REVISITING THE (BLOCK) JACOBI SUBSPACE ROTATION METHOD FOR THE SYMMETRIC EIGENVALUE PROBLEM

YOUSEF SAAD *

Abstract. The paper revisits the topic of block-Jacobi algorithms for the symmetric eigenvalue problem by proposing a few alternative versions. The main advantage of a block Jacobi method is that it is built entirely from computations with small dense matrices. The proposed mehod is based on a sequence of subspace rotations whose determination requires to solve small Riccati-like correction equation. The paper discusses theoretical and algorithmic aspects of the algorithm, and illustrates its behavior on a few simple examples.

Key words. Symmetric eigenvalue problem, Jacobi algorithm, Riccati equations,

1. Background and introduction.

The Jacobi Algorithm for computing all eigenvalues and vectors of symmetric matrices appeared in 1846 [29] and remained the preeminant method for diagonalizing symmetric matrices until the discovery of the QR algorithm in the early 1960's [17]. In 1960, a remarkable article by Forsythe and Henrici [16] introduced the cyclic version of the algorithm and established a number of convergence results, see also [38] for extensions of this work. After its diminished dominance, the Jacobi method regained some ground when it was advocated as a viable competitor to QR for highly parallel computers in early work on parallel algorithms, see, e.g., [44, 35, 23] among many other works.

However, the adoption of parallel versions of the Jacobi method failed to take hold for various reasons. One of these is that researchers have been able to adapt the QR algorithm to the new environments. For example, a reduction to tridiagonal form via the standard Householder transformations is not as effective for highly parallel platforms, so researchers developed alternatives [37, 4]. Another important reason for the lack of success of parallel Jacobi methods is that these tend to be somewhat complex to implement and that their parallelism is limited: at most $O(n)$ rotations can be applied simultaneously. In addition, the computational cost on a serial computer the Jacobi algorithm is higher than that of the QR algorithm, typically by a factor of about 3 to 5 times [20, 6].

Block Jacobi methods appeared starting in the mid-1980s [33, 3, 14, 30] as a means to improve parallel efficiency of the classic method. In spite of their appeal these methods still currently lag behind theor parallel counterparts of the QR algorithm. Block-Jacobi methods are designed for the new kind of hardware currently available based on Graphics Processing Units. On certain modern platforms, certain computations such as small matrix-matrix multiplications can be orders of magnitude faster when they are performed in reduced precision arithmetic. The most expensive parts of the operations in the block-Jacobi method discussed in this paper, are matrix-matrix products with matrices of selected sizes. In addition, some parts of the calculations can be performed in reduced precision without affecting the rest. A full fledged GPU implementation of the block algorithms discussed here will take some time to develop but this paper focusses on the algorithmic and theoretical aspects of the methods.

It may be useful to summarize the reasons why a Jacobi-type method, whether in standard or block form, can be advantageous, regardless of the computational

*Address: Computer Science & Engineering, University of Minnesota, Twin Cities. e-mail: saad@umn.edu. Work supported by NSF under grant DMS-2011324.

platform being used. The best known advantage of the standard (cyclic) Jacobi iteration comes from its iterative nature: the matrix is transformed directly into diagonal form by an iterative scheme and there is no pre-processing to reduce the matrix into a convenient initial form. Thus, it may require much less work than the alternatives in any situation where the matrix is already in almost in diagonal form. A related appealing situation is when one has a sequence of evolving matrices $A^{(k)}$ where the difference $\|A^{(k+1)} - A^{(k)}\|$ is small and where the changes occur in a small number of locations. The QR algorithm does not seem to be amenable to performing eigendecompostion updates of this type with similar efficiency. Finally, it is known that the Jacobi method tends to be more accurate than the Householder-QR combination for the small eigenvalues in the positive matrix case [20, 10].

A block form adds a number of appealing features to those already known for the scalar version. The most prominant of these is the ability to work with dense blocks of *arbitrary size*. This means that BLAS3-type computations can be optimized and very fast GPUs can be put to work to carry out these calculations. It is also possible to change block-size within the algorithm to exploit the convergence characteristics of the method: at the beginning the inner problems are harder to solve so a small block size may be more effective than a larger. Many other uses are potentially possible but not fully explored here. For example, since the method deals explicitly with rotating subspaces, the block variant presents the possibility of computing or updating a specific subspace and this can have wide range of applications. An illustrative example of this type will be discussed in Section 6.3 of the numerical experiments.

Section 2 lays-out the basic pieces of the proposed block Jacobi algorithm and presents a ‘Riccati correction equation’ that must be solved as part of the algorithm. Section 3, focuses on various schemes for solving this Riccati correction equation. Section 4 is a brief discussion on extending the algorithm to generalized eigenproblems and Section 5 is concerned with theoretical aspects. Finally, Section 6 provides a few illustrative examples and the paper ends with concluding remarks in Section 7.

2. Jacobi Subspace Rotations.

In the following we will make use of the Matlab notation to represent submatrices and arrays. Thus, given a matrix $A \in \mathbb{R}^{N\times N}$, $A(i_1 : i_2, j_1 : j_2)$ represents the submatrix of size $(i_2 - i_1 + 1) \times (j_2 - j_1 + 1)$ and denoted by G that consists of entries a_{ij} where $i_1 \le i \le i_2$ and $j_1 \le j \le j_2$. We are given 2 sets of row indices i_1, i_2, and two sets of column indices j_1, j_2 with

$$1 \le i_1 \le i_2 < j_1 \le j_2 \le N \ . \tag{2.1}$$

This simply corresponds to the selection of an arbitrary sub-block G in the strict upper triangular part of A (diagonal entries are avoided by the constraint $i_2 < j_1$). An illustration is shown on the left side of Figure 2.1. We will use the following dimensions throughout the paper:

$$n_1 = i_2 - i_1 + 1, \qquad n_2 = j_2 - j_1 + 1, \qquad n = n_1 + n_2. \tag{2.2}$$

Thus, the matrices H and M in the figure are both square and of size n_1 and n_2 respectively and G is a matrix of dimension $n_1 \times n_2$. For convenience, we may assume without loss of generality that the row dimension of Y is not less than its column dimension: $n_1 \ge n_2$.

Note that G can be rectangular, i.e., we may have $n_2 \ne n_1$. We will denote by I_k the idenity matrix of size $k \times k$ and when there is no ambiguity the subscript k will be omitted.

2.1. Problem setting. In analogy with the standard Jacobi method, we see that there are 4 blocks involved in the transformaton, namely: $G = A(i_1 : i_2, j_1 : j_2)$, $H = A(i_1 : i_2, i_1 : i_2)$, $M = A(j_1 : j_2, j_1 : j_2)$, and $G^T = A(j_1 : j_2, i_1 : i_2)$. The goal of the block-Jacobi algorithm is to zero out the whole block G (and G^T by symmetry) instead of a single entry as is done in the classical Jacobi algorithm.

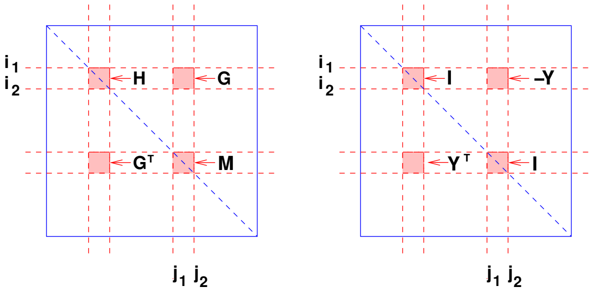

FIG. 2.1. *Illustration of block involved in elimination. Left side: blocks from original matrix A. Right: The block rotation matrix U*

The transformation we use to achieve this goal is formed as follows: Take an $N \times N$ identity matrix and replace the blocks that are in the same positions as H, G, G^T, M by $I_{n_1}, -Y, Y^T, I_{n_2}$ respectively, where Y is a certain $n_1 \times n_2$ matrix to be determined. Note that Y has the same shape as G. This is illustrated on the right side of the figure. We call $\mathcal{U}$ this $N \times N$ matrix.

We can group the four blocks in the shaded parts of the figure into contiguous blocks and define:

$$C = \begin{pmatrix} H & G \\ G^T & M \end{pmatrix}, \qquad U = \begin{pmatrix} I & -Y \\ Y^T & I \end{pmatrix}. \tag{2.3}$$

The matrices C and U are both of size $n \times n$. The matrix U, and therefore also $\mathcal{U}$, is not unitary but we observe that:

$$U^T U = \begin{pmatrix} I + YY^T & 0 \\ 0 & I + Y^T Y \end{pmatrix}. \tag{2.4}$$

Note in passing that the matrices UU^T and $U^T U$ are the same, i.e., U is normal and this is because it is equal to the identity plus a skew-symmetric matrix.

Since the matrix $\mathcal{U}$ is not unitary, $(\mathcal{U}^T \mathcal{U} \neq I)$ a post-scaling will be required to ensure that the combined transformation is a similarity transformation. We assume that the matrix U is multipled to the right by the block-diagonal matrix:

$$\begin{pmatrix} R_1^{-1} & 0 \\ 0 & R_2^{-1} \end{pmatrix} \qquad \text{with} \quad R_1^T R_1 = I + YY^T; \quad R_2^T R_2 = I + Y^T Y. \tag{2.5}$$

Incorporating this scaling will make $\mathcal{U}$ unitary. The matrices R_1, R_2 can be obtained from Cholesky factorizations.

The transformation $\mathcal{U}^T A \mathcal{U}$ will result in the following transformation of C:

$$U^T C U = \left[\begin{array}{c|c} H + YG^T + GY^T + YMY^T & G - HY + YM - YG^T Y \\ \hline G^T - Y^T H + MY^T - Y^T GY^T & M - Y^T G - G^T Y + Y^T HY \end{array} \right] \tag{2.6}$$

The goal is to make the (1,2) block equal to zero.

To find the block Y that eliminates the (1,2) block of (2.6) we need to solve the Riccati-type equation:

$$G - HY + YM - YG^T Y = 0. \tag{2.7}$$

This is a quadratic equation which is a generalized form of the Algebraic Riccati Equation [20] in that the matrices H and M are not necessarily of the same dimension, and so Y may be rectangular. This equation arises in a large number of applications, in addition to the classical context of optimal control [31, 5]. For example, the above equation, termed the 'Riccati correction equation' plays a crucial role when defining - subspace iteration methods and the correction equation in the Jacobi-Davidson method [1, 7, 41]. This will be discussed further in Section 3.4.

We can solve this equation in a number of ways. To simplify notation, we will often call $\tilde{A}$ the matrix obtained from A after a rotation is applied.

2.2. Comparison with the standard block-Jacobi approach. The standard block Jacobi method is known since the 1980s, e.g., [33, 3] and it is described in the classical Golub and Van Loan textbook [20]. Instead of a matrix U in the form of (2.3), the standard method only requires that U be a unitary matrix that diagonalizes the matrix C in (2.3). In other words, U is such that $U^T U = I$ and the matrix $U^T CU$ in (2.6) is diagonal. From one perspective a clear advantage of the classical approach is its simplicity. However, there are a few advantages in the approach proposed in this paper. The main ones are the following two:

1. There is a small advantage in cost. Specifically, due to the special form of U, the new scheme cuts the number of operation by 25%. See section (5.1) for details.
2. The approach enables the use of more progressive schemes in the sense that if G has already been made small from the application of previous steps, then it less inexpensive to find the needed transformation U to annihilate it.

We need to further clarify the second point. Although the matrix C is typically small, diagonalizing it by, e.g., the standard Householder tridiagonalization followed by the QR algorithm can cause a bottleneck in a highly parallel environment. A progressive scheme which takes advantage of a small G can be advantageous, especially in the case when the blocks sizes are not that small.

3. The general Riccati correction equation. We now consider Equation (2.7) in its general form. Although this equation has been encountered and studied in numerous papers in the past, it is important to point out that our context is somewhat different. Guo and Higham [24] study the Nonsymmetric Algebraic Riccati Equation in detail and prove a number of convergence results of Newton's method. Their theory is geared toward applications of Markov models where the matrix H in our notation is an M-Matrix. In fact the solution Y to the problem is nonnegative. The equation under consideration in this paper has some similarities to that of [24] in that Y is rectangular but our matrix C is symmetric. However, the biggest difference is that the matrices H and M have no specific properties apart from symmetry. In the usual setting of Grassmanian subspace iteration [7, 1, 41] and the correction equation in general, one of the 2 dimensions, say n_2 is very small relative to the other dimension, namely n_1.

In the case of the standard Continuous-time Algebraic Riccati Equation (CARE) the two matrix dimensions n_1 and n_2 are the same and it is assumed that $G = G^T$ is semi-positive definite while $M = -H^T$ in our notation, which is rather special. Also, in our case, the user has the freedom to select the block sizes n_1, n_2 and these will often be the same, except that 'boundary cases' arise when N is not divisible by the common block size.

In spite of these differences, many of the approaches that we will take are rather similar. The Riccati equation (2.7) can be converted into an invariant subspace problem, an approach that was proposed first by Alan Laub who devised an elegant method based on the Schur decomposition to solve the CARE problem [31].

3.1. Existence and characterization of a solution. We now return to equation (2.7). We define

$$S = M - G^T Y, \tag{3.1}$$

and rewrite Equation (2.7) as follows:

$$-HY + YS + G = 0. \tag{3.2}$$

Then, putting (3.1) and (3.2) together in matrix form leads to:

$$\begin{bmatrix} H & G \\ G^T & M \end{bmatrix} \begin{bmatrix} -Y \\ I \end{bmatrix} = \begin{bmatrix} -Y \\ I \end{bmatrix} S. \tag{3.3}$$

It is well-known that the *standard* Riccati equation can be formulated as an eigenvalue problem, or rather an invariant subspace problem, see, e.g., [32] and others. We have a similar situation here, but note that the basis of the invariant subspace is required to have a specific structure.

The form (3.3) of the correction Riccati equation will help us analyze Equation (2.7). We denote by Z, E and P the following matrices:

$$Z = \begin{bmatrix} -Y \\ I \end{bmatrix}, \quad E = \begin{bmatrix} O \\ I \end{bmatrix}, \quad P = ZE^T, \tag{3.4}$$

where E and Z are both of size $n \times n_2$, I is the identity matrix of size $n_2 \times n_2$, and P is therefore of size $n \times n$.

It is easy to verify that $P^2 = P$, and that $E^T(x - Px) = 0$. Therefore, P is a projector onto $\text{Span}(Z)$ and orthogonally to the range of E, which is the subspace of $\mathbb{R}^n$ spanned by the last n_2 columns of the canonical basis.

Equation (3.3) shows that if Y is solution to the Riccati equation then Z is the basis of an invariant subspace for C. It is easy to see that the reverse is also true: If there exists a matrix Y and a matrix S such that (3.3) holds then Y is solution of the general Riccati equation 2.7. In the end we can prove the following lemma that characterizes the general solution to (2.7).

LEMMA 3.1. *The following 3 conditions are mathematically equivalent:*

(i) The general Riccati equation 2.7 admits a solution Y;

(ii) There is an n_2 dimensional invariant subspace for C that has a basis of the form of Z in (3.4);

(iii) There is an n_2 dimensional invariant subspace for C such that $\det(E^T Z) \neq 0$ *where $Z = [z_1, z_2, \cdots, z_{n_2}]$ is any basis of this subspace.*

Proof. The first part, namely that (i) and (ii) are equivalent, was proved above. We will show that (ii) is true iff (iii) is true.

The necessary condition is trivial: If the condition is satisfied then there is an invariant subspace of the form Z in (3.4) and for this Z, the matrix $E^T Z = I$ has a nonzero determiant.

For the sufficient condition, first note that the condition $\det(E^T Z) \neq 0$ is invariant under a change of basis. Assume that we have an n_2 dimensional invariant subspace for C spanned by some basis Z. The assumption implies that $\det[E^T Z] \neq 0$. The invariance under C implies that there exists an $n_2 \times n_2$ matrix T such that $CZ = ZT$. We write Z in the form:

$$Z = \begin{bmatrix} -Y_1 \\ Y_2 \end{bmatrix}.$$

From the assumptions, $E^T Z = Y_2$ is noningular - therefore we can change bases by multiplying Z on the right by Y_2^{-1} and obtain the new basis $\hat{Z}$ where

$$\hat{Z} = \begin{bmatrix} -Y_1 Y_2^{-1} \\ I \end{bmatrix} \equiv \begin{bmatrix} -Y \\ I \end{bmatrix}, \tag{3.5}$$

for the invariant subspace. We clearly have: $C\hat{Z} = \hat{Z}S$ where $S = Y_2 T Y_2^{-1}$. This completes the proof. □

The lemma will help establish the existence of a solution for the Riccati equation (2.7) under any condition.

THEOREM 3.2. *Equation (3.3), and therefore also equation (2.7), always has a solution, i.e., there exist an $n_2 \times n_2$ matrix S and an $n_1 \times n_2$ matrix Y, such that (3.3) holds.*

Proof. The proof is constructive. Let the eigen-decompositon of C be

$$CQ = Q\Lambda \tag{3.6}$$

and consider the $n_2 \times n$ matrix Q_1 that consists of the bottom n_2 rows of Q. Since Q is unitary, these rows are linearly independent and therefore there must exist a subset of n_2 of the columns of Q_1 that are linearly independent. Let $p = [i_1, i_2, \cdots, i_{n_2}]$ be this set and let $Q_p = Q(:,p)$ i.e., a matrix formed of columns $i_1, \cdots, i_{n_2}$ of Q. We have $CQ_p = Q_p \Lambda_p$ where Λ_p is an $n_2 \times n_2$ diagonal matrix. In addition $E^T Q_2$ consists of the bottom n_2 rows of Q_2 which are linearly independent so $\det(E^T Q_2) \neq 0$. We are now in the situation of case (iii) of the previous lemma which can be invoked to complete the proof. □

Note that the theorem does not state a result on uniqueness because there is no uniqueness, not even up to basis transformation. Referring to the proof, we will generally have $\binom{n}{n_2}$ possible choices for a subset of columns (the order within each subset is not relevant) that satisfy the condition required by the proof. We say 'generally' because a few of these sets may lead to linear dependent columns. As a result, by 'solving' the equation, we only mean finding one solution among many. In fact the proof of the theorem tells us how we can build all the solutions once we have the eigenvalue decomposition. It is possible to ask the question: among all these solutions which one is the best, in the sense of leading to a numerically more stable computation. This issue will be revisited in Section 3.4.

The next three subsections will address numerical methods for solving the Riccati equation. We begin with Newton's method.

3.2. Solving the Riccati correction equation: The Newton approach. A classic approach to solving Equation (2.7) is to exploit Newton's iteration, see, e.g., [24, 7, 2, 25]. We define the residual matrix:

$$R(Y) \equiv G - HY + YM - YG^T Y, \tag{3.7}$$

Newton's method corresponds to setting the linear part around the current approximation to Y to zero at each step. If Y_k is the approximate solution at step k, then we seek a next iterate in the form $Y = Y_k + \Delta$. For a new approximation of this type the residual is:

$$\begin{aligned} R(Y_k + \Delta) &= G - H(Y_k + \Delta) + (Y_k + \Delta)M - (Y_k + \Delta)G^T(Y_k + \Delta) \\ &= R_k - (H + Y_k G^T)\Delta + \Delta(M - G^T Y_k) - \Delta G^T \Delta, \end{aligned} \tag{3.8}$$

where

$$R_k \equiv G - HY_k + Y_k M - Y_k G^T Y_k. \tag{3.9}$$

The matrix Δ is obtained by solving solving the Sylvester equation:

$$(H + Y_k G^T)\Delta - \Delta(M - G^T Y_k) = R_k. \tag{3.10}$$

Once Δ is available, the solution is updated as:

$$Y_{k+1} = Y_k + \Delta. \tag{3.11}$$

Equations (3.11)–(3.10)–(3.9) could also have been arrived at by simply applying Newton's method to equation (2.7), and noting that the differential of the linear mapping $R(Y)$ with respect to Y and applied to Δ is:

$$\left\langle \frac{dR}{dY}, \Delta \right\rangle = -[(H + YG^T)\Delta - \Delta(M - G^T Y)]. \tag{3.12}$$

Once Y_{k+1} is updated as in (3.11), then, according to (3.8) we have

$$R_{k+1} = -\Delta G^T \Delta. \tag{3.13}$$

In summary, the main steps for solving (2.7) are as shown in Algorithm 1.

Algorithm 1 Newton method for solving (2.7)

```
Start: Select $Y_0$ and compute $R_0$ from (3.9)
while $\|R_k\| > tol$ do
    Find $\Delta$ solution of (3.10)
    Compute $Y_{k+1} := Y_k + \Delta$
    Compute $R_{k+1} := -\Delta G^T \Delta$
end while
```

An equivalent formulation of the algorithm is to define Y_{k+1} directly from Y_k. It is easy to see from (3.10) and (3.9) that the matrix Y_{k+1} produced by the above algorithm satisfies the Sylvester equation:

$$(H + Y_k G^T)Y_{k+1} - Y_{k+1}(M - G^T Y_k) = G + Y_k G^T Y_k. \tag{3.14}$$

Clearly, relation (3.13) shows that convergence is quadratic at the limit as is expected. In our experiments we observed that for small block sizes, algorithm 1 often converges with the intitial guess $\Delta = 0$. Note that unconditional convergence was established in a different context where the matrix C is an M-matrix [24]. In spite of its appealing quadratic convergence, Newton's method has the disadvantage that it requires the solution of a Sylvester equation at each step. Its convergence can also be erratic sometimes. The method can be useful in some special situations, see, for example the experiments in Section 6.3. Next we describe a few alternatives based on invariant subspaces.

3.3. Manifold subspace iteration. As we now show, a method proposed by Chatelin in 1984 [9] is strongly related to the Newton approach seen above. We first give a brief description of Chatelin's method in terms of our notation. Given a matrix C, the problem is to find an invariant subspace as defined by a certain basis Z, i.e., to find $Z \in \mathbb{C}^{n\times m}$ such that

$$CZ = ZT \tag{3.15}$$

where T is a certain matrix in $\mathbb{R}^{m\times m}$. We select a matrix $E \in \mathbb{R}^{n\times m}$ and impose the condition: $E^T Z = I$. Upon multiplying (3.15) by E^T on the left we find that $T = E^T CZ$. Therefore, we need to find Z such that

$$\begin{cases} CZ & = & Z(E^T CZ) \\ E^T Z & = & I \end{cases}$$

The above pair of equations is a nonlinear system and Chatelin proposed to solve it with a Newton approach. This method belongs to the class of *Grassmann manifold* subspace iteration methods, see e.g., [11, 1, 45, 15, 8] among others. In these methods the Newton iteration acts on a subspace. A constraint, such as $E^T Z$ in our case, is imposed to extract one particular basis of the subspace among infinitely many others. Chatelin's method seems to be among the first in this class and has been mentioned in a few articles that deal with Grassmanian subspace iteration [11, 1, 45, 15].

We now explore the method in more detail. Let:

$$F(Z) = CZ - Z(E^T CZ). \tag{3.16}$$

and note that the Frechet differential of F at Z can be defined through the mapping:

$$F'(Z).\Theta = (I - ZE^T)C\Theta - \Theta(E^T CZ). \tag{3.17}$$

Suppose we want to perform one step of Newton's method. If Z is the current iterate and if the next iteration by Newton's method is $\widetilde{Z} = Z + \Theta$, then, it is clear from (3.16) and (3.17) that Θ must be solution of the Sylvester equation

$$(I - ZE^T)C\Theta - \Theta(E^T CZ) = -R \tag{3.18}$$

where $R = F(Z) \equiv CZ - Z(E^T CY)$. We now write Z in the form (3.4) and select E in the form defined in (3.4). In this situation:

$$\Theta = \begin{pmatrix} -\Delta \\ 0 \end{pmatrix}; \quad (I - ZE^T)C\Theta = \begin{pmatrix} -H\Delta - YG^T\Delta \\ 0 \end{pmatrix}.$$

Now since $E^T CZ = M - G^T Y (= S)$ we see that (3.18) yields a system of the form:

$$\begin{aligned} -H\Delta - YG^T\Delta + \Delta(M - G^T Y) &= -R \\ 0 &= 0. \end{aligned} \tag{3.19}$$

The second equation is vacous but the first one yields:

$$(H + YG^T)\Delta - \Delta(M - G^T Y) = R,$$

where R is the same as R_k in (3.10) without the indices. Therefore (3.19) and (3.10) are identical.

Proposition 3.3. *Newton's method for solving the Riccati equation* (2.7)*, as described by Algorithm 1, is mathematically equivalent to Chatelin's Simultaneous Newton iteration method.*

In the numerical experiments, we will utilize a variant of this method which is simply the standard manifold approach where the basis E of the left subspace in equation (3.18) is replaced by Z and Z is made orthogonal at each step:

$$(I - ZZ^T)C\Theta - \Theta(Z^TCZ) = -R \qquad \text{(Solve for } \quad \Theta) \tag{3.20}$$

$$Z := \texttt{qrf}(Z + \Theta) \qquad \text{(Update+Orthogonalize)} \tag{3.21}$$

where `qrf(`X`)` denotes the Q factor in the QR factorization of X.

3.4. The invariant subspace approach. Next we exploit the eigenvalue decomposition to solve (2.7). The approach described is closer to the traditional block form of the Jacobi method as descibed in, e.g., [33, 3, 14, 28]. The approach discussed here is based on the proof of Theorem 3.2 and it assumes that the eigendecomposition (3.6) of C is available. Following the construction of the proof we can see that all that is needed is a subset of n_2 columns of the matrix Q_1 that are linearly independent. Given the large number of possible choices (there are generally $\binom{n}{n_2}$ possible selections) we are led to ask the question which one is best. From the point of view of the block subspace iteration, any solution will do. From a computational point of view, we need to worry about accuracy. In computing the solution Y from (3.5), a matrix inversion is needed. Thus, we wish to select the columns of Q_1 in such a way that the matrix Y_2 is as well-conditioned as possible.

The problem of selecting the best k columns (or rows) of a given matrix has been investigated by many authors from different perspectives. Goreinov et. al [21] seem to be the first to have addressed the question from a theoretical angle, see also [22]. They prove that it is possible to select r columns and r rows to form what they call the pseudoskeleton submatrix that approximates some original $m \times n$ matrix with an order $O(\epsilon\sqrt{r}(\sqrt{m} + \sqrt{n}))$. They introduce the notion of volume which is simply the absolute value of the determinant, to measure the quality of the selected basis. Later a number of articles explored a related question but with a different goal, geared toward data-related applications. Specifically, the problem is to sample rows / columns with certain probabilities, see for example [51, 12, 13, 36]. Here again the same notion of volume plays a crucial role.

From a practical point of view, it is sufficient for our purpose to simply use a QR-based method analyzed in [51] and named the greedy algorithm. We will simplify notation by considering a matrix X of size $m \times n$ where $m < n$ and such that $XX^T = I$. The greedy QR-based approach is shown in Algorithm 2.

Algorithm 2 Greedy column selection

```
for $j = 1, \cdots, n_2$ do
    Find index $p_j$ of column of $X$ of largest norm. Let $v = x_{p_j}$ and $\rho_j = \|v\|$.
    Project out $v$ from $X$, i.e., $X := X - PX$, with $P = vv^T/\rho_j^2$
end for
```

Apart from its lack of normalization Algorithm 2 is nothing but the 'forward-looking' implementation of the modified Gram-Schmidt algorithm [20]. If $X_p = X(:,p)$ is the $n_2 \times n_2$ selected matrix then it is easy to quantify its volume in terms of the norms ρ_j produced in Algorithm 2.

Proposition 3.4. *Let p be the permutation selected by Algorithm 2 and $X_p = X(:,p)$ the matrix with columns $p_1, p_2, \cdots, p_{n_2}$ of X. Then,*

$$Vol(X_p) = \prod_{j=1}^{n_2} \rho_j. \tag{3.22}$$

Proof. This comes from the QR factorization $X_p = QR$ of X_p. It can be easily seen that the diagonal entries of R are just the scalars ρ_j generated by Algorithm 2. Then, $|\det X_p| = |\det Q \det R| = \rho_1 \cdots \rho_{n_2}$. □

Thus, Algorithm 2 can indeed be viewed as just a greedy approach to minimizing volume. Bounds have been established in [51] to show how far from the optimum (i.e., largest possible) volume one can get by using this approach. Because X has orthonormal rows, the situation we have is more favorable than what might be inferred from these bounds which address worst case scenarios. Indeed, a sort of universal bound was shown in the article [21] for the exact situation we have, namely for when $XX^T = I$. We restate the result (Lemma 2.1 of [21]).

Lemma 3.5 ([21]). *Let X be* any *$r \times n$ matrix ($r < n$) such that $XX^T = I$ and let X_p be the matrix of largest volume extracted from X. Then the smallest singular value σ_r of X_p satisfies:*

$$\sigma_r \geq \frac{1}{\sqrt{1 + r(n-r)}}. \tag{3.23}$$

What is remarkable about this result is that it does not depend on the given matrix but only on the dimensions.

The final algorithm for computing the solution Y to the Riccati equation from the eigendecomposition, can now be formulated, see Algorithm 3.

Algorithm 3 Invariant subspace approach for solving (2.7)

```
1: Compute the eigen-decomposition (3.6) and extract $X = Q(n_1 + 1 : n, :)$
2: Call Algorithm 2 to find permutation $p = [p_1, p_2, \cdots, p_{n_2}]$
3: Compute $Y$ from (3.5) where $Z = [Q(:,p_1), Q(:,p_1), \cdots, Q(:,p_{n_2})]$
```

4. In brief: Extension to the generalized eigenvalue problem. Similarly to the common block version of the Jacobi iteration [27, 26], it is also possible to extend the subspace rotation algorithm described in this paper to generalized eigenvalue problems of the form $Au = \lambda Bu$ where we only make an assumption of symmetry for both A and B. The main difference with the subspace rotation matrices $\mathcal{U}$ presented in Section 2 is that we now need to zero-out two blocks at the same time one for A and one for B. Referring to Figure 2.1, we now have two sets of blocks: H_a, G_a, M_a for A and H_b, G_b, M_b for B and the goal is annihilate *both* G_a and G_b. One single matrix Y has $n_1 n_2$ unknowns which will not allow to satisfy the $2n_1 n_2$ constraints needed to zero out the two blocks. For this we need to generalize U of (2.3) as follows:

$$U = \begin{pmatrix} I & -Y_1 \\ Y_2^T & I \end{pmatrix}. \tag{4.1}$$

This leads to a matching number of unknowns and equations. We now have two blocks in lieu of the (1,2) block of (2.6), one for A and one for B, both of which need to be set to zero:

$$G_a - H_a Y_1 + Y_2 M_a - Y_2 G_a^T Y_1 = 0 \tag{4.2}$$
$$G_b - H_b Y_1 + Y_2 M_b - Y_2 G_b^T Y_1 = 0. \tag{4.3}$$

The above equations represent a 'coupled system' of Algebraic Riccati equations and is encountered in some applications [19, 18]. It is of the same family of Riccati equations. Specifically, we have a mapping from $\mathbb{R}^{2n_1 n_2}$ to $\mathbb{R}^{2n_1 n_2}$ that includes a linear term and a quadratic one and the goal is to find the zero of the mapping. We implemented and tested a generalization of the invariant subspace approach described earlier, with the *MaxVol* technique. Details are omitted.

5. Analysis. This section addresses a few theoretical issues for the Jacobi subspace rotation algorithm. In a first subsection we will consider the computational cost of the algorithm and then we will analyse its convergence.

5.1. Computational cost. We consider an algorithm that annihilates square blocks of size $\nu \times \nu$ where ν is a parameter and the dimension N of the matrix is a multilple of ν. Thus, with this we have $n_1 = n_2 = \nu$. We can view the resulting matrix as a block matrix of size $(N/\nu) \times (N/\nu)$ with each entry being a $\nu \times \nu$ submatrix. We define $m \equiv N/\nu$. We would like to analyze the cost of a "complete sweep" in which each of the $m(m-1)/2$ blocks is annihilated exactly once. The goal is to compare the cost of the resulting block algorithm with that of a standard Jacobi algorithm. In counting arithmetic operations, we will only count multiplications. In what follows we discuss the costs of each of the main components of the algorithm.

Determining the matrix Y. This is needed for each transformation and it costs $O(\nu^3)$. We need not be more specific as this cost depends on which of the algorithms is used. There is an additional cost to determine the Cholesky factor of the matrix $\widetilde{U}^T U$ – also an ν^3 cost. We will group these operations as one and write the cost in the form $\beta \nu^3$.

Applying the subspace rotation. The first set of transformations, is applied to the right (columns). One of these is of the form

$$A(:, j1 : j2) := A(:, j1 : j2) - A(:, i1 : i2) * Y \tag{5.1}$$

which costs $\nu^3 \times (N/\nu) = \nu^2 N$ multiplies. The other is similar and involves the group of columns $i1 : i2$. Then, two similar operations are performed on the left (rows) and involves rows $i1 : i2$ and then rows $j1 : j2$. The total for these 4 operations whould therefore be $4 \times (\nu^2 N)$ multilications. However, we need to keep in mind that work is performed only on the upper part of the matrix so this must be halved to $2 \times (\nu^2 N)$ multilications.

Post-scaling the rotated blocks. We need to apply 4 scaling operations by the block-diagonal matrices of size $\nu \times \nu$. The cost depends on which form of scaling is applied. If we use Cholesky factors, then the matrices R_1, R_2 are upper triangular and so the cost of each scaling operation is $\approx \frac{1}{2}\nu^2 N$. There are four such operations, two from the left and two from the right and so the total for scaling is $2\nu^2 N$ - but again we only work on the upper half of the matrix so this becomes simply $\nu^2 N$.

Totals. The total for each transformation comes to $\beta\nu^3 + 3\nu^2 N$ and for a complete sweep of $m(m-1)/2 \approx \frac{1}{2}(N/\nu)^2$ we get the total of

$$T_A(N, \nu) \approx \frac{N^2}{2\nu^2}\left[\beta\nu^3 \ + \ 3\nu^2 N\right] = \frac{3}{2}N^3 + \frac{\beta}{2}\nu N^2 \ . \tag{5.2}$$

For the standard block-Jacobi method the transformation (5.1) becomes something like $B(:, j1 : j2) := B(:, j1 : j2) * V_1 + B(:, i1 : i2) * V_2$ where V_1, V_2 are $\nu \times \nu$ and so the cost of which is four times $2\nu^2 N$ for each transformation. The total over all blocks of a sweep becomes $2N^3 + \frac{\beta}{2}\nu N^2$ (after we taking into account a division by 2 due to symmetry).

This is to compare with (5.2) which indicates that the overall cost of the proposed scheme involves 25% fewer operations provided $\nu\beta$ remains small relative to N and the Cholesky form of scaling is used in the transformations.

Accumulation of block rotations. To the above cost we need to add the separate cost of accumulating the orthogonal transformations if, as is often the case, the eigenvectors are wanted. Here each transformation is applied on the right only and it includes two transformations similar to (5.1) each at the cost of $\nu^2 N$ multiplies each to which we add cost of 2 scalings at the cost of $\frac{1}{2}\nu^2 N$ each. Each transform will cost $3\nu^2 N$ and the total is

$$T_Q(N, \nu) \approx \frac{N^2}{2\nu^2} \times \nu^2 N = \frac{3}{2} N^3. \tag{5.3}$$

The traditional block algorithm requires a transformation on two block columns which, when taken together, amounts to multiplying a matrix of size $N \times (2\nu)$ by a square matrix of size $(2\nu) \times (2\nu)$ at the cost of $4\nu^2 N$. For a sweep of $m(m-1)/2$ rotations, this leads to a total of $\approx \frac{1}{2}m^2 \times 4\nu^2 N = 2N^3$ operations. Therefore, this part of the proposed algorithm also reduces the number of operations by about 25% relative to that of the classical block Jacobi method.

5.2. Convergence Analysis. A convergence analysis of the Block -Jacobi algorithm can be derived in a straightforward manner by adapting results that have been established in the classical scalar and block case. A number of results already exist for the block case, see, e.g., [14, 49, 30, 50] and interested readers are referred to these.

Here we will only briefly discuss one tool that can be invoked in an effort to relate the analysis to known techniques, namely the following definition of the matrix-of-norms:

$$\Omega(A) = \{\omega_{ij}\}_{i,j=1:m} \quad \text{with} \quad \omega_{ij} = \|A_{ij}\|_F. \tag{5.4}$$

This is an $m \times m$ matrix and we observe that we have $\|A\|_F = \|\Omega(A)\|_F$. Using this definition reduces the analysis in effect to a sort of classical Jacobi method applied to an $m \times m$ matrix.

For example let us assume that the classical Jacobi method is applied, where the block that has the biggest F-norm is annihilated at each time. If D_X is the block-diagonal matrix whose block-diagonal submatrices are the same as those of X and if $\tilde{A}$ is the matrix obtained from A after such a block rotation is applied, then the analysis of the scalar case [20] can be easily adapted to show the inequality:

$$\|\Omega(\tilde{A} - D_{\tilde{A}})\|_F^2 \leq \left[1 - \frac{2}{m(m-1)}\right] \|\Omega(A - D_A)\|_F^2, \tag{5.5}$$

which establishes global convergence in this situation. Here, we wish only to point out that the similar notion of vector-of-norms was introduced by François Robert [42] as a means of studying the convergence of vector sequences that arise from iterative methods. [1]

[1] This work won the very first PhD thesis prize at the Householder meeting (then called the

5.3. The problem of changing 'affiliation'. Forsythe and Henrici [16] proved that under some mild restrictions, the sequence of matrices obtained from the cyclic Jacobi iteration does indeed converge to a diagonal matrix. This required showing that a diagonal entry does not change affiliation during the algorithm in the sense that when $a_{ii}^{(k)}$ is closest to an eigenvalue λ at step k, then $a_{ii}^{(k+1)}$ cannot become closest to another sufficiently distinct eigenvalue μ. This was also discussed in Wilkinson [48, p. 268]. It is possible to show a similar, though not as complete, result for the block case.

In what follows eigenvalues of any given block A_{ii} are labeled decreasingly and we denote by $\lambda_l(A_{ii})$ the l-th eigenvalue of A_{ii}. We will use the notation:

$$dist(\lambda, A_{ii}) = \min_{\nu} |\lambda_\nu(A_{ii}) - \lambda|. \tag{5.6}$$

Here, we reverse the definition of affiliation by saying that an eigenvalue λ of A is affiliated with block (i,i) if

$$dist(\lambda, A_{ii}) \leq dist(\lambda, A_{jj}), \ \forall j. \tag{5.7}$$

In addition, λ is 'strictly affiliated' with block (i,i) if there exists a positive scalar δ (that depends on λ) such that the following holds:

$$dist(\lambda, A_{ii}) \leq dist(\lambda, A_{jj}) - \delta \quad \text{for } j \neq i. \tag{5.8}$$

Let us assume that at some step, λ is strictly affiliated with block (p,p). If we apply a rotation (p,q) at this step then eigenvalues in all blocks remain the same except those in blocks p and q. The question is: Can an eigenvalue's affiliation 'jump' from block p to block q (or vice versa)? We show two lemmas that will help answer the question.

Lemma 5.1. *Assume that we apply a rotation associated with block (p,q) and that the matrix Y utilized during this rotation satisfies $\|Y\|_2 \leq \epsilon$ and also that $\|G\|_2 \leq \epsilon$. Then after the rotation is performed we have*

$$|\lambda_l(\tilde{A}_{pp}) - \lambda_l(A_{pp})| \leq \epsilon^2 \left[2 + \|A_{pp}\| + \|A_{qq}\|\right] \tag{5.9}$$

$$|\lambda_l(\tilde{A}_{qq}) - \lambda_l(A_{qq})| \leq \epsilon^2 \left[2 + \|A_{pp}\| + \|A_{qq}\|\right]. \tag{5.10}$$

Proof. According to (2.6) $\tilde{A}_{pp}$ will be as follows:

$$\tilde{A}_{pp} = R_1^{-T}[A_{pp} + YA_{pq}^T + A_{pq}Y^T + YA_{qq}Y^T]R_1^{-1}$$

where $R_1^T R_1 = I + YY^T$. Consider, for any nonzero vector x in $\mathbb{R}^{n_1 \times n_1}$, the following Rayleigh quotient where we set $z = R_1^{-1}x$:

$$\begin{aligned}\frac{(\tilde{A}_{pp}x, x)}{(x,x)} &= \frac{(R_1^{-T}[A_{pp} + YA_{pq}^T + A_{pq}Y^T + YA_{qq}Y^T]R_1^{-1}x, x)}{(x,x)} \\ &= \frac{([A_{pp} + YA_{pq}^T + A_{pq}Y^T + YA_{qq}Y^T]z, z)}{(R_1 z, R_1 z)}.\end{aligned}$$

Gatlinburg conference)

Observe that $(R_1 z, R_1 z) = (R_1^T R_1 z, z) = ((I + YY^T)z, z) \equiv (z, z)(1 + \eta^2)$ where we set $\eta \equiv \|Y^T z\|/\|z\|$, and note that $\eta \leq \epsilon$. Therefore:

$$\begin{aligned}
\frac{(\tilde{A}_{pp}x, x)}{(x, x)} &= \frac{([A_{pp} + YA_{pq}^T + A_{pq}Y^T + YA_{qq}Y^T]z, z)}{(z, z)(1 + \eta^2)} \\
&= \frac{(A_{pp}z, z)}{(z, z)} \times \frac{1}{1 + \eta^2} + \frac{([YA_{pq}^T + A_{pq}Y^T + YA_{qq}Y^T]z, z)}{(z, z)(1 + \eta^2)} \\
&= \frac{(A_{pp}z, z)}{(z, z)} - \frac{(A_{pp}z, z)\eta^2}{(z, z)(1 + \eta^2)} + \frac{([YA_{pq}^T + A_{pq}Y^T + YA_{qq}Y^T]z, z)}{(z, z)(1 + \eta^2)},
\end{aligned}$$

where we used the relation $1/(1 + \eta^2) = 1 - \eta^2/(1 + \eta^2)$. This leads to the inequality

$$\begin{aligned}
\left| \frac{(\tilde{A}_{pp}x, x)}{(x, x)} - \frac{(A_{pp}z, z)}{(z, z)} \right| &\leq \left| \frac{(A_{pp}z, z)\eta^2}{(z, z)(1 + \eta^2} \right| + \left| \frac{([YA_{pq}^T + A_{pq}Y^T + YA_{qq}Y^T]z, z)}{(z, z)(1 + \eta^2)} \right| \\
&\leq \|A_{pp}\| \times \epsilon^2 + 2\epsilon^2 + \epsilon^2 \|A_{qq}\|. \qquad (5.11)
\end{aligned}$$

There is a one-to-one association between a given subspace S of dimension $l \leq n_1$, and the l dimensional subspace $R_1^{-1}S$. There is also a one-to-one correspondance from an arbitrary vector $x \in S$ to the vector $z = R_1^{-1}x$ and the Rayleigh quotients $(\tilde{A}_{pp}x, x)/(x, x)$ and $(A_{pp}z, z)/(z, z)$ are related by (5.11). We can now invoke the min-max principle which will show the result. A similar result can be established for the $(2, 2)$ block. □

With this, we can now prove the following result.

Lemma 5.2. *Let λ be strictly affiliated with block (p, p) at some step and assume that a block-rotation (p, q) is applied at this step where it is assumed that the block A_{pq} satisfies $\|A_{ij}\| \leq \epsilon$ and that we also have $\|Y\| \leq \epsilon$. Denote by $\tilde{A}_{pp}, \tilde{A}_{qq}$, the matrices in blocks p and q respectively obtained after the rotation is applied and define $\alpha \equiv 2 + \|A_{pp}\| + \|A_{qq}\|$. Then*

$$dist(\lambda, \tilde{A}_{qq}) \geq dist(\lambda, A_{pp}) \ + \ \delta - \alpha\epsilon^2 \qquad (5.12)$$

$$dist(\lambda, \tilde{A}_{pp}) \leq dist(\lambda, A_{pp}) \ + \ \alpha\epsilon^2 \ . \qquad (5.13)$$

Proof. Define the indices l, k such that:

$$dist(\lambda, \tilde{A}_{pp}) \equiv |\lambda - \lambda_l(\tilde{A}_{pp})|, \quad \text{and} \quad dist(\lambda, \tilde{A}_{qq}) \equiv |\lambda - \lambda_k(\tilde{A}_{qq})|.$$

Inequality (5.12) is based on the second triangle inequality:

$$|\lambda - \lambda_k(\tilde{A}_{qq})| \geq |\lambda - \lambda_k(A_{qq})| - |\lambda_k(A_{qq}) - \lambda_k(\tilde{A}_{qq})|. \qquad (5.14)$$

Our definition (5.8) shows that (recall that λ is strictly associated with (p, p))

$$|\lambda - \lambda_k(A_{qq})| \geq dist(\lambda, A_{qq}) \geq dist(\lambda, A_{pp}) + \delta. \qquad (5.15)$$

In addition Lemma 5.1 shows that

$$|\lambda_k(A_{qq}) - \lambda_k(\tilde{A}_{qq})| \leq \alpha\epsilon^2. \qquad (5.16)$$

Substituting (5.16) and (5.15) into (5.14) yields inequality (5.12).

To show inequality (5.13) we let l_0 be the index ν for which $|\lambda - \lambda_\nu(A_{pp})|$ is minimum. Then we write:

$$\begin{aligned} dist(\lambda, \tilde{A}_{pp}) = |\lambda - \lambda_l(\tilde{A}_{pp})| &\leq |\lambda - \lambda_{l_0}(\tilde{A}_{pp})| \\ &\leq |\lambda - \lambda_{l_0}(A_{pp})| + |\lambda_{l_0}(A_{pp}) - \lambda_{l_0}(\tilde{A}_{pp})| \\ &\leq dist(\lambda, A_{pp}) + \alpha\epsilon^2. \end{aligned}$$

which shows (5.13). This completes the proof. □

A simple consequence of the above lemma is that when ϵ^2 is small enough relative to δ then λ cannot change its affiliation.

Theorem 5.3. *Let the assumptions of Lemma 5.2 be satisfied and assume that in addition:*

$$\delta - 2\alpha\epsilon^2 > 0.$$

Then, after the block rotation (p, q) is applied, the eigenvalue λ can only be affilited with block (p, p) of $\tilde{A}$.

Proof. Inequality (5.13) yields $dist(\lambda, A_{pp}) \geq dist(\lambda, \tilde{A}_{pp}) - \alpha\epsilon^2$. Substituting this into (5.12) we obtain:

$$dist(\lambda, \tilde{A}_{qq}) \geq dist(\lambda, \tilde{A}_{pp}) \ + \ \delta - 2\alpha\epsilon^2.$$

When $\delta - 2\alpha\epsilon^2 > 0$ then

$$dist(\lambda, \tilde{A}_{qq}) > dist(\lambda, \tilde{A}_{pp}) \geq \min_i dist(\lambda, \tilde{A}_{ii}).$$

This means that λ cannot be affiliated with block (q, q). Since all blocks A_{jj} for $j \neq q$, and $j \neq p$ have not changed after the rotation, their eigenvalues are the same as before. Therefore λ is affiliated with either (p, p) or some block (j, j) with $j \neq p, q$.

Let now $j \neq p$ (and $\neq q$) and note that A_{jj} and $\tilde{A}_{jj}$ are identical so in what follows $dist(\lambda, \tilde{A}_{jj}) = dist(\lambda, A_{jj})$. Define the index l_0 such that $\min_\nu |\lambda - \lambda_\nu(A_{pp})| = |\lambda - \lambda_{l_0}(A_{pp})|$. Using (5.8) we obtain

$$dist(\lambda, A_{jj}) \geq dist(\lambda, A_{pp}) + \delta. \tag{5.17}$$

Equation (5.13) along with (5.17) yield the following inequality:

$$dist(\lambda, A_{jj}) \geq dist(\lambda, \tilde{A}_{pp}) + \delta - \alpha\epsilon^2 \geq dist(\lambda, \tilde{A}_{pp}) + (\delta - 2\alpha\epsilon^2) > dist(\lambda, \tilde{A}_{pp}) \tag{5.18}$$

which establishes the desired result that λ cannot be affiliated with block (j, j) when $j \neq p, q$. In the end λ can only be affiliated with block (p, p) of $\tilde{A}$. □

The above result may appear unnecessary at first. Indeed, we know that the off-diagonal blocks will converge to zero so in the end we wind up with a block-diagonal matrix that is similar to the original A and the eigenvalues of A can easily be recovered from all the different blocks. However, there are situations where some guarantee that the eigenvalues do not change affiliation can be useful. In case the matrix has already been diagonalized and we perturb the original matrix we may be interested in following say the largest k eigenvalues of the perturbed matrix. If these belong to a certain block, then we may need to only focus on that block if we know that affiliation has not changed. Without this, we would have to examine all the diagonal blocks. An experiment will provide an illustration in Section 6.3.

5.4. Quadratic convergence. Proofs of quadratic convergence for the classical or cyclic Jacobi iteration are rather involved, see, for example the discussion in [48], and related articles [47, 46, 43, 34]. Rather than trying to prove formal bounds, it may be more instructive for our purpose to explain the mechanism by which quadratic convergence is set into motion. Here we will make similar assumptions as those of the previous section. We will assume that after a number of whole cyclic sweeps, all blocks are small enough as the process is nearing convergence.

Assume that the sweep proceeds row-wise: we eliminate all those blocks in the strict upper triangular matrix by row: $(1,2),(1,3)\cdots(1,m)$ then $(2,3),(2,4)\cdots,$ $(2,m)$, ... When an eliminination is processed, we combine columns (right transformations) or rows (left transformations). The transform to zero-out the block in position (p,q) is as follows:

$$\tilde{A}(i,p) = A(i,p) + A(i,q) * Y^T \quad \text{for} \quad i = 1:m \tag{5.19}$$

$$\tilde{A}(i,q) = A(i,q) - A(i,p) * Y \quad \text{for} \quad i = 1:m \tag{5.20}$$

$$\tilde{A}(p,j) = A(p,j) + Y * A(q,j) \quad \text{for} \quad j = 1:m \tag{5.21}$$

$$\tilde{A}(q,j) = A(q,j) - Y^T * A(p,j) \quad \text{for} \quad j = 1:m \tag{5.22}$$

We are assumuing that throughout the sweep, the block Y is of order ϵ and each block $A(i,j)$ also has norm of order ϵ. The right and left transforms are followed by a normalization but this has little effect on the order of the terms. Figure (5.1) shows what happens when the 1st row in the sweep is processed. After (1,2) is eliminated and we proceed with eliminating block (1,3) the block (1,2) undergoes a row transformation of the form (5.21), namely $A(1,2) = A(1,2) + Y * A(3,2)$. Thus, block $(1,2)$ which was previously zeroed out, has now been filled-up with nonzero entries. However, because $A(1,2)$ is initially zero, and Y and $A(3,2)$ are both of order ϵ the block is replaced by terms of order ϵ^2. It can be verified that the same process is continued : either a zero block or a block of order ϵ^2 is combined with a product of a Y matrix and a block of A both of which are of order ϵ. For example in the next subspace rotation, namely rotation (1,4), the same block $A(1,2)$ which now has terms of order ϵ^2 is modified as $A(1,2) = A(1,2) + Y * A(3,2)$ so the terms will remain of order ϵ^2. It may be possible to formalize this with actual bounds but the complexity of the resulting bounds are not worth it. It may also be possible to exploit this knowledge in order to speed-up the convergence, but one must realize that this analysis is valid when we are near the end of the process at which point convergence is quite fast.

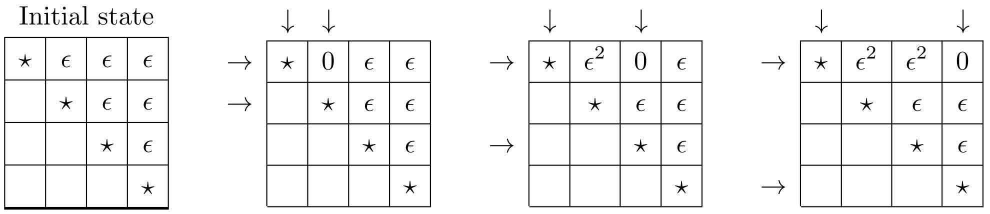

FIG. 5.1. *Elimination of blocks in first row: Rotations (1,2), (1,3), and (1,4)*

$$
\begin{array}{c|c|c|c|c|}
 & & \downarrow & \downarrow & \\
\hline
 & \star & \epsilon^2 & \epsilon^2 & 0 \\
\hline
\rightarrow & & \star & 0 & \epsilon \\
\hline
\rightarrow & & & \star & \epsilon \\
\hline
 & & & & \star \\
\hline
\end{array}
\qquad
\begin{array}{c|c|c|c|c|}
 & & \downarrow & & \downarrow \\
\hline
 & \star & \epsilon^2 & \epsilon^2 & \epsilon^2 \\
\hline
\rightarrow & & \star & \epsilon^2 & 0 \\
\hline
 & & & \star & \epsilon \\
\hline
\rightarrow & & & & \star \\
\hline
\end{array}
\qquad
\begin{array}{c|c|c|c|c|}
 & & & \downarrow & \downarrow \\
\hline
 & \star & \epsilon^2 & \epsilon^2 & \epsilon^2 \\
\hline
 & & \star & \epsilon^2 & \epsilon^2 \\
\hline
\rightarrow & & & \star & 0 \\
\hline
\rightarrow & & & & \star \\
\hline
\end{array}
$$

FIG. 5.2. *Elimination of blocks in 2nd and 3rd row: rotations (2,3), (2,4) and (3,4)*

6. Experiments.

The next experiments will illustrate some features of the algorithms described in this paper with a few small examples. All experiments are conducted in Matlab and we wish to emphasize that a parallel implementation would take time to develop and it is not within our scope.

6.1. Effect of the block-size.

We begin by illustrating the effect of block-size on convergence. Inequality (5.5) suggests that when m is smaller, i.e., when the block-size ν is larger, then convergence may be faster. Figure 6.1 seems to confirm this. The test involves a randomly generated matrix of size $N = 256$. We use our default algorithm which is the subspace approach of Section 3.4 with the Max-volume selection of the subspace. The block sizes used are $\nu = 4, 16, 64$. The 3 plots show the 2-norm of the matrix without its block-diagonal submatrices after each iteration.

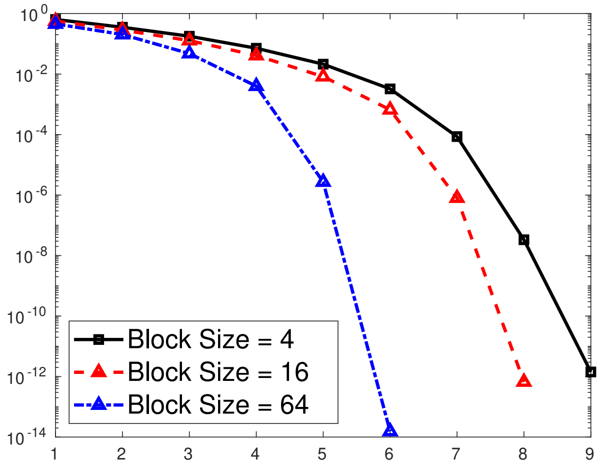

FIG. 6.1. *Comparison of convergence as the block-size changes*

6.2. Comparing the accuracy of a few methods.

In the following experiment, we explore the accuracy of a few of the methods discussed earlier from the point of view of their accuracy. We generate a 512 × 512 random symmetric matrix from which we extract the 5 principal square matrices of sizes: 32, 64, 128, 256, and 512. Along with these we select the block sizes 4, 8, 8, 16, 16 in the same order. We then apply 5 different methods based on the block Jacobi approach to each of these matrices:

1. Newton-Riccati iteration
2. Subspace approach - where the Max. Volume technique is applied

3. Subspace approach - where no Max. Volume technique is applied, i.e., the subspace selected corresponds to the one associated with the largest eigenvalues.
4. Manifold subspace iteration approach described *at the end of Section 3.3*
5. Classical block Jacobi method.

Note that the Newton-Riccati iteraion (1.) and the manifold subspace iteration (4.) both require solving Sylvester equations. In our implementation this is accomplished by calls to the Matlab function `sylvester()`. Assuming the eigenvalues are sorted in the same fashion, for each method we plot the average error:

$$\frac{1}{N}\sum_{i=1}^{N}\frac{|\lambda_i(\hat{A})-\lambda_i(A)|}{\lambda_i(A)},$$

where $\hat{A}$ is the block-diagonal matrix obtained from the method (the small entries outside the block-diagonal form are explicitly set to zero to reflect what is normally done in practice).

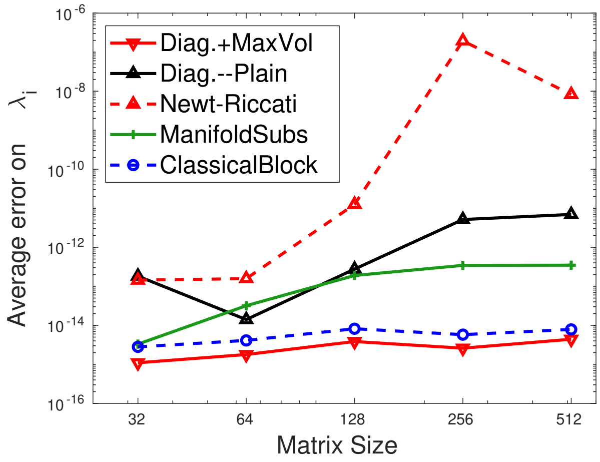

Fig. 6.2. *Comparing accuracies produced by five methods on matrices of increasing size*

The results are shown in Figure 6.2.

6.3. An application: Perturbed matrix case. The goal of the next experiment is to illustrate a technique that can be effective when applied with the proposed approach but that has no equivalent with the standard block approach. Let us assume that A has already been fully diagonalized by any method, i.e., we end up with

$$A = Q\Lambda Q^T, \tag{6.1}$$

where Λ is diagonal and Q is orthogonal. A common practical situation that arises is when one is interested in eigenvalues of a matrix obtained from A by slightly perturbing it in some locations, i.e., by adding a sparse symmetric matrix E to A:

$$\tilde{A} = A + E = Q(\Lambda + F)Q^T \quad \text{with} \quad F = Q^T E Q. \tag{6.2}$$

In what follows we assume that E is of small magnitude, e.g. $\|E\|/\|A\| \approx 10^{-2}$, but this can be alleviated. The goal then is to compute, for example, the largest ν eigenvalues of $\tilde{A}$, inexpensively. In fact a more interesting scenario is when A undergoes many consecutive such perturbations and the problem is to track the corresponding largest ν eigenvalues and vectors. While there are possible solutions that involve shift-and-invert iterations to refine the eigenvectors, the procedure based on a simplified version of the block Jacobi algorithm described in this paper provides a simpler and possibly more reliable technique.

Since our goal is to compute the dominant eigenvalues of

$$B = \Lambda + F \tag{6.3}$$

the first step is to compute B. If we are interested in the largest eigenvalues when we may have to apply a permutation in such a way that the largest entries of Λ are in the leading $\nu \times \nu$ block. It is assumed that this permutation is applied to B and Q accordingly. With this we cast B in the form

$$B = \begin{bmatrix} H & G \\ G^T & M \end{bmatrix}, \tag{6.4}$$

where now B is of dimension $N \times N$. Our goal is simply to annihilate the block G. In the notation of Section2, we have: $i_1 = 1, i_2 = \nu, j_1 = \nu + 1, j_2 = N$.

If we wish to apply Algorithm 1 then we note that the Sylvester equation (3.10) can be solved very effectively because the matrices $H + Y_k G^T$ and $M - G^T Y_k$ will remain nearly diagonal. Indeed, H and M are close to diagonal matrices and the sequence of approximate solutions as well as G are going to be small. As a result *we can solve the equations approximately by ignoring the off-diagonal entries*:

$$\begin{aligned} &\text{Set:} && H_k^{(D)} = \text{Diag}[H + Y_k G^T] \\ &\text{Set:} && M_k^{(D)} = \text{Diag}[M - G^T Y_k] \\ &\text{Solve for } \Delta: && H_k^{(D)} \Delta - \Delta M_k^{(D)} = R_k. \end{aligned} \tag{6.5}$$

What this means is that when viewing the Sylvester equation as a large linear system of size as $\nu.(N - \nu)$ then the whole system is nearly diagonal and we solve it approximately by ignoring its off-diagonal entries. We have the option of employing iterative refinement steps, until convergence to solve the linear system to some accuracy but we will proceed differently by employing Algorithm 1 in which we solve the diagonalized equation (6.5) instead of (3.10) in Line 3. However, since the Sylvester equation is solved only approximately, the update of the residual in Line 5 is no longer valid and we must use the explicit formula (3.9) instead which can be conveniently re-written as

$$R_k = (G - HY_k) + Y_k(M - Y_k G^T). \tag{6.6}$$

Algorithm 4 Diagonalized Newton-Riccati method for solving (2.7)

```
Start: Select $Y_0$ and compute $R_0$ from (3.9)
while $\|R_k\| > tol$ do
    Find $\Delta$ solution of (6.5)
    Compute $Y_{k+1} := Y_k + \Delta$
    Compute $R_{k+1}$ using (6.6)
end while
```

Note that, per Section 5.1, the cost of one iteration is $\approx 4\nu^2 N$ and a very small number of iteration (usuall 2 or 3) are needed. This is illustrated in the next experiment.

The following experiment deals with a model covariance matrix. Given sample points in $\mathbb{R}^2$ or $\mathbb{R}^3$ a model covariance matrix C is defined from a certain covariance function $k(r)$ as $C_{ij} = k(r_{ij})$ where R_{ij} is the distance between points i and j.

A popular choice for $k(r)$ is the *Matérn* covariance function given by [2]

$$k(r) = \frac{2^{1-\mu}}{\Gamma(\mu)} \left(\frac{\sqrt{2\mu} r}{l} \right)^{\mu} K_{\mu} \left(\frac{\sqrt{2\mu} r}{l} \right)$$

where K_μ is the modified Bessel function of the second kind. This function has two parameters: μ and l. In our example we take $\mu = 0.2, l = 0.1$. The sample points are those of a regular 32×32 grid on the square $[0,\ 1]^2$ of $\mathbb{R}^2$, leading to a dense matrix of size $N = 1024$.

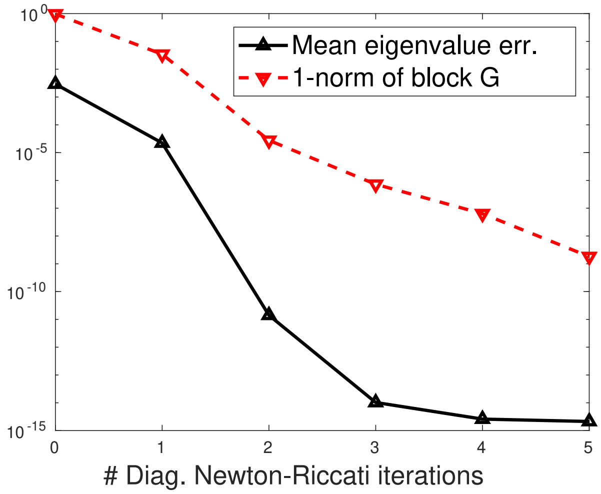

FIG. 6.3. *The Diagonaized Riccati-Newton block Jacobi method*

In the experiment, we diagonalize A and perturb it by random sparse generated by the matlab commands: `E = sprandn(N,N,5/N);` and `E = 1.e-02*(E+E');` The goal is to reduce the norm of the block G in (6.4), ideally to zero. In the experiments we take $\nu = 32$, i.e., we are interested in the largest 32 eigenvalues. We then apply Algorithm 4 to the matrix (6.3) where the `while` loop is replaced by a fixed number of iterations: We perform $0, 1, 2, \cdots, 5$ iterations of the loop that comprises Lines 3–5. Figure 6.3 show two curves. The first one plots the average error obtained on the $\nu = 32$ largest eigenvalues:

$$\frac{1}{\nu} \sum_{i=1}^{\nu} \frac{|\lambda_i(\hat{H}) - \lambda_i(B)|}{\lambda_i(B)},$$

where a hat symbol indicates a block after the transformation resulting from Algorithm 4 is applied. Along with this we also plot the 1-morm of the matrix $\hat{G}$ after

[2] In the commun notation used in the litterature, the parameter μ is replaced by ν.

the Jacobi rotation is performed (or just G when the number of iterations is zero). As can be seen within two iterations, the improvement to the eigenvalues is rather substantial. After 3 iterations, the full accuracy is achieved for the eigenvalues. At the same time the norm of the $(1,2)$ block $\hat{G}$ decreases gradually, diminishing by 2 to 3 orders of magnitude with each additional iteration.

We need to add that we may end-up in a situation where eigenvalues switch from the H block to the M block. We called this 'change of affiliation' in Section 5.3 and theorem 5.3 provides some theoretical guarantees under which this cannot happen.

We mentioned earlier that an alternative to the approach discussed in this section would be to apply a shift-and-invert technique to refine each eigenvector in turn. The key point that is exploited in both techniques is that the equations resulting from applying the procedure are rather easy to solve due to almost diagonal structure of the coefficient matrices involved. However, this particular alternative is not without issues if only because computing or refining individual eigenvectors can be difficult in the presence of clustered eigenvalues. Recently, Ogita and K. Aishima [39, 40] proposed another technique along the same lines as ours. Their approach takes a perturbative viewpoint to develop algorithms that also take advantage of the inexpensive nature of the sub-problems that need to be solved in the process. Although the main ideas behind the two methods are rather different, the resulting equations that need to be solved are similar.

7. Conclusion. The block-Jacobi method has a number of appealing features and has the potential to be a competitor to the usual Householder-QR combination in suitable, e.g., GPU-based, high-performance computer environments. The proposed alternative scheme discussed in this paper has some additional appealing features some of which stem from the more progressive nature of the method when compared with the classical one. The example in Section 6.3 in particular shows what can be done in a special situation where the matrix is slightly perturbed. There are many issues that have not yet been explored. In particular a full-fledged parallel GPU-based implementation is still lacking. The perturbation-based scheme of Section 6.3 points to a class of methods for tracking subspaces of slowly varying matrices, where the proposed scheme can play a role. More generally, the overall flexibility of the approach may prove useful in other similar special situations.

Data Availability. The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

Conflict of interest. The author declares that he has no conflict of interest.

Acknowledgments. The paper has benefitted from a careful reading from an anonymous referee who made a number of suggestions and pointed out two references on related work.

REFERENCES

[1] P. A. Absil, R. Mahoney, R. Sepulchre, and P. Van Dooren, *A grassmann–rayleigh quotient iteration for computing invariant subspaces*, SIAM Review, 44 (2002), pp. 57–73.

[2] D. A. Bini, B. Iannazzo, and F. Poloni, *A fast newton's method for a nonsymmetric algebraic riccati equation*, SIAM Journal on Matrix Analysis and Applications, 30 (2008), pp. 276–290.

[3] C. Bischof and C. Van Loan, *Computing the singular value decomposition on a ring of array processors*, in Large Scale Eigenvalue Problems, J. Cullum and R. A. Willoughby, eds., vol. 127 of North-Holland Mathematics Studies, North-Holland, 1986, pp. 51–66.

[4] C. H. Bischof, B. Lang, and X. Sun, *A framework for symmetric band reduction*, ACM Trans. Math. Softw., 26 (2000), pp. 581–601.

[5] S. Bittanti, A. Laub, and J. C. W. (eds.), *The Riccati equation*, Springer Verlag, Berlin, 1991.

[6] A. Björk, *Numerical methods in matrix computations*, Texts in applied mathematics, vol. 59, Springer, 2015.

[7] J. Brandts, *The riccati algorithm for eigenvalues and invariant subspaces of matrices with inexpensive action*, Linear Algebra and its Applications, 358 (2003), pp. 335–365.

[8] J. H. Brandts, *The riccati method for eigenvalues and invariant subspaces of matrices with inexpensive action*, Linear Algebra Appl., 358 (2003), pp. 333–363.

[9] F. Chatelin, *Simultaneous Newton's iteration for the eigenproblem*, in Defect Correction Methods. Computing Supplementum, vol 5, S. H. Böhmer K., ed., Vienna, 1984, Springer.

[10] J. Demmel and K. Veselić, *Jacobi's method is more accurate than QR*, SIAM Journal on Matrix Analysis and Applications, 13 (1992), pp. 1204–1245.

[11] J. W. Demmel, *Three methods for refining estimates of invariant subspaces*, Computing, 38 (1987), pp. 43–57.

[12] A. Deshpande and L. Rademacher, *Efficient volume sampling for row/column subset selection*, in 2010 IEEE 51st Annual Symposium on Foundations of Computer Science, 2010, pp. 329–338.

[13] A. Deshpande, L. Rademacher, S. Vempala, and G. Wang, *Matrix approximation and projective clustering via volume sampling*, Theory of Computing, 2 (2006), pp. 225–247.

[14] Z. Drmač, *A global convergence proof for cyclic Jacobi methods with block rotations*, SIAM Journal on Matrix Analysis and Applications, 31 (2010), pp. 1329–1350.

[15] A. Edelman, T. A. Arias, and S. T. Smith, *The geometry of algorithms with orthogonality constraints*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 303–353.

[16] G. E. Forsythe and P. Henrici, *The cyclic Jacobi method for computing the principal values of a complex matrix*, 1960.

[17] J. G. F. Francis, *The QR transformations, parts i and ii*, Computer J., 4 (1961-1962), pp. 362–363, and 332–345.

[18] G. Freiling, *A survey of nonsymmetric riccati equations*, Linear Algebra and its Applications, 351-352 (2002), pp. 243–270. Fourth Special Issue on Linear Systems and Control.

[19] G. Freiling, G. Jank, and H. Abou-Kandil, *On global existence of solutions to coupled matrix riccati equations in closed-loop nash games*, IEEE Transactions on Automatic Control, 41 (1996), pp. 264–269.

[20] G. H. Golub and C. F. V. Loan, *Matrix Computations, 4th edition*, Johns Hopkins University Press, Baltimore, MD, 4th ed., 2013.

[21] S. Goreinov, E. Tyrtyshnikov, and N. Zamarashkin, *A theory of pseudoskeleton approximations*, Linear Algebra and its Applications, 261 (1997), pp. 1–21.

[22] S. A. Goreinov, I. V. Oseledets, D. V. Savostyanov, E. E. Tyrtyshnikov, and N. L. Zamarashkin, *How to Find a Good Submatrix*, 2010, pp. 247–256.

[23] J. Gotze, S. Paul, and M. Sauer, *An efficient Jacobi-like algorithm for parallel eigenvalue computation*, IEEE Transactions on Computers, 42 (1993), pp. 1058–1065.

[24] C. Guo and N. J. Higham, *Iterative solution of a nonsymmetric algebraic riccati equation*, SIAM Journal on Matrix Analysis and Applications, 29 (2007), pp. 396–412.

[25] C.-H. Guo and A. J. Laub, *On a newton-like method for solving algebraic riccati equations*, SIAM Journal on Matrix Analysis and Applications, 21 (2000), pp. 694–698.

[26] V. Hari, *Globally convergent jacobi methods for positive definite matrix pairs*, Numerical Algorithms, 79 (2018), pp. 221–249.

[27] V. Hari, *On the global convergence of the block jacobi method for the positive definite generalized eigenvalue problem*, Calcolo, 58 (2021), p. 24.

[28] V. Hari, S. Singer, and S. Singer, *Full block j-jacobi method for hermitian matrices*, Linear Algebra and its Applications, 444 (2014), pp. 1–27.

[29] C. G. J. Jacobi, *über ein lechtes verfahren, die in der theorie säculastrüngen vorkommenden gleichungen numerisch aufzulösen*, Journal für die reine und angewandte Mathematik, Crelle's journal, 30 (1846), pp. 51–94.

[30] S. Kudo, K. Yasuda, and Y. Yamamoto, *Performance of the parallel block jacobi method with dynamic ordering for the symmetric eigenvalue problem*, JSIAM Letters (Japan Society for Industrial and Applied Mathematics), 10 (2018), p. 41–44.

[31] A. Laub, *A Schur method for solving algebraic Riccati equations*, IEEE Transactions on Au-

tomatic Control, 24 (1979), pp. 913–921.

[32] A. J. Laub, *Schur techniques in invariant imbedding methods for solving two point boundary value problems*, in Proceedings of the 21-st conference on decision and control, Orlando, Florida, Dec. 1982, 1982.

[33] C. V. Loan, *The block Jacobi method for computing the singular value decomposition*, in Computational and Combinatorial Methods in Systems Theory, C. Byrnes and A. Lindquist, eds., Amsterdam, 1986, Elsevier Science Publishers B.V. (North-Holland), pp. 245–255.

[34] G. Loizou, *On the Quadratic Convergence of the Jacobi Method for Normal Matrices*, The Computer Journal, 15 (1972), pp. 274–276.

[35] F. T. Luk and H. Park, *On parallel Jacobi orderings*, SIAM Journal on Scientific and Statistical Computing, 10 (1989), pp. 18–26.

[36] M. W. Mahoney et al., *Randomized algorithms for matrices and data*, Foundations and Trends® in Machine Learning, 3 (2011), pp. 123–224.

[37] A. Marek, V. Blum, R. Johanni, V. Havu, B. Lang, T. Auckenthaler, A. Heinecke, H.-J. Bungartz, and H. Lederer, *The ELPA library: scalable parallel eigenvalue solutions for electronic structure theory and computational science*, Journal of Physics: Condensed Matter, 26 (2014), p. 213201.

[38] L. Nazareth, *On the convergence of the cyclic jacobi method*, Linear Algebra and its Applications, 12 (1975), pp. 151–164.

[39] T. Ogita and K. Aishima, *Iterative refinement for symmetric eigenvalue decomposition*, Japan Journal of Industrial and Applied Mathematics, 35 (2018), pp. 1007–1035.

[40] ———, *Iterative refinement for symmetric eigenvalue decomposition ii: clustered eigenvalues*, Japan Journal of Industrial and Applied Mathematics, 36 (2019), pp. 435–459.

[41] B. Philippe and Y. Saad, *On correction equations and domain decomposition for computing invariant subspaces*, Computer Methods in Applied Mechanics and Engineering (special issue devoted to Domain Decomposition), 196 (2007), pp. 1471–1483.

[42] F. Robert, *Blocs-h-matrices et convergence des methodes iteratives classiques par blocs*, Linear Algebra and its Applications, 2 (1969), pp. 223–265.

[43] A. Ruhe, *On the quadratic convergence of the jabobi method for normal matrices*, BIT Numerical Mathematics, 7 (1967), pp. 305–313.

[44] A. H. Sameh, *On Jacobi and Jacobi-like algorithms for a parallel computer*, Math. Comp., 25 (1971), pp. 579–590.

[45] P. Van Dooren, *A generalized eigenvalue approach for solving Riccati equations*, SIAM Journal on Scientific and Statistical Computing, 2 (1981), pp. 121–135.

[46] H. P. M. van Kempen, *On the quadratic convergence of the special cyclic Jacobi method*, Numer. Math., 9 (1966), p. 19–22.

[47] J. H. Wilkinson, *Note on the quadratic convergence of the cyclic Jacobi process*, Num. Math., 4 (1962), p. 296–300.

[48] ———, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, 1965.

[49] Y. Yamamoto, Z. Lan, and S. Kudo, *Convergence analysis of the parallel classical block jacobi method for the symmetric eigenvalue problem*, JSIAM Letters (Japan Society for Industrial and Applied Mathematics), 6 (2014), pp. 57–60.

[50] Y. Yamamoto, G. Okša, and M. Vajteršic, *On convergence to eigenvalues and eigenvectors in the block-jacobi EVD algorithm with dynamic ordering*, Linear Algebra and its Applications, 622 (2021), pp. 19–45.

[51] A. Çivril and M. Magdon-Ismail, *On selecting a maximum volume sub-matrix of a matrix and related problems*, Theoretical Computer Science, 410 (2009), pp. 4801–4811.