# Algorithms and convergence of the Jacobi-type methods

**Perković, Ana**

University of Zagreb

FACULTY OF SCIENCE
DEPARTMENT OF MATHEMATICS

Ana Perković

# Algorithms and Convergence of the Jacobi-type Methods

DOCTORAL THESIS

Zagreb, 2025.

University of Zagreb

FACULTY OF SCIENCE

DEPARTMENT OF MATHEMATICS

Ana Perković

# Algorithms and Convergence of the Jacobi-type Methods

DOCTORAL THESIS

Supervisor:

Assoc. Prof. Erna Begović Kovač

Zagreb, 2025.

Sveučilište u Zagrebu

PRIRODOSLOVNO–MATEMATIČKI FAKULTET

MATEMATIČKI ODSJEK

Ana Perković

# Algoritmi i konvergencija metoda Jacobijeva tipa

DOKTORSKI RAD

Mentor:

izv. prof. dr. sc. Erna Begović Kovač

Zagreb, 2025.

# ACKNOWLEDGEMENTS

First, I want to express my gratitude to my supervisor Erna Begović Kovač for her guidance, patience, and insightful feedback throughout my Ph.D. journey. I also thank Nela Bosner, Zlatko Drmač, and Zoran Tomljanović who read my thesis and offered constructive feedback and suggestions. I gratefully acknowledge the financial support of Croatian Science Foundation under the project 5200 during my Ph.D. research. I want to thank my parents who always believed in me. Thanks to my brother, and Ivana for being like a sister to me. Lastly, thank you, Petar and Lovre, this accomplishment would not have been possible without your love and support.

# SUMMARY

The Jacobi eigenvalue algorithm is a well-known iterative method used for solving the eigenvalue problem of symmetric matrices. The process is based on matrix diagonalization. In this thesis we study several modifications of the Jacobi method. We work on both matrix and tensor numerical problems. First, we review and generalize the Eberlein method, which is a Jacobi-type method for diagonalization of an arbitrary matrix. We prove the global convergence of the Eberlein method under a broad class of generalized serial pivot strategies with permutations. Moreover, we discuss the cases of unique and multiple eigenvalues. Next, we consider block-partitioned matrices and introduce a block version of the Eberlein method. We give a convergence proof for the block Eberlein algorithm under the already mentioned class of generalized serial pivot strategies. Lastly, we study the methods for approximate tensor diagonalization. We propose an iterative Jacobi-type trace maximization algorithm for solving this problem on general tensors, as well as the structure-preserving variant for the symmetric tensors. We prove the global convergence for both of our algorithms. All theoretical work is accompanied by numerous numerical examples.

**Keywords**: Jacobi-type method, matrix diagonalization, pivot strategies, global convergence; tensor diagonalization.

# SAŽETAK

Jacobijev algoritam je poznata iterativna metoda za rješavanje problema svojstvenih vrijednosti za simetrične matrice. Postupak se temelji na dijagonalizaciji matrice. U ovoj se disertaciji bavimo modifikacijama Jacobijeve metode koje koristimo za rješavanje numeričkih problema za matrice i tenzore. U prvom dijelu rada proučavat ćemo Eberleininu metodu Jacobijevog tipa za dijagonalizaciju opće matrice. Poopćit ćemo Eberleininu metodu i dati dokaz globalne konvergencije za široku klasu tzv. generaliziranih serijalnih pivotnih strategija s permutacijama. Analizirat ćemo slučaj jednostrukih i višestrukih svojstvenih vrijednosti. Nadalje, promatrat ćemo matrice s blok-particijom te uvesti blok verziju Eberleinine metode. Dat ćemo dokaz konvergencije blok Eberleininog algoritma za već spomenutu klasu generaliziranih serijalnih pivotnih strategija. Naposljetku, promatrat ćemo problem približne dijagonalizacije tenzora. Predstavit ćemo iterativni algoritam Jacobijevog tipa temeljen na maksimizaciji traga tenzora. Konstruirat ćemo algoritam za opće tenzore te njegovu varijantu za simetrične tenzore u kojoj je očuvana polazna simetrična struktura. Za oba algoritma dokazat ćemo globalnu konvergenciju. Svi teorijski rezultati će biti popraćeni brojnim numeričkim primjerima.

Disertacija je podijeljena u četiri poglavlja. U prvom poglavlju dan je osvrt na rezultate iz literature. Opisan je realni i kompleksni Jacobijev algoritam za rješavanje problema svojstvenih vrijednosti i izvedene su formule za računanje kuteva transformacije. Opisano je nekoliko klasa pivotnih strategija. Poglavlje se nastavlja teorijom o Jacobijevim anihilatorima i operatorima koji se koriste u brojnim rezultatima o konvergenciji za standardnu Jacobijevu metodu i za druge metode Jacobijeva tipa. Koristit ćemo tu teoriju za dokaz konvergencije Eberleinine metode po elementima, ali i njene blok varijante. Prvo poglavlje završava osvrtom na teoriju o konvergenciji Jacobijeva algoritma.

Drugo poglavlje temelji se na članku [10] od Begović Kovač i Perković objavljenom

2024. godine. Prvo je dan pregled postojećih rezultata o konvergenciji Eberleinine metode, za realni i kompleksni slučaj. Glavni dio poglavlja je proširenje globalne konvergencije metode na široku klasu generaliziranih serijalnih pivotnih strategija. Pokaže se da, za proizvoljnu početnu matricu $A$, Eberleinina metoda konvergira uz bilo koju strategiju iz navedene klase. Niz matrica $A^{(k)}$, $k \geq 0$, koji se dobije nakon svake iteracije, konvergira prema normalnoj matrici. Niz hermitskih dijelova dobivenih matrica, $\left(A^{(k)} + (A^{(k)})^*\right)/2$, $k \geq 0$, konvergira prema dijagonalnoj matrici takvoj da su na dijagonali realni dijelovi svojstvenih vrijednosti od $A$. Ako sve svojstvene vrijednosti od $A$ imaju različite realne dijelove, niz $A^{(k)}$ konvergira prema dijagonalnoj matrici sa svojstvenim vrijednostima od $A$ na dijagonali. Inače, svojstvene vrijednosti s jednakim realnim dijelovima mogu dovesti do ne-nul van-dijagonalnih elemenata u dobivenoj matrici. Kroz numeričke primjere testirana je metoda na realnim i kompleksnim matricama, za početne matrice koje su unitarno dijagonalizabilne i za one koje to nisu. Promatrana je promjena u matričnoj van-dijagonalnoj normi, tj., udaljenosti od dijagonalne matrice. Nadalje, pokazana je blok struktura koja se pojavljuje ako početna matrica ima višestruke svojstvene vrijednosti. Naposljetku, pokazano je kako numerički riješiti problem kod ponavljajućih svojstvenih vrijednosti.

Treće poglavlje sadrži prijedlog novog blok Eberleininog algoritma. Dan je kratki uvod u blok matrice i blok algoritme. Opisana je blok verzija Eberleinine metode u kojoj su matrice podijeljene u blokove. Zatim, predložen je način za računanje transformacija $\mathbf{R}_k$ i $\mathbf{S}_k$, $k \geq 0$, te dan dokaz konvergencije algoritma uz generalizirane serijalne pivotne strategije. Rezultati konvergencije su u skladu s onima za Eberleininu metodu po elementima. Numeričkim testovima pokazano je kako blok algoritam radi za različite veličine blokova i za ponavljajuće realne dijelove svojstvenih vrijednosti.

Četvrto i posljednje poglavlje orijentirano je na približnu dijagonalizaciju tenzora. Temelji se na članku [11] od Begović Kovač i Perković objavljenom 2024. godine. Ovdje je detaljno objašnjena terminologija i pojmovi vezani uz tenzore koji se koriste u tenzorskom računu. Prvo je dan osvrt na postojeće algoritme za dijagonalizaciju tenzora. Zatim je iznesen prijedlog algoritma koji se temelji na maksimizaciji traga tenzora, kao u [65]. Algoritam koristi metodu alternirajućih najmanjih kvadrata (ALS). Naime, jedna iteracija algoritma na tenzoru reda $d$ sastojat će se od $d$ mikroiteracija. Pokazana

je globalna konvergencija našeg algoritma za opće tenzore. Preciznije, pokazano je da je svako gomilište dobiveno našim algoritmom stacionarna točka funkcije koju maksimiziramo. Ovaj rezultat istog je tipa kao rezultati konvergencije algoritama za dijagonalizaciju tenzora koji se baziraju na maksimizaciji Frobeniusove norme dijagonale tenzora. Konvergencija vrijedi za sve cikličke strategije uz dodatni uvjet na pivotni par $(p,q)$, zvan Łojasiewitzeva nejednakost gradijenta. Nadalje, naš algoritam maksimizacije traga prilagođen je kako bi se očuvala struktura simetričnih tenzora. U tom slučaju, svih $d$ rotacija koje djeluju u jednoj iteraciji moraju biti iste. Prema tome, ovo više nije ALS algoritam jer se trag maksimizira po svim modovima istovremeno. Ipak, dokaz konvergencije će ići uz bok dokazu za algoritam koji ne čuva strukturu. Numerički primjeri uključuju testove oba algoritma na tenzorima različitih redova, dijagonalizabilnih tenzora i onih koji to nisu. Promatrano je povećanje traga tenzora i smanjenje van-dijagonalne norme tenzora, te su dane usporedbe za različite cikličke pivotne strategije.

**Ključne riječi**: Metoda Jacobijeva tipa, dijagonalizacija matrice, pivotne strategije, globalna konvergencija, dijagonalizacija tenzora.

# CONTENTS

# INTRODUCTION

The Jacobi eigenvalue algorithm was initially proposed in 1846 by C. G. J. Jacobi, and rediscovered in the mid 20th century upon the appearance of modern computers. This is an iterative method for solving the symmetric eigenvalue problem with high relative accuracy, [23, 24, 62, 70]. Besides being known for its simplicity, it is well suited for parallelization, [57, 69]. The method has been modified to deal with different matrix structures, e.g., Hamiltonian matrices in [30], J-symmetric matrices in [60], Hermitian matrices in [41,63], matrices in anti-triangular Schur form in [59]. A variant of the Jacobi algorithm developed in [26] and [27] and used to compute the SVD of a general matrix outperformed other algorithms, for example QR algorithm in terms of speed, but also retained high relative accuracy property. The convergence of the Jacobi method has been studied by many authors, see e.g., [9, 38, 56, 58, 61], and the references therein.

For a starting matrix $A$, the main idea of the Jacobi method is to find the sequence of rotation matrices that, when applied to $A$ from both left (transposed rotation) and right (rotation), result with a diagonal matrix $D$, such that the diagonal entries of $D$ are eigenvalues of $A$. As the Frobenius norm is invariant to the orthogonal transformations, in each iterative step rotations $R_k$, $k \geq 0$, can be chosen to annihilate pivot element of the underlying matrix and, consequently, increase the sum of squares of the diagonal elements. This process is repeated for different pivot pairs until a diagonal matrix, or a good approximation of a diagonal matrix is obtained.

In this thesis, we are going to study Jacobi-type algorithms for the (approximate) matrix, block matrix, and tensor diagonalization. Matrices are denoted by capital letters, e.g. $A$, $B$, $C$. When we observe block-partitioned matrices, we use bold capital letters, for example $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$. On the other hand, tensors are denoted by calligraphic capital letters, e.g. $\mathscr{A}$, $\mathscr{B}$, $\mathscr{C}$. Our methods employ the main idea of the Jacobi eigenvalue algorithm.

Namely, they are all iterative and, in each iteration step, in the matrix case, a transformation is applied from the left- and from the right-hand side. In the tensor case, this means that a transformation is applied in all modes. In the Jacobi-type algorithms, one must first determine a suitable form of the transformation. Then, in each iteration, one must find the optimal transformation coefficients. Since these methods are iterative, their important property is convergence.

The Jacobi method and each of the Jacobi-type methods depend on a pivot strategy that defines an order in which the pivot positions are selected. The possible pivot positions are those in the upper triangle of a matrix and they determine the transformation matrices. Cyclic pivot strategies are the strategies in which, in the first set of iterations (first cycle), we take all possible pivot positions exactly once in some prescribed order. Then, pivot positions are repeated cycle-by-cycle, until convergence. The most well-known cyclic strategies are serial strategies. In the row-wise serial pivot strategy, pivot positions are taken row-by-row, from the first to the second to last row, and inside each row positions are taken from left to right. Similarly, in the column-wise serial pivot strategy, pivot positions are taken column-by-column from the second to last column, and in each column, positions are taken from top to bottom. In the matrix case, we are going to work with the so-called generalized serial pivot strategies from [39]. In the tensor case, our convergence proofs will be valid for any cyclic pivot strategy.

In contrast to the Jacobi method which solves the symmetric eigenvalue problem, the Eberlein method, proposed by Eberlein [28] in 1962, is a Jacobi-type diagonalization process for solving the eigenvalue problem on an arbitrary complex matrix. For a starting matrix $A$, in each step of the iterative process, transformation $T_k$, $k \geq 0$, is applied to the underlying matrix from the right-hand side, and inverse transformation is applied from the left-hand side. This transformation $T_k$ is constructed as a product of two non-singular elementary matrices, $T_k = R_k S_k$. The matrix $R_k$, $k \geq 0$, is a plane rotation, while $S_k$, $k \geq 0$, is a non-unitary elementary matrix. Rotation $R_k$ is chosen to annihilate the pivot element of the Hermitian part of the underlying matrix. On the other hand, transformation $S_k$ reduces the Frobenius norm of the underlying matrix. Although the method is old, nowadays it is interesting because it is very suitable for parallelization.

Veselić [75] proved the convergence under the classical Jacobi pivot strategy, but for

a modified Eberlein method where, in each step, the transformation $T_k$ is either equal to $R_k$ or to $S_k$. Hari [35] proved the global convergence of the original method under serial pivot strategies on real matrices. Pupovci and Hari [67] studied the convergence of the complex Eberlein method under weak wavefront pivot strategies. They also considered parallelization for the Eberlein method and proved its convergence under pivot strategies weakly equivalent to the modulus strategy from [58]. We extend their convergence results.

In the standard Jacobi method, instead of eliminating one pivot element in one iteration step, we can annihilate an entire block of elements. This way we get a block Jacobi algorithm. Begović and Hari [9] have given the most general result when they proved the convergence of the block Jacobi method under generalized serial pivot strategies. In general, block algorithms are more efficient than their element-wise counterparts. That is the motivation for modifying the element-wise matrix algorithm into a block matrix one. In particular, we observe the Eberlein method and introduce its block variation. The transformations $R_k$, $k \geq 0$, which annihilate the pivot element are replaced by the block transformations $\mathbf{R}_k$, $k \geq 0$, which diagonalize the pivot block. The non-unitary elementary matrices $S_k$, $k \geq 0$, that reduce the Frobenius norm become non-unitary block elementary matrices $\mathbf{S}_k$, $k \geq 0$, that reduce the Frobenius norm of the block matrix. Up to now, there has been no convergence theory for the block Eberlein method.

Lastly, we study Jacobi-type diagonalization methods for tensors. Tensor diagonalization has applications in independent component analysis [53], and signal processing problems, like blind source separation, image denoising, etc. See, e.g., [17, 71]. The problem has been studied as the orthogonal [8,54,55,74], and non-orthogonal [71] tensor diagonalization, for structured and unstructured tensors.

Formally, a tensor is an element of a tensor product of vector spaces. One can look at it as a $d$-dimensional matrix, where the dimension $d$ is called the order of the tensor. Hence, a scalar, a vector, and a matrix are zero-order, first-order, and second-order tensors, respectively. When we refer to a tensor, we assume that its order is at least three. Instead of matrix rows and columns, tensor has fibers in $d$ sides. Each side, or dimension, of a tensor is called a mode. For an easier computation, we represent a tensor by a matrix. Mode-$m$ matricization of a tensor $\mathscr{A}$ is a matrix $A_{(m)}$ such that the columns of $A_{(m)}$ are mode-$m$ fibers of the tensor. There are different tensor decompositions. We work with

Tucker decomposition originally introduced in [44]. It is a representation of a tensor $\mathscr{A}$ as a product of a core tensor $\mathscr{S}$ and $d$ matrices, one in each mode.

If a tensor $\mathscr{A}$ allows orthogonal diagonalization, then the core tensor $\mathscr{S}$ obtained from the orthogonal Tucker decomposition is diagonal. However, it is not always possible to completely diagonalize a tensor using orthogonal transformations. Even in the symmetric case, contrary to the symmetric matrices, it is known that symmetric tensors generally cannot be orthogonally diagonalized. Therefore, in most cases, a diagonal tensor $\mathscr{S}$ is not achievable. Hence, our goal is to maximize tensor diagonal in a certain way. In [8, 54, 55, 74], the authors developed the Jacobi-type algorithms for maximizing the Frobenius norm of the diagonal of $\mathscr{S}$. Contrary to this approach, Moravitz Martin and Van Loan [65] worked with an algorithm that maximizes the trace of $\mathscr{S}$, but without the proof of convergence. Inspired by [65], we propose an algorithm based on tensor-trace maximization.

The thesis is divided into four chapters. Chapter 1 is an overview of the results from literature. We describe the standard Jacobi algorithm for the eigenvalue problem, both for the real and the complex case, derive the relations for calculating the transformation angles and describe several classes of pivot strategies. Then, we explain the theory of the Jacobi annihilators and operators which is used to obtain some of the convergence results for the standard Jacobi and other Jacobi-type methods. We are going to use it to prove the convergence of the Eberlein method, both for the element-wise and for the block matrices. We end the first chapter an overview of the convergence theory for the Jacobi algorithm.

Chapter 2 is based on the paper [10] by Begović Kovač and Perković published in 2024. First, we review the existing convergence results for the Eberlein method, both for the real and for the complex case. We extend the global convergence result to a broad class of cyclic pivot strategies, the generalized serial pivot strategies. We discuss the cases of the unique and the multiple eigenvalues. We prove that, for an arbitrary starting matrix $A$, the Eberlein method under any pivot strategy from the specified class converges. The sequence of matrices $A^{(k)}$, $k \geq 0$, obtained after each iteration converges to normal matrix. The sequence of the Hermitian parts of the obtained matrices, $\left(A^{(k)} + (A^{(k)})^*\right)/2$, $k \geq 0$, converges to a diagonal matrix, where the diagonal entries are the real parts of the eigenvalues of $A$. If all eigenvalues of $A$ have different real parts, then the sequence $A^{(k)}$

converges to a diagonal matrix with eigenvalues on the diagonal. Otherwise, the eigenvalues with equal real parts may lead to non-zero off-diagonal elements in the obtained matrix. Within the numerical experiments we test the method on both complex and real matrices, for the starting matrices that can be diagonalized using unitary transformations (normal matrices) as well as for the matrices that cannot be diagonalized this way (matrices that are not normal). We examine the change in the matrix off-norm, that is, the distance from a diagonal matrix. Moreover, we show the obtained block structure that appears if the starting matrix has multiple eigenvalues. Finally, we explain how to overcome the issue that appears for the multiple eigenvalues.

Chapter 3 contains the newly proposed block Eberlein algorithm. We give a short introduction to block matrices and block algorithms. We describe our block version of the Eberlein method in which all matrices are partitioned into blocks. Next, we suggest how the transformations $\mathbf{R}_k$ and $\mathbf{S}_k$, $k \geq 0$, should be taken. We prove convergence of the proposed algorithm under the generalized serial block pivot strategies. The convergence results are alongside those for the element-wise Eberlein method. Within the numerical tests, we show how the block algorithm performs for different block sizes and for repeating real parts of the eigenvalues.

In Chapter 4, we focus on the approximate diagonalization of tensors. This chapter is based on the paper [11] by Begović Kovač and Perković published in 2024. Here, we explain in detail the terminology and tensor notions used in tensor computations. First, we give an overview of the existing algorithms for tensor diagonalization. Then, we propose an algorithm based on maximizing the trace of a tensor, like it is done in [65]. The algorithm uses the alternate least squares (ALS) technique. Thus, one iteration of the algorithm on an order-$d$ tensor will be made of $d$ microiterations. We prove the global convergence of our algorithm for general tensors. More precisely, we prove that every accumulation point obtained by our algorithm is a stationary point of the objective function. This result is of the same type as the convergence results for the tensor diagonalization algorithms based on the maximization of the Frobenius norm of the diagonal. It holds for every cyclic strategy assuming an additional condition, called Łojasiewitz gradient inequality, on the pivot pair $(p,q)$. Moreover, we adapt our trace maximization algorithm to obtain a structure-preserving algorithm for symmetric tensors. In the structure-preserving

case, all $d$ rotations applied in one iteration are the same. Therefore, this is no longer an ALS algorithm because the maximization is pursued through all modes at once. However, the convergence theory will be alongside the non-structured algorithm. Our numerical experiments include tests on the tensors of different orders, both for diagonalizable and for non-diagonalizable tensors. We inspect the increase of the tensor trace and the decrease of its off-norm. We compare different cyclic pivot strategies.

# 1. JACOBI METHOD AND RELATED RESULTS

In this chapter, we review the renowned Jacobi method for matrix diagonalization [46] for the real as well as the complex matrix. We describe well-known pivot strategies with an emphasis on the broad class of generalized serial pivot strategies that we work with later. We set forth the theory of the Jacobi annihilators and operators commonly used to prove convergence results for the Jacobi method. We state the convergence theory of the Jacobi method from the literature.

## 1.1. JACOBI ALGORITHM

In 1846, Carl Gustav Jacob Jacobi in his work *Über ein leichtes Verfahren, die in der Theorie der Säkularstörungen vorkommenden Gleichungen numerisch aufzulösen* (*On a simple procedure for numerically solving the equations occurring in the theory of secular perturbations*) [46] proposed an iterative method for finding eigenvectors and eigenvalues of a real symmetric matrix. The method uses plane rotations to reduce the matrix to a diagonal form. Compared to the other state-of-the-art diagonalization methods, the main advantage of the Jacobi method is its high relative accuracy, [23, 24, 62, 70]. The method has been modified to deal with different matrix structures [30,41,59,60,63] and to address various problems of numerical linear algebra [15, 26, 27, 64]. Its convergence has been extensively studied; see, for example, [9, 38, 56, 58, 61].

### 1.1.1. Real Jacobi algorithm

The fundamental theorem of symmetric matrices states that a matrix $A \in \mathbb{R}^{n\times n}$ is symmetric if and only if it has real eigenvalues and is orthogonally diagonalizable. In other words, there exists an orthogonal matrix $U \in \mathbb{R}^{n\times n}$ such that

$$A = U^T D U,$$

where $D \in \mathbb{R}^{n\times n}$ is a diagonal matrix. The equation above justifies the form of the Jacobi iterations that we are going to describe now.

For a real symmetric matrix $A \in \mathbb{R}^{n\times n}$, the Jacobi method is the iterative process

$$A^{(k+1)} = U_k^T A^{(k)} U_k, \quad k \geq 0, \quad A^{(0)} = A, \tag{1.1}$$

where $U_k = R(p_k, q_k, \varphi_k)$ are plane rotations

$$R(p_k,q_k,\varphi_k) = \begin{bmatrix} 1 & & & & & & & & & \\ & \ddots & & & & & & & & \\ & & 1 & & & & & & & \\ & & & \cos\varphi_k & & & -\sin\varphi_k & & & p_k \\ & & & & 1 & & & & & \\ & & & & & \ddots & & & & \\ & & & & & & 1 & & & \\ & & & \sin\varphi_k & & & \cos\varphi_k & & & q_k \\ & & & & & & & 1 & & \\ & & & & & & & & \ddots & \\ & & & & & & & & & 1 \end{bmatrix}. \tag{1.2}$$

Orthogonal matrices $U_k$ are called *elementary matrices* [6], as they differ from the identity only in a $2\times 2$ submatrix. Sometimes, an elementary matrix is called a core transformation, [6]. They are determined by a *pivot pair* $(p_k, q_k)$ and a *rotation angle* $\varphi_k$. For a fixed $k$, we say that the transformation from (1.1) rotates the element at the position $(p_k, q_k)$.

The method uses plane rotations to reduce the matrix to a diagonal form. After each step $k$, matrices $A^{(k)}$ become "more diagonal". In order to describe the meaning of distance from the diagonal form, we define the matrix *off-norm*. The off-norm of $A$ is the Frobenius norm of its off-diagonal part,

$$\mathrm{off}(A) = \|A - \mathrm{diag}(A)\|_F = \sqrt{\sum_{\substack{i,j=1 \\ i\neq j}}^{n} {a_{ij}}^2}. \tag{1.3}$$

It is easy to see that $A$ is a diagonal matrix if and only if $\mathrm{off}(A) = 0$. Observe that the off-norm is not a matrix norm, as $\mathrm{off}(A) = 0$ does not imply $A = 0$. Nevertheless, the off-norm is a matrix norm on the vector space of matrices with zero diagonal,

$$\{A \in \mathbb{R}^{n\times n} \,|\, \mathrm{diag}(A) = 0\}.$$

If $A$ is symmetric, it is sufficient to work only with the upper-diagonal part. We define

$$S(A) = \frac{\sqrt{2}}{2}\mathrm{off}(A) = \sqrt{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} {a_{ij}}^2}, \quad A = A^T, \tag{1.4}$$

which is again a matrix norm on the vector space of symmetric matrices with zero diagonal

$$\{A \in \mathbb{R}^{n\times n} \,|\, A = A^T, \mathrm{diag}(A) = 0\}.$$

Let us fix the iteration step $k$. Then we simplify the notation by setting $(p_k, q_k) = (p, q)$ and $\varphi_k = \varphi$. We first show how to calculate $\varphi$, or rather its sine and cosine, that maximally reduces the off-norm in one step of the Jacobi algorithm. Let $A'$ be the transformed symmetric matrix $A$ after one step of the Jacobi method, that is,

$$A' = R(p,q,\varphi)^T A R(p,q,\varphi). \tag{1.5}$$

Transformation (1.5) only changes the $p$th and $q$th row and column of $A$. For $r \neq p, q$ we have,

$$\begin{aligned} a'_{pr} &= a_{pr}\cos\varphi + a_{qr}\sin\varphi, a'_{qr} = -a_{pr}\sin\varphi + a_{qr}\cos\varphi, \\ a'_{rp} &= a_{rp}\cos\varphi + a_{rq}\sin\varphi, a'_{rq} = -a_{rp}\sin\varphi + a_{rq}\cos\varphi. \end{aligned}$$

We can easily see that the changes of these elements do not affect the off-norm because

$$\begin{aligned} (a'_{pr})^2 + (a'_{qr})^2 &= a_{pr}^2 + a_{qr}^2, \\ (a'_{rp})^2 + (a'_{rq})^2 &= a_{rp}^2 + a_{rq}^2. \end{aligned}$$

On the other hand, the $2\times 2$ submatrix of $A$ which is at the intersection of the $p$th and $q$th row and column is transformed as follows,

$$\begin{bmatrix} a'_{pp} & a'_{pq} \\ a'_{pq} & a'_{qq} \end{bmatrix} = \begin{bmatrix} \cos\varphi & \sin\varphi \\ -\sin\varphi & \cos\varphi \end{bmatrix} \begin{bmatrix} a_{pp} & a_{pq} \\ a_{pq} & a_{qq} \end{bmatrix} \begin{bmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{bmatrix}. \tag{1.6}$$

Thus, the optimal choice for the angle $\varphi$ is the one that annihilates $a_{qp}$ and $a_{pq}$, meaning $a'_{qp} = a'_{pq} = 0$. Then, $\text{off}^2(A') = \text{off}^2(A) - 2a_{pq}^2$. The transformation (1.6) then implies

$$a'_{pq} = a_{pq}\cos(2\varphi) - \frac{1}{2}(a_{pp} - a_{qq})\sin(2\varphi) = 0,$$

and

$$\text{tg}(2\varphi) = \frac{2a_{pq}}{a_{pp} - a_{qq}}. \tag{1.7}$$

Let

$$t = \text{tg}\,\varphi, \quad \lambda = 2a_{pq}\,\text{sign}(a_{pp} - a_{qq}), \quad \mu = |a_{pp} - a_{qq}|,$$

and use the double angle formula for tangent,

$$\text{tg}(2\varphi) = \frac{2t}{1-t^2},$$

in (1.7) to get

$$\frac{2t}{1-t^2} = \frac{\lambda}{\mu}.$$

This is a quadratic equation in variable $t$,

$$\lambda t^2 + 2\mu t - \lambda = t,$$

whose solutions are

$$t_{1,2} = \frac{-\mu \pm \sqrt{\mu^2 + \lambda^2}}{\lambda}.$$

We choose the rotation angle from the interval $\left[-\frac{\pi}{4}, \frac{\pi}{4}\right]$, and hence, $t$ and $\text{tg}(2\varphi)$ are of the same sign. Also, because $\mu$ is non-negative, $\text{tg}(2\varphi)$ and $\mu$ are of the same sign. Therefore, $t$ and $\mu$ are of the same sign, and we choose

$$t = \frac{-\mu + \sqrt{\mu^2 + \lambda^2}}{\lambda}.$$

It is necessary to multiply both numerator and denominator of $t$ by $\mu + \sqrt{\mu^2 + \lambda^2}$ to avoid the catastrophic cancellation and ensure the computational stability,

$$t = \frac{\lambda}{\mu + \sqrt{\mu^2 + \lambda^2}},$$

that is

$$t = \frac{2a_{pq}\,\text{sign}(a_{pp} - a_{qq})}{|a_{pp} - a_{qq}| + \sqrt{|a_{pp} - a_{qq}|^2 + 4a_{pq}^2}}. \tag{1.8}$$

The sine and cosine are then easily calculated,

$$\cos\varphi = \frac{1}{\sqrt{1+t^2}}, \quad \sin\varphi = \frac{t}{\sqrt{1+t^2}} = t\cos\varphi. \tag{1.9}$$

It remains to compute the diagonal elements $a'_{pp}$ and $a'_{qq}$. We know that orthogonal transformation does not change the trace, i.e.,

$$a'_{pp} + a'_{qq} = a_{pp} + a_{qq}.$$

We can easily compute $a'_{pp}$ with

$$a'_{pp} = \begin{bmatrix}\cos\varphi & \sin\varphi\end{bmatrix} \begin{bmatrix}a_{pp} & a_{pq}\\ a_{pq} & a_{qq}\end{bmatrix} \begin{bmatrix}\cos\varphi\\ \sin\varphi\end{bmatrix} = a_{pp} + a_{pq}\operatorname{tg}\varphi.$$

Using the trace indifference above, the other diagonal element is equal to

$$a'_{qq} = a_{qq} - a_{pq}\operatorname{tg}\varphi.$$

We summarize one step of the real Jacobi method in the Algorithm 1.

### 1.1.2. Complex Jacobi algorithm

For a Hermitian matrix $A \in \mathbb{C}^{n\times n}$, the complex Jacobi method is similar to the iterative process (1.1), but the transformation matrices are now complex plane rotations. It takes the form

$$A^{(k+1)} = U_k^* A^{(k)} U_k, \quad k \geq 0, \quad A^{(0)} = A,$$

where $U_k = R(p_k, q_k, \varphi_k, \alpha_k)$ are

$$R(p_k,q_k,\varphi_k,\alpha_k) = \begin{bmatrix}
1 & & & & & & & & & \\
& \ddots & & & & & & & & \\
& & 1 & & & & & & & \\
& & & \cos\varphi_k & & & & -e^{\imath\alpha_k}\sin\varphi_k & & \\
& & & & 1 & & & & & \\
& & & & & \ddots & & & & \\
& & & & & & 1 & & & \\
& & & e^{-\imath\alpha_k}\sin\varphi_k & & & & \cos\varphi_k & & \\
& & & & & & & & 1 & \\
& & & & & & & & & \ddots \\
& & & & & & & & & & 1
\end{bmatrix} \begin{matrix} \\ \\ \\ p_k \\ \\ \\ \\ q_k \\ \\ \\ \\ \end{matrix} . \tag{1.10}$$

**Algorithm 1** One step of the real Jacobi algorithm. $A \in \mathbb{R}^{n\times n}$ symmetric, pivot pair $(p,q)$.

**if** $a_{pq} \neq 0$ **then**

$\lambda = 2a_{pq}\,\mathrm{sign}(a_{pp} - a_{qq})$

$\mu = |a_{pp} - a_{qq}|$

$\nu = \sqrt{\lambda^2 + \mu^2}$

$t = \frac{\lambda}{\mu+\nu}$

$c = \frac{1}{\sqrt{1+t^2}}$

$s = tc$

$a_{pp} = a_{pp} + ta_{pq}$; $a_{qq} = a_{qq} - ta_{pq}$; $a_{pq} = 0$

**for** $r = 1,\ldots,p-1$ **do**

$x = ca_{rp} + sa_{rq}$

$a_{rq} = -sa_{rp} + ca_{rq}$; $a_{rp} = x$

**end for**

**for** $r = p+1,\ldots,q-1$ **do**

$x = ca_{pr} + sa_{rq}$

$a_{rq} = -sa_{pr} + ca_{rq}$; $a_{pr} = x$

**end for**

**for** $r = q+1,\ldots,n$ **do**

$x = ca_{pr} + sa_{qr}$

$a_{qr} = -sa_{pr} + ca_{qr}$; $a_{pr} = x$

**end for**

**end if**

Here, $\imath$ stands for the imaginary unit. Unitary matrices $U_k$ are again elementary matrices, that is, they differ from the identity only in a $2 \times 2$ submatrix. They are determined by a pivot pair $(p_k, q_k)$ and angles $\varphi_k$ and $\alpha_k$. Analogous to the real case, angle $\varphi_k$ is chosen to eliminate the elements $a^{(k)}_{p_k q_k}$ and $a^{(k)}_{q_k p_k}$ of $A^{(k)}$, while $\alpha_k$ is selected to minimize the off-norm of $A^{(k)}$. In the complex case, the definition of the off-norm should include the absolute values. We have

$$\operatorname{off}^2(A) = \sum_{\substack{i,j=1 \\ i \neq j}}^{n} |a_{ij}|^2.$$

We repeat the process of finding the rotation parameters in one step of the Jacobi algorithm. Let us fix the iteration step $k$. The off-norm again depends only on the $2 \times 2$ subproblem at the pivot position $(p, q)$,

$$\begin{bmatrix} a'_{pp} & a'_{pq} \\ a'_{pq} & a'_{qq} \end{bmatrix} = \begin{bmatrix} \cos\varphi & -e^{\imath\alpha}\sin\varphi \\ e^{-\imath\alpha}\sin\varphi & \cos\varphi \end{bmatrix}^* \begin{bmatrix} a_{pp} & a_{pq} \\ a_{pq} & a_{qq} \end{bmatrix} \begin{bmatrix} \cos\varphi & -e^{\imath\alpha}\sin\varphi \\ e^{-\imath\alpha}\sin\varphi & \cos\varphi \end{bmatrix}. \quad (1.11)$$

The optimal angle $\varphi$ is the one for which $a'_{qp} = a'_{pq} = 0$. After multiplying (1.11) with the rotation matrix from the left, we get

$$\begin{bmatrix} \cos\varphi & -e^{\imath\alpha}\sin\varphi \\ e^{-\imath\alpha}\sin\varphi & \cos\varphi \end{bmatrix} \begin{bmatrix} a'_{pp} & 0 \\ 0 & a'_{qq} \end{bmatrix} = \begin{bmatrix} a_{pp} & a_{pq} \\ a_{pq} & a_{qq} \end{bmatrix} \begin{bmatrix} \cos\varphi & -e^{\imath\alpha}\sin\varphi \\ e^{-\imath\alpha}\sin\varphi & \cos\varphi \end{bmatrix},$$

which is equal to

$$\begin{bmatrix} a'_{pp}\cos\varphi & -a'_{qq}e^{\imath\alpha}\sin\varphi \\ a'_{pp}e^{-\imath\alpha}\sin\varphi & a'_{qq}\cos\varphi \end{bmatrix} = \begin{bmatrix} a_{pp}\cos\varphi + a_{pq}e^{-\imath\alpha}\sin\varphi & -a_{pp}e^{\imath\alpha}\sin\varphi + a_{pq}\cos\varphi \\ a_{qp}\cos\varphi + a_{qq}e^{-\imath\alpha}\sin\varphi & -a_{qp}e^{\imath\alpha}\sin\varphi + a_{qq}\cos\varphi \end{bmatrix}.$$

In the upper equation, elements at positions $(1,1)$ and $(2,2)$, respectively, are equal which implies

$$a'_{pp}\cos\varphi = a_{pp}\cos\varphi + a_{pq}e^{-\imath\alpha}\sin\varphi,$$
$$a'_{qq}\cos\varphi = -a_{qp}e^{\imath\alpha}\sin\varphi + a_{qq}\cos\varphi.$$

By dividing both relations with $\cos\varphi$ we get the elements $a'_{pp}$ and $a'_{qq}$,

$$a'_{pp} = a_{pp} + a_{pq}e^{-\imath\alpha}\operatorname{tg}\varphi,$$
$$a'_{qq} = a_{qq} - a_{qp}e^{\imath\alpha}\operatorname{tg}\varphi.$$

Unitary transformations do not change the trace of the matrix so we have

$$a_{pp} + a_{qq} = a'_{pp} + a'_{qq} = a_{pp} + a_{pq}e^{-\imath\alpha}\operatorname{tg}\varphi + a_{qq} - a_{qp}e^{\imath\alpha}\operatorname{tg}\varphi,$$

which leads to the equality

$$a_{pq}e^{-\imath\alpha} = a_{qp}e^{\imath\alpha}.$$

Since $a_{pq}$ is the pivot element, we assume that $a_{pq} \neq 0$. Then, from the equality above we have

$$|a_{pq}|e^{\imath\alpha_{pq}}e^{-\imath\alpha} = |a_{pq}|e^{-\imath\alpha_{pq}}e^{\imath\alpha},$$

where we used the fact that $a_{pq} = |a_{pq}|e^{\imath\alpha_{pq}}$ and $a_{qp} = |a_{pq}|e^{-\imath\alpha_{pq}}$. Now we divide the equation by the right-hand side and get

$$e^{2\imath\alpha_{pq}-2\imath\alpha} = 1,$$

or equivalently,

$$2\imath\alpha_{pq} - 2\imath\alpha = 0.$$

Thus, $\alpha = \alpha_{pq}$. The optimal choice for $\alpha$ is then

$$\alpha = \arg(a_{pq}). \tag{1.12}$$

From the relation (1.11) we compute the expression for $a'_{pq}$,

$$\begin{aligned}
a'_{pq} &= \begin{bmatrix}\cos\varphi & e^{\imath\alpha}\sin\varphi\end{bmatrix}\begin{bmatrix}a_{pp} & a_{pq}\\ a_{pq} & a_{qq}\end{bmatrix}\begin{bmatrix}-e^{\imath\alpha}\sin\varphi\\ \cos\varphi\end{bmatrix}\\
&= -a_{pp}e^{\imath\alpha}\sin\varphi\cos\varphi - a_{qp}e^{2\imath\alpha}\sin^2\varphi + a_{pq}\cos^2\varphi + a_{qq}e^{\imath\alpha}\sin\varphi\cos\varphi\\
&= \left(-(a_{pp}-a_{qq})\cos\varphi\sin\varphi + a_{pq}e^{-\imath\alpha}(\cos^2\varphi - \sin^2\varphi)\right)e^{\imath\alpha}\\
&= \left(-\frac{1}{2}(a_{pp}-a_{qq})\sin 2\varphi + a_{pq}e^{-\imath\alpha}\cos 2\varphi\right)e^{\imath\alpha}.
\end{aligned}$$

The condition $a'_{pq} = 0$ indicates that

$$\frac{1}{2}(a_{pp}-a_{qq})\sin 2\varphi = a_{pq}e^{-\imath\alpha}\cos 2\varphi,$$

so the choice (1.12) implies

$$\operatorname{tg}(2\varphi) = \frac{2a_{pq}e^{-\imath\alpha}}{a_{pp}-a_{qq}} = \frac{2|a_{pq}|}{a_{pp}-a_{qq}}. \tag{1.13}$$

To find $\cos\varphi$ and $\sin\varphi$ in a stable way, we do as in the real case. Similarly as in (1.8), we get the tangent,

$$\operatorname{tg}\varphi = \frac{2|a_{pq}|\operatorname{sign}(a_{pp}-a_{qq})}{|a_{pp}-a_{qq}| + \sqrt{|a_{pp}-a_{qq}|^2 + 4|a_{pq}|^2}}. \tag{1.14}$$

Sine and cosine are then easily computed, equivalently to (1.9).

We summarize one step of the complex Jacobi method in the Algorithm 2. The complex conjugate of number $a$ is denoted by $a^*$.

**Algorithm 2** One step of the complex Jacobi algorithm. $A \in \mathbb{C}^{n\times n}$ Hermitian, pivot pair $(p,q)$.

**if** $a_{pq} \neq 0$ **then**

$\lambda = 2|a_{pq}|\,\mathrm{sign}(a_{pp} - a_{qq})$

$\mu = |a_{pp} - a_{qq}|$

$\nu = \sqrt{\lambda^2 + \mu^2}$

$t = \frac{\lambda}{\mu+\nu}$

$c = \frac{1}{\sqrt{1+t^2}}$

$s = tc$

$a_{pp} = a_{pp} + t|a_{pq}|;\ a_{qq} = a_{qq} - t|a_{pq}|;\ a_{pq} = 0;\ a_{qp} = 0$

$s^+ = e^{\imath\alpha} s$

$s^- = e^{-\imath\alpha} s$

**for** $r = 1,\ldots,p-1$ **do**

$x = ca_{rp} + s^- a_{rq}$

$a_{rq} = -s^+ a_{rp} + ca_{rq};\ a_{rp} = x$

**end for**

**for** $r = p+1,\ldots,q-1$ **do**

$x = ca_{pr} + s^+ a_{rq}^*$

$a_{rq} = -s^+ a_{pr}^* + ca_{rq};\ a_{pr} = x$

**end for**

**for** $r = q+1,\ldots,n$ **do**

$x = ca_{pr} + s^+ a_{qr}$

$a_{qr} = -s^- a_{pr} + ca_{qr};\ a_{pr} = x$

**end for**

**end if**

# 1.2. PIVOT STRATEGIES

In each iteration $k$ of the Algorithms 1 and 2, pivot position is selected according to a pivot strategy. In this section, we describe the well-known pivot strategies. Moreover, we define a large class of pivot strategies with which we will work in Section 3. These strategies were introduced in [7] and were studied later in [39] and [40].

For an $n \times n$ matrix, the set of all possible pivot pairs is denoted by $\mathscr{P}_n = \{(i,j) : 1 \leq i < j \leq n\}$. Notice that a pivot pair is chosen from the upper triangle of the matrix. The pivot strategy determines the order of the pivot pairs in the algorithm.

**Definition 1.2.1.** A *pivot strategy* is any function

$$I \colon \mathbb{N}_0 \to \mathscr{P}_n,$$

where $\mathbb{N}_0 = \{0,1,2,\ldots\}$.

In the *classical Jacobi strategy*, in each step $k$ of the algorithm, the pivot pair $(p,q) = (p_k,q_k)$ is chosen such that it contains the off-diagonal element with the largest absolute value,

$$|a_{pq}^{(k)}| = \max_{(i,j)\in\mathscr{P}_n} |a_{ij}^{(k)}|.$$

The search for a pivot pair requires going through the whole upper triangle of the matrix. This strategy is not very popular in practice because it slows down the process for large matrices. To overcome this problem it is better to know the pivot strategy in advance, not to just establish it on the go. To this end, from now on we work with the periodic pivot strategies. In such strategies $I$ is a periodic function. The pivot pairs are taken in some prescribed order which is repeated again and again until convergence. If $I$ has a period $T = N \equiv \frac{n(n-1)}{2}$ and if its image is equal to $\mathscr{P}_n$, then $I$ is called *a cyclic pivot strategy*. That is, cyclic pivot strategy goes through every pivot pair in some specific order, on repeat.

Let $\mathscr{O}(\mathscr{S})$ stand for the set of all finite sequences of elements from $\mathscr{S} \subseteq \mathscr{P}_n$, provided that each pair from $\mathscr{S}$ appears at least once in every sequence. Elements from $\mathscr{O}(\mathscr{S})$ are called *orderings* of $\mathscr{S}$,

$$\mathscr{O} = (p_0,q_0),(p_1,q_1),\ldots,(p_r,q_r) \in \mathscr{O}(\mathscr{S}).$$

The number of pairs $r$ contained in the ordering $\mathscr{O}$ is called its *length*. Pivot strategies and orderings are connected in the following way. A periodic strategy $I$ with period $T$ defines a sequence $\mathscr{O}_I$ which is an ordering of $\mathscr{S}$,

$$\mathscr{O}_I = I(0), I(1), \ldots, I(T-1) \in \mathscr{O}(\mathscr{S}).$$

Vice versa, if $\mathscr{O} = (p_0, q_0), (p_1, q_1), \ldots, (p_{T-1}, q_{T-1}) \in \mathscr{O}(\mathscr{S})$ is an ordering of pivot pairs from $\mathscr{S}$, then the corresponding pivot strategy $I_{\mathscr{O}}$ is defined by

$$I_{\mathscr{O}}(k) = (p_{\tau(k)}, q_{\tau(k)}),$$

where $0 \leq \tau(k) \leq T-1$ is determined by $k \equiv \tau(k) \pmod{T}$, $k \geq 0$. That is, strategy $I_{\mathscr{O}}$ takes the pivot pairs ordered as in $\mathscr{O}$, and then again and again.

Any transposition of two adjacent pivot pairs in $\mathscr{O} \in \mathscr{O}(\mathscr{S})$,

$$(p_r, q_r), (p_{r+1}, q_{r+1}) \to (p_{r+1}, q_{r+1}), (p_r, q_r),$$

assuming that the sets $\{p_r, q_r\}$ and $\{p_{r+1}, q_{r+1}\}$ are disjoint, is called an *admissible transposition* in $\mathscr{O}$. In that case, the rotation matrices $U_r$, $U_{r+1}$ from the Jacobi method commute because of their special structure. We, sometimes, also say that the pivot positions $(p_r, q_r)$ and $(p_{r+1}, q_{r+1})$ are commuting. We use several equivalence relations on $\mathscr{O}(\mathscr{S})$, $\mathscr{S} \subseteq \mathscr{P}_n$. (See, e.g., [40].)

**Definition 1.2.2.** Two sequences $\mathscr{O}, \mathscr{O}' \in \mathscr{O}(\mathscr{S})$, $\mathscr{S} \subseteq \mathscr{P}_n$, where the sequence $\mathscr{O}$ is given as $\mathscr{O} = (p_0, q_0)$, $(p_1, q_1), \ldots,$ $(p_r, q_r)$, are said to be

(i) *equivalent* if one can be obtained from the other by a finite set of admissible transpositions. We write $\mathscr{O} \sim \mathscr{O}'$.

(ii) *shift-equivalent* if $\mathscr{O} = [\mathscr{O}_1, \mathscr{O}_2]$ and $\mathscr{O}' = [\mathscr{O}_2, \mathscr{O}_1]$, where $[\ ,\ ]$ denotes concatenation; the length of $\mathscr{O}_1$ is called the shift length. We write $\mathscr{O} \overset{s}{\sim} \mathscr{O}'$.

(iii) *weak equivalent* if there exist $\mathscr{O}_i \in \mathscr{O}(\mathscr{S})$, $0 \leq i \leq t$, such that every two adjacent terms in the sequence $\mathscr{O} = \mathscr{O}_0, \mathscr{O}_1, \ldots, \mathscr{O}_t = \mathscr{O}'$ are equivalent or shift-equivalent. We write $\mathscr{O} \overset{w}{\sim} \mathscr{O}'$.

(iv) *reverse* if

$$\mathscr{O}' = (p_r, q_r), \ldots, (p_1, q_1), (p_0, q_0) \in \mathscr{O}(\mathscr{P}_n).$$

We write $\mathscr{O}' = \mathscr{O}^{\leftarrow}$.

(v) *permutation-equivalent* if there is a permutation q of the set $\mathscr{S}$ such that

$$\mathscr{O}' = (\mathrm{q}(p_0), \mathrm{q}(q_0)), (\mathrm{q}(p_1), \mathrm{q}(q_1)), \ldots, (\mathrm{q}(p_r), \mathrm{q}(q_r)).$$

We write $\mathscr{O} \overset{p}{\sim} \mathscr{O}'$ or $\mathscr{O}' = \mathscr{O}(\mathrm{q})$.

The Definition 1.2.2 (iii) implies that if $\mathscr{O} \overset{w}{\sim} \mathscr{O}'$, then there is a finite sequence $\mathscr{O} = \mathscr{O}_0$, $\mathscr{O}_1, \ldots, \mathscr{O}_t = \mathscr{O}'$ such that

$$\mathscr{O} \sim \mathscr{O}_1 \overset{s}{\sim} \mathscr{O}_2 \sim \mathscr{O}_3 \overset{s}{\sim} \mathscr{O}_4 \ldots \mathscr{O}' \quad \text{or} \quad \mathscr{O} \overset{s}{\sim} \mathscr{O}_1 \sim \mathscr{O}_2 \overset{s}{\sim} \mathscr{O}_3 \sim \mathscr{O}_4 \ldots \mathscr{O}'. \tag{1.15}$$

If there are two or more consecutive equivalence or shift-equivalence relations, they can be replaced by one such relation because of the transitive property of these equivalence relations. The chains from (1.15) that are connecting $\mathscr{O}$ and $\mathscr{O}'$ are in the *canonical form*. If the orderings $\mathscr{O}$ and $\mathscr{O}'$ are equivalent (shift-equivalent, weak equivalent, permutation-equivalent, reverse) then the same is said for the corresponding pivot strategies $I_{\mathscr{O}}$ and $I_{\mathscr{O}'}$.

Let us review some of the well-known and most frequently used cyclic strategies. The most intuitive cyclic strategies are the row-cyclic $I_{row} = I_{\mathscr{O}_{row}}$ and the column-cyclic strategy $I_{col} = I_{\mathscr{O}_{col}}$, collectively named *serial pivot strategies*. They correspond to the row-wise, and respectively, column-wise orderings,

$$\mathscr{O}_{row} = (1,2), (1,3), \ldots, (1,n), (2,3), (2,4), \ldots, (2,n), \ldots, (n-1,n),$$
$$\mathscr{O}_{col} = (1,2), (1,3), (2,3), (1,4), (2,4), (3,4), \ldots, (1,n), \ldots, (n-1,n).$$

To put it in words, row-wise orderings take upper-diagonal elements starting from the first row from left to right, then elements from the second row, then third etc., until the second to last row (with only one upper-diagonal element). Analogously, column-wise orderings take upper-diagonal elements starting from the second column (with only one upper-diagonal element), then elements from the third column from top to bottom, then fourth column etc., until the last column.

Another example of a cyclic strategy is the one derived from *antidiagonal ordering* $\mathscr{O}_{adiag}$. Here, a pivot pair $(p,q)$, $1 \le p < q \le n$ is followed by

$$\begin{aligned}
&(p+1,q-1) && \text{if } q-p>2,\\
&(1,p+q) && \text{if } q-p\le 2,\ p+q\le n,\\
&(p+q+1-n,n) && \text{if } q-p\le 2,\ n<p+q\le 2n-1,\\
&(1,2) && \text{if } q=n \text{ and } p=n-1.
\end{aligned}$$

We can describe the ordering $\mathscr{O}$ using a matrix $M_{\mathscr{O}} \in \mathbb{N}^{n\times n}$, $M_{\mathscr{O}} = (m_{pq})$, where $m_{ss} = *$ and

$$m_{pq} = m_{qp} = k, \quad \text{if } (p,q) = (p_k,q_k), \quad p \ne q.$$

If $I = I_{\mathscr{O}}$ then we write $M_I = M_{\mathscr{O}}$. For example, the matrix portrayals of orderings $\mathscr{O}_{row}$, $\mathscr{O}_{col}$ and the antidiagonal ordering $\mathscr{O}_{adiag}$ for $n = 5$ are

$$M_{row} = \begin{bmatrix} * & 0 & 1 & 2 & 3\\ 0 & * & 4 & 5 & 6\\ 1 & 4 & * & 7 & 8\\ 2 & 5 & 7 & * & 9\\ 3 & 6 & 8 & 9 & * \end{bmatrix}, \quad M_{col} = \begin{bmatrix} * & 0 & 1 & 3 & 6\\ 0 & * & 2 & 4 & 7\\ 1 & 2 & * & 5 & 8\\ 3 & 4 & 5 & * & 9\\ 6 & 7 & 8 & 9 & * \end{bmatrix}, \quad M_{adiag} = \begin{bmatrix} * & 0 & 1 & 2 & 4\\ 0 & * & 3 & 5 & 6\\ 1 & 3 & * & 7 & 8\\ 2 & 5 & 7 & * & 9\\ 4 & 6 & 8 & 9 & * \end{bmatrix},$$

where the matrix elements, starting from zero, mark the order of pivot pairs in $\mathscr{O}$. Hansen proved the equivalence of the row-wise and column-wise strategies [34]. More than that, he proved that after each cycle, two equivalent strategies $I_{\mathscr{O}}$ and $I_{\mathscr{O}'}$ generate the same matrix.

### 1.2.1. Wavefront and weakly wavefront strategies

In 1989, Shroff and Schreiber [69] defined *wavefront* strategies. For a pivot element in a wavefront strategy, the element directly above it and the element immediately to the left of it are rotated before it. Likewise, the element directly under it and immediately right to it are rotated after it. All three of the orderings $\mathscr{O}_{row}$, $\mathscr{O}_{col}$ and $\mathscr{O}_{adiag}$ are wavefront orderings.

**Definition 1.2.3.** Let $\mathscr{O} \in \mathscr{O}(\mathscr{P}_n)$ be a pivot sequence of length $N = \frac{n(n-1)}{2}$, and let $t(p,q)$ denote the place at which the pair $(p,q)$ appears in $\mathscr{O}$. Then $\mathscr{O}$ and $I_{\mathscr{O}}$ are a wavefront ordering and a wavefront strategy, respectively, if

$$t(p,q-1) < t(p,q) < t(p+1,q), \quad 1 \le p < q \le n.$$

The next easily proven lemma claims even more. For a pivot element in a wavefront strategy, all elements above and to the left of it are rotated before it. Likewise, all elements under and to the right of it are rotated after it.

**Lemma 1.2.4** (Shroff, Schreiber [69])**.** In a cyclic wavefront ordering, for all $1 \leq p < q \leq n$, and $1 \leq i < j \leq n$,

(i) $t(p,q) \geq t(i,j)$ if $i \leq p$ and $j \leq q$,

(ii) $t(p,q) \leq t(i,j)$ if $i \geq p$ and $j \geq q$.

The matrix below illustrates this result. During one cycle, the elements rotated before the pivot element $x_{pq}$ are denoted by $\triangleleft$, while elements rotated after it are denoted by $\triangleright$.

$$\begin{bmatrix} * & \triangleleft & \triangleleft & \triangleleft & \triangleleft & x & x \\ & * & \triangleleft & \triangleleft & x_{pq} & \triangleright & \triangleright \\ & & * & x & \triangleright & \triangleright & \triangleright \\ & & & * & \triangleright & \triangleright & \triangleright \\ & & & & * & \triangleright & \triangleright \\ & & & & & * & \triangleright \\ & & & & & & * \end{bmatrix}.$$

Shroff and Schreiber proved that wavefront strategies are exactly those equivalent to the serial pivot strategies. Next, we can define the following class of strategies.

**Definition 1.2.5.** A cyclic strategy is called *weak wavefront* strategy if it is weakly equivalent to a wavefront strategy.

As mentioned above, the order of executing the commuting rotations does not influence the transformation that represents one sweep/cycle of the method. This suggests that the Jacobi method is suitable for parallelization. Luk and Park [57, 58] proposed several parallel strategies. For example, the antidiagonal ordering is easily transformed into a parallel ordering by performing the rotations on the same antidiagonal at the same time,

$$\begin{bmatrix} * & 0 & 1 & 2 & 4 \\ 0 & * & 3 & 5 & 6 \\ 1 & 3 & * & 7 & 8 \\ 2 & 5 & 7 & * & 9 \\ 4 & 6 & 8 & 9 & * \end{bmatrix} \rightarrow \begin{bmatrix} * & 0 & 1 & 2 & 3 \\ 0 & * & 2 & 3 & 4 \\ 1 & 2 & * & 4 & 5 \\ 2 & 3 & 4 & * & 6 \\ 3 & 4 & 5 & 6 & * \end{bmatrix}.$$

This can be done because the pivot positions on the same antidiagonal are commuting.

To achieve more parallelism, we demonstrate another idea of Luk and Park for $n = 5$. Start with applying a shift equivalence of length 3 to the antidiagonals ordering. Do the admissible transposition of the pivot pairs $(1,2)$ and $(4,5)$. The resulting ordering is weakly equivalent to the starting antidiagonal one. Observe that the rotations on the antidiagonals commute. In addition to that, notice that the pairs $(1,2)$, $(3,5)$ and $(1,3)$, $(4,5)$ commute. If we perform the commuting rotations at the same time, we get a parallel ordering with two pivot pairs at every step. The evolution of the ordering is presented below.

$$\begin{bmatrix} * & 0 & 1 & 2 & 4 \\ 0 & * & 3 & 5 & 6 \\ 1 & 3 & * & 7 & 8 \\ 2 & 5 & 7 & * & 9 \\ 4 & 6 & 8 & 9 & * \end{bmatrix} \rightarrow \begin{bmatrix} * & \mathbf{3} & 4 & 5 & 7 \\ \mathbf{3} & * & 6 & 8 & 9 \\ 4 & 6 & * & 0 & 1 \\ 5 & 8 & 0 & * & \mathbf{2} \\ 7 & 9 & 1 & \mathbf{2} & * \end{bmatrix} \rightarrow \begin{bmatrix} * & 2 & 4 & 5 & 7 \\ 2 & * & 6 & 8 & 9 \\ 4 & 6 & * & 0 & 1 \\ 5 & 8 & 0 & * & 3 \\ 7 & 9 & 1 & 3 & * \end{bmatrix} \rightarrow \begin{bmatrix} * & 0 & 1 & 2 & 3 \\ 0 & * & 2 & 3 & 4 \\ 1 & 2 & * & 4 & 0 \\ 2 & 3 & 4 & * & 1 \\ 3 & 4 & 0 & 1 & * \end{bmatrix}.$$

The final ordering is called *parallel modulus ordering* [58]. Formally, the parallel modulus ordering follows the rule

$$t(p,q) \equiv (p+q-3)(\bmod n).$$

In general, at every step there are $\lfloor \frac{n}{2} \rfloor$ pivot pairs that are rotated in parallel. The modulus ordering is weakly equivalent to the antidiagonal one. Therefore, it is a weakly wavefront ordering.

### 1.2.2. Generalized serial strategies with permutations

Let us go back to the definition of equivalences on orderings and use it to define a broader class of pivot strategies. In terms of the Definition 1.2.2 (v), if two pivot orderings $\mathscr{O}, \mathscr{O}'$ are permutation-equivalent, then

$$M_{\mathscr{O}'} = P^T M_{\mathscr{O}} P,$$

where $P$ is permutation matrix defined with $Pe_i = e_{\mathrm{q}(i)}$, $1 \leq i \leq n$, and $M_{\mathscr{O}}$, $M_{\mathscr{O}'}$ are matrices describing the corresponding orderings. It is not difficult to show (see [7]) that one cycle of Jacobi method on matrix $P^T AP$ under the strategy $I_{\mathscr{O}}$ corresponds to, up to the sign of rotation angles, one cycle of Jacobi method on matrix $A$ under the strategy $\mathscr{O}'$.

**Proposition 1.2.6.** (Begović Kovač [7]) Let $A^{(N)}$ be the matrix obtained from $A$ after one cycle of Jacobi method using the pivot ordering $\mathscr{O}$. Let $P$ be a permutation matrix of order $n$, and q is such that $Pe_i = e_{\mathrm{q}(i)}$. If we apply the Jacobi method on $P^TAP$ under the strategy given by ordering $O' = (\mathrm{q}(p_0),\mathrm{q}(q_0)),(\mathrm{q}(p_1),\mathrm{q}(q_1)),\ldots,(\mathrm{q}(p_{N-1}),\mathrm{q}(q_{N-1}))$, after one cycle, we get $P^TA^{(N)}P$.

Now we define a very broad class of pivot strategies that is derived from the serial pivot strategies. We call them *generalized serial pivot strategies* [39].

Denote by $\Pi^{(l_1,l_2)}$ the set of all permutations of the set $\{l_1,l_1+1,l_1+2,\ldots,l_2\}$ for $l_1 < l_2$. Let

$$\mathscr{C}_c^{(n)} = \Big\{\mathscr{O} \in \mathscr{O}(\mathscr{P}_n) \mid \mathscr{O} = (1,2),(\tau_3(1),3),(\tau_3(2),3),\ldots,(\tau_n(1),n),\ldots \\ \ldots,(\tau_n(n-1),n), \quad \tau_j \in \Pi^{(1,j-1)}, 3 \le j \le n\Big\}.$$

The orderings from $\mathscr{C}_c^{(n)}$ go through the matrix column by column, starting from the second one, just like in the standard column strategy $I_{col}$. However, in each column pivot elements are chosen in some arbitrary order. If $\mathscr{O} \in \mathscr{C}_c^{(n)}$, then $\mathscr{O}$ is called a column-wise ordering with permutations. An example of an ordering $\mathscr{O} \in \mathscr{C}_c(5)$ is

$$M_{\mathscr{O}} = \begin{bmatrix} * & 0 & 2 & 3 & 9 \\ 0 & * & 1 & 5 & 6 \\ 2 & 1 & * & 4 & 8 \\ 3 & 5 & 4 & * & 7 \\ 9 & 6 & 8 & 7 & * \end{bmatrix}.$$

Similarly, the set of row-wise orderings with permutations is defined as

$$\mathscr{C}_r^{(n)} = \Big\{\mathscr{O} \in \mathscr{O}(\mathscr{P}_n) \mid \mathscr{O} = (n-1,n),(n-2,\tau_{n-2}(n-1)),(n-2,\tau_{n-2}(n)),\ldots \\ \ldots,(1,\tau_1(2)),\ldots,(1,\tau_1(n)) \quad \tau_i \in \Pi^{(i+1,n)}, 1 \le i \le n-2\Big\}.$$

The orderings from $\mathscr{C}_r^{(n)}$ start from the $(n-1)$st row, that is element on the position $(n-1,n)$. Then they go through the elements of the $(n-2)$nd row in some order, then the elements of the $(n-3)$rd row in some order, etc. At the end of the cycle there are the

first row elements in an arbitrary order. For example,

$$M_{\mathscr{O}} = \begin{bmatrix} * & 6 & 9 & 7 & 8 \\ 6 & * & 5 & 3 & 4 \\ 9 & 5 & * & 1 & 2 \\ 7 & 3 & 1 & * & 0 \\ 8 & 4 & 2 & 0 & * \end{bmatrix}$$

is an ordering $\mathscr{O} \in \mathscr{C}_r^{(5)}$. It can be shown that every ordering from $\mathscr{C}_r^{(n)}$ is permutation-equivalent to some ordering from $\mathscr{C}_c^{(n)}$, where the permutation is equal to

$$\begin{pmatrix} 1 & 2 & \cdots & n \\ n & n-1 & \cdots & 1 \end{pmatrix}.$$

Using $\mathscr{C}_c^{(n)}$ and $\mathscr{C}_r^{(n)}$, we can define two more sets of orderings. They contain orderings reversed to column-wise and row-wise orderings with permutations,

$$\overleftarrow{\mathscr{C}}_c^{(n)} = \left\{ \mathscr{O} \in \mathscr{O}(\mathscr{P}_n) \mid \mathscr{O}^{\leftarrow} \in \mathscr{C}_c^{(n)} \right\} \quad \text{and} \quad \overleftarrow{\mathscr{C}}_r^{(n)} = \left\{ \mathscr{O} \in \mathscr{O}(\mathscr{P}_n) \mid \mathscr{O}^{\leftarrow} \in \mathscr{C}_r^{(n)} \right\}.$$

For instance,

$$\begin{bmatrix} * & 9 & 7 & 5 & 1 \\ 9 & * & 8 & 4 & 2 \\ 7 & 8 & * & 6 & 0 \\ 5 & 4 & 6 & * & 3 \\ 1 & 2 & 0 & 3 & * \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} * & 0 & 2 & 3 & 1 \\ 0 & * & 4 & 6 & 5 \\ 2 & 4 & * & 8 & 7 \\ 3 & 6 & 8 & * & 9 \\ 1 & 5 & 7 & 9 & * \end{bmatrix}$$

are orderings from $\overleftarrow{\mathscr{C}}_c^{(5)}$ and $\overleftarrow{\mathscr{C}}_r^{(5)}$, respectively. Together, these four sets of orderings are called *serial orderings with permutations*,

$$\mathscr{C}_{sp}^{(n)} = \mathscr{C}_c^{(n)} \cup \overleftarrow{\mathscr{C}}_c^{(n)} \cup \mathscr{C}_r^{(n)} \cup \overleftarrow{\mathscr{C}}_r^{(n)}.$$

From the set of serial orderings with permutations we get a very large set of the pivot orderings if we derive an expansion of $\mathscr{C}_{sp}^{(n)}$ using the other equivalence relations from Definition 1.2.2. Let

$$\mathscr{C}_{sg}^{(n)} = \left\{ \mathscr{O} \in \mathscr{O}(\mathscr{P}_n) \mid \mathscr{O} \overset{w}{\sim} \mathscr{O}' \overset{p}{\sim} \mathscr{O}'' \text{ or } \mathscr{O} \overset{p}{\sim} \mathscr{O}' \overset{w}{\sim} \mathscr{O}'', \mathscr{O}'' \in \mathscr{C}_{sp}^{(n)} \right\},$$

where $\mathscr{O}' \in \mathscr{O}(\mathscr{P}_n)$. Strategies defined by orderings from $\mathscr{C}_{sg}^{(n)}$ are called *generalized serial pivot strategies with permutations*. In Chapter 2 we extend the convergence results of the Eberlein method, which is a Jacobi-type method, to the set $\mathscr{C}_{sg}^{(n)}$.

# 1.3. JACOBI ANNIHILATORS AND OPERATORS

Jacobi annihilators and operators were first introduced in 1968 by Henrici and Zimmermann [43]. They offer a different, more general perspective on the Jacobi method. Specifically, an appropriate instance of the Jacobi annihilator corresponds to a single step of the Jacobi algorithm, while a Jacobi operator with specific parameters coincides with one full cycle of the Jacobi method. They are frequently used as a tool to achieve convergence results for the Jacobi method [36–40]. We are going to use them in the same way in Chapter 2. In this section we first define complex Jacobi annihilators and operators. Then we comment on the simpler real case.

## 1.3.1. Complex case

Jacobi annihilators and operators are not applied on a matrix but on vectors representing off-diagonal part of a matrix. Set

$$c_j = \begin{bmatrix} a_{1j} \\ a_{2j} \\ \vdots \\ a_{j-1,j} \end{bmatrix} \quad \text{and} \quad r_i = [a_{i1}, a_{i2}, \ldots, a_{i,i-1}], \quad \text{for } 2 \leq i, j \leq n.$$

Let vecoff : $\mathbb{C}^{n\times n} \to \mathbb{C}^{2N}$, $N = \frac{n(n-1)}{2}$, be a function defined by

$$a = \text{vecoff}(A) = \left[c_2^T, c_3^T, \ldots, c_n^T, r_2, r_3, \ldots, r_n\right]^T \in \mathbb{C}^{2N}.$$

It is easy to check that vecoff is a linear operator and a surjection. If $A$ is Hermitian, $a$ is determined by strictly upper-triangular elements of $A$. In particular,

$$\text{vecoff}(A) := \begin{bmatrix} v \\ \overline{v} \end{bmatrix},$$

where $v = \left[c_2^T, c_3^T, \ldots, c_n^T\right]^T$. For example, vecoff transforms matrix

$$\begin{bmatrix} * & i & 1+2i & 3 \\ -i & * & 2 & 4-i \\ 1-2i & 2 & * & 5 \\ 3 & 4+i & 5 & * \end{bmatrix}$$

to vector

$$\begin{bmatrix} i & 1+2i & 2 & 3 & 4-i & 5 & -i & 1-2i & 2 & 3 & 4+i & 5 \end{bmatrix}^T.$$

From the previous example it is clear that vecoff is not invertible because it ignores matrix diagonal. Therefore, we define a restriction $\text{vecoff}_0 = \text{vecoff}|_{\mathbb{C}_0^{n\times n}}$, where $\mathbb{C}_0^{n\times n}$ is the set of all $n \times n$ complex matrices with zero diagonal. For a vector $a \in \mathbb{C}^{2N}$ it stands that

$$\text{vecoff}(\text{vecoff}_0^{-1}(a)) = a.$$

Furthermore, let $\nu_{pq} : \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$ be a linear operator that sets the matrix values at positions $(p,q)$ and $(q,p)$ to zero, while the rest of the matrix is unchanged. Using the operators vecoff and $\nu$ we define the Jacobi annihilator.

**Definition 1.3.1.** Let $U = R(p,q,\varphi,\alpha)$ be a complex rotation. Matrix $\mathscr{R}_{pq}(U)$ defined by

$$\mathscr{R}_{pq}(U)\,\text{vecoff}(A) = \text{vecoff}(\nu_{pq}(U^*AU)), \quad A \in \mathbb{C}^{n\times n}, \tag{1.16}$$

is called the Jacobi annihilator. The class of Jacobi annihilators $\mathscr{R}_{pq}^{\omega}$, $\omega \in [0,1]$, is a set

$$\mathscr{R}_{pq}^{\omega} = \left\{ \mathscr{R}_{pq}(U) \;\middle|\; U = R(p,q,\varphi,\alpha),\ \varphi \geq 0,\ \alpha \leq 2\pi,\ |\cos\varphi| \geq \omega \right\}.$$

If $\omega = 0$, $\mathscr{R}_{pq}$ is used instead of $\mathscr{R}_{pq}^{0}$.

The effect of the Jacobi annihilator on a vector $a$ can be described as follows. Firstly, set $A = \text{vecoff}_0^{-1}(a)$ as a two-dimensional representation of $a$. Matrix $A$ is then transformed by the rule $A' = U^*AU$, where $U = R(p,q,\varphi,\alpha)$ is a complex rotation with an arbitrary choice of $\varphi$. Contrary to a single step of the Jacobi algorithm, as the rotation angle is arbitrary, the element of $A'$ at the pivot position $(p,q)$ does not have to be zero. Nevertheless, to keep the property of canceling the pivot element, apply $\nu_{pq}$ on the matrix $A'$. The resulting matrix $A'_0$ is then again represented by a vector $a'$ using vecoff.

**Algorithm 3** Applying the Jacobi annihilator, $a' = \mathscr{R}_{pq}(U)a$

$a \in \mathbb{C}^{2N}$ arbitrary

$A = \text{vecoff}_0^{-1}(a)$

$A' = U^*AU$

$A'_0 = \nu_{pq}(A')$

$a' = \text{vecoff}(A'_0)$

If the angles $\varphi$ and $\alpha$ are not arbitrary, but chosen such that they cancel the pivot element, then this special case of the Algorithm 3 is equivalent to one step of the Jacobi algorithm. In that case applying $v_{pq}$ on $A'$ is unnecessary.

We give a theorem that explains the structure of the Jacobi annihilator and present an example of a Jacobi annihilator for a $4 \times 4$ Hermitian matrix.

**Theorem 1.3.2** (Begović Kovač [7]). Let $\mathscr{R} = \mathscr{R}(R(p,q,\varphi,\alpha))$ be a Jacobi annihilator. Let

$$\tau(i,j) = \begin{cases} (j-1)(j-2)/2 + i, & \text{for } 1 \le i < j \le n, \\ \tau(j,i) + N, & \text{for } 1 \le j < i \le n. \end{cases}$$

be the function that indicates the position of the element $a_{ij}$ in the vector vecoff$(A)$. Then $\mathscr{R}$ differs from the identity matrix $I_{2N}$ in exactly $2n-2$ submatrices defined by

$$\mathscr{R}_{\tau(p,q)\tau(p,q)} = 0, \quad \mathscr{R}_{\tau(q,p)\tau(q,p)} = 0,$$

and

$$\begin{bmatrix} \mathscr{R}_{\tau(r,p)\tau(r,p)} & \mathscr{R}_{\tau(r,p)\tau(r,q)} \\ \mathscr{R}_{\tau(r,q)\tau(r,p)} & \mathscr{R}_{\tau(r,q)\tau(r,q)} \end{bmatrix} = \begin{bmatrix} \cos\varphi & e^{-\imath\alpha}\sin\varphi \\ -e^{i\alpha}\sin\varphi & \cos\varphi \end{bmatrix},$$
$$\begin{bmatrix} \mathscr{R}_{\tau(p,r)\tau(p,r)} & \mathscr{R}_{\tau(p,r)\tau(q,r)} \\ \mathscr{R}_{\tau(q,r)\tau(p,r)} & \mathscr{R}_{\tau(q,r)\tau(q,r)} \end{bmatrix} = \begin{bmatrix} \cos\varphi & e^{\imath\alpha}\sin\varphi \\ -e^{-i\alpha}\sin\varphi & \cos\varphi \end{bmatrix},$$

where $1 \le r \le n$, $r \notin \{p,q\}$.

Let $A$ be a $4 \times 4$ Hermitian matrix and let $U = U(2,4,\varphi_{24},\alpha_{24})$. For simplicity, denote $s_{24} = \sin\varphi_{24}$ and $c_{24} = \cos\varphi_{24}$. Then $\mathscr{R}_{24}(U)$ is a $12 \times 12$ matrix that differs from the identity $I_{12}$ in two diagonal elements and four $2 \times 2$ submatrices. Generally, the annihilator cancels the pivot elements $(p,q)$ and $(q,p)$, so the diagonal elements on positions $\tau(p,q)$ and $\tau(q,p)$ must be zero. Particularly,

$$\mathscr{R}_{\tau(2,4)\tau(2,4)} = \mathscr{R}_{55} = 0, \quad \mathscr{R}_{\tau(4,2)\tau(4,2)} = \mathscr{R}_{11,11} = 0.$$

In the Theorem 1.3.2, for $r = 1$ we get that

$$\begin{bmatrix} \mathscr{R}_{\tau(1,2)\tau(1,2)} & \mathscr{R}_{\tau(1,2)\tau(1,4)} \\ \mathscr{R}_{\tau(1,4)\tau(1,2)} & \mathscr{R}_{\tau(1,4)\tau(1,4)} \end{bmatrix} = \begin{bmatrix} \mathscr{R}_{1,1} & \mathscr{R}_{1,4} \\ \mathscr{R}_{4,1} & \mathscr{R}_{4,4} \end{bmatrix} = \begin{bmatrix} c_{24} & e^{-\imath\alpha_{24}}s_{24} \\ -e^{\imath\alpha_{24}}s_{24} & c_{24} \end{bmatrix},$$
$$\begin{bmatrix} \mathscr{R}_{\tau(2,1)\tau(2,1)} & \mathscr{R}_{\tau(2,1)\tau(4,1)} \\ \mathscr{R}_{\tau(4,1)\tau(2,1)} & \mathscr{R}_{\tau(4,1)\tau(4,1)} \end{bmatrix} = \begin{bmatrix} \mathscr{R}_{7,7} & \mathscr{R}_{7,10} \\ \mathscr{R}_{10,7} & \mathscr{R}_{10,10} \end{bmatrix} = \begin{bmatrix} c_{24} & e^{\imath\alpha_{24}}s_{24} \\ -e^{-\imath\alpha_{24}}s_{24} & c_{24} \end{bmatrix}.$$

We get the position of the other two rotation matrices for $r = 3$. In summary, the Jacobi annihilator $\mathscr{R}_{24}(U)$ takes the form

$$\begin{bmatrix}
c_{24} & & & e^{-\imath\alpha_{24}}s_{24} & & & & & & & & \\
& 1 & & & & & & & & & & \\
& & c_{24} & & & & & & & & & e^{\imath\alpha_{24}}s_{24} \\
-e^{\imath\alpha_{24}}s_{24} & & & c_{24} & & & & & & & & \\
& & & & 0 & & & & & & & \\
& & & & & c_{24} & & & e^{-\imath\alpha_{24}}s_{24} & & & \\
& & & & & & c_{24} & & & e^{\imath\alpha_{24}}s_{24} & & \\
& & & & & & & 1 & & & & \\
& & & & & -e^{\imath\alpha_{24}}s_{24} & & & c_{24} & & & \\
& & & & & & -e^{-\imath\alpha_{24}}s_{24} & & & c_{24} & & \\
& & & & & & & & & & 0 & \\
& & -e^{-\imath\alpha_{24}}s_{24} & & & & & & & & & c_{24}
\end{bmatrix}.$$

It will be useful to know that the spectral norm of a Jacobi annihilator is equal to one, except for the case of the $2 \times 2$ annihilator, which is a zero matrix. This follows from the structure of the annihilators.

In the same way the Jacobi annihilator corresponds to a single step of the Jacobi method, the Jacobi operator corresponds to one sweep of the cyclic Jacobi method. We define Jacobi operators and the class of Jacobi operators.

**Definition 1.3.3.** Let $N = \frac{n(n-1)}{2}$, $\omega \in [0,1]$, and let the ordering $\mathscr{O} \in \mathbb{O}(\mathscr{P}_n)$ be given as $\mathscr{O} = (p_0,q_0),(p_1,q_1),\ldots,(p_{N-1},q_{N-1})$. Then

$$\mathscr{J}_{\mathscr{O}}^{\omega} = \{\mathscr{J} \mid \mathscr{J} = \mathscr{R}_{p_{N-1}q_{N-1}}\mathscr{R}_{p_{N-2}q_{N-2}}\ldots\mathscr{R}_{p_0q_0},\ \mathscr{R}_{p_kq_k} \in \mathscr{R}_{p_kq_k}^{\omega},\ 0 \le k \le N-1\}$$

is the class of Jacobi operators associated with the sequence $\mathscr{O}$ and $\omega$. The $2N \times 2N$ matrices $\mathscr{J}$ from $\mathscr{J}_{\mathscr{O}}^{\omega}$ are called Jacobi operators. If $\omega = 0$, we write $\mathscr{J}_{\mathscr{O}}$ instead of $\mathscr{J}_{\mathscr{O}}^{0}$.

Furthermore, Jacobi operators determined by mutually equivalent, shift-equivalent, weak-equivalent, or permutation-equivalent pivot orderings are connected. Let us explain this. By $\rho(X)$ we denote the spectral radius of the matrix $X$, that is, the largest absolute value of the elements in the spectrum of $X$. The following lemma holds.

**Lemma 1.3.4** (Begović Kovač [7])**.** Let $\mathscr{O}$, $\mathscr{O}' \in \mathbb{O}(\mathscr{P}_n)$ be two pivot orderings and let $\mathscr{J}_{\mathscr{O}}$, $\mathscr{J}_{\mathscr{O}'}$ be the Jacobi operators determined by $\mathscr{O}$ and $\mathscr{O}'$.

1. If $\mathscr{O} \sim \mathscr{O}'$, then $\mathscr{J}_{\mathscr{O}} = \mathscr{J}_{\mathscr{O}'}$.
2. If $\mathscr{O} \overset{s}{\sim} \mathscr{O}'$, then $\rho(\mathscr{J}_{\mathscr{O}}) = \rho(\mathscr{J}_{\mathscr{O}'})$.

3. If $\mathscr{O} \overset{w}{\sim} \mathscr{O}'$, then $\rho(\mathscr{J}_{\mathscr{O}}) = \rho(\mathscr{J}_{\mathscr{O}'})$.

4. If $\mathscr{O} \overset{p}{\sim} \mathscr{O}'$, then $\|\mathscr{J}_{\mathscr{O}}\|_2 = \|\mathscr{J}_{\mathscr{O}'}\|_2$.

### 1.3.2. Real case

The definitions of the Jacobi annihilator and operator get simpler if $A$ is a real symmetric matrix. In that case, we want the Jacobi annihilator, and consequently the Jacobi operator, to be real. To achieve that, we set the parameter $\alpha$ in Definition 1.3.1 to zero. In addition, it is sufficient to observe only the upper triangle of $A$. Therefore, we redefine the function vecoff. In the real case we have vecoff : $\mathbb{R}^{n\times n} \to \mathbb{R}^N$, $N = \frac{n(n-1)}{2}$,

$$a = \text{vecoff}(A) = \begin{bmatrix} c_2^T & c_3^T & \cdots & c_n^T \end{bmatrix}^T \in \mathbb{R}^N,$$

where $c_j$, $2 \le j \le n$, are strictly upper-diagonal parts of the columns of $A$, as in the complex case. For example, the operator vecoff acts as follows,

$$\begin{bmatrix} * & 0 & 1 & 2 \\ 0 & * & 3 & 4 \\ 1 & 3 & * & 5 \\ 2 & 4 & 5 & * \end{bmatrix} \xrightarrow{\text{vecoff}} \begin{bmatrix} 0 \\ 1 \\ 3 \\ 2 \\ 4 \\ 5 \end{bmatrix}.$$

Again, vecoff is not an injection. Hence, we need its restriction. Let $S_0$ be the set of all $n \times n$ symmetric matrices with zero diagonal. We define $\text{vecoff}_0 = \text{vecoff}|_{S_0}$, which is a bijection.

The properties of the real Jacobi annihilator are inherited from the complex one. The function $\text{vecoff}_0^{-1}$ maps an arbitrary vector $a \in \mathbb{R}^N$ to a matrix $A \in S_0$. Then $A$ is transformed to $A' = U^T A U$, using the rotation matrix $U = R(p,q,\varphi)$ with an arbitrary choice of $\varphi$. This transformation does not necessarily cancel the pivot element. To achieve this feature, we use the operator $\nu_{pq} : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$ that sets the matrix elements at positions $(p,q)$ and $(q,p)$ to zero, while the rest of the matrix remains unchanged. We get $A'_0 = \nu_{pq}(A')$. Finally, the matrix obtained $A'_0$ is again transformed into a vector $a'$ using $\text{vecoff}_0$. The real Jacobi annihilator is a matrix $\mathscr{R}_{pq}(U)$ that describes the linear transformation from $a$ to $a'$,

$$\mathscr{R}_{pq}(U)\,\text{vecoff}(A) = \text{vecoff}(\nu_{pq}(U^T A U)), \quad A \in \mathbb{R}^{n\times n}.$$

The class of Jacobi annihilators $\mathscr{R}^{\omega}_{pq}$, $\omega \in [0,1]$, is a set

$$\mathscr{R}^{\omega}_{pq} = \Big\{ \mathscr{R}_{pq}(U) \;\mid\; U = R(p,q,\varphi),\ 0 \le \varphi,\ |\cos\varphi| \ge \omega \Big\}.$$

If $\omega = 0$, $\mathscr{R}_{pq}$ is used instead of $\mathscr{R}^{0}_{pq}$.

Similarly to the complex case and Theorem 1.3.2, the next theorem describes the structure of the real Jacobi annihilator.

**Theorem 1.3.5** (Henrici, Zimmermann [43])**.** Let $\mathscr{R} = \mathscr{R}(R(p,q,\varphi_{pq}))$, be a Jacobi annihilator. Let

$$\tau(i,j) = (j-1)(j-2)/2 + i$$

be the function that indicates the position of the element $a_{ij}$ in the vector vecoff$(A)$. Then $\mathscr{R}$ differs from the identity matrix $I_N$ in one diagonal element determined by

$$\mathscr{R}_{\tau(p,q)\tau(p,q)} = 0,$$

and exactly $n-2$ principal $2\times 2$ submatrices defined by

$$\begin{bmatrix} \mathscr{R}_{\tau(r,p)\tau(r,p)} & \mathscr{R}_{\tau(r,p)\tau(r,q)} \\ \mathscr{R}_{\tau(r,q)\tau(r,p)} & \mathscr{R}_{\tau(r,q)\tau(r,q)} \end{bmatrix} = \begin{bmatrix} \cos\varphi_{pq} & -\sin\varphi_{pq} \\ \sin\varphi_{pq} & \cos\varphi_{pq}, \end{bmatrix}, \quad 1 \le r < p,$$

$$\begin{bmatrix} \mathscr{R}_{\tau(p,r)\tau(p,r)} & \mathscr{R}_{\tau(p,r)\tau(r,q)} \\ \mathscr{R}_{\tau(r,q)\tau(p,r)} & \mathscr{R}_{\tau(r,q)\tau(r,q)} \end{bmatrix} = \begin{bmatrix} \cos\varphi_{pq} & -\sin\varphi_{pq} \\ \sin\varphi_{pq} & \cos\varphi_{pq}, \end{bmatrix}, \quad p < r < q,$$

$$\begin{bmatrix} \mathscr{R}_{\tau(p,r)\tau(p,r)} & \mathscr{R}_{\tau(p,r)\tau(q,r)} \\ \mathscr{R}_{\tau(q,r)\tau(p,r)} & \mathscr{R}_{\tau(q,r)\tau(q,r)} \end{bmatrix} = \begin{bmatrix} \cos\varphi_{pq} & -\sin\varphi_{pq} \\ \sin\varphi_{pq} & \cos\varphi_{pq}, \end{bmatrix}, \quad q < r \le n.$$

Using the parity of sine and cosine and the definition of annihilator, it is easy to see that

$$\mathscr{R}(R(p,q,\varphi_{pq}))^T = \mathscr{R}(R(p,q,-\varphi_{pq})).$$

Using Theorem 1.3.5, we construct an example of a Jacobi annihilator for $A \in \mathbb{R}^{5\times 5}$, $N = 10$. Let $U = R(2,3,\varphi_{23})$, and denote by $s_{23} = \sin\varphi_{23}$, $c_{23} = \cos\varphi_{23}$. Then $\mathscr{R} = \mathscr{R}_{23}(U)$ is a $10\times 10$ matrix that differs from the identity $I_{10}$ in one diagonal element and three $2\times 2$ submatrices. Generally, the annihilator cancels the pivot element $(p,q)$, so the diagonal element at position $\tau(p,q)$ must be zero. Particularly,

$$\mathscr{R}_{\tau(2,3)\tau(2,3)} = \mathscr{R}_{33} = 0.$$

In Theorem 1.3.5, for $r = 1$ we get that

$$\begin{bmatrix} \mathscr{R}_{\tau(1,2)\tau(1,2)} & \mathscr{R}_{\tau(1,2)\tau(1,3)} \\ \mathscr{R}_{\tau(1,3)\tau(1,2)} & \mathscr{R}_{\tau(1,3)\tau(1,3)} \end{bmatrix} = \begin{bmatrix} \mathscr{R}_{11} & \mathscr{R}_{12} \\ \mathscr{R}_{21} & \mathscr{R}_{22} \end{bmatrix} = \begin{bmatrix} c_{23} & -s_{23} \\ s_{23} & c_{23} \end{bmatrix}.$$

We get the position of the other two rotation matrices for $r = 4$ and $r = 5$. In summary, the Jacobi annihilator $\mathscr{R}_{23}(U)$ takes the form

$$\mathscr{R}_{23}(U) = \begin{bmatrix} c_{23} & -s_{23} & & & & & & & \\ s_{23} & c_{23} & & & & & & & \\ & & 0 & & & & & & \\ & & & 1 & & & & & \\ & & & & c_{23} & -s_{23} & & & \\ & & & & s_{23} & c_{23} & & & \\ & & & & & & 1 & & \\ & & & & & & & c_{23} & -s_{23} \\ & & & & & & & s_{23} & c_{23} \\ & & & & & & & & & 1 \end{bmatrix}.$$

Real Jacobi operators and the class of Jacobi operators is defined analogously to the complex case. Moreover, Lemma 1.3.4 holds in the real case.

# 1.4. CONVERGENCE OF THE JACOBI METHOD

In this section, we define the global convergence of the Jacobi method and list several convergence results from the literature.

**Definition 1.4.1.** Jacobi method converges for a Hermitian (symmetric) matrix $A$ if the sequence

$$A^{(0)} = A, A^{(1)}, A^{(2)}, \ldots,$$

generated by the complex (real) Jacobi method converges toward a diagonal matrix $\Lambda$, such that the diagonal elements of $\Lambda$ are eigenvalues of $A$. The Jacobi method converges globally if it converges for every starting Hermitian (symmetric) matrix $A$.

In the rest of the chapter, when addressing the convergence of the Jacobi method we mean global convergence. In 1960 Forsythe and Henrici proved the convergence of the Jacobi method under the row cyclic strategy.

**Theorem 1.4.2** (Forsythe, Henrici [31])**.** Let the rotation angle $\varphi_k$ satisfy

$$\varphi_k \in [a,b], \quad -\frac{\pi}{2} < a < b < \frac{\pi}{2},$$

and apply the sequence of the Jacobi transformations on a Hermitian matrix $A$. If off-diagonal elements are annihilated using the cyclic row-wise strategy, then the Jacobi method is globally convergent.

The Forsythe-Henrici condition is also often told using an equivalent formulation employing the cosine of the rotation angle. That is, the cosine of the rotation angle should be bounded from below by some $\omega > 0, |\cos\varphi_k| \geq \omega > 0$.

It is known that if the Jacobi method converges for some strategy $I_{\mathscr{O}}$, this implies the convergence of the method for all strategies equivalent to $I_{\mathscr{O}}$. That is to say, in 1963 Hansen [34] proved that after each cycle, two equivalent strategies $I_{\mathscr{O}}$ and $I_{\mathscr{O}'}$ generate the same matrix. This means that two equivalent strategies produce exactly the same subsequence $A^{(kN)}$, for each cycle $k \geq 0$. Therefore, the Jacobi method converges under the set of wavefront strategies.

**Theorem 1.4.3** (Hansen [34]). Let $I_{\mathscr{O}}$ and $I_{\mathscr{O}'}$ be two equivalent cyclic strategies that begin with the same pivot element. Let $A^{(N)}$ and $A'^{(N)}$ be matrices obtained after one full cycle of the Jacobi method under strategies $I_{\mathscr{O}}$ and $I_{\mathscr{O}'}$, respectively. Then,

$$A^{(N)} = A'^{(N)}.$$

More than that, in 1989 Shroff and Shreiber proved that the convergence under $I_{\mathscr{O}}$ implies the convergence for all strategies weakly equivalent to $I_{\mathscr{O}}$.

**Proposition 1.4.4** (Shroff, Schreiber [69]). Let $\mathscr{O} \overset{w}{\sim} \mathscr{O}'$, $\mathscr{O}, \mathscr{O}' \in \mathscr{O}(\mathscr{P}_n)$. If the cyclic Jacobi method converges under the strategy $I_{\mathscr{O}}$, then it also converges under the strategy $I_{\mathscr{O}'}$.

As a consequence of the result, the Jacobi method converges under the class of weakly wavefront strategies. The convergence of matrices $A^{(k)}$, $k \geq 0$ to a diagonal matrix $\Lambda$ is equivalent to the convergence of their off-norms to zero [61].

Since we are going to explore the use of generalized serial pivot strategies with permutations, defined in Section 1.2, we are going to need some newer results for the Jacobi method under such strategies. Begović Kovač and Hari [40] in 2021 produced a result for a sequence of Jacobi operators defined by the ordering $\mathscr{O} \in \mathscr{C}_{sg}^{(n)}$.

**Theorem 1.4.5** (Begović Kovač, Hari [40]). Let $\mathscr{O} \in \mathscr{C}_{sg}^{(n)}$. Suppose that $\mathscr{O} \overset{p}{\sim} \mathscr{O}' \overset{w}{\sim} \mathscr{O}''$ or $\mathscr{O} \overset{w}{\sim} \mathscr{O}' \overset{p}{\sim} \mathscr{O}''$, $\mathscr{O}'' \in \mathscr{C}_{sp}^{(n)}$ and that the weak equivalence relation is in canonical form containing $d$ shift equivalences. Then for any $d+1$ Jacobi operators $\mathscr{J}_1, \mathscr{J}_2, \ldots, \mathscr{J}_{d+1} \in \mathscr{J}_{\mathscr{O}}^{\omega}$, $0 < \omega \leq 1$, there is a constant $\zeta_{n,\omega}$ depending only on $n$ and $\omega$ such that it holds

$$\|\mathscr{J}_1 \mathscr{J}_2 \cdots \mathscr{J}_{d+1}\|_2 \leq \zeta_{n,\omega}, \quad 0 \leq \zeta_{n,\omega} < 1.$$

Using the general result for Jacobi operators from Theorem 1.4.5, the authors proved the convergence of the Jacobi method under the concerning strategies.

**Corollary 1.4.6** (Begović Kovač, Hari [40]). Let $A$ be a Hermitian matrix of order $n$. Let $\mathscr{O} \in \mathscr{C}_{sg}^{(n)}$. Suppose that $\mathscr{O} \overset{p}{\sim} \mathscr{O}' \overset{w}{\sim} \mathscr{O}''$ or $\mathscr{O} \overset{w}{\sim} \mathscr{O}' \overset{p}{\sim} \mathscr{O}''$, $\mathscr{O}'' \in \mathscr{C}_{sp}^{(n)}$ and that the weak equivalence relation is in canonical form containing $d$ shift equivalences. Let $A'$ be obtained from $A$ by applying $d+1$ cycles of the Jacobi method defined by strategy $I_{\mathscr{O}}$. If

all rotation angles satisfy $\varphi_k \in \left[-\frac{\pi}{4}, \frac{\pi}{4}\right]$, $k \leq 0$, then there is a constant $\gamma_n$ depending only on $n$ such that

$$\text{off}^2(A') \leq \gamma_n \, \text{off}^2(A), \quad 0 \leq \gamma_n < 1.$$

In this thesis, we do not work on the original Jacobi method, but on its variants, the so-called Jacobi-type methods. However, as we are going to see in the rest of the thesis, many concepts presented in this chapter can be adopted for the Jacobi-type methods.

# 2. CONVERGENCE OF THE EBERLEIN DIAGONALIZATION METHOD UNDER THE GENERALIZED SERIAL PIVOT STRATEGIES

One of the generalizations of the Jacobi method is known as the Eberlein method. The Eberlein method, originally proposed in 1962 by Patricia J. Eberlein [28], is a Jacobi-type process for solving the eigenvalue problem of an arbitrary matrix. It is one of the first efficient norm-reducing methods of this type. The iterative process on a general matrix $A \in \mathbb{C}^{n\times n}$ takes the form

$$A^{(k+1)} = T_k^{-1} A^{(k)} T_k, \quad k \geq 0, \tag{2.1}$$

where $A^{(0)} = A$ and

$$T_k = R_k S_k$$

are non-singular elementary matrices, the same as in [6]. In particular, matrices $R_k$ are plane rotations and $S_k$ are non-unitary elementary matrices. Transformations $R_k$ are chosen to annihilate the pivot element of the matrix $\left(A^{(k)} + (A^{(k)})^*\right)/2$, while transformations $S_k$ reduce the Frobenius norm of $A^{(k)}$. In Eberlein's experiments, the matrices $A^{(k)}$, $k \geq 0$, given by the process (2.1) converge to a normal matrix. Eberlein proved this convergence only under a specific non-cyclic pivot strategy.

Veselić [75] studied a slightly altered Eberlein algorithm where in the $k$th step only one transformation is applied, $R_k$ or $S_k$, but not both at the same time. He proved the convergence of this modified method under the classical Jacobi pivot strategy. Specifically, he

showed that, for an arbitrary $n \times n$ starting matrix $A^{(0)}$, the sequence $A^{(k)}$, $k \geq 0$, converges to a block diagonal normal matrix. At the same time, the sequence $\left(A^{(k)} + (A^{(k)})^*\right)/2$ converges to a diagonal matrix $\mathrm{diag}(\mu_1, \mu_2, \ldots, \mu_n)$, where $\{\mu_1, \mu_2, \ldots, \mu_n\}$ are the real parts of the eigenvalues of $A$. Later, Hari [35] proved the global convergence of the original method under the column/row cyclic pivot strategy on real matrices. In [67] Hari and Pupovci proved the convergence of the Eberlein method on complex matrices with the pivot strategies that are weakly equivalent to the row cyclic strategy. In the same paper authors considered the parallel method and proved its convergence under the pivot strategies that are weakly equivalent to the modulus strategy.

In this chapter, we extend the global convergence result for the Eberlein method to a significantly broader class of the cyclic pivot strategies — generalized serial strategies with permutations, explained earlier in Section 1.2. We consider the method in the form given in [67]. Our new result is the global convergence of the Eberlein method under the generalized serial pivot strategies with permutations. It is given in Theorem 2.3.3. We show that for an arbitrary $n \times n$ starting matrix $A^{(0)}$, the sequence $A^{(k)}$, $k \geq 0$, converges to a block diagonal normal matrix. At the same time, the sequence $\left(A^{(k)} + (A^{(k)})^*\right)/2$ converges to a diagonal matrix $\mathrm{diag}(\mu_1, \mu_2, \ldots, \mu_n)$, where $\{\mu_1, \mu_2, \ldots, \mu_n\}$ are the real parts of the eigenvalues of $A$. Moreover, we present several numerical examples and discuss the cases of the unique and the multiple eigenvalues.

The chapter is organized as follows. In Section 2.1 we describe the Eberlein method, its complex and real variant, while in Section 2.2 we state its convergence theory results from the literature up to now. The main part of the chapter is contained in Section 2.3 where we prove the convergence of the method under the generalized serial pivot strategies. Finally, in Section 2.4 we present the results of our numerical tests.

# 2.1. THE EBERLEIN METHOD

As was mentioned in the introduction of this chapter, there are several variations of the Eberlein method. The method can be applied to complex matrices using the transformations $T_k \in \mathbb{C}^{n\times n}$, $k \geq 0$, or one can observe the real method with $T_k \in \mathbb{R}^{n\times n}$, $k \geq 0$. Here we mostly focus on the complex method. We describe it in Subsection 2.1.1. In Subsection 2.1.2 we outline the real case. We use the notation $\imath = \sqrt{-1}$. For a complex number $x$, $\mathrm{Re}(x)$ stands for the real part of $x$ and $\mathrm{Im}(x)$ stands for its imaginary part.

## 2.1.1. Complex case

The Eberlein method is an iterative Jacobi-type method used to find the eigenvalues and eigenvectors of an arbitrary matrix $A \in \mathbb{C}^{n\times n}$. One iteration step of the method is given by the relation (2.1). In the $k$th iteration, transformation $T_k$ is a elementary matrix that differs from the identity only in one of its $2\times 2$ principal submatrices $\widehat{T}_k$ determined by the pivot pair $(p_k, q_k)$,

$$\widehat{T}_k = \begin{bmatrix} t^{(k)}_{p_k p_k} & t^{(k)}_{p_k q_k} \\ t^{(k)}_{q_k p_k} & t^{(k)}_{q_k q_k} \end{bmatrix}.$$

Matrix $T_k$ is set to be the product of two nonsingular matrices, a plane rotation $R_k$ and a non-unitary elementary matrix $S_k$. That is, $T_k = R_k S_k$. Denote the $k$th pivot pair by $(p,q) = (p_k, q_k)$. The pivot pair is the same for both $R_k$ and $S_k$, and consequently for $T_k$. In addition to $(p,q)$, matrices $R_k$ and $S_k$ depend on the transformation angles $\alpha_k, \varphi_k$, and $\beta_k, \psi_k$, respectively. The pivot submatrix $\widehat{T}_k$ is equal to $\widehat{T}_k = \widehat{R}_k \widehat{S}_k \in \mathbb{C}^{2\times 2}$, where

$$\widehat{R}_k = \begin{bmatrix} \cos\varphi_k & -e^{\imath\alpha_k}\sin\varphi_k \\ e^{-\imath\alpha_k}\sin\varphi_k & \cos\varphi_k \end{bmatrix}, \quad \widehat{S}_k = \begin{bmatrix} \cosh\psi_k & -\imath e^{\imath\beta_k}\sinh\psi_k \\ \imath e^{-\imath\beta_k}\sinh\psi_k & \cosh\psi_k \end{bmatrix}. \tag{2.2}$$

We are going to show how to choose $\widehat{R}_k$ and $\widehat{S}_k$ in the Eberlein method. The process (2.1) can be written with an intermediate step,

$$A^{(k)} \to \widetilde{A}^{(k)} \to A^{(k+1)},$$

where

$$\widetilde{A}^{(k)} = R_k^* A^{(k)} R_k,$$
$$A^{(k+1)} = S_k^{-1} \widetilde{A}^{(k)} S_k, \quad k \geq 0.$$

The transformations effect only the elements from the $p$th and $q$th row and column of $A^{(k)}$. The elements of $\tilde{A}^{(k)} = (\tilde{a}_{ij}^{(k)})$ are computed as follows:

$$\begin{aligned}
\tilde{a}_{ij}^{(k)} &= a_{ij}^{(k)} \text{ if } (i,j) \text{ and } (p,q) \text{ are disjoint,}\\
\tilde{a}_{pi}^{(k)} &= a_{pi}^{(k)} \cos\varphi_k + e^{\imath\alpha_k} a_{qi}^{(k)} \sin\varphi_k,\\
\tilde{a}_{ip}^{(k)} &= a_{ip}^{(k)} \cos\varphi_k + e^{-\imath\alpha_k} a_{iq}^{(k)} \sin\varphi_k,\\
\tilde{a}_{qi}^{(k)} &= a_{qi}^{(k)} \cos\varphi_k - e^{-\imath\alpha_k} a_{pi}^{(k)} \sin\varphi_k,\\
\tilde{a}_{iq}^{(k)} &= a_{iq}^{(k)} \cos\varphi_k - e^{\imath\alpha_k} a_{ip}^{(k)} \sin\varphi_k,\\
\tilde{a}_{pp}^{(k)} &= \frac{1}{2}(a_{pp}^{(k)} + a_{qq}^{(k)} + d_{pq}^{(k)} \cos 2\varphi_k + \xi_{pq}^{(k)} \sin 2\varphi_k),\\
\tilde{a}_{qq}^{(k)} &= \frac{1}{2}(a_{pp}^{(k)} + a_{qq}^{(k)} - d_{pq}^{(k)} \cos 2\varphi_k - \xi_{pq}^{(k)} \sin 2\varphi_k),\\
\tilde{a}_{pq}^{(k)} &= \frac{1}{2} e^{\imath\alpha_k} (\eta_{pq}^{(k)} - d_{pq}^{(k)} \sin 2\varphi_k + \xi_{pq}^{(k)} \cos 2\varphi_k),\\
\tilde{a}_{qp}^{(k)} &= \frac{1}{2} e^{-\imath\alpha_k} (-\eta_{pq}^{(k)} - d_{pq}^{(k)} \sin 2\varphi_k + \xi_{pq}^{(k)} \cos 2\varphi_k),
\end{aligned} \tag{2.3}$$

where

$$\begin{aligned}
d_{pq}^{(k)} &= a_{pp}^{(k)} - a_{qq}^{(k)},\\
\xi_{pq}^{(k)} &= (a_{pq}^{(k)} + a_{qp}^{(k)}) \cos\alpha_k - \imath(a_{pq}^{(k)} - a_{qp}^{(k)}) \sin\alpha_k,\\
\eta_{pq}^{(k)} &= (a_{pq}^{(k)} - a_{qp}^{(k)}) \cos\alpha_k - \imath(a_{pq}^{(k)} + a_{qp}^{(k)}) \sin\alpha_k.
\end{aligned}$$

Similarly, elements of $A^{(k+1)}$ are then obtained from $\tilde{A}^{(k)}$ by the following rules:

$$\begin{aligned}
a_{ij}^{(k+1)} &= \tilde{a}_{ij}^{(k)} \text{ if } (i,j) \text{ and } (p,q) \text{ are disjoint,}\\
a_{pi}^{(k+1)} &= \tilde{a}_{pi}^{(k)} \cosh\psi_k + \imath e^{\imath\beta_k} \tilde{a}_{qi}^{(k)} \sinh\psi_k,\\
a_{ip}^{(k+1)} &= \tilde{a}_{ip}^{(k)} \cosh\psi_k + \imath e^{-\imath\beta_k} \tilde{a}_{iq}^{(k)} \sinh\psi_k,\\
a_{qi}^{(k+1)} &= \tilde{a}_{qi}^{(k)} \cosh\psi_k - \imath e^{-\imath\beta_k} \tilde{a}_{pi}^{(k)} \sinh\psi_k,\\
a_{iq}^{(k+1)} &= \tilde{a}_{iq}^{(k)} \cosh\psi_k - \imath e^{\imath\beta_k} \tilde{a}_{ip}^{(k)} \sinh\psi_k,\\
a_{pp}^{(k+1)} &= \frac{1}{2}(\tilde{a}_{pp}^{(k)} + \tilde{a}_{qq}^{(k)} + \tilde{d}_{pq}^{(k)} \cosh 2\psi_k + \imath\tilde{\xi}_{pq}^{(k)} \sinh 2\psi_k),\\
a_{qq}^{(k+1)} &= \frac{1}{2}(\tilde{a}_{pp}^{(k)} + \tilde{a}_{qq}^{(k)} - \tilde{d}_{pq}^{(k)} \cosh 2\psi_k - \imath\tilde{\xi}_{pq}^{(k)} \sinh 2\psi_k),\\
a_{pq}^{(k+1)} &= \frac{1}{2} e^{\imath\beta_k} (\tilde{\eta}_{pq}^{(k)} - \imath\tilde{d}_{pq}^{(k)} \sinh 2\psi_k + \tilde{\xi}_{pq}^{(k)} \cosh 2\psi_k),\\
a_{qp}^{(k+1)} &= \frac{1}{2} e^{-\imath\beta_k} (-\tilde{\eta}_{pq}^{(k)} - \imath\tilde{d}_{pq}^{(k)} \sinh 2\psi_k + \tilde{\xi}_{pq}^{(k)} \cosh 2\psi_k),
\end{aligned} \tag{2.4}$$

where

$$\begin{aligned}
\tilde{d}_{pq}^{(k)} &= \tilde{a}_{pp}^{(k)} - \tilde{a}_{qq}^{(k)}, \\
\tilde{\xi}_{pq}^{(k)} &= (\tilde{a}_{pq}^{(k)} + \tilde{a}_{qp}^{(k)})\cos\beta_k - \imath(\tilde{a}_{pq}^{(k)} - \tilde{a}_{qp}^{(k)})\sin\beta_k, \\
\tilde{\eta}_{pq}^{(k)} &= (\tilde{a}_{pq}^{(k)} - \tilde{a}_{qp}^{(k)})\cos\beta_k - \imath(\tilde{a}_{pq}^{(k)} + \tilde{a}_{qp}^{(k)})\sin\beta_k.
\end{aligned} \tag{2.5}$$

Let

$$\begin{aligned}
B^{(k)} &= \frac{1}{2}(A^{(k)} + (A^{(k)})^*), \\
\widetilde{B}^{(k)} &= R_k^* B^{(k)} R_k.
\end{aligned} \tag{2.6}$$

The matrix $B^{(k)}$ is the Hermitian part of $A^{(k)}$ and, likewise, $\widetilde{B}^{(k)}$ is the Hermitian part of $\widetilde{A}^{(k)}$. Next, let $C$ be an operator defined by

$$C(A) = AA^* - A^*A. \tag{2.7}$$

We denote $C(A^{(k)}) = (c_{ij}^{(k)})$, $C(\widetilde{A}^{(k)}) = (\tilde{c}_{ij}^{(k)})$, $B^{(k)} = (b_{ij}^{(k)})$ and $\widetilde{B}^{(k)} = (\tilde{b}_{ij}^{(k)})$. Obviously, $C(A) = 0$, if and only if $A$ is a normal matrix. The definition of $C(A)$ is linked to one of the measures of nonnormality of matrices given by Elsner and Paardekooper in [29]. The reason $C(A)$ is introduced because the convergence theorem 2.3.3 is going to state that matrices $A^{(k)}$, $k \geq 0$, from the Eberlein process converge to a normal matrix.

The rotation $R_k$ is chosen so that the element of $B^{(k)}$ at the pivot position $(p,q)$ is annihilated. The real number $\alpha_k$, as well as the sine and cosine of $\varphi_k$ in (2.2) are calculated from the following expressions,

$$\alpha_k = \arg(b_{pq}^{(k)}), \tag{2.8}$$

$$\tan 2\varphi_k = \frac{2|b_{pq}^{(k)}|}{b_{pp}^{(k)} - b_{qq}^{(k)}}, \quad |\varphi_k| \leq \frac{\pi}{4}. \tag{2.9}$$

These formulas are the same as for the complex Jacobi method on Hermitian matrices, (1.12) and (1.13), applied on the matrix $B^{(k)}$. Then, in order to get $\sin\varphi$ and $\cos\varphi$ in a stable way, we use formulas equivalent to (1.14) and (1.9),

$$\begin{aligned}
\tan\varphi_k &= \frac{2|b_{pq}^{(k)}|\operatorname{sign}(b_{pp}^{(k)} - b_{qq}^{(k)})}{|b_{pp}^{(k)} - b_{qq}^{(k)}| + \sqrt{|b_{pp}^{(k)} - b_{qq}^{(k)}|^2 + 4|b_{pq}^{(k)}|^2}}, \\
\cos\varphi_k &= \frac{1}{\sqrt{1+\tan^2\varphi_k}}, \ \sin\varphi_k = \frac{\tan\varphi_k}{\sqrt{1+\tan^2\varphi_k}}.
\end{aligned} \tag{2.10}$$

On the other hand, $S_k$ is chosen to reduce the Frobenius norm of $A^{(k)}$. Set

$$\Delta_k := \|A^{(k)}\|_F^2 - \|A^{(k+1)}\|_F^2.$$

Eberlein [28] proved that

$$\begin{aligned}\Delta_k = \|\widetilde{A}^{(k)}\|_F^2 - \|A^{(k+1)}\|_F^2 &= g_{pq}^{(k)}(1-\cosh 2\psi_k) - h_{pq}^{(k)}\sinh 2\psi_k \\ &+ \frac{1}{2}(|\tilde{\xi}_{pq}^{(k)}|^2 + |\tilde{d}_{pq}^{(k)}|^2)(1-\cosh 4\psi_k) + \mathrm{Im}(\tilde{\xi}_{pq}^{(k)}\tilde{d}_{pq}^{(k)*})\sinh 4\psi_k,\end{aligned}$$

where

$$\begin{aligned} g_{pq}^{(k)} &= \sum_{\substack{i=1\\ i\neq p,q}}^{n} |\tilde{a}_{ip}^{(k)}|^2 + |\tilde{a}_{pi}^{(k)}|^2 + |\tilde{a}_{iq}^{(k)}|^2 + |\tilde{a}_{qi}^{(k)}|^2, \\ h_{pq}^{(k)} &= -\mathrm{Re}(l_{pq}^{(k)})\sin\beta_k + \mathrm{Im}(l_{pq}^{(k)})\cos\beta_k, \\ l_{pq}^{(k)} &= 2\sum_{\substack{i=1\\ i\neq p,q}}^{n} (\tilde{a}_{pi}^{(k)}\tilde{a}_{qi}^{(k)*} - \tilde{a}_{ip}^{(k)*}\tilde{a}_{iq}^{(k)}). \end{aligned} \tag{2.11}$$

It is shown in [28] that the choice of $\beta_k$ and $\psi_k$ such that

$$\tan\beta_k = -\frac{\mathrm{Re}(\tilde{c}_{pq}^{(k)})}{\mathrm{Im}(\tilde{c}_{pq}^{(k)})}, \tag{2.12}$$

$$\tanh\psi_k = \frac{\mathrm{Im}(\tilde{\xi}_{pq}^{(k)}\tilde{d}_{pq}^{(k)*}) - h_{pq}^{(k)}/2}{g_{pq}^{(k)} + 2(|\tilde{\xi}_{pq}^{(k)}|^2 + |\tilde{d}_{pq}^{(k)}|^2)}, \tag{2.13}$$

implies

$$\Delta_k \geq \frac{1}{3}\frac{|\tilde{c}_{pq}^{(k)}|^2}{\|A^{(k)}\|_F^2} \geq \frac{1}{3}\frac{|\tilde{c}_{pq}^{(k)}|^2}{\|A\|_F^2}, \quad k \geq 1. \tag{2.14}$$

The values of $\beta_k$ and $\psi_k$ determined by (2.12) and (2.13) are an approximation of the solution that maximizes $\Delta_k$. We compute hyperbolic cosine and sine from (2.13) as

$$\cosh\psi_k = \frac{1}{\sqrt{1-\tanh^2\psi_k}},\ \sinh\psi_k = \frac{\tanh\psi_k}{\sqrt{1-\tanh^2\psi_k}},$$

that is, we take positive values of both functions.

Instead of (2.13), we can use a simpler formula for computing $\psi_k$,

$$\tanh\psi_k = \frac{\mathrm{Re}(\tilde{c}_{pq}^{(k)})\sin\beta_k - \mathrm{Im}(\tilde{c}_{pq}^{(k)})\cos\beta_k}{g_{pq}^{(k)} + 2(|\tilde{\xi}_{pq}^{(k)}|^2 + |\tilde{d}_{pq}^{(k)}|^2)}. \tag{2.15}$$

Let us prove that

$$\mathrm{Im}(\tilde{\xi}_{pq}^{(k)}\tilde{d}_{pq}^{(k)*}) - \frac{h_{pq}^{(k)}}{2} = \mathrm{Re}(\tilde{c}_{pq}^{(k)})\sin\beta_k - \mathrm{Im}(\tilde{c}_{pq}^{(k)})\cos\beta_k. \tag{2.16}$$

Recall that the pivot element of $C(\tilde{A}^{(k)})$ is

$$\tilde{c}_{pq}^{(k)} = \sum_{i=1}^{n} (\tilde{a}_{pi}^{(k)} \tilde{a}_{qi}^{(k)*} - \tilde{a}_{ip}^{(k)*} \tilde{a}_{iq}^{(k)}).$$

Let

$$\tilde{c}_{pq}^{(k)} = \tilde{\chi}_{pq}^{(k)} + \frac{l_{pq}^{(k)}}{2}, \tag{2.17}$$

where

$$\tilde{\chi}_{pq}^{(k)} = \tilde{a}_{pp}^{(k)} \tilde{a}_{qp}^{(k)*} - \tilde{a}_{pp}^{(k)*} \tilde{a}_{pq}^{(k)} + \tilde{a}_{pq}^{(k)} \tilde{a}_{qq}^{(k)*} - \tilde{a}_{qp}^{(k)*} \tilde{a}_{qq}^{(k)}.$$

Using simple manipulation of the real and imaginary parts we get

$$\begin{aligned}
\tilde{\chi}_{pq}^{(k)} &= \tilde{a}_{qp}^{(k)*} \tilde{d}_{pq}^{(k)} - \tilde{a}_{pq}^{(k)} \tilde{d}_{pq}^{(k)*} \\
&= \mathrm{Re}(\tilde{a}_{qp}^{(k)*} \tilde{d}_{pq}^{(k)} - \tilde{a}_{pq}^{(k)} \tilde{d}_{pq}^{(k)*}) + \imath \mathrm{Im}(\tilde{a}_{qp}^{(k)*} \tilde{d}_{pq}^{(k)} - \tilde{a}_{pq}^{(k)} \tilde{d}_{pq}^{(k)*}) \\
&= \mathrm{Re}(\tilde{a}_{qp}^{(k)*} \tilde{d}_{pq}^{(k)}) - \mathrm{Re}(\tilde{a}_{pq}^{(k)} \tilde{d}_{pq}^{(k)*}) + \imath \mathrm{Im}(\tilde{a}_{qp}^{(k)*} \tilde{d}_{pq}^{(k)}) - \imath \mathrm{Im}(\tilde{a}_{pq}^{(k)} \tilde{d}_{pq}^{(k)*}) \\
&= \mathrm{Re}(\tilde{a}_{qp}^{(k)} \tilde{d}_{pq}^{(k)*}) - \mathrm{Re}(\tilde{a}_{pq}^{(k)} \tilde{d}_{pq}^{(k)*}) - \imath \mathrm{Im}(\tilde{a}_{qp}^{(k)} \tilde{d}_{pq}^{(k)*}) - \imath \mathrm{Im}(\tilde{a}_{pq}^{(k)} \tilde{d}_{pq}^{(k)*}) \\
&= -\mathrm{Re}\left( (\tilde{a}_{pq}^{(k)} - \tilde{a}_{qp}^{(k)}) \tilde{d}_{pq}^{(k)*} \right) - \imath \mathrm{Im}\left( (\tilde{a}_{pq}^{(k)} + \tilde{a}_{qp}^{(k)}) \tilde{d}_{pq}^{(k)*} \right).
\end{aligned} \tag{2.18}$$

From the relations (2.5) and (2.18) we obtain

$$\begin{aligned}
&\mathrm{Re}(\tilde{\chi}_{pq}^{(k)}) \sin\beta_k - \mathrm{Im}(\tilde{\chi}_{pq}^{(k)}) \cos\beta_k \\
&\quad = -\mathrm{Re}\left( (\tilde{a}_{pq}^{(k)} - \tilde{a}_{qp}^{(k)}) \tilde{d}_{pq}^{(k)*} \right) \sin\beta_k + \mathrm{Im}\left( \tilde{a}_{pq}^{(k)} + \tilde{a}_{qp}^{(k)}) \tilde{d}_{pq}^{(k)*} \right) \cos\beta_k \\
&\quad = \mathrm{Im}\left( (\tilde{a}_{pq}^{(k)} + \tilde{a}_{qp}^{(k)}) \tilde{d}_{pq}^{(k)*} \cos\beta_k - \imath (\tilde{a}_{pq}^{(k)} - \tilde{a}_{qp}^{(k)}) \tilde{d}_{pq}^{(k)*} \sin\beta_k \right) \\
&\quad = \mathrm{Im}\left( \left( (\tilde{a}_{pq}^{(k)} + \tilde{a}_{qp}^{(k)}) \cos\beta_k - \imath (\tilde{a}_{pq}^{(k)} - \tilde{a}_{qp}^{(k)}) \sin\beta_k \right) \tilde{d}_{pq}^{(k)*} \right) \\
&\quad = \mathrm{Im}(\tilde{\xi}_{pq}^{(k)} \tilde{d}_{pq}^{(k)*}).
\end{aligned} \tag{2.19}$$

The equation (2.16) now follows from the relations (2.11), (2.17) and (2.19).

We summarize the procedure in Algorithm 4. One should keep in mind that it is not needed to formulate matrices $\widetilde{A}^{(k)}$ explicitly, only the $p$th and the $q$th row and column.

### 2.1.2. Real case

Suppose that $A$ is a real matrix and we wish for the iterates $A^{(k)}$ to stay real during the process (2.1). In order to satisfy this request we modify the complex algorithm. Firstly, we can take $\alpha_k = \pi$ and $\beta_k = \pi/2$. This implies

$$\widehat{R}_k = \begin{bmatrix} \cos\varphi_k & \sin\varphi_k \\ -\sin\varphi_k & \cos\varphi_k \end{bmatrix}, \quad \widehat{S}_k = \begin{bmatrix} \cosh\psi_k & \sinh\psi_k \\ \sinh\psi_k & \cosh\psi_k \end{bmatrix}.$$

**Algorithm 4** Eberlein method

**Input:** $A \in \mathbb{C}^{n\times n}$

**Output:** matrix $A^{(k)}$

$A^{(0)} = A$

$k = 0$

**repeat**

Choose pivot pair $(p,q)$ according to the pivot strategy.

Find $\alpha_k$ using (2.8), and $\sin\varphi_k$, $\cos\varphi_k$ using (2.10).

$\widetilde{A}^{(k)} = R_k^* A^{(k)} R_k$

Find $\beta_k$ using (2.12), and $\sin\psi_k$, $\cos\psi_k$ using (2.15).

$A^{(k+1)} = S_k^{-1}\widetilde{A}^{(k)} S_k$

$k = k+1$

**until** convergence

With the same intermediate step as before, the transformations effect only the elements of the $p$th and $q$th row and column of $A^{(k)}$. We update the relations (2.3) for the elements of $\tilde{A}^{(k)}$ keeping in mind that $\alpha_k = \pi$ and $\beta_k = \pi/2$:

$$
\begin{aligned}
\tilde{a}_{ij}^{(k)} &= a_{ij}^{(k)} \text{ if } (i,j) \text{ and } (p,q) \text{ are disjoint,}\\
\tilde{a}_{pi}^{(k)} &= a_{pi}^{(k)}\cos\varphi_k - a_{qi}^{(k)}\sin\varphi_k,\\
\tilde{a}_{ip}^{(k)} &= a_{ip}^{(k)}\cos\varphi_k - a_{iq}^{(k)}\sin\varphi_k,\\
\tilde{a}_{qi}^{(k)} &= a_{qi}^{(k)}\cos\varphi_k + a_{pi}^{(k)}\sin\varphi_k,\\
\tilde{a}_{iq}^{(k)} &= a_{iq}^{(k)}\cos\varphi_k + a_{ip}^{(k)}\sin\varphi_k,\\
\tilde{a}_{pp}^{(k)} &= \frac{1}{2}(a_{pp}^{(k)} + a_{qq}^{(k)} + d_{pq}^{(k)}\cos 2\varphi_k + \xi_{pq}^{(k)}\sin 2\varphi_k),\\
\tilde{a}_{qq}^{(k)} &= \frac{1}{2}(a_{pp}^{(k)} + a_{qq}^{(k)} - d_{pq}^{(k)}\cos 2\varphi_k - \xi_{pq}^{(k)}\sin 2\varphi_k),\\
\tilde{a}_{pq}^{(k)} &= \frac{1}{2}(-\eta_{pq}^{(k)} + d_{pq}^{(k)}\sin 2\varphi_k - \xi_{pq}^{(k)}\cos 2\varphi_k),\\
\tilde{a}_{qp}^{(k)} &= \frac{1}{2}(\eta_{pq}^{(k)} + d_{pq}^{(k)}\sin 2\varphi_k - \xi_{pq}^{(k)}\cos 2\varphi_k),
\end{aligned}
\tag{2.20}
$$

where

$$\begin{aligned} d_{pq}^{(k)} &= a_{pp}^{(k)} - a_{qq}^{(k)}, \\ \xi_{pq}^{(k)} &= -(a_{pq}^{(k)} + a_{qp}^{(k)}), \\ \eta_{pq}^{(k)} &= a_{qp}^{(k)} - a_{pq}^{(k)}. \end{aligned}$$

Similarly, from the relations (2.4), elements of $A^{(k+1)}$ are then obtained by the following rules:

$$\begin{aligned} a_{ij}^{(k+1)} &= \tilde{a}_{ij}^{(k)} \text{ if } (i,j) \text{ and } (p,q) \text{ are disjoint}, \\ a_{pi}^{(k+1)} &= \tilde{a}_{pi}^{(k)} \cosh\psi_k - \tilde{a}_{qi}^{(k)} \sinh\psi_k, \\ a_{ip}^{(k+1)} &= \tilde{a}_{ip}^{(k)} \cosh\psi_k + \tilde{a}_{iq}^{(k)} \sinh\psi_k, \\ a_{qi}^{(k+1)} &= \tilde{a}_{qi}^{(k)} \cosh\psi_k - \tilde{a}_{pi}^{(k)} \sinh\psi_k, \\ a_{iq}^{(k+1)} &= \tilde{a}_{iq}^{(k)} \cosh\psi_k + \tilde{a}_{ip}^{(k)} \sinh\psi_k, \\ a_{pp}^{(k+1)} &= \frac{1}{2}(\tilde{a}_{pp}^{(k)} + \tilde{a}_{qq}^{(k)} + \tilde{d}_{pq}^{(k)} \cosh 2\psi_k + \tilde{e}_{pq}^{(k)} \sinh 2\psi_k), \\ a_{qq}^{(k+1)} &= \frac{1}{2}(\tilde{a}_{pp}^{(k)} + \tilde{a}_{qq}^{(k)} - \tilde{d}_{pq}^{(k)} \cosh 2\psi_k - \tilde{e}_{pq}^{(k)} \sinh 2\psi_k), \\ a_{pq}^{(k+1)} &= \frac{1}{2}(\tilde{a}_{pq}^{(k)} + \tilde{a}_{qp}^{(k)} + \tilde{d}_{pq}^{(k)} \sinh 2\psi_k + \tilde{e}_{pq}^{(k)} \cosh 2\psi_k), \\ a_{qp}^{(k+1)} &= \frac{1}{2}(\tilde{a}_{pq}^{(k)} + \tilde{a}_{qp}^{(k)} - \tilde{d}_{pq}^{(k)} \sinh 2\psi_k - \tilde{e}_{pq}^{(k)} \cosh 2\psi_k), \end{aligned} \tag{2.21}$$

where

$$\begin{aligned} \tilde{d}_{pq}^{(k)} &= \tilde{a}_{pp}^{(k)} - \tilde{a}_{qq}^{(k)}, \\ \tilde{e}_{pq}^{(k)} &= \tilde{a}_{pq}^{(k)} - \tilde{a}_{qp}^{(k)}. \end{aligned}$$

As before, we do not need to calculate the angles $\varphi$ and $\psi$ directly. It is sufficient to find the matrices $\widehat{R}_k$ and $\widehat{S}_k$. As in the complex case, $\varphi_k$ is selected to annihilate the pivot element of $B^{(k)}$ while $\psi_k$ is chosen to reduce $\|A^{(k)}\|_F$. The angle $\varphi_k$ is calculated from the relation similar to (2.9),

$$\tan 2\varphi_k = \frac{2b_{pq}^{(k)}}{b_{qq}^{(k)} - b_{pp}^{(k)}}, \quad |\varphi_k| \leq \frac{\pi}{4}.$$

This formula is the same as (1.7), using the real Jacobi method on symmetric matrices applied on the matrix $B^{(k)}$. Considering that $\beta_k = \pi/2$ and that all the elements of $A^{(k)}$ are

real, the formula (2.15) for $\psi_k$ is transformed into

$$\tanh \psi_k = \frac{\tilde{c}_{pq}^{(k)}}{g_{pq}^{(k)} + 2\left((\tilde{e}_{pq}^{(k)})^2 + (\tilde{d}_{pq}^{(k)})^2\right)},$$

where $g_{pq}^{(k)}$ is the same as in the complex case.

## 2.2. CONVERGENCE RESULTS FROM THE LITERATURE

In this section, we give an overview of the convergence results for the Eberlein method under different pivot strategies. Let $A$ be an arbitrary complex (real) matrix and let the sequence

$$A^{(0)} = A, A^{(1)}, A^{(2)}, \ldots,$$

be generated by the complex (real) Eberlein process (2.1). In studying the convergence of the Eberlein method, we focus on several features. Firstly, we want the sequence of Hermitian parts of $A^{(k)}$, that is, the matrices $B^{(k)}$, $k \geq 0$, to converge to a diagonal matrix. Secondly, we want the sequence $(A^{(k)}, k \geq 0)$, to converge to a normal matrix $\Lambda$.

We are going to show that convergence of $(B^{(k)},\ k \geq 0)$ to a diagonal matrix will happen due to the the rotation part of the transformations $T_k$, that is, plane rotations $R_k$, which annihilate the pivot elements of $B^{(k)}$. On the other hand, the convergence of $(A^{(k)},\ k \geq 0)$ to a normal matrix will be the part of the transformation $T_k$ that reduces the Frobenius norm of $A^{(k)}$. Further on, if all the real parts of the eignevalues of $A$ are different, then $(A^{(k)},\ k \geq 0)$ will converge to a diagonal matrix $\Lambda$. Otherwise, if $A$ has some eigenvalues with equal real parts, $\Lambda$ will be a block diagonal matrix, such that the diagonal block sizes correspond to the number of times the same real part appears in the spectrum of $A$.

In each iteration $k$ of the Algorithm 4, pivot position is selected according to the pivot strategy. In [28], numerical experiments showed that the real and complex Eberlein method converged under the cyclic row-wise strategy. Nevertheless, the convergence for the real case was not given, while the complex case convergence was proved only under a specific non-cyclic strategy.

**Theorem 2.2.1** (Eberlein [28])**.** Let $A \in \mathbb{C}^{n\times n}$ and let $(A^{(k)}, k \geq 0)$ be a sequence generated by the Eberlein method. At every step $k$ the pivot pair $(p,q) = (p_k, q_k)$ is chosen such that

$$4|c_{pq}^{(k)}|^2 + (c_{pp}^{(k)} - c_{qq}^{(k)})^2$$

is greater or equal to the average of all possible results for

$$4|c_{ij}^{(k)}|^2 + (c_{ii}^{(k)} - c_{jj}^{(k)})^2, \quad 1 \le i < j \le n.$$

Then

$$\lim_{k\to\infty} \|C(A^{(k)})\|_F^2 = 0,$$

i.e., for $k$ sufficiently large, $A^{(k)}$ is arbitrarily close to being normal.

Veselić [75] proved the convergence of a slightly modified Eberlein method for real matrices. In this altered algorithm, only one transformation is applied in the $k$th step, either $R_k$ or $S_k$, but not both. He proved the convergence under the classical Jacobi pivot strategy.

**Theorem 2.2.2** (Veselić [75])**.** Let $A \in \mathbb{R}^{n\times n}$. Let $(A^{(k)}, k \ge 0)$ be a sequence generated by (2.1) where $T_k = R_k$ or $S_k$, depending on which of the numbers

$$a^{(k)} = \max_{\substack{i,j \\ i\neq j}} |a_{ij}^{(k)} + a_{ji}^{(k)}| \quad \text{or} \quad c^{(k)} = \max_{\substack{i,j \\ i\neq j}} \sqrt{|c_{ij}^{(k)}|}$$

is larger. The pivot pair $(p_k, q_k) = (p, q)$ is chosen to be that for which $|a_{pq}^{(k)} + a_{qp}^{(k)}|$ $\left(\text{or } \sqrt{|c_{pq}^{(k)}|}\right)$, $p \neq q$, achieves its maximum. Then

(i) The sequence $(A^{(k)}, k \ge 0)$ tends to a normal matrix, that is,

$$\lim_{k\to\infty} C(A^{(k)}) = 0.$$

(ii) The sequence of matrices $(B^{(k)}, k \ge 0)$ tends to a fixed diagonal matrix,

$$\lim_{k\to\infty} B^{(k)} = \operatorname{diag}(\mu_1, \mu_2, \ldots, \mu_n),$$

where $\mu_i$, $1 \le i \le n$, are real parts of the eigenvalues of $A$.

(iii) If $\mu_i \neq \mu_j$, then $\lim_{k\to\infty} a_{ij}^{(k)} = 0$.

(iv) If $\mu_i = \mu_j$ for a fixed pair $i \neq j$, and $\mu_r \neq \mu_i$ for all $r \neq i, j$, then $\lim_{k\to\infty} a_{ij}^{(k)} = \mu_{ij}$, where $\mu_{ij}$ is the imaginary part of an eigenvalue corresponding to $\mu_i$.

Hari [35] proved the convergence of the Eberlein method for real matrices under the wavefront strategies.

**Theorem 2.2.3** (Hari [35]). Let $A \in \mathbb{R}^{n\times n}$ and let $(A^{(k)}, k \geq 0)$ be a sequence generated by the Eberlein method under a wavefront pivot strategy. Then

(i) The sequence $(A^{(k)}, k \geq 0)$ tends to a normal matrix, that is,

$$\lim_{k\to\infty} C(A^{(k)}) = 0.$$

(ii) The sequence of matrices $(B^{(k)}, k \geq 0)$ tends to a fixed diagonal matrix,

$$\lim_{k\to\infty} B^{(k)} = \operatorname{diag}(\mu_1, \mu_2, \ldots, \mu_n),$$

where $\mu_i$, $1 \leq i \leq n$, are real parts of the eigenvalues of $A$.

(iii) If $\mu_i \neq \mu_j$, then $\lim_{k\to\infty} a_{ij}^{(k)} = 0$.

(iv) If $\mu_i = \mu_j$ for a fixed pair $i \neq j$, and $\mu_r \neq \mu_i$ for all $r \neq i, j$, then $\lim_{k\to\infty} a_{ij}^{(k)} = \mu_{ij}$, where $\mu_{ij}$ is the imaginary part of an eigenvalue corresponding to $\mu_i$.

In [67] Pupovci and Hari provided the convergence proof for the complex Eberlein method under the weakly wavefront strategies. In addition to that, they proved the convergence of the complex method under a parallel modulus strategy and the strategies that are weakly equivalent to it.

**Theorem 2.2.4** (Pupovci, Hari [67]). Let $A \in \mathbb{C}^{n\times n}$ and let $(A^{(k)}, k \geq 0)$ be a sequence generated by the Eberlein method under a weakly wavefront pivot strategy. Then

(i) The sequence $(A^{(k)}, k \geq 0)$ tends to a normal matrix, that is,

$$\lim_{k\to\infty} C(A^{(k)}) = 0.$$

(ii) The sequence of matrices $(B^{(k)}, k \geq 0)$ tends to a fixed diagonal matrix,

$$\lim_{k\to\infty} B^{(k)} = \operatorname{diag}(\mu_1, \mu_2, \ldots, \mu_n),$$

where $\mu_i$, $1 \leq i \leq n$, are real parts of the eigenvalues of $A$.

(iii) If $\mu_i \neq \mu_j$, then $\lim_{k\to\infty} a_{ij}^{(k)} = 0$ and $\lim_{k\to\infty} a_{ji}^{(k)} = 0$.

## 2.3. Convergence of the Eberlein method under the generalized serial strategies

We prove that the iterative process (2.1) converges under any pivot ordering $\mathscr{O} \in \mathscr{C}_{sg}^{(n)}$ described in Section 1.2. First, we list several auxiliary results from the literature and their direct implications. We use the notation introduced in Section 2.1.

(i) (Eberlein [28]) For $\|A^{(k)}\|_F^2$ we have

$$\Delta_k = \|A^{(k)}\|_F^2 - \|A^{(k+1)}\|_F^2 = \|\widetilde{A}^{(k)}\|_F^2 - \|A^{(k+1)}\|_F^2 \geq 0. \tag{2.22}$$

(ii) Since the sequence $(\|A^{(k)}\|_F^2, k \geq 0)$ is non-increasing and bounded from below by zero, it is convergent. Therefore, inequalities (2.22) and (2.14) imply

$$\lim_{k\to\infty} \tilde{c}_{pq}^{(k)} = 0. \tag{2.23}$$

(iii) (Hari [35]) For $\widetilde{A}^{(k)} = R_k^* A^{(k)} R_k$, $k \geq 0$, and

$$E^{(k)} = A^{(k+1)} - \widetilde{A}^{(k)}, \tag{2.24}$$

we have

$$\|E^{(k)}\|_F^2 \leq \frac{3}{2} n^2 |\tilde{c}_{pq}^{(k)}|. \tag{2.25}$$

(iv) (Hari [35]) For $\widetilde{B}^{(k)} = R_k^* B^{(k)} R_k$, $k \geq 0$, and

$$F^{(k)} = B^{(k+1)} - \widetilde{B}^{(k)}, \tag{2.26}$$

we have

$$\|F^{(k)}\|_F^2 \leq \frac{3}{2} n^2 |\tilde{c}_{pq}^{(k)}|. \tag{2.27}$$

(v) For any $k \geq 0$, we have

$$\begin{aligned} C(\widetilde{A}^{(k)}) &= C(R_k^* A^{(k)} R_k) \\ &= R_k^* A^{(k)} (A^{(k)})^* R_k - R_k^* (A^{(k)})^* A^{(k)} R_k \\ &= R_k^* (A^{(k)} (A^{(k)})^* - (A^{(k)})^* A^{(k)}) R_k \\ &= R_k^* C(A^{(k)}) R_k. \end{aligned} \tag{2.28}$$

Further, we prove the following two auxiliary propositions.

**Proposition 2.3.1.** Let $(x_k, k \geq 0)$ be a sequence of nonnegative real numbers such that

$$x_{k+1} = \gamma x_k + c_k, \quad 0 \leq \gamma < 1. \tag{2.29}$$

If $\lim_{k\to\infty} c_k = 0$, then

$$\lim_{k\to\infty} x_k = 0.$$

*Proof.* First, we show that the sequence (2.29) is bounded from above. Take

$$C = \max\{x_0, \sup_k c_k\}.$$

We prove the boundedness by mathematical induction. For $k = 0$,

$$x_0 \leq C \leq \frac{C}{1-\gamma} =: M, \quad \text{for } 0 \leq \gamma < 1.$$

Assume that $x_k \leq M$ for some given $k$. Then, for $k+1$,

$$x_{k+1} = \gamma x_k + c_k \leq \gamma M + C = \gamma M + (1-\gamma)M = M.$$

Therefore, $x_k \leq M$ for any $k \geq 0$.

For the limit superior, we take $\limsup_{k\to\infty} x_k = L \in \mathbb{R}$. Then,

$$L = \limsup_{k\to\infty} x_{k+1} \leq \gamma \limsup_{k\to\infty} x_k + \limsup_{k\to\infty} c_k = \gamma L.$$

Since $0 \leq \gamma < 1$, the upper inequality can hold only with $L = 0$. Since $(x_k)_k$ is the sequence of nonnegative real numbers, $\liminf_{k\to\infty} x_k \geq 0$. This implies that

$$\limsup_{k\to\infty} x_k = \liminf_{k\to\infty} x_k = 0$$

and $\lim_{k\to\infty} x_k = 0$. ■

**Proposition 2.3.2.** Let $H \neq 0$ be a Hermitian matrix. Let $(H^{(k)}, k \geq 0)$ be a sequence generated by applying the following iterative process on $H$,

$$H^{(k+1)} = R_k^* H^{(k)} R_k + M^{(k)}, \quad H^{(0)} = H, \quad k \geq 0, \tag{2.30}$$

where $R_k$ are complex plane rotations acting in the $(p_k, q_k)$ plane, $p_k < q_k$, with the rotation angles $|\varphi_k| \leq \frac{\pi}{4}$, $k \geq 0$. Suppose that the pivot strategy is defined by an ordering $\mathscr{O} \in \mathscr{C}_{sg}^{(n)}$ and that

$$\lim_{k\to\infty} \operatorname{off}(M^{(k)}) = 0. \tag{2.31}$$

Then,

$$\lim_{k\to\infty}\left|h_{p_kq_k}^{(k+1)}\right|=0 \quad \text{and} \quad \lim_{k\to\infty}\left|h_{q_kp_k}^{(k+1)}\right|=0 \tag{2.32}$$

imply

$$\lim_{k\to\infty}\operatorname{off}(H^{(k)})=0.$$

*Proof.* The proof uses the idea of the proof of Theorem 3.8 from [40].

To simplify the notation, let $(p,q)=(p_k,q_k)$ denote the pivot pair at step $k$. Transformation $R_k^*H^{(k)}R_k$ does not annihilate the elements on positions $(p,q)$ and $(q,p)$ of $H^{(k)}$, but we can write it as

$$R_k^*H^{(k)}R_k=\nu_{pq}(R_k^*H^{(k)}R_k)+(R_k^*H^{(k)}R_k)_{pq}\left(e_pe_q^*\right)+(R_k^*H^{(k)}R_k)_{qp}\left(e_qe_p^*\right), \tag{2.33}$$

where $e_r$ is the $r$th column vector of the identity matrix $I_n$ and $\nu_{pq}$ is as in (1.3.1). By using the vecoff operator on equation (2.30) and the definition of a Jacobi annihilator (1.3.1), from the relation (2.33) we get

$$\chi^{(k+1)}=\mathscr{R}_{p_kq_k}(R_k)\chi^{(k)}+m^{(k)}, \quad k\geq 0, \tag{2.34}$$

where $\chi^{(k)}=\operatorname{vecoff}(H^{(k)})$, and

$$\begin{aligned} m^{(k)} &= \operatorname{vecoff}\left(M^{(k)}+(R_k^*H^{(k)}R_k)_{pq}\left(e_pe_q^*\right)+(R_k^*H^{(k)}R_k)_{qp}\left(e_qe_p^*\right)\right)\\ &= \operatorname{vecoff}(M^{(k)})+(R_k^*H^{(k)}R_k)_{pq}e_{\tau(p,q)}+(R_k^*H^{(k)}R_k)_{qp}e_{\tau(q,p)}\\ &= \operatorname{vecoff}(M^{(k)})+(h_{pq}^{(k+1)}-M_{pq}^{(k)})e_{\tau(p,q)}+(h_{qp}^{(k+1)}-M_{qp}^{(k)})e_{\tau(q,p)}. \end{aligned} \tag{2.35}$$

Here, $\tau(p,q)$ stands for the position of the matrix element $x_{pq}$ in the vectorization $\operatorname{vecoff}(X)$ and $e_{\tau(p,q)}$ is the column vector of the identity matrix $I_{2N}$ with one on position $\tau(p,q)$. Relation (2.35) and the assumptions (2.31), (2.32) imply that

$$\lim_{k\to\infty}m^{(k)}=0. \tag{2.36}$$

We denote the matrix obtained from $H$ after $t$ cycles of the process (2.30) by $H^{(tN)}$. Vector $\chi^{(tN)}=\operatorname{vecoff}(H^{(tN)})$ can be written as

$$\chi^{(tN)}=\mathscr{J}_{\mathscr{O}}^{[tN]}\chi^{((t-1)N)}+m^{[tN]}, \quad t\geq 1.$$

The Jacobi operator $\mathscr{J}_{\mathscr{O}}^{[tN]}$ that appears in the upper equation is determined by the ordering $\mathscr{O}=(p_0,q_0),(p_1,q_1),\ldots,(p_{N-1},q_{N-1})\in\mathscr{O}(\mathscr{P}_n)$ and by the Jacobi annihilators,

$$\mathscr{J}_{\mathscr{O}}^{[tN]}=\mathscr{R}_{p_{N-1},q_{N-1}}(R_{tN-1})\mathscr{R}_{p_{N-2},q_{N-2}}(R_{tN-2})\cdots\mathscr{R}_{p_1,q_1}(R_{(t-1)N+1})\mathscr{R}_{p_0,q_0}(R_{(t-1)N}),$$

while

$$\begin{aligned} m^{[tN]} = &\mathscr{R}_{p_{N-1},q_{N-1}}(R_{tN-1})\cdots\mathscr{R}_{p_1,q_1}(R_{(t-1)N+1})\mathscr{R}_{p_0,q_0}(R_{(t-1)N})m^{((t-1)N)} \\ &+\mathscr{R}_{p_{N-1},q_{N-1}}(R_{tN-1})\cdots\mathscr{R}_{p_1,q_1}(R_{(t-1)N+1})m^{((t-1)N+1)} \\ &+\cdots+\mathscr{R}_{p_{N-1},q_{N-1}}(R_{tN-1})m^{(tN-1)}. \end{aligned} \tag{2.37}$$

From the fact that the spectral norm of any Jacobi annihilator is equal to one (or zero if it is a $2\times 2$ annihilator), the relation (2.37) indicates that

$$\|m^{[tN]}\|_2 \le \|m^{((t-1)N)}\|_2 + \|m^{((t-1)N+1)}\|_2 + \cdots + \|m^{(tN-2)}\|_2 + \|m^{(tN-1)}\|_2, \quad t \ge 1.$$

Thus, from the limit (2.36) we get

$$\lim_{t\to\infty} m^{[tN]} = 0. \tag{2.38}$$

Since $\mathscr{O} \in \mathscr{C}_{sg}^{(n)}$, i.e., the pivot strategy is generalized serial, suppose that $\mathscr{O} \overset{p}{\sim} \mathscr{O}' \overset{w}{\sim} \mathscr{O}''$ or $\mathscr{O} \overset{w}{\sim} \mathscr{O}' \overset{p}{\sim} \mathscr{O}''$, $\mathscr{O}'' \in \mathscr{C}_{sp}^{(n)}$, and that the weak equivalence relation is in the canonical form containing exactly $d$ shift equivalences. For $d+1$ consecutive cycles we get

$$\chi^{((t+d)N)} = \mathscr{J}_{\mathscr{O}}^{[(t+d)N]}\cdots\mathscr{J}_{\mathscr{O}}^{[(t+1)N]}\mathscr{J}_{\mathscr{O}}^{[tN]}\chi^{((t-1)N)} + m_{[d+1]}^{[tN]}, \quad t \ge 1, \tag{2.39}$$

where

$$\begin{aligned} m_{[d+1]}^{[tN]} = &\mathscr{J}_{\mathscr{O}}^{[(t+d)N]}\cdots\mathscr{J}_{\mathscr{O}}^{[(t+1)N]}m^{[tN]} + \mathscr{J}_{\mathscr{O}}^{[(t+d)N]}\cdots\mathscr{J}_{\mathscr{O}}^{[(t+2)N]}m^{[(t+1)N]} \\ &+\cdots+\mathscr{J}_{\mathscr{O}}^{[(t+d)N]}m^{[(t+d-1)N]} + m^{[(t+d)N]}. \end{aligned}$$

Similarly as before, the property of the spectral norm of the Jacobi operator implies

$$\|m_{[d+1]}^{[tN]}\|_2 \le \|m^{[tN]}\|_2 + \|m^{[(t+1)N]}\|_2 + \cdots + \|m^{[(t+d-1)N]}\|_2 + \|m^{[(t+d)N]}\|_2,$$

and using the limit (2.38) we get

$$\lim_{t\to\infty} m_{[d+1]}^{[tN]} = 0.$$

To the Jacobi operators from (2.39) we can apply the Theorem 1.4.5. We get

$$\|\mathscr{J}_{\mathscr{O}}^{[(t+d)N]}\cdots\mathscr{J}_{\mathscr{O}}^{[(t+1)N]}\mathscr{J}_{\mathscr{O}}^{[tN]}\|_2 \le \gamma_n, \quad 0 \le \gamma_n < 1. \tag{2.40}$$

Looking at the spectral norm of (2.39) and using the bound (2.40) we obtain

$$\begin{aligned}\|\chi^{((t+d)N)}\|_2 &\le \|\mathscr{J}_{\mathscr{O}}^{[(t+d)N]}\cdots\mathscr{J}_{\mathscr{O}}^{[(t+1)N]}\mathscr{J}_{\mathscr{O}}^{[tN]}\|_2\|\chi^{((t-1)N)}\|_2+\|m_{[d+1]}^{[tN]}\|_2\\ &\le \gamma_n\|\chi^{((t-1)N)}\|_2+\|m_{[d+1]}^{[tN]}\|_2.\end{aligned}$$

Considering that $0\le\gamma_n<1$ and $\|m_{[d+1]}^{[tN]}\|_2\to 0$, as $t\to\infty$, we employ the Proposition 2.3.1 which yields $\lim_{t\to\infty}\chi^{(tN)}=0$. Therefore, iterations obtained after each cycle converge to zero.

Additionally, for iterations $0<k<N$ within one cycle, from the relation (2.34) we have

$$\begin{aligned}&\chi^{((t-1)N+k)}\\ &\quad=\mathscr{R}_{p_{k-1},q_{k-1}}(R_{(t-1)N+k-1})\cdots\mathscr{R}_{p_1,q_1}(R_{(t-1)N+1})\mathscr{R}_{p_0,q_0}(R_{(t-1)N})\chi^{((t-1)N)}\\ &\quad+\mathscr{R}_{p_{k-1},q_{k-1}}(R_{(t-1)N+k-1})\cdots\mathscr{R}_{p_1,q_1}(R_{(t-1)N+1})\mathscr{R}_{p_0,q_0}(R_{(t-1)N})m^{((t-1)N)}\\ &\quad+\mathscr{R}_{p_{k-1},q_{k-1}}(R_{(t-1)N+k-1})\cdots\mathscr{R}_{p_1,q_1}(R_{(t-1)N+1})m^{((t-1)N+1)}\\ &\quad+\cdots+\mathscr{R}_{p_{k-1},q_{k-1}}(R_{(t-1)N+k-1})m^{((t-1)N+k-1)}.\end{aligned}$$

In the same manner as before we get the inequality

$$\begin{aligned}&\|\chi^{((t-1)N+k)}\|_2\\ &\quad\le\|\chi^{((t-1)N)}\|_2+\|m^{((t-1)N)}\|_2+\|m^{((t-1)N+1)}\|_2+\cdots+\|m^{((t-1)N+k-1)}\|_2\\ &\quad\le\|\chi^{((t-1)N)}\|_2+k\max_{0\le r\le k-1}\|m^{((t-1)N+r)}\|_2.\end{aligned}$$

Thus, $\lim_{t\to\infty}\|\chi^{((t-1)N+k)}\|_2=0$, and it follows

$$\lim_{k\to\infty}\|\chi^{(k)}\|_2=0.$$

Finally, because $\mathrm{off}(H^{(k)})=\|\chi^{(k)}\|_2$, $k\ge 0$, we have $\lim_{k\to\infty}\mathrm{off}(H^{(k)})=0$. ■

Now we can prove the convergence theorem for the Eberlein method under the generalized serial orderings with permutations, $\mathscr{O}\in\mathscr{C}_{sg}^{(n)}$.

**Theorem 2.3.3.** Let $A\in\mathbb{C}^{n\times n}$ and let $(A^{(k)},k\ge 0)$ be a sequence generated by the Eberlein method under a generalized serial pivot strategy defined by an ordering $\mathscr{O}\in\mathscr{C}_{sg}^{(n)}$. Let the matrices $B^{(k)}$ be defined as in (2.6), and the matrices $C(A^{(k)})$ as in (2.7). Then

(i) The sequence of the off-norms $(\mathrm{off}(B^{(k)}), k \geq 0)$ tends to zero,

$$\lim_{k\to\infty} \mathrm{off}(B^{(k)}) = 0.$$

(ii) The sequence $(A^{(k)}, k \geq 0)$ tends to a normal matrix, that is,

$$\lim_{k\to\infty} C(A^{(k)}) = 0.$$

(iii) The sequence of matrices $(B^{(k)}, k \geq 0)$ tends to a fixed diagonal matrix,

$$\lim_{k\to\infty} B^{(k)} = \mathrm{diag}(\mu_1, \mu_2, \ldots, \mu_n),$$

where $\mu_i$, $1 \leq i \leq n$, are real parts of the eigenvalues of $A$.

(iv) If $\mu_i \neq \mu_j$, then $\lim_{k\to\infty} a_{ij}^{(k)} = 0$ and $\lim_{k\to\infty} a_{ji}^{(k)} = 0$.

*Proof.* (i) For $F^{(k)}$ defined as in (2.26) we have

$$B^{(k+1)} = R_k^* B^{(k)} R_k + F^{(k)}, \quad k \geq 0. \tag{2.41}$$

On the pivot position $(p,q)$ in the step $k$ we have

$$b_{pq}^{(k+1)} = \tilde{b}_{pq}^{(k)} + f_{pq}^{(k)},$$

where $F^{(k)} = (f_{ij}^{(k)})$.

Relations (2.27) and (2.23) imply $\lim_{k\to\infty} F^{(k)} = 0$ and $\lim_{k\to\infty} f_{pq}^{(k)} = 0$. Furthermore, the rotation $R_k$ is chosen to annihilate $\tilde{b}_{pq}^{(k)}$. It annihilates $\tilde{b}_{qp}^{(k)}$, as well, because $B^{(k)}$ is Hermitian. Therefore, $\lim_{k\to\infty} b_{pq}^{(k+1)} = 0$ and $\lim_{k\to\infty} b_{qp}^{(k+1)} = 0$. Matrix $B^{(0)} = B$ is Hermitian by the definition and the iterative process (2.41) satisfies the assumptions of the Proposition 2.3.2. Hence,

$$\lim_{k\to\infty} \mathrm{off}(B^{(k)}) = 0.$$

(ii) For $E^{(k)}$ defined as in (2.24) we have

$$C(A^{(k+1)}) = C(\tilde{A}^{(k)} + E^{(k)}).$$

Then,

$$\begin{aligned} C(A^{(k+1)}) &= (\tilde{A}^{(k)}+E^{(k)})(\tilde{A}^{(k)}+E^{(k)})^* - (\tilde{A}^{(k)}+E^{(k)})^*(\tilde{A}^{(k)}+E^{(k)}) \\ &= \tilde{A}^{(k)}(\tilde{A}^{(k)})^* + E^{(k)}(\tilde{A}^{(k)})^* + (\tilde{A}^{(k)}+E^{(k)})(E^{(k)})^* \\ &\quad - (\tilde{A}^{(k)})^*\tilde{A}^{(k)} - (E^{(k)})^*\tilde{A}^{(k)} - (\tilde{A}^{(k)}+E^{(k)})^*E^{(k)} \\ &= C(\tilde{A}^{(k)}) + A^{(k+1)}(E^{(k)})^* - (A^{(k+1)})^*E^{(k)} + E^{(k)}(\tilde{A}^{(k)})^* - (E^{(k)})^*\tilde{A}^{(k)} \\ &= C(\tilde{A}^{(k)}) + W^{(k)}, \end{aligned} \tag{2.42}$$

where

$$W^{(k)} = A^{(k+1)}(E^{(k)})^* - (A^{(k+1)})^*E^{(k)} + E^{(k)}(\tilde{A}^{(k)})^* - (E^{(k)})^*\tilde{A}^{(k)}.$$

Moreover, applying the relation (2.28), we can write (2.42) as

$$C(A^{(k+1)}) = R_k^* C(A^{(k)})R_k + W^{(k)}.$$

Using the properties of the norm and the inequality (2.22) we get

$$\begin{aligned} &\|W^{(k)}\|_F \\ &\quad \le \|A^{(k+1)}(E^{(k)})^*\|_F + \|(A^{(k+1)})^*E^{(k)}\|_F + \|E^{(k)}(\tilde{A}^{(k)})^*\|_F + \|(E^{(k)})^*\tilde{A}^{(k)}\|_F \\ &\quad \le \|A^{(k+1)}\|_F\|E^{(k)}\|_F + \|A^{(k+1)}\|_F\|E^{(k)}\|_F + \|E^{(k)}\|_F\|\tilde{A}^{(k)}\|_F + \|E^{(k)}\|_F\|\tilde{A}^{(k)}\|_F \\ &\quad = 2\|E^{(k)}\|_F(\|A^{(k+1)}\|_F + \|\tilde{A}^{(k)}\|_F) \\ &\quad \le 4\|E^{(k)}\|_F\|\tilde{A}^{(k)}\|_F, \end{aligned}$$

and

$$\|W^{(k)}\|_F^2 \le 16\|E^{(k)}\|_F^2\|\tilde{A}^{(k)}\|_F^2.$$

It follows from the relations (2.24) and (2.25) that

$$\|W^{(k)}\|_F^2 \le 16\|E^{(k)}\|_F^2\|A\|_F^2 \le 24n^2|\tilde{c}_{pq}^{(k)}|\|A\|_F^2.$$

Thus, relation (2.23) implies

$$\lim_{k\to\infty} \|W^{(k)}\|_F = 0. \tag{2.43}$$

We consider the off-diagonal and the diagonal part of $C(A^{(k)})$ separately. Similarly as for matrices $B^{(k)}$, on the pivot position $(p,q)$ in the step $k$ we have

$$c_{pq}^{(k+1)} = \tilde{c}_{pq}^{(k)} + w_{pq}^{(k)},$$

where $W^{(k)} = (w_{ij}^{(k)})$. Relations (2.23) and (2.43) imply $\lim_{k\to\infty} c_{pq}^{(k+1)} = 0$. It is easy to check that matrices $C(A^{(k)})$, $k \geq 0$, are Hermitian. Then $\lim_{k\to\infty} c_{qp}^{(k+1)} = 0$ and we can use the Proposition 2.3.2 again. We get

$$\lim_{k\to\infty} \mathrm{off}(C(A^{(k)})) = 0. \tag{2.44}$$

It remains to show that

$$\lim_{k\to\infty} c_{ii}^{(k)} = 0.$$

Set $A^{(k)} = B^{(k)} + Z^{(k)}$, where $(B^{(k)})$ is Hermitian, as in (2.6), and $Z^{(k)}$ is skew-Hermitian. Then,

$$\begin{aligned} C(A^{(k)}) &= (B^{(k)} + Z^{(k)})(B^{(k)} + Z^{(k)})^* - (B^{(k)} + Z^{(k)})^*(B^{(k)} + Z^{(k)}) \\ &= B^{(k)}B^{(k)^*} + B^{(k)}Z^{(k)^*} + Z^{(k)}B^{(k)^*} + Z^{(k)}Z^{(k)^*} \\ &\quad - B^{(k)^*}B^{(k)} - B^{(k)^*}Z^{(k)} - Z^{(k)^*}B^{(k)} - Z^{(k)^*}Z^{(k)} \\ &= 2(Z^{(k)}B^{(k)} - B^{(k)}Z^{(k)}). \end{aligned} \tag{2.45}$$

The diagonal element of $C(A^{(k)})$ is given by

$$c_{ii}^{(k)} = 2\sum_{j=1}^{n} \left( z_{ij}^{(k)} b_{ji}^{(k)} - b_{ij}^{(k)} z_{ji}^{(k)} \right).$$

It is proven in part $(i)$ that $\lim_{k\to\infty} \mathrm{off}(B^{(k)}) = 0$, that is,

$$\lim_{k\to\infty} b_{ij}^{(k)} = 0, \quad \text{for } i \neq j.$$

Thus,

$$\lim_{k\to\infty} c_{ii}^{(k)} = 2\left( z_{ii}^{(k)} b_{ii}^{(k)} - b_{ii}^{(k)} z_{ii}^{(k)} \right) = 0. \tag{2.46}$$

Relations (2.44) and (2.46) imply the assertion $(ii)$ of the theorem.

(iii) In part $(i)$ of the proof we showed that matrices $B^{(k)}$ tend to a diagonal matrix. The fact that the diagonal elements of the matrix $\lim_{k\to\infty} B^{(k)}$ correspond to the real parts of the eigenvalues of $A$ is then proved as in [67], using the assertion $(ii)$ of this theorem.

(iv) Using the relation (2.45) and parts $(i)$–$(iii)$ of the theorem it follows that

$$\begin{aligned} 0 &= \lim_{k\to\infty} c_{ij}^{(k)} = 2\lim_{k\to\infty}\sum_{r=1}^{n}\left(z_{ir}^{(k)}b_{rj}^{(k)} - b_{ir}^{(k)}z_{rj}^{(k)}\right) \\ &= 2\lim_{k\to\infty}\left(z_{ij}^{(k)}b_{jj}^{(k)} - b_{ii}^{(k)}z_{ij}^{(k)}\right) = 2(\mu_j-\mu_i)\lim_{k\to\infty} z_{ij}^{(k)}, \quad 1\le i,j\le n. \end{aligned}$$

If $\mu_i \neq \mu_j$, then $\lim_{k\to\infty} z_{ij}^{(k)} = 0$. Finally, since $\lim_{k\to\infty} b_{ij}^{(k)} = 0$ for $i \neq j$, we get

$$a_{ij}^{(k)} = b_{ij}^{(k)} + z_{ij}^{(k)} \to 0 \quad \text{and} \quad a_{ji}^{(k)} = (b_{ij}^{(k)})^* - (z_{ij}^{(k)})^* \to 0, \quad \text{as } k\to\infty.$$

■

Therefore, starting with an $n\times n$ matrix $A$, the Eberlein method under a pivot strategy defined by any generalized serial pivot ordering converges to some matrix $\Lambda$. If all real parts of the eigenvalues of $A$ are different, then $\Lambda$ is a diagonal matrix. If the real parts $\mu_i$ and $\mu_j$ of the eigenvalues of $A^{(0)}$ are the same, then we cannot claim that the corresponding off-diagonal elements $a_{ij}^{(k)}$ and $a_{ji}^{(k)}$ tend to zero. This can result with blocks on the diagonal of $\Lambda$. Assuming that the diagonal elements of $\Lambda$ are arranged such that their real parts appear in decreasing order, based on Theorem 2.3.3, we reach the following conclusion. The matrix $\Lambda$ is a block diagonal matrix with block sizes corresponding to the number of times the same real part appears in the spectrum of $A$.

The eigenvalues with different real parts can be read from the diagonal of $\Lambda$. Pairs of complex conjugate eigenvalues with non-repeating real parts, if they create a block, will correspond to $2\times 2$ matrices with $\mathrm{Re}(a_{ii}) = \mu_i$. Such eigenvalues are easy to read from $2\times 2$ blocks. For the repeating real parts, the blocks can be bigger. In our numerical tests, we observed that the blocks appear in case there are complex eigenvalues with the same real but different imaginary parts. The size of such a block corresponds to the number of those eigenvalues with the same real parts. In contrast, repeating real or complex eigenvalues did not create blocks in practice. In order to find all eigenvalues of $A$, it remains to find the eigenvalues of the bigger blocks of $\Lambda$. To that end, for example, the nonsymmetric Jacobi algorithm for the computation of the Schur form discussed in [64] can be applied.

Another approach to find the eigenvalues contained in the blocks of $\Lambda$ is as follows. Let $d$ be a random nonzero complex number. It is easy to check that for any $n\times n$ matrix

$M$ with the spectrum $(\lambda_1, \lambda_2, \ldots, \lambda_n)$, the spectrum of $dM$ is equal to $(d\lambda_1, d\lambda_2, \ldots, d\lambda_n)$. Let us show this. Let $\lambda$ be an eigenvalue of $M$, and let $x$ be a corresponding eigenvector. That is,

$$Mx = \lambda x.$$

We multiply both sides by $d$ and get,

$$dMx = d\lambda x = \lambda dx.$$

Therefore, $d\lambda$ is an eigenvalue of $dM$, with the same corresponding eigenvector $x$.

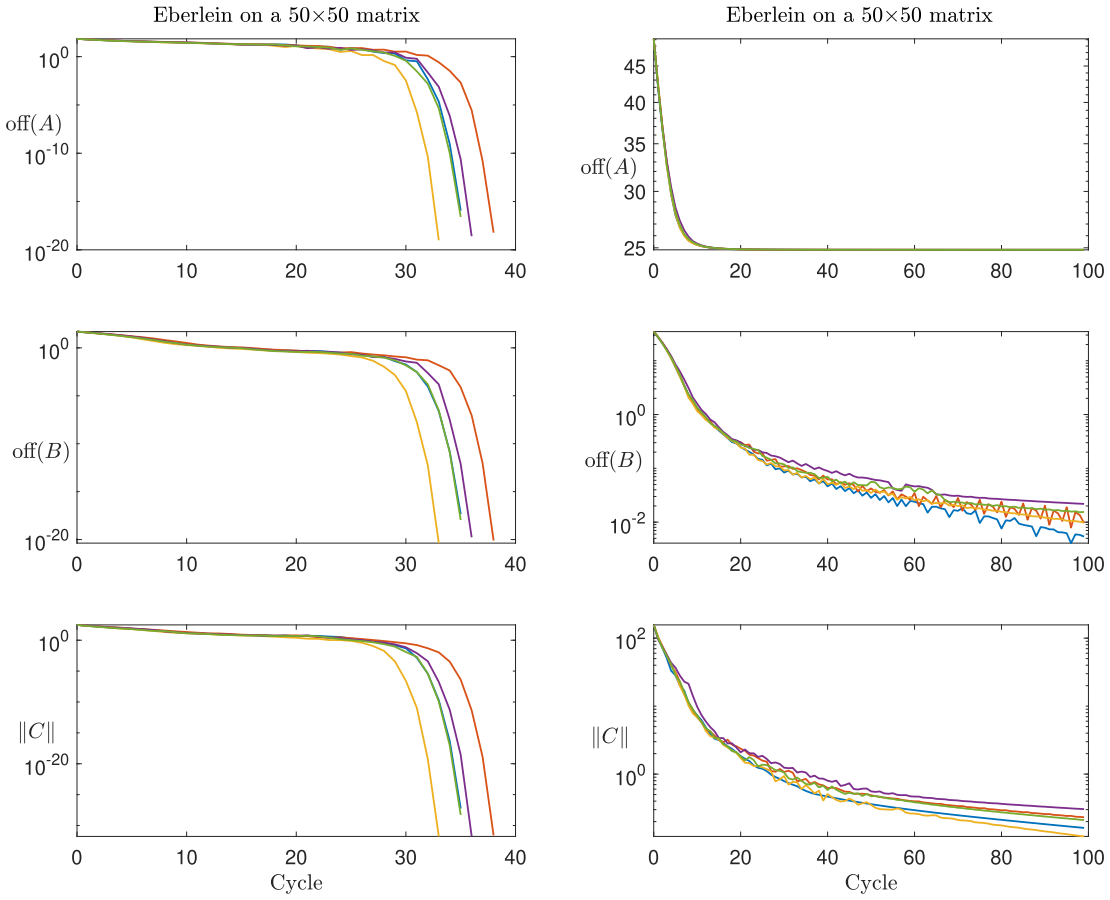
Now we return to our goal of finding the eigenvalues contained in the blocks of $\Lambda$. We take $d$ to be a complex number with nontrivial imaginary part, that is, $\mathrm{Im}(d) \neq 0$. We multiply the obtained block diagonal matrix $\Lambda$ by $d$. The eigenvalues of $d\Lambda$ are equal to the eigenvalues of $\Lambda$ multiplied by $d$. Therefore, if there are complex eigenvalues of $\Lambda$ with the same real but different imaginary parts, they turn into eigenvalues of $d\Lambda$ with different real (and imaginary) parts. That being said, applying the Eberlein method again, this time to $d\Lambda$, yields a diagonal matrix $\Lambda_d$. The algorithm applied to $d\Lambda$ will converge more quickly because the starting matrix is already nearly diagonal. After we get $\Lambda_d$, all eigenvalues of $A$ are found simply by dividing the eigenvalues of $\Lambda_d$ by $d$.

In order to avoid doing the Eberlein method twice, we can do the preconditioning step, scaling by $d$, on the starting matrix $A$. Then we apply the Eberlein method to $dA$ to obtain a diagonal matrix $\Lambda_d$. Again, diagonal elements of $\Lambda_d$ are multiples of the eigenvalues of $A$. Thus, we divide them by $d$ to get the eigenvalues of $A$. We can always do this procedure to bypass the possible diagonal blocks and not concern ourselves with the repeating real parts of the eigenvalues. Therefore, we can assume that the matrix $A$ does not have eigenvalues with repeating real parts, but different imaginary parts. In conclusion, this means that, in practice, the sequence $(A^{(k)},\ k \geq 0)$ will converge to a diagonal matrix carrying the eigenvalues of $A$.

## 2.4. NUMERICAL RESULTS

Numerical tests of the Algorithm 4 under the generalized pivot strategies with permutations are presented in this section. All experiments are done in Matlab R2021a.

To depict the performance of the Eberlein algorithm, we observe three quantities; $\text{off}(A^{(k)})$, $\text{off}(B^{(k)})$, and $\|C(A^{(k)})\|_F$. The results are presented in logarithmic scale. The algorithm is terminated when the change in the off-norm of $B^{(k)}$ becomes small enough, $10^{-8}$. According to Theorem 2.3.3, both $\text{off}(B^{(k)})$ and $\|C(A^{(k)})\|_F$ should converge to zero.




(a) Complex algorithm, random $A \in \mathbb{C}^{50\times 50}$. (b) Real algorithm, random $A \in \mathbb{R}^{50\times 50}$.

Figure 2.1: Change in $\text{off}(A^{(k)})$, $\text{off}(B^{(k)})$ and $\|C(A^{(k)})\|_F$ for different pivot strategies.

In Figure 2.1 the results of the Eberlein algorithm on a non-structured random complex matrix are shown, as well as the results of the real Eberlein algorithm on a non-structured random real matrix. We test the algorithm under different pivot strategies. Each line represents the results of a different pivot strategy $I_{\mathscr{O}}$, $\mathscr{O} \in \mathscr{C}_{sg}^{(n)}$. Strategies are randomly chosen at the beginning of the algorithm. No pivot strategy is superior to others. A strategy that leads to the fastest convergence on one matrix will be slow on a different matrix. We observe that $\text{off}(B^{(k)})$ and $\|C(A^{(k)})\|_F$ converge to zero in both complex and real algorithm, although the convergence is slower for the real algorithm. In the complex case $\text{off}(A)$ converges to zero, as well. That is, the matrix is diagonalized. However, this is not the case for the real algorithm. The reason is that the real algorithm formed the blocks for the eigenvalues with the same real part.

The algorithm is significantly faster if it is applied on a normal matrix, see, for example, [32, 56]. We construct a unitarily diagonalizable (i.e., normal) $400 \times 400$ matrix $A = A^{(0)}$, such that we multiply some chosen complex diagonal matrix from the left- and right-hand side by a random unitary matrix. In Figure 2.2, we see the results of the Eberlein method under a randomly chosen pivot strategy $I_{\mathscr{O}}$, $\mathscr{O} \in \mathscr{C}_{sg}^{(n)}$, applied on a diagonalizable complex matrix. Here we do not show $\|C(A^{(k)})\|_F$ because $A^{(0)}$ is normal, that is $C(A^{(0)}) = 0$, and it stays normal during the process. For this reason, transformations $S_k$ are equal to the identity matrix $I_n$.

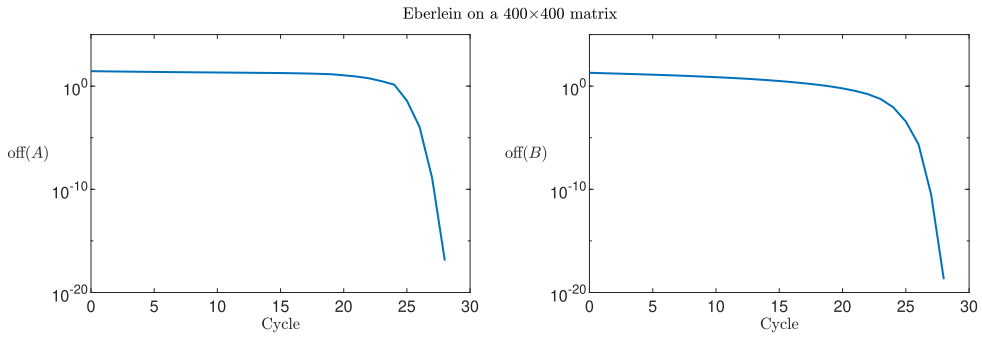



Figure 2.2: Progress of the off-norms of $A^{(k)}$ and $B^{(k)}$ for a unitarily diagonalizable complex matrix.

In order to show the block diagonal structure of $A^{(k)}$ discussed at the end of the previous section, we applied the Eberlein method to the matrices from $\mathbb{C}^{10\times 10}$ and $\mathbb{C}^{50\times 50}$. To generate the starting matrix $A$, we first set the upper triangular matrix $T$ to have the specified diagonal elements. Then we multiply $T$ by a random unitary matrix $Q$, $A = Q^*TQ$. In our implementation of the algorithm, we introduce an additional condition so that the real values of the diagonal elements appear in decreasing order. That is achieved by, if necessary, translating the angle $\alpha_k$ by $\pi/2$ in the $k$th step of the process. The evolution of the matrix structure of the iterates is shown in Figure 2.3. Specifically, the figure shows the logarithm of the absolute values of the elements of $A^{(k)}$. The lighter squares represent the elements that are larger in absolute value. According to Theorem 2.3.3, the algorithm should converge to a block diagonal matrix in both cases described below.

In Figure 2.3a, we have a $10\times 10$ matrix with the spectrum

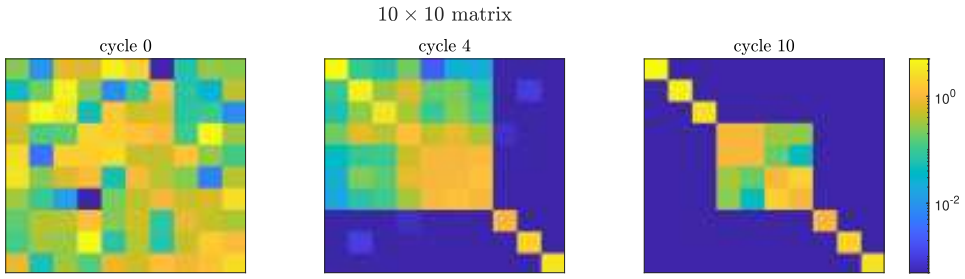
$$\{5,4,3,1\pm 2i,1\pm i,-1,-2,-3\}.$$

Thus, we deal with distinctive eigenvalues and there are two complex conjugate pairs of eigenvalues with the same real part. On the other hand, in Figure 2.3b, we have a $50\times 50$ matrix. Its spectrum consists of two random complex numbers of multiplicity ten and three pairs of complex conjugate complex numbers, each of multiplicity five.

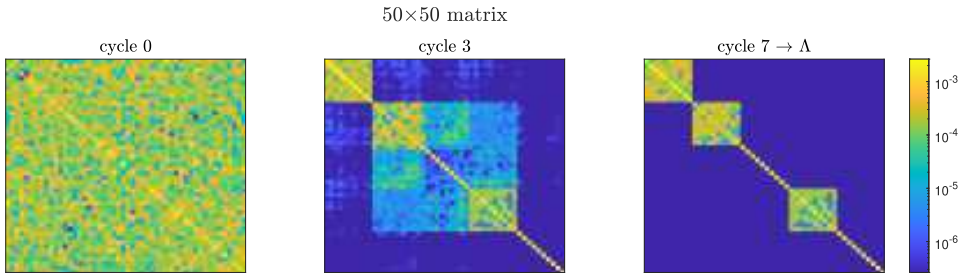
For both matrices, after a few cycles we can faintly see the diagonal blocks. After a few more cycles the block diagonal structure is clear. For the first matrix, the obtained $4\times 4$ block has eigenvalues that are (approximately) $1\pm i$ and $1\pm 2i$. The rest of the diagonal carries the real eigenvalues of the original matrix. Furthermore, for the second matrix we see three blocks that correspond to three pairs of complex conjugate eigenvalues. The rest of the diagonal corresponds to two repeating eigenvalues, and they do not form blocks despite the tenfold multiplicity of each eigenvalue. Compared to the part that formed the blocks, for the repeating eigenvalues there are no other eigenvalues with the same real, but different imaginary part.

To corroborate the discussion at the end of the previous section, for the $50\times 50$ matrix we multiply the obtained block diagonal matrix $\Lambda$ with a complex number $d$ with non-zero imaginary part. In Figure 2.3c we see that running the Eberlein method again on $d\Lambda$ fully diagonalizes the matrix. We point out that we get the same result if we do the

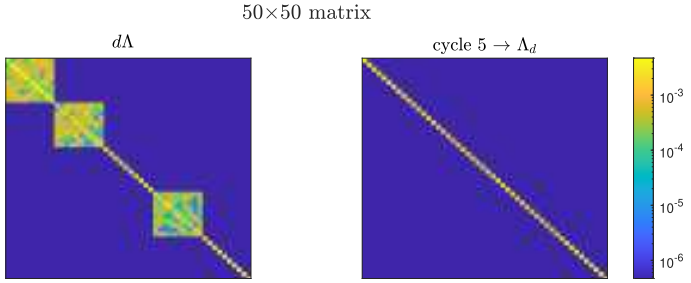
multiplication $dA$ at the beginning and apply the Eberlein method only once to obtain a diagonal matrix $\Lambda_d$. The eigenvalues of the starting matrix are recovered by dividing the values on the diagonal of $\Lambda_d$ by $d$.




(a) Two complex conjugate pairs of eigenvalues with the same real part that formed a $4 \times 4$ diagonal block.




(b) Three complex conjugate eigenvalues formed $10 \times 10$ diagonal blocks, while the rest of the diagonal carries two repeating eigenvalues.




(c) Applying Eberlein method again on $d\Lambda$ fully diagonalized the matrix.

Figure 2.3: Block diagonal structure and solution.

In Figure 2.4 we test the accuracy of the Eberlein method. The top graph demonstrates that the Eberlein method on a random $50 \times 50$ matrix converged to the same solution as the Matlab `eig` function. The bottom graphs show the relative errors in the real and imaginary parts of the obtained eigenvalues with respect to the solutions obtained by the Matlab function `eig`.

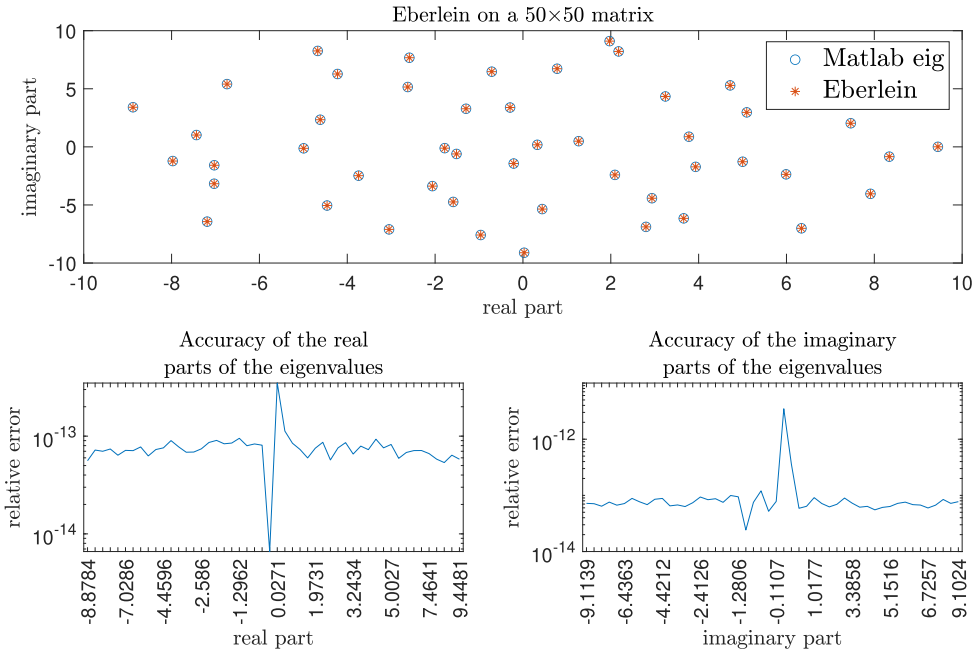



Figure 2.4: Accuracy of the Eberlein method in comparison to the Matlab eig function.

Furthermore, we want to test the accuracy of the eigenvectors generated by the Eberlein algorithm. Because matrices $A^{(k)}$ converge to a matrix $\Lambda$, as $k$ tends to infinity, the sequence of transformations $T_k$ also converges to some non-singular matrix $V$. That is, from the Eberlein algorithm we get

$$A \approx V^{-1} \Lambda V.$$

Columns of $V$ correspond to eigenvectors of $A$. We compare columns of $V$ with the appropriate eigenvectors given by the Matlab function `eig`. We scale both vectors such that their first coordinate is one and then observe their difference component-wise. We can only do this kind of comparison for one-dimensional eigenspaces. In Figure 2.5 we show the accuracy of three randomly chosen eigenvectors for a random $50 \times 50$ matrix.

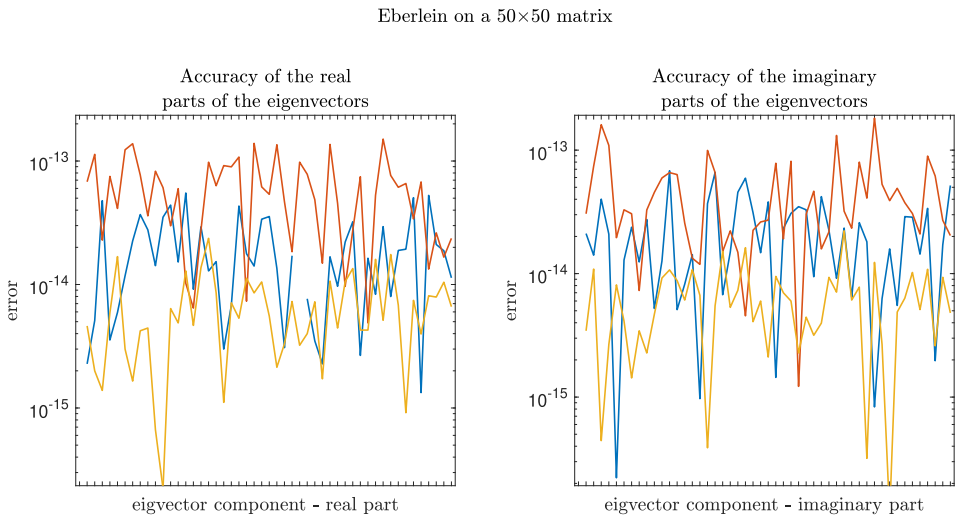



Figure 2.5: Accuracy of the eigenvectors from the Eberlein method in comparison to the Matlab eig function.

In summary, in this section we showed the numerical behaviour of the Eberlein algorithm. The numerical results depict the theoretical results given in Theorem 2.3.3. For $B^{(k)} = \frac{1}{2}(A^{(k)} + (A^{(k)*})$, the sequence $(\text{off}(B^{(k)}), k \geq 0)$ converges to zero, that is, the Hermitian part of $A^{(k)}$ converges to a diagonal matrix. For $C(A^{(k)}) = A^{(k)}(A^{(k)})^* - (A^{(k)})^* A^{(k)}$, the sequence $(C(A^{(k)}), k \geq 0)$ converges to zero, that is, $A^{(k)}$ converges to a normal matrix. Moreover, we showed that if the real parts of the eigenvalues of $A$ are different, then $A^{(k)}$ converges to a diagonal matrix. Otherwise, the blocks corresponding to the repeating eigenvalues may be formed. Regarding the accuracy of the method, we compared it to the Matlab `eig` function, and the results are satisfying.

# 3. BLOCK EBERLEIN DIAGONALIZATION METHOD

In this chapter we propose a new type of the Eberlein method, the block Eberlein method. In general, block algorithms assume a block structure of a given matrix. Contrary to the element-wise algorithms that work on matrix elements, block algorithms work on $n \times n$ blocks of elements at once. Therefore, instead of going through the matrix sequentially element by element, we take sets of elements (blocks) and do the corresponding computation on the entire block. After that we move onto the next block. Scalar operations from the element-wise algorithms are replaced by matrix operations, while zero and one become zero matrix and identity matrix. If we take a block algorithm with $1 \times 1$ blocks, then the block algorithm becomes an element-wise algorithm. On the modern computers, block algorithms are usually more efficient than their element-wise counterparts.

Recall that one step of the element-wise Eberlein method consists of two parts. For the underlying matrix $A$ we first annihilate the pivot element of the matrix $B = (A + A^*)/2$, that is, the Hermitian part of $A$, using a unitary transformation. This corresponds to the diagonalization of a $2 \times 2$ pivot submatrix of $B$. Secondly, we use non-singular complex rotation to reduce the Frobenius norm of $A$. Hence, one step of the block Eberlein algorithm should have two parts, as well. In the first part we are going to diagonalize the pivot block of **B**. Compared to the element-wise case, where this meant annihilating one element of $B$, now we will need to diagonalize a pivot submatrix. This can be done in different ways, for example, by using the complex Jacobi algorithm on the pivot block of the Hermitian matrix **B**. In the second part the goal is to reduce the Frobenius norm of **A** and this is a challenging part of the Block Eberlein algorithm.

First,we give a short introduction to the block matrices and present the block Eberlein

method. Furthermore, we suggest a core algorithm for finding the norm-reducing transformation. Our main result is the convergence theorem that corresponds to the one proved in Chapter 2 for the element-wise case. In particular, we prove that the block Eberlein method converges under the generalized serial block pivot strategies with permutations. Finally, we perform numerical tests for the proposed block method and present the results.

## 3.1. On the block matrices

We first give a short introduction to block matrices. We denote block matrices in boldface capital letters, e.g., **A**, **B**, **C**. Let

$$\pi = (n_1, n_2, \ldots, n_m) \tag{3.1}$$

be an integer partition of $n \in \mathbb{N}$, where $n_i \geq 1$, for all $1 \leq i \leq m$, and $n_1 + n_2 + \cdots + n_m = n$. The partition $\pi$ determines *the block partition* of an $n \times n$ matrix **A**,

$$\mathbf{A} = \begin{bmatrix} A_{11} & A_{12} & \ldots & A_{1m} \\ A_{21} & A_{22} & \ldots & A_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ A_{m1} & A_{m2} & \ldots & A_{mm} \end{bmatrix} \begin{matrix} n_1 \\ n_2 \\ \vdots \\ n_m \end{matrix} . \tag{3.2}$$

Diagonal blocks $A_{ii}$, $1 \leq i \leq m$, are square matrices, while the off-diagonal blocks can be rectangular. Generally, block $A_{ij}$ has dimension $n_i \times n_j$, for all $1 \leq i, j \leq m$. If $\pi = (1, 1, \ldots 1)$, then the block matrix is actually an element-wise matrix. For example, given below is an $8 \times 8$ matrix represented as two block matrices using different partitions, $\pi_1 = (2, 3, 2, 1)$ and $\pi_2 = (1, 3, 4)$, respectively,

$$\left[\begin{array}{cc|ccc|cc|c} * & * & * & * & * & * & * & * \\ * & * & * & * & * & * & * & * \\ \hline * & * & * & * & * & * & * & * \\ * & * & * & * & * & * & * & * \\ * & * & * & * & * & * & * & * \\ \hline * & * & * & * & * & * & * & * \\ * & * & * & * & * & * & * & * \\ \hline * & * & * & * & * & * & * & * \end{array}\right], \qquad \left[\begin{array}{c|ccc|cccc} * & * & * & * & * & * & * & * \\ \hline * & * & * & * & * & * & * & * \\ * & * & * & * & * & * & * & * \\ * & * & * & * & * & * & * & * \\ \hline * & * & * & * & * & * & * & * \\ * & * & * & * & * & * & * & * \\ * & * & * & * & * & * & * & * \\ * & * & * & * & * & * & * & * \end{array}\right].$$

Recall that the block elementary matrix is a matrix differing from the identity only in a $2 \times 2$ submatrix. For a partition $\pi$ and a pivot pair $(p, q)$, the $n \times n$ *block elementary*

*matrix* $\mathbf{E}_{pq}$ differs from the identity matrix in an $(n_p+n_q)\times(n_p+n_q)$ block submatrix. For $p<q$ it takes the form

$$\mathbf{E}_{pq}=\begin{bmatrix} I & & & & \\ & E_{pp} & & E_{pq} & \\ & & I & & \\ & E_{pq} & & E_{qq} & \\ & & & & I \end{bmatrix}\begin{matrix} \\ n_p \\ \\ n_q \\ \\ \end{matrix}.$$

For $p=q$, the block elementary matrix is simply

$$\mathbf{E}_{pp}=\begin{bmatrix} I & & \\ & E_{pp} & \\ & & I \end{bmatrix}\begin{matrix} \\ n_p \\ \\ \end{matrix},$$

but we are going to assume that $p\neq q$. The $(n_p+n_q)\times(n_p+n_q)$ block submatrix $\widehat{\mathbf{E}}_{pq}$,

$$\widehat{\mathbf{E}}_{pq}=\begin{bmatrix} E_{pp} & E_{pq} \\ E_{qp} & E_{qq} \end{bmatrix},$$

is called *the pivot submatrix* of $\mathbf{E}_{pq}$. We denote the function that maps $\widehat{\mathbf{E}}_{pq}$ to an $n\times n$ matrix $\mathbf{E}_{pq}$ by $\mathscr{E}$. We write

$$\mathbf{E}_{pq}=\mathscr{E}(p,q,\widehat{\mathbf{E}}_{pq}).$$

Let us call attention to the effect of multiplying a block matrix by a block elementary matrix on the left and right-hand side. In particular, let $\mathbf{A}$ be a block matrix with block partition $\pi$ as in (3.2), and let $\mathbf{E}_{pq}$ be a block elementary matrix determined by a pivot pair $(p,q)$ and partition $\pi$. Multiplying $\mathbf{A}$ by $\mathbf{E}_{pq}$ from the left-hand side, that is, $\mathbf{E}_{pq}\mathbf{A}$, changes only the $p$th and $q$th block rows of $\mathbf{A}$. The rest of the blocks of $\mathbf{A}$ remain the same. Similarly, multiplication from the right-hand side, $\mathbf{A}\mathbf{E}_{pq}$, alters only the $p$th and $q$th block columns of $\mathbf{A}$. Together, multiplying $\mathbf{E}_{pq}\mathbf{A}\mathbf{E}_{pq}$, changes exclusively the $p$th and $q$th block rows and columns of $\mathbf{A}$, leaving the other blocks intact. This is completely analogous to the element-wise case where a matrix $A$ is multiplied from the left and right-hand side by an elementary matrix.

## 3.2. BLOCK EBERLEIN METHOD

We are going to illustrate our algorithm for the block Eberlein method. Let $\mathbf{A}$ be an arbitrary $n \times n$ block matrix of the form (3.2) with the partition $\pi$ as in (3.1). The block Eberlein method is the iterative process

$$\mathbf{A}^{(k+1)} = \mathbf{T}_k^{-1}\mathbf{A}^{(k)}\mathbf{T}_k, \quad k \geq 0, \tag{3.3}$$

where $\mathbf{A}^{(0)} = \mathbf{A}$, and

$$\mathbf{T}_k = \mathbf{R}_k\mathbf{S}_k, \quad k \geq 0,$$

are non-singular block elementary matrices. The block matrices $\mathbf{T}_k$, and consequently $\mathbf{R}_k$ and $\mathbf{S}_k$, have the same block partition as the matrix $\mathbf{A}^{(k)}$. In our case, the partition $\pi$ is fixed throughout the process, so we omit it in the notation. In a general case, it would be possible to have an adaptive partition that is changing throughout the process. As it was mentioned earlier, if $\pi = (1,1,\ldots,1)$, i.e., all blocks are in fact just elements, the block Eberlein method comes down to the element-wise Eberlein method.

The same way as in the Chapter 2, the process (3.3) can be written with an intermediate step,

$$\mathbf{A}^{(k)} \to \widetilde{\mathbf{A}}^{(k)} \to \mathbf{A}^{(k+1)},$$

where

$$\widetilde{\mathbf{A}}^{(k)} = \mathbf{R}_k^*\mathbf{A}^{(k)}\mathbf{R}_k, \tag{3.4}$$

$$\mathbf{A}^{(k+1)} = \mathbf{S}_k^{-1}\widetilde{\mathbf{A}}^{(k)}\mathbf{S}_k, \quad k \geq 0. \tag{3.5}$$

Let $\mathbf{B}^{(k)}$ be the Hermitian part of $\mathbf{A}^{(k)}$,

$$\mathbf{B}^{(k)} = \frac{1}{2}\left(\mathbf{A}^{(k)} + (\mathbf{A}^{(k)})^*\right). \tag{3.6}$$

Let $(p,q) = (p_k, q_k)$ be the pivot pair in the $k$th step. The pivot submatrix of $\mathbf{A}^{(k)}$ is given by

$$\widehat{\mathbf{A}}_{pq}^{(k)} = \begin{bmatrix} A_{pp}^{(k)} & A_{pq}^{(k)} \\ A_{qp}^{(k)} & A_{qq}^{(k)} \end{bmatrix},$$

while the corresponding submatrix of $\mathbf{B}^{(k)}$ is of the same form,

$$\widehat{\mathbf{B}}_{pq}^{(k)} = \begin{bmatrix} B_{pp}^{(k)} & B_{pq}^{(k)} \\ B_{qp}^{(k)} & B_{qq}^{(k)} \end{bmatrix}.$$

Because $\mathbf{B}^{(k)}$ is Hermitian, the submatrix $\widehat{\mathbf{B}}_{pq}^{(k)}$ is also Hermitian. Thus, $\widehat{\mathbf{B}}_{qp}^{(k)} = \left(\widehat{\mathbf{B}}_{pq}^{(k)}\right)^*$. The $k$th step consists of two parts, the first part corresponding to the relation (3.4) and the second one corresponding to (3.5). In the first part we are looking for the unitary block elementary matrix $\mathbf{R}_k$ that diagonalizes the pivot submatrix $\widehat{\mathbf{B}}_{pq}^{(k)}$. In order to achieve this, we find the unitary $(n_p+n_q)\times(n_p+n_q)$ matrix $\widehat{\mathbf{R}}_{pq}^{(k)}$,

$$\widehat{\mathbf{R}}_{pq}^{(k)} = \begin{bmatrix} R_{pp}^{(k)} & R_{pq}^{(k)} \\ R_{qp}^{(k)} & R_{qq}^{(k)} \end{bmatrix},$$

such that

$$\begin{bmatrix} R_{pp}^{(k)} & R_{pq}^{(k)} \\ R_{qp}^{(k)} & R_{qq}^{(k)} \end{bmatrix}^* \begin{bmatrix} B_{pp}^{(k)} & B_{pq}^{(k)} \\ B_{qp}^{(k)} & B_{qq}^{(k)} \end{bmatrix} \begin{bmatrix} R_{pp}^{(k)} & R_{pq}^{(k)} \\ R_{qp}^{(k)} & R_{qq}^{(k)} \end{bmatrix} = \begin{bmatrix} \Lambda_{pp}^{(k+1)} & 0 \\ 0 & \Lambda_{qq}^{(k+1)} \end{bmatrix},$$

where $\Lambda_{pp}^{(k+1)}$ and $\Lambda_{qq}^{(k+1)}$ are diagonal matrices. Then, we set

$$\mathbf{R}_k = \mathscr{E}(p,q,\widehat{\mathbf{R}}_{pq}^{(k)}).$$

We determine $\widehat{\mathbf{R}}_{pq}^{(k)}$ by applying the complex Jacobi method (see [39]) to the Hermitian matrix $\widehat{\mathbf{B}}_{pq}^{(k)}$. Instead of the Jacobi method, other diagonalization methods can be used, as well. In the second part we need to find the non-singular (and non-unitary) block elementary matrix $\mathbf{S}_k$ that reduces the Frobenius norm of $\widetilde{\mathbf{A}}^{(k)}$. Similar to the first step, we find a non-unitary $(n_p+n_q)\times(n_p+n_q)$ matrix $\widehat{\mathbf{S}}_{pq}^{(k)}$,

$$\widehat{\mathbf{S}}_{pq}^{(k)} = \begin{bmatrix} S_{pp}^{(k)} & S_{pq}^{(k)} \\ S_{qp}^{(k)} & S_{qq}^{(k)} \end{bmatrix},$$

and set

$$\mathbf{S}_k = \mathscr{E}(p,q,\widehat{\mathbf{S}}_{pq}^{(k)}). \tag{3.7}$$

The second part is more difficult than the first part. We are going to describe it in details in Section 3.3. The procedure for the full block Eberlein method is given in Algorithm 5.

**Algorithm 5** Block Eberlein method

**Input:** $\mathbf{A} \in \mathbb{C}^{n\times n}$

**Output:** matrix $\mathbf{A}^{(k)}$, block elementary matrix $\mathbf{T}$

$\mathbf{A}^{(0)} = \mathbf{A}$, $\mathbf{T}_0 = I$

$k = 0$

**repeat**

Choose block pivot pair $(p,q)$ according to the pivot strategy.

Find $\widehat{\mathbf{R}}_{pq}^{(k)}$ which diagonalizes the Hermitian matrix $\widehat{\mathbf{B}}_{pq}^{(k)}$ using complex Jacobi algorithm.

Set $\mathbf{R}_k = \mathscr{E}(p,q,\widehat{\mathbf{R}}_{pq}^{(k)})$.

$\widetilde{\mathbf{A}}^{(k)} = \mathbf{R}_k^* \mathbf{A}^{(k)} \mathbf{R}_k$

Find $\widehat{\mathbf{S}}_{pq}^{(k)}$ which reduces Frobenius norm of $\widetilde{\mathbf{A}}^{(k)}$ using Algorithm 6.

Set $\mathbf{S}_k = \mathscr{E}(p,q,\widehat{\mathbf{S}}_{pq}^{(k)})$.

$\mathbf{A}^{(k+1)} = \mathbf{S}_k^{-1} \widetilde{\mathbf{A}}^{(k)} \mathbf{S}_k$

$\mathbf{T}_{k+1} = \mathbf{T}_k \mathbf{R}_k \mathbf{S}_k$

$k = k+1$

**until** convergence

In the block algorithm pivot pairs refer to blocks. For the partition $\pi = (n_1, n_2, \ldots, n_m)$, the set of all possible pivot pairs is $\mathscr{P}_m = \{(i,j) : 1 \le i < j \le m\}$. A *block pivot strategy* is any function

$$I \colon \mathbb{N}_0 \to \mathscr{P}_m.$$

The same way as we built the set $\mathscr{C}_{sg}^{(n)}$, we build the set of generalized serial block pivot orderings with permutations, $\mathscr{B}_{sg}^{(m)}$, defined in [7]. First, we define the set of column-wise (row-wise) orderings with permutations,

$$\mathscr{B}_c^{(m)} = \Big\{ \mathscr{O} \in \mathscr{O}(\mathscr{P}_m) \mid \mathscr{O} = (1,2), (\tau_3(1),3), (\tau_3(2),3), \ldots,$$

$$\ldots, (\tau_m(1),m), \ldots, (\tau_m(m-1),m), \quad \tau_j \in \Pi^{(1,j-1)}, 3 \le j \le m \Big\},$$

$$\mathscr{B}_r^{(m)} = \Big\{ \mathscr{O} \in \mathscr{O}(\mathscr{P}_m) \mid \mathscr{O} = (m-1,m),(m-2,\tau_{m-2}(m-1)),(m-2,\tau_{m-2}(m)),\dots$$
$$\dots,(1,\tau_1(2)),\dots,(1,\tau_1(m)) \quad \tau_i \in \Pi^{(i+1,m)}, 1 \le i \le m-2 \Big\}.$$

The orderings from $\mathscr{B}_c^{(m)}$ ($\mathscr{B}_r^{(m)}$) go through the matrix block column by block column (block row by block row), starting from the second one (the second to last one). In each block column (block row) pivot elements are chosen in some arbitrary order. Then, using $\mathscr{B}_c^{(n)}$ and $\mathscr{B}_r^{(n)}$ we define two more sets of orderings. They contain orderings reversed to block column-wise and block row-wise orderings with permutations,

$$\overleftarrow{\mathscr{B}}_c^{(m)} = \Big\{ \mathscr{O} \in \mathscr{O}(\mathscr{P}_m) \mid \mathscr{O}^{\leftarrow} \in \mathscr{B}_c^{(m)} \Big\} \quad \text{and} \quad \overleftarrow{\mathscr{B}}_r^{(m)} = \Big\{ \mathscr{O} \in \mathscr{O}(\mathscr{P}_m) \mid \mathscr{O}^{\leftarrow} \in \mathscr{B}_r^{(m)} \Big\}.$$

Together, these four sets of orderings are called *serial block orderings with permutations*,

$$\mathscr{B}_{sp}^{(m)} = \mathscr{B}_c^{(m)} \cup \overleftarrow{\mathscr{B}}_c^{(m)} \cup \mathscr{B}_r^{(m)} \cup \overleftarrow{\mathscr{B}}_r^{(m)}.$$

Finally, we get a very large set of the block pivot orderings if we derive an expansion of $\mathscr{B}_{sp}^{(m)}$ using weak and permutation equivalence relations from the Definition 1.2.2. Let

$$\mathscr{B}_{sg}^{(m)} = \Big\{ \mathscr{O} \in \mathscr{O}(\mathscr{P}_m) \mid \mathscr{O} \overset{w}{\sim} \mathscr{O}' \overset{p}{\sim} \mathscr{O}'' \text{ or } \mathscr{O} \overset{p}{\sim} \mathscr{O}' \overset{w}{\sim} \mathscr{O}'', \mathscr{O}'' \in \mathscr{B}_{sp}^{(m)} \Big\},$$

where $\mathscr{O}' \in \mathscr{O}(\mathscr{P}_m)$. Strategies defined by orderings from $\mathscr{B}_{sg}^{(m)}$ are called *generalized serial block pivot strategies with permutations*. In Section 3.4 we prove the convergence of the block Eberlein method under this broad class of strategies.

## 3.3. CORE ALGORITHM FOR FINDING THE NORM-REDUCING TRANSFORMATION $\mathbf{S}_k$

We now describe the details of computing the block elementary matrix $\mathbf{S}_k$ from the relation (3.5) that reduces the Frobenius norm of the underlying matrix $\widetilde{\mathbf{A}}^{(k)}$ obtained in (3.4).

Recall that multiplying $\mathbf{E}_{pq}\mathbf{A}\mathbf{E}_{pq}$, changes only the $p$th and $q$th block rows and block columns of $\mathbf{A}$, leaving the other blocks intact. That being said, the block elementary matrix $\mathbf{S}_k$ affects the $\widetilde{\mathbf{A}}^{(k)}$ by reducing the Frobenius norm of the pivot block rows and block columns. On the other hand, finding the block submatrix $\widehat{\mathbf{S}}_{pq}^{(k)}$ requires the same block pivot rows and columns. This is in correspondence with the element-wise Eberlein method, where we needed the entire pivot rows and columns to compute the transformation $\widehat{S}_k$ from (2.2). Let us point out that computing the block elementary matrix $\mathbf{R}_k$, or rather finding the block submatrix $\widehat{\mathbf{R}}_{pq}^{(k)}$, requires only the pivot submatrix $\widehat{\mathbf{A}}_{pq}^{(k)}$. This is the reason the second part of the $k$th step of the block Eberlein process (3.3), finding $\mathbf{S}_k$, is more complicated and numerically exhausting.

Here we construct the core algorithm for finding $\mathbf{S}_k$, for some fixed step $k \geq 0$. The input arguments are the block matrix $\widetilde{\mathbf{A}}^{(k)} \in \mathbb{C}^{n\times n}$ with block partition $\pi$ and the pivot pair $(p,q) = (p_k, q_k)$. The goal of reducing the Frobenius norm of $\widetilde{\mathbf{A}}^{(k)}$ can be achieved in more than one way. Our core algorithm is the iterative process

$$\approx^{(l+1)}_{\mathbf{A}} = \mathbf{S}_l^{-1} \approx^{(l)}_{\mathbf{A}} \mathbf{S}_l, \quad l \geq 0, \tag{3.8}$$

where $\approx^{(0)}_{\mathbf{A}} = \widetilde{\mathbf{A}}^{(k)}$, and $\mathbf{S}_l \in \mathbb{C}^{n\times n}$, $l \geq 0$, are block elementary matrices with the same block partition $\pi$. Each transformation $\mathbf{S}_l$ reduces the Frobenius norm of $\approx^{(l)}_{\mathbf{A}}$. We apply them iteratively in order to get as big reduction as possible. Note that, although the relation (3.8) involves $n \times n$ matrices, matrices $\mathbf{S}_l$ differ from the identity only in the pivot submatrix. Then, the pivot submatrix $\widehat{\mathbf{S}}_{pq}^{(k)}$ is computed as a product of the pivot submatrices of $\mathbf{S}_l$. The next iterate $\mathbf{A}^{(k+1)}$ is obtained using (3.7) and (3.5).

Let us describe the construction of $\mathbf{S}_l$, for some fixed $l \geq 0$. The matrix $\mathbf{S}_l$ is computed in the same way as the non-unitary matrix $S_k$ from the element-wise Eberlein method studied in Chapter 2 (see relation (2.2)). It depends on an index pair $(r,s) = (r_l, s_l)$ and

transformation angles $\beta_l$ and $\psi_l$, like in (2.2). It has the following form,

$$\mathbf{S}_l = \begin{bmatrix} 1 & & & & & & & & & & \\ & \ddots & & & & & & & & & \\ & & 1 & & & & & & & & \\ & & & \cos\psi_l & & & & -\imath e^{\imath\beta_l}\sin\psi_l & & & r \\ & & & & 1 & & & & & & \\ & & & & & \ddots & & & & & \\ & & & & & & 1 & & & & \\ & & & \imath e^{-\imath\beta_l}\sin\psi_l & & & & \cos\psi_l & & & s \\ & & & & & & & & 1 & & \\ & & & & & & & & & \ddots & \\ & & & & & & & & & & 1 \end{bmatrix} . \tag{3.9}$$

The transformation angles are calculated using the formulas (2.12) and (2.15). Notice that the latter formula requires all elements from the $r$th and $s$th rows and columns of $\overset{\approx}{\mathbf{A}}^{(l)}$. That is why our iterations (3.8) work on whole matrices instead of the pivot submatrices. Since the outer algorithm is a block algorithm, we focus on the $(p,q)$ pivot submatrix of $\overset{\approx}{\mathbf{A}}^{(l)}$. This means that we choose pairs $(r,s)$ exclusively from the upper triangle of the $(p,q)$ pivot submatrix. To be precise, we have

$$\begin{gathered} n_1+\cdots+n_{p-1}+1 \le r \le n_1+\cdots+n_{p-1}+n_p-1, \\ r < s \le n_1+\cdots+n_{p-1}+n_p, \\ \text{or} \\ n_1+\cdots+n_{q-1}+1 \le r \le n_1+\cdots+n_{q-1}+n_q-1, \\ r < s \le n_1+\cdots+n_{q-1}+n_q, \\ \text{or} \\ n_1+\cdots+n_{p-1}+1 \le r \le n_1+\cdots+n_{p-1}+n_p, \\ n_1+\cdots+n_{q-1}+1 \le s \le n_1+\cdots+n_{q-1}+n_q. \end{gathered} \tag{3.10}$$

Finally, the pivot submatrix $\widehat{\mathbf{S}}_{pq}^{(k)}$ is computed as a product of $\widehat{\mathbf{S}}_{rs}^{(l)}$, $l \ge 0$. The procedure is summarized in Algorithm 6.

In theory, the Algorithm 6 should run until the maximal reduction of the Frobenius norm is achieved. This is numerically challenging as it requires many iterations. However, in practice, there are several other stopping criteria that can be used in Algorithm 6. In our implementation, we choose pairs $(r,s)$ from the set defined by (3.10) according to

**Algorithm 6** Finding $\widehat{\mathbf{S}}_{pq}^{(k)}$

**Input:** $\widetilde{\mathbf{A}}^{(k)} \in \mathbb{C}^{n\times n}$, block pivot pair $(p,q)$

**Output:** $\mathbf{A}^{(k+1)} \in \mathbb{C}^{n\times n}$, $\widehat{\mathbf{S}}_{pq}^{(k)} \in \mathbb{C}^{(n_p+n_q)\times(n_p+n_q)}$

$\widetilde{\widehat{\mathbf{A}}}^{(0)} = \widetilde{\mathbf{A}}^{(k)}$, $\widehat{\mathbf{S}}_{pq}^{(0)} = I_{n_p+n_q}$

$l = 0$

**repeat**

Choose pair $(r,s)$ from (3.10).

Find $(n_p+n_q)\times(n_p+n_q)$ block matrix $\widehat{\mathbf{S}}_{rs}^{(l)}$.

$\mathbf{S}_l = \mathscr{E}(p,q,\widehat{\mathbf{S}}_{rs}^{(l)})$

$\widetilde{\widehat{\mathbf{A}}}^{(l+1)} = \mathbf{S}_l^{-1}\widetilde{\widehat{\mathbf{A}}}^{(l)}\mathbf{S}_l$

$\widehat{\mathbf{S}}_{pq}^{(l+1)} = \widehat{\mathbf{S}}_{pq}^{(l)}\widehat{\mathbf{S}}_{rs}^{(l)}$

$l = l+1$

**until** stopping criterion is satisfied

$\mathbf{A}^{(k+1)} = \widetilde{\widehat{\mathbf{A}}}^{(l)}$, $\widehat{\mathbf{S}}_{pq}^{(k)} = \widehat{\mathbf{S}}_{pq}^{(l)}$

the row-wise ordering in the pivot submatrix $(p,q)$, taking each pair exactly once. Then, the number of iterations in the Algorithm 6 is equal to the number of all possible pairs $(r,s)$, i.e., $l = (n_p+n_q)(n_p+n_q-1)/2$. That is sufficient, in practice, to achieve the convergence of the Algorithm 5, without being too computationally exhausting. Another option would be taking only one pair $(r,s)$ for each pivot submatrix $(p,q)$, i.e., stopping when $l = 1$. For example, for each $(p,q)$ we can randomly choose one pair $(r,s)$. This way block algorithm converges in practice, although much slower, provided that for each block $(p,q)$, all possible pairs $(r,s)$ are chosen enough times.

We are going to illustrate how to find $\widehat{\mathbf{S}}_{pq}^{(k)}$ using the Algorithm 6. Let the pivot submatrix be a $4\times 4$ matrix, and let $(p,q)$ be the given pivot pair. To start, we set $\widehat{\mathbf{S}}_{pq}^{(0)} = I \in \mathbb{C}^{4\times 4}$. Then, we update this matrix by consecutively multiplying it with the computed matrices $\widehat{\mathbf{S}}_{rs}^{(l)}$, for $l = 0,1,\ldots,5$. Index pairs $(r,s)$ are chosen from the upper triangle of the $4\times 4$ pivot submatrix in the row-wise ordering,

$$\begin{bmatrix} * & 0 & 1 & 2 \\ 0 & * & 3 & 4 \\ 1 & 3 & * & 5 \\ 2 & 4 & 5 & * \end{bmatrix}.$$

To simplify the notation, for $l = 1, 2, \ldots, 6$, we donote $\cosh \psi_l$ and $\sinh \psi_l$ by $\mathrm{ch}_l$ and $\mathrm{sh}_l$, respectively. After the first iteration, $l = 1$, we have

$$\widehat{\mathbf{S}}_{pq}^{(1)} = \widehat{\mathbf{S}}_{12}^{(0)} = \begin{bmatrix} \mathrm{ch}_1 & -\imath e^{\imath\beta_1}\,\mathrm{sh}_1 & 0 & 0 \\ \imath e^{-\imath\beta_1}\,\mathrm{sh}_1 & \mathrm{ch}_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Notice that in the first iteration, the elements in the first and second column are the only ones that were transformed. This is the property we mentioned before of multiplying a matrix by an elementary matrix from the right-hand side. Next, we obtain the matrix $\widehat{\mathbf{S}}_{13}^{(1)}$ and multiply it with $\widehat{\mathbf{S}}_{pq}^{(1)}$. The product $\widehat{\mathbf{S}}_{pq}^{(2)} = \widehat{\mathbf{S}}_{pq}^{(1)}\widehat{\mathbf{S}}_{13}^{(1)} = \widehat{\mathbf{S}}_{12}^{(0)}\widehat{\mathbf{S}}_{13}^{(1)}$, takes the form

$$\begin{bmatrix} \mathrm{ch}_1\,\mathrm{ch}_2 & -\imath e^{\imath\beta_1}\,\mathrm{sh}_1 & -\imath e^{\imath\beta_2}\,\mathrm{ch}_1\,\mathrm{sh}_2 & 0 \\ \imath e^{-\imath\beta_1}\,\mathrm{sh}_1\,\mathrm{ch}_2 & \mathrm{ch}_1 & e^{\imath(-\beta_1+\beta_2)}\,\mathrm{sh}_1\,\mathrm{sh}_2 & 0 \\ \imath e^{-\imath\beta_2}\,\mathrm{sh}_2 & 0 & \mathrm{ch}_2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Observe that in this iteration only six elements of the previously computed $\widehat{\mathbf{S}}_{pq}^{(1)}$ were changed, the ones in the first and third column. After the next iteration, $l = 3$, we have

$$\widehat{\mathbf{S}}_{pq}^{(3)} = \widehat{\mathbf{S}}_{12}^{(0)}\widehat{\mathbf{S}}_{13}^{(1)}\widehat{\mathbf{S}}_{14}^{(2)},$$

which is equal to

$$\begin{bmatrix} \mathrm{ch}_1\,\mathrm{ch}_2\,\mathrm{ch}_3 & -\imath e^{\imath\beta_1}\,\mathrm{sh}_1 & -\imath e^{\imath\beta_2}\,\mathrm{ch}_1\,\mathrm{sh}_2 & -\imath e^{\imath\beta_3}\,\mathrm{ch}_1\,\mathrm{ch}_2\,\mathrm{sh}_3 \\ \imath e^{-\imath\beta_1}\,\mathrm{sh}_1\,\mathrm{ch}_2 & \mathrm{ch}_1 & e^{\imath(-\beta_1+\beta_2)}\,\mathrm{sh}_1\,\mathrm{sh}_2 & e^{i(-\beta_1+\beta_3)}\,\mathrm{sh}_1\,\mathrm{ch}_2\,\mathrm{sh}_3 \\ \imath e^{-\imath\beta_2}\,\mathrm{sh}_2\,\mathrm{ch}_3 & 0 & \mathrm{ch}_2 & e^{\imath(-\beta_2+\beta_3)}\,\mathrm{sh}_2\,\mathrm{sh}_3 \\ \imath e^{-\imath\beta_3}\,\mathrm{sh}_3 & 0 & 0 & \mathrm{ch}_3 \end{bmatrix}.$$

Now, all elements in the first and fourth column were transformed. At the end of the cycle, the product of all six submatrices $\widehat{\mathbf{S}}_{12}^{(0)}, \ldots, \widehat{\mathbf{S}}_{34}^{(5)}$, transformed all elements of the starting pivot submatrix $\widehat{\mathbf{S}}_{pq}^{(0)}$. The resulting matrix, for $l = 5$, is

$$\widehat{\mathbf{S}}_{pq}^{(k)} = \widehat{\mathbf{S}}_{pq}^{(5)} = \widehat{\mathbf{S}}_{12}^{(0)}\widehat{\mathbf{S}}_{13}^{(1)}\widehat{\mathbf{S}}_{14}^{(2)}\widehat{\mathbf{S}}_{23}^{(3)}\widehat{\mathbf{S}}_{24}^{(4)}\widehat{\mathbf{S}}_{34}^{(5)},$$

which is given in (3.11).

$$
\begin{bmatrix}
\mathrm{ch}_1\,\mathrm{ch}_2\,\mathrm{ch}_3 & \begin{array}{l}\mathrm{ch}_5\Big(-\imath e^{\imath\beta_1}\,\mathrm{sh}_1\,\mathrm{ch}_4\\ \quad +e^{\imath(\beta_2-\beta_4)}\,\mathrm{ch}_1\,\mathrm{sh}_2\,\mathrm{sh}_4\Big)\\ \quad +e^{\imath(\beta_3-\beta_5)}\,\mathrm{ch}_1\,\mathrm{ch}_2\,\mathrm{sh}_3\,\mathrm{sh}_5\end{array} & \begin{array}{l}\mathrm{ch}_6\Big(-e^{\imath(\beta_1+\beta_4)}\,\mathrm{sh}_1\,\mathrm{sh}_4\\ \quad -\imath e^{\imath\beta_2}\,\mathrm{ch}_1\,\mathrm{sh}_2\,\mathrm{ch}_4\Big)\\ +\imath e^{-\imath\beta_6}\,\mathrm{sh}_6\Big(-e^{\imath(\beta_1+\beta_5)}\,\mathrm{sh}_1\,\mathrm{ch}_4\,\mathrm{sh}_5\\ \quad -\imath e^{\imath(\beta_2-\beta_4+\beta_5)}\,\mathrm{ch}_1\,\mathrm{sh}_2\,\mathrm{sh}_4\,\mathrm{sh}_5\\ \quad -\imath e^{\imath\beta_3}\,\mathrm{ch}_1\,\mathrm{ch}_2\,\mathrm{sh}_3\,\mathrm{ch}_5\Big)\end{array} & \begin{array}{l}-\imath e^{\imath\beta_6}\,\mathrm{sh}_6\Big(-e^{\imath(\beta_1+\beta_4)}\,\mathrm{sh}_1\,\mathrm{sh}_4\\ \quad -\imath e^{\imath\beta_2}\,\mathrm{ch}_1\,\mathrm{sh}_2\,\mathrm{ch}_4\Big)\\ +\mathrm{ch}_6\Big(-e^{\imath(\beta_1+\beta_5)}\,\mathrm{sh}_1\,\mathrm{ch}_4\,\mathrm{sh}_5\\ \quad -\imath e^{\imath(\beta_2-\beta_4+\beta_5)}\,\mathrm{ch}_1\,\mathrm{sh}_2\,\mathrm{sh}_4\,\mathrm{sh}_5\\ \quad -\imath e^{\imath\beta_3}\,\mathrm{ch}_1\,\mathrm{ch}_2\,\mathrm{sh}_3\,\mathrm{ch}_5\Big)\end{array}\\
\imath e^{-\imath\beta_1}\,\mathrm{sh}_1\,\mathrm{ch}_2\,\mathrm{ch}_3 & \begin{array}{l}\mathrm{ch}_1\,\mathrm{ch}_4\,\mathrm{ch}_5\\ +\imath e^{\imath(-\beta_1+\beta_2-\beta_4)}\,\mathrm{sh}_1\,\mathrm{sh}_2\,\mathrm{sh}_4\,\mathrm{ch}_5\\ +\imath e^{\imath(-\beta_1+\beta_3-\beta_5)}\,\mathrm{sh}_1\,\mathrm{ch}_2\,\mathrm{sh}_3\,\mathrm{sh}_5\end{array} & \begin{array}{l}\mathrm{ch}_6\Big(-\imath e^{\imath\beta_4}\,\mathrm{ch}_1\,\mathrm{sh}_4\\ \quad +e^{\imath(\beta_2-\beta_1)}\,\mathrm{sh}_1\,\mathrm{sh}_2\,\mathrm{ch}_4\Big)\\ +\imath e^{-\imath\beta_6}\,\mathrm{sh}_6\Big(-\imath e^{\imath\beta_5}\,\mathrm{ch}_1\,\mathrm{ch}_4\,\mathrm{sh}_5\\ \quad +e^{\imath(-\beta_1+\beta_2-\beta_4+\beta_5)}\,\mathrm{sh}_1\,\mathrm{sh}_2\,\mathrm{sh}_4\,\mathrm{sh}_5\\ \quad +e^{\imath(-\beta_1+\beta_3)}\,\mathrm{sh}_1\,\mathrm{ch}_2\,\mathrm{sh}_3\,\mathrm{ch}_5\Big)\end{array} & \begin{array}{l}-\imath e^{\imath\beta_6}\,\mathrm{sh}_6\Big(-\imath e^{\imath\beta_4}\,\mathrm{ch}_1\,\mathrm{sh}_4\\ \quad +e^{\imath(-\beta_1+\beta_2)}\,\mathrm{sh}_1\,\mathrm{sh}_2\,\mathrm{ch}_4\Big)\\ +\mathrm{ch}_6\Big(-\imath e^{\imath\beta_5}\,\mathrm{ch}_1\,\mathrm{ch}_4\,\mathrm{sh}_5\\ \quad +e^{\imath(-\beta_1+\beta_2-\beta_4+\beta_5)}\,\mathrm{sh}_1\,\mathrm{sh}_2\,\mathrm{sh}_4\,\mathrm{sh}_5\\ \quad +e^{\imath(\beta_3-\beta_1)}\,\mathrm{sh}_1\,\mathrm{ch}_2\,\mathrm{sh}_3\,\mathrm{ch}_5\Big)\end{array}\\
\imath e^{-\imath\beta_2}\,\mathrm{sh}_2\,\mathrm{ch}_3 & \begin{array}{l}\imath e^{-\imath\beta_4}\,\mathrm{ch}_2\,\mathrm{sh}_4\,\mathrm{ch}_5\\ +\imath e^{\imath(-\beta_2+\beta_3-\beta_5)}\,\mathrm{sh}_2\,\mathrm{sh}_3\,\mathrm{sh}_5\end{array} & \begin{array}{l}\mathrm{ch}_2\,\mathrm{ch}_4\,\mathrm{ch}_6\\ +\imath e^{-\imath\beta_6}\,\mathrm{sh}_6\Big(e^{\imath(-\beta_4+\beta_5)}\,\mathrm{ch}_2\,\mathrm{sh}_4\,\mathrm{sh}_5\\ \quad +e^{\imath(-\beta_2+\beta_3)}\,\mathrm{sh}_2\,\mathrm{sh}_3\,\mathrm{ch}_5\Big)\end{array} & \begin{array}{l}-\imath e^{\imath\beta_6}\,\mathrm{ch}_2\,\mathrm{ch}_4\,\mathrm{sh}_6\\ +\mathrm{ch}_6\Big(\imath e^{\imath(-\beta_4+\beta_5)}\,\mathrm{ch}_2\,\mathrm{sh}_4\,\mathrm{sh}_5\\ \quad +e^{\imath(\beta_3-\beta_2)}\,\mathrm{sh}_2\,\mathrm{sh}_3\,\mathrm{ch}_5\Big)\end{array}\\
\imath e^{-\imath\beta_3}\,\mathrm{sh}_3 & \imath e^{-\imath\beta_5}\,\mathrm{ch}_3\,\mathrm{sh}_5 & \imath e^{-\imath\beta_6}\,\mathrm{ch}_3\,\mathrm{ch}_5\,\mathrm{sh}_6 & \mathrm{ch}_3\,\mathrm{ch}_5\,\mathrm{ch}_6
\end{bmatrix}
\quad . \tag{3.11}
$$

# 3.4. CONVERGENCE OF THE BLOCK EBERLEIN METHOD

Recall that in the Algorithm 6, the block elementary matrix $\mathbf{S}_k$ is computed as a product of matrices $\mathbf{S}_l$, $l = 0,\dots,L-1$, where $L = (n_p+n_q)(n_p+n_q-1)/2$, and $(p,q) = (p_k,q_k)$ is the block pivot pair. That is, we stop the algorithm after one sweep of the pivot submatrix. Notice that matrices $\mathbf{S}_l$ also depend on $k$, but we omit it in the notation for the sake of simplicity. The same is true for block matrices $\widetilde{\widetilde{\mathbf{A}}}^{(l)}$ that denote the intermediate steps between $\widetilde{\mathbf{A}}^{(k)}$ and $\mathbf{A}^{(k+1)}$. For a fixed $k$, we have,

$$\mathbf{S}_k = \prod_{l=0}^{L-1} \mathbf{S}_l. \tag{3.12}$$

Therefore,

$$\widetilde{\widetilde{\mathbf{A}}}^{(l+1)} = \mathbf{S}_l^{-1} \widetilde{\widetilde{\mathbf{A}}}^{(l)} \mathbf{S}_l, \quad l = 0,\dots,L-1, \tag{3.13}$$

where $\widetilde{\widetilde{\mathbf{A}}}^{(0)} = \widetilde{\mathbf{A}}^{(k)} = \mathbf{R}_k^* \mathbf{A}^{(k)} \mathbf{R}_k$, and $\mathbf{A}^{(k+1)} = \widetilde{\widetilde{\mathbf{A}}}^{(L)}$. Let $(r_l,s_l)$ be the index pair chosen in the $l$th step. According to (2.14), reduction of the Frobenius norm of $\widetilde{\widetilde{\mathbf{A}}}^{(l)}$ is non-negative, that is, we have

$$\|\widetilde{\widetilde{\mathbf{A}}}^{(l)}\|_F^2 - \|\widetilde{\widetilde{\mathbf{A}}}^{(l+1)}\|_F^2 \geq \frac{1}{3} \frac{|\widetilde{\widetilde{c}}_{r_l s_l}^{(l)}|^2}{\|\mathbf{A}^{(k)}\|_F^2} \geq 0, \tag{3.14}$$

where $C(\widetilde{\widetilde{\mathbf{A}}}^{(l)}) = (\widetilde{\widetilde{c}}_{ij}^{(l)})$.

In the next propositions and the following paragraphs, we show that the assertions analogous to (2.22), (2.23), (2.25), (2.27), and (2.28) are also valid in the block case.

**Proposition 3.4.1.** Let $\mathbf{A}^{(k)}$, $k \geq 0$, be a sequence generated by applying the iterative process (3.3) on a matrix $\mathbf{A}$. Then, for $\|\mathbf{A}^{(k)}\|_F^2$ we have

$$\Delta_k = \|\mathbf{A}^{(k)}\|_F^2 - \|\mathbf{A}^{(k+1)}\|_F^2 \geq 0. \tag{3.15}$$

*Proof.* In the $k$th step of the process (3.3), we observe the reduction of the Frobenius norm for $\mathbf{A}^{(k)}$, which is a sum of the reductions for all $\widetilde{\widetilde{\mathbf{A}}}^{(l)}$, $l \in \{0,1,\dots,L-1\}$. From the inequality (3.14) it follows,

$$\Delta_k = \|\mathbf{A}^{(k)}\|_F^2 - \|\mathbf{A}^{(k+1)}\|_F^2 = \|\widetilde{\mathbf{A}}^{(k)}\|_F^2 - \|\mathbf{A}^{(k+1)}\|_F^2$$
$$= \sum_{l=0}^{L-1} \left( \|\widetilde{\widetilde{\mathbf{A}}}^{(l)}\|_F^2 - \|\widetilde{\widetilde{\mathbf{A}}}^{(l+1)}\|_F^2 \right) \geq \sum_{l=0}^{L-1} \frac{1}{3} \frac{|\widetilde{\widetilde{c}}_{r_l s_l}^{(l)}|^2}{\|\mathbf{A}^{(k)}\|_F^2} \geq 0.$$

■

**Proposition 3.4.2.** Let $\mathbf{A}^{(k)}$, $k \geq 0$, be a sequence generated by applying the iterative process (3.3). We have

$$\lim_{k\to\infty} \frac{\mathrm{off}(\widehat{C}(\widetilde{\mathbf{A}}^{(k)})_{pq})}{\|\mathbf{A}^{(k)}\|_F^2} = 0, \tag{3.16}$$

where $\widehat{C}(\widetilde{\mathbf{A}}^{(k)})_{pq}$ is the pivot submatrix of $C(\widetilde{\mathbf{A}}^{(k)})$.

*Proof.* From the previous proposition we see that the sequence $(\|\mathbf{A}^{(k)}\|_F^2, k \geq 0)$, is non-increasing. Since it is bounded from below, it is convergent. Then, inequality (2.14) implies

$$\lim_{k\to\infty} \sum_{l=0}^{L-1} \frac{|\widetilde{\widetilde{c}}_{r_l s_l}^{(l)}|^2}{\|\mathbf{A}^{(k)}\|_F^2} = 0. \tag{3.17}$$

The notation of the limit in (3.17) makes sense because matrices $\widetilde{\widetilde{\mathbf{A}}}^{(l)}$, and therefore $C(\widetilde{\widetilde{\mathbf{A}}}^{(l)})$, $l = 0, \ldots, L-1$, depend on $k$. The limit (3.17) implies

$$\lim_{k\to\infty} \frac{\widetilde{\widetilde{c}}_{r_l s_l}}{\|\mathbf{A}^{(k)}\|_F^2} = 0, \quad l = 0, \ldots, L-1. \tag{3.18}$$

The matrix $C(\widetilde{\mathbf{A}}^{(k)})$ is Hermitian, and thus, $\lim_{k\to\infty} \widetilde{\widetilde{c}}_{s_l r_l} / \|\mathbf{A}^{(k)}\|_F^2 = 0$, $l = 0, \ldots, L-1$. The assertion (3.16) follows directly from the fact that the index pairs $(r_l, s_l)$, $l = 0, \ldots, L-1$, are chosen from the upper-triangle of the pivot submatrix $\widehat{C}(\widetilde{\mathbf{A}}^{(k)})_{pq}$. ■

**Proposition 3.4.3.** Let $\mathbf{A}^{(k)}$, $k \geq 0$, be a sequence generated by applying the iterative process (3.3). For $\widetilde{\mathbf{A}}^{(k)} = \mathbf{R}_k^* \mathbf{A}^{(k)} \mathbf{R}_k$, $k \geq 0$, and

$$\mathbf{E}^{(k)} = \mathbf{A}^{(k+1)} - \widetilde{\mathbf{A}}^{(k)}, \tag{3.19}$$

we have

$$\|\mathbf{E}^{(k)}\|_F^2 \leq \frac{3}{2} n^2 \sum_{l=0}^{L-1} \frac{|\widetilde{\widetilde{c}}_{r_l s_l}^{(l)}|^2}{\|\mathbf{A}^{(k)}\|_F^2}. \tag{3.20}$$

*Proof.* Similarly as in Chapter 2, we write $\mathbf{E}_k$ as

$$\mathbf{E}^{(k)} = \mathbf{A}^{(k+1)} - \widetilde{\mathbf{A}}^{(k)} = \sum_{l=0}^{L-1} \left( \widetilde{\widetilde{\mathbf{A}}}^{(l+1)} - \widetilde{\widetilde{\mathbf{A}}}^{(l)} \right).$$

From the inequalities (2.24) and (2.25) we get

$$\|\mathbf{E}^{(k)}\|_F^2 \leq \sum_{l=0}^{L-1} \|\widetilde{\widetilde{\mathbf{A}}}^{(l+1)} - \widetilde{\widetilde{\mathbf{A}}}^{(l)}\|_F^2 \leq \sum_{l=0}^{L-1} \frac{3}{2} n^2 \frac{|\widetilde{\widetilde{c}}_{r_l s_l}^{(l)}|^2}{\|\mathbf{A}^{(k)}\|_F^2} = \frac{3}{2} n^2 \sum_{l=0}^{L-1} \frac{|\widetilde{\widetilde{c}}_{r_l s_l}^{(l)}|^2}{\|\mathbf{A}^{(k)}\|_F^2}.$$

■

**Proposition 3.4.4.** Let $\mathbf{A}^{(k)}$, $k \geq 0$, be a sequence generated by applying the iterative process (3.3). For $\widetilde{\mathbf{B}}^{(k)} = \mathbf{R}_k^* \mathbf{B}^{(k)} \mathbf{R}_k$, $k \geq 0$, and

$$\mathbf{F}^{(k)} = \mathbf{B}^{(k+1)} - \widetilde{\mathbf{B}}^{(k)}, \tag{3.21}$$

we have

$$\|\mathbf{F}^{(k)}\|_F^2 \leq \frac{3}{2} n^2 \sum_{l=0}^{L-1} \frac{|\widetilde{\widetilde{c}}_{r_l s_l}^{(l)}|^2}{\|\mathbf{A}^{(k)}\|_F^2}. \tag{3.22}$$

*Proof.* The proof is similar as the proof of Proposition 3.4.3, only instead of inequalities (2.24) and (2.25) we use inequalities (2.26) and (2.27). Then,

$$\|\mathbf{F}^{(k)}\|_F^2 \leq \sum_{l=0}^{L-1} \|\widetilde{\widetilde{\mathbf{B}}}^{(l+1)} - \widetilde{\widetilde{\mathbf{B}}}^{(l)}\|_F^2 \leq \frac{3}{2} n^2 \sum_{l=0}^{L-1} \frac{|\widetilde{\widetilde{c}}_{r_l s_l}^{(l)}|^2}{\|\mathbf{A}^{(k)}\|_F^2}.$$

■

Furthermore, for any $k \geq 0$, we have

$$C(\widetilde{\mathbf{A}}^{(k)}) = \mathbf{R}_k^* C(\mathbf{A}^{(k)}) \mathbf{R}_k. \tag{3.23}$$

In Chapter 2, Proposition 2.3.2 plays a major role in the proof of Theorem 2.3.3, the convergence of the element-wise Eberlein method under the generalized pivot strategies. Specifically, the proposition is used to prove that the off-norms of $B^{(k)}$ and $C(A^{(k)})$ tend to zero. In the block case, this role is going to be fulfilled by Theorem 3.4.5. Before we state this result, we take a step back and observe a general Jacobi-type process

$$\mathbf{A}^{(k+1)} = \mathcal{T}_k^{-1} \mathbf{A}^{(k)} \mathcal{T}_k, \quad k \geq 0, \tag{3.24}$$

where $\mathbf{A}^{(0)} = \mathbf{A}$, and $\mathcal{T}_k$, $k \geq 0$, are block elementary matrices. We give three assumptions on the process (3.24):

**A1** block pivot strategy is defined by an ordering $\mathscr{O} \in \mathscr{B}_{sg}^{(m)}$;

**A2** there exists a sequence of unitary block elementary matrices $\mathbf{U}_k$, $k \geq 0$, such that

$$\lim_{k\to\infty}(\mathcal{T}_k - \mathbf{U}_k) = 0; \tag{3.25}$$

**A3** for the diagonal blocks $U_{p_k p_k}^{(k)}$ and $\sigma^{(k)} = \sigma_{\min}(U_{p_k p_k}^{(k)})$ we have

$$\sigma = \liminf_{k\to\infty} \sigma^{(k)} > 0,$$

where $\sigma_{\min}(X)$ is the smallest singular value of matrix $X$.

The assumption **A1** determines the strategies that we work with: generalized serial block pivot strategies with permutations. If $\mathcal{T}_k$ are unitary transformations, then the assumption **A2** is trivially true. On the other hand, in the block Eberlein process, matrices $\mathcal{T}_k = \mathbf{T}_k$ are non-unitary. Therefore, we need to prove that the assumption **A2** holds in the block Eberlein case. Regarding the assumption **A3**, recall that the necessary condition for the convergence of the Jacobi method was that the cosine of all the transformation angles $\varphi_k$ are bounded from below by some strictly positive constant [31]. The assumption **A3** is a generalization of this condition for the block elementary matrices.

The next theorem refers to the convergence of the iterative process (3.26). Like we mentioned before, we are going to use it in order to prove the convergence of the block Eberlein algorithm.

**Theorem 3.4.5.** Let $\mathbf{H}$ be an $n \times n$ matrix with the partition $\pi = (n_1, \ldots, n_m)$. Let $\mathbf{H}^{(k)}$, $k \geq 0$, be a sequence generated by applying the iterative process

$$\mathbf{H}^{(k+1)} = \mathbf{U}_k^* \mathbf{H}^{(k)} \mathbf{U}_k + \mathbf{M}^{(k)}, \quad \mathbf{H}^{(0)} = \mathbf{H}, \quad k \geq 0. \tag{3.26}$$

If the assumptions **A1** and **A3** are true, then the following relations are equivalent,

(i)
$$\lim_{k\to\infty} \frac{\operatorname{off}(\widehat{\mathbf{A}}_{p_k q_k}^{(k+1)})}{\|\mathbf{A}^{(k)}\|_F} = 0,$$

(ii)
$$\lim_{k\to\infty} \frac{\operatorname{off}(\mathbf{A}^{(k)})}{\|\mathbf{A}^{(k)}\|_F} = 0.$$

*Proof.* The proof is essentially the same as the proof of Theorem 7.1. from [7]. ■

Now, let us focus back to the block Eberlein method, which is a special case of the general Jacobi-type process. Before we use the results from Theorem 3.4.5 in order to prove the convergence of our block method, we need to show that the assumptions **A2** and **A3** are satisfied for the process (3.3).

Matrices $\mathbf{S}_l$ are chosen to reduce the Frobenius norm of $\overset{\approx}{\mathbf{A}}^{(l)}$. Consequently, matrices $\mathbf{S}_k$ reduce the Frobenius norm of $\mathbf{A}^{(k)}$. As it was said in the proof of the Proposition 3.4.2, the sequence $\|\mathbf{A}^{(k)}\|_F$, $k \geq 0$, is clearly non-increasing and bounded from below by zero. Hence, it is convergent. It follows that, as the process progresses, the elementary matrices $\mathbf{S}_k$ have a smaller and smaller influence on the norm of $\mathbf{A}^{(k)}$. Precisely, in the relation (3.12) matrices $\mathbf{S}_l$, $l = 0,\ldots,L-1$, are of the form (3.9), where the transformation angles are calculated using (2.15). Then, the limit (3.18) implies that,

$$\lim_{k\to\infty} \tanh \psi_l = 0, \quad \text{for} \quad l = 0,\ldots,L-1,$$

which indicates

$$\lim_{k\to\infty} \sinh \psi_l = 0, \quad \text{and} \quad \lim_{k\to\infty} \cosh \psi_l = 1, \quad \text{for} \quad l = 0,\ldots,L-1,$$

because we take positive value for the hyperbolic cosine. If follows that for each $l = 0,\ldots,L-1$, we have that $\mathbf{S}_l$ tends to identity, as $k \to \infty$. Then, relation (3.12) implies that the matrices $S_k$ converge to $I_n$, as well. Therefore, the sequence $\mathbf{T}_k = \mathbf{R}_k\mathbf{S}_k$, $k \geq 0$, tends to $\mathbf{U}_k = \mathbf{R}_k$, $k \geq 0$. Since $\mathbf{R}_k$ are unitary matrices, the assumption **A2** is true for the block Eberlein process.

It is left to show that the assumption **A3** is true as well. The assumption **A3** is true if the matrices $\mathbf{U}_k$ are the so-called UBC transformations defined in [25].

**Definition 3.4.6.** A class of unitary transformations with a given $2 \times 2$ block partition is called a class of *UBC (Uniformly Bounded Cosine) transformations*, if the singular values of the diagonal blocks can be bounded from below by a function of the dimension.

Furthermore, in [25] Drmač proved that for every unitary $n \times n$ matrix $\mathbf{U}$ and for every partition $\pi = (n_1, n_2)$, $n_1 + n_2 = n$, there exists a permutation matrix $P$ such that for the leading $n_1 \times n_1$ block of $\mathbf{U}' = \mathbf{U}P$ we have

$$\sigma_{\min}(\mathbf{U}'_{11}) \geq \gamma_\pi > 0,$$

where $\gamma_\pi$ is a constant depending only on $n_1$ and $n_2$. Hari [38] showed that the strictly positive lower bound depends only on the dimension $n$.

In the block Eberlein process, multiplying the unitary transformation $\mathbf{R}_k$ in (3.4) with a permutation matrix $P_k$ such that $\mathbf{R}_k P_k$ is a UBC transformation will not change off$(\widetilde{\mathbf{A}}^{(k)})$, nor will it disturb zeros in the matrix $\widetilde{\mathbf{B}}^{(k)} = \mathbf{R}_k^* \mathbf{B}^{(k)} \mathbf{R}_k$. Therefore, without the loss of generality, we can use UBC matrices in (3.4). We have showed that the sequence of transformations $\mathbf{T}_k$ in the block Eberlein process tends to the unitary matrices of the form $\mathbf{U}_k = \mathbf{R}_k$. Hence, as $\mathbf{R}_k$ are UBC matrices, the assumption **A3** is satisfied for the block Eberlein process (3.3).

We are now ready to prove the main result of this chapter, a generalization of the Theorem 2.3.3 to the block case. We look at the convergence of the block Eberlein method under the strategies determined by $\mathscr{B}_{sg}^{(m)}$, thus, the assumption **A1** is satisfied. Moreover, we demonstrated that the process (3.3) satisfies **A2** and **A3**, as well. That is, we showed that the block Eberlein process is associated with a general Jacobi-type process (3.24).

**Theorem 3.4.7.** Let $\mathbf{A} \in \mathbb{C}^{n\times n}$ be a block matrix with partition $\pi = (n_1, \ldots, n_m)$ as in (3.2), and let $(\mathbf{A}^{(k)}, k \geq 0)$ be a sequence generated by the block Eberlein method under a generalized serial pivot strategy defined by an ordering $\mathscr{O} \in \mathscr{B}_{sg}^{(m)}$. Let the matrices $\mathbf{B}^{(k)}$ be defined as in (3.6), and the matrices $C(\mathbf{A}^{(k)})$ as in (2.7). Then

(i) The sequence of the off-norms $(\text{off}(\mathbf{B}^{(k)}), k \geq 0)$ tends to zero,

$$\lim_{k\to\infty} \text{off}(\mathbf{B}^{(k)}) = 0.$$

(ii) The sequence $(\mathbf{A}^{(k)}, k \geq 0)$ tends to a normal matrix, that is,

$$\lim_{k\to\infty} C(\mathbf{A}^{(k)}) = 0.$$

(iii) The sequence of matrices $(\mathbf{B}^{(k)}, k \geq 0)$ tends to a fixed diagonal matrix,

$$\lim_{k\to\infty} \mathbf{B}^{(k)} = \text{diag}(\mu_1, \mu_2, \ldots, \mu_n),$$

where $\mu_i$, $1 \leq i \leq n$, are real parts of the eigenvalues of $A$.

(iv) If $\mu_i \neq \mu_j$, then $\lim_{k\to\infty} a_{ij}^{(k)} = 0$ and $\lim_{k\to\infty} a_{ji}^{(k)} = 0$.

*Proof.* The proof follows the proof of the Theorem 2.3.3.

(i) For $\mathbf{F}^{(k)}$ defined as in (3.21) we have

$$\mathbf{B}^{(k+1)} = \mathbf{R}_k^* \mathbf{B}^{(k)} \mathbf{R}_k + \mathbf{F}^{(k)}, \quad k \geq 0. \tag{3.27}$$

For the pivot submatrix determined by the pivot pair $(p,q) = (p_k, q_k)$, we have

$$\widehat{\mathbf{B}}_{pq}^{(k+1)} = \widehat{\widetilde{\mathbf{B}}}_{pq}^{(k)} + \widehat{\mathbf{F}}_{pq}^{(k)},$$

where $\widehat{\widetilde{\mathbf{B}}}_{pq}^{(k)}$ is the pivot submatrix of $\widetilde{\mathbf{B}}^{(k)}$.

Relations (3.22) and (3.17) imply $\lim_{k\to\infty} \mathbf{F}^{(k)} = 0$ and $\lim_{k\to\infty} \widehat{\mathbf{F}}_{pq}^{(k)} = 0$. Furthermore, the rotation $\mathbf{R}_k$ is chosen to diagonalize $\widehat{\widetilde{\mathbf{B}}}_{pq}^{(k)}$. Therefore,

$$\lim_{k\to\infty} \operatorname{off}(\widehat{\mathbf{B}}_{pq}^{(k+1)}) = 0.$$

Because Frobenius norm of $\mathbf{A}^{(k)}$, $k \geq 0$, converges, the same is true for the sequence $\|\mathbf{B}^{(k)}\|_F$, and we have

$$\lim_{k\to\infty} \frac{\operatorname{off}(\widehat{\mathbf{B}}_{pq}^{(k+1)})}{\|\mathbf{B}^{(k)}\|_F} = 0.$$

The assumptions **A1** and **A3** are satisfied for the block Eberlein process. The iterative process (3.27), therefore, satisfies the assumptions of Theorem 3.4.5. Hence,

$$\lim_{k\to\infty} \frac{\operatorname{off}(\mathbf{B}^{(k)})}{\|\mathbf{B}^{(k)}\|_F} = 0.$$

The limit

$$\lim_{k\to\infty} \operatorname{off}(\mathbf{B}^{(k)}) = 0,$$

is true, as well, because $\|\mathbf{B}^{(k)}\|_F$, $k \geq 0$ converges.

(ii) For $\mathbf{E}^{(k)}$ defined as in (3.19) we have

$$C(\mathbf{A}^{(k+1)}) = C(\widetilde{\mathbf{A}}^{(k)} + \mathbf{E}^{(k)}).$$

Then, in the same manner as in (2.42),

$$C(\mathbf{A}^{(k+1)}) = C(\widetilde{\mathbf{A}}^{(k)}) + \mathbf{W}^{(k)}, \tag{3.28}$$

where

$$\mathbf{W}^{(k)} = \mathbf{A}^{(k+1)}(\mathbf{E}^{(k)})^* - (\mathbf{A}^{(k+1)})^* \mathbf{E}^{(k)} + \mathbf{E}^{(k)}(\tilde{\mathbf{A}}^{(k)})^* - (\mathbf{E}^{(k)})^* \widetilde{\mathbf{A}}^{(k)}.$$

Using the properties of the norm and the inequality (3.15), in the same way as we did in the proof of the Theorem 2.3.3 (ii), we get

$$\|\mathbf{W}^{(k)}\|_F \leq 4\|\mathbf{E}^{(k)}\|_F \|\widetilde{\mathbf{A}}^{(k)}\|_F,$$

and

$$\|\mathbf{W}^{(k)}\|_F^2 \leq 16\|\mathbf{E}^{(k)}\|_F^2 \|\widetilde{\mathbf{A}}^{(k)}\|_F^2.$$

It follows from the relations (3.19) and (3.20) that

$$\|\mathbf{W}^{(k)}\|_F^2 \leq 16\|\mathbf{E}^{(k)}\|_F^2 \|\mathbf{A}\|_F^2 \leq 24n^2 \sum_{l=0}^{L-1} \frac{|\widetilde{\widetilde{c}}_{r_l s_l}^{(l)}|^2}{\|\mathbf{A}^{(k)}\|_F^2} \|\mathbf{A}\|_F^2.$$

Thus, relation (3.17) implies

$$\lim_{k\to\infty} \|\mathbf{W}^{(k)}\|_F = 0. \tag{3.29}$$

Next, we consider the off-diagonal and the diagonal part of $C(\mathbf{A}^{(k)})$ separately. Applying the relation (3.23), we can write (3.28) as

$$C(\mathbf{A}^{(k+1)}) = \mathbf{R}_k^* C(\mathbf{A}^{(k)}) \mathbf{R}_k + \mathbf{W}^{(k)}. \tag{3.30}$$

Hence, similarly as for the matrices $\mathbf{B}^{(k)}$, for the pivot submatrix in the step $k$ we have

$$\widehat{C}(\mathbf{A}^{(k+1)})_{pq} = \widehat{C}(\widetilde{\mathbf{A}}^{(k)})_{pq} + \widehat{\mathbf{W}}_{pq}^{(k)},$$

where $\widehat{C}(\widetilde{\mathbf{A}}^{(k)})_{pq}$ and $\widehat{C}(\mathbf{A}^{(k+1)})_{pq}$ are pivot submatrices of $C(\widetilde{\mathbf{A}}^{(k)})$ and $C(\mathbf{A}^{(k+1)})$, respectively. Relations (3.16) and (3.29) imply $\lim_{k\to\infty} \mathrm{off}(\widehat{C}(\mathbf{A}^{(k+1)})_{pq}) = 0$. Because the sequence $\|\mathbf{A}^{(k)}\|_F$, $k \geq 0$, is convergent, so is the sequence $\|C(\mathbf{A}^{(k)})\|_F$, $k \geq 0$, and we have

$$\lim_{k\to\infty} \frac{\mathrm{off}(\widehat{C}(\mathbf{A}^{(k+1)})_{pq})}{\|C(\mathbf{A}^{(k)})\|_F} = 0.$$

We can once again use Theorem 3.4.5, this time on the sequence $C(\mathbf{A}^{(k)})$, $k \geq 0$. We get

$$\lim_{k\to\infty} \frac{\mathrm{off}(C(\mathbf{A}^{(k+1)})_{pq})}{\|C(\mathbf{A}^{(k)})\|_F} = 0,$$

and

$$\lim_{k\to\infty} \mathrm{off}(C(\mathbf{A}^{(k)})) = 0. \tag{3.31}$$

It remains to show that, for the diagonal elements of $C(\mathbf{A}^{(k)})$, we have

$$\lim_{k\to\infty} c_{ii}^{(k)} = 0, \quad i = 1,\ldots,n.$$

Set $\mathbf{A}^{(k)} = \mathbf{B}^{(k)} + \mathbf{Z}^{(k)}$, where $(\mathbf{B}^{(k)})$ is Hermitian, as in (3.6), and $\mathbf{Z}^{(k)}$ is skew-Hermitian part of $\mathbf{A}^{(k)}$. Repeating the same calculation as in the relation (2.45), we obtain

$$C(\mathbf{A}^{(k)}) = 2(\mathbf{Z}^{(k)}\mathbf{B}^{(k)} - \mathbf{B}^{(k)}\mathbf{Z}^{(k)}). \tag{3.32}$$

The diagonal element of $C(\mathbf{A}^{(k)})$ is then given by

$$c_{ii}^{(k)} = 2\sum_{j=1}^{n} \left(z_{ij}^{(k)} b_{ji}^{(k)} - b_{ij}^{(k)} z_{ji}^{(k)}\right).$$

It is proven in part $(i)$ that $\lim_{k\to\infty} \mathrm{off}(\mathbf{B}^{(k)}) = 0$, that is,

$$\lim_{k\to\infty} b_{ij}^{(k)} = 0, \quad \text{for } i \neq j.$$

Thus,

$$\lim_{k\to\infty} c_{ii}^{(k)} = 2\left(z_{ii}^{(k)} b_{ii}^{(k)} - b_{ii}^{(k)} z_{ii}^{(k)}\right) = 0. \tag{3.33}$$

Now, relations (3.31) and (3.33) imply the assertion (ii) of the theorem.

(iii) In part (i) of the proof we showed that matrices $\mathbf{B}^{(k)}$ tend to a diagonal matrix. The fact that the diagonal elements of the matrix $\lim_{k\to\infty} \mathbf{B}^{(k)}$ correspond to the real parts of the eigenvalues of $\mathbf{A}$ is then proved as in [67], using the assertion (ii) of this theorem.

(iv) This part is proved as the corresponding part (iv) of Theorem 2.3.3, but instead of the relation (2.45) we use the relation (3.32) for block matrices.

■

Let us recapitulate. Starting with an $n \times n$ complex block matrix $\mathbf{A}^{(0)} = \mathbf{A}$ with the partition $\pi = (n_1,\ldots,n_m)$, the sequence of block matrices $(\mathbf{A}^{(k)}, k \geq 0)$generated by the block Eberlein method under any generalized serial block pivot strategy converges to a normal matrix $\Lambda$. The Hermitian parts of $\mathbf{A}^{(k)}$ converge to a diagonal matrix with the real parts of the eigenvalues of $\mathbf{A}$, $\mu_i$, $i = 1,\ldots n$, on the diagonal. In addition, if two

eigenvalues of $\mathbf{A}$ have different real parts, that is, if $\mu_i \neq \mu_j$, then the corresponding off-diagonal elements $a_{ij}^{(k)}$ and $a_{ji}^{(k)}$ tend to zero. Consequently, if all the eigenvalues of $\mathbf{A}$ have different real parts, $\Lambda$ is a diagonal matrix. On the other hand, if the real parts of two or more eigenvalues are equal, then the matching off-diagonal elements might not vanish, resulting in non-trivial off-diagonal blocks in $\Lambda$. These off-diagonal blocks do not necessarily match the partition $\pi$. They can stretch across and/or be divided in several blocks determined by $\pi$. This complication can be solved by preconditioning the matrix $\mathbf{A}$.

Same as in the element-wise case, we can precondition the starting matrix $\mathbf{A}$ by scaling it with $d \in \mathbb{C}$, where $\mathrm{Im}(d) \neq 0$. Then, we apply the block Eberlein method to $d\mathbf{A}$ which, with probability one, does not have eigenvalues with the same real and different imaginary parts. This process results with a diagonal matrix $\Lambda_d$. Diagonal elements of $\Lambda_d$ are multiples of the eigenvalues of $\mathbf{A}$. Finally, we simply divide them by $d$ to get the eigenvalues of $\mathbf{A}$.

## 3.5. NUMERICAL RESULTS

Numerical tests of the Algorithm 5 under the row-wise pivot strategy are presented in this section. All experiments were performed in Matlab R2021a.

To depict the performance of the block Eberlein algorithm, we observe three quantities; off($\mathbf{A}^{(k)}$), off($\mathbf{B}^{(k)}$), and $\|C(\mathbf{A}^{(k)})\|_F$, and how they change after each cycle, just like we did in Chapter 2 for the element-wise algorithm. The results are presented in logarithmic scale. The algorithm is terminated when the change in the off-norm of $\mathbf{B}^{(k)}$ becomes small enough, $10^{-8}$. Acording to the Theorem 3.4.7, we expect both off($\mathbf{B}^{(k)}$) and $\|C(\mathbf{A}^{(k)})\|_F$ to converge to zero, while the convergence of off($\mathbf{A}^{(k)}$) depends on the eigenvalues of the starting matrix $\mathbf{A}$. If $\mathbf{A}$ is a normal matrix, then it remains normal throughout the process. Thus, there is no need to observe $\|C(\mathbf{A}^{(k)})\|_F$ for normal matrices. For simplicity, we take the partition $\pi = (n_1, n_2, \ldots, n_m)$ to have all blocks of the same size, $n_1 = n_2 = \ldots = n_m$. Obviously, the size of the blocks depends on the number of blocks, $m$. We test the algorithm for different block sizes, one, two, five, and ten. Each line in the figures represents the results for a different block size. Using the blocks of size one should come down to the element-wise Eberlein method.

In order to show the block structure of $\mathbf{A}^{(k)}$ discussed at the end of the previous section, we will apply the Eberlein method on matrices with repeating real parts of the eigenvalues. Additionally, we are going to test the accuracy of the block Eberlein method for different block sizes and compare it to the element-wise Eberlein. We are going to show the relative errors in the real and imaginary parts of values obtained on the diagonal of $A^{(k)}$, regarding the eigenvalues obtained by the Matlab `eig` function.

Let us first present the test matrices.

1. **TestMatrix1**: Matrix $\mathbf{A}_1 \in \mathbb{C}^{n\times n}$ is constructed as a random matrix:
   - `A_1=randn(n)+1i*randn(n);`
2. **TestMatrix2**: Matrix $\mathbf{A}_2 \in \mathbb{C}^{n\times n}$ is constructed as a random matrix with the desired condition number $c$.
   - `U=orth(randn(n)+1i*randn(n)); V=orth(randn(n)+1i*randn(n));`

- `s=randn(n,1)+1i*randn(n,1);` ($s$ is a vector)
- `s = s(1)*( 1-((c-1)/c)*(s(1)-s)/(s(1)-s(end)));` (linear stretch of existing $s$)
- `A_2 = U*diag(s)*V';`

3. **TestMatrix3**: Matrix $\mathbf{A}_3 \in \mathbb{C}^{n\times n}$ is constructed as an ill-conditioned matrix with fast decaying eigenvalues.

   - $\Sigma = \mathrm{diag}((1+\imath)^{-d}, (2+2\imath)^{-d}, \ldots, (n+n\imath)^{-d})$;
   - `Q=orth(rand(n)+1i*rand(n));`
   - `A_3=Q*Σ*Q';`

   Matrix $\mathbf{A}_3$ is a normal matrix, meaning that $C(\mathbf{A}_3) = \mathbf{A}_3\mathbf{A}_3^* - \mathbf{A}_3^*\mathbf{A}_3 = 0$.

4. **TestMatrix4**: The spectrum of $\mathbf{A}_4 \in \mathbb{C}^{n\times n}$ consists of a random complex number and a pair of complex conjugate numbers, with multiplicities $m_1$ and $m_2$, respectively. The multiplicities of the eigenvalues add up to $n$, that is, $m_1 + 2m_2 = n$.

   - `a_1=rand(1)+1i*rand(1); a_2=rand(1)+1i*rand(1);`
   - `a=[repelem(a_1,m_1),repelem([a_2,a_2'],m_2);`
   - $\Sigma =$ `diag(a);`
   - `Q=orth(rand(n)+1i*rand(n));`
   - `A_4=Q*Σ*Q';`

   Preconditioning step:

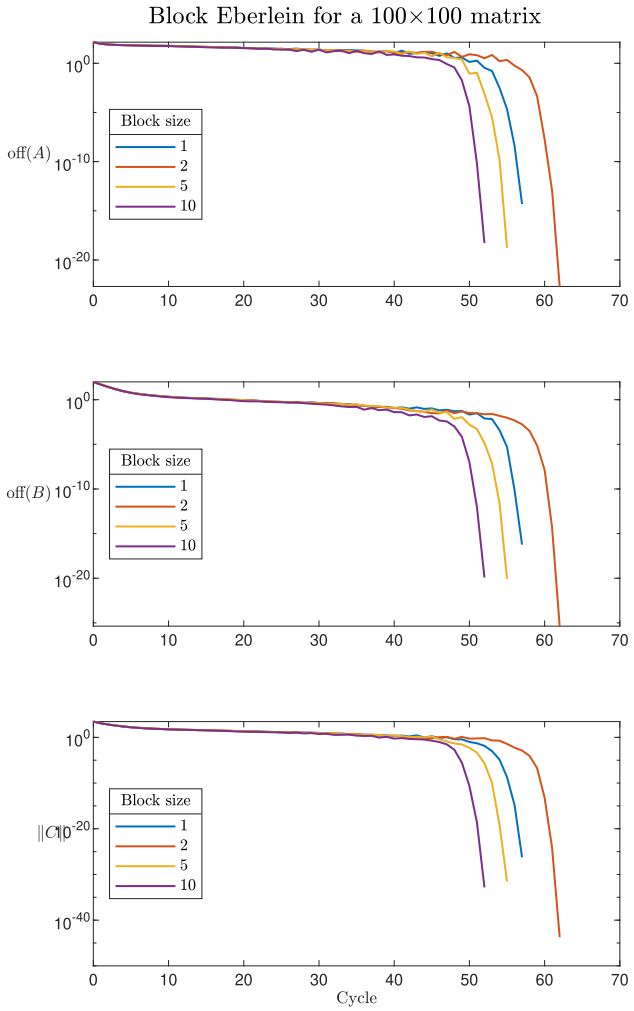
   - `d=rand(1)+1i*rand(1);`
   - `dA_4=d*A_4;`

   Matrices $\mathbf{A}_4$ and $d\mathbf{A}_4$ are also normal.

We analyze the test matrices and present the results in the following subsections.

### 3.5.1. TestMatrix1

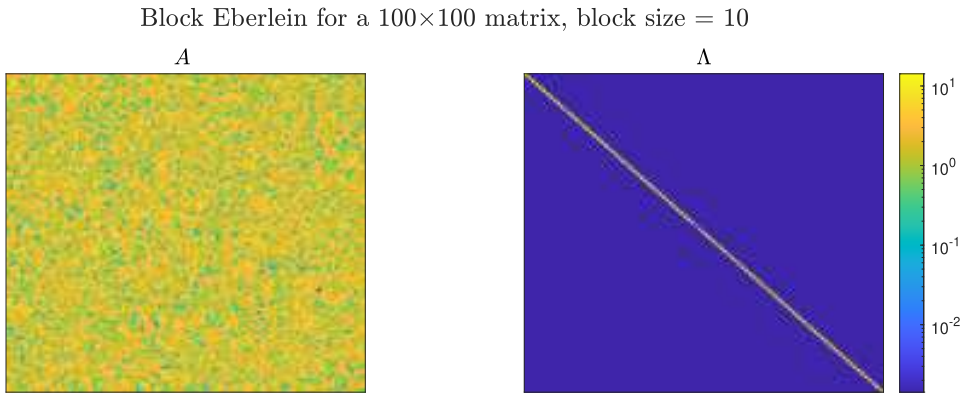
We consider TestMatrix1 for $n = 100$. Figure 3.1 shows the results of the block Eberlein method applied to the matrix $\mathbf{A}_1$. Generically, random matrices are not normal and have different eigenvalues. We expect that off$(\mathbf{B}^{(k)})$ and $\|C(\mathbf{A}^{(k)})\|$, $k \geq 0$, converge to zero for all block sizes. However, the larger the blocks, the less cycles are needed for these values to converge. This observation repeats itself in almost all of the following examples. Furthermore, because all eigenvalues are simple, we expect off$(\mathbf{A}^{(k)})$, $k \geq 0$. In fact, Figure 3.1a confirms our prediction.




(a) Change in off$(\mathbf{A}^{(k)})$, off$(\mathbf{B}^{(k)})$ and $\|C(\mathbf{A}^{(k)})\|_F$ for different sized blocks.

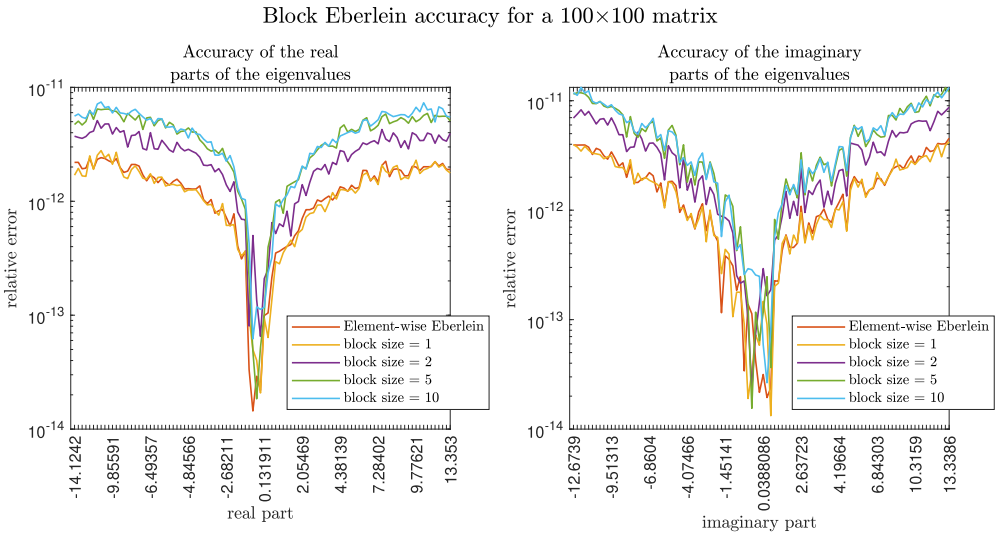
Figure 3.1: Results for **TestMatrix1**, for $n = 100$.

In Figure 3.1b we see the structure for the starting matrix $\mathbf{A}_1$ and the matrix $\Lambda$ obtained by the block Eberlein algorithm, for a block size equal to five. In particular, the figure shows the logarithm of the absolute values of the elements of $\mathbf{A}^{(k)}$. Lighter shaded squares represent the elements larger in absolute value. The obtained matrix $\Lambda$ is diagonal, and $\Lambda$ should carry eigenvalues of $\mathbf{A}_1$ on the diagonal.




(b) The starting matrix $\mathbf{A}_1$ and fully diagonal matrix $\Lambda$ obtained from the block Eberlein algorithm.

In Figure 3.1c we can see the accuracy of the block Eberlein method in comparison to the Matlab `eig` function. More precisely, as $\Lambda$ is diagonal, it carries approximations of the eigenvalues of the starting matrix. That is, we see that the relative errors in both real and imaginary parts of the obtained eigenvalues are around $10^{-12}$.

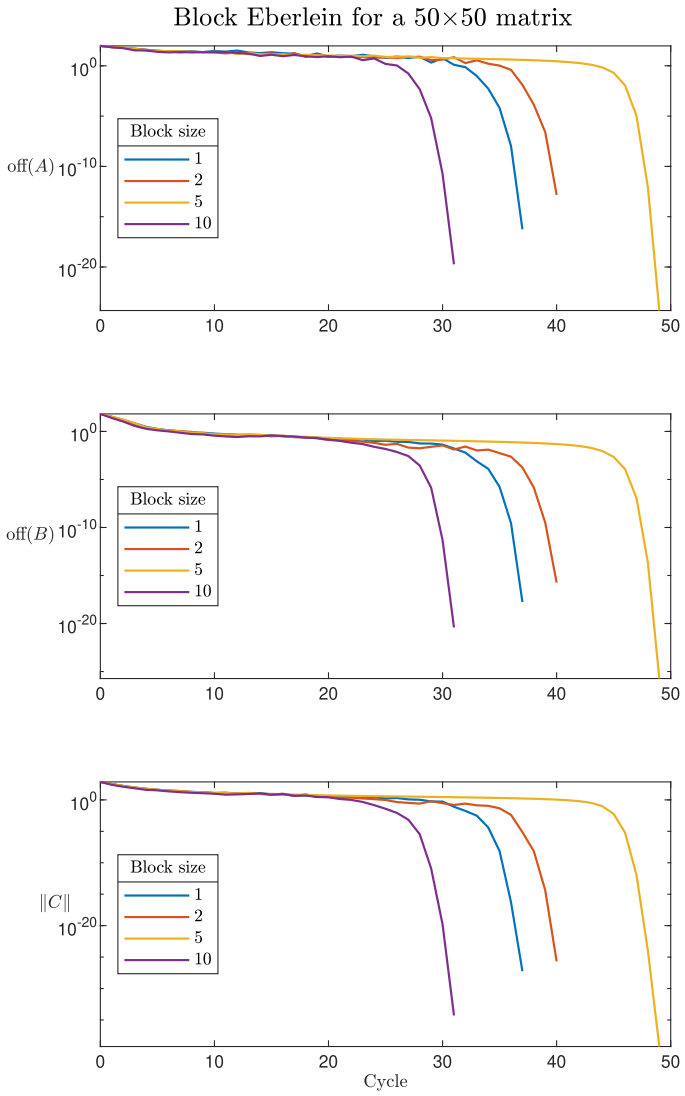



(c) Accuracy of the block Eberlein method in comparison to the Matlab eig function.

Figure 3.1: Results for **TestMatrix1**, for $n = 100$.

### 3.5.2. TestMatrix2

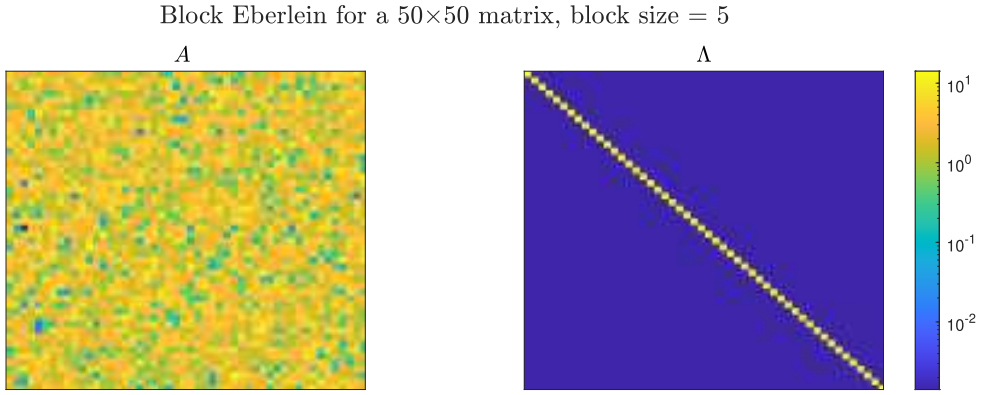
This example is constructed to demonstrate how does the block Eberlein method works on random well-conditioned matrices. In Figure 3.2 we show the results of the Algorithm 5 on a $50 \times 50$ non-structured random complex matrix $\mathbf{A}_2$, with low condition number, $c = 2$. Figure 3.2a shows that $\text{off}(\mathbf{A}^{(k)})$, $\text{off}(\mathbf{B}^{(k)})$ and $\|\mathbf{C}^{(k)}\|_F$, $k \geq 0$ converge to zero for all block sizes.




(a) Change in $\text{off}(\mathbf{A}^{(k)})$, $\text{off}(\mathbf{B}^{(k)})$ and $\|C(\mathbf{A}^{(k)})\|_F$ for different sized blocks.

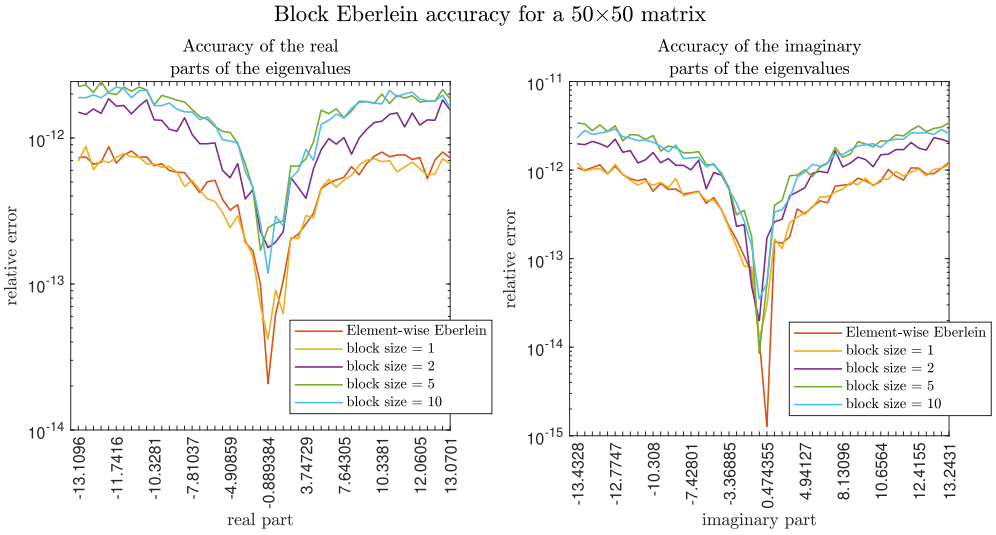
Figure 3.2: Results for **TestMatrix2**, for $n = 50$ and $c = 2$.

In Figure 3.2b we see the structure of the starting matrix $\mathbf{A}_2$ and the matrix $\Lambda$ obtained by the Algorithm 5, for a block size of five. This partcular matrix $\Lambda$ is diagonal.




(b) The starting matrix $\mathbf{A}_2$ and fully diagonal matrix $\Lambda$ obtained from the block Eberlein algorithm.

In Figure 3.2c we can see the accuracy of Algorithm 5 in comparison to the Matlab `eig` function. Again, we see that the relative errors in real and imaginary parts of the obtained eigenvalues are around $10^{-12}$.




(c) Accuracy of the block Eberlein method in comparison to the Matlab eig function.

Figure 3.2: Results for **TestMatrix2**, for $n = 50$ and $c = 2$.

In addition to the accuracy of the eigenvalues, we want to test the accuracy of the eigenvectors generated by the block Eberlein algorithm. We already showed that the transformations $T_k$ converge to block rotations $\mathbf{R}_k$. More than that, from the fact that matrices $\mathbf{A}^{(k)}$ converge to a matrix $\Lambda$, the sequence of $T_k$ also converges to some non-singular matrix $V$. Similar as in the element-wise case, from the block Eberlein algorithm we get

$$\mathbf{A} \approx V^{-1} \Lambda V.$$

Columns of $V$ correspond to eigenvectors of $\mathbf{A}$. Again, we compare columns of $V$ with the appropriate eigenvectors given by the Matlab function `eig`. First, we scale both vectors such that their first coordinate is one and then observe their difference component-wise. Let us point out that we can only do this kind of comparison for one-dimensional eigenspaces. All eigenvalues of matrix $A_2$ are different, and therefore all its eigenspaces are one-dimensional. In Figure 2.5 we show the accuracy of three randomly chosen eigenvectors for matrix $A_2$.

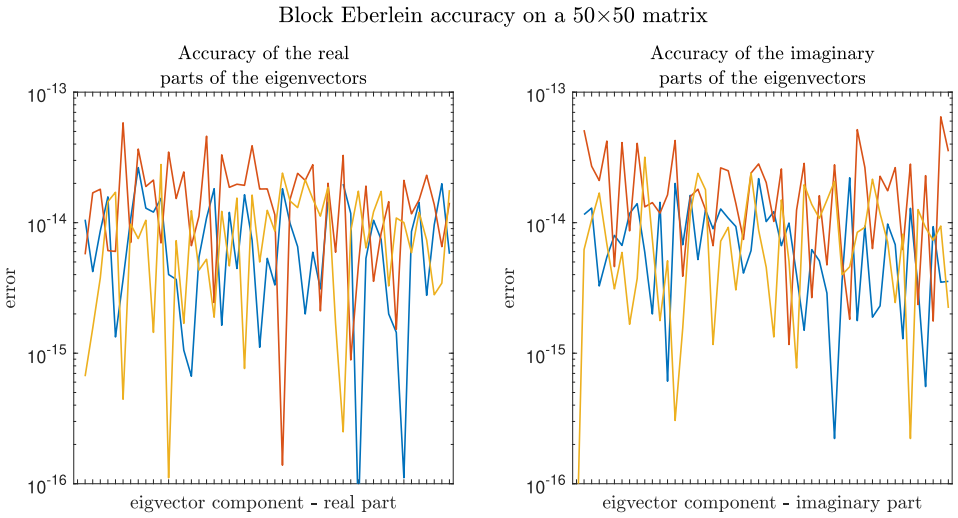
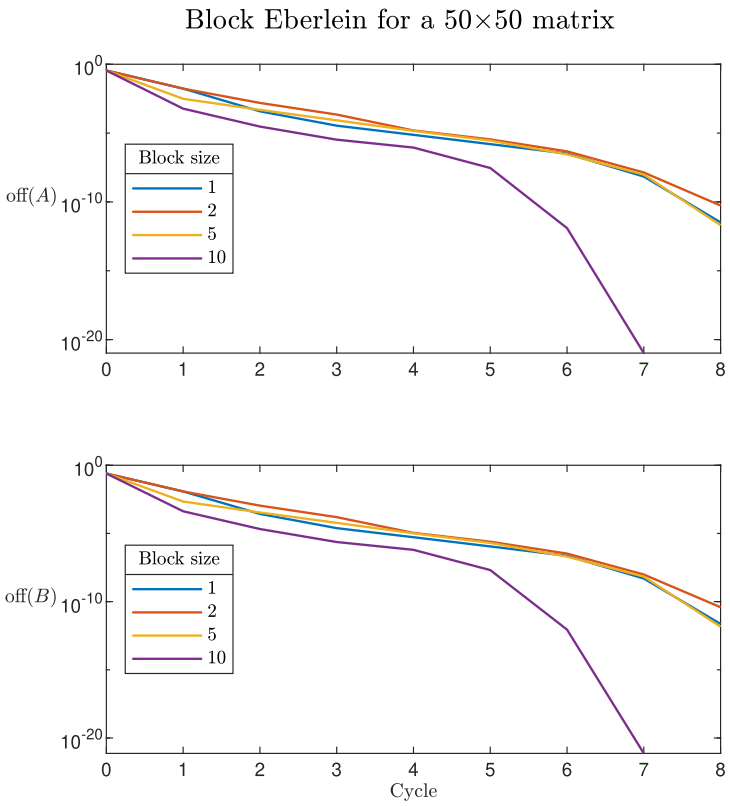



Figure 3.3: Accuracy of the eigenvectors from the Eberlein method in comparison to the Matlab eig function.

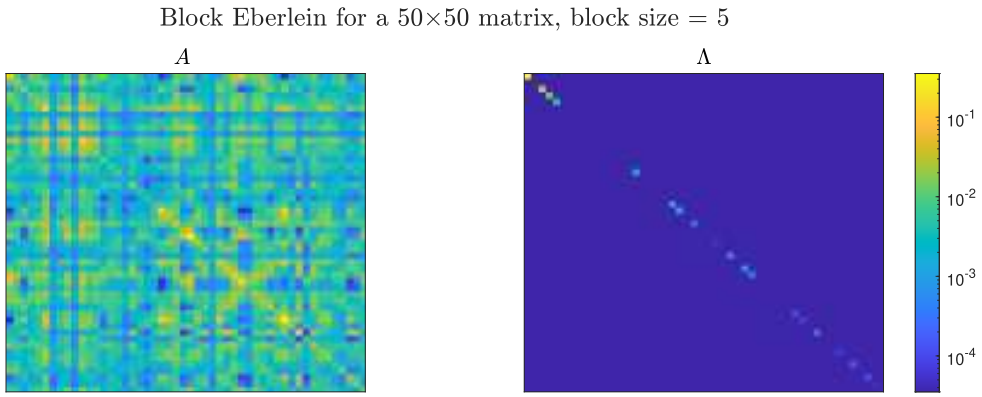
### 3.5.3. TestMatrix3

For the TestMatrix3, we consider $n = 50$ and $d = 3$. The constructed matrix $\mathbf{A}_3$ is a normal matrix. Hence, we do not observe the Frobenius norms of $C(\mathbf{A}^{(k)})$, since all $C(\mathbf{A}^{(k)})$, $k \geq 0$,are zero matrices. Again, both off$(\mathbf{A}^{(k)})$ and off$(\mathbf{B}^{(k)})$, $k \geq 0$, converge to zero for all block sizes (see Figure 3.4a). As expected, the block Eberlein algorithm converged to a fully diagonalized matrix $\Lambda$ (see Figure 3.4b). Although the condition number of $\mathbf{A}_3$ (cond$(\mathbf{A}_3) = 1.25 \cdot 10^5$) is significantly larger than the condition number of $\mathbf{A}_2$ (cond$(\mathbf{A}_2) = 2$), the accuracy for the TestMatrix3 is not inferior (see Figure 3.4c).

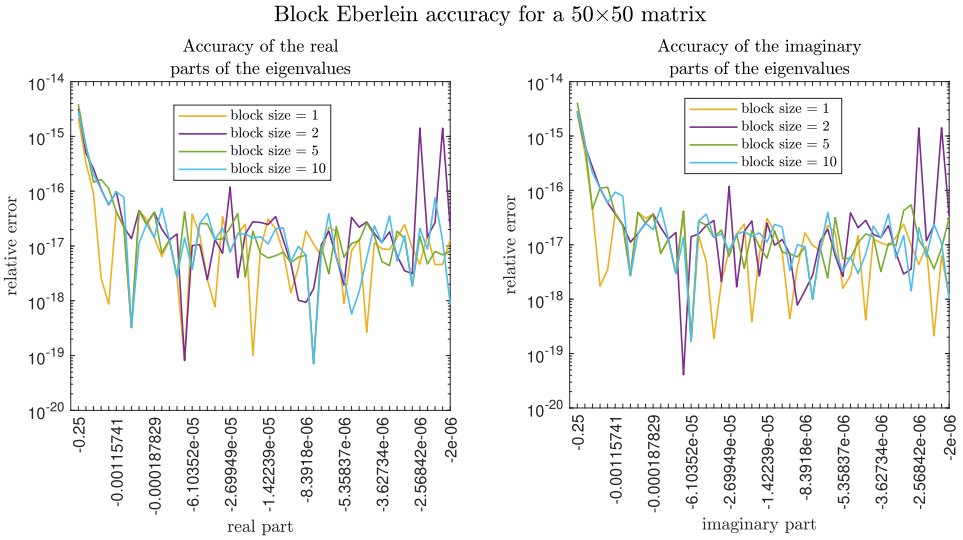



(a) Change in off$(\mathbf{A}^{(k)})$ and off$(\mathbf{B}^{(k)})$ for different sized blocks.

Figure 3.4: Results for **TestMatrix3**, for $n = 50$ and $d = 3$.




(b) The starting matrix $\mathbf{A}_3$ and fully diagonal matrix $\Lambda$ obtained from the block Eberlein algorithm.




(c) Accuracy of the block Eberlein method in comparison to the Matlab eig function.

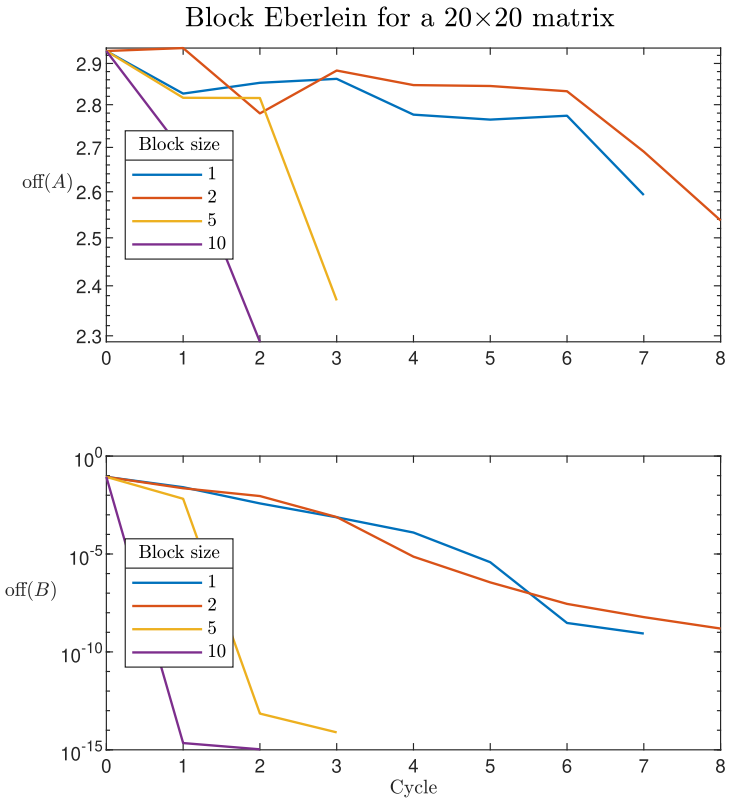
Figure 3.4: Results for **TestMatrix3**, for $n = 50$ and $d = 3$.

### 3.5.4. TestMatrix4

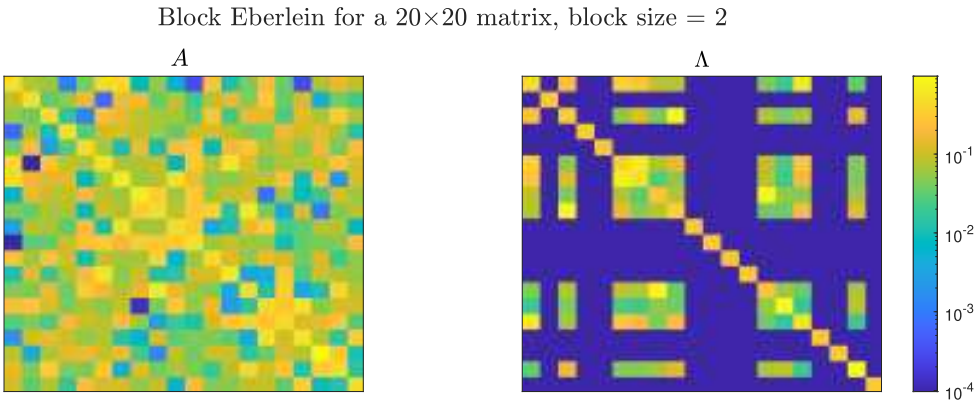
This example is constructed to demonstrate that the block diagonal structure of $\Lambda$, discussed in the element-wise Eberlein in Chapter 2, also turns up in the block Eberlein case. Recall that, in the element-wise Eberlein algorithm, in Figure 2.3, we introduced an additional condition so that the real values of the diagonal elements appear in decreasing order. Due to that and Theorem 2.3.3, the algorithm converged to a block diagonal matrix. Specifically, it is because obtained diagonal elements corresponding to the eigenvalues with the same real part were arranged successively, and the matching off-diagonal elements that had not converged to zero formed a block. In contrast to that, in the block case, this condition would not imply a block diagonal matrix. The reason is that the eigenvalues with the same real part may be located in more than one block. If we could assume that all the eigenvalues sharing the same real part are located inside the same block, then the resulting matrix would be block diagonal. Still, this is not something that is assumed. Thus, the non-zero off-diagonal elements need not to appear only in the diagonal blocks, but rather in the off-diagonal blocks.

We consider TestMatrix4 for $n = 20$, $m_1 = 10$, $m_2 = 5$. The spectrum of $\mathbf{A}_4$ consists of a random complex number $a_1$ of multiplicity ten, and a pair of complex conjugate numbers, $a_2$ and $a_2^*$, each of multiplicity five. That is, the real part of $a_1$ appears 10 times in the spectrum of $\mathbf{A}_4$, while the corresponding imaginary parts are all equal. Furthermore, the real part of $a_2$ (and $a_2^*$) also appears 10 times in the spectrum of $\mathbf{A}_4$. The imaginary parts of $a_2$ and $a_2^*$ are different and each appears five times in the spectrum of $\mathbf{A}_4$.

In Figures 3.5a and 3.5b we see that by applying the block Eberlein method on $\mathbf{A}_4$, the off-norm $\text{off}(\mathbf{A}^{(k)})$, $k \geq 0$, does not converge to zero and we obtain a block matrix, but not diagonal matrix, $\Lambda$. Matrix $\Lambda$ is nearly diagonal. The non-zero off-diagonal values correspond to the pair of complex conjugate eigenvalues $a_2$ and $a_2^*$. The repeating eigenvalue $a_1$ appears on the diagonal while the matching off-diagonal parts are trivial, despite the tenfold multiplicity. Compared to the part with non-zero off-diagonal, for the repeating eigenvalue $a_1$ there are no other eigenvalues with the same real, but different imaginary part.



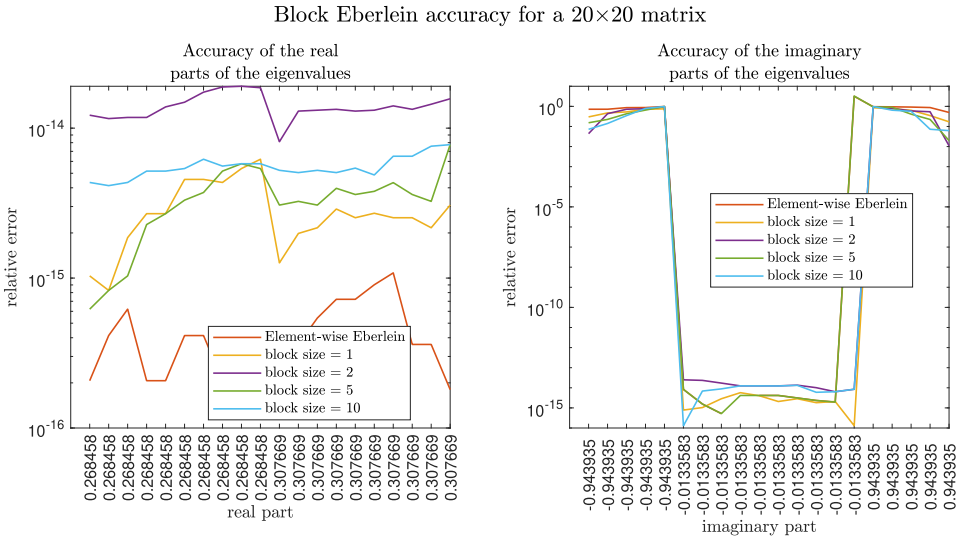

(a) Change in off($\mathbf{A}^{(k)}$) and off($\mathbf{B}^{(k)}$) for different sized blocks.




(b) The starting matrix $\mathbf{A}_4$ and block matrix $\Lambda$ obtained from the block Eberlein algorithm.

Figure 3.5: Results for **TestMatrix4**, for $n = 20$, $m_1 = 10$, $m_2 = 5$.

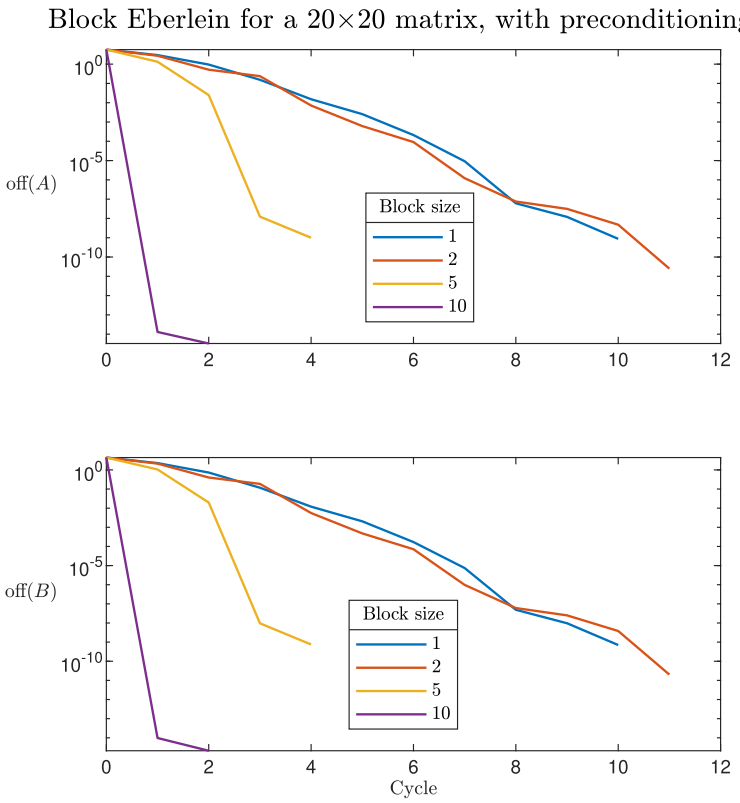
The real and imaginary parts of values obtained on the diagonal of $\Lambda$ are compared to the eigenvalues of the starting matrix $\mathbf{A}_4$. The results are shown in Figure 3.5c. The accuracy of the eigenvalue $a_1$ is very good, both for the real and imaginary part, as $a_1$ appears on the diagonal of $\Lambda$. That is the case for the real parts of the eigenvalues $a_2$ and $a_2^*$, as well. Because of the non-trivial blocks that correspond to $a_2$ and $a_2^*$, we can not expect the same accuracy for their imaginary parts.



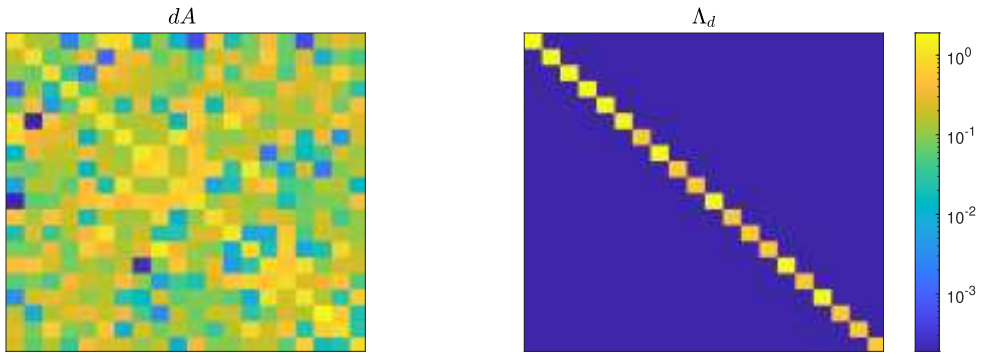

(c) Accuracy of the block Eberlein method in comparison to the Matlab eig function.

Figure 3.5: Results for **TestMatrix4**, for $n = 20$, $m_1 = 10$, $m_2 = 5$.

We solve the problem and simply avoid the discussion about the repeating real parts of eigenvalues by preconditioning the starting matrix. As in the element-wise case, we are going to multiply the starting matrix $\mathbf{A}_4$ with a complex number $d$ such that $\mathrm{Im}(d) \neq 0$. Then, with probability one, matrix $d\mathbf{A}_4$ has no eigenvalues with the same real and different imaginary parts. Applying the block Eberlein method on $d\mathbf{A}_4$ yields a fully diagonal matrix $\Lambda_d$, as seen in Figures 3.6a and 3.6b. Eigenvalues of $\mathbf{A}_4$ are retrieved by dividing the values on the diagonal of $\Lambda_d$ by $d$. According to the Figure 3.6c, both real and imaginary part of all eigenvalues are again highly accurate.

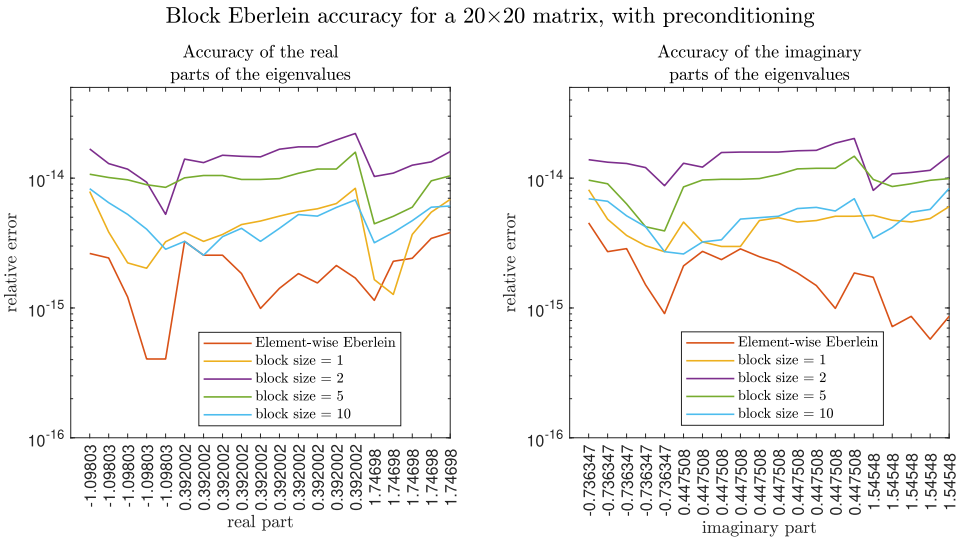



(a) Change in off($\mathbf{A}^{(k)}$) and off($\mathbf{B}^{(k)}$) for different sized blocks.




(b) The starting matrix $d\mathbf{A}_4$ and fully diagonal matrix $\Lambda_d$ obtained from the block Eberlein algorithm.

Figure 3.6: Results for **TestMatrix4**, for $n = 20$ and $m_1 = 10$, $m_2 = 5$.




(c) Accuracy of the block Eberlein method in comparison to the Matlab eig function.

Figure 3.6: Results for **TestMatrix4**, for $n = 20$, $m_1 = 10$, $m_2 = 5$.

In summation, this section shows how the block Eberlein algorithm behaves numerically. Similarly as for the element-wise Eberlein, the Hermitian part of $\mathbf{A}^{(k)}$ converges to a diagonal matrix, while $\mathbf{A}^{(k)}$, $k \geq 0$ converges to a normal matrix. Moreover, if the real parts of the eigenvalues of $\mathbf{A}$ are different, then $\mathbf{A}^{(k)}$ converges to a diagonal matrix with eigenvalues on the diagonal. Otherwise, the eigenvalues with equal real parts may contribute to non-zero off-diagonal elements. In comparison with the Matlab `eig` function, our block algorithm achieves satisfactory accuracy results, in line with the element-wise Eberlein method.

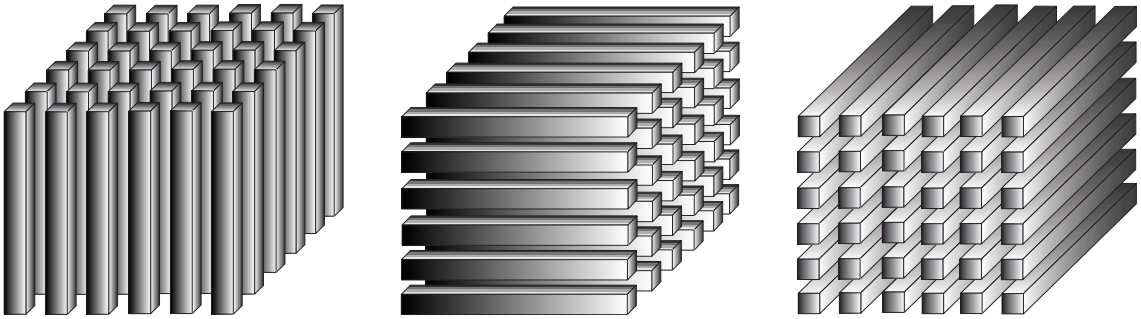
# 4. JACOBI-TYPE METHODS FOR TENSOR DIAGONALIZATION

## 4.1. ON THE HIGHER-ORDER TENSORS

Higher-order tensors, that is, multiway arrays of order three or more, have in recent decades found a wide spectrum of applications, including numerical linear algebra [14], multiway data analyses [50], signal processing [17, 19, 66, 71], image processing and machine learning [4]. More examples of applications can be found in [49].

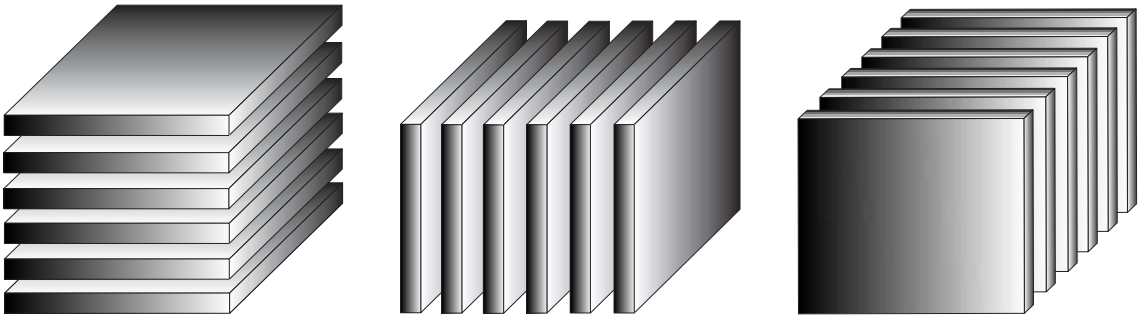
A tensor is an element of the tensor product of vector spaces, $\mathbb{R}^{n_1} \otimes \mathbb{R}^{n_2} \otimes \cdots \otimes \mathbb{R}^{n_d}$, [12, 33, 52]. It can be observed as a multiway array from $\mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$, [20]. A $d$-tuple $(n_1, n_2, \ldots, n_d)$ defines tensor *dimensions* or *modes*. The number of dimensions $d$ is called the *order* of a tensor. In this chapter we denote vectors, or the first order tensors, by lowercase letters (e.g. $a, b, \ldots$). Matrices, or second order tensors, are denoted by uppercase letters (e.g. $A, B, \ldots$), while tensors of order three or more are denoted by calligraphic letters (e.g. $\mathscr{A}, \mathscr{B}, \ldots$). The element of tensor $\mathscr{A}$ on position $(i_1, i_2, \ldots, i_d)$ is denoted by $a_{i_1 i_2 \cdots i_d}$. A tensor can be considered as a set of vectors. Analogously to matrix columns and rows, these vectors are called tensor *fibers*. They are defined by fixing all indices except one. Hence, matrix columns are mode-1 fibers, and matrix rows are mode-2 fibers. Higher-order tensor has fibers in $d$ modes. See Fig. 4.1 for illustration. Paint 3D was used to generate tensor illustrations in this chapter.

One can also observe two-dimensional sections of a tensor, called *slices*. They are defined by fixing all indices except two. The 3rd-order tensor has horizontal, frontal, and lateral slices (see Fig. 4.2).



(a) Mode-1 fibers (b) Mode-2 fibers (c) Mode-3 fibers

Figure 4.1: Fibers of a 3rd-order tensor



(a) Horizontal slices (b) Lateral slices (c) Frontal slices

Figure 4.2: Slices of a 3rd-order tensor

It is often useful to have a matrix reperesentation of a tensor. This is going to simplify complicated tensor computations. We achieve that by reordering tensor elements into a matrix. Mode-$m$ *matricization* or *unfolding* of an $n_1 \times n_2 \cdots \times n_d$ tensor $\mathscr{A}$ is an $n_m \times (n_1 \cdots n_{m-1} n_{m+1} \cdots n_d)$ matrix $A_{(m)}$, such that the columns of $A_{(m)}$ are mode-$m$ fibers of $\mathscr{A}$. Formally, in mode-$m$ matricization, tensor element $(i_1, i_2, \ldots, i_d)$ maps to the matrix element $(i_m, j)$, where

$$j = 1 + \sum_{\substack{k=1, \\ k \neq m}}^{d} (i_k - 1) N_k, \quad N_k = \prod_{\substack{l=1, \\ l \neq m}}^{k-1} n_l.$$

Following this mapping rule there are, in total, $d$ different unfoldings of an order-$d$ tensor. For example, let $\mathscr{A} \in \mathbb{R}^{4 \times 3 \times 2}$ be given by its two frontal slices

$$\mathscr{A}(:,:,1) = \begin{bmatrix} 1 & 5 & 9 \\ 2 & 6 & 10 \\ 3 & 7 & 11 \\ 4 & 8 & 12 \end{bmatrix}, \quad \mathscr{A}(:,:,2) = \begin{bmatrix} 13 & 17 & 21 \\ 14 & 18 & 22 \\ 15 & 19 & 23 \\ 16 & 20 & 24 \end{bmatrix}.$$

There are three matricizations of tensor $\mathscr{A}$,

$$A_{(1)} = \begin{bmatrix} 1 & 5 & 9 & 13 & 17 & 21 \\ 2 & 6 & 10 & 14 & 18 & 22 \\ 3 & 7 & 11 & 15 & 19 & 23 \\ 4 & 8 & 12 & 16 & 20 & 24 \end{bmatrix},$$

$$A_{(2)} = \begin{bmatrix} 1 & 2 & 3 & 4 & 13 & 14 & 15 & 16 \\ 5 & 6 & 7 & 8 & 17 & 18 & 19 & 20 \\ 9 & 10 & 11 & 12 & 21 & 22 & 23 & 24 \end{bmatrix},$$

and

$$A_{(3)} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 & 21 & 22 & 23 & 24 \end{bmatrix}.$$

Column ordering in mode-$m$ matricization mapping given above can be defined differently [47]. As long as it is consistent with the related tensor calculations, the specific column ordering does not matter. However, the matricization rule that we use is the one that is commonly used in the literature.

We now define tensors with symmetric and anti-symmetric structure. A tensor of order $d$ with each mode of the same length, $\mathscr{A} \in \mathbb{R}^{n \times n \times \cdots \times n}$, is called *symmetric* if

$$a_{\ldots i \ldots j \ldots} = a_{\ldots j \ldots i \ldots},$$

for any pair of indices $(i, j)$, $1 \leq i, j \leq n$. Some authors call it a *supersymmetric* tensor [13, 48]. This means that the entries of a symmetric tensor are invariant to index permutations. In a third-order tensor $\mathscr{A}$ we have

$$a_{ijk} = a_{ikj} = a_{jik} = a_{jki} = a_{kij} = a_{kji},$$

for any indices $1 \leq i, j, k \leq n$. Consequently, all unfoldings of a symmetric tensor are equal,

$$A_{(1)} = A_{(2)} = \cdots = A_{(d)}.$$

For example, third order tensor $\mathscr{A}$ given with its slices,

$$\mathscr{A}(:,:,1) = \begin{bmatrix} 1 & 5 & 6 \\ 5 & 4 & 0 \\ 6 & 0 & 7 \end{bmatrix}, \quad \mathscr{A}(:,:,2) = \begin{bmatrix} 5 & 4 & 0 \\ 4 & 2 & 9 \\ 0 & 9 & 8 \end{bmatrix}, \quad \mathscr{A}(:,:,3) = \begin{bmatrix} 6 & 0 & 7 \\ 0 & 9 & 8 \\ 7 & 8 & 3 \end{bmatrix}.$$

is symmetric, and all matricizations are in the form of

$$A_{(1)} = A_{(2)} = A_{(3)} = \begin{bmatrix} 1 & 5 & 6 & 5 & 4 & 0 & 6 & 0 & 7 \\ 5 & 4 & 0 & 4 & 2 & 9 & 0 & 9 & 8 \\ 6 & 0 & 7 & 0 & 9 & 8 & 7 & 8 & 3 \end{bmatrix}.$$

We say that a tensor $\mathscr{A} \in \mathbb{R}^{n \times n \times \cdots \times n}$ is *antisymmetric* if

$$a_{\ldots i \ldots j \ldots} = -a_{\ldots j \ldots i \ldots},$$

for every pair of indices $(i, j)$. It follows from this property that

$$a_{\ldots i \ldots i \ldots} = -a_{\ldots i \ldots i \ldots} = 0.$$

Hence, only non-trivial elements of an antisymmetric tensor are the ones with all indices distinct. An example of an antisymmetric third order tensor is given below,

$$\mathscr{A}(:,:,1) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -2 \\ 0 & 2 & 0 \end{bmatrix}, \quad \mathscr{A}(:,:,2) = \begin{bmatrix} 0 & 0 & 2 \\ 0 & 0 & 0 \\ -2 & 0 & 0 \end{bmatrix}, \quad \mathscr{A}(:,:,3) = \begin{bmatrix} 0 & -2 & 0 \\ 2 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

Order-$d$ tensor $\mathscr{A}$ is called *diagonal* when an entry $a_{i_1 i_2 \cdots i_d}$ is zero if it has at least two different indices, $i_l \neq i_m$. Hence, only entries $a_{i \cdots i}$, for $i = 1, \ldots, d$, can be non-trivial. An example of order-3 dimension-3 diagonal tensor is given below,

$$\mathscr{A}(:,:,1) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \mathscr{A}(:,:,2) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \mathscr{A}(:,:,3) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 3 \end{bmatrix}.$$

Moreover, multiplying a matrix by other matrices from the left- or right-hand side can be generalized to multiplying a tensor by matrices from each of the $d$ sides. The *mode-$m$ product* of tensor $\mathscr{A} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$ and matrix $X \in \mathbb{R}^{p \times n_m}$ is a tensor $\mathscr{B}$,

$$\mathscr{B} = \mathscr{A} \times_m X \in \mathbb{R}^{n_1 \times \cdots \times n_{m-1} \times p \times n_{m+1} \times \cdots \times n_d},$$

such that

$$B_{(m)} = XA_{(m)}. \tag{4.1}$$

This can be expressed element-wise as

$$(\mathscr{A} \times_m X)_{i_1 \cdots i_{m-1} j i_{m+1} \cdots i_d} = \sum_{i_m=1}^{n_m} a_{i_1 i_2 \cdots i_d} x_{j i_m}.$$

Another property is inherited from the matrix case — if $\mathscr{A}$ defines a multilinear operator, the mode-$m$ product with the matrix $X$ relates to a change of basis.

Let us discuss these notions for a second order tensor, i.e. matrix $A \in \mathbb{R}^{m \times n}$. Mode-1 matricization of $A$ is $A$, while mode-2 matricization of $A$ is its transpose, that is

$$A_{(1)} = A \quad \text{and} \quad A_{(2)} = A^T. \tag{4.2}$$

Furthermore, let $B = A \times_1 X$, that is $B_{(1)} = XA_{(1)}$. Using (4.2) we have $B = XA$. Therefore, mode-1 product of matrix $A$ with matrix $X$ is equivalent to multiplying $A$ by $X$ from the left-hand side. Similarly, let $C = A \times_2 Y$ or, equivalently, $C_{(2)} = YA_{(2)}$. Then we have

$$C^T = YA^T \implies C = AY^T.$$

Hence, mode-2 product of matrix $A$ with matrix $Y$ is equivalent to multiplying $A$ by $Y^T$ from the right-hand side. We can now observe what happens when multiplying simultaneously in both modes. We get

$$\begin{aligned} A \times_1 X \times_2 Y &= (XA) \times_2 Y = XAY^T, \\ A \times_2 Y \times_1 X &= (AY^T) \times_1 X = XAY^T. \end{aligned} \tag{4.3}$$

Clearly, the order of multiplication in distinct modes is irrelevant. On the other hand, order of multiplication is important when multiplying in the same mode,

$$A \times_1 X \times_1 Y = (XA) \times_1 Y = YXA = A \times_1 (YX). \tag{4.4}$$

Properties (4.3) and (4.4) for simultaneous mode multiplication are true for order-$d$ tensors in general. We have

$$\mathscr{A} \times_m X \times_n Y = \mathscr{A} \times_n Y \times_m X, \quad m \neq n. \tag{4.5}$$

$$\mathscr{A} \times_m X \times_m Y = \mathscr{A} \times_m (YX). \tag{4.6}$$

We are now ready to define *Tucker decomposition* [72] of a tensor, named after Ledyard R. Tucker. It was originally proposed by Hitchcock [44]. It decomposes order-$d$ tensor $\mathscr{A} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$ into a *core tensor* $\mathscr{S}$ multiplied in all modes with matrices $U_i \in \mathbb{R}^{n_i \times n_i}$, $i = 1, \ldots, d$, respectively. Decomposition is written as

$$\mathscr{A} = \mathscr{S} \times_1 U_1 \times_2 U_2 \cdots \times_d U_d, \tag{4.7}$$

where $\mathscr{S}$ is of same order and dimension as tensor $\mathscr{A}$.

The notion of tensor rank is different from matrix rank in a sense that it is not defined as the number of linearly independent mode-$m$ fibers. Instead, we first define *rank one tensor* as any tensor $\mathscr{A} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$ that can be written as an outer product of $d$ vectors, that is

$$\mathscr{A} = u^{(1)} \circ u^{(2)} \circ \cdots \circ u^{(d)}. \tag{4.8}$$

The vector outer product is denoted by $\circ$ and we have that, element-wise,

$$a_{i_1 i_2 \cdots i_d} = u^{(1)}_{i_1} u^{(2)}_{i_2} \cdots u^{(d)}_{i_d}, \quad \text{for all } 1 \leq i_m \leq n_m, \quad 1 \leq m \leq d.$$

In other words, an entry of a tensor is the product of the corresponding vector entries. Notice that if in (4.8) we have $u^{(1)} = u^{(2)} = \cdots = u^{(d)}$, then the corresponding rank one tensor is symmetric because its elements are indifferent to index permutations.

In general, each tensor can be written as a linear combination of rank one tensors,

$$\mathscr{A} = \sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \cdots \sum_{i_d=1}^{n_d} a_{i_1 i_2 \cdots i_d} \left( e^{(1)}_{i_1} \circ e^{(2)}_{i_2} \circ \cdots \circ e^{(d)}_{i_d} \right), \tag{4.9}$$

where $e^{(m)}_{i_m} \in \mathbb{R}^{n_m}$, $1 \leq m \leq d$, is a unit vector with $i_m$ entry equal to one. The decomposition (4.9) is actually a decomposition with respect to the canonical basis for $\mathbb{R}^{n_1} \otimes \mathbb{R}^{n_2} \otimes \cdots \otimes \mathbb{R}^{n_d}$. *Tensor rank* is defined as the smallest number $r$ such that $\mathscr{A}$ can be written as a linear combination of $r$ rank one tensors,

$$\operatorname{rank}(\mathscr{A}) = \min\{r \mid \mathscr{A} = \sum_{i=1}^{r} \lambda_i u^{(1)}_i \circ u^{(2)}_i \circ \cdots \circ u^{(d)}_i\}.$$

Figure 4.3: Order-3 tensor $\mathscr{A} = u^{(1)} \circ u^{(2)} \circ u^{(3)}$ is of rank one.

Tensor decomposition

$$\mathscr{A} = \sum_{i=1}^{r} \lambda_i u_i^{(1)} \circ u_i^{(2)} \circ \cdots \circ u_i^{(d)},$$

when $r$ is minimal is called *tensor rank decomposition*. It was introduced by Hitchcock [44] in 1927. It was later rediscovered by Harshman [42] who named it PARAFAC (for *parallel factors*), and, separately, by Carrol and Chang [16] who called it CANDECOMP (*canonical decomposition* ). It is often referred to as CP (CANDECOMP/ PARAFAC) decomposition.

The *inner product* of two tensors $\mathscr{A},\mathscr{B} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$ is defined as

$$\langle \mathscr{A}, \mathscr{B} \rangle = \sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \cdots \sum_{i_d=1}^{n_d} a_{i_1 i_2 \cdots i_d} b_{i_1 i_2 \cdots i_d}. \tag{4.10}$$

The tensor norm induced from the inner product (4.10) is a generalization of the Frobenius norm for matrices. It is given by

$$\|\mathscr{A}\|_F = \sqrt{\langle \mathscr{A}, \mathscr{A} \rangle} = \sqrt{\sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \cdots \sum_{i_d=1}^{n_d} a_{i_1 i_2 \ldots i_d}^2}.$$

## 4.2. PROBLEM DESCRIPTION

In this chapter, we observe the tensor generalization of the matrix singular value decomposition,

$$A = U\Sigma V^T, \tag{4.11}$$

where $U$ and $V$ are orthogonal matrices, and $\Sigma$ is a diagonal matrix. For a tensor $\mathscr{A}$ it is in the form of Tucker decomposition

$$\mathscr{A} = \mathscr{S} \times_1 U_1 \times_2 U_2 \cdots \times_d U_d, \tag{4.12}$$

where matrices $U_i$, $i = 1, \ldots, d$, are orthogonal, and the core tensor $\mathscr{S}$ plays the role of the diagonal matrix $\Sigma$ from (4.11). Relation (4.12) can also be written as

$$\mathscr{S} = \mathscr{A} \times_1 U_1^T \times_2 U_2^T \cdots \times_d U_d^T. \tag{4.13}$$

A tensor is said to be *orthogonally diagonalizable* if it can be transformed into a diagonal tensor using orthogonal transformations in each mode. That is, for a diagonalizable tensor $\mathscr{A}$, we can find orthogonal matrices $U_i$, $i = 1, \ldots, d$, such that the core tensor $\mathscr{S}$ is diagonal. The matrices $U_i$ represent change of bases in each mode $i$. It is known that matrices can always be orthogonally diagonalized, and said diagonalization is achieved using SVD. In contrast to matrices, a general tensor can not be diagonalized [68]. More specifically, not even every symmetric tensor can be diagonalized by orthogonal transformations. We are going to explain why.

We can observe the tensor digonalization problem from another perspective. To keep the notation simple, in the decomposition (4.12), let $d = 3$, and $U = U_1$, $V = U_2$, and $W = U_3$. Every tensor $\mathscr{A} \in \mathbb{R}^{n \times n \times n}$ can be written as a linear combination of rank one tensors as

$$\mathscr{A} = \mathscr{S} \times_1 U \times_2 V \times_3 W = \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} \sigma_{ijk} (u_i \circ v_j \circ w_k), \tag{4.14}$$

where $u_i$, $v_j$ and $w_k$ are $i$th, $j$th and $k$th column of orthogonal matrices $U$, $V$ and $W$, respectively, and $\sigma_{ijk}$ is the element of tensor $\mathscr{S}$ in the position $(i, j, k)$. Diagonalizing a tensor is then equivalent to finding vectors $u_i$, $v_j$ and $w_k$, such that $\sigma_{ijk} = 0$ unless $i = j = k$. Hence, if a tensor is diagonalizable, it can be decomposed in a way that (4.14)

has at most $n$ summands,

$$\mathscr{A} = \sum_{i=1}^{n} \sigma_{iii}(u_i \circ v_i \circ w_i). \tag{4.15}$$

To show that not even every symmetric tensor can be diagonalized, let $U = V = W$. The *symmetric rank* of a symmetric tensor is the smallest number $r$ such that tensor $\mathscr{A}$ can be decomposed as

$$\mathscr{A} = \sum_{i=1}^{r} \lambda_i(u_i \circ u_i \circ u_i). \tag{4.16}$$

If $r$ is minimal, decomposition (4.16) is known as *symmetric rank decomposition* [20]. The symmetric rank of a generic symmetric tensor of order $d \geq 3$, according to Alexander-Hirschowitz theorem [3], is

$$\left\lceil \frac{1}{n}\binom{n+d-1}{d} \right\rceil,$$

except in finite number of cases where it should be increased by one. This number exceeds the dimension $n$. On the other hand, according to (4.15), a diagonalizable tensor has a rank not greater than $n$. To sum up, the change of bases that will diagonalize a generic symmetric tensor does not exist.

Hence, we try to find the *approximate* tensor diagonalization, such that tensor $\mathscr{S}$ is *as close as possible* to a diagonal one. We measure the distance of a tensor $\mathscr{A}$ from a diagonal tensor using the tensor *off-norm* that is defined as

$$\operatorname{off}^2(\mathscr{A}) = \|\mathscr{A}\|_F^2 - \|\operatorname{diag}(\mathscr{A})\|_F^2.$$

The off-norm is actually the Frobenius norm of the off-diagonal part of the tensor. In order to obtain the approximate tensor diagonalization, we want to minimize the off-norm

$$\operatorname{off}(\mathscr{A}) \to \min.$$

Using the fact that Frobenius norm is invariant to orthogonal transformations, the problem is equivalent to maximizing the (squared) Frobenius norm of the diagonal,

$$\sum_{i=1}^{n} a_{i\dots i}^2 \to \max. \tag{4.17}$$

The *relative off-norm* of $\mathscr{A}$ is given as

$$\frac{\operatorname{off}(\mathscr{A})}{\|\mathscr{A}\|_F},$$

and it can also be used to measure distance from $\mathscr{A}$ to a diagonal tensor. Obviously, a diagonal tensor has the relative off-norm equal to zero. The relative off-norm of a general tensor is close to one.

In our maximization problem, the goal is to make off-diagonal elements become less and less significant as opposed to diagonal elements. Keeping (4.13) in mind, we define an iterative process,

$$\mathscr{A}^{(k)} = \mathscr{A}^{(k-1)} \times_1 (R_{U_1,k})^T \times_2 (R_{U_2,k})^T \cdots \times_d (R_{U_d,k})^T, \quad k > 0, \quad \mathscr{A}^{(0)} = \mathscr{A}, \tag{4.18}$$

where $R_{U_1,k}, R_{U_2,k}, \cdots, R_{U_d,k} \in \mathbb{R}^{n \times n}$ are plane rotations. They depend on an index pair $(p_k, q_k)$, called the *pivot pair*, and a rotation angle $\phi_k$ as follows,

$$R_{U_l,k} = R(p_k, q_k, \phi_k) = \begin{bmatrix} 1 & & & & & & & & & \\ & \ddots & & & & & & & & \\ & & 1 & & & & & & & \\ & & & \cos\phi_k & & & -\sin\phi_k & & & p_k \\ & & & & 1 & & & & & \\ & & & & & \ddots & & & & \\ & & & & & & 1 & & & \\ & & & \sin\phi_k & & & \cos\phi_k & & & q_k \\ & & & & & & & 1 & & \\ & & & & & & & & \ddots & \\ & & & & & & & & & 1 \end{bmatrix} \tag{4.19}$$

It is crucial to notice that the rotations in (4.18) change only elements of $\mathscr{A}^{(k-1)}$ with indices containing $p_k$ or $q_k$. This enables us to reduce the problem to a $2 \times 2 \times \cdots \times 2$ subproblem, which we go through in detail later. After $k$ iterations (4.18) we get

$$\mathscr{A}^{(k)} = \mathscr{A} \times_1 (U_1^{(k)})^T \times_2 (U_2^{(k)})^T \cdots \times_d (U_d^{(k)})^T, \tag{4.20}$$

where $U_l^{(0)} = I_n$, and the orthogonal matrices $U_l^{(k)}$ can be expressed as

$$U_l^{(k)} = U_l^{(k-1)} R_{U_l,k}, \quad l = 1, \ldots, d.$$

We study two different approaches to the iteration process (4.18). The first is to maximize the Frobenius norm of the diagonal of the iterates $\mathscr{A}^{(k)}$. In Section 4.3 we give the Jacobi-type algorithms and the convergence results from literature [8, 45, 54, 55, 65, 74].

The second approach is to maximize the trace of the iterates $\mathscr{A}^{(k)}$. We design a Jacobi-type algorithm that maximizes the trace and prove its global convergence in Section 4.4. In Section 4.5 we present the results of our numerical experiments.

## 4.3. Maximization of the Frobenius norm of the diagonal

In this section we observe only third order tensors. Therefore, decomposition (4.12) becomes simpler,

$$\mathscr{A} = \mathscr{S} \times_1 U \times_2 V \times_3 W, \tag{4.21}$$

where $U, V$ and $W$ are orthogonal matrices of appropriate dimension. The goal is to find $U, V, W$ such that core tensor $\mathscr{S}$ is as close to a diagonal tensor as possible. The approach we take in this section is to maximize Frobenius norm of the diagonal of $\mathscr{S}$. Let $O_n$ be the set of $n \times n$ orthogonal matrices. We define the objective function $f : O_n \times O_n \times O_n \to \mathbb{R}$ as

$$f(U,V,W) = \| \operatorname{diag}(\mathscr{A} \times_1 U^T \times_2 V^T \times_3 W^T) \|_F^2. \tag{4.22}$$

Various authors have designed Jacobi-type algorithms [8, 54, 55, 65, 74] for tensor diagonalization that solve the problem of maximizing (4.22). We summarize the ideas behind the algorithms and state the main convergence results.

The core idea of these methods is to reduce an $n \times n \times n$ problem to a $2 \times 2 \times 2$ subproblem. In the following text, we denote matrices and tensors of dimension two with a hat, that is, $\widehat{A} \in \mathbb{R}^{2\times 2}$, $\widehat{\mathscr{A}} \in \mathbb{R}^{2\times 2\times 2}$. For a pivot pair $(p,q)$, $1 \le p < q \le n$, subtensor $\widehat{\mathscr{A}}$ is constructed as

$$\widehat{\mathscr{A}}(:,:,1) = \begin{bmatrix} a_{ppp} & a_{pqp} \\ a_{qpp} & a_{qqp} \end{bmatrix}, \quad \widehat{\mathscr{A}}(:,:,2) = \begin{bmatrix} a_{ppq} & a_{pqq} \\ a_{qpq} & a_{qqq} \end{bmatrix}. \tag{4.23}$$

Then the subproblem is

$$\widehat{\mathscr{S}} = \widehat{\mathscr{A}} \times_1 \widehat{R}_U^T \times_2 \widehat{R}_V^T \times_3 \widehat{R}_W^T, \tag{4.24}$$

where $\widehat{R}_U, \widehat{R}_V, \widehat{R}_W \in \mathbb{R}^{2\times 2}$ are plane rotations, and $\widehat{\mathscr{S}} \in \mathbb{R}^{2\times 2\times 2}$ is given as

$$\widehat{\mathscr{S}}(:,:,1) = \begin{bmatrix} \sigma_{ppp} & \sigma_{pqp} \\ \sigma_{qpp} & \sigma_{qqp} \end{bmatrix}, \quad \widehat{\mathscr{S}}(:,:,2) = \begin{bmatrix} \sigma_{ppq} & \sigma_{pqq} \\ \sigma_{qpq} & \sigma_{qqq} \end{bmatrix}. \tag{4.25}$$

In the iterative process

$$\mathscr{A}^{(k)} = \mathscr{A}^{(k-1)} \times_1 R_{U,k}^T \times_2 R_{V,k}^T \times_3 R_{W,k}^T, \tag{4.26}$$

each iteration $k$ consists of three steps:

Step 1. Choosing a pivot pair $(p_k, q_k)$;

Step 2. Computing rotation matrices $\widehat{R}_{U,k}, \widehat{R}_{V,k}, \widehat{R}_{W,k} \in \mathbb{R}^{2\times 2}$ which maximize the Frobenius norm of the diagonal of a corresponding subtensor $\widehat{\mathscr{A}^{(k)}}$ in (4.26);

Step 3. Updating orthogonal matrices $U^{(k)}, V^{(k)}, W^{(k)}$ with

$$U^{(k)} = U^{(k-1)} R_{U,k}, \quad V^{(k)} = V^{(k-1)} R_{V,k}, \quad W^{(k)} = W^{(k-1)} R_{W,k}, \tag{4.27}$$

where $R_{U,k}, R_{V,k}, R_{V,k} \in \mathbb{R}^{n\times n}$ are formed as in (4.19) using corresponding matrices from the Step 2.

We then have

$$\mathscr{A}^{(k)} = \mathscr{A} \times_1 (U^{(k)})^T \times_2 (V^{(k)})^T \times_3 (W^{(k)})^T. \tag{4.28}$$

These steps are repeated until convergence. Usually, the stopping criteria for the iterative process (4.26) is a fixed number of iterations or the change in Frobenius norm of the diagonal of $\mathscr{A}^{(k)}$ becoming small enough. The algorithms vary in Step 2., finding the orthogonal matrices that solve the subproblem, as well as in Step 1. which is important for the convergence results.

In [65], Van Loan and Moravitz Martin proposed an algorithm for third order tensors that is a generalization of the Jacobi SVD algorithm for matrices. Decomposition (4.14) of the tensor $\widehat{\mathscr{A}}$ is computed such that $\sigma_{ppp}^2 + \sigma_{qqq}^2$ is maximized.

Let us define operators we need for this algorithm, vec and reshape. Operator vec, if used on a matrix $A \in \mathbb{R}^{m\times n}$ yields a vector in $\mathbb{R}^{mn}$ that is formed by stacking columns of $A$, from the first to the $n$th column. Analogously, for a tensor $\mathscr{A} \in \mathbb{R}^{n\times n\times n}$, the vector $\mathrm{vec}(\mathscr{A}) \in \mathbb{R}^{n^3}$ is formed by stacking mode-1 fibers of $\mathscr{A}$. We can write the subtensor $\widehat{\mathscr{A}}$ from (4.23) as

$$\mathrm{vec}(\widehat{\mathscr{A}}) = \begin{bmatrix} a_{ppp} & a_{qpp} & a_{pqp} & a_{qqp} & a_{ppq} & a_{qpq} & a_{pqq} & a_{qqq} \end{bmatrix}^T. \tag{4.29}$$

Operator reshape rearranges vector elements into a matrix, the opposite of what vec does. If $b \in \mathbb{R}^{mn}$, then $\mathrm{reshape}(b, m, n)$ returns an $m \times n$ matrix whose columns are sub-arrays of $b$, of length $m$. For example,

$$\mathrm{reshape}(\mathrm{vec}(\widehat{\mathscr{A}}), 2, 4) = \begin{bmatrix} a_{ppp} & a_{pqp} & a_{ppq} & a_{pqq} \\ a_{qpp} & a_{qqp} & a_{qpq} & a_{qqq} \end{bmatrix},$$

which is actually equal to mode-1 unfolding of $\widehat{\mathscr{A}}$.

Vector outer product $u \circ v \circ w \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ can be expressed as a vector $w \otimes v \otimes u \in \mathbb{R}^{n_1 n_2 n_3}$, where $\otimes$ denotes the Kronecker product,

$$\operatorname{vec}(u \circ v \circ w) = w \otimes v \otimes u.$$

Computations in [65] involve (4.14) in the vector form,

$$\operatorname{vec}(\mathscr{A}) = \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} \sigma_{ijk}(w_k \otimes v_j \otimes u_i), \tag{4.30}$$

where $u_i$, $v_j$ and $w_k$ are $i$th, $j$th and $k$th column of orthogonal matrices $U$, $V$ and $W$, respectively. The equation above can also be represented as matrix and vector product,

$$a = (W \otimes V \otimes U) \cdot \sigma,$$

or equivalently,

$$\sigma = (W^T \otimes V^T \otimes U^T) \cdot a,$$

where $a = \operatorname{vec}(\mathscr{A})$ and $\sigma = \operatorname{vec}(\mathscr{S})$. In the $2 \times 2 \times 2$ subproblem (4.24) we want to find plane rotations $\widehat{R}_U, \widehat{R}_V, \widehat{R}_W \in \mathbb{R}^{2\times 2}$ which maximize $\sigma_{ppp}^2 + \sigma_{qqq}^2$ in

$$\begin{bmatrix} \sigma_{ppp} \\ \sigma_{qpp} \\ \sigma_{pqp} \\ \sigma_{qqp} \\ \sigma_{ppq} \\ \sigma_{qpq} \\ \sigma_{pqq} \\ \sigma_{qqq} \end{bmatrix} = (\widehat{R}_W^T \otimes \widehat{R}_V^T \otimes \widehat{R}_U^T) \begin{bmatrix} a_{ppp} \\ a_{qpp} \\ a_{pqp} \\ a_{qqp} \\ a_{ppq} \\ a_{qpq} \\ a_{pqq} \\ a_{qqq} \end{bmatrix}. \tag{4.31}$$

Solving the subproblem consists of three microiterations. We hold two variables constant while varying the third:

$$\begin{aligned} \widehat{\sigma}_1 &= (I \otimes I \otimes \widehat{R}_U^T)\widehat{a}, \\ \widehat{\sigma}_2 &= (I \otimes \widehat{R}_V^T \otimes I)\widehat{\sigma}_1, \\ \widehat{\sigma}_3 &= (\widehat{R}_W^T \otimes I \otimes I)\widehat{\sigma}_2, \end{aligned}$$

that is, holding two of the modes fixed, we optimize in only one mode at a time. This approach is known as alternating least squares (ALS). Indeed, after the third microiteration we get

$$\widehat{\sigma}_3 = (\widehat{R}_W^T \otimes I \otimes I)(I \otimes \widehat{R}_V^T \otimes I)(I \otimes I \otimes \widehat{R}_U^T)\widehat{a} = (\widehat{R}_W^T \otimes \widehat{R}_V^T \otimes \widehat{R}_U^T)\widehat{a}.$$

Each of the microiterations above are equivalent with respect to permutation of tensor elements. In other words, each of the steps is performed in the same way, but looking from a different perspective, i.e. mode. Therefore, we only show how to find $\widehat{R}_U$ in

$$\widehat{\sigma} = (I \otimes I \otimes \widehat{R}_U^T)\widehat{a} = \begin{bmatrix} \widehat{R}_U^T & & & \\ & \widehat{R}_U^T & & \\ & & \widehat{R}_U^T & \\ & & & \widehat{R}_U^T \end{bmatrix} \begin{bmatrix} a_{ppp} \\ a_{qpp} \\ a_{pqp} \\ a_{qqp} \\ a_{ppq} \\ a_{qpq} \\ a_{pqq} \\ a_{qqq} \end{bmatrix},$$

such that $\sigma_{ppp}^2 + \sigma_{qqq}^2$ is maximized. It is sufficient to find $\widehat{R}_U$ which maximizes the Frobenius norm of the diagonal of $\widehat{R}_U^T \widehat{A}$, where

$$\widehat{A} = \begin{bmatrix} a_{ppp} & a_{pqq} \\ a_{qpp} & a_{qqq} \end{bmatrix}.$$

Let the SVD of matrix $\widehat{A}$ be

$$\widehat{A} = \tilde{U} \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix} \begin{bmatrix} c & s \\ -s & c \end{bmatrix},$$

where $(c,s)$ is a cosine/sine pair. If we do have a $2 \times 2$ matrix $Z$ such that the Frobenius norm of the diagonals of

$$Z \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix} \begin{bmatrix} c & s \\ -s & c \end{bmatrix}$$

is maximized, we then set $\widehat{R}_U^T = Z\tilde{U}^T$. It is shown in [65] that $Z$ is constructed using right singular vector of matrix $M$ associated with the smallest singular value, where

$$M = \begin{bmatrix} \sigma_2 s & \sigma_1 c \\ \sigma_1 s & \sigma_2 c \end{bmatrix}.$$

The method from [65] for case $2 \times 2 \times 2$ is given in the Algorithm 7.

**Algorithm 7** Jacobi compress $2\times 2\times 2$ [65]

**Input:** $\widehat{\mathscr{A}} \in \mathbb{R}^{2\times 2\times 2}$

**Output:** orthogonal matrices $\widehat{R}_U, \widehat{R}_V, \widehat{R}_W \in \mathbb{R}^{2\times 2}$, $\widehat{\sigma}$

$\widehat{\sigma_0} = \text{vec}(\widehat{\mathscr{A}})$

**for** l=1,2,3 (each mode) **do**

$\%$ Solves $\widehat{\sigma_1} = (I \otimes I \otimes \widehat{R}_U^T)\ \widehat{\sigma}_0$

$B = \text{reshape}(\widehat{\sigma}_{l-1}, 2, 4)$ (or according to the current mode $l$)

$\widehat{\Sigma}_1 = B(1:2, 1:2), \widehat{\Sigma}_2 = B(1:2, 3:4)$

$\widehat{A} = [\widehat{\Sigma}_1(:,1)\ \ \widehat{\Sigma}_2(:,2)]$

$\left[\tilde{U}, \tilde{S}, \tilde{V}\right] = \textbf{svd}(\widehat{A}), \quad M = \begin{bmatrix} \sigma_2 s & \sigma_1 c \\ \sigma_1 s & \sigma_2 c \end{bmatrix}$

$\left[\bar{U}, \bar{S}, \bar{V}\right] = \textbf{svd}(M), \quad Z = \begin{bmatrix} \bar{v}_{12} & \bar{v}_{22} \\ -\bar{v}_{22} & \bar{v}_{12} \end{bmatrix}$

$R_l^T = Z\tilde{U}^T$

$\widehat{\sigma}_l = \text{vec}(\left[R_l^T \widehat{\Sigma}_1 \mid R_l^T \widehat{\Sigma}_2\right])$

**end for**

$\widehat{\sigma} = \widehat{\sigma}_3, \widehat{R}_W = R_2, \widehat{R}_V = R_3, \widehat{R}_U = R_1$

Lastly, we say something about choosing pivot pair $(p_k, q_k) = (p, q)$, $1 \le p < q \le n$, in each iteration. In any cyclic pivot strategy the idea is to sweep through the whole tensor making a cycle, and then repeat until convergence. On the other hand, Jacobi compress method of Van Loan and Moravitz consists of three sweeps, one for every orientation/mode of the tensor, and then repeating the process until convergence. If tensor $\mathscr{A}$ is diagonalizable, Jacobi compress algorithm yields $U$, $V$ and $W$ which diagonalize it, that is, $\mathscr{S}$ is diagonal. For non-diagonalizable tensors, numerical convergence is seen in practice. However, to the best of our knowledge, the proof of convergence of Algorithm 7 is not given.

Begović Kovač [8] designed similar algorithm along with the proof of convergence. Each iteration $k$ is again in the form of (4.26). Matrices $R_{U,k}, R_{V,k}, R_{W,k}$ are set to be the plane rotations that depend on a pivot pair $(p, q)$ and an angle $\phi$. Using the ALS approach, the iteration $k$ consists of three microiterations. Again, in each microiteration we hold two

variables constant and vary the third one. We have

$$\mathscr{B} = \mathscr{A}^{(k-1)} \times_1 R_{U,k}^T \times_2 I \times_3 I, \tag{4.32}$$

$$\mathscr{C} = \mathscr{B} \times_1 I \times_2 R_{V,k}^T \times_3 I, \tag{4.33}$$

$$\mathscr{A}^{(k)} = \mathscr{C} \times_1 I \times_2 I \times_3 R_{W,k}^T, \tag{4.34}$$

where $\mathscr{B}$ and $\mathscr{C}$ are intermediate steps. As before, combining the three expressions above using the mode-$m$ properties (4.5) and (4.6) we get the iterative process (4.26). Next, matrices $U,V,W$ are updated using (4.27) and the process is repeated until convergence. While the pivot pair is the same for all three matrices $R_{U,k}, R_{V,k}, R_{W_k}$ (in a single iteration $k$), the angle $\phi$ is computed to maximize the Frobenius norm of the diagonal in each microiteration, and is generally different for each rotation. We now give the method to compute the desired angle. For a given pivot pair $(p,q)$, it is sufficient to look only at a $2\times 2\times 2$ subproblem. Let the subtensors $\widehat{\mathscr{A}}$ and $\widehat{\mathscr{S}}$ be as in (4.23) and (4.25), respectively. We observe the first microiteration (4.32) and calculate $R_{U,k}$. Looking at the mode-1 matricizations, we have

$$\widehat{B}_{(1)} = R_{U,k}^T \widehat{A}_{(1)}, \tag{4.35}$$

or element-wise,

$$\begin{bmatrix} b_{ppp} & b_{pqp} & b_{ppq} & b_{pqq} \\ b_{qpp} & b_{qqp} & b_{qpq} & b_{qqq} \end{bmatrix} = \begin{bmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} a_{ppp} & a_{pqp} & a_{ppq} & a_{pqq} \\ a_{qpp} & a_{qqp} & a_{qpq} & a_{qqq} \end{bmatrix}.$$

Rotation angle $\phi$ is chosen to maximize the function

$$g(\phi) = b_{ppp}^2 + b_{qqq}^2 = (\cos\phi a_{ppp} + \sin\phi a_{qpp})^2 + (-\sin\phi a_{pqq} + \cos\phi a_{qqq})^2. \tag{4.36}$$

The angle $\phi$ we want to find must satisfy $g'(\phi) = 0$, thus we get

$$\tan(2\phi) = \frac{2(a_{ppp}a_{qpp} - a_{pqq}a_{qqq})}{a_{ppp}^2 + a_{qqq}^2 - a_{pqq}^2 - a_{qpp}^2}. \tag{4.37}$$

There is no need to explicitly calculate $\phi$ for $R_{U,k}$. Actually, it is sufficient to find the sine and cosine of $\phi$. To do this efficiently, define

$$\lambda = 2(a_{ppp}a_{qpp} - a_{pqq}a_{qqq}) \cdot \operatorname{sign}\left(a_{ppp}^2 + a_{qqq}^2 - a_{pqq}^2 - a_{qpp}^2\right),$$
$$\mu = \left|a_{ppp}^2 + a_{qqq}^2 - a_{pqq}^2 - a_{qpp}^2\right|,$$

and $t = \tan\phi$. After some calculations we obtain the quadratic equation in $t$,

$$\lambda t^2 + 2\mu t - \lambda = 0,$$

with solutions

$$t_1 = \frac{-\mu + \sqrt{\mu^2+\lambda^2}}{\lambda}, \quad t_2 = \frac{-\mu - \sqrt{\mu^2+\lambda^2}}{\lambda}.$$

It is necessary to multiply both numerator and denominator of $t_1$ by $\mu + \sqrt{\mu^2+\lambda^2}$ to avoid the catastrophic cancellation,

$$t_1 = \frac{\lambda}{\mu + \sqrt{\mu^2+\lambda^2}}.$$

We get,

$$\cos\phi_i = \frac{1}{\sqrt{1+t_i^2}}, \quad \sin\phi_i = \frac{t_i}{\sqrt{1+t_i^2}} = t_i\cos\phi_i, \quad i = 1,2.$$

Finally, calculate both solutions and use the one that gives bigger value of the function (4.36). The other rotation angles are computed analogously, with respect to remaining mode-$m$ matricizations.

Although this algorithm converges in practice for every cyclic strategy, there is an additional condition for choosing pivot pairs that ensures convergence in theory. We choose a pair $(p_k, q_k) = (p, q)$ as the pivot pair only if at least one of the following conditions is satisfied:

$$|\langle \nabla_U f, U\dot{R}(p,q,0)\rangle| \geq \eta \|\nabla_U f\|_2, \tag{4.38}$$

$$|\langle \nabla_V f, V\dot{R}(p,q,0)\rangle| \geq \eta \|\nabla_V f\|_2, \tag{4.39}$$

$$|\langle \nabla_W f, W\dot{R}(p,q,0)\rangle| \geq \eta \|\nabla_W f\|_2, \tag{4.40}$$

where $0 < \eta \leq \frac{2}{n}$ and $\dot{R}(p,q,0) = \frac{\partial}{\partial\phi}R(p,q,\phi)\Big|_{\phi=0}$. These inequalities are called the *Łojasiewitz gradient inequalities* and are a commonly used tool in proving convergence of non-linear optimization algorithms [1, 5, 51], and specifically tensor decomposition algorithms [73]. In Algorithm 8 we give Begović Kovač's method for approximate orthogonal tensor diagonalization [8]. The convergence of Algorithm 8 is given in Theorem 4.3.1.

**Theorem 4.3.1.** Every accumulation point $(U, V, W)$ obtained by Algorithm 8 is a stationary point of the function $f$ defined by (4.22).

**Algorithm 8** Jacobi-type algorithm for the approximate tensor diagonalization [8]

**Input:** $\mathscr{A} \in \mathbb{R}^{n\times n\times n}$.

**Output:** orthogonal matrices $U, V, W$

$\mathscr{A}^{(0)} = \mathscr{A}$

$U_0 = V_0 = W_0 = I_n$

$k = 1$

**repeat**

Choose pivot pair $(p,q)$.

**if** $(p,q)$ satisfies (4.38) **then**

Find $\cos\phi_k$ and $\sin\phi_k$ for $R_{U,k}$.

$\mathscr{B} = \mathscr{A}^{(k-1)} \times_1 R_{U,k}^T$

$U^{(k)} = U^{(k-1)} R_{U,k}$

**end if**

**if** $(p,q)$ satisfies (4.39) **then**

Find $\cos\phi_k$ and $\sin\phi_k$ for $R_{V,k}$.

$\mathscr{C} = \mathscr{B} \times_2 R_{V,k}^T$

$V^{(k)} = V^{(k-1)} R_{V,k}$

**end if**

**if** $(p,q)$ satisfies (4.40) **then**

Find $\cos\phi_k$ and $\sin\phi_k$ for $R_{W,k}$.

$\mathscr{A}^{(k)} = \mathscr{C} \times_3 R_{W,k}^T$

$W^{(k)} = W^{(k-1)} R_{W,k}$

**end if**

**until** convergence

Next, we are focusing on the approximate diagonalization of a symmetric tensor. Previous algorithms are designed for a general real tensor without any specific structure. During one iteration, multiplying with different rotation matrices in each mode does not preserve symmetry. Therefore, we must make some adjustments to the problem statement. Recall that a symmetric tensor is invariant to index permutations and therefore all of its unfoldings are equal. Also, due to its symmetric structure, orthogonal matrices $U, V$ and $W$ in decomposition (4.21) must be equal,

$$\mathscr{A} = \mathscr{S} \times_1 U \times_2 U \times_3 U. \tag{4.41}$$

Consequently, objective function becomes simpler. The goal is to find orthogonal matrix $U$ which maximizes the function $f_s : O_n \to \mathbb{R}$, given as

$$f_s(U) = \|diag(\mathscr{A} \times_1 U^T \times_2 U^T \times_3 U^T)\|_F^2. \tag{4.42}$$

In the $k$th iteration of the symmetry preserving algorithm we have

$$\mathscr{A}^{(k)} = \mathscr{A}^{(k-1)} \times_1 (R_k)^T \times_2 (R_k)^T \times_3 (R_k)^T, \quad \mathscr{A}^{(0)} = \mathscr{A}. \tag{4.43}$$

where $R_k$ is a plane rotation chosen to maximize the Frobenius norm of the diagonal of $\mathscr{A}^{(k)}$. Again, matrices $R_k$ depend on the pivot pair and the rotation angle.

In [55], Li, Ushevich, and Comon give some convergence results regarding the uniqueness of the stationary point for the cyclic Jacobi algorithm given below in the Algorithm 9. It does not use the Łojasiewitz gradient inequality to decide on a pivot pair, but rather goes through all of the pairs in a row-wise manner. In this case we do not have the proof of convergence for Algorithm 9.

**Algorithm 9** Cyclic Jacobi algorithm

**Input:** $\mathscr{A} \in \mathbb{R}^{n\times n\times n}$ symmetric, $\delta_0 > 0$, $U_0 = I_n$

**Output:** orthogonal matrix $U$

$\mathscr{A}^{(0)} = \mathscr{A}$

$k = 1$

**repeat**

Choose pivot pair $(p_k, q_k)$ according to the row-wise cyclic strategy

Find $\theta_k$ that maximizes the function

$$g_s(\theta) = f_s(U^{(k-1)}R(p_k, q_k, \theta)) \tag{4.44}$$

Set $R_k = R(p_k, q_k, \theta_k)$

$\mathscr{A}^{(k)} = \mathscr{A}^{(k-1)} \times_1 R_k^T \times_2 R_k^T \times_3 R_k^T$

$U^{(k)} = U^{(k-1)}R_k$

**until** convergence

$U = U^{(k)}$

In [54], the same authors provided another Jacobi type algorithm for this problem, along with a convergence proof. Jacobi-PC algorithm is a cyclic Jacobi-type algorithm that uses a proximal term. Additional assumption on the smooth function $f_s$ is that it is periodic with period $\pi/2$, that is

$$f_s(UR(p,q,\theta)) = f_s(UR(p,q,\theta+\pi/2)).$$

The method is given in Algorithm 10. In each iteration, the angle $\theta_k$ which maximizes the function $\tilde{g}_s$ can be computed algebraically by solving a polynomial equation.

**Theorem 4.3.2** (Li, Ushevich, Comon [54])**.** Every sequence $U^{(k)}$, $k \geq 0$ generated by Algorithm 10 converges to a stationary point $U \in O_n$ for any starting point $U^{(0)}$.

**Algorithm 10** Jacobi-PC algorithm [54]

**Input:** $\mathscr{A} \in \mathbb{R}^{n\times n\times n}$ symmetric, $\delta_0 > 0$, $U_0 = I_n$

**Output:** orthogonal matrix $U$

$\mathscr{A}^{(0)} = \mathscr{A}$

$k = 1$

**repeat**

Choose pivot pair $(p_k, q_k)$ from some cyclic strategy

Find $\theta_k$ that maximizes the function

$$\tilde{g}_s(\theta) = f_s(U^{(k-1)}R(p_k, q_k, \theta)) - \delta_0\gamma(\theta),$$

where

$$\gamma(\theta) = 2sin^2(\theta)cos^2(\theta).$$

Set $R_k = R(p_k, q_k, \theta_k)$

$\mathscr{A}^{(k)} = \mathscr{A}^{(k-1)} \times_1 R_k^T \times_2 R_k^T \times_3 R_k^T$

$U^{(k)} = U^{(k-1)}R_k$

**until** convergence

$U = U^{(k)}$

When it comes to tensor diagonalization, the most general case is observed in [74] by the same authors. They design and prove the convergence of the gradient-based Jacobi-type algorithms for approximate diagonalization of a complex tensor on the unitary group $\mathscr{U}_n$. They consider a class of functions that generalize the joint approximate Hermitian diagonalization of tensors. For tensors $\mathscr{A}_l$ of order $d_l$, integers $t_l$, $0 \leq t_l \leq d_l$, and $\alpha_l \in \mathbb{R}$, $l = 1, 2, \ldots, L$, define the objective function

$$f_{\mathbb{C}}(U) = \sum_{l=1}^{L} \alpha_l \|diag(\mathscr{A}_l \times_1 U^H \cdots \times_{t_l} U^H \times_{t_l+1} U^T \cdots \times_{d_l} U^T)\|_F^2, \tag{4.45}$$

where $U^T$ and $U^H$ represent the transpose and Hermitian transpose of the matrix $U$, respectively. The conjugate transpose is applied only in the first $t_l$ modes, and in the remaining $d_l - t_l$ modes, the (non-conjugate) transpose is applied. To solve the joint approximate diagonalization of tensors $\mathscr{A}_l$, the goal is to maximize the function $f_{\mathbb{C}}$. This

general case, for $L = 1$ and $t_1 = 0$, also includes symmetric diagonalization problem covered before in (4.42). Using Łojasiewitz gradient inequality in their complex Jacobi-type algorithm enabled them to prove that every accumulation point $\overline{U}$ of the generated sequence $\{U^{(k)}\}_{k\geq 0}$ is a stationary point and they were able to obtain global convergence rates. Moreover, an accumulation point $\overline{U}$ is the limit point if it satisfies certain regularity conditions. In addition to that, the speed of convergence is linear if the Hessian at $\overline{U}$ has full rank.

# 4.4. TRACE MAXIMIZATION

We take another approach to approximately diagonalize a tensor. Inspired by the algorithm of Moravitz Martin and Van Loan [65], instead of maximizing the Frobenius norm of the diagonal, we maximize the trace of $\mathscr{S}$ using the ALS approach. That is, for a given tensor $\mathscr{A} \in \mathbb{R}^{n\times n\times\cdots\times n}$ of order $d \geq 3$, we are looking for its decomposition

$$\mathscr{S} = \mathscr{A} \times_1 U_1^T \times_2 U_2^T \cdots \times_n U_n^T, \tag{4.46}$$

such that the trace of the core tensor $\mathscr{S}$,

$$\operatorname{tr}(\mathscr{S}) = \sum_{i=1}^{d} \mathscr{S}_{ii\cdots i}, \tag{4.47}$$

is maximized.

Apart from the paper [65], trace maximization was addressed in [21] and [74]. Of those papers, only [74] offers the convergence proof for their algorithm, but exclusively for tensors of even order. Here we prove the convergence of our algorithm for tensors of order $d \geq 3$. The convergence results are analogous to those from [8, 54]. Since we are maximizing the trace, our objective function is different than those in [8, 54], and it is a function of $d$ variables because we are solving the problem for tensors of order $d$. In particular, we are going to prove that every accumulation point $(U_1, U_2, \ldots, U_d)$ obtained by our algorithm is a stationary point of the function $f$ defined by (4.48). Moreover, we adapt our trace maximization algorithm to obtain the structure-preserving algorithm for symmetric tensors. Such algorithm will no longer be an ALS algorithm, since we need to optimize over all modes at once, but the convergence theory will be along side the non-structured ALS algorithm.

## 4.4.1. Algorithm for the general non-structured tensors

Trace maximization of the tensor $\mathscr{S}$ from (4.46) is equivalent to the problem of finding $d$ orthogonal matrices $U_1, U_2, \ldots, U_d$ that maximize the objective function

$$f(U_1, U_2, \ldots, U_d) = \operatorname{tr}(\mathscr{A} \times_1 U_1^T \times_2 U_2^T \cdots \times_d U_d^T). \tag{4.48}$$

To solve this problem we develop a Jacobi-type algorithm using the ALS approach where each iteration contains $d$ microiterations. In one microiteration we fix $d-1$ matrices and

solve the optimization problem for only one matrix, i.e. we optimize in only one mode at a time. In the $k$th iteration of the iterative process (4.18) we apply $d$ plane rotations onto the underlying tensor $\mathscr{A}^{(k-1)}$, one in each mode. As before, each plane rotation depends on the pivot pair and the angle $\phi$. The pivot pair is the same for all matrices $R_{U_l,k}$, $l = 1,\ldots,d$, but the rotation angle is, in general, different for each of those matrices.

The results of $d$ microiterations building the $k$th iteration are denoted by $\mathscr{A}_l^{(k)}$, $l = 1,\ldots,d$. They are computed as

$$\mathscr{A}_l^{(k)} = \mathscr{A}_{l-1}^{(k)} \times_1 I \cdots \times_{l-1} I \times_l R_{U_l,k}^T \times_{l+1} I \cdots \times_d I, \quad l = 1,\ldots,d. \tag{4.49}$$

We set

$$\mathscr{A}_0^{(k)} = \mathscr{A}^{(k-1)}, \quad \mathscr{A}^{(k)} = \mathscr{A}_d^{(k)}.$$

Relations (4.49) can also be written as matrix products

$$(\mathscr{A}_l^{(k)})_{(l)} = R_{U_l,k}^T (\mathscr{A}_{l-1}^{(k)})_{(l)}, \quad l = 1,\ldots,d, \tag{4.50}$$

where each rotation $R_{U_l,k}$ changes only two rows in the corresponding mode-$l$ matricization $(\mathscr{A}_{l-1}^{(k)})_{(l)}$. This scheme is well defined because combining all microiterations (4.49) gives the iteration (4.18). Using the properties of mode-$m$ product (4.5) and (4.6) we get

$$\begin{aligned}\mathscr{A}^{(k)} = ((\mathscr{A}^{(k-1)} \times_1 R_{U_1,k}^T \times_2 I \cdots \times_d I) \times_1 I \times_2 R_{U_2,k}^T \times_3 I \cdots \times_d I) \cdots \\ \cdots \times_1 I \cdots \times_{d-1} I \times_d R_{U_d,k}^T = \mathscr{A}^{(k-1)} \times_1 R_{U_1,k}^T \times_2 R_{U_2,k}^T \cdots \times_d R_{U_d,k}^T.\end{aligned}$$

In the $k$th iteration of the algorithm we have tensor $\mathscr{A}^{(k)}$ and matrices $U_l^{(k)}$, $1 \le l \le d$. For the current pivot position $(p_k, q_k)$ we seek to find the rotation matrix $R_{U_1,k}$. Using $R_{U_1,k}$ we update the transformation matrix $U_1^{(k)}$ and form the auxiliary tensor $\mathscr{A}_1^{(k)}$,

$$\mathscr{A}_1^{(k)} = \mathscr{A}^{(k-1)} \times_1 R_{U_1,k}.$$

Since

$$\begin{aligned}\mathscr{A}^{(k-1)} \times_1 R_{U_1,k} &= (\mathscr{A} \times_1 (U_1^{(k-1)})^T \times_2 (U_2^{(k-1)})^T \cdots \times_d (U_d^{(k-1)})^T) \times_1 R_{U_1,k}^T \\ &= \mathscr{A} \times_1 \left(R_{U_1,k}^T (U_1^{(k-1)})^T\right) \times_2 (U_2^{(k-1)})^T \cdots \times_d (U_d^{(k-1)})^T,\end{aligned}$$

it follows that

$$U_1^{(k)} = U_1^{(k-1)} R_{U_1,k}.$$

We repeat the same computation for modes $l = 2, \dots d$, one by one, and do the updates

$$\mathscr{A}_l^{(k)} = \mathscr{A}_{l-1}^{(k)} \times_l R_{U_l,k}^T,$$
$$U_l^{(k)} = U_l^{(k-1)} R_{U_l,k}.$$

We still need to explain how we choose pivot positions $(p_k, q_k)$ and rotations $R_{U_1,k}$, $k \geq 0$. Regarding the choice of the pivot pairs, our algorithm uses cyclic pivot strategies. That means that we go through all possible pivot pairs $(p,q)$, $1 \leq p < q \leq n$, in some prescribed order, making a cycle, and then repeat that same cycle until convergence. As we are going to see in Subsection 4.4.3, the convergence results hold for any cyclic pivot strategy. Still, in order to ensure the convergence we need to set an additional condition (Łojasiewitz gradient inequality) — pivot pair $(p,q)$ must satisfy the condition

$$|\langle \nabla_{U_l} f(U_1, U_2, \dots, U_d), U_l \dot{R}(p,q,0) \rangle| \geq \eta \|\nabla_{U_l} f(U_1, U_2, \dots, U_d)\|_2, \tag{4.51}$$

for at least one mode $l$, $1 \leq l \leq d$, where $0 < \eta \leq \frac{2}{n}$ and $\dot{R}(p,q,0) = \frac{\partial}{\partial \phi} R(p,q,\phi)\Big|_{\phi=0}$. If a pair $(p,q)$ does not satisfy the condition (4.51) for any $l$, we move onto the next pair. Even though this condition may seem restrictive, we will show in Subsection 4.4.3 that for every $l = 1, \dots, d$, there exists at least one viable pivot pair. Thus, the algorithm will not stop because the condition (4.51) is not fulfilled.

Now, let us see how the rotation angles are calculated. We fix the index $k$ and assume that the pivot pair is $(p_k, q_k) = (p,q)$, $1 \leq p < q \leq n$. We observe an order-$d$ subtensor $\widehat{\mathscr{A}} \in \mathbb{R}^{2 \times 2 \times \cdots \times 2}$ of $\mathscr{A}$. We need to find $2 \times 2$ rotations $\widehat{R}_{U_l}$, $l = 1, \dots, d$, such that the trace of the subtensor

$$\widehat{\mathscr{S}} = \widehat{\mathscr{A}} \times_1 \widehat{R}_{U_1}^T \times_2 \widehat{R}_{U_2}^T \cdots \times_d \widehat{R}_{U_d}^T$$

is maximized. To this end we use mode-$l$ matricizations from (4.50). This gives

$$(\widehat{\mathscr{A}_l})_{(l)} = \widehat{R}_{U_l}^T (\widehat{\mathscr{A}_{l-1}})_{(l)}, \quad l = 1, \dots, d. \tag{4.52}$$

Since the mode-$l$ matricization is obtained by arranging all mode-$l$ fibers into columns, elements in the same column have all indices the same except the $l$th one. Therefore, relation (4.52) can be written as

$$\begin{bmatrix} a_{p\dots p}^{(l)} & \cdots & a_{q\dots qpq\dots q}^{(l)} \\ a_{p\dots pqp\dots p}^{(l)} & \cdots & a_{q\dots q}^{(l)} \end{bmatrix} = \begin{bmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} a_{p\dots p}^{(l-1)} & \cdots & a_{q\dots qpq\dots q}^{(l-1)} \\ a_{p\dots pqp\dots p}^{(l-1)} & \cdots & a_{q\dots q}^{(l-1)} \end{bmatrix},$$

where in matrices $(\widehat{\mathscr{A}_{l-1}})_{(l)}$ and $(\widehat{\mathscr{A}_l})_{(l)}$ the elements at the position $(2,1)$ have the $l$th index equal to $q$ and the elements at the position $(1,d)$ have the $l$th index equal to $p$. In order to maximize the trace of $(\widehat{\mathscr{A}_l})_{(l)}$ we define the function

$$\begin{aligned} g_l(\phi) &= \operatorname{tr}((\widehat{\mathscr{A}_l})_{(l)}) = a^{(l)}_{p\dots p} + a^{(l)}_{q\dots q} \\ &= (\cos\phi\, a^{(l-1)}_{p\dots p} + \sin\phi\, a^{(l-1)}_{p\dots pqp\dots p}) + (-\sin\phi\, a^{(l-1)}_{q\dots qpq\dots q} + \cos\phi\, a^{(l-1)}_{q\dots q}). \end{aligned} \tag{4.53}$$

Setting the derivative of $g_l$ to zero leads to the equation

$$\begin{aligned} 0 = g_l{}'(\phi) &= -\sin\phi\, a^{(l-1)}_{p\dots p} + \cos\phi\, a^{(l-1)}_{p\dots pqp\dots p} - \cos\phi\, a^{(l-1)}_{q\dots qpq\dots q} - \sin\phi\, a^{(l-1)}_{q\dots q} \\ &= -\sin\phi\,(a^{(l-1)}_{p\dots p} + a^{(l-1)}_{q\dots q}) + \cos\phi\,(a^{(l-1)}_{p\dots pqp\dots p} - a^{(l-1)}_{q\dots qpq\dots q}). \end{aligned}$$

By rearranging this equation and dividing it by $\cos\phi$ we get the formula for the tangent of the rotation angle

$$\tan\phi = \frac{a^{(l-1)}_{p\dots pqp\dots p} - a^{(l-1)}_{q\dots qpq\dots q}}{a^{(l-1)}_{p\dots p} + a^{(l-1)}_{q\dots q}}. \tag{4.54}$$

This procedure is the same for all $l = 1,2,\dots,d$.

The explicit angles for $\widehat{R}^T_{U_1},\dots,\widehat{R}^T_{U_d}$ are not needed in order to perform the transformations (4.49). We only need sine and cosine of the corresponding angles. We compute those from (4.54) using the transformation formulas

$$\cos\phi_i = \frac{1}{\pm\sqrt{1+\tan^2\phi}}, \quad \sin\phi_i = \frac{t}{\pm\sqrt{1+\tan^2\phi}} = \tan\phi\cos\phi_i, \quad i = 1,2. \tag{4.55}$$

We calculate both solutions and take the one that gives a bigger value of the function $g_l$ from (4.53). Notice that both function values will have the same absolute value but a different sign because $\sin\phi_2 = -\sin\phi_1$ and $\cos\phi_2 = -\cos\phi_1$. Therefore, we can take the angle $\phi_i$, $i = 1,2$, which gives a positive value of the function $g_l$.

We sum up this subsection in Algorithm 11.

Input arguments in Algorithm 11 are the initial tensor $\mathscr{A}^{(0)}$ and the starting approximations of the orthogonal transformations $U^{(0)}_l$, $1 \le l \le d$. A simple starting point is to set $\mathscr{A}^{(0)} = \mathscr{A}$, and take $U^{(0)}_l$ as identity matrices,

$$\mathscr{A}^{(0)} = \mathscr{A}, \quad U^{(0)}_l = I, \quad 1 \le l \le d. \tag{4.56}$$

We call (4.56) an identity initialization. It works very well in most of the cases.

**Algorithm 11** Tensor-trace maximization

**Input:** $\mathscr{A}^{(0)} \in \mathbb{R}^{n\times n\times\cdots\times n}$, $U_l^{(0)} \in \mathbb{R}^{n\times n}$, $l = 1,\ldots,d$

**Output:** orthogonal matrices $U_l$, $l = 1,\ldots,d$

$k = 1$

**repeat**

Choose pivot pair $(p,q)$.

$\mathscr{A}_0^{(k)} = \mathscr{A}^{(k-1)}$

**for** l=1:d **do**

**if** $(p,q)$ satisfies (4.51) for $l$ **then**

Find $\cos\phi_k$ and $\sin\phi_k$ for $R_{U_l,k}$ using (4.54).

$\mathscr{A}_l^{(k)} = \mathscr{A}_{l-1}^{(k)} \times_l R_{U_l,k}^T$

$U_l^{(k)} = U_l^{(k-1)} R_{U_l,k}$

**end if**

**end for**

$\mathscr{A}^{(k)} = \mathscr{A}_d^{(k)}$

**until** convergence

However, identity initialization is not an option if, for example, $\mathscr{A}$ is an antisymmetric tensor. Recall that the only non-trivial elements of an antisymmetric tensor are the ones with all indices different. Therefore, in equation (4.54) for the tangent of the rotation angle, both the numerator and the denominator are equal to zero, and the algorithm fails. That is why we must use a different initialization. One solution to this problem is to precondition the tensor $\mathscr{A}$ using the HOSVD [22]. We have

$$\mathscr{A} = \widetilde{\mathscr{S}} \times_1 \widetilde{U}_1 \times_2 \widetilde{U}_2 \cdots \times_d \widetilde{U}_d,$$

where $\widetilde{U}_l$ are matrices of left singular vectors of matricizations $A_{(l)}$, $1 \le l \le d$. Then, the HOSVD initialization is given by

$$\begin{aligned} \mathscr{A}^{(0)} &= \mathscr{A} \times_1 \widetilde{U}_1^T \times_2 \widetilde{U}_2^T \cdots \times_d \widetilde{U}_d^T, \\ U_l^{(0)} &= \widetilde{U}_l, \quad 1 \le l \le d. \end{aligned} \tag{4.57}$$

In Section 4.5 we will further discuss these two initializations.

### 4.4.2. Structure-preserving algorithm for the symmetric tensors

Algorithm 11 does not preserve the tensor structure since it applies different rotations in different modes. Still, it can be modified to preserve the symmetry of the starting tensor by setting the transformation matrices from (4.12) to be the same in each mode. That means that now, for a symmetric tensor $\mathscr{A}$, we are looking for the decomposition of the form

$$\mathscr{A} = \mathscr{S} \times_1 U \times_2 U \cdots \times_d U, \tag{4.58}$$

where $U$ is orthogonal.

As we did before, we can write the core tensor $\mathscr{S}$ as

$$\mathscr{S} = \mathscr{A} \times_1 U^T \times_2 U^T \cdots \times_d U^T.$$

Thus, for a symmetric tensor $\mathscr{A}$ we need to find the orthogonal matrix $U$ that maximizes the objective function

$$f_s(U) = \operatorname{tr}(\mathscr{A} \times_1 U^T \times_2 U^T \cdots \times_d U^T). \tag{4.59}$$

Now, in the $k$th iteration of the algorithm we have

$$\mathscr{A}^{(k)} = \mathscr{A}^{(k-1)} \times_1 R_k^T \times_2 R_k^T \cdots \times_d R_k^T, \quad k \geq 0, \quad \mathscr{A}^{(0)} = \mathscr{A}, \tag{4.60}$$

where $R_k$ is a plane rotation of the form (4.19). It is interesting to notice that in the matrix case, $d = 2$, the trace would remain constant throughout the iterations (4.60) because

$$\operatorname{tr}(U^T A U) = \operatorname{tr}(A),$$

but that is not the case for the higher order tensors.

Rotations $R_k$ depend on the pivot position and the rotation angle. Pivot positions are chosen in any cyclic order, same as in Algorithm 11, with the condition that the pair $(p,q)$ is taken as a pivot pair if it satisfies the inequality

$$|\langle \nabla f_s(U), U\dot{R}(p,q,0) \rangle| \geq \eta \|\nabla f_s(U)\|_2, \tag{4.61}$$

which is analogous to the condition (4.51).

When choosing the rotation angle, we now need to consider all modes at once. Hence, this is not an ALS algorithm. Because of that, the formula for the tangent of the rotation

angle is more complicated than the one from (4.54). We get a polynomial equation in $\tan\phi$, where the order of the polynomial is equal to the order of the tensor $d$. Here, we derive such equation for $d = 3$ and $d = 4$. This calculation follows the same steps for $d > 4$.

Let $d = 3$. Again, we observe a two-dimensional subproblem

$$\widehat{\mathscr{A}} = \widehat{\mathscr{S}} \times_1 R \times_2 R \times_3 R,$$

for a fixed pivot pair $(p,q)$, where

$$\widehat{\mathscr{A}}(:,:,1) = \begin{bmatrix} a_{ppp} & a_{pqp} \\ a_{qpp} & a_{qqp} \end{bmatrix}, \quad \widehat{\mathscr{A}}(:,:,2) = \begin{bmatrix} a_{ppq} & a_{pqq} \\ a_{qpq} & a_{qqq} \end{bmatrix},$$

and

$$R = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix}.$$

We choose the angle $\phi$ that maximizes the function

$$g_s(\phi) = \operatorname{tr}(\widehat{\mathscr{S}}) = \operatorname{tr}(\widehat{\mathscr{A}} \times_1 R^T \times_2 R^T \times_3 R^T).$$

Using the fact that $\widehat{\mathscr{A}}$ is a symmetric tensor, function $g_s$ can be written as

$$\begin{aligned} g_s(\phi) &= \cos^3\phi a_{ppp} + 2\cos^2\phi\sin\phi a_{ppq} + \cos\phi\sin^2\phi a_{pqq} + \cos^2\phi\sin\phi a_{ppq} \\ &\quad + 2\cos\phi\sin^2\phi a_{pqq} + \sin^3\phi a_{qqq} - \cos^2\phi\sin\phi a_{pqq} + 2\cos\phi\sin^2\phi a_{ppq} \\ &\quad - \sin^3\phi a_{ppp} + \cos^3\phi a_{qqq} - 2\cos^2\phi\sin\phi a_{pqq} + \cos\phi\sin^2\phi a_{ppq} \\ &= \cos^3\phi(a_{ppp} + a_{qqq}) + 3\cos^2\phi\sin\phi(a_{ppq} - a_{pqq}) \\ &\quad + 3\cos\phi\sin^2\phi(a_{ppq} + a_{pqq}) + \sin^3\phi(a_{qqq} - a_{ppp}). \end{aligned}$$

We have

$$\begin{aligned} 0 = g_s'(\phi) &= 3\cos^3\phi(a_{ppq} - a_{pqq}) + 3\cos^2\phi\sin\phi(2a_{ppq} + 2a_{pqq} - a_{ppp} - a_{qqq}) \\ &\quad + 3\cos\phi\sin^2\phi(a_{qqq} - a_{ppp} - 2a_{ppq} + 2a_{pqq}) - 3\sin^3\phi(a_{ppq} + a_{pqq}). \end{aligned}$$

Dividing this equation by $-3\cos^3\phi$ we obtain the cubic equation for $t = \tan\phi$,

$$\begin{aligned} &(a_{ppq} + a_{pqq})t^3 + (a_{ppp} - a_{qqq} + 2a_{ppq} - 2a_{pqq})t^2 \\ &\qquad + (a_{ppp} + a_{qqq} - 2a_{ppq} - 2a_{pqq})t + (a_{pqq} - a_{ppq}) = 0. \end{aligned} \quad (4.62)$$

Now, let $d = 4$. The two-dimensional subproblem is in the form of

$$\widehat{\mathscr{A}} = \widehat{\mathscr{S}} \times_1 R \times_2 R \times_3 R \times_4 R,$$

for a fixed pivot pair $(p,q)$, where

$$\widehat{\mathscr{A}}(:,:,1,1) = \begin{bmatrix} a_{pppp} & a_{pqpp} \\ a_{qppp} & a_{qqpp} \end{bmatrix}, \quad \widehat{\mathscr{A}}(:,:,2,1) = \begin{bmatrix} a_{ppqp} & a_{pqqp} \\ a_{qpqp} & a_{qqqp} \end{bmatrix},$$

$$\widehat{\mathscr{A}}(:,:,1,2) = \begin{bmatrix} a_{pppq} & a_{pqpq} \\ a_{qppq} & a_{qqpq} \end{bmatrix}, \quad \widehat{\mathscr{A}}(:,:,2,2) = \begin{bmatrix} a_{ppqq} & a_{pqqq} \\ a_{qpqq} & a_{qqqq} \end{bmatrix},$$

and

$$R = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix}.$$

We choose the angle $\phi$ that maximizes the function

$$g_s(\phi) = \operatorname{tr}(\widehat{\mathscr{S}}) = \operatorname{tr}(\widehat{\mathscr{A}} \times_1 R^T \times_2 R^T \times_3 R^T \times_4 R^T).$$

Using the symmetric property of $\widehat{\mathscr{A}}$, function $g_s$ becomes

$$\begin{aligned} g_s(\phi) = {} & \cos^4\phi(a_{pppp} + a_{qqqq}) - 4\cos^3\phi\sin\phi(a_{pqqq} - a_{pppq}) \\ & + 12\cos^2\phi\sin^2\phi a_{ppqq} + 4\cos\phi\sin^3\phi(a_{pqqq} - a_{pppq}) + \sin^4\phi(a_{pppp} + a_{qqqq}). \end{aligned}$$

Differentiating over $\phi$ and setting the derivative to zero yields

$$\begin{aligned} 0 = g_s'(\phi) = {} & -4\cos^4\phi(a_{pqqq} - a_{pppq}) + 4\cos^3\phi\sin\phi(6a_{ppqq} - a_{pppp} - a_{qqqq}) \\ & + 24\cos^2\phi\sin^2\phi(a_{pqqq} - a_{pppq}) - 4\cos\phi\sin^3\phi(6a_{ppqq} - a_{pppp} - a_{qqqq}) \\ & - 4\sin^4\phi(a_{pqqq} - a_{pppq}). \end{aligned}$$

Dividing this equation by $-4\cos^4\phi$ we obtain the quartic equation for $t = \tan\phi$,

$$\begin{aligned} & (a_{pqqq} - a_{pppq})t^4 - (a_{pppp} + a_{qqqq} - 6a_{ppqq})t^3 - 6(a_{pqqq} - a_{pppq})t^2 \\ & \qquad + (a_{pppp} + a_{qqqq} - 6a_{ppqq})t + (a_{pqqq} - a_{pppq}) = 0. \end{aligned} \tag{4.63}$$

Depending on the order $d$ of the tensor $\mathscr{A}$, we solve equations (4.62) or (4.63) for $t$ and calculate $\cos\phi$ and $\sin\phi$ using the formulas (4.55). Then we take a real solution that gives the highest value of the function $g_s$. From a theoretical point of view, solutions of (4.62) and (4.63) can be calculated using a rather complicated formula for the roots of the general cubic/quatric equation. In practice, we use Matlab function `roots`.

**Algorithm 12** Symmetry-preserving tensor-trace maximization

**Input:** $\mathscr{A} \in \mathbb{R}^{n \times n \times \cdots \times n}$ symmetric

**Output:** orthogonal matrix $U$

$\mathscr{A}^{(0)} = \mathscr{A}$

$U^{(0)} = I_n$

$k = 1$

**repeat**

Choose pivot pair $(p,q)$.

**if** $(p,q)$ satisfies (4.61) **then**

Find $\cos\phi_k$ and $\sin\phi_k$ for $R_k$ using polynomial equation

$\mathscr{A}^{(k)} = \mathscr{A}^{(k-1)} \times_1 R_k^T \times_2 R_k^T \cdots \times_d R_k^T$

$U^{(k)} = U^{(k-1)} R_k$

**end if**

**until** convergence

This calculation follows the same steps for $d > 4$. The complete procedure using the identity initialization is given in Algorithm 12.

We have observed one intriguing thing. Instead of Algorithm 12 for symmetric tensors, one can take its modification where the rotation angle is chosen as the optimal angle in only one (e.g. first) mode. On the contrary, in the computation of (4.62) and (4.63) when choosing the optimal angle we considered all modes at once. The advantage when optimizing the angle in only one mode is that the computation is much simpler, we get a linear equation in $\tan\phi$, the same as in (4.54). The modification is given in Algorithm 13. Our convergence proof is valid only if the rotation angle is optimal regarding all modes at once, but the modified algorithm has some interesting properties that can be seen in Figures 4.11 and 4.12 in Section 4.5. Note that Algorithm 13 does not lead to the same process as Algorithm 11, because it still applies the same rotation in all modes, unlike the Algorithm 11 when applied to a symmetric tensor.

**Algorithm 13** Mode-1 modification of Algorithm 12

**Input:** $\mathscr{A} \in \mathbb{R}^{n\times n\times\cdots\times n}$ symmetric

**Output:** orthogonal matrix $U$

$\mathscr{A}^{(0)} = \mathscr{A}$

$U^{(0)} = I_n$

$k = 1$

**repeat**

Choose pivot pair $(p,q)$.

**if** $(p,q)$ satisfies (4.61) **then**

Find $\cos\phi_k$ and $\sin\phi_k$ for $R_k$ using (4.54) for $l = 1$,

$$\tan\phi_k = \frac{a^{(k-1)}_{qp\ldots p} - a^{(k-1)}_{q\ldots qp}}{a^{(k-1)}_{p\ldots p} + a^{(k-1)}_{q\ldots q}}.$$

$\mathscr{A}^{(k)} = \mathscr{A}^{(k-1)} \times_1 R_k^T \times_2 R_k^T \cdots \times_d R_k^T$

$U^{(k)} = U^{(k-1)} R_k$

**end if**

**until** convergence

### 4.4.3. Convergence of the tensor-trace maximization algorithm

The convergence of the new algorithms is analogous to the convergence results from [8, 54]. Compared to the algorithms where the squares of the diagonal elements are maximized, maximization of the trace leads to a simpler algorithm. In this section we are going to show that Algorithm 11 and Algorithm 12 converge to the stationary points of the objective functions (4.48) and (4.59), respectively. The proofs follow the basic idea from the paper [45] that was adopted in [8].

First, we define the function $\tilde{f}\colon \mathbb{R}^{n\times n}\times\mathbb{R}^{n\times n}\times\cdots\times\mathbb{R}^{n\times n}\to\mathbb{R}$,

$$\begin{aligned}\tilde{f}(U_1,U_2,\ldots,U_d) &= \operatorname{tr}(\mathscr{A}\times_1 U_1^T\times_2 U_2^T\cdots\times_d U_d^T) \\ &= \sum_{r=1}^{n}\left(\sum_{i_1,\ldots,i_d=1}^{n} a_{i_1 i_2\ldots i_d} u_{i_1 r,(1)} u_{i_2 r,(2)}\ldots u_{i_d r,(d)}\right).\end{aligned} \tag{4.64}$$

Function $\tilde{f}$ is the extension of the objective function $f$ from (4.48) to the set of all square matrices. We calculate $\nabla_{U_l}\tilde{f}$, $1 \leq l \leq d$, element-wise as

$$\begin{aligned}\frac{\partial \tilde{f}}{\partial u_{mr,(l)}} &= \sum_{i_1,\ldots,i_{l-1},i_{l+1},\ldots,i_d=1}^{n} a_{i_1\ldots i_{l-1}mi_{l+1}\ldots i_d} u_{i_1 r,(1)} \cdots u_{i_{l-1}r,(l-1)} u_{i_{l+1}r,(l+1)} \cdots u_{i_d r,(d)} \\ &= (\mathscr{A} \times_1 U_1^T \cdots \times_{l-1} U_{l-1}^T \times_{l+1} U_{l+1}^T \cdots \times_d U_d^T)_{r\ldots rmr\ldots r}.\end{aligned}$$

Firstly, we compute $\nabla\tilde{f}(U_1,U_2,\ldots,U_d)$. Then, in order to get $\nabla f(U_1,U_2,\ldots,U_d)$, we project $\nabla\tilde{f}$ onto the tangent space at $(U_1,U_2,\ldots,U_d)$ to the manifold $O_n \times O_n \times \cdots \times O_n$. In the simpler case, for $d = 1$, the tangent space at $U$ to $O_n$ is of the form [2]

$$T_U O_n = \{U\Omega : \Omega^T = -\Omega\} = U\boldsymbol{\mathcal{S}}_{\text{skew}}(n),$$

where $\boldsymbol{\mathcal{S}}_{\text{skew}}(n)$ is the set of all $n \times n$ skew-symmetric matrices. The projection of $\nabla_U\tilde{f}$ onto the tangent space at $U$ to $O_n$ is

$$\text{Proj}(\nabla_U\tilde{f}) = U\Lambda(U),$$

where the operator $\Lambda\colon O_n \to \boldsymbol{\mathcal{S}}_{\text{skew}}(n)$, is defined as

$$\Lambda(U) := \frac{U^T\nabla_U\tilde{f} - (\nabla_U\tilde{f})^T U}{2}. \tag{4.65}$$

Then, we have,

$$\begin{aligned}\nabla f(U_1,U_2,\ldots,U_d) &= \begin{bmatrix} \nabla_{U_1} f(U_1,U_2,\ldots,U_d) & \cdots & \nabla_{U_d} f(U_1,U_2,\ldots,U_d) \end{bmatrix} \\ &= \text{Proj}\begin{bmatrix} \nabla_{U_1}\tilde{f}(U_1,U_2,\ldots,U_d) & \cdots & \nabla_{U_d}\tilde{f}(U_1,U_2,\ldots,U_d) \end{bmatrix} \\ &= \begin{bmatrix} U_1\Lambda(U_1) & \cdots & U_d\Lambda(U_d) \end{bmatrix}.\end{aligned}$$

Using the operator $\Lambda$, we can simplify the convergence condition (4.51). For $1 \leq l \leq d$, we have

$$\|\nabla_{U_l} f(U_1,U_2,\ldots,U_d)\|_2 = \|U_l\Lambda(U_l)\|_2 = \|\Lambda(U_l)\|_2,$$

and

$$\langle \nabla_{U_l} f(U_1,U_2,\ldots,U_d), U_l\dot{R}(p,q,0)\rangle = \langle U_l\Lambda(U_l), U_l\dot{R}(p,q,0)\rangle = \langle \Lambda(U_l), \dot{R}(p,q,0)\rangle.$$

Since

$$\dot{R}(p,q,0) = \begin{bmatrix} 0 & & & & & & \\ & \ddots & & & & & \\ & & 0 & & -1 & & \\ & & & \ddots & & & \\ & & 1 & & 0 & & \\ & & & & & \ddots & \\ & & & & & & 0 \end{bmatrix} \begin{matrix} \\ \\ p \\ \\ q \\ \\ \\ \end{matrix},$$

and $\Lambda(U_l)$ is a skew-symmetric matrix, we get

$$\langle \Lambda(U_l), \dot{R}(p,q,0) \rangle = -2\Lambda(U_l)_{pq}.$$

Therefore, the condition (4.51) can be written as

$$2|\Lambda(U_l)_{pq}| \geq \eta \|\Lambda(U_l)\|_2. \tag{4.66}$$

Now it is easy to prove that it is always possible to find a pivot pair that satisfies the convergence condition. Lemma 4.4.1 is a straightforward generalization of Lemma 3.2 from [8].

**Lemma 4.4.1.** For any differentiable function $f\colon O_n \times O_n \times \cdots \times O_n \to \mathbb{R}$, orthogonal matrices $U_1, U_2, \ldots, U_d \in O_n$, and $0 < \eta \leq \frac{2}{n}$ it is always possible to find index pairs $(p_{U_l}, q_{U_l})$, $1 \leq l \leq d$, such that

$$|\langle \nabla_{U_l} f(U_1, U_2, \ldots, U_d), U_l \dot{R}(p_{U_l}, q_{U_l}, 0) \rangle| \geq \eta \|\nabla_{U_l} f(U_1, U_2, \ldots, U_d)\|_2,$$

where $\dot{R}(p,q,0) = \frac{\partial}{\partial \phi} R(p,q,\phi)\Big|_{\phi=0}$.

*Proof.* For any $l = 1, \ldots, d$ it is always possible to find an index pair $(p,q)$ such that

$$|\Lambda(U_l)_{pq}| \geq \frac{1}{n} \|\Lambda(U_l)\|_2.$$

For $\eta = \frac{2}{n}$, we get

$$2|\Lambda(U_l)_{p_{U_l} q_{U_l}}| \geq \eta \|\Lambda(U_l)\|_2,$$

that is, inequality (4.66) is satisfied for $(p_{U_l}, q_{U_l}) = (p,q)$. As inequalities (4.66) and (4.51) are equivalent, this proves the lemma. ■

In Lemma 4.4.2 we show that if $(U_1, U_2, \ldots, U_d)$ is not a stationary point of the function $f$, then applying one step of the Algorithm 11 to any point in the small enough neighbourhood of $(U_1, U_2, \ldots, U_d)$ would increase the value of $f$. The proof of Lemma 4.4.2 follows the steps of the proof of Lemma 3.4 from [8].

**Lemma 4.4.2.** Let $\{U_l^{(k)}\}_{k\geq 0}$, $1 \leq l \leq d$, be the sequences generated by Algorithm 11. Let $\overline{U}_1, \overline{U}_2, \ldots, \overline{U}_d$ be a $d$-tuple of orthogonal matrices satisfying $\nabla f(\overline{U}_1, \overline{U}_2, \ldots, \overline{U}_d) \neq 0$. Then there exist $\varepsilon > 0$ and $\delta > 0$ such that

$$\|U_l^{(k-1)} - \overline{U}_l\|_2 < \varepsilon, \quad \forall l = 1, \ldots, d,$$

implies

$$f(U_1^{(k)}, U_2^{(k)}, \ldots, U_d^{(k)}) - f(U_1^{(k-1)}, U_2^{(k-1)}, \ldots, U_d^{(k-1)}) \geq \delta. \tag{4.67}$$

*Proof.* Here we denote $R_{l,k} = R(p_k, q_k, \phi_{U_l,k})$. For a fixed iteration $k$ we define $d$ functions $h_k^{(l)} \colon \mathbb{R} \to \mathbb{R}$, $l = 1, 2, \ldots, d$, as

$$\begin{aligned} h_k^{(1)}(\phi_1) &= f(U_1^{(k-1)} R(p_k, q_k, \phi_1), U_2^{(k-1)}, \ldots, U_d^{(k-1)}), \\ h_k^{(d)}(\phi_d) &= f(U_1^{(k-1)} R_{1,k}, U_2^{(k-1)} R_{2,k}, \ldots, U_{d-1}^{(k-1)} R_{d-1,k}, U_d^{(k-1)} R(p_k, q_k, \phi_d)), \end{aligned}$$

and

$$h_k^{(l)}(\phi_l) = f(U_1^{(k-1)} R_{1,k}, \ldots, U_{l-1}^{(k-1)} R_{l-1,k}, U_l^{(k-1)} R(p_k, q_k, \phi_l), U_{l+1}^{(k-1)}, \ldots, U_d^{(k-1)}),$$

for $2 \leq l \leq d-1$. The rotation angle in Algorithm 11 is chosen such that

$$\max_{\phi_l} h_k^{(l)}(\phi_l) = h_k^{(l)}(\phi_{U_l,k}) = f(U_1^{(k-1)} R_{1,k}, \ldots, U_l^{(k-1)} R_{l,k}, U_{l+1}^{(k-1)}, \ldots, U_d^{(k-1)}), \quad 1 \leq l \leq d.$$

Moreover, we know that after each microiteration $l$ in the Algorithm 11 the value of the objective function $f$ does not decrease, that is,

$$\begin{aligned} f(U_1^{(k)}, U_2^{(k)}, \ldots, U_d^{(k)}) &\geq f(U_1^{(k)}, U_2^{(k)}, \ldots, U_{d-1}^{(k)}, U_d^{(k-1)}) \\ &\geq \cdots \geq f(U_1^{(k)}, U_2^{(k-1)}, \ldots, , U_d^{(k-1)}) \\ &\geq f(U_1^{(k-1)}, U_2^{(k-1)}, \ldots, U_d^{(k-1)}). \end{aligned} \tag{4.68}$$

To prove the inequality (4.67) we need at least one sharp inequality in (4.68).

Since $\nabla f(\overline{U}_1, \overline{U}_2, \ldots, \overline{U}_d) \neq 0$, we have

$$\nabla_{U_l} f(\overline{U}_1, \overline{U}_2, \ldots, \overline{U}_d) \neq 0,$$

for at least one partial gradient $\nabla_{U_l} f$, $1 \leq l \leq d$. Let us assume that $m$, $1 \leq m \leq d$, is the smallest index such that

$$\nabla_{U_m} f(\overline{U}_1, \overline{U}_2, \ldots, \overline{U}_d) \neq 0. \tag{4.69}$$

Since $f$ is $C^\infty$, from the relation (4.69) it follows that there exists $\varepsilon > 0$ such that

$$\mu := \min\{\|\nabla_{U_m} f(U_1, U_2, \ldots, U_d)\|_2 : \ \|U_m - \overline{U}_m\|_2 < \varepsilon\} > 0. \tag{4.70}$$

Let $U_m^{(k-1)}$ be such that $\|U_m^{(k-1)} - \overline{U}_m\|_2 < \varepsilon$, and $U_l^{(k)}$, $1 \leq l \leq d$, generated by one iteration from the Algorithm 11. Then

$$\begin{aligned} & f(U_1^{(k)}, U_2^{(k)}, \ldots, U_d^{(k)}) - f(U_1^{(k-1)}, U_2^{(k-1)}, \ldots, U_d^{(k-1)}) \\ & \quad \geq f(U_1^{(k)}, \ldots, U_m^{(k)}, U_{m+1}^{(k-1)}, \ldots, U_d^{(k-1)}) - f(U_1^{(k)}, \ldots, U_{m-1}^{(k)}, U_m^{(k-1)}, \ldots, U_d^{(k-1)}) \\ & \quad \geq h_k^{(m)}(\phi_m) - h_k^{(m)}(0), \end{aligned} \tag{4.71}$$

is true for any angle $\phi_m$. We are going to find a particular $\phi_m$ that will ensure that $h_k^{(m)}(\phi_m) - h_k^{(m)}(0) > 0$. We need the Taylor expansion of the function $h_k^{(m)}$ around 0. It is given by

$$h_k^{(m)}(\phi_m) = h_k^{(m)}(0) + (h_k^{(m)})'(0)\phi_m + \frac{1}{2}(h_k^{(m)})''(\xi)\phi_m^2, \quad 0 < \xi < \phi_m. \tag{4.72}$$

Denote

$$M = \max_{0<\xi<\phi_m} |(h_k^{(m)})''(\xi)| < \infty.$$

Then we can write the Taylor expansion (4.72) as

$$h_k^{(m)}(\phi_m) - h_k^{(m)}(0) \geq (h_k^{(m)})'(0)\phi_m - \frac{1}{2}M\phi_m^2. \tag{4.73}$$

Therefore, using relations (4.71) and (4.73) we obtain

$$f(U_1^{(k)}, U_2^{(k)}, \ldots, U_d^{(k)}) - f(U_1^{(k-1)}, U_2^{(k-1)}, \ldots, U_d^{(k-1)}) \geq (h_k^{(m)})'(0)\phi_m - \frac{1}{2}M\phi_m^2. \tag{4.74}$$

The derivative $(h_k^{(m)})'(\phi_m)$ is calculated as

$$\langle \nabla_{U_m} f(U_1^{(k)}, \ldots, U_{m-1}^{(k)}, U_m^{(k-1)} R(p_k, q_k, \phi_m), U_{m+1}^{(k-1)}, \ldots, U_d^{(k-1)}), U_m^{(k-1)} \dot{R}(p_k, q_k, \phi_m) \rangle.$$

From the fact that $R(p_k, q_k, 0) = I$, we get the value of $(h_k^{(m)})'$ at $\phi_m = 0$,

$$(h_k^{(m)})'(0) = \langle \nabla_{U_m} f(U_1^{(k)}, \ldots, U_{m-1}^{(k)}, U_m^{(k-1)}, \ldots, U_d^{(k-1)}), U_m^{(k-1)} \dot{R}(p_k, q_k, 0) \rangle. \quad (4.75)$$

Hence, Lemma 4.4.1 and equation (4.75) imply

$$|(h_k^{(m)})'(0)| \geq \eta \|\nabla_{U_m} f(U_1^{(k)}, \ldots, U_{m-1}^{(k)}, U_m^{(k-1)}, \ldots, U_d^{(k-1)})\|_2. \quad (4.76)$$

Relation (4.70) together with the inequality (4.76) gives the lower bound on $|(h_k^{(m)})'(0)|$,

$$|(h_k^{(m)})'(0)| \geq \eta\mu > 0. \quad (4.77)$$

Finally, we go back to inequality (4.74). For $\phi_m = \frac{1}{M}(h_k^{(m)})'(0)$, using the relation (4.77), we get

$$\begin{aligned} f(U_1^{(k)}, U_2^{(k)}, \ldots, U_d^{(k)}) - f(U_1^{(k-1)}, U_2^{(k-1)}, \ldots, U_d^{(k-1)}) \\ \geq \frac{1}{M}((h_k^{(m)})'(0))^2 - \frac{1}{2M}((h_k^{(m)})'(0))^2 \\ \geq \frac{1}{2M}\eta^2\mu^2 = \delta > 0. \end{aligned}$$

■

Using Lemma 4.4.2 we are going to prove that Algorithm 11 converges to a stationary point of the objective function.

**Theorem 4.4.3.** Every accumulation point $(\overline{U}_1, \overline{U}_2, \ldots, \overline{U}_d)$ obtained by Algorithm 11 is a stationary point of the function $f$ defined by (4.48).

*Proof.* Suppose that $\overline{U}_l$ are the accumulation points of the sequences $\{U_l^{(j)}\}_{j\geq 1}$, $1 \leq l \leq d$, generated by Algorithm 11. Then there are subsequences $\{U_l^{(j)}\}_{j\in\mathscr{K}_l}$, $\mathscr{K}_l \subseteq \mathbb{N}$, such that

$$\{U_l^{(j)}\}_{j\in\mathscr{K}_l} \to \overline{U}_l, \quad 1 \leq l \leq d.$$

Further on, suppose that

$$\nabla f(\overline{U}_1, \overline{U}_2, \ldots, \overline{U}_d) \neq 0. \quad (4.78)$$

Then, for any $\varepsilon > 0$ there are $K_l \in \mathscr{K}_l$, $1 \leq l \leq d$ such that

$$\|U_l^{(k-1)} - \overline{U}_l\|_2 < \varepsilon, \quad \forall l = 1, \ldots, d,$$

for every $k > K$, $K = \max\{K_l : 1 \leq l \leq d\}$, and Lemma 4.4.2 implies that

$$f(U_1^{(k)}, U_2^{(k)}, \ldots, U_d^{(k)}) - f(U_1^{(k-1)}, U_2^{(k-1)}, \ldots, U_d^{(k-1)}) \geq \delta,$$

for some $\delta > 0$. Therefore, we have

$$f(U_1^{(k)}, U_2^{(k)}, \ldots, U_d^{(k)}) \to \infty,$$

when $k \to \infty$.

Since $f$ is a continuous function, convergence of $(U_1^{(j)}, U_2^{(j)}, \ldots, U_d^{(j)})$ implies the convergence of $f(U_1^{(j)}, U_2^{(j)}, \ldots, U_d^{(j)})$ and we got a contradiction. Hence,

$$\nabla f(\overline{U}_1, \overline{U}_2, \ldots, \overline{U}_d) = 0,$$

that is, $(\overline{U}_1, \overline{U}_2, \ldots, \overline{U}_d)$ is a stationary point of the function $f$. ■

### 4.4.4. Convergence of the structure-preserving tensor-trace maximization algorithm

To prove the convergence of Algorithm 12 we follow the same scheme as for Algorithm 11. We should keep two things in mind. First, the function that is being maximized by the Algorithm 12 is a function in only one variable. Second, unlike the Algorithm 11, this is not an ALS algorithm. These two facts will actually simplify the lemmas needed for the proof.

Instead of Lemma 4.4.1, we can now use Lemma 3.1 from [54].

**Lemma 4.4.4** (Li, Ushevich, Comon [54])**.** For every differentiable function $f_s \colon O_n \to \mathbb{R}$, $U \in O_n$, and $0 < \eta \leq \frac{2}{n}$ it is always possible to find index pair $(p,q)$ such that (4.61) holds.

Lemma 4.4.5 is similar to Lemma 4.4.2, but instead of $d$ microiterations we observe one iteration.

**Lemma 4.4.5.** Let $U^{(k)}$, $k \geq 0$, be the sequence generated by Algorithm 12. For $\overline{U} \in O_n$, let $\nabla f(\overline{U}) \neq 0$. Then there exist $\varepsilon > 0$ and $\delta > 0$ such that $\|U^{(k-1)} - \overline{U}\|_2 < \varepsilon$ implies

$$f_s(U^{(k)}) - f_s(U^{(k-1)}) \geq \delta.$$

*Proof.* The proof follows the same reasoning as the proof of Lemma 4.4.2.

For a fixed iteration $k$ we define the function $h_k\colon \mathbb{R} \to \mathbb{R}$,

$$h_k(\phi) = f_s(U^{(k-1)}R(p_k, q_k, \phi)).$$

The rotation angle in Algorithm 12 is chosen in such a way that

$$\max_{\phi} h_k(\phi) = h_k(\phi_{U,k}) = f_s(U^{(k-1)}R(p_k, q_k, \phi_{U,k}) = f_s(U^{(k)}).$$

Moreover, we have

$$h_k(0) = f_s(U^{(k-1)}R(p_k, q_k, 0) = f_s(U^{(k-1)}).$$

Thus,

$$f_s(U^{(k)}) - f_s(U^{(k-1)}) = h_k(\phi_{U,k}) - h_k(0) \geq h_k(\phi) - h_k(0), \tag{4.79}$$

for any angle $\phi$. We want to find some $\phi$ such that $h_k(\phi) - h_k(0) > 0$.

We use the Taylor expansion of the function $h_k$ around 0,

$$h_k(\phi) = h_k(0) + h_k'(0)\phi + \frac{1}{2}h_k''(\xi)\phi^2, \quad 0 < \xi < \phi.$$

For

$$M = \max_{0<\xi<\phi} |h_k''(\xi)| < \infty,$$

it follows from the relation (4.79) that

$$f_s(U^{(k)}) - f_s(U^{(k-1)}) \geq h_k'(0)\phi - \frac{1}{2}M\phi^2. \tag{4.80}$$

Using Lemma 4.4.4 we get

$$\begin{aligned} h_k'(0) &= \langle \nabla f_s(U^{(k-1)})R(p_k, q_k, 0), U^{(k-1)}\dot{R}(p_k, q_k, 0)\rangle \\ &= \langle \nabla f_s(U^{(k-1)}), U^{(k-1)}\dot{R}(p_k, q_k, 0)\rangle \\ &\geq \eta \|\nabla f_s(U^{(k-1)})\|_2. \end{aligned} \tag{4.81}$$

Since $\|U^{(k-1)} - \overline{U}\|_2 < \varepsilon$, there exists $\varepsilon > 0$ such that

$$\mu := \min\{\|\nabla f_s(U)\|_2 : \ \|U - \overline{U}\|_2 < \varepsilon\} > 0, \tag{4.82}$$

and it follows from (4.81) and (4.82) that

$$|h_k'(0)| \geq \eta\mu.$$

Then, from the inequality (4.80), for $\phi = \frac{1}{M}h_k'(0)$, we get

$$f_s(U^{(k)}) - f_s(U^{(k-1)}) \geq \frac{1}{2}\frac{h_k'(0)^2}{M} \geq \frac{1}{2M}\eta^2\mu^2 = \delta > 0.$$

■

Now we can prove the convergence of Algorithm 12.

**Theorem 4.4.6.** Every accumulation point $U$ obtained by Algorithm 12 is a stationary point of the function $f_s$ defined by (4.59).

*Proof.* The proof is analogous to the proof of Theorem 4.4.3. Instead of Lemma 4.4.2 it uses Lemma 4.4.5. ■

We end this section with the expression for $\nabla f_s$. We define the extension of the objective function $f_s$ as $\tilde{f}_s \colon \mathbb{R}^{n\times n} \to \mathbb{R}$,

$$\tilde{f}_s(U) = \operatorname{tr}(\mathscr{A} \times_1 U^T \times_2 U^T \cdots \times_d U^T) = \sum_{r=1}^{n}\left(\sum_{i_1,\ldots,i_d=1}^{n} a_{i_1 i_2 \ldots i_d} u_{i_1 r} u_{i_2 r} \cdots u_{i_d r}\right)$$

$$= \sum_{r=1}^{n}\sum_{m=1}^{n}\sum_{k=1}^{d}\binom{d}{k}\left(\sum_{\substack{i_{k+1},\ldots,i_d=1 \\ i_{k+1},\ldots,i_d\neq m}}^{n} a_{m\ldots m i_{k+1}\ldots i_d} u_{mr}^k u_{i_{k+1}r}\cdots u_{i_d r}\right).$$

Element-wise, the gradient of $\tilde{f}_s$ is given by

$$\frac{\partial \tilde{f}_s}{\partial u_{mr}} = \sum_{k=1}^{d}\binom{d}{k} k u_{mr}^{k-1}\left(\sum_{\substack{i_{k+1},\ldots,i_d=1 \\ i_{k+1},\ldots,i_d\neq m}}^{n} a_{m\ldots m i_{k+1}\ldots i_d} u_{i_{k+1}r}\cdots u_{i_d r}\right)$$

$$= \sum_{k=1}^{d}\binom{d}{k} k u_{mr}^{k-1}\,(\mathscr{A} \times_{k+1} U_0 \cdots \times_d U_0)_{\underbrace{m\cdots m}_{k}\underbrace{r\cdots r}_{d-k}},$$

where $U_0$ is a matrix equal to $U$ in all entries except for the $m$th row where the entries of $U_0$ are equal to zero. Then, $\nabla f_s$ is the projection of $\nabla \tilde{f}_s$ onto the tangent space at $U$ to the manifold $O_n$. That is,

$$\nabla f_s(U) = \operatorname{Proj} \nabla \tilde{f}_s(U) = U\Lambda(U),$$

where the operator $\Lambda$ is defined by

$$\Lambda(U) := \frac{U^T \nabla_U \tilde{f}_s - (\nabla_U \tilde{f}_s)^T U}{2}.$$

## 4.5. NUMERICAL EXPERIMENTS

In the final section of this chapter we present the results of our numerical experiments. All the tests are done in Matlab R2021a.

For both Algorithm 11 and Algorithm 12 we observe two values in each iteration — trace and relative off-norm of a current tensor. The trace is the objective function which is expected to increase in each microiteration and converge to some value. The algorithms stop when the change of the trace after one cycle is less than $10^{-4}$. The relative off-norm of a tensor $\mathscr{A}$ is given by

$$\frac{\text{off}(\mathscr{A})}{\|\mathscr{A}\|_F}.$$

Obviously, the relative off-norm of a diagonal tensor is equal to zero. On the contrary, the relative off-norm of a random tensor is close to one.

The algorithms are applied on general random tensors and random tensors that can be diagonalized using orthogonal transformations. Random tensor entries are drawn from the uniform distribution in the interval $[0,1]$. Orthogonally diagonalizable tensors are constructed such that we take a diagonal tensor with random uniformly distributed entries from $[0,1]$ on the diagonal and multiply it in each mode with random orthogonal matrices (obtained from QR decomposition of random matrices).

Figure 4.4 shows the convergence of the trace and the relative off-norm in the Algorithm 11 for diagonalizable $20\times 20\times 20$, $10\times 10\times 10\times 10$ and $5\times 5\times 5\times 5\times 5\times 5$ tensors, for different values of $\eta$ from (4.51). One can observe that for larger $\eta$, $\eta=\frac{1}{n}$, the trace converges to a lower value than for smaller $\eta$. Moreover, in these examples for $\eta=\frac{1}{n}$ the relative off-norm converges to a number greater than zero, while for the smaller $\eta$ it converges to zero. This means that for $\eta=\frac{1}{n}$ the algorithm converges to a different stationary point, the one that is not a diagonal tensor, than for smaller $\eta$. Therefore, from our observations, we recommend using a smaller $\eta$.

We repeat the same experiment as the one described above, but this time on non-diagonalizable $20\times 20\times 20$, $10\times 10\times 10\times 10$ and $5\times 5\times 5\times 5\times 5\times 5$ tensors. Here one cannot expect the relative off-norm to become equal to zero. The results are shown in Figure 4.5. Same as in Figure 4.4, for $\eta=\frac{1}{n}$ we get the convergence to a different, less desirable, stationary point of the function $f$.

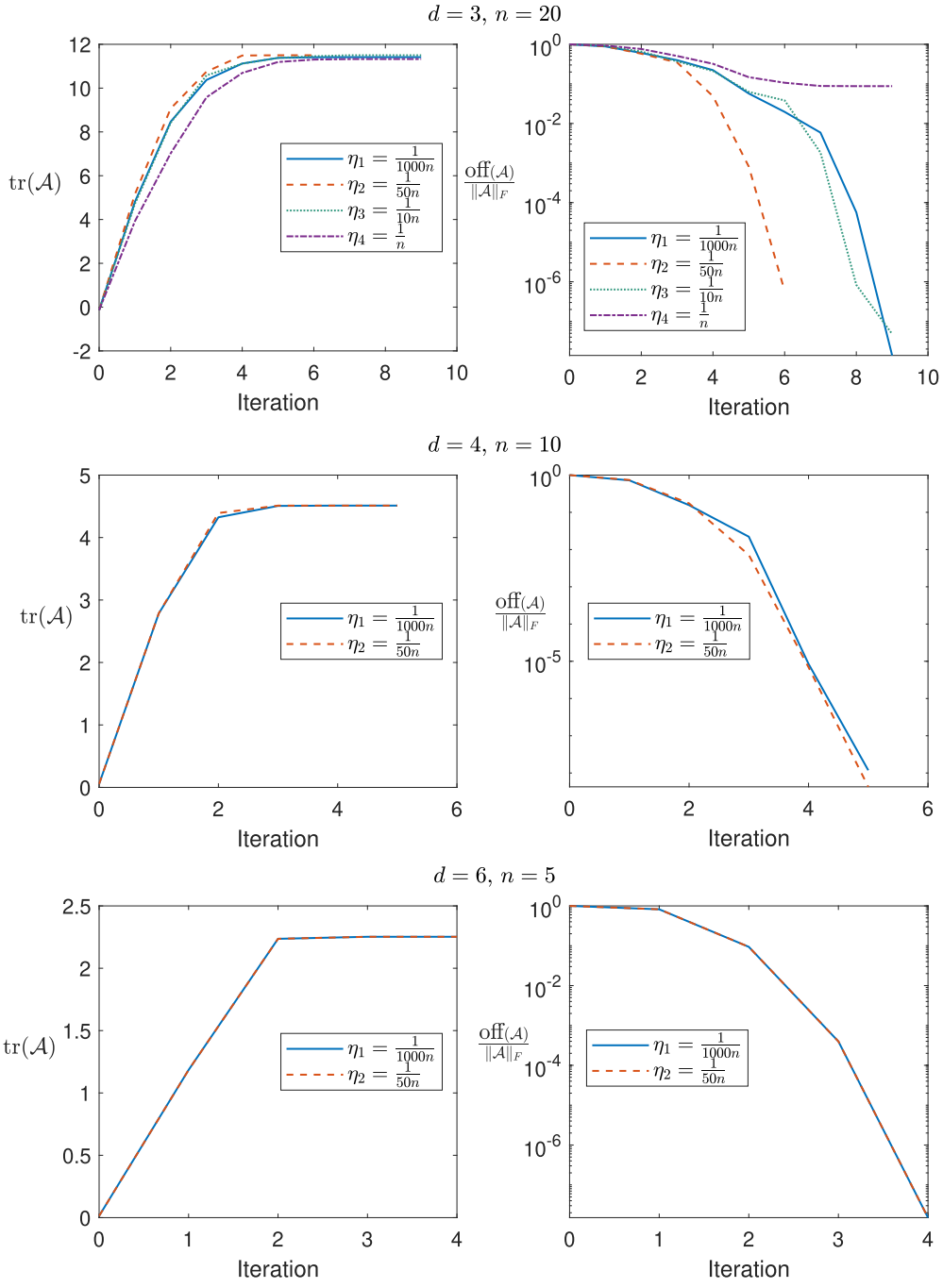



Figure 4.4: Convergence of Algorithm 11 for different values of $\eta$ on tensors of order 3, 4 and 6 that are diagonalizable using orthogonal transformations.

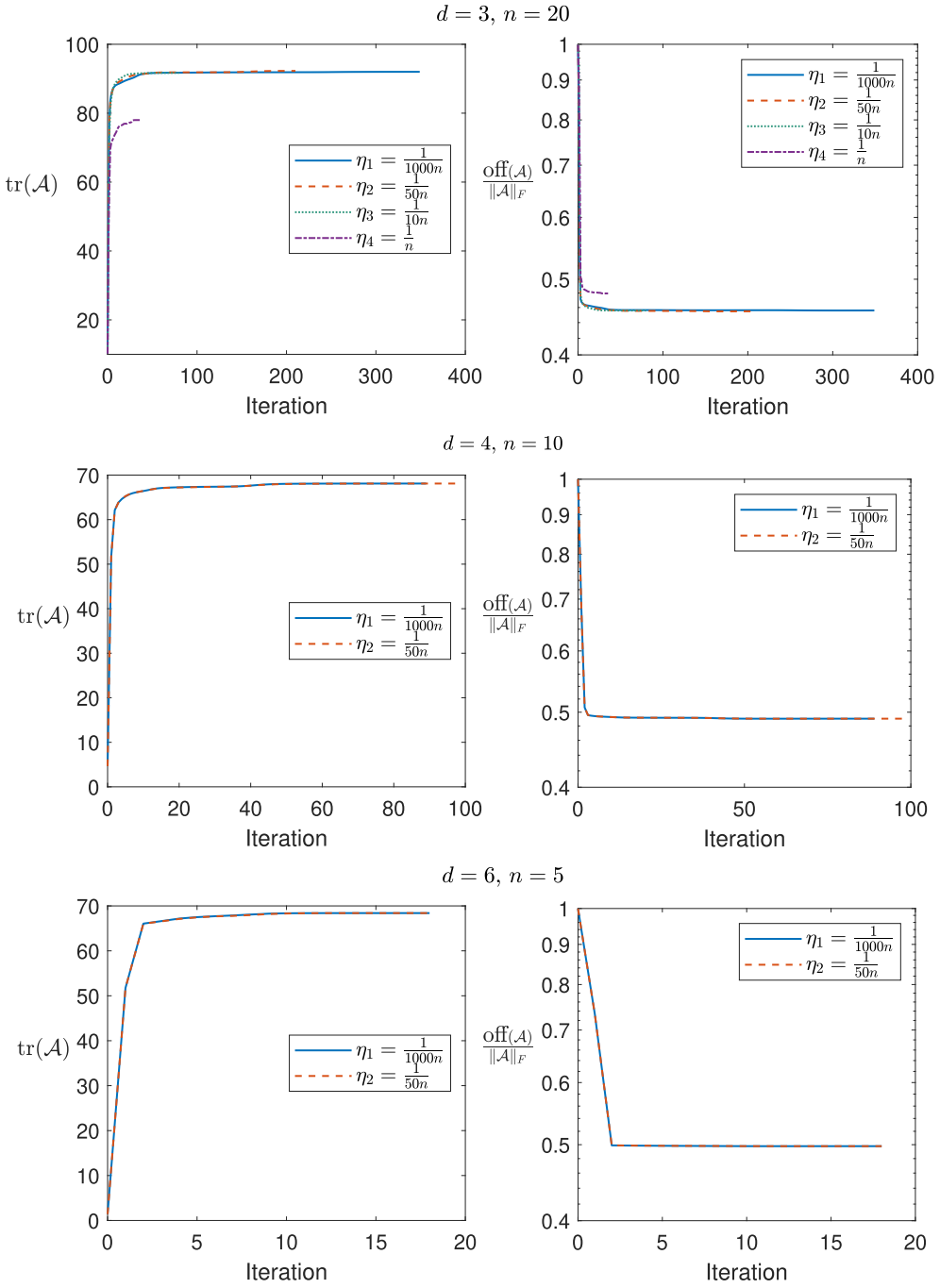



Figure 4.5: Convergence of Algorithm 11 for different values of $\eta$ on random tensors of order 3 and 6.

The bar graphs in Figure 4.6 show how $\eta$ affects the number of microiterations in each iteration of the Algorithm 11. The test is done on non-diagonalizable tensors of order $d = 3$, 4, 5 and 6. If $\eta$ is bigger, the condition (4.51) is more restrictive and more microiterations are skipped. For example, for $d = 3$ and $\eta = \frac{1}{n}$, 38.6% of iterations contain only one microiteration and only 12.7% contain maximum number of microiterations. On the other hand, for $\eta = \frac{1}{1000n}$, 99.8% of the iterations consist of all three microiterations.

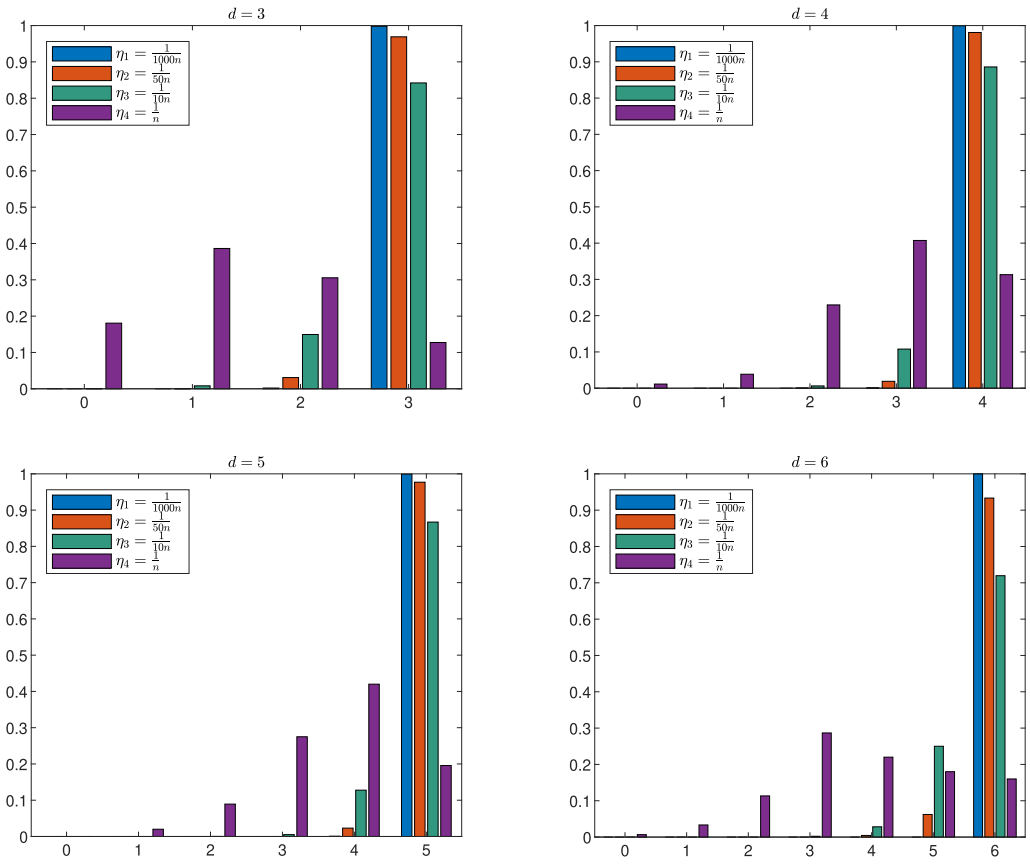



Figure 4.6: Portion of the number of microiterations within one iteration for different values of $\eta$ on random tensors of order $d = 3, 4, 5, 6$.

In Section 4.4 we discussed different initialization strategies for the Algorithm 11. In Figure 4.7 we compare the identity initialization given by (4.56) with the HOSVD initialization given by (4.57). The results are shown for a non-diagonalizable $10 \times 10 \times 10 \times 10$ tensor during the first 10 iterations. When a tensor is preconditioned using HOSVD it becomes closer to a diagonal one. Thus, the starting trace value is higher in the case of the HOSVD initialization than for the identity initialization. Also, the starting relative

off-norm is much closer to the limit value, significantly under the value of 1. Regardless, the algorithm that uses the identity initialization catches up after the first few iterations. Both initializations give equally good approximations and converge to the same value.

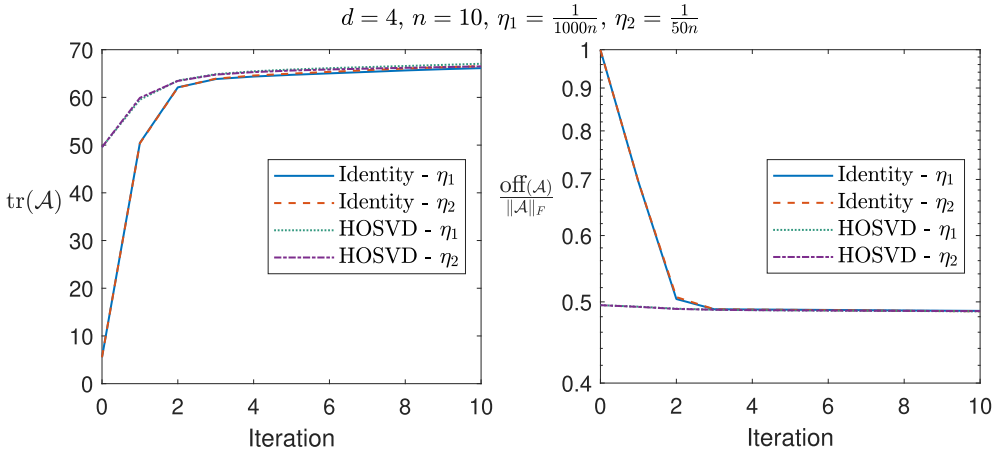



Figure 4.7: Comparison of different initialization strategies for Algorithm 11 for a random tensor of order 4 during the fist 10 iterations.

In Figure 4.8 we consider the trace maximization algorithm opposed to the Jacobi-type algorithm that maximizes the squares of the diagonal elements. Although the trace maximization is not equivalent to the maximization of the Frobenius norm of the diagonal, our numerical examples show that the off-norm of a tensor is decreasing, that is, the Frobenius norm of the diagonal is increasing, when the trace is increasing. We observe the performance of Algorithm 11 and Algorithm 8 [8] on two random $20 \times 20 \times 20$ tensors, one orthogonally diagonalizable, and one non-diagonalizable. We can see that the results of both algorithms are comparable.

Next, we compare Algorithm 11 with another trace maximizing method from [65]. The other algorithm is similar to Algorithm 7 as it constructs orthogonal matrices $\widehat{R}_U$, $\widehat{R}_V$ and $\widehat{R}_W$ from (4.31) using SVD of a certain matrix. To the best of our knowledge, there is no proof of convergence of this Jacobi compress trace algorithm, even though it numerically converges after just a few iterations. In Figure 4.9 we compare the two methods on the same $20 \times 20 \times 20$ tensors. We see that the trace in the Jacobi compress algorithm does not converge. Regardless, the relative off-norm for both non-diagonalizable and orthogonally diagonalizable tensor is comparable for the Jacobi compress trace algorithm and our Algorithm 11. Additionally, it numerically converges after just a few iterations.

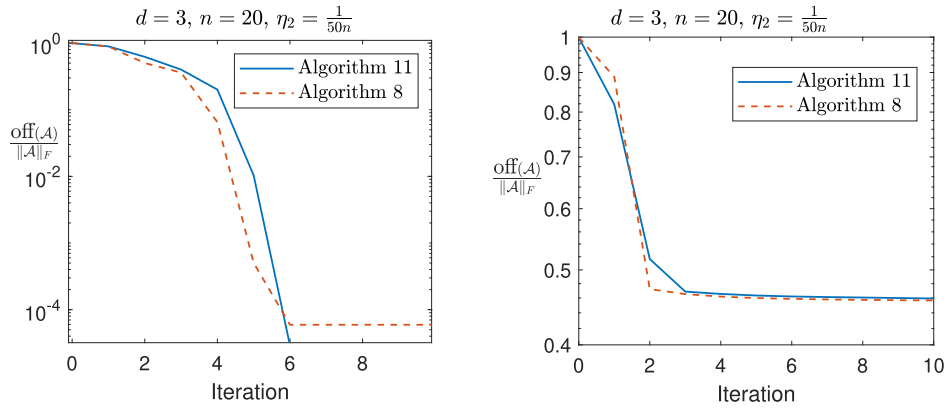



Figure 4.8: Trace maximization compared to the maximization of the squares of the diagonal elements for one diagonalizable and one random tensor during the first 10 iterations.

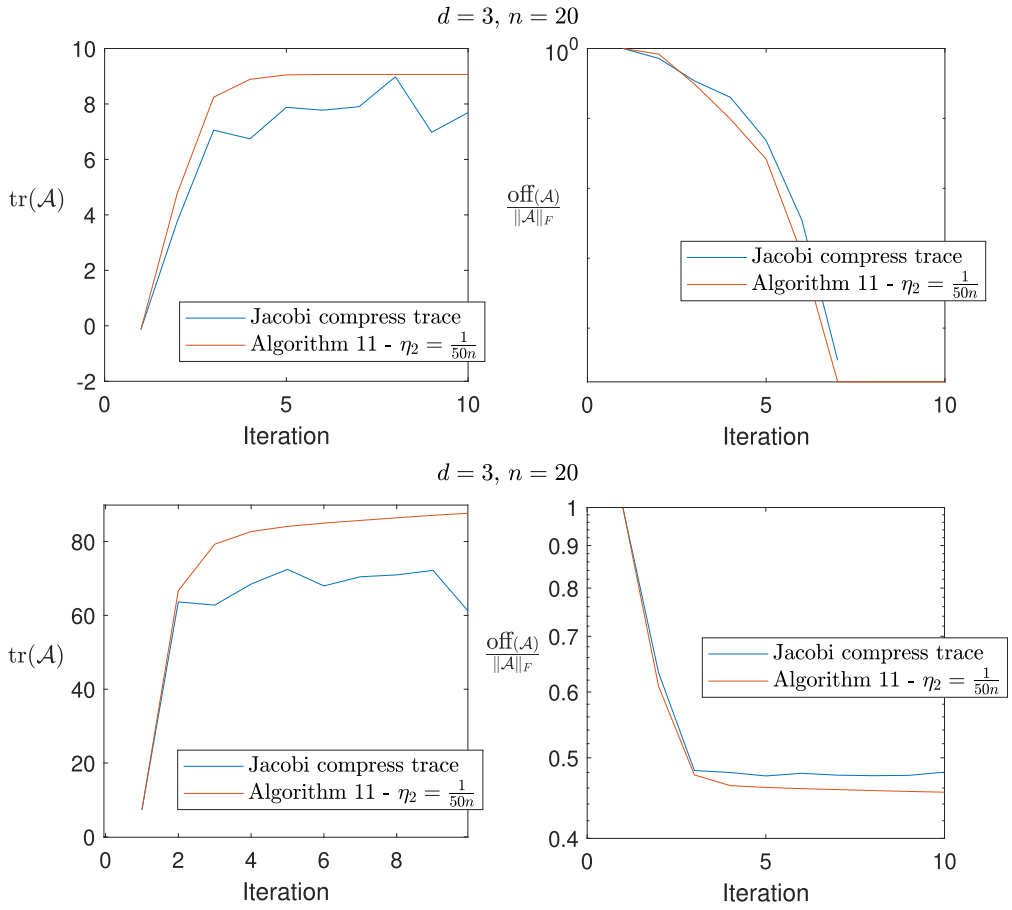



Figure 4.9: Trace maximization using Algorithm 11 compared to the Jacobi compress algorithm from [65] for one diagonalizable and one random tensor during the first 10 iterations.

Lastly, we observe the symmetric case. We present the convergence results of Algorithm 12 for a diagonalizable $20 \times 20 \times 20$ tensor in Figure 4.10.

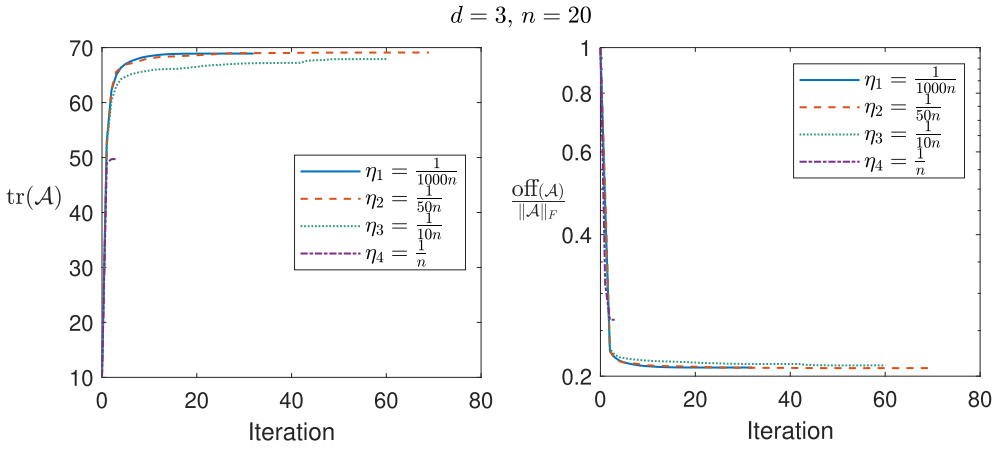



Figure 4.10: Convergence of Algorithm 12 on a random 3rd-order tensor.

Additionally to the Algorithm 12 we observe its modification, a hybrid approach between the Algorithms 11 and 12. In the Algorithm 12 the rotation matrix is chosen by optimizing the angle with respect to all modes, which leads to a polynomial equation of order $d$, (4.62) or (4.63). As this can be computationally challenging, we investigated another approach. From the relation (4.54) used in the Algorithm 11 we compute the rotation angle that is optimal in one mode $l$, for example $l = 1$. Because of the symmetry, it does not matter which mode we consider. Then we apply this same rotation in all modes, as it is done in the Algorithm 12, to preserve the symmetry. This mode-1 modification is given in Algorithm 13. We present its performance in Figure 4.11, where we compare its results with the Algorithm 12. The convergence results from Subsection 4.4.4 hold for Algorithm 12, but not for the modified Algorithm 13. In practice we observed that both Algorithm 13 and 12 converge to the same solution, although the modification converges slower. Analogous results were obtained for the non-diagonalizable tensors.

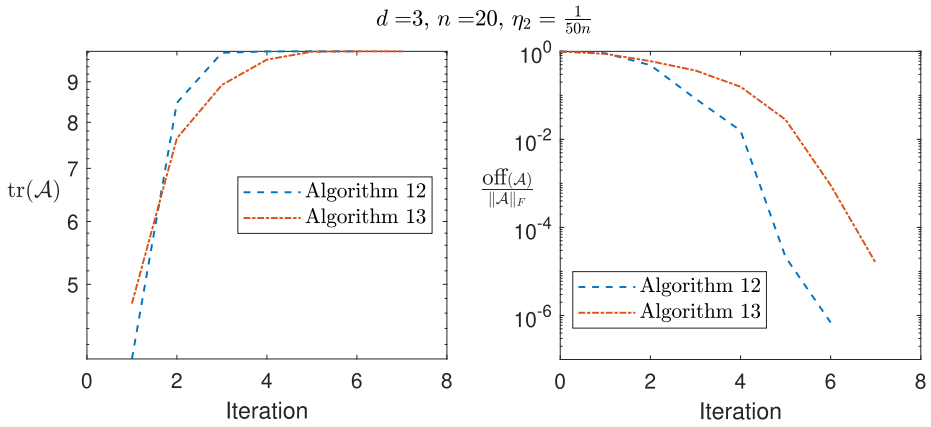



Figure 4.11: Convergence of Algorithms 12 and 13 on a tensor of order 3 that is diagonalizable using orthogonal transformations.

Still, the symmetry-preserving Algorithm 12 has some limitations when dealing with tensors of even order, specifically tensors with both positive and negative elements on the diagonal approximation [21]. Although the convergence theorem is still valid, the acquired stationary point of the objective function is not its global maximum. This behaviour can be seen in Figure 4.12 where we apply the Algorithm 11, the Algorithm 12, as well as its modification Algorithm 13 on a random orthogonally diagonalizable symmetric 4th-order tensor. A symmetric 4th-order diagonalizable tensor is constructed by taking a diagonal tensor with diagonal elements drawn uniformly from $[-1,1]$, that is multiplied in each mode by a same random orthogonal matrix. In numerical experiments we observed some interesting things. Algorithm 11 yielded matrices $U_l$, $1 \leq l \leq 4$, equal up to sign, specifically $-U_1 = U_2 = U_3 = U_4$. Moreover, obtained diagonal elements are good approximations of the absolute values of eigenvalues of the starting tensor, except for one value which is of negative sign. As seen in Figure 4.12, Algorithm 13 does not find the maximal trace. However, the approximation is as good as for the Algorithm 11 (in terms of the off-norm). Additionally, the obtained diagonal elements are good approximations of the eigenvalues of the starting tensor.

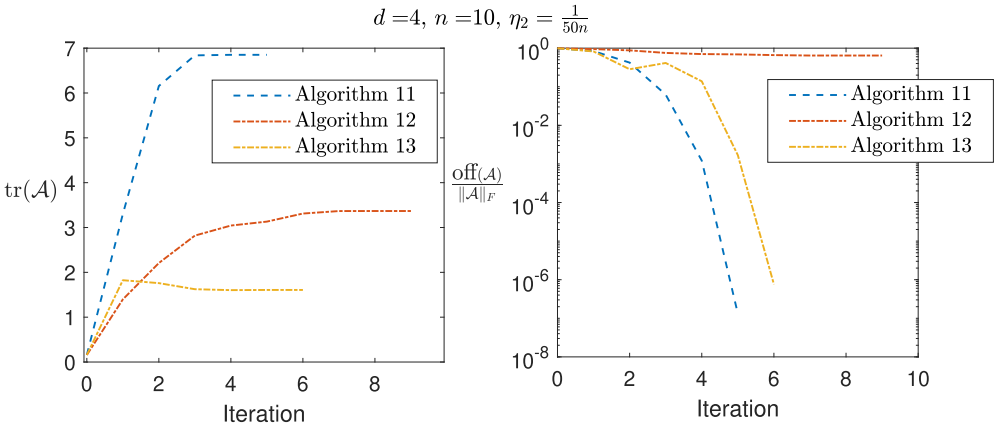



Figure 4.12: Convergence of Algorithm 11, Algorithm 12, and its modification Algorithm 13 on a symmetric 4th-order tensor that is diagonalizable using orthogonal transformations.

# CONCLUSION

The goal of this thesis was to improve some old algorithms and propose new Jacobi-type algorithms to solve the problem of matrix and tensor diagonalization. In the matrix case, we expanded the convergence results for the Eberlein method to generalized serial pivot strategies with permutations. Next, we proposed a block variant of the Eberlein method and proved its convergence under the same broad class of pivot strategies. In the tensor case, we designed two new algorithms for the approximate tensor diagonalization. One is an algorithm for general tensors, and the other one is a structure-preserving algorithm for symmetric tensors. We proved that both algorithms converge globally, that is, for every starting tensor, for every cyclic pivot strategy. All theoretical results were accompanied by various numerical examples.

This work opened up numerous questions and research directions. In order to improve the computation time of the Eberlein algorithm and other highly accurate Jacobi-type methods, one can try implementing mixed-precision arithmetic. This could especially be beneficial for the block Eberlein algorithm, particularly the part for finding the norm-reducing transformation $\mathbf{S}_k$, which is rather slow compared to the state-of-the-art algorithms. In the tensor case, the complexity of the computations grows exponentially with order $d$. This problem could be approached with new randomization techniques for tensor diagonalization, as well as for similar multilinear problems. Moreover, parallel pivot strategies for Jacobi-type algorithms on tensors are worth exploring.

# BIBLIOGRAPHY


[1] P.-A. Absil, R. Mahony, and B. Andrews. Convergence of the iterates of descent methods for analytic cost functions. *SIAM J. Optim.*, 16(2):531–547, 2005. ↑ 116.

[2] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*, volume 78. Mathematics of Computation - Math. Comput., 12 2008. ↑ 132.

[3] J. Alexander and A. Hirschowitz. Polynomial interpolation in several variables. *J. Algebraic Geom.*, 4(2):201–222, 1995. ↑ 107.

[4] A. Anandkumar, R. Ge, D. Hsu, S. M. Kakade, and M. Telgarsky. Tensor decompositions for learning latent variable models. *J. Mach. Learn. Res.*, 15:2773–2832, 2014. ↑ 99.

[5] H. Attouch, J. Bolte, and B. F. Svaiter. Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward-backward splitting, and regularized Gauss-Seidel methods. *Math. Program.*, 137(1-2):91–129, 2013. ↑ 116.

[6] J. L. Aurentz, T. Mach, L. Robol, R. Vandebril, and D. S. Watkins. *Core-chasing algorithms for the eigenvalue problem*, volume 13 of *Fundamentals of Algorithms*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2018. ↑ 8, 34.

[7] E. Begović. *Konvergencija blok Jacobijevih metoda*. PhD thesis, University of Zagreb, Faculty of Science, Zagreb, 2014. ↑ 16, 21, 22, 26, 27, 68, 79.

[8] E. Begović Kovač. Convergence of a Jacobi-type method for the approximate orthogonal tensor diagonalization. *Calcolo*, 60(1):Paper No. 3, 20, 2023. ↑ 3, 4, 108, 110, 114, 116, 117, 122, 131, 133, 134, 144.

[9] E. Begović Kovač and V. Hari. Convergence of the complex block Jacobi methods under the generalized serial pivot strategies. *Linear Algebra Appl.*, 699:421–458, 2024. ↑ 1, 3, 7.

[10] E. Begović Kovač and A. Perković. Convergence of the Eberlein diagonalization method under generalized serial pivot strategies. *Electron. Trans. Numer. Anal.*, 60:238–255, 2024. ↑ iii, 4.

[11] E. Begović Kovač and A. Perković. Trace maximization algorithm for the approximate tensor diagonalization. *Linear Multilinear Algebra*, 72(3):429–450, 2024. ↑ iv, 5.

[12] N. Bourbaki. *Elements of mathematics. Algebra, Part I: Chapters 1-3*. Hermann, Paris; Addison-Wesley Publishing Co., Reading, MA, 1974. ↑ 99.

[13] A. Brini, R. Q. Huang, and A. G. B. Teolis. The umbral symbolic method for supersymmetric tensors. *Adv. Math.*, 96(2):123–193, 1992. ↑ 102.

[14] A. Bunse-Gerstner, R. Byers, and V. Mehrmann. Numerical methods for simultaneous diagonalization. *SIAM J. Matrix Anal. Appl.*, 14(4):927–949, 1993. ↑ 99.

[15] A. Bunse-Gerstner and H. Faß bender. A Jacobi-like method for solving algebraic Riccati equations on parallel computers. *IEEE Trans. Automat. Control*, 42(8):1071–1084, 1997. ↑ 7.

[16] J. D. Carroll and J. Chang. Analysis of individual differences in multidimensional scaling via an n-way generalization of 'Eckart–Young' decomposition. *Psychometrika*, 35(3):283–319, 1970. ↑ 105.

[17] A. Cichocki, D. Mandic, C. Caiafa, A.-H. Phan, G. Zhou, Q. Zhao, and L. De Lathauwer. Multiway component analysis: tensor decompositions for signal processing applications. *IEEE Sig. Process. Mag.*, 32(2):145–163, 2015. ↑ 3, 99.

[18] P. Comon. Tensor diagonalization, a useful tool in signal processing. *IFAC Proceedings Volumes*, 27(8):77–82, 1994.

[19] P. Comon. Tensors: a brief introduction. *IEEE Signal Processing Magazine*, 31(3):44–53, 2014. ↑ 99.

[20] P. Comon, G. Golub, L.-H. Lim, and B. Mourrain. Symmetric tensors and symmetric tensor rank. *SIAM J. Matrix Anal. Appl.*, 30(3):1254–1279, 2008. ↑ 99, 107.

[21] P. Comon and M. Sorensen. Tensor diagonalization by orthogonal transforms. *Report ISRN I3S-RR-2007-06-FR*, 2007. ↑ 122, 147.

[22] L. De Lathauwer, B. De Moor, and J. Vandewalle. A multilinear singular value decomposition. *SIAM J. Matrix Anal. Appl.*, 21(4):1253–1278, 2000. ↑ 126.

[23] J. Demmel and K. Veselić. Jacobi's method is more accurate than QR. *SIAM J. Matrix Anal. Appl.*, 13(4):1204–1245, 1992. ↑ 1, 7.

[24] F. M. Dopico, P. Koev, and J. M. Molera. Implicit standard Jacobi gives high relative accuracy. *Numer. Math.*, 113(4):519–553, 2009. ↑ 1, 7.

[25] Z. Drmač. A global convergence proof for cyclic Jacobi methods with block rotations. *SIAM J. Matrix Anal. Appl.*, 31(3):1329–1350, 2009. ↑ 79.

[26] Z. Drmač and K. Veselić. New fast and accurate Jacobi SVD algorithm I. *SIAM J. Matrix Anal. Appl.*, 29(4):1322–1342, 2008. ↑ 1, 7.

[27] Z. Drmač and K. Veselić. New fast and accurate Jacobi SVD algorithm II. *SIAM J. Matrix Anal. Appl.*, 29(4):1343–1362, 2008. ↑ 1, 7.

[28] P. J. Eberlein. A Jacobi-like method for the automatic computation of eigenvalues and eigenvectors of an arbitrary matrix. *J. Soc. Indust. Appl. Math.*, 10(1):74–88, 1962. ↑ 2, 34, 39, 44, 47.

[29] L. Elsner and M. H. C. Paardekooper. On measures of nonnormality of matrices. *Linear Algebra Appl.*, 92:107–123, 1987. ↑ 38.

[30] H. Fassbender, D. S. Mackey, and N. Mackey. Hamiltonian and Jacobi come full circle: Jacobi algorithms for structured Hamiltonian eigenproblems. *Linear Algebra Appl.*, 332-334:37–80, 2001. ↑ 1, 7.

[31] G. E. Forsythe and P. Henrici. The cyclic Jacobi method for computing the principal values of a complex matrix. *Trans. Amer. Math. Soc.*, 94:1–23, 1960. ↑ 31, 78.

[32] H. H. Goldstine and L. P. Horwitz. A procedure for the diagonalization of normal matrices. *J. Assoc. Comput. Mach.*, 6:176–195, 1959. ↑ 58.

[33] W. Greub. *Multilinear algebra*. Universitext. Springer-Verlag, New York-Heidelberg, second edition, 1978. ↑ 99.

[34] E. R. Hansen. On cyclic Jacobi methods. *J. Soc. Indust. Appl. Math.*, 11:448–459, 1963. ↑ 19, 31, 32.

[35] V. Hari. On the global convergence of the Eberlein method for real matrices. *Numer. Math.*, 39:361–369, 1982. ↑ 3, 35, 45, 46, 47.

[36] V. Hari. On the convergence of cyclic Jacobi-like processes. *Linear Algebra Appl.*, 81:105–127, 1986. ↑ 24.

[37] V. Hari. On block Jacobi annihilators. *Proceedings of ALGORITMY*, pages 429–439, 2009. ↑ 24.

[38] V. Hari. Convergence to diagonal form of block Jacobi-type methods. *Numer. Math.*, 129(3):449–481, 2015. ↑ 1, 7, 24, 80.

[39] V. Hari and E. B. Kovač. Convergence of the cyclic and quasi-cyclic block Jacobi methods. *Electron. Trans. Numer. Anal.*, 46:107–147, 2017. ↑ 2, 16, 22, 24, 67.

[40] V. Hari and E. B. Kovač. On the convergence of complex Jacobi methods. *Linear multilinear algebra*, 69(3):489–514, 2021. ↑ 16, 17, 24, 32, 49.

[41] V. Hari, S. Singer, and S. Singer. Full block $J$–Jacobi method for Hermitian matrices. *Linear Algebra Appl.*, 444:1–27, 2014. ↑ 1, 7.

[42] R. A. Harshman. Foundations of the parafac procedure: Models and conditions for an ”explanatory” multi-modal factor analysis. *UCLA Working Papers in Phonetics*, 16(84), 1970. ↑ 105.

[43] P. Henrici and K. Zimmermann. An estimate for the norms of certain cyclic Jacobi operators. *Linear Algebra Appl.*, 1(4):489–501, 1968. ↑ 24, 29.

[44] F. L. Hitchcock. The expression of a tensor or a polyadic as a sum of products. *Journal of Mathematics and Physics*, 6(1-4):164–189, 1927. ↑ 4, 104, 105.

[45] M. Ishteva, P.-A. Absil, and P. Van Dooren. Jacobi algorithm for the best low multilinear rank approximation of symmetric tensors. *SIAM J. Matrix Anal. Appl.*, 34(2):651–672, 2013. ↑ 108, 131.

[46] C. Jacobi. Über ein leichtes Verfahren die in der Theorie der Säcularstörungen vorkommenden Gleichungen numerisch aufzulösen*). *Journal für die reine und angewandte Mathematik*, 1846(30):51–94, 1846. ↑ 7.

[47] H. A. L. Kiers. Towards a standardized notation and terminology in multiway analysis. *J. Chemometrics*, 14:105–122, 2000. ↑ 101.

[48] E. Kofidis and P. A. Regalia. On the best rank-1 approximation of higher-order supersymmetric tensors. *SIAM J. Matrix Anal. Appl.*, 23(3):863–884, 2001/02. ↑ 102.

[49] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM Rev.*, 51(3):455–500, 2009. ↑ 99.

[50] P. M. Kroonenberg. *Applied multiway data analysis*. Wiley Series in Probability and Statistics. Wiley-Interscience [John Wiley & Sons], Hoboken, NJ, 2008. ↑ 99.

[51] C. Lageman. *Convergence of gradient-like dynamical systems and optimization algorithms*. PhD thesis, Universität Würzburg, 2007. ↑ 116.

[52] S. Lang. *Algebra*, volume 211 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, third edition, 2002. ↑ 99.

[53] L. D. Lathauwer, B. D. Moor, and J. Vandewalle. Independent component analysis and (simultaneous) third-order tensor diagonalization. *IEEE Transactions on Signal Processing*, 41:2262–2271, 2001. ↑ 3.

[54] J. Li, K. Usevich, and P. Comon. Globally convergent Jacobi-type algorithms for simultaneous orthogonal symmetric tensor diagonalization. *SIAM J. Matrix Anal. Appl.*, 39(1):1–22, 2018. ↑ 3, 4, 108, 110, 119, 120, 122, 131, 137.

[55] J. Li, K. Usevich, and P. Comon. On approximate diagonalization of third order symmetric tensors by orthogonal transformations. *Linear Algebra Appl.*, 576:324–351, 2019. ↑ 3, 4, 108, 110, 118.

[56] G. Loizou. On the quadratic convergence of the Jacobi method for normal matrices. *Comput. J.*, 15:274–276, 1972. ↑ 1, 7, 58.

[57] F. T. Luk and H. Park. On parallel Jacobi orderings. *SIAM J. Sci. Statist. Comput.*, 10(1):18–26, 1989. ↑ 1, 20.

[58] F. T. Luk and H. Park. A proof of convergence for two parallel Jacobi SVD algorithms. *IEEE Trans. Comput.*, 38(6):806–811, 1989. ↑ 1, 3, 7, 20, 21.

[59] D. S. Mackey, N. Mackey, C. Mehl, and V. Mehrmann. Numerical methods for palindromic eigenvalue problems: Computing the anti-triangular Schur form. *Numer. Linear Algebra Appl.*, 16(1):63–86, 2009. ↑ 1, 7.

[60] D. S. Mackey, N. Mackey, and F. Tisseur. Structured tools for structured matrices. *Electron. J. Linear Algebra*, 10:106–145, 2003. ↑ 1, 7.

[61] W. F. Mascarenhas. On the convergence of the Jacobi method for arbitrary orderings. *SIAM J. Matrix Anal. Appl.*, 16(4):1197–1209, 1995. ↑ 1, 7, 32.

[62] J. Matejaš. Accuracy of the Jacobi method on scaled diagonally dominant symmetric matrices. *SIAM J. Matrix Anal. Appl.*, 31(1):133–153, 2009. ↑ 1, 7.

[63] C. Mehl. Jacobi-like algorithms for the indefinite generalized Hermitian eigenvalue problem. *SIAM J. Matrix Anal. Appl.*, 25:964–985, 2004. ↑ 1, 7.

[64] C. Mehl. On asymptotic convergence of nonsymmetric Jacobi algorithms. *SIAM J. Matrix Anal. Appl.*, 30(1):291–311, 2008. ↑ 7, 55.

[65] C. D. Moravitz Martin and C. F. Van Loan. A Jacobi-type method for computing orthogonal tensor decompositions. *SIAM J. Matrix Anal. Appl.*, 30(3):1219–1232, 2008. ↑ iv, 4, 5, 108, 110, 111, 112, 113, 114, 122, 144, 145.

[66] A. H. Phan, P. Tichavský, and A. Cichocki. Blind source separation of single channel mixture using tensorization and tensor diagonalization. *Lecture Notes in Computer Science, Springer*, 10169, 2017. ↑ 99.

[67] D. Pupovci and V. Hari. On the convergence of parallelized Eberlein methods. *Rad. Mat.*, 8:249–267, 1992. ↑ 3, 35, 46, 54, 83.

[68] E. Robeva. Orthogonal decomposition of symmetric tensors. *SIAM J. Matrix Anal. Appl.*, 37(1):86–102, 2016. ↑ 106.

[69] G. Shroff and R. Schreiber. On the convergence of the cyclic Jacobi method for parallel block orderings. *SIAM J. Matrix Anal. Appl.*, 10(3):326–346, 1989. ↑ 1, 19, 20, 32.

[70] I. Slapničar. Highly accurate symmetric eigenvalue decomposition and hyperbolic SVD. *Linear Algebra Appl.*, 358:387–424, 2003. ↑ 1, 7.

[71] P. Tichavský, A. H. Phan, and A. Cichocki. Non-orthogonal tensor diagonalization. *Signal Process*, 138:313–320, 2017. ↑ 3, 99.

[72] L. R. Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31:279–311, 1966. ↑ 104.

[73] A. Uschmajew. A new convergence proof for the higher-order power method and generalizations. *Pac. J. Optim.*, 11(2):309–321, 2015. ↑ 116.

[74] K. Usevich, J. Li, and P. Comon. Approximate matrix and tensor diagonalization by unitary transformations: convergence of Jacobi-type algorithms. *SIAM J. Optim.*, 30(4):2998–3028, 2020. ↑ 3, 4, 108, 110, 120, 122.

[75] K. Veselić. A convergent Jacobi method for solving the eigenproblem of arbitrary real matrices. *Numer. Math.*, 25:179–184, 1976. ↑ 2, 34, 45.

# CURRICULUM VITAE

Ana (Bokšić) Perković, was born on November 21, 1995, in Makarska, Croatia, where she finished primary school. She graduated from high school at III. gymnasium Split. In 2014, she enrolled in University of Zagreb to study mathematics at the Faculty of Science. She got her master's degree in mathematics in 2019, when she entered the doctoral programme at the Department of Mathematics, Faculty of Science, University of Zagreb.

From 2020, she works as a teaching assistant at the Faculty of Chemical Engineering and Technology, University of Zagreb.

She participated in three conferences, the 11th Conference on Applied Mathematics and Scientific Computing (ApplMath22), September 5–9, 2022, Brijuni, Croatia, with a poster, the 7th Faculty of Science PhD Student Symposium, April 21-22, 2023, in Zagreb, Croatia, with a talk, both with the title *Trace maximization algorithm for the approximate tensor diagonalization*, and the 12th Conference on Applied Mathematics and Scientific Computing (ApplMath24), September 23–27, 2024, in Dubrovnik, Croatia, with a talk *Convergence of the Eberlein diagonalization method under generalized serial pivot strategies*. She has two papers published, both are co-written with her supervisor Erna Begović Kovač.

She was a participant of summer school organized by CIME (International Mathematical Summer Center), September 6–10, 2021, Cetraro, Italy, where the theme was Matrix nearness problems and eigenvalue optimization.