic orderings experimentally and show that the former can speedup the convergence for ill-conditioned matrices considerably with little extra cost.

Keywords symmetric eigenvalue problem, Jacobi method, parallel computing

Research Activity Group Algorithms for Mastrix / Eigenvalue Problems and their Applications

1. Introduction

Let A be an $n \times n$ real symmetric matrix and suppose that we want to compute all the eigenvalues and eigenvectors of A. Such a problem arises in various fields of science and engineering. The standard procedure for this problem consists of three steps, namely, transformation of A to a tridiagonal matrix, solution of the resulting tridiagonal eigenvalue problem and back-transformation to recover the eigenvectors of A. However, in highly parallel environments, the tridiagonalization step can become a severe performance bottleneck due to its small parallel granularity [1]. As an alternative to the tridiagonalization based methods, the block Jacobi method has attracted attention recently. It is a natural extension of the (point) Jacobi method for the symmetric eigenvalue problem to the block case and is based on the idea of making the matrix closer and closer to diagonal by annihilating off-diagonal blocks step by step. Thanks to blocking, the block Jacobi method has larger parallel granularity than tridiagonalization based methods and can outperform the latter in some cases [2].

There are many variants of the block Jacobi methods, which differ by the order of annihilating the off-diagonal blocks. In *cyclic ordering* methods, each off-diagonal block is annihilated once in a period called sweep in a predetermined order. In *dynamic ordering* methods, the off-diagonal block to be annihilated at each step is determined based on the run-time information, such as the Frobenius norms of the blocks. Among the latter class of methods, the *classical ordering*, in which the off-diagonal block with the largest Frobenius norm is annihilated at each step, is expected to achieve faster convergence than other orderings, based on the analogy of the point Jacobi methods. This ordering has been extended by Bečka et al. to *parallel dynamic ordering* [3,4], which chooses multiple off-diagonal blocks so that the sum of their squared Frobenius norms is maximal under the constraint that they can be annihilated simultaneously, and annihilates them in parallel. Although this ordering was originally proposed for the block Jacobi SVD (singular value decomposition) method, it should be promising also for the eigenvalue problem because it can attain both fast reduction of the off-diagonal norm and large-grain parallelism at the same time.

The aim of this paper is twofold. First, we will present an improved global convergence theorem for the block Jacobi method for the eigenvalue problem with parallel dynamic ordering. Such a theorem has been given in [5], but there, the constant appearing in the linear convergence bound is the same as that for the sequential classical ordering. Hence, the bound does not reflect the convergence acceleration effect due to annihilating multiple off-diagonal blocks at once. We will derive a tighter bound by taking into account the effect of multiple-block annihilation. Second, we compare the parallel performance and convergence speed experimentally with those of parallel cyclic orderings. The parallel dynamic ordering requires extra costs to compute the Frobenius norms of off-diagonal blocks and to find the set of off-diagonal blocks to be annihilated simultaneously. These costs need to be sufficiently small for the ordering to be competitive. In addition, it is of interest to see how the convergence speed of the parallel dynamic ordering differs from those of cyclic orderings for different types of matrices. We investigate these aspects experimentally in a shared memory parallel environment.

The rest of this paper is structured as follows. In Section 2, we introduce the parallel block Jacobi method with dynamic ordering and its so-called greedy implementation. In Section 3, an improved global convergence theorem for the method is presented. Experimental results that show the superiority of the parallel dynamic ordering to cyclic orderings are provided in Section 4. Finally, Section 5 gives some conclusion.

2. The parallel block Jacobi method using dynamic ordering

Let p be the number of processors and $q = 2p$. We partition the matrix $A^{(0)} = A$ into a $q \times q$ block structure and denote the (I, J) block by $A_{I,J}$. We also denote the submatrix of A consisting of the $I_1, I_2, \ldots, I_\ell$ block rows and $J_1, J_2, \ldots, J_m$ block columns by $A_{I_1 I_2 \ldots I_\ell, J_1 J_2 \ldots J_m}$. The set of all block rows or columns is denoted by '*'. In the kth step of the classical block Jacobi method, we pick up an off-diagonal block $A^{(k)}_{X,Y}$ $(X \neq Y)$ with the largest Frobenius norm and apply an orthogonal similarity transformation that annihilates $A^{(k)}_{X,Y}$ and $A^{(k)}_{Y,X}$ to $A^{(k)}$. Specifically, let

$$A^{(k)}_{XY,XY} = \begin{pmatrix} A^{(k)}_{X,X} & A^{(k)}_{X,Y} \\ A^{(k)}_{Y,X} & A^{(k)}_{Y,Y} \end{pmatrix}$$

and $P^{(k)}_{XY,XY}$ be the orthogonal eigenvector matrix of $A^{(k)}_{XY,XY}$. Moreover, let $P^{(k)}$ be the $n \times n$ orthogonal matrix obtained by embedding $P^{(k)}_{XY,XY}$ into the $n \times n$ identity matrix, in the same way as $A^{(k)}_{XY,XY}$ is embedded in $A^{(k)}$. Then, $A^{(k+1)} = \left(P^{(k)}\right)^\top A^{(k)} P^{(k)}$ is the desired orthogonal similarity transformation. We also let $V^{(0)} = I_n$ and update $V^{(k)}$ by $V^{(k+1)} = V^{(k)} P^{(k)}$ to compute the eigenvector matrix of A.

Let $A^{(k)}_{X',Y'}$ be another off-diagonal block such that $X' \neq X, Y$ and $Y' \neq X, Y$ and $P'^{(k)}$ be the $n \times n$ orthogonal matrix to annihilate $A^{(k)}_{X',Y'}$ and $A^{(k)}_{Y',X'}$. Since $P^{(k)}$ updates only the Xth and Yth block rows (or columns), while $P'^{(k)}$ updates only the X'th and Y'th block rows (or columns), they commute with each other and the two blocks $A^{(k)}_{X,Y}$ and $A^{(k)}_{X',Y'}$ (along with $A^{(k)}_{Y,X}$ and $A^{(k)}_{Y',X'}$) can be annihilated simultaneously. More generally, if the indices $X_1, Y_1, X_2, Y_2, \ldots, X_p, Y_p$ are all different, that is, if they are some permutation of $1, 2, \ldots, 2p$, the p off-diagonal blocks $A^{(k)}_{X_1,Y_1}, A^{(k)}_{X_2,Y_2}, \ldots, A^{(k)}_{X_p,Y_p}$ can be annihilated in parallel, along with $A^{(k)}_{Y_1,X_1}, A^{(k)}_{Y_2,X_2}, \ldots, A^{(k)}_{Y_p,X_p}$. If $X_1, Y_1, X_2, Y_2, \ldots, X_p, Y_p$ can be chosen to maximize $\sum_{\ell=1}^{p} \|A^{(k)}_{X_\ell,Y_\ell}\|_F^2$, a generalization of the classical block Jacobi method will result. This is the idea of parallel dynamic ordering proposed by Bečka et al [3].

The problem of maximizing $\sum_{\ell=1}^{p} \|A^{(k)}_{X_\ell,Y_\ell}\|_F^2$ under the constraint $\{X_1, Y_1, X_2, Y_2, \ldots, X_p, Y_p\} = \{1, 2, \ldots, 2p\}$ can be formulated as a maximum weight matching problem of a complete graph with $2p$ vertices, if we regard the Ith block row/column as vertex I, the (I, J) off-diagonal block as an edge between vertices I and J, and $w_{I,J} = \|A^{(k)}_{I,J}\|_F^2$ as the weight attached to the edge. We can solve this problem approximately by the greedy algorithm, that is, by first choosing X_1 and Y_1 so that $w_{X_1,Y_1} = \max_{1 \le X < Y \le q} w_{X,Y}$, then choosing X_2 and Y_2 so that $w_{X_2,Y_2} = \max_{1 \le X < Y \le q,\, X_2 \neq X_1, Y_1,\, Y_2 \neq X_1, Y_1} w_{X,Y}$, and so on. The algorithm of parallel block Jacobi method using dynamic ordering is shown as Algorithm 1. Here, MWM($\{w_{I,J}\}$) is a function that finds the maximum weight matching $(X_1, Y_1), (X_2, Y_2), \ldots, (X_p, Y_p)$ by the greedy algorithm given the weights $w_{I,J}$ $(1 \le I < J \le q)$.

Algorithm 1 Parallel block Jacobi method with dynamic ordering

```
$A^{(0)} = A$
$V^{(0)} = I_n$
$k = 0$
while $w_{\max} > \epsilon$ do
    Compute $w_{I,J} = \|A^{(k)}_{I,J}\|_F$ for $1 \le I < J \le q$.
    $w_{\max} = \max_{1 \le I < J \le q} w_{I,J}$.
    $(X_1, Y_1), (X_2, Y_2), \ldots, (X_p, Y_p) = \mathrm{MWM}(\{w_{I,J}\})$.
    for $\ell = 1, 2, \ldots, p$ do                ▷ parallel loop
        Compute the eigenvector matrix $P^{(k)}_{X_\ell Y_\ell, X_\ell Y_\ell}$.
        $\hat{A}^{(k)}_{X_\ell Y_\ell,*} = \left(P^{(k)}_{X_\ell Y_\ell, X_\ell Y_\ell}\right)^\top A^{(k)}_{X_\ell Y_\ell,*}$.
    end for
    for $\ell = 1, 2, \ldots, p$ do                ▷ parallel loop
        $A^{(k+1)}_{*,X_\ell Y_\ell} = \hat{A}^{(k)}_{*,X_\ell Y_\ell} P^{(k)}_{X_\ell Y_\ell, X_\ell Y_\ell}$.
        $V^{(k+1)}_{*,X_\ell Y_\ell} = V^{(k)}_{*,X_\ell Y_\ell} P^{(k)}_{X_\ell Y_\ell, X_\ell Y_\ell}$.
    end for
    $k := k + 1$
end while
```

3. Improved global convergence theorem

In [5], the following global convergence theorem has been proved for the parallel block Jacobi method with dynamic ordering.

Theorem 1 *In the parallel block Jacobi method with dynamic ordering and greedy strategy, the off-diagonal elements of $A^{(k)}$ converges to zero as $k \to \infty$.*

More specifically, the following linear convergence bound has been shown in the proof.

$$\sum_{I \neq J} \|A^{(k+1)}_{I,J}\|_F^2 \le \left(1 - \frac{2}{q(q-1)}\right) \sum_{I \neq J} \|A^{(k)}_{I,J}\|_F^2. \qquad (1)$$

Although (1) is sufficient to guarantee global convergence, the constant $1 - 2/[q(q-1)]$ on the right-hand side is not satisfactory because it is the same as the constant appearing in the linear convergence bound of the sequential classical ordering (see [5, Theorem 3.1]). This means that the bound (1) does not reflect the convergence acceleration effect due to annihilating $2p$ blocks at each step. We will improve this in this section.

To this end, we first note that the $q \times q$ blocks of $A^{(k)}$ can be divided into p^2 mutually disjoint 2×2 blocks, $A^{(k)}_{X_\ell Y_\ell, X_m Y_m}$ $(1 \le \ell, m \le p)$. By the orthogonal transformation at step k, each of these 2×2 blocks are transformed as

$$A^{(k+1)}_{X_\ell Y_\ell, X_m Y_m} = \left(P^{(k)}_{X_\ell Y_\ell, X_\ell Y_\ell}\right)^\top A^{(k)}_{X_\ell Y_\ell, X_m Y_m} P^{(k)}_{X_m Y_m, X_m Y_m}.$$

Since $P^{(k)}_{X_\ell Y_\ell, X_\ell Y_\ell}$ $(1 \le \ell \le p)$ is orthogonal, this transformation does not change the Frobenius norm of $A^{(k)}_{X_\ell Y_\ell, X_m Y_m}$. When $\ell \neq m$, the four blocks contained in $A^{(k)}_{X_\ell Y_\ell, X_m Y_m}$ are all off-diagonal blocks, so the contribution from $A^{(k)}_{X_\ell Y_\ell, X_m Y_m}$ to the change of $\sum_{I\neq J}\|A^{(k)}_{I,J}\|_F^2$ is zero. When $\ell = m$, only two of the four blocks, namely $A^{(k)}_{X_\ell,Y_\ell}$ and $A^{(k)}_{Y_\ell,X_\ell}$, are off-diagonal blocks and they are annihilated. So the contribution to the change of $\sum_{I\neq J}\|A^{(k)}_{I,J}\|_F^2$ is $-\left(\|A^{(k)}_{X_\ell,Y_\ell}\|_F^2+\|A^{(k)}_{Y_\ell,X_\ell}\|_F^2\right) = -2\|A^{(k)}_{X_\ell,Y_\ell}\|_F^2$. By summing up this over ℓ, we have

$$\sum_{I\neq J}\|A^{(k+1)}_{I,J}\|_F^2 = \sum_{I\neq J}\|A^{(k)}_{I,J}\|_F^2 - 2\sum_{\ell=1}^{p}\|A^{(k)}_{X_\ell,Y_\ell}\|_F^2.$$

Eq. (1) is obtained by combining this with the following loose lower bound:

$$\sum_{\ell=1}^{p}\|A^{(k)}_{X_\ell,Y_\ell}\|_F^2 \geq \|A^{(k)}_{X_1,Y_1}\|_F^2 \geq \frac{2}{q(q-1)}\sum_{I<J}\|A^{(k)}_{I,J}\|_F^2, \qquad (2)$$

where we used the fact that $A^{(k)}_{X_1,Y_1}$ is the off-diagonal block with the largest Frobenius norm.

In the following, we will derive a tighter lower bound on the left-hand side of (2). Let G_{2p} be a complete graph with $2p$ vertices and nonnegative edge weights $w_{I,J} = \|A^{(k)}_{I,J}\|_F^2$ and M_{2p} its approximate maximum weight matching obtained by the greedy algorithm. Moreover, for an edge-weighted graph $G = (V,E,w)$, we denote its sum of weights by $S(G) = \sum_{e\in E} w(e)$. Then,

$$\sum_{I<J}\|A^{(k)}_{I,J}\|_F^2 = S(G_{2p}), \quad \sum_{\ell=1}^{p}\|A^{(k)}_{X_\ell,Y_\ell}\|_F^2 = S(M_{2p}).$$

We will prove the following lemma.

Lemma 2 *For any complete graph G_{2m}, $m \geq 1$, with nonnegative weights and its approximate maximum weight matching M_{2m} obtained by the greedy algorithm,*

$$S(M_{2m}) \geq \frac{S(G_{2m})}{4m-3}. \qquad (3)$$

Proof We show (3) by induction.

When $m = 1$, G_2 has only one edge and the greedy algorithm chooses it. So, $S(M_2) = S(G_2)$ and (3) is satisfied.

Next, assume that (3) holds for some $m \geq 1$. Let us consider applying the greedy algorithm to a complete graph $G_{2(m+1)}$. This is equivalent to choosing the edge e with the heaviest weight, removing it from $G_{2(m+1)}$ along with other $4m$ edges incident to the endpoints of e, and applying the greedy algorithm recursively to the remaining complete graph, which we call G_{2m}. Now, write the sum of the weights of the $4m+1$ edges removed from $G_{2(m+1)}$ as $tS(G_{2(m+1)})$, where $0 \leq t \leq 1$. Then, since e is the heaviest edge, $w(e) \geq tS(G_{2(m+1)})/(4m+1)$. On the other hand, from the induction hypothesis, the matching M_{2m} of G_{2m} obtained by the greedy algorithm satisfies

$$S(M_{2m}) \geq \frac{S(G_{2m})}{4m-3} = (1-t)\frac{S(G_{2(m+1)})}{4m-3}.$$

Thus, the matching $M_{2(m+1)}$ of $G_{2(m+1)}$ obtained by the greedy algorithm satisfies

$$S(M_{2(m+1)}) = w(e) + S(M_{2m}) \geq t\,\frac{S(G_{2(m+1)})}{4m+1} + (1-t)\,\frac{S(G_{2(m+1)})}{4m-3}.$$

Minimizing the right-hand side with respect to $t \in [0,1]$ gives

$$S(M_{2(m+1)}) \geq \frac{S(G_{2(m+1)})}{4(m+1)-3}.$$

Hence, the induction is completed and the lemma is proved.

(QED)

This lemma enables us to replace $2/[q(q-1)]$ with $1/(4p-3)$ in the right-hand side of (2). As a result, we have the following improved global convergence theorem.

Theorem 3 *In the parallel block Jacobi method with dynamic ordering and greedy strategy, the off-diagonal elements of $A^{(k)}$ satisfies the following inequality and therefore converges to zero as $k \to \infty$.*

$$\sum_{I\neq J}\|A^{(k+1)}_{I,J}\|_F^2 \leq \left(1-\frac{1}{4p-3}\right)\sum_{I\neq J}\|A^{(k)}_{I,J}\|_F^2. \qquad (4)$$

Note that (4) and (1) are valid even when $p = 1$. In this case, $q = 2$ and $A^{(k)}$ is partitioned into a 2×2 block structure. Thus there are only two off-diagonal blocks and they become zero after one step of annihilation. Consequently, both sides of (4) and (1) vanish and the equations hold.

4. Performance evaluation

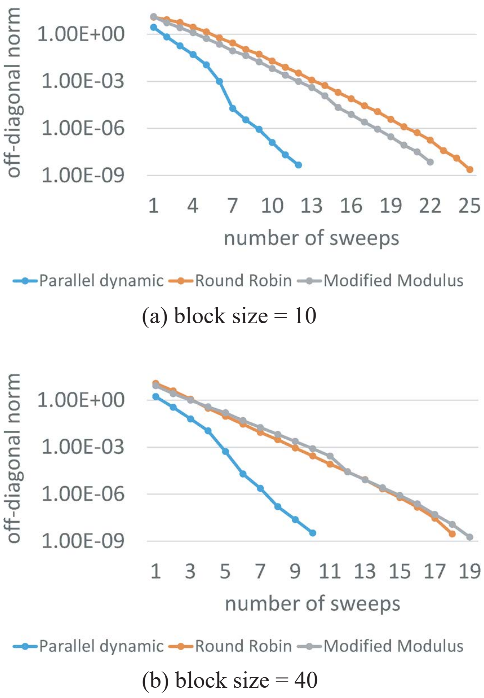
In this section, we evaluate the parallel execution time and convergence properties of the parallel block Jacobi method with dynamic ordering experimentally. To this end, we implemented the block Jacobi method with three different ordering strategies: the parallel dynamic (PD) ordering, round robin (RR) ordering [6] and the modified modulus (MM) ordering [7]. The latter two are representative parallel cyclic orderings. The programs were written in C and parallelized using OpenMP. The stopping criterion was $\max_{1\leq i<j\leq n}|a^{(k)}_{ij}| < 10^{-10}$. To compute the orthogonal eigenvector matrix $P^{(k)}_{X_\ell Y_\ell, X_\ell Y_\ell}$ (see line 9 of Algorithm 1), we used LAPACK routine DSYEVD, which employs the Householder tri-diagonalization and the divide-and-conquer methods. The computational environments are listed in Table 1.

As a test matrix, we used a symmetric matrix A with a prescribed condition number α generated as follows: we first constructed a diagonal matrix D with condition number α by setting its ith diagonal element to $\alpha^{-(i-1)/(n-1)}$ $(1 \leq i \leq n)$ and then computed A as $A = QDQ^\top$, where Q is a random orthogonal matrix. We set $n = 1600$ and varied the block size n/q from 10 to 80. Since there were only 10 cores in the CPU, we allocated multiple values of ℓ to one core in lines 8 and 12 of Algorithm 1 when $n/q = 10$, 20 and 40.

Convergence properties We varied the condition number as $\alpha = 10, 10^5$ and 10^{10}. For $\alpha = 10$, the convergence speed of PD was much the same as that of

Table 1. The computational environment.

Item	Specification
CPU	Intel Xeon E5-2660V2 2.2GHz (10 cores)
Compiler	icc version 16.0.0
Library	MKL version 11.3.0



(a) block size = 10

(b) block size = 40

Fig. 1. Convergence history for the ill-conditioned matrix.

MM, while RR was significantly slower. As the condition number was increased, however, the convergence of MM deteriorated rapidly and a marked difference was seen between the convergence speed of PD and MM. The convergence histories for the case of $\alpha = 10^{10}$ are shown in Fig. 1. Here, we picked up only the results for block size = 10 and 40 due to limitation of space. These results suggest that PD is more robust against ill-conditioned matrices than cyclic orderings. The graphs for the $\alpha = 10$ and $\alpha = 10^5$ cases, as well as the results for random matrices, can be found in [8].

Parallel execution time Compared with cyclic ordering strategies, PD requires two extra steps for each k: computation of the weights $w_{I,J}$ and computation of the approximate maximum weight matching. The former requires $O(n^2/p)$ work per processor, while the latter requires $O(p^3)$ work, if a naive implementation of the greedy algorithm (scanning all the remaining edges to determine each edge in the matching) is employed. Both of these are of smaller order than the work for the update of $A^{(k)}$ and $V^{(k)}$, which are of $O(n^3/p^2)$ per processor. Hence, it is expected that the time for these extra steps are negligible. We confirm this by comparing the execution times for the three ordering strategies.

The total execution times for the case of $n = 1600$ and $\alpha = 10^{10}$ are shown in Fig. 2. As can be seen from the graph, the times for the Frobenius norm computation and maximum weight matching computation account for only small portion of the total execution time, as expected theoretically. Combining this result with the excellent convergence property of PD, we can conclude that the parallel dynamic ordering is a promising ordering strategy for parallelizing the block Jacobi method.

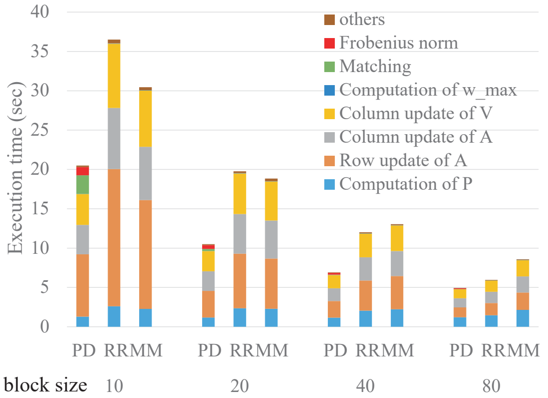


Fig. 2. Execution time for the ill-conditioned matrix.

5. Conclusion

In this paper, we presented an improved global convergence theorem for the parallel block Jacobi method with dynamic ordering. We also confirmed the effectiveness of the method, especially for ill-conditioned matrices, by numerical experiments. Future work includes performance evaluation of the method using a larger number of processors and matrices from real applications.

Acknowledgments

We express our sincere gratitude to the anonymous referee, whose comments helped us to improve the paper. This study is supported by JSPS KAKENHI Nos. 16KT0016, 17H02828, 17K19966 and 17J07747. The present affiliation of Shuhei Kudo is RIKEN Center for Computational Science.

References

[1] T. Fukaya and T. Imamura, Performance evaluation of the Eigen Exa eigensolver on Oakleaf-FX: tridiagonalization versus pentadiagonalization, in: Proc. of IPDPS 2015 Workshops, pp.960–969, IEEE Computer Society, 2015.

[2] S. Kudo, Implementation of the block Jacobi method for the symmetric eigenvalue problem on the K computer and verification of its performance (in Japanese), Master's Thesis, Graduate School of System Informatics, Kobe University, 2015.

[3] M. Bečka, G. Okša and M. Vajteršic, Dynamic ordering for a parallel block-Jacobi SVD algorithm, Parallel Comput., **28** (2002), 243–262.

[4] M. Bečka, G. Okša and M. Vajteršic, New dynamic orderings for the parallel one-sided block-Jacobi SVD algorithm. Parallel Process. Lett., **25** (2015), 1550003.

[5] Y. Yamamoto, Z. Lan and S. Kudo, Convergence analysis of the parallel classical block Jacobi method for the symmetric eigenvalue problem, JSIAM Letters, **6** (2014), 57–60.

[6] R. P. Brent and F. T. Luk, The solution of singular value and symmetric eigenvalue problems on multi-processor arrays, SIAM J. Sci. Stat. Comput., **6** (1985), 69–84.

[7] S. Singer, S. Singer, V. Novaković, A. UšĆumlić and V. Dunjko, Novel modifications of parallel Jacobi algorithms, Numer. Algorithms, **59** (2012), 1–27.

[8] K. Yasuda, Solution of the symmetric eigenvalue problem based on the parallel classical block Jacobi method (in Japanese), Graduation Thesis, The University of Electro-Communications, 2018.