University of Zagreb
Department of Mathematics

Nela Bosner

Fast Methods for Large Scale Singular Value Decomposition

Doctoral Thesis

Ph.D. Supervisor: dr. sc. Zlatko Drmač

Zagreb, 2006

I thank my supervisor professor Zlatko Drmač for giving me the opportunity to work on such interesting problems and for reading all my papers, and I thank professor Jesse Barlow for very instructive collaboration. I also wish to thank several people who helped me in the realization of this work: Ratimir Komarica for carefully reading the text and correcting spelling and grammatical mistakes, despite the fact that he is not a mathematician; Daniel Kressner and Michiel Hochstenbach for reading some parts of the work with great enthusiasm; Zvonimir Bujanović for performing large number of numerical tests; Nenad Antonić for fixing all the problems with computers which occurred during the extensive testing process; and finally Borjan Komarica who helped me to overcome all the difficulties on the way to PhD degree.

Contents

Chapter 1

Introduction

This thesis deals with two major topics in numerical linear algebra: the singular value decomposition (SVD) and the eigenvalue problem. Both represent a standard tool for numerical solutions of problems which appear in many applications in various fields of natural sciences. The singular values and eigenvalues are connected, and they are usually considered to be two aspects of the same problem. The singular value decomposition of a rectangular matrix $A \in \mathbb{R}^{m\times n}$, where without loss of generality $m \geq n$, is a decomposition obtained by orthogonal transformation which produces a diagonal matrix:

$$A = U\Sigma V^T,$$

where

$$U \in \mathbb{R}^{m\times m},\ U^TU = I, \quad V \in \mathbb{R}^{n\times n},\ V^TV = I, \quad \Sigma \in \mathbb{R}^{m\times n},\ \Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_n).$$

The diagonal elements of Σ, $\sigma_i \geq 0$ are called singular values.

There are several natural parallels between singular values and eigenvalues:

- for the singular value σ_i of the matrix A, σ_i^2 is an eigenvalue of the matrices A^TA and AA^T,
- for the singular value σ_i of the matrix A, $\pm\sigma_i$ are eigenvalues of the matrix $\begin{bmatrix} 0 & A^T \\ A & 0 \end{bmatrix}$,
- for the eigenvalue λ_i of the symmetric matrix A, $|\lambda_i|$ is a singular value of A.

With the accelerating speed of computers, we can now solve eigenvalue and singular value problems of very large dimension. Thus our algorithms need to be efficient with large matrices. On the other hand, computers do not use real arithmetic, they use floating point arithmetic, and they do not produce exact solutions. Because of that our algorithms need to be efficient in modern architectures and accurate in floating point arithmetic.

In this work two new algorithms are proposed: one for finding the singular value decomposition, and one for solving the partial eigenvalue problem of a symmetric positive definite matrix. They are both designed to be efficient, and numerical analysis for

both algorithms confirms that they are also accurate. In addition to the algorithms, a new perturbation result for singular value approximations obtained from subspaces is presented, which measures relative error in singular values by means of angles of appropriate subspaces. This result gives us new insight into approximate solutions of the singular value decomposition.

The work is organized as follows. In §2 we analyze the singular value decomposition, which includes basic definitions and properties, applications, perturbation theory and most important existing methods for computing the SVD. This chapter serves as an introduction for the new bidiagonalization algorithm presented in §3. The new bidiagonalization algorithm was proposed by Barlow in [2], and it constitutes the first step in computing the singular value decomposition. In §3 an elegant proof of numerical stability of the bidiagonalization algorithm is presented, obtained independently from the results in [2]. Further, it is shown that despite of possible loss of orthogonality of columns in the computed matrix U, the bidiagonalization algorithm can be used as a tool for solving several problems in numerical linear algebra with high accuracy. The efficiency of the algorithm is also considered. The new bidiagonalization algorithm has fewer floating point operations than other standard algorithms, but in the original version the usage of the hierarchical structure of memory was not optimized. Hence, the new algorithm, executed on a computer, was slower than the algorithm implemented in LAPACK [1], and it required modification to optimize the time spent on transfer between different types of computer memory. This was the reason for developing the block version of the bidiagonalization algorithm, which is proven to be numerically stable. The parallel version of the algorithm was also considered, since the new bidiagonalization is more suitable for parallel computing than the standard algorithms. Extensive testing was performed for all versions of the bidiagonalization algorithm, the tests showed that on our computers the block and parallel versions of the new algorithm were faster than the algorithms in LAPACK [1] and ScaLAPACK [7].

In §4, we deal with the symmetric eigenvalue problem, and analyzes some aspects of the problem, such as: basic definitions and properties, applications, perturbation theory and most important existing subspace methods for solving the partial eigenvalue problem. Again, this chapter serves as an introduction for the new subspace method described in §5, which is called *multispace*. Multispace is a combination of multigrid approach and of two very well known subspace methods: inverse iteration and the block Lanczos method. Inverse iteration is known to stagnate when eigenvalues are not conveniently distributed, and the new multigrid approach is designed to speed up the convergence. A convergence rate for multispace is also presented, proving that the whole process converges to an invariant subspace. The numerical examples are presented at the end of this chapter.

§6 presents the new perturbation results for singular value approximations.

At last, we have introduce some notation. First of all, we will consider real matrices $A \in \mathbb{R}^{m\times n}$ or $A \in \mathbb{R}^{n\times n}$, and without loss of generality we will assume that $m \geq n$. If $m < n$ then we can take A^T, where ^{T} denotes the transposed matrix. The matrices will be denoted by capital Latin letters, the vectors by small letters, the subspaces by calligraphic letters, and in most cases Greek letters will be used to denote scalar values

such as parameters and angles.

Let $x \in \mathbb{R}^n$ be an n dimensional vector, then we denote $x = [x_i]$, or

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix},$$

which means that the i-th component of x is equal to x_i. Next, let $A \in \mathbb{R}^{m \times n}$ be an $m \times n$ matrix, then we denote $A = [a_{ij}]$, or

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix},$$

where a_{ij} is the element in the position (i, j). Sometimes, we will also use MATLAB notation:

$$\begin{aligned} x(i) &= x_i, \\ x(i:j) &= \begin{bmatrix} x_i \\ \vdots \\ x_j \end{bmatrix}, \\ A(i,j) &= a_{ij}, \\ A(i:j, k:\ell) &= \begin{bmatrix} a_{ik} & \cdots & a_{i\ell}, \\ \vdots & & \vdots \\ a_{jk} & \cdots & a_{j\ell} \end{bmatrix}, \\ A(:, k:\ell) &= A(1:m, k:\ell), \\ A(i:j, :) &= A(i:j, 1:n). \end{aligned}$$

The matrices I_k and 0_k denote $k \times k$ identity and zero matrix, respectively.

The scalar product used in the work is the standard scalar product in $\mathbb{R}^n$

$$\begin{aligned} \langle x, y \rangle &= x^T y = \sum_{i=1}^{n} x_i y_i, \qquad \text{or} \\ \langle x, y \rangle_A &= x^T A y, \qquad \text{for a symmetric positive definite matrix } A. \end{aligned}$$

We also use the Euclidean vector norm

$$\|x\|_2 = \sqrt{x^T x},$$

and two matrix norms

$$\begin{aligned} \|A\|_2 &= \max_{\|x\|_2 = 1} \|Ax\|_2 = \sqrt{\mathrm{spr}(A^T A)}, \\ \|A\|_F &= \sqrt{\mathrm{trace}(A^T A)} = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij}^2}, \end{aligned}$$

where for $B \in \mathbb{R}^{n\times n}$

$$\mathrm{spr}(B) = \max\{|\lambda| : \lambda \text{ is an eigenvalue of } B\},$$

denotes spectral radius, and

$$\mathrm{trace}(B) = \sum_{i=1}^{n} b_{ii}.$$

Since we are dealing with finite precision arithmetic, the computed quantities in numerical analysis are denoted by $\tilde{}$, and the exact quantities by $\hat{}$. Hence, we denote quantity x computed in finite precision arithmetic by $\tilde{x}$, and $\hat{x}$ presents an exact, sometimes only theoretical entity, which is obtained in exact arithmetic in some phase of the computation. The unit roundoff error of a computer is denoted by ε.

Chapter 2

The Singular Value Decomposition

2.1 Definitions and Properties

The singular value decomposition (SVD) of a real matrix $A \in \mathbb{R}^{m\times n}$ is a very powerful computation tool for solving many problems in numerical linear algebra. From the theoretical point of view, it is also used in numerical analysis as a decomposition that reveals important information about the matrix and the problem it is involved with. The singular value decomposition was discovered independently by Beltrami in 1873 [4] and Jordan in 1874 [54] during their research on bilinear forms. Since then, many mathematicians have been working on discovering its properties, both in exact and finite precision arithmetic, and developing algorithms for its computation. The SVD decomposition is described in the following theorem.

Theorem 2.1.1 (Singular Value Decomposition (SVD), [35, p. 71]). *If $A \in \mathbb{R}^{m\times n}$ is a real $m \times n$ matrix, then there exist orthogonal matrices*

$$U = [u_1, \ldots, u_m] \in \mathbb{R}^{m\times m} \quad \textit{and} \quad V = [v_1, \ldots, v_n] \in \mathbb{R}^{n\times n}$$

such that

$$U^T A V = \Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_p) \in \mathbb{R}^{m\times n}, \qquad p = \min\{m, n\}, \tag{2.1}$$

where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p \geq 0$.

Proof. From the definition of matrix 2 norm, there exists $v_1 \in \mathbb{R}^n$ with $\|v_1\|_2 = 1$, such that $\|A\|_2 = \|Av_1\|_2$. Let $u_1 \in \mathbb{R}^n$ be a vector with $\|u_1\|_2 = 1$, which satisfies

$$Av_1 = \sigma_1 u_1, \qquad \text{where} \quad \sigma_1 = \|A\|_2. \tag{2.2}$$

Since any orthonormal set of vectors can be extended to an orthonormal basis, it is possible to find $V_{1,2} \in \mathbb{R}^{n\times(n-1)}$ and $U_{1,2} \in \mathbb{R}^{m\times(m-1)}$ so that

$$\mathbf{V}_1 = \begin{bmatrix} v_1 & V_{1,2} \end{bmatrix} \in \mathbb{R}^{n\times n} \quad \text{and} \quad \mathbf{U}_1 = \begin{bmatrix} u_1 & U_{1,2} \end{bmatrix} \in \mathbb{R}^{m\times m}$$

are orthogonal. $\mathbf{U}_1^T A \mathbf{V}_1$ has the following structure:

$$\mathbf{U}_1^T A \mathbf{V}_1 = \begin{bmatrix} \sigma_1 u_1^T u_1 & u_1^T A V_{1,2} \\ \sigma_1 U_{1,2}^T u_1 & U_{1,2}^T A V_{1,2} \end{bmatrix} = \begin{bmatrix} \sigma_1 & w_1^T \\ 0 & A_2 \end{bmatrix} = \bar{A}_1,$$

where $w_1 = V_{1,2}^T A^T u_1$ and $A_2 = U_{1,2}^T A V_{1,2}$. Since

$$\begin{aligned} \|\bar{A}_1\|_2^2(\sigma_1^2 + w_1^T w_1) &= \|\bar{A}_1\|_2^2 \left\| \begin{bmatrix} \sigma_1 \\ w_1 \end{bmatrix} \right\|_2^2 \geq \left\| \bar{A}_1 \begin{bmatrix} \sigma_1 \\ w_1 \end{bmatrix} \right\|_2^2 = \\ &= (\sigma_1^2 + w_1^T w_1)^2 + w_1^T A_2^T A_2 w_1 \geq (\sigma_1^2 + w_1^T w_1)^2 \end{aligned}$$

we have $\|\bar{A}_1\|_2^2 \geq \sigma_1^2 + w_1^T w_1$. From the definition of σ_1 in (2.2) it follows that

$$\sigma_1^2 = \|A\|_2^2 = \|\bar{A}_1\|_2^2 \geq \sigma_1^2 + w_1^T w_1,$$

and this implies that $w_1 = 0$, and

$$\mathbf{U}_1^T A \mathbf{V}_1 = \begin{bmatrix} \sigma_1 & 0 \\ 0 & A_2 \end{bmatrix}.$$

The rest of the proof is done by applying the same technique to A_2, and (2.1) follows from the induction argument. □

Remark 2.1.2. *For every $A \in \mathbb{R}^{m \times n}$, Theorem 2.1.1 implies that (see [89, pp. 30–31])*

$$U^T A V = \begin{bmatrix} \Sigma_+ & 0 \\ 0 & 0 \end{bmatrix} \begin{matrix} r \\ m-r \end{matrix}$$
$$\qquad\quad r \quad n-r$$

where $r = \operatorname{rank}(A)$, and $\Sigma_+ = \operatorname{diag}(\sigma_1, \ldots, \sigma_r)$, with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$.

Remark 2.1.3. *For $m \geq n$ and every matrix $A \in \mathbb{R}^{m \times n}$ with $\operatorname{rank}(A) = r \leq n$, Theorem 2.1.1 implies that the matrix A can be factorized as*

$$A = U \Sigma V^T, \tag{2.3}$$

or for the partition

$$U = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \qquad V = \begin{bmatrix} V_1 & V_2 \end{bmatrix}$$
$$\quad r \quad m-r \qquad\qquad r \quad n-r$$

where $U_1 \in \mathbb{R}^{m \times r}$ and $V_1 \in \mathbb{R}^{n \times r}$, as

$$A = U_1 \Sigma_+ V_1^T = \sum_{i=1}^{r} \sigma_i u_i v_i^T. \tag{2.4}$$

There are many notions connected with SVD, and the most important of them are as follows:

- The factorization described in equation (2.4) is the **abbreviated SVD** and the right equation represents the **SVD expansion**.
- The nonnegative values σ_i are the **singular values** of A.

- The set[1] of all singular values of A is denoted by $\sigma(A) = \{\sigma_1, \ldots, \sigma_p\}$, where $p = \min\{m, n\}$.
- The vectors u_i are the **left singular vectors**.
- The vectors v_i are the **right singular vectors**.

Besides these basic notions, there are many matrix properties which are based on the SVD. Before expressing their definitions we should note that by comparing columns in equations $AV = \Sigma V$ and $A^T U = \Sigma V$ it follows that

$$\left.\begin{array}{r} Av_i = \sigma_i u_i \\ A^T u_i = \sigma_i v_i \end{array}\right\} \quad i = 1, \ldots, p = \min\{m, n\}.$$

Three important characteristics of a matrix A are immediately available from the singular value decomposition of A.

Corollary 2.1.4 ([35, p. 72]). *If the SVD of $A \in \mathbb{R}^{m\times n}$ is given by Theorem 2.1.1, and if we define r by*

$$\sigma_1 \geq \cdots \geq \sigma_r > \sigma_{r+1} = \cdots = \sigma_p = 0, \qquad p = \min\{m, n\},$$

then

- $\operatorname{rank}(A) = r$
- $\operatorname{null}(A) = \operatorname{span}\{v_{r+1}, \ldots, v_n\}$
- $\operatorname{range}(A) = \operatorname{span}\{u_1, \ldots, u_r\}$

A further important property of the singular value decomposition is that there exists a connection between the singular values and the eigenvalues.

Definition 2.1.5 ([35, pp. 332–333]). *The **eigenvalues** of a square matrix $A \in \mathbb{R}^{n\times n}$ are n roots of its **characteristic polynomial** $p(\lambda) = \det(\lambda I - A)$. The set[2] of these roots is called the **spectrum** and is denoted by*

$$\lambda(A) = \{\lambda_1, \ldots, \lambda_n\}.$$

If $\lambda \in \lambda(A)$ then a nonzero vector $x \in \mathbb{R}^n$, which satisfies

$$Ax = \lambda x$$

*is called an **eigenvector**.*

[1] Since the singular values can be multiple, $\sigma(A)$ is in fact a multiset. It can also be regarded as an element of the quotient space $\mathbb{R}^p/_{S_p}$, where S_p is the symmetric group on the finite set $\{1, \ldots, p\}$.

[2] Again, since the eigenvalues can be multiple, $\lambda(A)$ is in fact a multiset. It can also be regarded as an element o the quotient space $\mathbb{C}^n/_{S_n}$, where S_n is the symmetric group on the finite set $\{1, \ldots, n\}$.

It can be easily verified that, for a matrix $A \in \mathbb{R}^{m\times n}$ of rank $r \leq \min\{m,n\}$, matrices $A^TA \in \mathbb{R}^{n\times n}$ and $AA^T \in \mathbb{R}^{m\times m}$ are symmetric and positive semidefinite. The relationship between the singular value decomposition of the matrix A, and the spectral decomposition of the above two matrices is shown in the following theorem.

Theorem 2.1.6. *If $A \in \mathbb{R}^{m\times n}$, then the following holds:*

- $$V^TA^TAV = \operatorname{diag}(\sigma_1^2,\ldots,\sigma_r^2,\underbrace{0,\ldots,0}_{n-r}), \quad \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$$

  that is, the squares of singular values of the matrix A are the eigenvalues of the matrix A^TA, and $n-r$ of them are zeros. The columns of the matrix V are the corresponding eigenvectors.

- $$U^TAA^TU = \operatorname{diag}(\sigma_1^2,\ldots,\sigma_r^2,\underbrace{0,\ldots,0}_{m-r}), \quad \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$$

  that is, the squares of singular values of the matrix A are the eigenvalues of the matrix AA^T, and $m-r$ of them are zeros. The columns of the matrix U are the corresponding eigenvectors.

There is another way to associate singular values with eigenvalues.

Theorem 2.1.7 ([35, p. 427]). *Let $A \in \mathbb{R}^{m\times n}$, $m \geq n$, then the Jordan–Wielandt matrix*

$$\begin{bmatrix} 0 & A^T \\ A & 0 \end{bmatrix} \in \mathbb{R}^{(m+n)\times(m+n)}$$

is a symmetric matrix with eigenvalues equal to $\{\sigma_1,\ldots,\sigma_n,-\sigma_1,\ldots,-\sigma_n,0,\ldots,0\}$, where in the case when $m > n$, zero has multiplicity $m-n$. Moreover, if $A = U\Sigma V^T$ is the singular value decomposition of A with $U = [U_1\ U_2]$, $U_1 \in \mathbb{R}^{m\times n}$ and $U_2 \in \mathbb{R}^{m\times(m-n)}$, then

$$\begin{bmatrix} 0 & A^T \\ A & 0 \end{bmatrix} = Q\operatorname{diag}(\sigma_1,\ldots,\sigma_n,-\sigma_1,\ldots,-\sigma_n,\underbrace{0,\ldots,0}_{m-n})Q^T,$$

where $Q \in \mathbb{R}^{(m+n)\times(m+n)}$ is an orthogonal matrix, defined by

$$Q = \frac{1}{\sqrt{2}}\begin{bmatrix} V & V & 0 \\ U_1 & -U_1 & \sqrt{2}U_2 \end{bmatrix}.$$

Singular values are important for the characterization of unitarily invariant norms.

Definition 2.1.8. *A norm $\|\cdot\|$ on $\mathbb{C}^{m\times n}$ is **unitarily invariant** if it satisfies*

$$\|U^*AV\| = \|A\|$$

for all unitary matrices U and V.

Let $A = U\Sigma V^T$ be the singular value decomposition of $A \in \mathbb{R}^{m\times n}$. Since U and V are real orthogonal matrices, and hence unitary,

$$\|A\| = \|\Sigma\|,$$

for every unitarily invariant norm $\|\cdot\|$. Thus $\|A\|$ is a function Φ of the singular values of A, with certain properties. The properties of the matrix norm suggest the following definition.

Definition 2.1.9. *A function $\Phi : \mathbb{R}^n \to \mathbb{R}$ is a **symmetric gauge function** if it satisfies the following conditions.*

1. $x \neq 0 \Rightarrow \Phi(x) > 0$, *for* $x \in \mathbb{R}^n$.
2. $\Phi(\rho x) = |\rho|\Phi(x)$, *for* $x \in \mathbb{R}^n$ *and* $\rho \in \mathbb{R}$.
3. $\Phi(x + y) \leq \Phi(x) + \Phi(y)$, *for* $x, y \in \mathbb{R}^n$.
4. *For any permutation matrix* P *and for* $x \in \mathbb{R}^n$ *we have* $\Phi(Px) = \Phi(x)$.
5. $\Phi(|x|) = \Phi(x)$, *for* $x \in \mathbb{R}^n$.

If Φ is a symmetric gauge function and if $\|\cdot\|_\Phi$ is defined by

$$\|A\|_\Phi = \Phi(\sigma_1, \ldots, \sigma_p), \qquad p = \min\{m, n\}, \tag{2.5}$$

where $\sigma_1,\ldots,\sigma_p$ are the singular values of A, then the following theorem will describe the relationship between $\|\cdot\|_\Phi$ and unitarily invariant norms.

Theorem 2.1.10 (von Neumann [89, p. 78]). *Let Φ be a symmetric gauge function on $\mathbb{R}^p$, where $p = \min\{m, n\}$, and let $\|\cdot\|_\Phi$ be defined by (2.5). Then $\|\cdot\|_\Phi$ is a unitarily invariant norm on $\mathbb{C}^{m\times n}$. Conversely, if $\|\cdot\|$ is a unitarily invariant norm on $\mathbb{C}^{m\times n}$, then there is a symmetric gauge function Φ on $\mathbb{R}^p$ such that $\|A\| = \|A\|_\Phi$ for all $A \in \mathbb{C}^{m\times n}$.*

The most important matrix norms $\|\cdot\|_2$ and $\|\cdot\|_F$ are unitarily invariant norms, and the immediate consequence of Theorem 2.1.6 and Theorem 2.1.10 is that they can be characterized in terms of the singular values in the following way:

$$\|A\|_F = \sqrt{\operatorname{trace}(A^T A)} = \sqrt{\sum_{i=1}^{p} \sigma_i^2}, \qquad p = \min\{m, n\} \tag{2.6}$$

$$\|A\|_2 = \sqrt{\operatorname{spr}(A^T A)} = \sigma_1. \tag{2.7}$$

The SVD also indicates how near the given matrix is to the closest matrix of lower rank.

Theorem 2.1.11 ([35, p. 73]). *Let the SVD of $A \in \mathbb{R}^{m\times n}$ be given by Theorem 2.1.1. If $k < r = \mathrm{rank}(A)$ and*

$$A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^T$$

then, for every unitarily invariant norm $\|\cdot\|$,

$$\min_{\mathrm{rank}(B)\le k} \|A - B\| = \|A - A_k\|,$$

and specially

$$\begin{aligned} \min_{\mathrm{rank}(B)\le k} \|A - B\|_2 &= \|A - A_k\|_2 = \sigma_{k+1}, \\ \min_{\mathrm{rank}(B)\le k} \|A - B\|_F &= \|A - A_k\|_F = \sqrt{\sum_{i=k+1}^{r} \sigma_i^2}. \end{aligned}$$

Theorem 2.1.11 says that the smallest singular value of A is the 2 norm distance of A to the set of all rank-deficient matrices. The drawback of Theorem 2.1.11 is that A_k, which is the best rank k approximation to the matrix A, generally differs from A in all its elements. In some applications it is necessary to find a best rank k approximation to A that leaves some columns of A fixed. Let us assume that the fixed columns are at the beginning of the matrix A, and let

$$A = [\ A_1 \quad A_2\ ], \tag{2.8}$$

where A_1 has ℓ columns. Then, we are considering the following problem: find a matrix $A_{k,2}$ such that $\mathrm{rank}([\ A_1 \quad A_{k,2}\ ]) \le k$, and

$$\min_{\mathrm{rank}([\ A_1 \quad B_2\ ])\le k} \|[\ A_1 \quad A_2\ ] - [\ A_1 \quad B_2\ ]\| = \|[\ A_1 \quad A_2\ ] - [\ A_1 \quad A_{k,2}\ ]\|, \tag{2.9}$$

for unitarily invariant matrix norm $\|\cdot\|$. Let us denote by $\mathbf{H}_k$ the operator that maps A onto A_k from Theorem 2.1.11, with the convention that if k is greater than the number of columns of A, then $\mathbf{H}_k$ is the identity. The following theorem solves the given problem.

Theorem 2.1.12 ([32, pp. 319–321]). *Let $A \in \mathbb{R}^{m\times n}$ be partitioned as in (2.8) where A_1 has ℓ columns, and let $p = \mathrm{rank}(A_1)$. Let P denote the orthogonal projection onto the column space of A and $P^\perp$ the orthogonal projection onto its orthogonal complement. If $p \le k$ then the matrix*

$$A_{k,2} = PA_2 + \mathbf{H}_{k-p}(P^\perp A_2)$$

satisfies (2.9).

There are many important properties of the singular values of an $m \times n$ matrix. First of them claims that the singular values satisfy the following "minimax" characterization.

Theorem 2.1.13 ([35, p. 428]). *If $A \in \mathbb{R}^{m\times n}$, then*

$$\sigma_k = \max_{\substack{\mathcal{S}\subset\mathbb{R}^n,\ \mathcal{T}\subset\mathbb{R}^m \\ \dim(\mathcal{S})=k \\ \dim(\mathcal{T})=k}} \min_{\substack{x\in\mathcal{S},\ y\in\mathcal{T} \\ \|x\|_2=1,\ \|y\|_2=1}} y^T A x = \max_{\substack{\mathcal{S}\subset\mathbb{R}^n \\ \dim(\mathcal{S})=k}} \min_{\substack{x\in\mathcal{S} \\ \|x\|_2=1}} \|Ax\|_2 \quad k = 1,\ldots,\min\{m,n\}.$$

Some other properties follow from the previous theorem.

Corollary 2.1.14 (Interlacing Property [35, p. 428]). *Let $A = [\ a_1\ \cdots\ a_n\ ] \in \mathbb{R}^{m\times n}$ be a column partitioning with $m \geq n$. If $A_k = [\ a_1\ \cdots\ a_k\ ]$ then for $k = 1,\ldots,n-1$ the following interlacing property holds:*

$$\sigma_i(A) \geq \sigma_i(A_k) \geq \sigma_{i+n-k}(A), \quad i = 1,\ldots,k,$$

2.2 Applications of the SVD

What follows is a list of problems in numerical linear algebra that may be solved by means of the singular value decomposition.

2.2.1 Computing the Inverse of a Nonsingular Square Matrix

A square matrix $A \in \mathbb{R}^{n\times n}$ is nonsingular if and only if $\sigma_i \neq 0$, $i = 1,\ldots,n$. Then from equation (2.3) it follows that its inverse is given by

$$A^{-1} = V\Sigma^{-1}U^T, \qquad \Sigma^{-1} = \operatorname{diag}(\sigma_1^{-1},\ldots,\sigma_n^{-1}),$$

where $A^{-1} \in \mathbb{R}^{n\times n}$ is such that $AA^{-1} = A^{-1}A = I$.

2.2.2 Computing the Pseudo–Inverse of a Matrix

We can extend the concept of inverse to singular and even rectangular matrices. Such a pseudo–inverse must satisfy weaker conditions than the standard inverse. One way to define pseudo–inverse is the following. For the rectangular matrix $A \in \mathbb{R}^{m\times n}$, matrix $X \in \mathbb{R}^{n\times m}$ is its pseudo–inverse if and only if X satisfies the *Moore–Penrose conditions* [79]:

1. $AXA = A$
2. $XAX = X$
3. $(AX)^T = AX$
4. $(XA)^T = XA$

The exact form of the pseudo–inverse will be defined by means of the SVD.

Theorem 2.2.1 ([35, p. 243]). *Let $A \in \mathbb{R}^{m\times n}$, then there exists a unique matrix $X \in \mathbb{R}^{n\times m}$ which satisfies the Moore–Penrose conditions. This matrix is of the form*

$$A^\dagger = V \begin{bmatrix} \Sigma_+^{-1} & 0 \\ 0 & 0 \end{bmatrix} U^T, \quad \text{where } A = U \begin{bmatrix} \Sigma_+ & 0 \\ 0 & 0 \end{bmatrix} V^T$$

is the singular value decomposition of the matrix A, defined in Theorem 2.1.1.

The pseudo–inverse is the unique minimal Frobenius norm solution of the problem

$$\min_{X\in\mathbb{R}^{n\times m}} \|AX - I_m\|_F.$$

From the Moore–Penrose conditions it follows that $AA^\dagger$ and $A^\dagger A$ are orthogonal projections onto range(A) and range(A^T), respectively:

$$AA^\dagger = U_1 U_1^T, \qquad A^\dagger A = V_1 V_1^T.$$

2.2.3 The Condition Number of a Matrix

The condition number of a matrix $A \in \mathbb{R}^{m\times n}$ in 2 norm is given by

$$\kappa_2(A) = \|A\|_2 \|A^\dagger\|_2.$$

From the observations above and Theorem 2.1.1 we can conclude that

$$\kappa_2(A) = \frac{\sigma_1}{\sigma_n}.$$

2.2.4 Solving the Orthogonal Procrustes Problem

The solution of the orthogonal Procrustes problem satisfies:

$$\min_{Q\in\mathbb{R}^{n\times n},\ Q^TQ=I_n} \|AQ - B\|_F, \tag{2.10}$$

where $A, B \in \mathbb{R}^{m\times n}$ and $m \geq n$. By the orthogonality of the matrix Q, the problem (2.10) transforms into

$$\min \|AQ - B\|_F^2 = \min(\text{trace}(A^TA) + \text{trace}(B^TB) - 2\ \text{trace}(Q^TA^TB)),$$

and is equivalent to the problem of maximizing trace(Q^TA^TB). The maximizing Q can be found by calculating the SVD of A^TB, [35, p. 582]. If

$$U^T(A^TB)V = \Sigma = \text{diag}(\sigma_1, \ldots, \sigma_n)$$

is the SVD of the matrix A^TB and we define the orthogonal matrix Z by $Z = V^TQ^TU$, then

$$\text{trace}(Q^TA^TB) = \text{trace}(Q^TU\Sigma V^T) = \text{trace}(Z\Sigma) = \sum_{i=1}^{n} z_{ii}\sigma_i \leq \sum_{i=1}^{n} \sigma_i.$$

The upper bound is attained by setting $Q = UV^T$ for then $Z = I_n$.

In the unbalanced case (see [27]), when $Q \in \mathbb{R}^{n\times k}$, $B \in \mathbb{R}^{m\times k}$ and $n > k$, the problem

$$\min_{Q\in\mathbb{R}^{n\times k},\ Q^TQ=I_k} \|AQ - B\|_F, \tag{2.11}$$

can also be solved by means of singular value decomposition. If we take the QR factorization of A

$$A = P \begin{bmatrix} R \\ 0 \end{bmatrix},$$

where $P = [\ P_1 \quad P_2\ ] \in \mathbb{R}^{m\times m}$ is orthogonal, $P_1 \in \mathbb{R}^{m\times n}$, and $R \in \mathbb{R}^{n\times n}$, we obtain an equivalent problem

$$\min_{Q\in\mathbb{R}^{n\times k},\ Q^TQ=I_k} \|RQ - P_1^TB\|_F,$$

with the square matrix R. Thus we can assume that the matrix A is square without loss of generality. By using Lagrange multipliers, the minimization problem (2.11) can be reduced to the *secular equation*

$$F(L) = Q_L^TQ_L - I = 0, \tag{2.12}$$

where Q_L is a solution of the *normal equation*

$$A^TAQ + QL = A^TB, \tag{2.13}$$

and $L = [\ell_{ij}] \in \mathbb{R}^{k\times k}$ is the symmetric matrix of Lagrangian multipliers. Let $A = U\Sigma V^T$ be the singular value decomposition of A, with $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_n)$, and let $L = W\Lambda W^T$ be the spectral decomposition of the fixed symmetric matrix L, with $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_k)$, then there is a unique solution of (2.13) if the eigenvalues λ_j of L satisfy

$$\lambda_j + \sigma_i^2 \neq 0, \qquad i = 1, \ldots, n,\ j = 1, \ldots, k.$$

Now, for $\bar{Q} = V^TQW$ and $\bar{B} = U^TBW$ we obtain equivalent minimization form

$$\min_{\bar{Q}\in\mathbb{R}^{n\times k},\ \bar{Q}^T\bar{Q}=I_k} \|\Sigma\bar{Q} - \bar{B}\|_F,$$

with its normal equation equal to

$$\Sigma^2\bar{Q} + \bar{Q}\Lambda = \Sigma\bar{B},$$

which can be easily solved. Its solution is of the form

$$\bar{Q}_L = [\bar{q}_{ij}], \quad \text{with} \quad \bar{q}_{ij} = \frac{\sigma_i\bar{b}_{ij}}{\sigma_i^2 + \lambda_j},$$

and then

$$F(L) = [F(L)_{ij}], \quad \text{with} \quad F(L)_{ij} = \sum_{s=1}^{k} \frac{\sigma_s^2\bar{b}_{si}\bar{b}_{sj}}{(\sigma_s^2 + \lambda_i)(\sigma_s^2 + \lambda_j)} - \delta_{ij}.$$

2.2.5 Finding the Intersection of Null Spaces

Let $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{p\times n}$ be given, and consider the problem of finding an orthonormal basis for $\text{null}(A) \cap \text{null}(B)$. One way of solving the problem is to exploit the following theorem.

Theorem 2.2.2 ([35, p. 583]). *Suppose $A \in \mathbb{R}^{m\times n}$ and let $\{z_1, \ldots, z_s\}$ be an orthonormal basis for* $\text{null}(A)$. *Define $Z = [z_1, \ldots, z_s]$ and let $\{w_1, \ldots, w_t\}$ be an orthonormal basis for* $\text{null}(BZ)$ *where $B \in \mathbb{R}^{p\times n}$. If $W = [w_1, \ldots, w_t]$, then the columns of ZW form an orthonormal basis for* $\text{null}(A) \cap \text{null}(B)$.

The SVD is used to compute the orthonormal basis $\{y_i\}$ of $\text{null}(A) \cap \text{null}(B)$ in the following way:

- Compute the SVD $U_A^T A V_A = \text{diag}(\sigma_i)$, $\quad r = \text{rank}(A)$
- $C = BV_A(:, r+1:n)$ $\quad$ from Corollary 2.1.4
- Compute the SVD $U_C^T C V_C = \text{diag}(\gamma_i)$, $\quad q = \text{rank}(C)$
- $Y = V_A(:, r+1:n)V_C(:, q+1:n-r)$ $\quad$ from Corollary 2.1.4

where $Y = [y_1, \ldots, y_{n-r-q}]$.

2.2.6 Finding Angles Between Subspaces

Let $\mathcal{X}$ and $\mathcal{Y}$ be subspaces of $\mathbb{R}^m$ whose dimensions satisfy

$$p = \dim(\mathcal{X}) \geq \dim(\mathcal{Y}) = q \geq 1.$$

The *principle angles* $\theta_1, \ldots, \theta_q \in [0, \pi/2]$ between $\mathcal{X}$ and $\mathcal{Y}$ are defined [35, p. 584] recursively by

$$\cos(\theta_k) = \max_{x\in\mathcal{X}} \max_{y\in\mathcal{Y}} x^T y = x_k^T y_k$$

subject to:

$$\begin{array}{ll} \|x\|_2 = \|y\|_2 = 1 & \\ x^T x_i = 0 & i = 1, \ldots, k-1 \\ y^T y_i = 0 & i = 1, \ldots, k-1 \end{array}$$

The vectors $\{x_1, \ldots, x_q\}$ and $\{y_1, \ldots, y_q\}$ are called the *principal vectors* between the subspaces $\mathcal{X}$ and $\mathcal{Y}$, and the principal angles defined as above satisfy $0 \leq \theta_1 \leq \cdots \leq \theta_q \leq \pi/2$. If $p = q$ then

$$\text{dist}(\mathcal{X}, \mathcal{Y}) = \|P_{\mathcal{X}} - P_{\mathcal{Y}}\|_2 = \sqrt{1 - \cos(\theta_p)^2} = \sin(\theta_p), \tag{2.14}$$

is the distance between equidimensional subspaces, where $P_{\mathcal{X}}$ and $P_{\mathcal{Y}}$ are orthogonal projections onto $\mathcal{X}$ and $\mathcal{Y}$. In that case the angle $\angle(\mathcal{X}, \mathcal{Y})$ can be defined as [90]

$$\begin{aligned} \angle(\mathcal{X}, \mathcal{Y}) &= \arcsin(\text{dist}(\mathcal{X}, \mathcal{Y})) = \theta_p \\ &= \max_{x\in\mathcal{X}} \angle(x, \mathcal{Y}) = \max_{x\in\mathcal{X}} \min_{y\in\mathcal{Y}} \angle(x, y) \\ &= \max_{y\in\mathcal{Y}} \angle(y, \mathcal{X}) = \max_{y\in\mathcal{Y}} \min_{x\in\mathcal{X}} \angle(x, y). \end{aligned} \tag{2.15}$$

Moreover

$$\angle(x, \mathcal{Y}) = \min_{y \in \mathcal{Y}} \angle(x, y) = \min_{y \in \mathcal{Y}} \arccos\left(\frac{x^T y}{\|x\|_2 \|y\|_2}\right), \tag{2.16}$$

and all vectors x and y are taken to be different from zero. For more information on angles between subspaces, see [36] and [90].

If the columns of $X \in \mathbb{R}^{m \times p}$ and $Y \in \mathbb{R}^{m \times q}$ define orthonormal bases for $\mathcal{X}$ and $\mathcal{Y}$ respectively [35, p. 585], then

$$\max_{\substack{x \in \mathcal{X} \\ \|x\|_2 = 1}} \max_{\substack{y \in \mathcal{Y} \\ \|y\|_2 = 1}} x^T y = \max_{\substack{u \in \mathbb{R}^p \\ \|u\|_2 = 1}} \max_{\substack{v \in \mathbb{R}^q \\ \|v\|_2 = 1}} u^T (X^T Y) v.$$

From the minimax characterization of the singular values in Theorem 2.1.13, it follows that if

$$U^T (X^T Y) V = \operatorname{diag}(\sigma_1, \ldots, \sigma_q), \quad \text{with} \quad \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_q \tag{2.17}$$

is the SVD of $X^T Y$, then we may define x_k, y_k and θ_k by

$$\begin{bmatrix} x_1 & \ldots & x_q \end{bmatrix} = XU, \tag{2.18}$$

$$\begin{bmatrix} y_1 & \ldots & y_q \end{bmatrix} = YV, \tag{2.19}$$

$$\cos(\theta_k) = \sigma_k, \qquad k = 1, \ldots, q. \tag{2.20}$$

2.2.7 Finding the Intersection of Subspaces

The same procedure can be used to compute an orthogonal basis for $\operatorname{range}(A) \cap \operatorname{range}(B)$ where $A \in \mathbb{R}^{m \times p}$ and $B \in \mathbb{R}^{m \times q}$.

Theorem 2.2.3 ([35, p. 586]). *Let $\cos(\theta_k)$ for $k = 1, \ldots, q$, $U = \begin{bmatrix} u_1 & \cdots & u_q \end{bmatrix}$ and $V = \begin{bmatrix} v_1 & \cdots & v_q \end{bmatrix}$ be defined by (2.17)–(2.20). If the index s is defined by $1 = \cos(\theta_1) = \cdots = \cos(\theta_s) > \cos(\theta_{s+1})$, then we have*

$$\operatorname{range}(A) \cap \operatorname{range}(B) = \operatorname{span}\{u_1, \ldots, u_s\} = \operatorname{span}\{v_1, \ldots, v_s\}.$$

2.2.8 Solving the Linear Least Squares Problem

Let $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$ be given, and consider a problem of finding vector $x \in \mathbb{R}^n$ such that it minimizes the following functional

$$\min_{x \in \mathbb{R}^n} \|Ax - b\|_2.$$

In case the matrix A is rank deficient, we are searching for the minimizer x with minimal 2 norm. Then, its solution can be found by using the SVD of the matrix A.

Theorem 2.2.4 ([35, p. 242]). *Suppose $A = U \Sigma V^T$ is the SVD of $A \in \mathbb{R}^{m \times n}$ with $r = \operatorname{rank}(A)$. If $U = [u_1, \ldots, u_m]$ and $V = [v_1, \ldots, v_n]$ are column partitionings and $b \in \mathbb{R}^m$, then*

$$x_{LS} = \sum_{i=1}^{r} \frac{u_i^T b}{\sigma_i} v_i$$

minimizes $\|Ax - b\|_2$ *and has the smallest* 2 *norm of all minimizers. Moreover*

$$r_{LS}^2 = \|Ax_{LS} - b\|_2^2 = \sum_{i=r+1}^{m} (u_i^T b)^2.$$

In fact, $x_{LS} = A^\dagger b$ and $r_{LS} = \|(I - AA^\dagger)b\|_2$. All solution of the linear least squares problem are given by

$$\hat{x}_{LS} = x_{LS} + V_2 y, \qquad y \in \mathbb{R}^{n-r},$$

where V_2 is null basis matrix for A.

2.2.9 Solving the Linear Total Least Squares Problem

Let $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{m\times k}$ be given, and we want to solve the following *total least squares (TLS) problem* (see [51] and [34])

$$\min_{\text{range}(B+R)\subseteq\text{range}(A+E)} \|D \cdot [\ E \quad R\ ] \cdot T\|_F, \tag{2.21}$$

where $E \in \mathbb{R}^{m\times n}$, $R \in \mathbb{R}^{m\times k}$, and the matrices $D = \text{diag}(d_1, \ldots, d_m)$ and $T = \text{diag}(t_1, \ldots, t_{n+k})$ are nonsingular weight matrices. If $[E_0\ R_0]$ solves (2.21), then any $X \in \mathbb{R}^{n\times k}$ that satisfies

$$(A + E_0)X = B + R_0$$

is said to be a *TLS solution.*

Theorem 2.2.5 ([35, p. 577]). *Let A, B, D, and T be as above and assume* $m \geq n+k$. *Let*

$$C = D[\ A \quad B\ ]T = \begin{bmatrix} C_1 & C_2 \\ n & k \end{bmatrix}$$

have the SVD given by $U^T CV = \text{diag}(\sigma_1, \ldots, \sigma_{n+k}) = \Sigma$ *where U, V, and* Σ *are partitioned as follows:*

$$U = \begin{bmatrix} U_1 & U_2 \\ n & k \end{bmatrix}, \qquad V = \begin{bmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \\ n & k \end{bmatrix} \begin{matrix} n \\ k \end{matrix},$$

$$\Sigma = \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \\ n & k \end{bmatrix} \begin{matrix} n \\ k \end{matrix}.$$

If $\sigma_n(C_1) > \sigma_{n+1}$, *then the matrix* $[E_0\ R_0]$ *defined by*

$$D[\ E_0 \quad R_0\ ]T = -U_2\Sigma_2[\ V_{12}^T \quad V_{22}^T\ ]$$

solves (2.21). If $T_1 = \text{diag}(t_1, \ldots, t_n)$ *and* $T_2 = \text{diag}(t_{n+1}, \ldots, t_{n+k})$ *then the matrix*

$$X_{TLS} = -T_1 V_{12} V_{22}^{-1} T_2^{-1}$$

exists and is the unique solution to $(A + E_0)X = B + R_0$.

Remark 2.2.6 ([35, pp. 578–579]). *If $\sigma_n = \sigma_{n+1}$ then the TLS problem may still have a solution, although it may not be unique. In this case, it may be desirable to single out a "minimal norm" solution.*

Many problems in physics, biology, chemistry, informatics and other fields of science can be reduced to a linear algebra problem. This is the reason why the singular value decomposition is widely used as a tool for solving such problems. In the rest of this section several examples will be presented, which illustrate application of the SVD in other sciences.

2.2.10 Integral Equations and the Truncated Singular Value Decomposition

Many physical, geological or medical measurements can be modelled by a *Fredholm integral equation*. We will be focused on the *Fredholm integral equation of the first kind*.

Definition 2.2.7 ([62, pp. 62–63]). *Let $\Delta = [a, b] \subseteq \mathbb{R}$ be a segment in $\mathbb{R}$ and let $C(\Delta)$ denotes the Banach space of continuous real functions on Δ, with the norm defined by*

$$\|x\| = \max\{|x(t)| \ : \ t \in \Delta\}.$$

Let $k : \Delta' \times \Delta \longrightarrow \mathbb{R}$ be a continuous function, where Δ' is a segment not necessarily equal to Δ. Then, for every $x \in C(\Delta)$, the function $t \longmapsto k(s,t)x(t)$ is Riemann integrable and the function

$$y(s) = \int_a^b k(s,t)x(t)dt, \qquad s \in \Delta' \tag{2.22}$$

*is continuous on Δ'. Equation (2.22) represents the **Fredholm integral equation of the first kind**.*

Equation (2.22) also defines a mapping $K : C(\Delta) \longrightarrow C(\Delta')$ such that $x \longmapsto y$, and it can be substituted by the equation

$$y = Kx. \tag{2.23}$$

*The operator K is referred to as the **Fredholm integral operator**, and the function k as the **kernel of the Fredholm integral operator** K.*

The most usual problem concerning the Fredholm integral operator is finding the function $x \in C(\Delta)$ for given $y \in C(\Delta')$, such that (2.22) holds (see [92]).

The Fredholm integral operators have several important properties, which are presented in the following theorems and definitions.

Theorem 2.2.8 ([62, p. 65]). *If $k : \Delta' \times \Delta \longrightarrow \mathbb{R}$ is a continuous function, then the operator K*

$$y = Kx$$

is a continuous linear operator from the Banach space $C(\Delta)$ to the Banach space $C(\Delta')$. The norm of the operator K is given by

$$\|K\| = \max_{s\in\Delta'} \int_{\Delta} |k(s,t)|dt.$$

Definition 2.2.9 ([62, p. 200]). *Let $\mathcal{X}$ and $\mathcal{Y}$ be normed spaces. Linear operator $K : \mathcal{X} \longrightarrow \mathcal{Y}$ is **compact** if it maps a unit ball from the space $\mathcal{X}$ into a relative compact set in the space $\mathcal{Y}$. The relative compact set is a set where every sequence has a Cauchy subsequence.*

Definition 2.2.10 ([62, p. 226]). *The space $C(\Delta)$ is unitary with the scalar product defined by*

$$\langle x, y\rangle = \int_{\Delta} x(t)y(t)dt.$$

The Hilbert space $L_2(\Delta)$ is the completion of the space $C(\Delta)$.

Theorem 2.2.11 ([62, p. 226]). *If the kernel k of the operator (2.22) is continuous on $\Delta' \times \Delta$, then the operator K in (2.23) is a compact operator from the unitary space $C(\Delta)$ to the Banach space $C(\Delta')$.*

Since the unitary space $C(\Delta)$ is dense in the Hilbert space $L_2(\Delta)$, and since the continuous operator K is defined on the unitary space $C(\Delta)$, the operator K can be expanded by continuity to the continuous operator $\tilde{K} : L_2(\Delta) \longrightarrow C(\Delta')$. Since K is a compact operator, $\tilde{K}$ is also a compact operator. So, the operator defined by (2.22) is a compact operator from $L_2(\Delta)$ to $C(\Delta')$.

Corollary 2.2.12 ([62, p. 227]). *If the kernel k of the operator (2.22) is continuous on $\Delta' \times \Delta$, then the operator K in (2.23) is compact from the Hilbert space $L_2(\Delta)$ to the unitary space $C(\Delta')$.*

Theorem 2.2.13 ([62, p. 212]). *Let $\mathcal{X}$ and $\mathcal{Y}$ be Hilbert spaces and let $K : \mathcal{X} \longrightarrow \mathcal{Y}$ be a compact operator with infinite range. Then, there exist orthonormal sequences $\{e_i\}$ in $\mathcal{X}$, and $\{f_i\}$ in $\mathcal{Y}$, and a sequence of real numbers $\{\boldsymbol{\sigma}_i\}$ where*

$$\boldsymbol{\sigma}_1 \geq \boldsymbol{\sigma}_2 \geq \cdots \geq \boldsymbol{\sigma}_i \geq \cdots > 0, \qquad \lim_{i\to\infty} \boldsymbol{\sigma}_i = 0,$$

such that every $x \in \mathcal{X}$ can be expressed as

$$x = x_0 + \sum_{i=1}^{\infty} \langle x, e_i\rangle e_i, \qquad \text{where} \quad Kx_0 = 0,$$

and

$$Kx = \sum_{i=1}^{\infty} \boldsymbol{\sigma}_i \langle x, e_i\rangle f_i. \tag{2.24}$$

*Equation (2.24) is the **Schmidt representation of the operator** K. The scalars $\boldsymbol{\sigma}_i > 0$ are referred to as* singular values of the operator K.

The presented results show that the Fredholm integral operator is compact and that its singular values tend to zero. Hence, solving the Fredholm integral equation represents an ill-posed problem, and the solution is extremely sensitive to measurement noise of the input parameters.

A numerical procedure for solving the Fredholm integral equation is described in [83] and [84]. The equation (2.22) is usually used in physics to model instrument distortion in measuring an unknown function $x(t)$. The first step in the discretization of the Fredholm integral equation would be replacement of the equation (2.22) by a system of equations

$$y_i = \int_a^b K_i(t)x(t)dt + \xi_i, \quad i = 1, \ldots, m,$$

where $K_i(t) = K(s_i, t)$ are well known response functions of the instruments, $y_i = y(s_i)$ are measured values, corresponding to a discrete mesh $s_1, s_2, \ldots, s_m$ of collocation points, and ξ_i are random, zero–mean measuring errors. The next step is discretization of the integral by means of numerical integration, where the error of the discretization should be smaller than the measuring errors. Thus the initial infinite dimensional problem (2.22) is transformed into the finite dimensional problem

$$\bar{y} = \bar{K}\bar{x} + \xi, \tag{2.25}$$

where $\bar{y} = [y_i] \in \mathbb{R}^m$ is the vector of measurements, $\bar{K} \in \mathbb{R}^{m\times n}$ is a known matrix with $m \geq n$, and $\bar{x} \in \mathbb{R}^n$ is an unknown vector whose components are either discrete point estimates of $x(t)$ on some mesh $t_1, t_2, \ldots, t_n$, or the unknown coefficient in an expansion of $x(t)$ in terms of some set of basis functions. The vector $\xi \in \mathbb{R}^m$ is a vector of random measuring errors satisfying

$$E(\xi) = 0, \qquad E(\xi\xi^T) = S^2,$$

where E is the expectation operator, $0 \in \mathbb{R}^m$ is the zero vector and $S^2 \in \mathbb{R}^{m\times m}$ is the positive definite variance-covariance matrix for ξ. In most problems the measurement errors are assumed to be are statistically independent, so

$$S^2 = \mathrm{diag}(s_1^2, \ldots, s_m^2),$$

where $s_1^2, \ldots, s_m^2$ are known standard deviations of the error.

Remark 2.2.14. *It should be noted that the discretization of an ill-posed Fredholm integral equation of the first kind yields an ill-conditioned linear system. In general, the higher the dimensions of the discretization matrix, the closer the finite-dimensional problem to the ill-posed continuous problem and, consequently, the more ill conditioned the algebraic problem becomes [37]. That means that singular values of the matrix $\bar{K}$ decay rapidly, and that its condition number is large. The smallest singular values are usually of the same order as the roundoff error. The computation of the solution as $\bar{x} = \bar{K}^\dagger\bar{y}$ is extremely unstable, and computed $\bar{x}$ is useless.*

For example, for the computation of the particle size distribution in photon correlation spectroscopy [31], a Fredholm integral equation of the first kind has to be solved, where

$$k(s, t) = e^{-st}.$$

When the Fredholm integral operator is discretized, the obtained singular values distributions are shown in Figure 2.1.

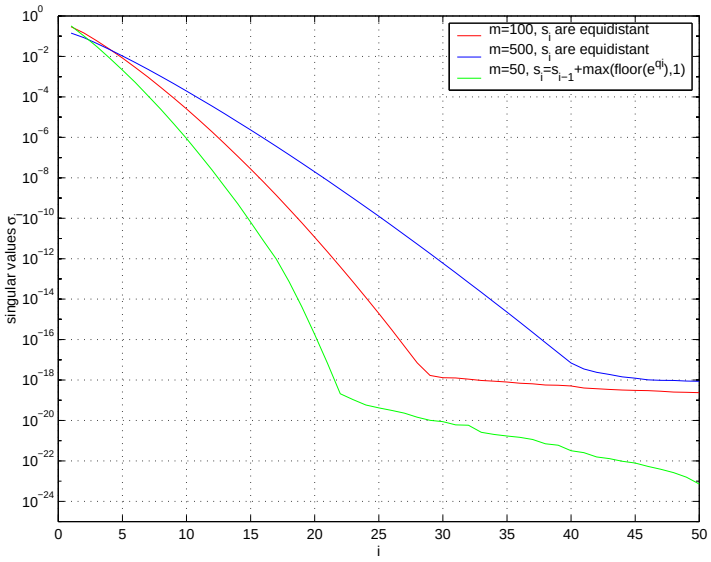


Figure 2.1: Singular values distribution of the discretized Fredholm integral operator.

As we can see from Figure 2.1, increasing the number of collocation points improves the situation just a little bit, nevertheless the problem remains ill–posed. The singular values decay so fast that they drop below the machine epsilon of the single precision somewhere around the 20*-th singular value.*

Usually, it is assumed that the errors are samples from a multivariate normal distribution:

$$\xi \sim N(0, S^2).$$

It is advantageous to scale (2.25) with the matrix S^{-1} as in [83]. Let

$$b = S^{-1}\bar{y}, \quad A = S^{-1}\bar{K}, \quad \eta = S^{-1}\xi.$$

Then by [83], (2.25) is transformed to

$$b = A\bar{x} + \eta, \qquad \eta \sim N(0, I_m), \tag{2.26}$$

where $I_m \in \mathbb{R}^{m\times m}$ is the identity matrix. If $\tilde{x}$ is an approximation of $\bar{x}$, then its residual $\tilde{r} = b - A\tilde{x}$ should be an approximation of $\eta = b - A\bar{x}$. Thus, an approximation $\tilde{x}$ is acceptable only if $\tilde{r}$ is a plausible sample from the $N(0, I_m)$ distribution. Further, by [83] it follows that

$$\|b - A\bar{x}\|_2^2 \sim \chi^2(m),$$

where $\chi^2(m)$ denotes the Chi–squared distribution with m degrees of freedom, and hence

$$E(\|b - A\bar{x}\|_2^2) = m, \qquad \mathrm{Var}(\|b - A\bar{x}\|_2^2) = 2m.$$

There are several ways to solve the problem (2.26). In [84] Rust proposed the following criteria for an approximation $\bar{x}$ to be accepted as a good approximation

1. The elements of $\tilde{r}$ should be distributed like $N(0,1)$.
2. The elements of $\tilde{r}$ should comprise a white noise time–series.
3. The squared norm $\tilde{r}^T\tilde{r}$ should lie in some interval $[m-\kappa\sqrt{2m}, m+\kappa\sqrt{2m}]$, with $|\kappa| \leq 2$.

Methods used for solving (2.26) are:

Solving the least squares problem

The solution is of the form

$$\tilde{x} = A^{\dagger}b.$$

The matrix A has always full column rank but it is ill-conditioned, so the components of $\tilde{x}$ are very sensitive to small perturbations in the components of b. The presence of measuring errors leads to the solution approximation which is totaly unreal. For example, in some physical measurement a very smooth solution is expected. Instead of this, the computed solution approximation oscillates wildly around the exact solution. The least squares solution usually does not satisfy any of Rust's criteria.

Regularization

The most widely used method for stabilizing the wildly oscillating least squares solution is to introduce a constraint on the solution $\tilde{x}$ of the form ([84])

$$\|Q(\tilde{x}-\tilde{x}_0)\|_2^2 \leq \beta^2.$$

Here, $\tilde{x}_0$ is an optional initial approximation of $\bar{x}$, Q is a matrix representation of the linear operator for the constraint, and β^2 is a constant determining the strength of the constraint. The approximation $\tilde{x}_\lambda$ is obtained by solving

$$\min\left(\|b-A\tilde{x}_\lambda\|_2^2 + \lambda\|Q(\tilde{x}_\lambda-\tilde{x}_0)\|_2^2\right),$$

where the parameter λ is a Lagrange multiplier whose value depends on the value of β^2. The solution is of the form

$$\tilde{x}_\lambda = (A^TA+\lambda^2Q^TQ)^{-1}(A^Tb+\lambda^2Q^TQ\tilde{x}_0).$$

The success of the regularization depends on the choice of the value λ. There are several ways to choose the optimal λ, which satisfy all of Rust's criteria, see [84].

Truncated singular value decomposition (TSVD)

Another often used method for solving such an ill–posed problem is the **truncated singular value decomposition**, which uses a rank $p < \min\{m,n\}$ approximation. If $A = U\Sigma V^T$ is the SVD of the matrix A, then by the result of Theorem 2.1.11,

$$A_p = \sum_{i=1}^{p} \sigma_i u_i v_i^T,$$

is the best rank p approximation of A. For $m \geq n$, let the matrices $U \in \mathbb{R}^{m\times m}$, $V \in \mathbb{R}^{n\times n}$ and $\Sigma \in \mathbb{R}^{m\times n}$ be partitioned as follows

$$U = \underset{p}{\left[\begin{array}{c} U_1 \end{array}\right.} \underset{n-p}{\begin{array}{c} U_2 \end{array}} \underset{m-n}{\left.\begin{array}{c} U_3 \end{array}\right]} \qquad V = \underset{p}{\left[\begin{array}{c} V_1 \end{array}\right.} \underset{n-p}{\left.\begin{array}{c} V_2 \end{array}\right]} \qquad \Sigma = \begin{array}{c} \left[\begin{array}{cc} \Sigma_1 & 0 \\ 0 & \Sigma_2 \\ 0 & 0 \end{array}\right] \\ \begin{array}{cc} p & n-p \end{array} \end{array} \begin{array}{l} p \\ n-p \\ m-n \end{array}$$

where $\sigma_p > \zeta\sigma_1$ and $\sigma_{p+1} < \zeta\sigma_1$ for some tolerance ζ. Then

$$A_p = U_1\Sigma_1 V_1^T.$$

Solving the least squares problem leads to minimization of the $\|r_{svd}\|_2^2 = \|A\bar{x} - b\|_2^2$, where

$$\|r_{svd}\|_2^2 = \|\Sigma_1 V_1^T\bar{x} - U_1^T b\|_2^2 + \|\Sigma_2 V_2^T\bar{x} - U_2^T b\|_2^2 + \|U_3^T b\|_2^2, \tag{2.27}$$

which is equivalent to the minimization of the first two terms in (2.27). The truncated SVD sets $\sigma_i = 0$ for $i = p+1, \ldots, n$ and minimizes only the first term in (2.27). The same result would be obtained if we solved the least squares problem for the matrix A_p

$$\min \|r_{tsvd}\|_2^2 = \min(\|\Sigma_1 V_1^T\tilde{x} - U_1^T b\|_2^2 + \|U_2^T b\|_2^2 + \|U_3^T b\|_2^2).$$

The important thing is to find a proper tolerance ζ or rank p so that it represents a compromise between the residual and the solution norm, keeping them both relatively small. Finally, the solution of the truncated singular value decomposition is of the form

$$\tilde{x} = \sum_{i=1}^{p} \frac{u_i^T b}{\sigma_i} v_i.$$

As Rust mentioned in [83], even for the most ill–posed problems, the matrix A is not rank deficient, so there is no good reason for setting any of the singular values to zero. This is the reason, why he proposed the next method.

Truncated vector $U^T b$

Rust in [83] suggested that instead of zeroing some of the singular values, one should zero those components of $U^T b$ that consist mostly of the random error. His idea is to pick a truncated level τ and require that solution approximation should satisfy

$$(V^T\tilde{x})_i = \begin{cases} \dfrac{(U^T b)_i}{\sigma_i} & , \text{ if } |U^T b|_i > \tau \\ 0 & , \text{ if } |U^T b|_i \leq \tau \end{cases} \qquad i = 1, \ldots, n.$$

The success of the proposed method depends again on the choice of the truncated level τ. The most simple way is to try several values of τ and choose the optimal truncated level so that solution approximation satisfies all of Rust's criteria.

The Fredholm integral equation appears in many problems such as, for example, opto–thermal skin measurements [93] and magnetic resonance imaging [55].

Example 2.2.15 (Geophysics). *An interesting example of a Fredholm integral equation can be found in [86] and [87]. It is concerned with gravity surveying. Variations of the density of subsurface rock give rise to variations of the gravity field at the Earth surface. Therefore, from measurements of the gravity field at the Earth surface, one can calculate density variations of subsurface rock. Variations of the vertical component of the gravity field $g(s)$ along a line s at the surface are related to variations $f(t)$ of the mass density along a line t ($0 \leq t \leq 1$) at depth d below the surface by the Fredholm integral equation of the first kind*

$$g(s) = \int_0^1 k(s,t)f(t)dt \tag{2.28}$$

with the kernel

$$k(s,t) = \frac{d}{(d^2 + (s-t)^2)^{3/2}}.$$

Dimensional constants, such as the gravity constant, have been omitted. In discrete form, we can write the relation between measurements $\bar{g} = [g_1, \ldots, g_m]^T$ of gravity variations at m points along a line at the surface and variations of the density $\bar{f} = [f_1, \ldots, f_n]$ at n points along a subsurface line as a linear regression model

$$\bar{g} = \bar{K}\bar{f} + \epsilon,$$

where ϵ is a vector of measurement errors, and the $m \times n$ matrix $\bar{K}$ is a discrete representation of the integral operator (2.28).

For the concrete example synthetic measurements of gravity variations g where taken at $m = 15$ equally spaced points along the line $0 \leq s \leq 1$. The $m \times n$ matrix $\bar{K}$ relates gravity variations at the $m = 15$ points along the surface to density variations f at $n = m = 15$ points at a depth $d = 0.25$ below the points of the surface measurements. The standard deviation of the measurement error ϵ is about 0.1.

First, when the problem is solved as $\bar{f} = \bar{K}^{-1}\bar{g}$, using the full SVD, the least squares estimate f_{SVD} oscillates on the scale of the discretization grid. From the model's point of view, the solution f_{SVD} does not seem to represent plausible density variations. The singular values of the matrix $\bar{K}$ are shown in Figure 2.2

We can see that the singular values are again approaching zero very rapidly, and that is the reason for the bad solution approximation.

Figures 2.3, 2.4, 2.5, 2.6 and 2.7 present the exact solution (solid line) and solution approximations (dashed line) obtained using TSVD with rank r equal to 4, 5, 6, 7 *and* 8. *As we can see, these estimates are reasonable estimates of the actual density variations. Figures 2.3 and 2.4 show that the best estimates are obtained for $r = 4$ and $r = 5$.*

2.2.11 Other Examples

Example 2.2.16 (Image processing). *Storing an image requires a large amount of computer memory, especially if high resolution is required. There is a lot of redundancy*

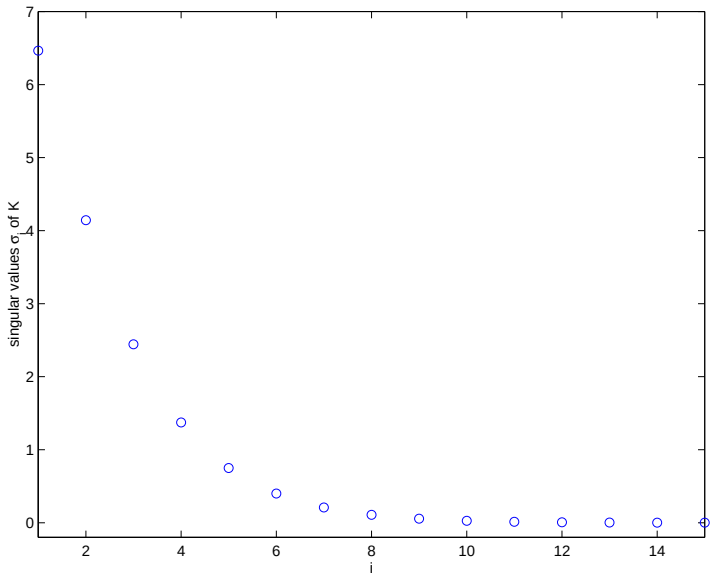


Figure 2.2: Singular value distribution of the discretized Fredholm integral operator.

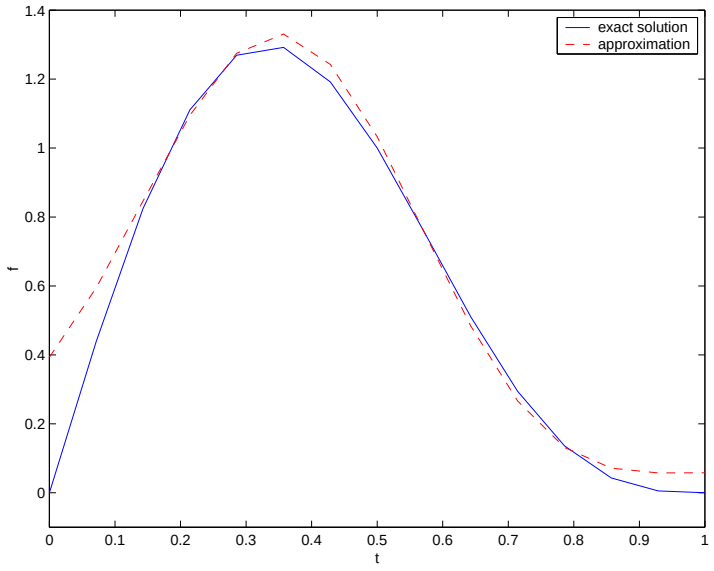


Figure 2.3: TSVD solution for $r = 4$.

among this data , and the SVD is used to reduce the amount of data and still to preserve important information. The idea is very simple: the image data are organized as an $m \times n$ *matrix* A*, and then its best rank* k *approximation is computed for suitably small* k*, as described in Theorem 2.1.11. Thus, instead of storing all* mn *elements of the matrix* A *we have to store only the elements of* $U(1:m, 1:k)$*,* $V(1:n, 1:k)$ *and* k *diagonal elements of* $\Sigma(1:k, 1:k)$*, where* $A = U\Sigma V^T$*. So, there is all together* $k(m+n+1)$ *elements to store, and this can be much less then* mn *if* $k \ll \min\{m, n\}$*.*

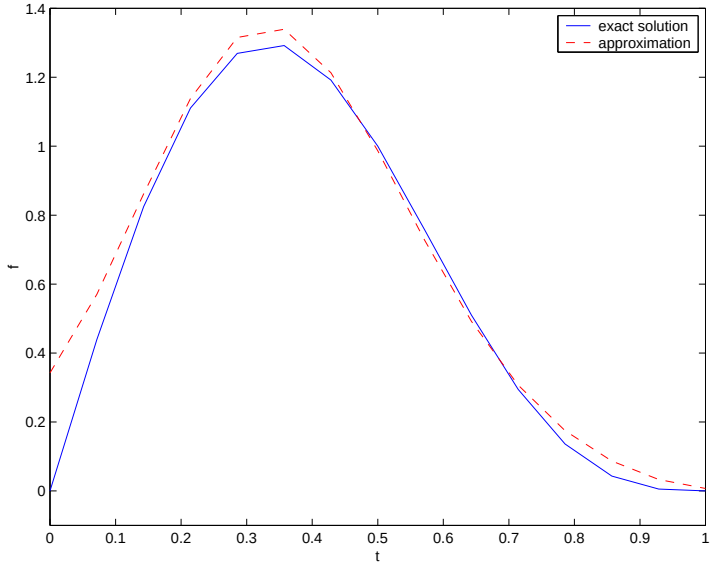


Figure 2.4: TSVD solution for $r = 5$.

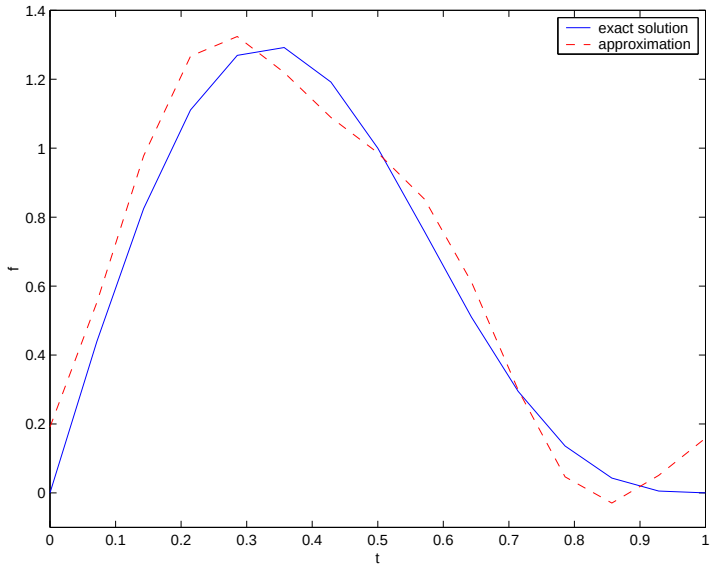


Figure 2.5: TSVD solution for $r = 6$.

The criteria for choosing the best k requires balancing storage reduction with good image quality.

For example, let us use the famous clown image which is an example from MATLAB *(Figure 2.8). We need a* 200×320 *matrix for storing this image, consisting of* 64000 *elements.*

If we use the rank 10 SVD approximation, that requires 5210 elements or about 8.14% of the original storage requirement. The quality of this new picture is not satisfactory

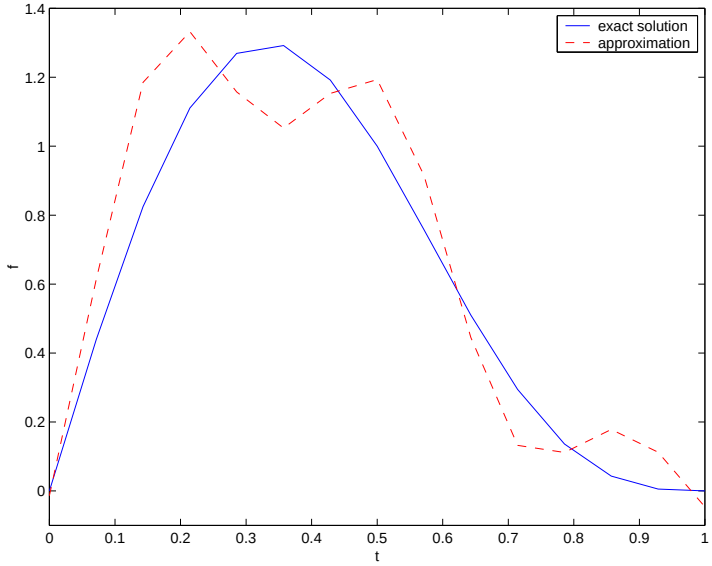


Figure 2.6: TSVD solution for $r = 7$.

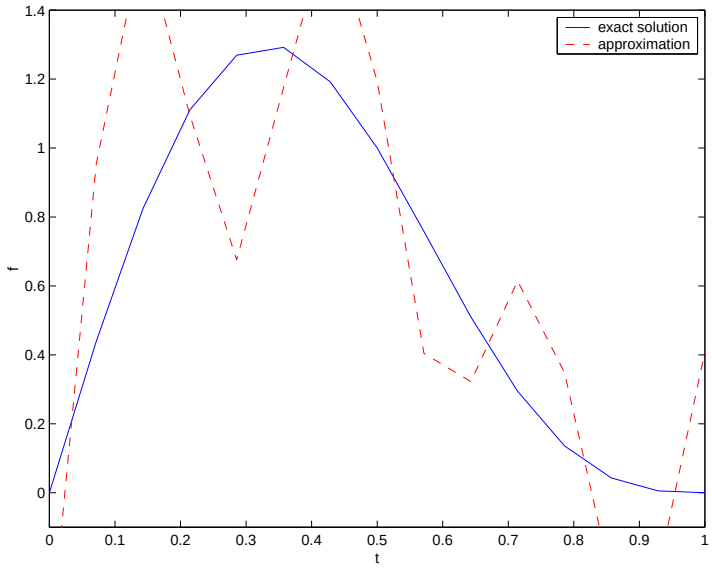


Figure 2.7: TSVD solution for $r = 8$.

(Figure 2.9).

The next image is produced by a rank 50 *approximation, with* 26050 *elements requiring* 40.70% *of the original storage amount. The quality of the image is now satisfactory (Figure 2.10). For more information on application of the SVD in image processing see [72] and [94].*

The same approach can be used for plotting surfaces. For example, suppose we want

Figure 2.8: Image of the clown: 100%

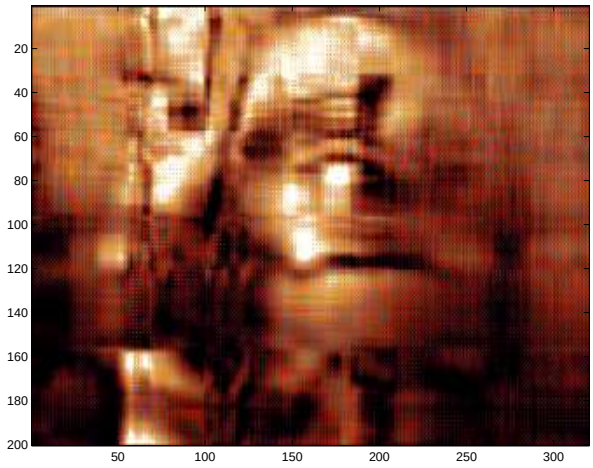


Figure 2.9: Image of the clown: 8.14%

to plot the graph of the following function (Figure 2.11)

$$f(x,y) = \frac{1}{250}(x^2y - x^2 - y^2 + 175) \qquad (x,y) \in [-5,5] \times [-5,5].$$

In order to do that, first we have to define a mesh on the square $[-5,5] \times [-5,5]$*, and then we have to plot points representing the function values in each mesh node. The points are then connected in an approximative surface. We can divide the initial square in small* 0.5×0.5 *squares, producing all together* $21 \times 21 = 441$ *mesh points, denoted by*

$$(x_i, y_i), \qquad i,j = 0,\ldots,20, \quad x_i = -5 + 0.5i,\ y_j = -5 + 0.5j.$$

These points are further organized in a 21×21 *matrix* $A = [a_{ij}]$*, where*

$$a_{ij} = f(x_i, y_j),$$

Figure 2.10: Image of the clown: 40.70%

which is then used for plotting. Thus, we have to store all 441 *elements.*

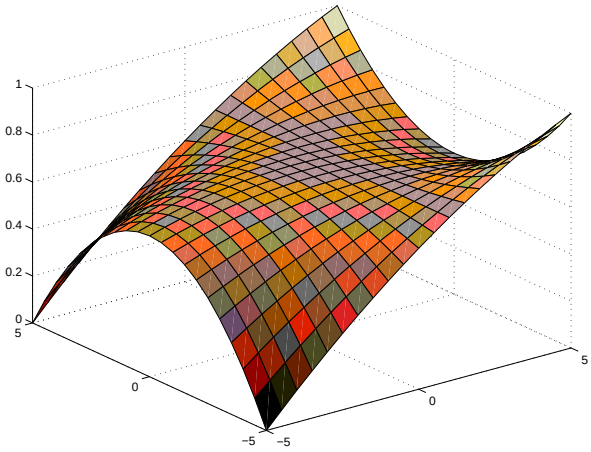


Figure 2.11: 3D plot of the function $f(x,y) = \frac{1}{250}(x^2y - x^2 - y^2 + 175)$: 100%

By computing the SVD of the matrix A, we can note that only the first two singular values of A are nontrivial. Hence, the matrix A has rank 2*, and the only possible choices for storage reduction are rank* 1 *and rank* 2 *approximations. The rank* 1 *approximation requires storage of only* 43 *elements which represents* 9.75% *of the original storage requirements. The resulting plot is not very accurate (Figure 2.12).*

The rank 2 *approximation requires storage of* 86 *elements which represents* 19.50% *of the original storage requirements. The produced plotting is exactly the same as the original (Figure 2.13).*

Example 2.2.17 (Internet traffic modelling). *The singular value decomposition is usually used to extract important information from an abundant amount of data. This is*

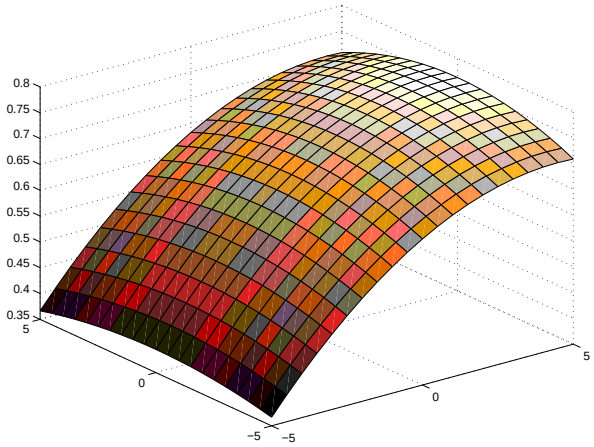


Figure 2.12: 3D plot of the function $f(x,y) = \frac{1}{250}(x^2y - x^2 - y^2 + 175)$: 9.75%

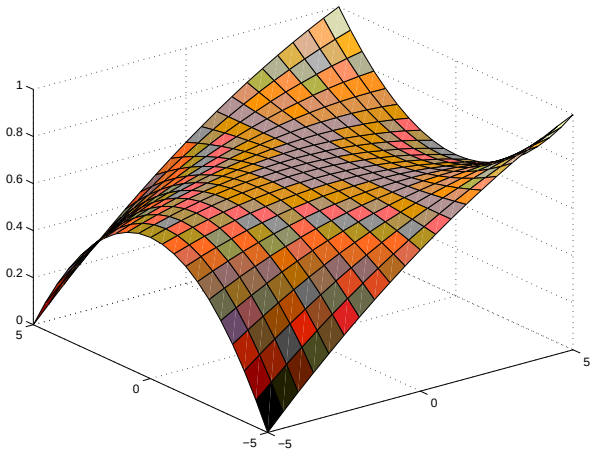


Figure 2.13: 3D plot of the function $f(x,y) = \frac{1}{250}(x^2y - x^2 - y^2 + 175)$: 19.50%

done in an example taken from [96], where internet traffic was analyzed. The main goal in [96] was to find patterns of internet traffic trace, such as weekly and daily patterns. The traffic intensity was observed, and its status was registered every minute in 49 *successive days. The collected data was organized in a* 49×1440 *matrix* X*, where each row represents one day, and each column is with respect to one minute within a day.*

From Figure 2.14 it is clear that there exist weekly patterns, and weekday–weekend and day–night effects. After performing the SVD on the data matrix $X = \sum_{i=1}^{49} \sigma_i u_i v_i^T$*, the following conclusions can be drawn:*

- *the* $\sigma_1 u_1 v_1^T$ *component contains the average information of the day and the specific minute in a day*
- *the* $\sigma_2 u_2 v_2^T$ *component contains the difference between weekend and weekdays and day–night effect*

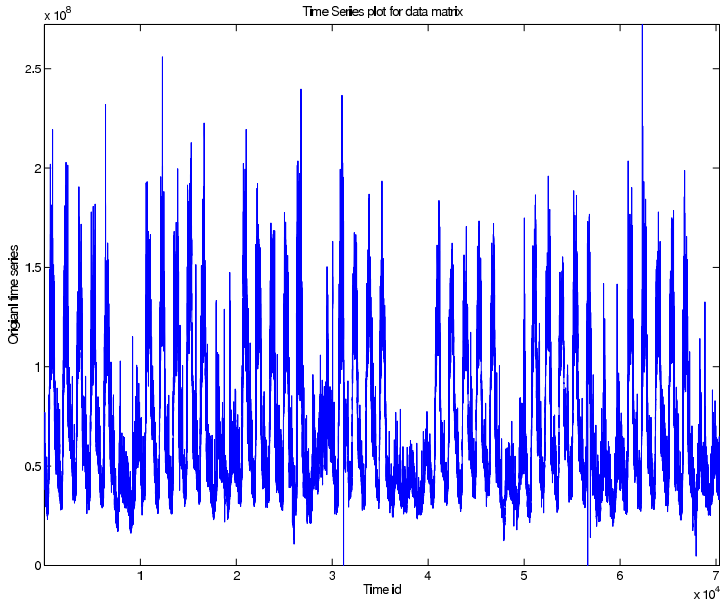


Figure 2.14: Original time series of internet traffic intensity.

- *the* $\sigma_3 u_3 v_3^T$ *component contains information related to some outliers*

The analysis of the singular vectors produces the following results:

- u_1 *has a weekly pattern and contains information related to the total network traffic*
- u_2 *also contains a clear weekly effect, and is strongly correlated with the variance of each day*
- u_3 *shows the special information the outliers might have*
- v_1 *shows the average daily traffic shape with the day–night effect*
- v_2 *also shows the day–night effect with the difference between weekdays and the weekend*
- v_3 *shows strong variability during the day, and this is probably because of the strong influence of the outliers*

In this example the SVD is used to get prediction with use of fewer components.

Example 2.2.18 (Genetics). *One of the most interesting problems in modern science is the decoding of genes. Recently, a huge advance in technology and data analysis has occurred, which enables us today a better understanding of the connection between genes and all the features of an organism. One possible approach to this problem is to analyze the data obtained from* microarray *experiments. The task is to find the structure of the gene network (or to reverse–engineer the network), which describes interactions between genes in a selected biological process and is illustrated in Figure 2.15. Solving this task*

requires a very large amount of experimental data, which is expensive to obtain. To overcome the problem of data shortage and computational inefficiency, several genetic researchers have adopted a linear model and have used the singular value decomposition to reconstruct the network architecture [95].

The method for such a reconstruction consists of two steps. The first step is the application of the SVD for constructing a set of feasible solutions that are consistent with the measured data. The second step is the robust regression which is used for selection of the most sparse one as the solution. The reason for doing the second step is that earlier works on gene regulatory networks and bioinformatics databases suggested that naturally occurring gene networks are sparse, i.e., generally each gene interacts with only a small percentage of all the genes in the entire genome.

In [95] they considered only systems that are operating near a steady state, so that the dynamics can be approximated by a linear system of ordinary differential equations:

$$\dot{x}_i(t) = -\lambda_i x_i(t) + \sum_{j=1}^{n} w_{ij} x_j(t) + b_i(t) + \xi_i(t), \quad i = 1, \ldots, n. \tag{2.29}$$

Here, x_i is the concentration of the i-th mRNA which reflects the expression level of the gene, λ_i is the self-degradation rate for the i-th gene, b_i is the external stimulus on the i-th gene, and ξ_i represents noise. The matrix element $w_{ij} \in \mathbb{R}$ describes the type and strength of the influence of the j-th gene on the i-th gene, with a positive sign indicating activation, a negative sign indicating repression, and a zero indicating no interaction.

To obtain parameters in the equation (2.29), the authors of [95] used a prescribed stimulus $[b_1, b_2, \ldots, b_n]^T$, and they used a microarray to simultaneously measure the concentrations of n different mRNAs, i.e., $[x_1, x_2, \ldots, x_n]^T$. Repeating this procedure m times, they obtained m measurements, organized as a matrix $X \in \mathbb{R}^{n\times m}$:

$$X = \begin{bmatrix} x_1^1 & x_1^2 & \cdots & x_1^m \\ x_2^1 & x_2^2 & \cdots & x_2^m \\ \vdots & \vdots & \ddots & \vdots \\ x_n^1 & x_n^2 & \cdots & x_n^m \end{bmatrix}.$$

The x_i^j represents the concentration of the i-th mRNA in the j-th experiment, with similar notations for $\dot{X}$ and B. The equation (2.29) can then be rewritten as

$$\dot{X} = AX + B, \qquad X, \dot{X}, B \in \mathbb{R}^{n\times m}, \quad A = [a_{ij}] \in \mathbb{R}^{n\times n}, \tag{2.30}$$

where noise is neglected and

$$a_{ij} = w_{ij} - \delta_{ij}\lambda_i.$$

The goal of the reverse engineering is to use measured data B, X, and $\dot{X}$ to deduce A and hence the connectivity matrix $W = [w_{ij}]$. First, by transposing equation (2.30), we obtain

$$(X^T)(A^T) = (\dot{X}^T) - (B^T), \tag{2.31}$$

where A is unknown. Because of the high costs of the measurements, typically $m \ll n$, thus (2.31) is an underdetermined linear system. The SVD is used to decompose X^T as

$$X^T = U\Sigma V^T, \qquad X^T, U \in \mathbb{R}^{m\times n}, \quad \Sigma, V \in \mathbb{R}^{n\times n},$$

in order to obtain $(X^T)^\dagger = (X^\dagger)^T$. *Let* $\mathrm{rank}(X) = r$. *Then one particular solution for* A *is given by*

$$A_0 = (\dot{X} - B)X^\dagger,$$

and the general solution is given by the affine space

$$A = A_0 + CV^T, \qquad C \in \mathbb{R}^{n\times n}, \tag{2.32}$$

where $C = [c_{ij}]$ *and* $c_{ij} = 0$ *for* $j \leq r$. *The family of solutions in equation (2.32) represents all the possible networks that are consistent with the microarray data.*

The second step of the procedure will find the most suitable solution A.

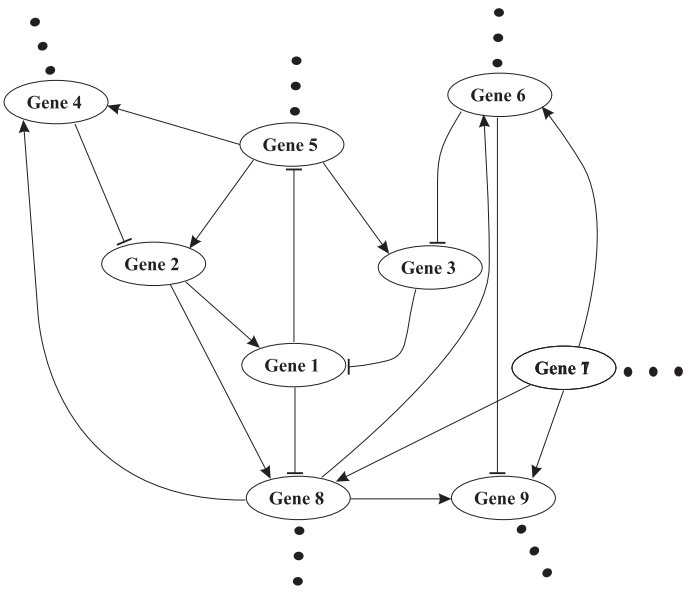
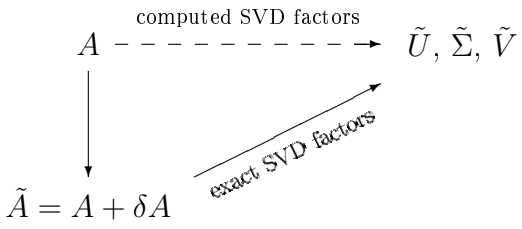


Figure 2.15: Schematic of a nonlinear gene network.

2.3 Perturbation Theory

When the singular value decomposition of a matrix $A \in \mathbb{R}^{m\times n}$ $(m \geq n)$ is computed in finite precision arithmetic, the exact factors $U \in \mathbb{R}^{m\times n}$, $\Sigma \in \mathbb{R}^{n\times n}$ and $V \in \mathbb{R}^{n\times n}$ will not be obtained in most cases. Matrices $\tilde{U}$, $\tilde{\Sigma}$ and $\tilde{V}$ will be computed instead. Numerical analysis of the method, used for computing the SVD, results in a matrix $\tilde{A}$ such that some or all of the computed matrices $\tilde{U}$, $\tilde{\Sigma}$ and $\tilde{V}$ are its exact SVD factors.



The relation between the exact and the computed factors is given by perturbation theory, which compares the matrices A and $\tilde{A}$. Basically, the perturbation theory will produce bounds on the errors in computed SVD factors. The error bounds can be divided in two different categories:

1. singular value error bounds,
2. singular subspace error bounds.

2.3.1 Singular Value Error Bounds

First, let us take a look at additive perturbations of a matrix.

Theorem 2.3.1 (Mirsky–Lidskii-Wielandt [66, p. 23]). *Let $A \in \mathbb{C}^{m\times n}$, $m \geq n$, then for any unitarily invariant norm $\|\cdot\|$, we have*

$$\|\Sigma - \tilde{\Sigma}\| \leq \|A - \tilde{A}\|.$$

Specially, for $\Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_n)$ and $\tilde{\Sigma} = \operatorname{diag}(\tilde{\sigma}_1, \ldots, \tilde{\sigma}_n)$, we have

$$\max_{i=1,\ldots,n} |\sigma_i - \tilde{\sigma}_i| \leq \|A - \tilde{A}\|_2,$$

$$\sqrt{\sum_{i=1}^{n} (\sigma_i - \tilde{\sigma}_i)^2} \leq \|A - \tilde{A}\|_F.$$

Theorem 2.3.1 claims that the absolute backward error norm is the upper bound for absolute error in singular values. But, the absolute errors are not always the best way of measuring errors. Backward analysis of a method which computes the SVD of the matrix A, usually results with a bound

$$\|\tilde{A} - A\|_2 \leq \eta_2 \|A\|_2, \quad \text{or} \quad \|\tilde{A} - A\|_F \leq \eta_F \|A\|_F,$$

where η_2 and η_F are some multiples of machine roundoff ε. Thus, if we want to look at the relative singular value error, we are going to obtain

$$\frac{|\sigma_i - \tilde{\sigma}_i|}{\sigma_i} \leq \eta_2 \frac{\sigma_1}{\sigma_i} \leq \eta_2 \kappa_2(A),$$

where $\kappa_2(A) = \|A\|_2 \|A^\dagger\|_2$ is the condition number. This means that if A is an ill conditioned matrix, the small singular values might be computed with large relative error.

The next step would be to look at the relative errors in singular values.

Theorem 2.3.2 ([52, p. 174]). *Let $A \in \mathbb{C}^{m\times n}$ for $m \geq n$. If* $\operatorname{range}(\tilde{A}) \subset \operatorname{range}(A)$ *then*

$$\frac{|\sigma_i - \tilde{\sigma}_i|}{\sigma_i} \leq \|A^\dagger(\tilde{A} - A)\|_2, \qquad i = 1, \ldots, n.$$

If $\operatorname{range}(\tilde{A}^*) \subset \operatorname{range}(A^*)$ *then*

$$\frac{|\sigma_i - \tilde{\sigma}_i|}{\sigma_i} \leq \|(\tilde{A} - A)A^\dagger\|_2, \qquad i = 1, \ldots, n.$$

When A has full column rank the second range condition in Theorem 2.3.2 is automatically satisfied.

Theorem 2.3.3 ([52, p. 174]). *Let $A \in \mathbb{C}^{m\times n}$, $m \geq n$, and let A have full column rank, then*

$$\max_{i=1,\ldots,n} \frac{|\sigma_i - \tilde{\sigma}_i|}{\sigma_i} \leq \|(\tilde{A} - A)A^{\dagger}\|_2.$$

Multiplicative perturbations are much more suitable for this case, so the next results will deal with such perturbations. Let us start with Ostrowsky–type bounds.

Theorem 2.3.4 ([52, p. 190]). *Let $A \in \mathbb{C}^{m\times n}$, $m \geq n$, and let $\tilde{A} = SAT$, where S and T are nonsingular. Then*

$$\frac{\sigma_i}{\|S^{-1}\|_2\|T^{-1}\|_2} \leq \tilde{\sigma}_i \leq \sigma_i\|S\|_2\|T\|_2.$$

Theorem 2.3.5 ([52, p. 192]). *Let $A \in \mathbb{C}^{m\times n}$, $m \geq n$, and let $\tilde{A} = SAT$, where S and T are nonsingular. Then*

$$\max_{i=1,\ldots,n} \frac{|\sigma_i - \tilde{\sigma}_i|}{\sigma_i} \leq \max\{\|I - SS^*\|_2, \|I - T^*T\|_2\}.$$

Thus, the relative error in singular values of $\tilde{A}$ is small if S and T are close to unitary matrices.

There are some other useful error measures for singular values. One of them is

$$\chi(\alpha, \tilde{\alpha}) = \frac{|\alpha - \tilde{\alpha}|}{\sqrt{|\alpha\tilde{\alpha}|}},$$

the so-called χ *relative distance* between real numbers α and $\tilde{\alpha}$. We define $\chi(0, 0) = 0$. The relation between this relative distance and the standard relative error is given in the following proposition.

Proposition 2.3.6 ([66, pp. 15–16]). *Let $\alpha, \tilde{\alpha} \in \mathbb{R}$. If $0 \leq \epsilon < 1$, then*

$$\frac{|\alpha - \tilde{\alpha}|}{|\alpha|} \leq \epsilon \Rightarrow \chi(\alpha, \tilde{\alpha}) \leq \frac{\epsilon}{\sqrt{1-\epsilon}}, \tag{2.33}$$

if $0 \leq \epsilon < 2$, then

$$\chi(\alpha, \tilde{\alpha}) \leq \epsilon \Rightarrow \max\left\{\frac{|\alpha - \tilde{\alpha}|}{|\alpha|}, \frac{|\alpha - \tilde{\alpha}|}{|\tilde{\alpha}|}\right\} \leq \left(\frac{\epsilon}{2} + \sqrt{1 + \frac{\epsilon^2}{4}}\right)\epsilon. \tag{2.34}$$

Asymptotically,

$$\lim_{\tilde{\alpha}\to\alpha} \frac{\chi(\alpha, \tilde{\alpha})}{\frac{|\alpha-\tilde{\alpha}|}{|\alpha|}} = 1,$$

thus (2.33) and (2.34) are at least asymptotically sharp.

Theorem 2.3.7 ([64, p. 397]). *Let $A \in \mathbb{C}^{m\times n}$, $m \geq n$, and suppose that $\tilde{A} = SAT$ where S and T are nonsingular. Then,*

$$\begin{aligned} \max_{i=1,\ldots,n} \chi(\sigma_i, \tilde{\sigma}_i) &\leq \frac{1}{2}\|S^* - S^{-1}\|_2 + \frac{1}{2}\|T^* - T^{-1}\|_2, \\ \sqrt{\sum_{i=1}^{n} \chi^2(\sigma_i, \tilde{\sigma}_i)} &\leq \frac{1}{2}\|S^* - S^{-1}\|_F + \frac{1}{2}\|T^* - T^{-1}\|_F. \end{aligned}$$

If T is unitary then

$$\begin{aligned} \max_{i=1,\ldots,n} \chi(\sigma_i, \tilde{\sigma}_i) &\leq \||S|^{1/2} - |S|^{-1/2}\|_2, \\ \sqrt{\sum_{i=1}^{n} \chi^2(\sigma_i, \tilde{\sigma}_i)} &\leq \||S|^{1/2} - |S|^{-1/2}\|_F, \end{aligned}$$

where

$$|S| = (S^*S)^{1/2}.$$

2.3.2 Singular Subspace Error Bounds

When comparing two subspaces $\mathcal{X}$ and $\mathcal{Y}$ of dimension k, the most natural measure is the angle matrix $\Theta(X, Y)$ between these two subspaces, where X and Y are orthonormal bases for $\mathcal{X}$ and $\mathcal{Y}$, and

$$\Theta(X, Y) = \arccos(X^T Y Y^T X)^{1/2}, \quad \text{with} \quad \|\sin\Theta(X, Y)\|_2 = \|Y_\perp^T X\|_2.$$

$Y_\perp$ is orthonormal basis of $\mathcal{Y}^\perp$, and $\mathcal{Y}^\perp$ is orthogonal complement of $\mathcal{Y}$. From section 2.2 and [90]

$$\|\sin\Theta(X, Y)\|_2 = \sin(\theta_k) = \sin\angle(\mathcal{X}, \mathcal{Y}), \qquad \|\sin\Theta(X, Y)\|_F = \sqrt{\sum_{i=1}^{k} \sin^2(\theta_i)}.$$

Theorem 2.3.8 (Wedin [67, p. 5]). *Let $m \geq n$, and let $A \in \mathbb{C}^{m\times n}$ and $\tilde{A} \in \mathbb{C}^{m\times n}$ have the following SVDs*

$$A = U\Sigma V^* = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^* \\ V_2^* \end{bmatrix}, \tag{2.35}$$

$$\tilde{A} = \tilde{U}\tilde{\Sigma}\tilde{V}^* = \begin{bmatrix} \tilde{U}_1 & \tilde{U}_2 \end{bmatrix} \begin{bmatrix} \tilde{\Sigma}_1 & 0 \\ 0 & \tilde{\Sigma}_2 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{V}_1^* \\ \tilde{V}_2^* \end{bmatrix}, \tag{2.36}$$

where $U, \tilde{U} \in \mathbb{C}^{m\times m}$ are unitary, $V, \tilde{V} \in \mathbb{C}^{n\times n}$ are unitary, $U_1, \tilde{U}_1 \in \mathbb{C}^{m\times k}$, $V_1, \tilde{V}_1 \in \mathbb{C}^{n\times k}$, and

$$\Sigma_1 = \mathrm{diag}(\sigma_1, \ldots, \sigma_k), \qquad \Sigma_2 = \mathrm{diag}(\sigma_{k+1}, \ldots, \sigma_n), \tag{2.37}$$

$$\tilde{\Sigma}_1 = \mathrm{diag}(\tilde{\sigma}_1, \ldots, \tilde{\sigma}_k), \qquad \tilde{\Sigma}_2 = \mathrm{diag}(\tilde{\sigma}_{k+1}, \ldots, \tilde{\sigma}_n), \tag{2.38}$$

with $1 \leq k < n$. *Let us define the residuals*

$$R_R = \tilde{A}V_1 - U_1\Sigma_1 = (\tilde{A} - A)V_1 \quad \textit{and} \quad R_L = \tilde{A}^*U_1 - V_1\Sigma_1 = (\tilde{A}^* - A^*)U_1.$$

If

$$\delta = \min\left\{ \min_{i=1,\ldots,k,\ j=1,\ldots,n-k} |\sigma_i - \tilde{\sigma}_{k+j}|, \min_{i=1,\ldots,k} \sigma_i \right\} > 0,$$

then

$$\sqrt{\|\sin\Theta(U_1, \tilde{U}_1)\|_F^2 + \|\sin\Theta(V_1, \tilde{V}_1)\|_F^2} \leq \frac{\sqrt{\|R_R\|_F^2 + \|R_L\|_F^2}}{\delta}.$$

The scalar δ represents the absolute gap between singular values of Σ_1 and $\tilde{\Sigma}_2$. The next theorem involves the relative gap, but first we have to define one more error measure for singular values:

$$\rho_p(\alpha, \tilde{\alpha}) = \frac{|\alpha - \tilde{\alpha}|}{\sqrt[p]{|\alpha|^p + |\tilde{\alpha}|^p}}, \quad \text{for } 1 \leq p \leq \infty,$$

and ρ_p is the so-called p *relative distance* between real numbers α and $\tilde{\alpha}$. We define $\rho_p(0, 0) = 0$. The relation between this relative distance and the standard relative error is given in the following proposition.

Proposition 2.3.9 ([66, pp. 10–11]). *Let* $\alpha, \tilde{\alpha} \in \mathbb{R}$. *If* $0 \leq \epsilon < 1$, *then*

$$\frac{|\alpha - \tilde{\alpha}|}{|\alpha|} \leq \epsilon \Rightarrow \rho_p(\alpha, \tilde{\alpha}) \leq \frac{\epsilon}{\sqrt[p]{1 + (1 - \epsilon)^p}}, \tag{2.39}$$

and

$$\rho_p(\alpha, \tilde{\alpha}) \leq \epsilon \Rightarrow \max\left\{ \frac{|\alpha - \tilde{\alpha}|}{|\alpha|}, \frac{|\alpha - \tilde{\alpha}|}{|\tilde{\alpha}|} \right\} \leq \frac{2^{1/p}\epsilon}{1 - \epsilon}. \tag{2.40}$$

Asymptotically,

$$\lim_{\tilde{\alpha} \to \alpha} \frac{\rho_p(\alpha, \tilde{\alpha})}{\frac{|\alpha - \tilde{\alpha}|}{|\alpha|}} = 2^{1/p},$$

thus (2.39) and (2.40) are at least asymptotically sharp.

Theorem 2.3.10 ([67, p. 7]). *Let* $A \in \mathbb{C}^{m \times n}$ $(m \geq n)$, *and* $\tilde{A} = S^*AT$ *be two matrices with SVDs (2.35), (2.36), (2.37) and (2.38), where* S *and* T *are nonsingular. Let*

$$\eta_2 = \begin{cases} \min\left\{ \min_{i=1,\ldots,k,\ j=1,\ldots,n-k} \rho_2(\sigma_i, \tilde{\sigma}_{k+j}), \min_{i=1,\ldots,k} \rho_2(\sigma_i, 0) \right\}, & \textit{if } m > n, \\ \min_{i=1,\ldots,k,\ j=1,\ldots,n-k} \rho_2(\sigma_i, \tilde{\sigma}_{k+j}), & \textit{otherwise.} \end{cases}$$

If $\eta_2 > 0$, *then*

$$\sqrt{\|\sin\Theta(U_1, \tilde{U}_1)\|_F^2 + \|\sin\Theta(V_1, \tilde{V}_1)\|_F^2} \leq$$
$$\leq \frac{\sqrt{\|(I - S^*)U_1\|_F^2 + \|(I - S^{-1})U_1\|_F^2 + \|(I - T^*)V_1\|_F^2 + \|(I - T^{-1})V_1\|_F^2}}{\eta_2}.$$

2.4 Methods for Computing the SVD

2.4.1 Jacobi–Type Methods

Jacobi–type methods are iterative methods for computing the singular value decomposition of a matrix $A = [a_{ij}] \in \mathbb{R}^{m\times n}$. They are based on pre- and post-multiplication with Jacobi rotations. The important measure for the convergence of such methods is

$$\mathrm{off}(A) = \sum_{i\neq j} a_{ij}^2.$$

The one–sided Jacobi SVD algorithm

The Jacobi method for spectral decomposition was introduced by Jacobi in 1846 [53], and many variants of the method have been developed since. The one–sided Jacobi SVD algorithm applies multiplication with Jacobi rotations only to one side of the matrix $A \in \mathbb{R}^{m\times n}$, and it is described in [16]. If, for example, $m \gg n$, a QR factorization of the matrix A is performed in order to reduce the matrix dimension. The Jacobi algorithm is then applied to the $n \times n$ upper triangular factor or to its transpose. The algorithm generates a sequence of matrices $A^{(k)}$ as follows,

$$\begin{array}{rclcrcl} V^{(0)} & = & I_n & & A^{(0)} & = & A \\ V^{(k+1)} & = & V_k V^{(k)} & & A^{(k+1)} & = & A^{(k)} V_k^T \end{array} \tag{2.41}$$

where V_k is a plane rotation, acting only on the i_k-th and the j_k-th columns. The rotation acting on the i-th and the j-th rows or columns is defined by

$$G_{i,j}(\theta) = \begin{bmatrix} 1 & & & \vdots & & & & \vdots & & & & \\ & \ddots & & \vdots & & & & \vdots & & 0 & & \\ & & 1 & \vdots & & & & \vdots & & & & \\ \cdots & \cdots & \cdots & \cos(\theta) & \cdots & \cdots & \cdots & \sin(\theta) & \cdots & \cdots & \cdots & i \\ & & & \vdots & 1 & & & \vdots & & & & \\ & & & \vdots & & \ddots & \vdots & & & & & \\ & & & \vdots & & & 1 & \vdots & & & & \\ \cdots & \cdots & \cdots & -\sin(\theta) & \cdots & \cdots & \cdots & \cos(\theta) & \cdots & \cdots & \cdots & j \\ & & & \vdots & & & & \vdots & 1 & & & \\ & 0 & & \vdots & & & & \vdots & & \ddots & & \\ & & & \vdots & & & & \vdots & & & 1 & \\ & & & i & & & & j & & & & \end{bmatrix}$$

V_k is defined as

$$V_k = G_{i_k,j_k}(\psi_k),$$

and if $A^{(k)} = [a_1^{(k)}, \ldots, a_n^{(k)}]$ is a column partition of the matrix $A^{(k)}$, then the iteration of the Jacobi algorithm (2.41) can be described as

$$\begin{bmatrix} a_{i_k}^{(k+1)} & a_{j_k}^{(k+1)} \end{bmatrix} = \begin{bmatrix} a_{i_k}^{(k)} & a_{j_k}^{(k)} \end{bmatrix} \cdot \begin{bmatrix} \cos(\psi_k) & -\sin(\psi_k) \\ \sin(\psi_k) & \cos(\psi_k) \end{bmatrix}.$$

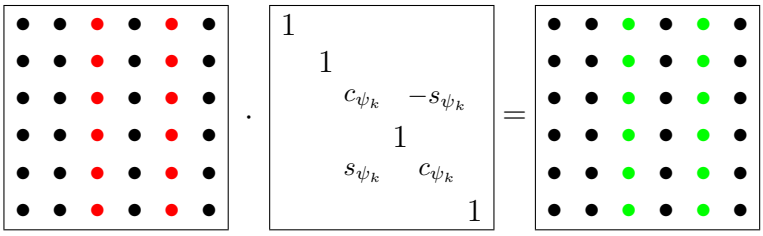
The angle ψ_k is chosen so that

$$(a_{i_k}^{(k+1)})^T a_{j_k}^{(k+1)} = 0,$$

which ensures that

$$\mathrm{off}((A^{(k+1)})^T A^{(k+1)}) = \mathrm{off}((A^{(k)})^T A^{(k)}) - 2|(a_{i_k}^{(k)})^T a_{j_k}|^2, \quad k \geq 0.$$

One iteration of the one–sided Jacobi SVD algorithm can be visualized as follows.



where $c_{\psi_k} = \cos(\psi_k)$, $s_{\psi_k} = \sin(\psi_k)$, • represent elements of the matrix $A^{(k)}$ that will be affected by V_k, and • are elements of $A^{(k+1)}$ that are changed, and are different from the corresponding elements of $A^{(k)}$. The two columns denoted by green bullets will be orthogonal, due to the choice of angle ψ_k.

The pivot indices i_k and j_k are usually chosen according to row– or column–cyclic pivot strategy, or so that $|(a_{i_k}^{(k)})^T a_{j_k}^{(k)}|$ is maximal. Actually, the Jacobi SVD algorithm is equivalent to the symmetric Jacobi algorithm applied to the matrix $A^T A$. Thus, the sequence $A^{(k)}$ converges to a diagonal matrix

$$\lim_{k\to\infty} (A^{(k)})^T A^{(k)} = \Sigma^2 = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_n^2),$$

and with

$$V = \lim_{k\to\infty} (V^{(k)})^T,$$

the following factorization is computed

$$A^T A = V \Sigma^2 V^T,$$

where

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

has orthogonal columns. This implies that U can be computed as

$$U = A^{(\infty)} \Sigma^{-1} \qquad \text{if } \det \Sigma \neq 0.$$

The one-sided Jacobi algorithm computes singular values with high relative accuracy, as it is shown in [16]. Let $\tilde{A}_s^{(k)}$ be the matrix with unit columns, obtained as

$$\tilde{A}_s^{(k)} = \tilde{A}^{(k)} (\tilde{D}^{(k)})^{-1}, \quad \text{with} \quad \tilde{D}^{(k)} = \mathrm{diag}(\|\tilde{A}^{(k)}(:,1)\|_2, \ldots, \|\tilde{A}^{(k)}(:,n)\|_2),$$

where $\tilde{A}^{(k)}$ is a matrix computed in finite precision arithmetic in the k-th step of the Jacobi algorithm, when the stopping criterion has been satisfied. If the computed singular values of $\tilde{A}^{(k)}$ are denoted by $\tilde{\sigma}_i^{(k)}$, $i = 1, \ldots, n$, then the following relation hold:

$$\frac{|\sigma_i - \tilde{\sigma}_i^{(k)}|}{\sigma_i} \leq \eta \max_{j=1,\ldots,k} \kappa_2(\tilde{A}_s^{(j)}), \quad i = 1, \ldots, n,$$

where η is some multiple of the unit roundoff ε.

New fast and accurate Jacobi SVD algorithm has been proposed recently in [24] and [25].

The Kogbetliantz algorithm

The Kogbetliantz algorithm is a two–sided Jacobi SVD algorithm for square matrices, and it was introduced in [60] and [61]. In [43] a modification of the original algorithm was proposed. The Kogbetliantz algorithm was applied to the triangular matrix, after a QR factorization with column pivoting has been applied to the original matrix. So we will assume that $A \in \mathbb{R}^{n\times n}$ is upper triangular, and that

$$|a_{11}| \geq |a_{22}| \geq \cdots \geq |a_{nn}|.$$

The algorithm generates a sequence of matrices $A^{(k)}$ as follows,

$$\begin{array}{lll} U^{(0)} = I_m & V^{(0)} = I_n & A^{(0)} = A \\ U^{(k+1)} = U_k U^{(k)} & V^{(k+1)} = V_k V^{(k)} & A^{(k+1)} = U_k A^{(k)} V_k^T \end{array} \tag{2.42}$$

where U_k and V_k are plane rotations, acting only on the i_k-th and the j_k-th rows and columns. Let us define

$$U_k = G_{i_k,j_k}(\phi_k), \qquad V_k = G_{i_k,j_k}(\psi_k),$$

and let us focus only on the rows and the columns, where all the action is going on. Then the iteration of the Kogbetliantz algorithm (2.42) can be described as

$$\begin{bmatrix} a_{i_k i_k}^{(k+1)} & a_{i_k j_k}^{(k+1)} \\ a_{j_k i_k}^{(k+1)} & a_{j_k j_k}^{(k+1)} \end{bmatrix} = \begin{bmatrix} \cos(\phi_k) & \sin(\phi_k) \\ -\sin(\phi_k) & \cos(\phi_k) \end{bmatrix} \cdot \begin{bmatrix} a_{i_k i_k}^{(k)} & a_{i_k j_k}^{(k)} \\ a_{j_k i_k}^{(k)} & a_{j_k j_k}^{(k)} \end{bmatrix} \cdot \begin{bmatrix} \cos(\psi_k) & -\sin(\psi_k) \\ \sin(\psi_k) & \cos(\psi_k) \end{bmatrix}.$$

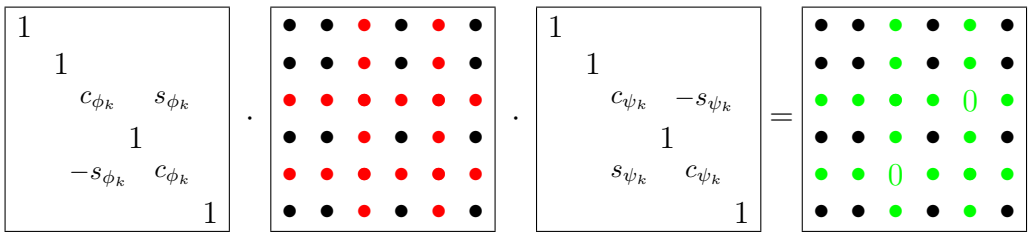
The angles ϕ_k and ψ_k are chosen so that

$$a_{i_k j_k}^{(k+1)} = a_{j_k i_k}^{(k+1)} = 0,$$

which ensures that

$$\mathrm{off}(A^{(k+1)}) = \mathrm{off}(A^{(k)}) - |a_{i_k j_k}^{(k)}|^2, \quad k \geq 0.$$

One iteration of the Kogbetliantz algorithm can be visualized as follows.



where $c_{\phi_k} = \cos(\phi_k)$, $s_{\phi_k} = \sin(\phi_k)$, $c_{\psi_k} = \cos(\psi_k)$, $s_{\psi_k} = \sin(\psi_k)$, • represents elements of the matrix $A^{(k)}$ that will be affected by U_k and V_k, and • are elements of $A^{(k+1)}$ that are changed, and are different from the corresponding elements of $A^{(k)}$.

The pivot indices i_k and j_k are usually chosen according to row– or column–cyclic pivot strategy. The sequence $A^{(k)}$ converges to a diagonal matrix

$$\lim_{k\to\infty} A^{(k)} = \Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_n),$$

and with

$$U = \lim_{k\to\infty} (U^{(k)})^T, \qquad V = \lim_{k\to\infty} (V^{(k)})^T,$$

all the SVD factors are computed, and

$$A = U\Sigma V^T.$$

2.4.2 Methods Based on Bidiagonalization

An algorithm belonging to this group consists of two steps. The first step uses orthogonal transformations to reduce the matrix $A \in \mathbb{R}^{m\times n}$, for $m \geq n$, to a bidiagonal form:

$$A = UBV^T, \qquad U \in \mathbb{R}^{m\times n}, \text{ and } B, V \in \mathbb{R}^{n\times n},$$

where U is orthonormal, V is orthogonal and B is bidiagonal

$$B = \begin{bmatrix} \psi_1 & \phi_2 & & & \\ & \psi_2 & \phi_3 & & \\ & & \ddots & \ddots & \\ & & & \psi_{n-1} & \phi_n \\ & & & & \psi_n \end{bmatrix}. \tag{2.43}$$

This process is called *bidiagonalization*. The second step is the application of a fast algorithm for computing the singular value decomposition of a bidiagonal matrix (see subsection 2.4.3).

Several bidiagonalization algorithms appeared in the past, and will be listed here. A new version of one–sided bidiagonalization will be presented in Chapter 3.

The Householder bidiagonalization

The Householder bidiagonalization algorithm is based on pre– and post–multiplications with Householder reflectors, such that at the end of the process the following relation is achieved

$$\begin{bmatrix} B \\ 0 \end{bmatrix} = U_n \cdots U_1 A V_1 \cdots V_{n-2}, \quad U = U_1 \cdots U_n, \; V = V_1 \cdots V_{n-2},$$

where U_k and V_k are the Householder reflectors, and $0 \in \mathbb{R}^{(m-n)\times n}$. This bidiagonalization is described by Golub and Kahan in [33]. A Householder reflector $H \in \mathbb{R}^{n\times n}$ is defined as

$$H = I_n - vv^T, \qquad \|v\|_2 = \sqrt{2},$$

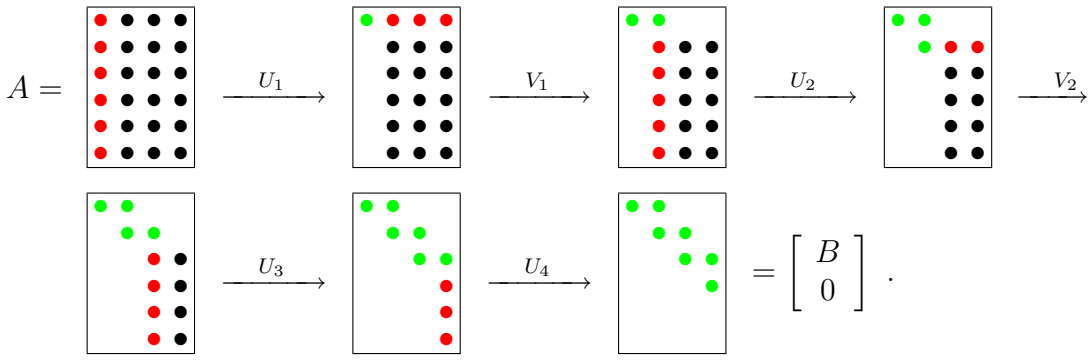
thus

$$H^T = H, \qquad H^2 = I_n.$$

For any $x, y \in \mathbb{R}^n$, $x \neq y$ with $\|x\|_2 = \|y\|_2$, a Householder reflector H can be found, such that $Hx = y$. In that case, the vector v is of the form

$$v = \frac{\sqrt{2}}{\|x - y\|_2}(x - y).$$

In the bidiagonalization process, Householder reflectors U_k are chosen to annihilate elements of the matrix A bellow the main diagonal, and V_k are chosen to annihilate elements above the superdiagonal. The computation and the application of the Householder reflectors U_k and V_k are interlaced: in the k-th step U_k will annihilate all the elements bellow the main diagonal in the k-th column, and V_k will annihilate all the elements right to the superdiagonal in the k-th row. The process is shown bellow.



The elements denoted by • are crucial for the next step of the algorithm. The column or row of red bullets denotes the vector x, such that the next Householder reflector H (lower block diagonal part of U_k or V_k) will depend on it. The Householder reflector H will be chosen so that $Hx = \pm\|x\|_2 e_1$. The elements denoted by • are computed values after the application of the Householder reflector.

The Lawson–Hanson–Chan bidiagonalization

When $m \gg n$, an efficient algorithm can be obtained if the QR factorization is performed before bidiagonalization. The bidiagonalization is then applied to the triangular square matrix with much smaller dimensions than the original matrix. This idea was mentioned in [63, p. 119] and analyzed in [11]. Let

$$A = Q \begin{bmatrix} R \\ 0 \end{bmatrix}, \qquad Q \in \mathbb{R}^{m\times m},\ R \in \mathbb{R}^{n\times n}$$

be the QR factorization of the matrix $A \in \mathbb{R}^{m\times n}$, with Q orthogonal and R upper triangular. Then, the bidiagonalization of the matrix R will produce

$$R = U_R B V^T, \qquad U_R,\ B,\ V \in \mathbb{R}^{n\times n},$$

with U_R and V orthogonal, and B bidiagonal. The final bidiagonalization of the matrix A is obtained as

$$A = Q \begin{bmatrix} U_R & 0 \\ 0 & I_{m-n} \end{bmatrix} \begin{bmatrix} B \\ 0 \end{bmatrix} V^T,$$

and

$$U = Q \begin{bmatrix} U_R & 0 \\ 0 & I_{m-n} \end{bmatrix}.$$

This algorithm involves fewer operations than the Householder bidiagonalization whenever $m \geq 5n/3$, and if U is not accumulated.

Both methods, the Householder bidiagonalization and the Lawson–Hanson–Chan bidiagonalization, can guarantee only small absolute errors in the computed singular values. If these algorithms are performed in finite precision arithmetic, and if the singular values of the computed bidiagonal matrix $\tilde{B}$ are denoted by $\tilde{\sigma}_i$, $i = 1, \ldots, n$, then they are the exact singular values of the matrix $A + \delta A$, and the following relations hold:

$$\begin{aligned} \|\delta A\|_2 &\leq \eta \|A\|_2, \quad \text{and thus} \\ \max_i |\sigma_i - \tilde{\sigma}_i| &\leq \eta \|A\|_2, \end{aligned}$$

where η is a moderate polynomial of matrix dimensions times the unit roundoff ε.

The Lanczos bidiagonalization

This algorithm, introduced by Golub and Kahan in [33], is a generalization of the symmetric Lanczos algorithm. It is based on a simple recurrence. Let b be a starting vector, then we define u_0, v_1 and ϕ_1 by

$$u_0 = 0, \qquad \phi_1 v_1 = b,\ \phi_1 = \|b\|_2,$$

and for $k = 1, 2, \ldots$ compute

$$\begin{aligned} \psi_k u_k &= A v_k - \phi_k u_{k-1}, \\ \phi_{k+1} v_{k+1} &= A^T u_k - \psi_k v_k, \end{aligned}$$

where ϕ_k and ψ_k are nonnegative, and are chosen so that $\|u_k\|_2 = \|v_k\|_2 = 1$. If we define

$$U_k = [u_1, \ldots, u_k], \quad V_k = [v_1, \ldots, v_k],$$

then the above equations can be written as

$$\begin{aligned} \phi_1 V_k e_1 &= b, \\ AV_k &= U_k B_k, \\ A^T U &= V_{k+1} B_{k+1,k}^T, \end{aligned}$$

where $B_k \in \mathbb{R}^{k\times k}$ is upper bidiagonal

$$B_k = \begin{bmatrix} \psi_1 & \phi_2 & & & \\ & \psi_2 & \phi_3 & & \\ & & \ddots & \ddots & \\ & & & \psi_{k-1} & \phi_k \\ & & & & \psi_k \end{bmatrix},$$

and $B_{k,k+1}$ is equal to B_{k+1} without the last row. Paige showed in [77] that the matrices U_k and V_k are orthogonal, and because of this orthogonality the process will stop at $k \leq \min(m, n)$ with either $\phi_k = 0$ or $\psi_k = 0$. The Lanczos bidiagonalization implemented in finite precision arithmetic can produce matrices $\tilde{U}_k$ and $\tilde{V}_k$ which are not numerically orthogonal. Thus, some sort of reorthogonalization is required.

The Ralha one–sided bidiagonalization

This algorithm is proposed by Ralha in [80], [82] and [81], and its main characteristic is that the Householder reflectors are applied only from one side of the matrix A. More attention will be devoted to this algorithm here, because the new bidiagonal algorithm, which will be analyzed in this thesis, is its modification. The main steps of the algorithm are as follows:

- **Triorthogonalization**
  The matrix is post-multiplied by a sequence of $n-2$ Householder reflectors $\mathbf{V}_k$:

  $$A_0 = A, \; A_k = A_{k-1}\mathbf{V}_k, \quad k = 1, \ldots, n-2.$$

  The Householder reflectors $\mathbf{V}_k$ are chosen so that

  $$\text{for } V = \mathbf{V}_1 \cdots \mathbf{V}_{n-2} \qquad F = A_{n-2} = AV \quad \text{is triorthogonal,}$$

  which means, that for columns f_i, f_j of F

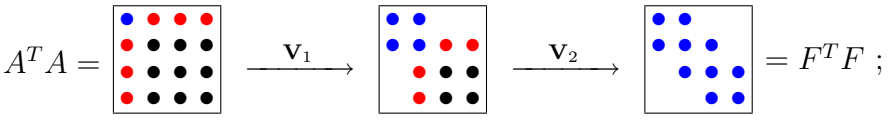
  $$f_i^T f_j = 0, \quad |i-j| > 1.$$

  This is equivalent way to say that $F^T F$ is tridiagonal.

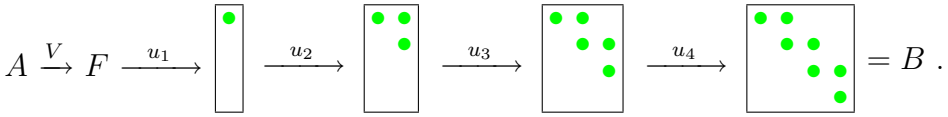
- **A variant of the Gram–Schmidt orthogonalization**
  The columns of F are orthogonalized only against adjacent columns, producing
  $$F = UB,$$
  where $U = [u_1, \ldots, u_n] \in \mathbb{R}^{m\times n}$ is orthogonal and $B \in \mathbb{R}^{n\times n}$ is the required upper bidiagonal matrix, whose singular values are those of A.

The steps of the Ralha algorithm are shown bellow.

Implicitly:



Explicitly:



The elements denoted by • will be used in the next step of the algorithm. The column or row of red bullets denotes the vector z_k, such that the next Householder reflector V_k will depend on it. The Householder reflector V_k will be chosen so that $V_k z_k = \pm\|z_k\|_2 e_1$, and $\mathbf{V}_k = \begin{bmatrix} I_k & 0 \\ 0 & V_k \end{bmatrix}$. The elements denoted by • are computed values of F^TF after the application of the Householder reflector. The computed elements of B are denoted by •.

Finally, the complete algorithm is given in Algorithm 2.4.1.

Algorithm 2.4.1 (The Ralha one–sided bidiagonalization). *For $A \in \mathbb{R}^{m\times n}$, $\mathrm{rank}(A) = n > 2$, this algorithm computes orthonormal $U = [u_1, \ldots, u_n]$, bidiagonal B and orthogonal $V = V^{(n-2)}$ such that $A = UBV^T$.*

$A_0 = A$; $V^{(0)} = I$;
{*Implicit triorthogonalization*}
for $k = 1 : n-2$
  $z_k = A_{k-1}(:, k+1:n)^T A_{k-1}(:, k)$;
  if $z_k \neq 0$ *find a Householder transformation V_k such that $V_k z_k = \gamma_k e_1$;*
    $A_k(:, k+1:n) = A_{k-1}(:, k+1:n)V_k$;
    $\mathbf{V}_k = \begin{bmatrix} I_k & 0 \\ 0 & V_k \end{bmatrix}$; $V^{(k)} = V^{(k-1)}\mathbf{V}_k$;
  else
    $V_k = I_{n-k}$;
  end
end

{*The Gram–Schmidt orthogonalization*}
$F = [f_1, \ldots, f_n] = A_{n-2}$;
$s_1 = f_1$;
$\psi_1 = \|s_1\|_2$;
$u_1 = s_1/\psi_1$;
for $k = 2 : n$
  $\phi_k = u_{k-1}^T f_k$;
  $s_k = f_k - \phi_k u_{k-1}$;
  $\psi_k = \|s_k\|_2$;
  $u_k = s_k/\psi_k$;
end

The version of the Gram–Schmidt orthogonalization in this algorithm substitutes pre-multiplication with a sequence of Householder reflection in the Householder bidiagonalization. Thus, the second part of the algorithm requires only $O(mn)$ flops instead of $O(n^2 m)$. The other important characteristic is that the algorithm is one-sided. That means that most of the algorithm can be expressed by simple operations on columns of the transformed matrix, and it can be efficiently implemented on **multiprocessor systems with distributed memory**. On the other hand, when the algorithm is implemented in finite precision arithmetic, there is a possible loss of triorthogonality of the computed matrix $\tilde{F}$. This means that $\tilde{F}^T\tilde{F}$ can be far from tridiagonal form, and this method *may not be numerically backward stable.* There is also a possible great loss of orthogonality of the computed matrix $\tilde{U}$.

2.4.3 Methods for the Bidiagonal SVD

After bidiagonalization, SVD of the bidiagonal matrix has to be performed to complete the task of computing the singular value decomposition of a general matrix:

$$B = U_B \Sigma V_B^T.$$

The final singular value decomposition is then achieved by

$$A = (UU_B)\Sigma(VV_B)^T.$$

There are several methods for computing bidiagonal SVD. They all assume that the bidiagonal matrix B (2.43) is unreduced. If $\phi_{k+1} = 0$ for some k, then

$$B = \begin{bmatrix} B_1 & 0 \\ 0 & B_2 \end{bmatrix} \begin{matrix} k \\ n-k \end{matrix}$$
$$\qquad\quad k \quad n-k$$

and the original SVD problem is reduced to two smaller problems involving matrices B_1 and B_2.

Three such methods are listed below in historical order.

The Golub–Kahan bidiagonal SVD

This method is also described in [33], and is based on the *implicit-shift QR* steps applied to the tridiagonal matrix $T_0 = B_0^T B_0$, where $B_0 = B$:

$$\begin{aligned} T_k - \lambda_k I &= UR \quad \text{(QR factorization)} \\ T_{k+1} &= RU + \lambda_k I, \quad k = 0, 1, \ldots \end{aligned}$$

which will produce a new tridiagonal T_{k+1} with $T_{k+1} = U^T T_k U$, but the matrix $T_{k+1} = B_{k+1}^T B_{k+1}$ is never explicitly formed. The shift λ_k is chosen to improve the convergence. The steps of the algorithm are following:

- Compute the eigenvalue λ_k of

$$T_k(n-1:n, n-1:n) = \begin{bmatrix} \psi_{n-1}^2 + \phi_{n-1}^2 & \psi_{n-1}\phi_n \\ \psi_{n-1}\phi_n & \psi_n^2 + \phi_n^2 \end{bmatrix}$$

  that is closer to $\psi_n^2 + \phi_n^2$.

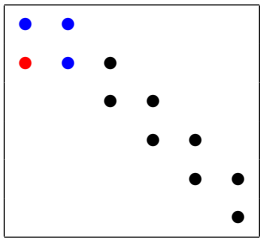
- Compute $c_{\theta_1} = \cos(\theta_1)$ and $s_{\theta_1} = \sin(\theta_1)$ such that

$$\begin{bmatrix} c_{\theta_1} & s_{\theta_1} \\ -s_{\theta_1} & c_{\theta_1} \end{bmatrix}^T \begin{bmatrix} \psi_1^2 - \lambda_k \\ \psi_1\phi_2 \end{bmatrix} = \begin{bmatrix} * \\ 0 \end{bmatrix}$$

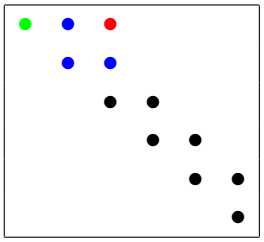
  and set a Givens rotation $V_1 = G_{1,2}(\theta_1)$.

- Compute Givens rotations $V_2, \ldots, V_{n-1}$ so that if $V^{(k)} = V_1 \cdots V_{n-1}$ then $T_{k+1} = (V^{(k)})^T T_k V^{(k)}$ is tridiagonal and $V^{(k)} e_1 = V_1 e_1$. This is done by "chasing the bulge" in the bidiagonal matrix B_k:

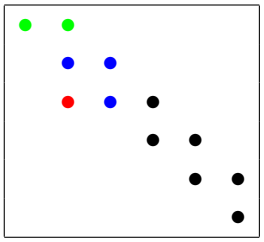
$$B_{k,1} \longleftarrow B_k V_1 \quad =$$



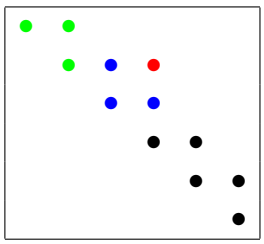
$$B_{k,2} \longleftarrow U_1^T B_{k,1} \quad =$$



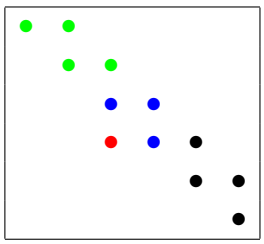
$$B_{k,3} \longleftarrow B_{k,2} V_2 \quad =$$



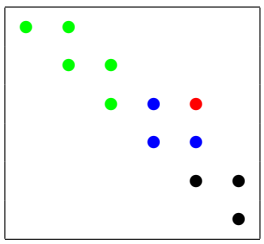
$$B_{k,4} \longleftarrow U_2^T B_{k,3} =$$



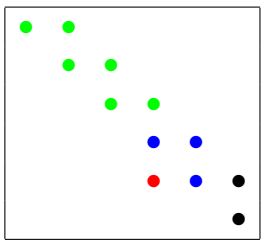
$$B_{k,5} \longleftarrow B_{k,4} V_3 =$$



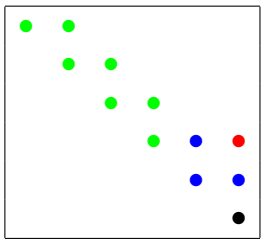
$$B_{k,6} \longleftarrow U_3^T B_{k,5} =$$



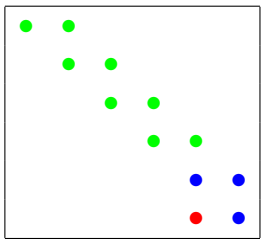
$$B_{k,7} \longleftarrow B_{k,6} V_4 =$$



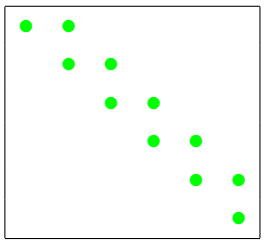
$$B_{k,8} \longleftarrow U_4^T B_{k,7} =$$



$$B_{k,9} \longleftarrow B_{k,8} V_5 =$$



$$B_{k,10} \longleftarrow U_5^T B_{k,9} =$$



$$= B_{k+1}$$

This step terminates with a new bidiagonal matrix B_{k+1} which is related to B_k as follows

$$B_{k+1} = (U_{n-1}^T \cdots U_1^T) B_k (V_1 \cdots V_{n-1}) = (U^{(k)})^T B_k V^{(k)}.$$

The whole process converges to a diagonal matrix, thus

$$\lim_{k\to\infty} B_k = \Sigma.$$

The Demmel–Kahan bidiagonal SVD

This algorithm is a variation of the Golub–Kahan algorithm, and is called the *implicit zero-shift QR* algorithm. Demmel and Kahan noticed in [15] that for $\lambda_k = 0$ Golub–Kahan's implicit-shift QR factorization can be modified in such a way that in finite precision arithmetic every entry of $\tilde{B}_{k+1}$ can be computed from $\tilde{B}_k$ to nearly full machine precision. This implies that, while the Golub–Kahan algorithm guarantees small absolute error in computed singular values:

$$|\tilde{\sigma}_i - \sigma_i| \leq O(\varepsilon)\|B\|_2, \quad i = 1, \ldots, n,$$

the Demmel–Kahan algorithm ensures that the computed singular values have small relative error:

$$\frac{|\tilde{\sigma}_i - \sigma_i|}{\sigma_i} \leq O(\varepsilon), \quad i = 1, \ldots, n.$$

In fact, the new algorithm is a hybrid of the standard QR and the implicit zero-shift QR. The standard QR is used when the condition number of B is modest. If the condition number is large, then the implicit zero-shift QR is used instead. Contrary to the standard Golub–Kahan QR algorithm where the bulge is always chased downwards, implicit zero-shift QR chooses to chase the bulge up or down, depending on which direction will speed up the convergence.

Algorithm 2.4.2 (Implicit Zero–Shift QR Algorithm). *Let, for $k = 0, 1, \ldots$, B_k be an $n \times n$ bidiagonal matrix with diagonal entries $\psi_1^{(k)}, \ldots, \psi_n^{(k)}$ and super-diagonal entries $\phi_1^{(k)}, \ldots, \phi_{n-1}^{(k)}$. The following algorithm computes a new bidiagonal matrix B_{k+1} with entries $\psi_1^{(k+1)}, \ldots, \psi_n^{(k+1)}$ and $\phi_1^{(k+1)}, \ldots, \phi_{n-1}^{(k+1)}$ corresponding to one step of the QR iteration with zero shift:*

$oldc = 1$;
$c = 1$;
for $i = 1 : n - 1$
  $[r, c, s] = \mathbf{rot}(\psi_i^{(k)} \cdot c, \phi_i^{(k)})$;
  if $i > 1$
    $\psi_{i-1}^{(k+1)} = olds \cdot r$;
  end
  $[\psi_i^{(k+1)}, oldc, olds] = \mathbf{rot}(oldc \cdot r, \psi_{i+1}^{(k)} \cdot s)$;
end

$h = \psi_n^{(k)} \cdot c;$
$\phi_{n-1}^{(k+1)} = h \cdot olds;$
$\psi_n^{(k+1)} = h \cdot oldc;$

function $[r, c, s] = \mathbf{rot}(f, g)$
{*The function* **rot**() *takes* f *and* g *as inputs and returns* r*, and a Givens rotation with* $c = \cos(\theta)$ *and* $s = \sin(\theta)$ *such that*

$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix} \cdot \begin{bmatrix} f \\ g \end{bmatrix} = \begin{bmatrix} r \\ 0 \end{bmatrix}.$$

}

Differential qd algorithms

The differential qd algorithm was developed from the *Cholesky LR* transformations (similar to QR) by Fernando and Parlett in [29]. The algorithm obtains maximal relative accuracy for all singular values as the Demmel–Kahan implicit zero-shift QR, but is at least four times faster. It also allows non-zero shifts for increasing the convergence.

We will start with zero-shift, and the Cholesky LR algorithm applied to tridiagonal $T = B^T B$. Let $T_0 = B_0^T B_0$, $B_0 = B$, then for $k = 0, 1, \ldots$ $T_{k+1} = B_k B_k^T$ is also tridiagonal, and we define its Cholesky factorization as

$$T_{k+1} = B_{k+1}^T B_{k+1}.$$

It can be shown that there exists an orthogonal matrix Q such that

$$B_k^T = Q B_{k+1}$$

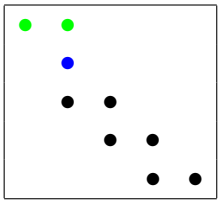
so B_{k+1} is the triangular QR factor of B_k^T.

The matrix Q may be written as a product of $(n-1)$ Givens rotations

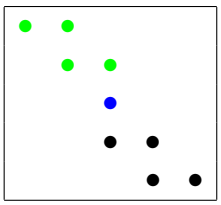
$$Q = G_1 G_2 \cdots G_{n-1}.$$

The annihilation of the subdiagonal elements of B_k^T is done as follows

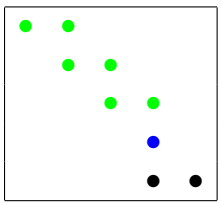
$$B_{k,1}^T \longleftarrow G_1^T B_k^T \quad =$$



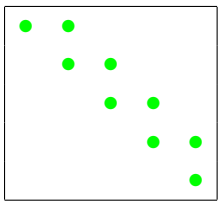
$$B_{k,2}^T \longleftarrow G_2^T B_{k,1}^T \quad =$$



$$B_{k,3}^T \longleftarrow G_3^T B_{k,2}^T =$$



$$B_{k,4}^T \longleftarrow G_4^T B_{k,3}^T =$$



$$= B_{k+1}$$

This procedure yields the *dqd* algorithm by eliminating square roots.

Algorithm 2.4.3 (dqd). *Let, for $k = 0, 1, \ldots$, B_k be an $n \times n$ bidiagonal matrix with diagonal entries $\psi_1^{(k)}, \ldots, \psi_n^{(k)}$ and superdiagonal entries $\phi_1^{(k)}, \ldots, \phi_{n-1}^{(k)}$. The following algorithm computes a new bidiagonal matrix B_{k+1} with entries $\psi_1^{(k+1)}, \ldots, \psi_n^{(k+1)}$ and $\phi_1^{(k+1)}, \ldots, \phi_{n-1}^{(k+1)}$ corresponding to one step of the Cholesky LR iteration:*

$\phi_n^{(k)} = 0$;
for $i = 1 : n$
  $q_i^{(k)} = (\psi_i^{(k)})^2$; $e_i^{(k)} = (\phi_i^{(k)})^2$;
end
$d = q_1^{(k)}$;
for $i = 1 : n - 1$
  $q_i^{(k+1)} = d + e_i^{(k)}$;
  $e_i^{(k+1)} = e^{(k)} \cdot (q_{i+1}^{(k)} / q_i^{(k+1)})$;
  $d = d \cdot (q_{i+1}^{(k)} / q_i^{(k+1)})$;
end
$q_n^{(k+1)} = d$;
$\phi_n^{(k+1)} = 0$;
for $i = 1 : n$
  $\psi_i^{(k+1)} = \sqrt{q_i^{(k+1)}}$; $\phi_i^{(k+1)} = \sqrt{e_i^{(k+1)}}$;
end

The whole process converges to a diagonal matrix, thus

$$\lim_{k \to \infty} B_k = \Sigma.$$

A shift $\lambda_k \neq 0$ can be introduced into the qd algorithm, so that

$$B_{k+1}^T B_{k+1} = B_k B_k^T - \lambda_k I.$$

To keep B_{k+1} real, the shift must satisfy $\lambda_k \leq \sigma_n(B_k)^2$, where $\sigma_n(B_k)$ is the smallest singular value of B_k. All the modifications in the algorithm involve terms with $q_i^{(k+1)}$.

Algorithm 2.4.4 (dqds). *Let, for $k = 0, 1, \ldots$, B_k be an $n \times n$ bidiagonal matrix with diagonal entries $\psi_1^{(k)}, \ldots, \psi_n^{(k)}$ and superdiagonal entries $\phi_1^{(k)}, \ldots, \phi_{n-1}^{(k)}$. The following algorithm computes a new bidiagonal matrix B_{k+1} with entries $\psi_1^{(k+1)}, \ldots, \psi_n^{(k+1)}$ and $\phi_1^{(k+1)}, \ldots, \phi_{n-1}^{(k+1)}$ corresponding to one step of the Cholesky LR iteration with shift:*

$\phi_n^{(k)} = 0$;
for $i = 1 : n$
  $q_i^{(k)} = (\psi_i^{(k)})^2$; $e_i^{(k)} = (\phi_i^{(k)})^2$;
end
$d = q_1^{(k)} - \lambda_k$;
for $i = 1 : n - 1$
  $q_i^{(k+1)} = d + e_i^{(k)}$;
  $e_i^{(k+1)} = e^{(k)} \cdot (q_{i+1}^{(k)} / q_i^{(k+1)})$;
  $d = d \cdot (q_{i+1}^{(k)} / q_i^{(k+1)}) - \lambda_k$;
end
$q_n^{(k+1)} = d$;
$\phi_n^{(k+1)} = 0$;
for $i = 1 : n$
  $\psi_i^{(k+1)} = \sqrt{q_i^{(k+1)}}$; $\phi_i^{(k+1)} = \sqrt{e_i^{(k+1)}}$;
end

We should note that, for singular values $\sigma_i(B_k)$ and $\sigma_i(B_{k+1})$ of the matrices B_k and B_{k+1} respectively, the following statement holds

$$\sigma_i(B_{k+1})^2 = \sigma_i(B_k)^2 - \lambda_k,$$

so that

$$\lim_{k \to \infty} B_k = \Delta,$$

where

$$\Delta^2 = \Sigma^2 - (\lambda_0 + \lambda_1 + \cdots) I.$$

Recently, Dhillon and Parlett developed in [17] a new method for computing eigenvalues of a symmetric tridiagonal matrix, called *Multiple Relatively Robust Representations (MRRR)*. Let $T \in \mathbb{R}^{n \times n}$ be a symmetric tridiagonal matrix, and let $T = L_0 D_0 L_0^T$ be its factorization, where L_0 is unit lower bidiagonal and D is diagonal. The algorithm is based on the compositions of the form

$$L_c D_c L_c^T = L_p D_p L_p^T - \tau I,$$

where τ is a suitable chosen shift. Basically, each new factorization $L_c D_c L_c^T$ corresponds to a cluster of eigenvalues. This new algorithm requires only $O(n^2)$ operations for computing the whole spectral decomposition, and guaranties that the computed eigenvectors are orthogonal to working accuracy and have small residual norms with respect to the original matrix T. This algorithm has been extended by Großer and Lang in [39] to the stable computation of the bidiagonal SVD. In [91] more efficient implementation of the bidiagonal SVD using MRRR is described. The new algorithm is based on the simultaneous computation of eigenvalues and eigenvectors of the matrices $B^T B$, BB^T and $\begin{bmatrix} 0 & B \\ B^T & 0 \end{bmatrix}$ by using so-called coupling relations.

Chapter 3

The Barlow One–sided Bidiagonalization

3.1 The Algorithm

The efficiency of the Ralha bidiagonalization on the one hand, and the numerical instability of the same algorithm on the other, was the motivation for developing its modification. The required modification should have retained the same operation count, but should have improved numerical stability. On the IWASEP 4 workshop held 2002 in Split, Croatia, Barlow proposed a modification of Ralha's algorithm, which seemed to satisfy both of the requirements. The changes in the algorithm were minimal, but subtle, some operations exchanged their places, and one vector was obtained from a different matrix, thus the number of operations remained the same. The proof of numerical stability of the new algorithm was given by Barlow, Bosner and Drmač in [2], and is rather technical. A much simpler version of the same proof will be presented in the next section of this thesis.

In contrast to the Ralha bidiagonalization, which is based on the implicit tridiagonalization of the matrix A^TA, Barlow's algorithm is based on direct bidiagonalization of the matrix A, like the Householder bidiagonalization and the Lanczos approach. In the new algorithm one step of the Gram–Schmidt orthogonalization and post-multiplication with one Householder reflector are performed simultaneously.

Once again, if $A \in \mathbb{R}^{m\times n}$ is given, the algorithm finds matrices $U \in \mathbb{R}^{m\times n}$, $V,\ B \in \mathbb{R}^{n\times n}$, such that

$$A = UBV^T, \qquad U \in \mathbb{R}^{m\times n}, \text{ and } B, V \in \mathbb{R}^{n\times n},$$

where U is orthonormal, V is orthogonal and B is bidiagonal

$$B = \begin{bmatrix} \psi_1 & \phi_2 & & & \\ & \psi_2 & \phi_3 & & \\ & & \ddots & \ddots & \\ & & & \psi_{n-1} & \phi_n \\ & & & & \psi_n \end{bmatrix}. \tag{3.1}$$

The Barlow bidiagonalization can be described in its simplest form as follows

- $A_0 = A$
- For $k = 1, 2, \ldots,$
  - u_k is produced from the k-th column of A_{k-1} by orthogonalization against u_{k-1} (if $k > 1$), and normalization

    $$U_k = [u_1, \ldots, u_k] \quad k = 1, \ldots, n$$

  - The Householder reflector $\mathbf{V}_k$ is chosen so that

    $$U_k^T A_{k-1} \mathbf{V}_k = B_k \in \mathbb{R}^{k \times n}, \qquad B_k \text{ is bidiagonal,} \tag{3.2}$$

    and the matrix A_{k-1} is postmultiplied with $\mathbf{V}_k$

    $$A_k = A_{k-1} \mathbf{V}_k \quad k = 1, \ldots, n-2.$$

- End of loop
- V is produced by accumulation of the Householder reflectors

$$V = \mathbf{V}_1 \cdots \mathbf{V}_{n-2}, \qquad F = A_{n-2} = AV. \tag{3.3}$$

The main difference between Ralha's and Barlow's algorithm is that in the Barlow bidiagonalization, transformations with the Householder reflectors and the Gram–Schmidt orthogonalization are interlaced and not separated as in the Ralha bidiagonalization. The criteria for choosing the Householder reflectors are also different.

If we define

$$F = [f_1, \ldots, f_n] = A_{n-2},$$

then the matrix F is also implicitly triorthogonal, and $F^T F$ is tridiagonal. In case when the matrix A has full column rank, then from condition (3.2) it follows that

$$U^T F = B, \quad \Longrightarrow \quad F = UB, \qquad \text{where } U = U_n.$$

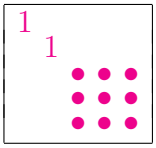
Thus

$$F^T F = B^T U^T U B = B^T B = T,$$

where T is a tridiagonal matrix.

The steps of the Barlow bidiagonalization are visualized in Figure 3.1. The elements denoted by • will be used in the next step of the algorithm to compute vector z_k, such that the next Householder reflector $\mathbf{V}_k$ will depend on it. The Householder reflector $\mathbf{V}_k$ is defined as

$$\mathbf{V}_k = \begin{bmatrix} I_k & 0 \\ 0 & V_k \end{bmatrix} = \tag{3.4}$$



and the Householder reflector $V_k \in \mathbb{R}^{(n-k)\times(n-k)}$ will be chosen so that $V_k z_k = \pm\|z_k\|_2 e_1$. The elements denoted by • are computed columns of F after application of the Householder reflector, and in the next steps they will not be changed. The computed elements of B are denoted by •.

The details of the Barlow bidiagonalization are shown in Algorithm 3.1.1.

Algorithm 3.1.1 (The Barlow one–sided bidiagonalization). *For $A \in \mathbb{R}^{m\times n}$, $\mathrm{rank}(A) = n > 2$, this algorithm computes orthonormal $U = [u_1, \ldots, u_n]$, bidiagonal B having the form (3.1), and orthogonal V such that $A = UBV^T$.*

(1) $A_0 = A$;
(2) $f_1 = A(:,1)$; $\psi_1 = \|f_1\|_2$;
(3) $u_1 = f_1/\psi_1$;
for $k = 1\colon n-2$
  (4) $z_k = A_{k-1}(:,k+1\colon n)^T u_k$;
  (5) $[\gamma_k, v_k]$=**householder**(z_k);
  (6) $A_k(:,1\colon k) = A_{k-1}(:,1\colon k)$;
  (7) $A_k(:,k+1\colon n) = A_{k-1}(:,k+1\colon n) - A_{k-1}(:,k+1\colon n)v_k v_k^T$;
  (8) $f_{k+1} = A_k(:,k+1)$; $\phi_{k+1} = u_k^T f_{k+1}$; $(\phi_{k+1} = \gamma_k;)$
  (9) $s_{k+1} = f_{k+1} - \phi_{k+1}u_k$; $\psi_{k+1} = \|s_{k+1}\|_2$;
  (10) $u_{k+1} = s_{k+1}/\psi_{k+1}$;
end;
(11) $f_n = A_{n-2}(:,n)$; $\phi_n = u_{n-1}^T f_n$;
(12) $s_n = f_n - \phi_n u_{n-1}$; $\psi_n = \|s_n\|_2$;
(13) $u_n = s_n/\psi_n$;
(14) $V^T =$ **householder_product**$(v_1, \ldots, v_{n-2})$
end.

The auxiliary functions **householder**() and **householder_product**() are defined as follows

function $[\gamma, v]$=**householder**(z)
{*The function* **householder**() *computes* γ *and* v *such that, for* $V = I - vv^T$, $Vz = \gamma e_1$.}
(1) $n = \mathrm{length}(z)$;
(2) $\gamma = \|z\|_2$;
if $\gamma > 0$
  (3) $\gamma = -\mathrm{sign}(z(1))\gamma$;
  (4) $t(1) = z(1) - \gamma$;
  (5) $t(2\colon n) = z(2\colon n)$;
  (6) $v = \sqrt{2}t/\|t\|_2$;
else
  (7) $v = 0$;
end;

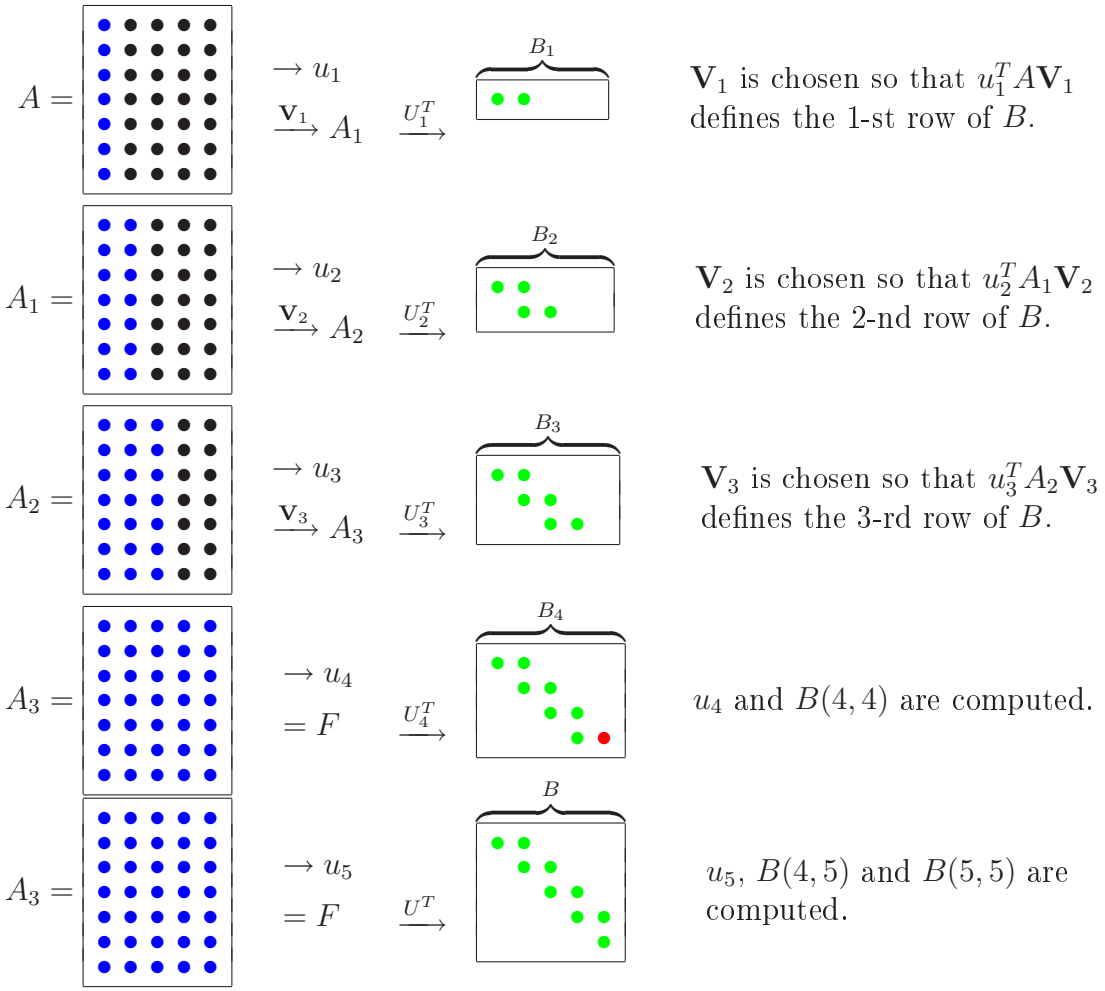


Figure 3.1: The Barlow one–sided bidiagonalization algorithm

function V^T = **householder_product**$(v_1, \ldots, v_n)$
{*The function* **householder_product**() *computes a matrix* V^T *as a product of* n *Householder reflectors, where* $V^T = \mathbf{V}_n \cdots \mathbf{V}_1$, $\mathbf{V}_k$ *for* $k = 1, \ldots, n$ *are defined in relation (3.4), and* $V_k = I - v_k v_k^T$. *Accumulation of the Householder reflectors are done by block algorithm implemented in the LAPACK routine* `sorgbr()` *[1].*}

Remark 3.1.2. *This is the first version of the Barlow algorithm, which mostly resembles the Ralha algorithm. Barlow noticed (see [2]) that computation of* ϕ_{k+1} *in step* ***(8)*** *as a scalar product is completely redundant, since in exact arithmetic it is equivalent to* $\phi_{k+1} = \gamma_k$. *Still, the numerical analysis in Theorem 3.2.6 is done for this original version, and in Remark 3.2.7 the same is done for the alternative choice of* ϕ_{k+1}. *It turns out that both versions give the same error bound, but the second one has less floating point operations, and thus it is more favorable.*

As we can see the main difference between the Ralha and the Barlow bidiagonaliza-

tions is the way the vector z_k is computed. In the Ralha bidiagonalization it is

$$z_k = A_{k-1}(:, k+1:n)^T A_{k-1}(:, k)$$

and in Barlow's modification

$$z_k = A_{k-1}(:, k+1:n)^T u_k.$$

Although these two bidiagonalizations are mathematically equivalent in exact arithmetic, numerically they differ. The difference between the computations of the vector z_k is responsible for the Barlow bidiagonalization being numerically stable, as it will be shown in Theorem 3.2.6.

Algorithm 3.1.1 will not break down in case when non of ψ_k is zero. The procedure how to proceed with the algorithm when $\psi_k = 0$, is described in [2, Section 5]. It uses Givens rotations to produce a compact bidiagonal factorization, with a bidiagonal matrix of smaller dimension and with all diagonal elements different from zero. So, from now on we can assume that Algorithm 3.1.1 will not break down, and that $\psi_k \neq 0$, for all $k = 1, \ldots, n$.

3.2 Numerical Stability

The goal of this section is to prove that the Barlow algorithm is backward stable. That means that the singular values of the matrix A computed in finite precision arithmetic are exact singular values of a matrix not far away from A. The difference between these two matrices, measured in $\|\cdot\|_2$ or $\|\cdot\|_F$ matrix norms, is called the **backward error**. The backward error is considered to be small if it is smaller than $\xi\|A\|_2$ or $\eta\|A\|_F$, where ξ and η are bounded by the machine roundoff ε times a moderate polynomial of matrix dimensions.

In the numerical analysis that will follow we will use the following notation. The values computed in finite precision arithmetic will be denoted by $\tilde{}$ and occasionally by $\bar{}$, while exactly computed values will be denoted by $\hat{}$. These exact values will serve only for analytical purposes, and are never actually computed.

First we will list some auxiliary results concerning the numerical analysis of the Householder QR factorization, presented by Higham in [47]. These results are necessary for the analysis of post-multiplication of the matrix A with Householder reflectors.

Lemma 3.2.1 ([47, p. 365]). *Let $x \in \mathbb{R}^m$. Consider the following construction of $\tau \in \mathbb{R}$ and $v \in \mathbb{R}^m$ such that $Px = \gamma e_1$, where $P = I - \tau v v^T$ is a Householder reflector with $\tau = 2/(v^T v)$:*

$$\begin{aligned} v &= x \\ s &= \operatorname{sign}(x(1))\|x\|_2 \\ \gamma &= -s \\ v(1) &= v(1) + s \\ \tau &= \frac{1}{s \cdot v(1)} \end{aligned}$$

In floating point arithmetic the computed $\tilde{\tau}$ *and* $\tilde{v}$ *satisfy* $\tilde{v}(2:m) = v(2:m)$ *and*

$$\begin{aligned} \tilde{\tau} &= \tau(1+\theta_{4m+8}), \\ \tilde{v}(1) &= v(1)(1+\theta_{m+2}), \end{aligned}$$

where $|\theta_k| \leq k\varepsilon + O(\varepsilon^2)$.

Remark 3.2.2 ([47, p. 366]). *If we write the Householder reflectors in the form* $I - \hat{v}\hat{v}^T$, *where* $\|\hat{v}\|_2 = \sqrt{2}$, *we can rewrite the results from Lemma 3.2.1 as*

$$\tilde{v} = \hat{v} + \delta\hat{v}, \quad |\delta\hat{v}| \leq O(m)\varepsilon|\hat{v}|, \quad \text{for } \hat{v} \in \mathbb{R}^m, \quad \|\hat{v}\|_2 = \sqrt{2}. \tag{3.5}$$

Lemma 3.2.3 ([47, pp. 366–367]). *Let* $b \in \mathbb{R}^m$ *and consider the computation of* $y = \tilde{P}b = (I - \tilde{v}\tilde{v}^T)b = b - \tilde{v}(\tilde{v}^T b)$, *where* $\tilde{v} \in \mathbb{R}^m$ *satisfies (3.5). The computed* $\tilde{y}$ *satisfies*

$$\tilde{y} = (\hat{P} + \delta\hat{P})b, \qquad \|\delta\hat{P}\|_F \leq O(m)\varepsilon,$$

where $\hat{P} = I - \hat{v}\hat{v}^T$.

Lemma 3.2.4 ([47, pp. 367–368]). *Consider the sequence of transformations*

$$A_{k+1} = P_k A_k, \qquad k = 1, \ldots, r,$$

where $A_1 = A \in \mathbb{R}^{m\times n}$ *and* $P_k = I - v_k v_k^T \in \mathbb{R}^{m\times m}$ *is a Householder reflector. Assume that the transformations are performed using computed Householder vectors* $\tilde{v}_k \approx \hat{v}_k$ *that satisfy (3.5). The computed matrix* $\tilde{A}_{r+1}$ *satisfies*

$$\tilde{A}_{r+1} = \hat{Q}^T(A + \delta A),$$

where $\hat{Q}^T = \hat{P}_r\hat{P}_{r-1}\cdots\hat{P}_1$ *and* δA *satisfies the normwise and componentwise bounds*

$$\begin{aligned} \|\delta A\|_F &\leq O(rm)\varepsilon\|A\|_F, \\ |\delta A| &\leq O(rm^2)\varepsilon G|A|, \qquad \|G\|_F = 1. \end{aligned}$$

(In fact, we can take $G = m^{-1}ee^T$, *where* $e = [1, 1, \ldots, 1]^T$.*) In the special case* $n = 1$, *so that* $A \equiv a$, *we have* $\tilde{a}_{r+1} = (\hat{Q} + \delta\hat{Q})a$ *with* $\|\delta\hat{Q}\|_F \leq O(rm)\varepsilon$.

Theorem 3.2.5 ([47, p. 368]). *Let* $\tilde{R} \in \mathbb{R}^{m\times n}$ *be the computed upper trapezoidal QR factor of* $A \in \mathbb{R}^{m\times n}$ $(m \geq n)$ *obtained via the Householder QR algorithm. Then there exists an orthogonal* $\hat{Q} \in \mathbb{R}^{m\times m}$ *such that*

$$A + \delta A = \hat{Q}\tilde{R},$$

where $\|\delta A\|_F \leq O(nm)\varepsilon\|A\|_F$ *and* $|\delta A| \leq O(nm^2)\varepsilon G|A|$, *with* $\|G\|_F = 1$. *The matrix* $\hat{Q}$ *is given explicitly as* $\hat{Q} = (\hat{P}_n\hat{P}_{n-1}\cdots\hat{P}_1)^T$, *where* $\hat{P}_k$ *is the Householder matrix that corresponds to exact application of the* k*-th step of the algorithm to* $\tilde{A}_k$.

Next, Björck and Paige in [6] observed that the modified Gram–Schmidt orthogonalization is mathematically and numerically equivalent to the Householder QR factorization of the augmented matrix

$$\begin{bmatrix} 0_n \\ A \end{bmatrix} \in \mathbb{R}^{(m+n)\times n}. \tag{3.6}$$

The results on the Householder reflectors: Lemma 3.2.1, Remark 3.2.2, Lemma 3.2.3, Lemma 3.2.4 and Theorem 3.2.5, can now be applied to the augmented matrix (3.6) and the Gram–Schmidt orthogonalization. This approach will be used in the numerical analysis of the Barlow bidiagonalization.

The main result on numerical backward stability of the Barlow bidiagonalization is given in the following theorem.

Theorem 3.2.6. *If $\tilde{B}$ is the bidiagonal matrix computed by Algorithm 3.1.1 without breakdown (all ψ_k's different from zero), then there exist an $(m+n)\times(m+n)$ orthogonal matrix $\hat{P}$, an orthogonal $n\times n$ matrix $\hat{V}$, and backward perturbations ΔA, δA such that*

$$\begin{bmatrix} \tilde{B} \\ 0 \end{bmatrix} = \hat{P}^T \begin{bmatrix} \Delta A \\ A+\delta A \end{bmatrix} \hat{V}, \quad \left\| \begin{bmatrix} \Delta A \\ \delta A \end{bmatrix} \right\|_F \leq \xi \|A\|_F, \tag{3.7}$$

where $0 \leq \xi \leq O(mn+n^3)\varepsilon$. The computed approximation $\tilde{V}$ of the matrix $\hat{V}$ satisfies $\|\tilde{V}-\hat{V}\|_F \leq O(n^2)\varepsilon$. Further, there exist an orthonormal $\hat{U}$ and a perturbation $\delta\hat{A}$ such that

$$A+\delta\hat{A} = \hat{U}\tilde{B}\hat{V}^T, \quad \|\delta\hat{A}\|_F \leq \sqrt{2}\xi\|A\|_F. \tag{3.8}$$

Proof. Let us first explore the details of Algorithm 3.1.1 in exact arithmetic. The algorithm consists of application of the Householder reflectors from the right, and the Gram–Schmidt orthogonalization. We define Householder reflectors $\mathbf{V}_1,\ldots,\mathbf{V}_{n-2} \in \mathbb{R}^{n\times n}$ in exact arithmetic as

$$\mathbf{V}_k = \begin{bmatrix} I_k & 0 \\ 0 & V_k \end{bmatrix}, \qquad k = 1,\ldots,n-2,$$

where I_k is an $k\times k$ identity matrix, and $V_k \in \mathbb{R}^{(n-k)\times(n-k)}$ is a $(n-k)\times(n-k)$ Householder reflector

$$V_k = I_{n-k} - v_k v_k^T,$$

such that

$$V_k z_k = \gamma_k e_1.$$

In the exact arithmetic γ_k is equal to ϕ_{k+1}, as shown in [2]. From Algorithm 3.1.1, we can also see that $A_k = A_{k-1}\mathbf{V}_k$.

Further the process can be represented as

$$\underbrace{\overbrace{\underbrace{A_0}_{get\ B(:,1)}\cdot\mathbf{V}_1}^{get\ B(:,2)}\cdot\mathbf{V}_2\cdot\mathbf{V}_3\cdot\cdots\cdots}_{get\ B(:,4)} \tag{3.9}$$

After producing A_k in the k-th step of the algorithm, the $(k+1)$-th column of B is computed. Simultaneously, the k-th row of B is also computed, as shown in Figure 3.1. Since each transformation $A_k = A_{k-1}\mathbf{V}_k$ affects only the columns $k+1:n$ and the first k columns are left unchanged, we conclude from the above diagram that the same matrix B is computed if we apply all $\mathbf{V}_k$'s first, and then the Gram–Schmidt orthogonalizations. In other words, if we set $F = A_{n-2}$, then we can use $f_{k+1} = F(:, k+1)$ instead of $f_{k+1} = A_k(:, k+1)$. Note that this separation of the implicit Householder tridiagonalization and the Gram–Schmidt computation of U and B is artificial, and is introduced only for the purposes of the analysis. The fact that these two processes are interwoven is crucial for the numerical properties of the algorithm.

Everything mentioned above also applies to finite precision arithmetic and the matrices $\tilde{A}^{(k)}$, $\tilde{\mathbf{V}}_k$ for $k = 1, \ldots, n-2$, and $F = \tilde{A}^{(n-2)}$, $\tilde{B}$. Note that F is never computed in the Algorithm 3.1.1, and it is only used in the analysis. Thus, the computed matrix $\tilde{A}^{(n-2)}$ will be denoted by F instead of $\tilde{F}$.

The proof of numerical stability of Algorithm 3.1.1 is technical and rather complicated. Thus the proof will be divided into four steps, concerning four important points in the proof.

Step 1: The Householder transformations

In finite precision arithmetic, the computed matrix F is obtained as

$$F = \tilde{A}_{n-2} = \mathrm{fl}((\cdots((A \cdot \tilde{\mathbf{V}}_1) \cdot \tilde{\mathbf{V}}_2) \cdots) \cdot \tilde{\mathbf{V}}_{n-2}),$$

where $\tilde{\mathbf{V}}_k$, $k = 1, \ldots, n-2$ are computed Householder reflectors. By Lemma 3.2.4 there exists an exactly orthogonal matrix $\hat{V} = \hat{\mathbf{V}}_1 \cdots \hat{\mathbf{V}}_{n-2}$, such that

$$F = (A + \delta_1 A)\hat{V}, \qquad \|\delta_1 A\|_F \leq \eta_F \|A\|_F, \ \eta_F \leq O(n^2)\varepsilon, \tag{3.10}$$

where

$$\hat{\mathbf{V}}_k = \begin{bmatrix} I_k & 0 \\ 0 & \hat{V}_k \end{bmatrix}, \qquad k = 1, \ldots, n-2,$$

and $\hat{V}_k \in \mathbb{R}^{(n-k)\times(n-k)}$ is an $(n-k) \times (n-k)$ exact Householder reflector $\hat{V}_k = I_{n-k} - \hat{v}_k \hat{v}_k^T$, such that $\hat{V}_k \tilde{z}_k = \hat{\gamma}_k e_1$.

On the other hand, for $k = 1, \ldots, n-2$ we can write

$$\tilde{\mathbf{V}}_k = I - \tilde{\mathbf{v}}_k \tilde{\mathbf{v}}_k^T, \qquad \hat{\mathbf{V}}_k = I - \hat{\mathbf{v}}_k \hat{\mathbf{v}}_k^T,$$

with

$$\tilde{\mathbf{v}}_k = \begin{bmatrix} 0_{k\times 1} \\ \tilde{v}_k \end{bmatrix}, \qquad \hat{\mathbf{v}}_k = \begin{bmatrix} 0_{k\times 1} \\ \hat{v}_k \end{bmatrix},$$

where $\tilde{v}_k$ represents the computed Householder vector described in Lemma 3.2.1 and Remark 3.2.2. Further, it follows

$$\begin{aligned} \tilde{\mathbf{V}}_k &= I - \tilde{\mathbf{v}}_k \tilde{\mathbf{v}}_k^T = I - (\hat{\mathbf{v}}_k + \delta\hat{\mathbf{v}}_k)(\hat{\mathbf{v}}_k + \delta\hat{\mathbf{v}}_k)^T = \\ &= \underbrace{I - \hat{\mathbf{v}}_k \hat{\mathbf{v}}_k^T}_{\hat{\mathbf{V}}_k} \underbrace{- \hat{\mathbf{v}}_k \delta\hat{\mathbf{v}}_k^T - \delta\hat{\mathbf{v}}_k \hat{\mathbf{v}}_k^T - \delta\hat{\mathbf{v}}_k \delta\hat{\mathbf{v}}_k^T}_{\delta\hat{\mathbf{V}}_k} = \\ &= \hat{\mathbf{V}}_k + \delta\hat{\mathbf{V}}_k \end{aligned}$$

where by Remark 3.2.2

$$\|\delta\hat{\mathbf{V}}_k\|_F \leq O(n)\varepsilon.$$

The computed matrix $\tilde{V}$ is obtained as

$$\tilde{V} = \mathrm{fl}(\tilde{\mathbf{V}}_1 \cdots \tilde{\mathbf{V}}_{n-2}) = \tilde{\mathbf{V}}_1 \cdots \tilde{\mathbf{V}}_{n-2} + \delta_1\hat{V},$$

and by using a result on the matrix product from [47, p. 78] we can bound the error $\delta_1\hat{V}$ with

$$\|\delta_1\hat{V}\|_F \leq O(n^2)\varepsilon.$$

Finally, we have

$$\begin{aligned}
\tilde{V} &= (\hat{\mathbf{V}}_1 + \delta\hat{\mathbf{V}}_1)\cdots(\hat{\mathbf{V}}_{n-1} + \delta\hat{\mathbf{V}}_{n-2}) + \delta_1\hat{V} = \\
&= \hat{\mathbf{V}}_1\cdots\hat{\mathbf{V}}_{n-2} + [(\hat{\mathbf{V}}_1 + \delta\hat{\mathbf{V}}_1)\cdots(\hat{\mathbf{V}}_{n-2} + \delta\hat{\mathbf{V}}_{n-2}) - \hat{\mathbf{V}}_1\cdots\hat{\mathbf{V}}_{n-2}] + \delta_1\hat{V} = \\
&= \hat{V} + \delta\hat{V}
\end{aligned}$$

where

$$\|\delta\hat{V}\|_F \leq O(n^2)\varepsilon.$$

Step 2: The Gram–Schmidt orthogonalization

Since the computation of $\tilde{B}$ from $F = [f_1, \ldots, f_n]$ corresponds to the modified Gram–Schmidt algorithm, we use results from [6] and represent the computation in equivalent form, as the Householder QR factorization of

$$\begin{bmatrix} 0 \\ F \end{bmatrix} = \begin{bmatrix} 0 \\ A + \delta_1 A \end{bmatrix} \hat{V}.$$

Consider the computation of the k–th column of $\tilde{B}$. An application of the results on floating point computation from [47] reveals that

$$\begin{aligned}
&\tilde{\psi}_1 = \mathrm{fl}(\|f_1\|_2) = \|f_1\|_2 - \delta\tilde{\psi}_1, \qquad \text{where} \quad |\delta\tilde{\psi}_1| \leq O(m)\varepsilon\|f_1\|_2, \\
&\tilde{u}_1 = \mathrm{fl}\left(\frac{f_1}{\tilde{\psi}_1}\right) = \hat{u}_1 + \delta\hat{u}_1, \qquad \text{where} \quad \hat{u}_1 = \frac{f_1}{\|f_1\|_2}, \ \text{ and } \ \|\delta\hat{u}_1\|_2 \leq O(m)\varepsilon.
\end{aligned}$$

Furthermore, for $k = 1, 2, \ldots$ we have

$$\begin{aligned}
\tilde{\phi}_{k+1} &= \mathrm{fl}(\tilde{u}_k^T f_{k+1}) = \hat{u}_k^T f_{k+1} + \delta\tilde{\phi}_{k+1}, \ \ |\delta\tilde{\phi}_{k+1}| \leq O(m)\varepsilon\|f_{k+1}\|_2, \\
\tilde{s}_{k+1} &= \mathrm{fl}(f_{k+1} - \tilde{\phi}_{k+1}\tilde{u}_k) = f_{k+1} - \hat{\phi}_{k+1}\hat{u}_k + \delta\tilde{s}_{k+1},
\end{aligned}$$

where

$$\begin{aligned}
\|\delta\tilde{s}_{k+1}\|_2 &\leq O(m)\varepsilon\|f_{k+1}\|_2, \\
\hat{\phi}_{k+1} &= \hat{u}_k^T f_{k+1}, \qquad\qquad |\hat{\phi}_{k+1}| \leq \|f_{k+1}\|_2.
\end{aligned}$$

Following an idea of Björck and Paige [6], we write this computation as

$$\begin{aligned}\begin{bmatrix} \tilde{\phi}_{k+1}e_k \\ \tilde{s}_{k+1} \end{bmatrix} &= \begin{bmatrix} \hat{\phi}_{k+1}e_k \\ f_{k+1} - \hat{\phi}_{k+1}\hat{u}_k \end{bmatrix} + \begin{bmatrix} \delta\tilde{\phi}_{k+1}e_k \\ \delta\tilde{s}_{k+1} \end{bmatrix} = \\ &= \hat{P}_k \left\{ \begin{bmatrix} 0 \\ f_{k+1} \end{bmatrix} + \hat{P}_k \begin{bmatrix} \delta\tilde{\phi}_{k+1}e_k \\ \delta\tilde{s}_{k+1} \end{bmatrix} \right\}, \\ \hat{P}_k &= I_{m+n} - \begin{bmatrix} -e_k \\ \hat{u}_k \end{bmatrix} \begin{bmatrix} -e_k^T & \hat{u}_k^T \end{bmatrix},\end{aligned}$$

where e_k denotes the k-th column of the identity matrix I_n. Note that $\hat{P}_k^2 = I_{m+n}$. Further, the values $\tilde{\psi}_{k+1} = \mathrm{fl}(\|\tilde{s}_{k+1}\|_2)$, $\tilde{u}_{k+1} = \mathrm{fl}(\tilde{s}_{k+1}/\tilde{\psi}_{k+1})$ satisfy

$$\begin{aligned}\tilde{\psi}_{k+1} &= \|\tilde{s}_{k+1}\|_2 - \delta\tilde{\psi}_{k+1}, \quad |\delta\tilde{\psi}_{k+1}| \le O(m)\varepsilon\|f_{k+1}\|_2, \\ \tilde{u}_{k+1} &= \hat{u}_{k+1} + \delta\hat{u}_{k+1}, \quad \hat{u}_{k+1} = \frac{\tilde{s}_{k+1}}{\|\tilde{s}_{k+1}\|_2}, \quad \|\delta\hat{u}_{k+1}\| \le O(m)\varepsilon. \end{aligned} \tag{3.11}$$

Thus, the computation of the $(k+1)$-th column of $\tilde{B}$ can be written as

$$\begin{aligned}&\begin{bmatrix} \tilde{\phi}_{k+1}e_k + \tilde{\psi}_{k+1}e_{k+1} \\ 0 \end{bmatrix} = \begin{bmatrix} \tilde{\phi}_{k+1}e_k + \|\tilde{s}_{k+1}\|_2 e_{k+1} \\ 0 \end{bmatrix} - \begin{bmatrix} \delta\tilde{\psi}_{k+1}e_{k+1} \\ 0 \end{bmatrix} = \\ &= \hat{P}_{k+1} \left\{ \begin{bmatrix} \tilde{\phi}_{k+1}e_k \\ \tilde{s}_{k+1} \end{bmatrix} - \hat{P}_{k+1} \begin{bmatrix} \delta\tilde{\psi}_{k+1}e_{k+1} \\ 0 \end{bmatrix} \right\} = \\ &= \hat{P}_{k+1} \left\{ \hat{P}_k \begin{bmatrix} 0 \\ f_{k+1} \end{bmatrix} + \begin{bmatrix} \delta\tilde{\phi}_{k+1}e_k \\ \delta\tilde{s}_{k+1} \end{bmatrix} - \hat{P}_{k+1} \begin{bmatrix} \delta\tilde{\psi}_{k+1}e_{k+1} \\ 0 \end{bmatrix} \right\} = \\ &= \hat{P}_{k+1}\hat{P}_k \left\{ \begin{bmatrix} 0 \\ f_{k+1} \end{bmatrix} + \begin{bmatrix} \Delta f_{k+1} \\ \delta f_{k+1} \end{bmatrix} \right\}, \quad \left\| \begin{bmatrix} \Delta f_{k+1} \\ \delta f_{k+1} \end{bmatrix} \right\|_2 \le O(m)\varepsilon\|f_{k+1}\|_2.\end{aligned}$$

In case when $k = 0$ and when the first column of $\tilde{B}$ is computed, we can write $\tilde{\phi}_1 = 0$, $\tilde{s}_1 = f_1$, $\hat{P}_0 = I_{m+n}$. Hence, we can conclude that

$$|\, \|\tilde{B}(:,k+1)\|_2 - \|f_{k+1}\|_2 \,| \le O(m)\varepsilon\|f_{k+1}\|_2.$$

Putting all columns of $\tilde{B}$ together, we get

$$\begin{aligned}\begin{bmatrix} \tilde{B} \\ 0 \end{bmatrix} &= \left[\begin{bmatrix} \tilde{\psi}_1 e_1 \\ 0 \end{bmatrix}, \begin{bmatrix} \tilde{\phi}_2 e_1 + \tilde{\psi}_2 e_2 \\ 0 \end{bmatrix}, \ldots, \begin{bmatrix} \tilde{\phi}_n e_{n-1} + \tilde{\psi}_n e_n \\ 0 \end{bmatrix} \right] = \\ &= \left[\hat{P}_1 \begin{bmatrix} \Delta f_1 \\ f_1 + \delta f_1 \end{bmatrix}, \hat{P}_2\hat{P}_1 \begin{bmatrix} \Delta f_2 \\ f_2 + \delta f_2 \end{bmatrix}, \ldots, \hat{P}_n\hat{P}_{n-1} \begin{bmatrix} \Delta f_n \\ f_n + \delta f_n \end{bmatrix} \right],\end{aligned}$$

and using the fact that

$$\begin{aligned}\hat{P}_i \begin{bmatrix} \tilde{B}(:,j) \\ 0 \end{bmatrix} &= \left(I - \begin{bmatrix} -e_i \\ \hat{u}_i \end{bmatrix} \begin{bmatrix} -e_i^T & \hat{u}_i^T \end{bmatrix} \right) \begin{bmatrix} \tilde{\phi}_j e_{j-1} + \tilde{\psi}_j e_j \\ 0 \end{bmatrix} = \\ &= \begin{bmatrix} \tilde{\phi}_j e_{j-1} + \tilde{\psi}_j e_j \\ 0 \end{bmatrix} - \begin{bmatrix} -e_i \\ \hat{u}_i \end{bmatrix} (-e_i^T(\tilde{\phi}_j e_{j-1} + \tilde{\psi}_j e_j)) = \\ &= \begin{bmatrix} \tilde{\phi}_j e_{j-1} + \tilde{\psi}_j e_j \\ 0 \end{bmatrix} = \begin{bmatrix} \tilde{B}(:,j) \\ 0 \end{bmatrix}, \quad \text{for all } i \ne j, j-1,\end{aligned}$$

we obtain

$$\begin{aligned}\begin{bmatrix} \tilde{B} \\ 0 \end{bmatrix} &= \left[\hat{P}_n\hat{P}_{n-1}\cdots\hat{P}_2\hat{P}_1\begin{bmatrix} \Delta f_1 \\ f_1+\delta f_1 \end{bmatrix}, \hat{P}_n\hat{P}_{n-1}\cdots\hat{P}_2\hat{P}_1\begin{bmatrix} \Delta f_2 \\ f_2+\delta f_2 \end{bmatrix},\right.\\ &\quad \hat{P}_n\hat{P}_{n-1}\cdots\hat{P}_3\hat{P}_2\begin{bmatrix} \Delta f_3 \\ f_3+\delta f_3 \end{bmatrix}, \hat{P}_n\hat{P}_{n-1}\cdots\hat{P}_4\hat{P}_3\begin{bmatrix} \Delta f_4 \\ f_4+\delta f_4 \end{bmatrix},\\ &\quad \left.\cdots, \hat{P}_n\hat{P}_{n-1}\hat{P}_{n-2}\begin{bmatrix} \Delta f_{n-1} \\ f_{n-1}+\delta f_{n-1} \end{bmatrix}, \hat{P}_n\hat{P}_{n-1}\begin{bmatrix} \Delta f_n \\ f_n+\delta f_n \end{bmatrix}\right].\end{aligned}$$

The k–th column of the computed bidiagonal matrix is of the form

$$\hat{P}_n\hat{P}_{n-1}\cdots\hat{P}_k\hat{P}_{k-1}\begin{bmatrix} \Delta f_k \\ f_k+\delta f_k \end{bmatrix},$$

and the desired form is

$$\hat{P}_n\hat{P}_{n-1}\cdots\hat{P}_2\hat{P}_1\begin{bmatrix} \hat{\Delta} f_k \\ f_k+\hat{\delta} f_k \end{bmatrix} = \hat{P}^T\begin{bmatrix} \hat{\Delta} f_k \\ f_k+\hat{\delta} f_k \end{bmatrix}, \quad \hat{P}=\hat{P}_1\hat{P}_2\cdots\hat{P}_{n-1}\hat{P}_n.$$

The first two columns ($k=1,2$) are already in the desired form and $\hat{\Delta} f_k = \Delta f_k$, $\hat{\delta} f_k = \delta f_k$. For $k \geq 3$ we write

$$\begin{bmatrix} \tilde{B}(:,k) \\ 0 \end{bmatrix} = (\hat{P}_n\hat{P}_{n-1}\cdots\hat{P}_k\hat{P}_{k-1}\overbrace{\hat{P}_{k-2}\cdots\hat{P}_2\hat{P}_1)(\hat{P}_1\hat{P}_2\cdots\hat{P}_{k-2})}^{I}\begin{bmatrix} \Delta f_k \\ f_k+\delta f_k \end{bmatrix},$$

then

$$\begin{aligned}&\hat{P}_1\hat{P}_2\cdots\hat{P}_{k-2}\begin{bmatrix} \Delta f_k \\ f_k+\delta f_k \end{bmatrix} = \hat{P}_1\cdots\hat{P}_{k-3}\left\{\hat{P}_{k-2}\left\{\begin{bmatrix} 0 \\ f_k \end{bmatrix}+\begin{bmatrix} \Delta f_k \\ \delta f_k \end{bmatrix}\right\}\right\} =\\ &= \hat{P}_1\cdots\hat{P}_{k-3}\left\{\begin{bmatrix} 0 \\ f_k \end{bmatrix}+\begin{bmatrix} e_{k-2} \\ -\hat{u}_{k-2} \end{bmatrix}(\hat{u}_{k-2}^T f_k)+\hat{P}_{k-2}\begin{bmatrix} \Delta f_k \\ \delta f_k \end{bmatrix}\right\} =\\ &= \hat{P}_1\cdots\hat{P}_{k-3}\left\{\begin{bmatrix} 0 \\ f_k \end{bmatrix}+\begin{bmatrix} \Delta_{k-2} f_k \\ \delta_{k-2} f_k \end{bmatrix}+\hat{P}_{k-2}\begin{bmatrix} \Delta f_k \\ \delta f_k \end{bmatrix}\right\} =\\ &= \hat{P}_1\cdots\hat{P}_{k-4}\left\{\begin{bmatrix} 0 \\ f_k \end{bmatrix}+\begin{bmatrix} \Delta_{k-3} f_k \\ \delta_{k-3} f_k \end{bmatrix}+\hat{P}_{k-3}\begin{bmatrix} \Delta_{k-2} f_k \\ \delta_{k-2} f_k \end{bmatrix}+\hat{P}_{k-3}\hat{P}_{k-2}\begin{bmatrix} \Delta f_k \\ \delta f_k \end{bmatrix}\right\} =\\ &= \begin{bmatrix} 0 \\ f_k \end{bmatrix}+\begin{bmatrix} \Delta_1 f_k \\ \delta_1 f_k \end{bmatrix}+\hat{P}_1\begin{bmatrix} \Delta_2 f_k \\ \delta_2 f_k \end{bmatrix}+\hat{P}_1\hat{P}_2\begin{bmatrix} \Delta_3 f_k \\ \delta_3 f_k \end{bmatrix}+\cdots\\ &\quad \cdots+\hat{P}_1\cdots\hat{P}_{k-3}\begin{bmatrix} \Delta_{k-2} f_k \\ \delta_{k-2} f_k \end{bmatrix}+\hat{P}_1\cdots\hat{P}_{k-2}\begin{bmatrix} \Delta f_k \\ \delta f_k \end{bmatrix} =\\ &= \begin{bmatrix} 0 \\ f_k \end{bmatrix}+\begin{bmatrix} \hat{\Delta} f_k \\ \hat{\delta} f_k \end{bmatrix}, \quad \text{where} \quad \begin{bmatrix} \Delta_j f_k \\ \delta_j f_k \end{bmatrix} = \begin{bmatrix} e_j \\ -\hat{u}_j \end{bmatrix}(\hat{u}_j^T f_k), \quad j=1,\ldots,k-2,\end{aligned}$$

and

$$\begin{bmatrix} \hat{\Delta} f_k \\ \hat{\delta} f_k \end{bmatrix} = \hat{P}_1\cdots\hat{P}_{k-2}\begin{bmatrix} \Delta f_k \\ \delta f_k \end{bmatrix}+\begin{bmatrix} \Delta_1 f_k \\ \delta_1 f_k \end{bmatrix}+\sum_{j=2}^{k-2}\hat{P}_1\cdots\hat{P}_{j-1}\begin{bmatrix} \Delta_j f_k \\ \delta_j f_k \end{bmatrix}.$$

Hence,

$$\begin{aligned}\begin{bmatrix}\tilde{B}\\0\end{bmatrix} &= \hat{P}^T\left[\begin{bmatrix}\hat{\Delta}f_1\\ f_1+\hat{\delta}f_1\end{bmatrix},\ldots,\begin{bmatrix}\hat{\Delta}f_k\\ f_k+\hat{\delta}f_k\end{bmatrix},\ldots,\begin{bmatrix}\hat{\Delta}f_n\\ f_n+\hat{\delta}f_n\end{bmatrix}\right]=\\ &= \hat{P}^T\left\{\begin{bmatrix}0\\F\end{bmatrix}+\begin{bmatrix}\Delta F\\ \delta F\end{bmatrix}\right\},\end{aligned}$$

where, after suitable reordering of the entries in the sums,

$$\begin{aligned}\begin{bmatrix}\Delta F\\ \delta F\end{bmatrix} = &\left[\begin{bmatrix}\Delta f_1\\ \delta f_1\end{bmatrix},\begin{bmatrix}\Delta f_2\\ \delta f_2\end{bmatrix},\ldots,\hat{P}_1\cdots\hat{P}_{k-2}\begin{bmatrix}\Delta f_k\\ \delta f_k\end{bmatrix},\ldots,\hat{P}_1\cdots\hat{P}_{n-2}\begin{bmatrix}\Delta f_n\\ \delta f_n\end{bmatrix}\right]+\\ &+\left[0,0,\begin{bmatrix}\Delta_1 f_3\\ \delta_1 f_3\end{bmatrix},\begin{bmatrix}\Delta_1 f_4\\ \delta_1 f_4\end{bmatrix},\begin{bmatrix}\Delta_1 f_5\\ \delta_1 f_5\end{bmatrix},\ldots,\begin{bmatrix}\Delta_1 f_k\\ \delta_1 f_k\end{bmatrix},\ldots,\begin{bmatrix}\Delta_1 f_n\\ \delta_1 f_n\end{bmatrix}\right]+\\ &+\hat{P}_1\left[0,0,0,\begin{bmatrix}\Delta_2 f_4\\ \delta_2 f_4\end{bmatrix},\begin{bmatrix}\Delta_2 f_5\\ \delta_2 f_5\end{bmatrix},\ldots,\begin{bmatrix}\Delta_2 f_k\\ \delta_2 f_k\end{bmatrix},\ldots,\begin{bmatrix}\Delta_2 f_n\\ \delta_2 f_n\end{bmatrix}\right]+\\ &+\hat{P}_1\hat{P}_2\left[0,0,0,0,\begin{bmatrix}\Delta_3 f_5\\ \delta_3 f_5\end{bmatrix},\begin{bmatrix}\Delta_3 f_6\\ \delta_3 f_6\end{bmatrix},\ldots,\begin{bmatrix}\Delta_3 f_k\\ \delta_3 f_k\end{bmatrix},\ldots,\begin{bmatrix}\Delta_3 f_n\\ \delta_3 f_n\end{bmatrix}\right]+\\ &+\hat{P}_1\hat{P}_2\hat{P}_3\left[0,0,0,0,0,\begin{bmatrix}\Delta_4 f_6\\ \delta_4 f_6\end{bmatrix},\begin{bmatrix}\Delta_4 f_7\\ \delta_4 f_7\end{bmatrix},\ldots,\begin{bmatrix}\Delta_4 f_k\\ \delta_4 f_k\end{bmatrix},\ldots,\begin{bmatrix}\Delta_4 f_n\\ \delta_4 f_n\end{bmatrix}\right]+\\ &+\cdots+\hat{P}_1\hat{P}_2\cdots\hat{P}_{n-3}\left[0,0,0,0,0,0,\ldots,0,0,\begin{bmatrix}\Delta_{n-2} f_n\\ \delta_{n-2} f_n\end{bmatrix}\right].\end{aligned}$$

Taking norms, we obtain

$$\begin{aligned}\left\|\begin{bmatrix}\Delta F\\ \delta F\end{bmatrix}\right\|_F &\le O(m)\varepsilon\|F\|_F+\sum_{j=1}^{n-2}\sqrt{\sum_{k=j+2}^{n}\left\|\begin{bmatrix}\Delta_j f_k\\ \delta_j f_k\end{bmatrix}\right\|_2^2}\le\\ &\le O(m)\varepsilon\|F\|_F+\sqrt{2}\sum_{j=1}^{n-2}\|\hat{u}_j^T\begin{bmatrix}f_{j+2} & f_{j+3} & \cdots & f_n\end{bmatrix}\|_2\le\\ &\le O(m)\varepsilon\|F\|_F+\sqrt{2}\sum_{j=1}^{n-2}(\|\tilde{u}_j^T\begin{bmatrix}f_{j+2} & f_{j+3} & \cdots & f_n\end{bmatrix}\|_2+\\ &\qquad+\|\delta\hat{u}_j^T\begin{bmatrix}f_{j+2} & f_{j+3} & \cdots & f_n\end{bmatrix}\|_2)\le\\ &\le O(m)\varepsilon\|F\|_F+\sqrt{2}\sum_{j=1}^{n-2}(\|\tilde{u}_j^T\tilde{A}_{n-2}(:,j+2:n)\|_2+\\ &\quad+\|\delta\hat{u}_j\|_2\|F(:,j+2:n)\|_F).\end{aligned}\tag{3.12}$$

Step 3: Estimation of the backward error

It remains to estimate the products

$$\hat{u}_j^T f_k=\tilde{u}_j^T f_k-\delta\hat{u}_j^T f_k,\quad \text{for } j=1,\ldots,n-2,\ k=j+2,\ldots,n.$$

Since $|\delta\hat{u}_j^T f_k|\le O(m)\varepsilon\|f_k\|_2$, it remains to estimate the products $\tilde{u}_j^T f_k=\tilde{u}_j^T\tilde{A}_{j+\ell}(:,j+\ell+1)$, $\ell=1,\ldots,n-j-1$.

In exact arithmetic, $u_j \perp f_{j+2}, \ldots, f_n$ because the triorthogonalization of F implies

$$u_j \in \mathrm{span}\{f_1, \ldots, f_j\} \subset \mathrm{span}\{f_{j+2}, \ldots, f_n\}^{\perp}. \tag{3.13}$$

The situation in finite precision arithmetic is different, and relation (3.13) does not have to hold. This is the key point where the difference between the Ralha bidiagonalization and the Barlow bidiagonalization plays an important role. While the Ralha bidiagonalization relies only on relation (3.13), the Barlow bidiagonalization deals more explicitly with the scalar products $\tilde{u}_j^T f_k$ by using a different formula for the vectors $z_1, \ldots, z_{n-2}$.

The computed vector $\tilde{z}_j \in \mathbb{R}^{n-j}$ satisfies

$$\begin{aligned} \tilde{z}_j^T &= \mathrm{fl}(\tilde{u}_j^T \tilde{A}_{j-1}(:, j+1:n)) = \tilde{u}_j^T \tilde{A}_{j-1}(:, j+1:n) + \delta\tilde{z}_j^T, \\ \|\delta\tilde{z}_j\|_2 &\leq O(m)\varepsilon \|\tilde{A}_{j-1}(:, j+1:n)\|_2. \end{aligned} \tag{3.14}$$

This estimation follows from numerical analysis of the scalar product and the matrix–vector product, described in [47, pp. 68–78], where

$$|\delta\tilde{z}_j(i)| \leq O(m)\varepsilon \|\tilde{A}_{j-1}(:, j+i)\|_2 \|\tilde{u}_j\|_2 \leq O(m)\varepsilon \|\tilde{A}_{j-1}(:, j+i)\|_2, \quad i = 1, \ldots, n-j.$$

Let $\hat{V}_j = I - \hat{v}_j \hat{v}_j^T$, where $\hat{v}_j \in \mathbb{R}^{n-j}$, $\|\hat{v}_j\|_2 = \sqrt{2}$, be a Householder reflector such that

$$\tilde{z}_j^T \hat{V}_j = [\|\tilde{z}_j\|_2 \ 0 \ 0 \ \ldots \ 0], \tag{3.15}$$

and let $\tilde{v}_j = \hat{v}_j + \delta\hat{v}_j$ be the computed approximation of $\hat{v}_j$, where by Lemma 3.2.1 and Remark 3.2.2

$$|\delta\hat{v}_j(i)| \leq O(n-j)\varepsilon |\hat{v}_j(i)|,$$

and let $\tilde{V}_j = I - \tilde{v}_j \tilde{v}_j$. Note that $\hat{V}_j$ is exactly orthogonal, while $\tilde{V}_j$ is numerically orthogonal. In the algorithm, $\tilde{V}_j$ is used to compute

$$\begin{aligned} \tilde{A}_j(:, j+1:n) &= \mathrm{fl}(\tilde{A}_{j-1}(:, j+1:n)\tilde{V}_j) = \\ &= \tilde{A}_{j-1}(:, j+1:n)\hat{V}_j + E_j, \end{aligned} \tag{3.16}$$

where E_j contains the error from the floating–point application of $\tilde{V}_j$ plus the difference between $\tilde{V}_j$ and $\hat{V}_j$. Using the results from Lemma 3.2.4 we have

$$\|E_j\|_F \leq O(n-j)\varepsilon \|\tilde{A}_{j-1}(:, j+1:n)\|_F. \tag{3.17}$$

This implies that

$$\begin{aligned} \tilde{u}_j^T \tilde{A}_j(:, j+1:n) &= [\|\tilde{z}_j\| \ 0 \ 0 \ \ldots \ 0] - \delta\tilde{z}_j^T \hat{V}_j + \tilde{u}_j^T E_j, \\ \|\tilde{u}_j^T \tilde{A}_j(:, j+2:n)\|_2 &\leq \|\delta\tilde{z}_j\|_2 + \|\tilde{u}_j\|_2 \|E_j\|_F \leq O(m)\varepsilon \|\tilde{A}_{j-1}(:, j+1:n)\|_F. \end{aligned} \tag{3.18}$$

Recall that we need an estimate of $\tilde{u}_j^T f_k = \tilde{u}_j^T \tilde{A}_{j+\ell}(:, j+\ell+1)$, $\ell \geq 1$. Therefore, consider the next $\ell \geq 1$ right–handed Householder $(n-i) \times (n-i)$ transformations

$\hat{V}_i$, $\tilde{V}_i$, $i = j+1, \ldots, j+\ell$. Since we are interested in columns $j+2, \ldots, n$, we consider all these transformations as $(n-j-1)\times(n-j-1)$ by writing $\hat{\mathbf{V}}_i^{(j)} = I \oplus \hat{V}_i$, $\tilde{\mathbf{V}}_i^{(j)} = I \oplus \tilde{V}_i$. Similarly as in (3.16), we obtain

$$\begin{aligned} \tilde{A}_{j+\ell}(:, j+2:n) &= \mathrm{fl}(((\tilde{A}_j(:, j+2:n)\tilde{\mathbf{V}}_{j+1}^{(j)})\cdots)\tilde{\mathbf{V}}_{j+\ell}^{(j)}) = \\ &= \tilde{A}_j(:, j+2:n)\hat{\mathbf{V}}_{j+1}^{(j)}\cdots\hat{\mathbf{V}}_{j+\ell}^{(j)} + E_{j,\ell}, \qquad (3.19) \\ \text{where } \|E_{j,\ell}\|_F &\leq O(\ell n)\varepsilon\|\tilde{A}_j(:, j+2:n)\|_F. \end{aligned}$$

Now, relations (3.19) and (3.18) imply

$$\begin{aligned} \tilde{u}_j^T\tilde{A}_{j+\ell}(:, j+2:n) &= \tilde{u}_j^T\tilde{A}_j(:, j+2:n)\hat{\mathbf{V}}_{j+1}^{(j)}\cdots\hat{\mathbf{V}}_{j+\ell}^{(j)} + \tilde{u}_j^T E_{j,\ell} = \\ &= (-\delta\tilde{z}_j^T\hat{V}_j + \tilde{u}_j^T E_j)(2:n-j)\hat{\mathbf{V}}_{j+1}^{(j)}\cdots\hat{\mathbf{V}}_{j+\ell}^{(j)} + \tilde{u}_j^T E_{j,\ell}, \end{aligned}$$

and we conclude that the following bound holds:

$$\begin{aligned} \|\tilde{u}_j^T\tilde{A}_{j+\ell}(:, j+2:n)\|_2 &\leq \|\delta\tilde{z}_j\|_2 + (\|E_j\|_F + \|E_{j,\ell}\|_F)\|\tilde{u}_j\|_2 \leq \\ &\leq O(m)\varepsilon\|\tilde{A}_{j-1}(:, j+1:n)\|_F + \\ &\quad + O(n-j)\varepsilon\|\tilde{A}_{j-1}(:, j+1:n)\|_F + \\ &\quad + O(\ell n)\varepsilon\|\tilde{A}_j(:, j+2:n)\|_F \leq \\ &\leq O(m+n^2)\varepsilon\|\tilde{A}_{n-2}(:, j+1:n)\|_F. \end{aligned}$$

Now, for $k = j+\ell+1$, $\ell \geq 1$, we can write

$$\begin{aligned} |\tilde{u}_j^T f_k| &= |\tilde{u}_j^T f_{j+\ell+1}| = |\tilde{u}_j^T\tilde{A}_{j+\ell}(:, j+\ell+1)| \leq \\ &\leq O(m+n^2)\varepsilon\|\tilde{A}_{n-2}\|_F = O(m+n^2)\varepsilon\|F\|_F. \qquad (3.20) \end{aligned}$$

In fact, the whole vector $\tilde{u}_j^T[f_{j+2}, f_{j+3}, \ldots, f_n] = \tilde{u}_j^T\tilde{A}_{n-2}(:, j+2:n)$ can be estimated as

$$\|\tilde{u}_j^T\tilde{A}_{n-2}(:, j+2:n)\|_2 \leq O(m+n^2)\varepsilon\|F\|_F,$$

and then by (3.12) it follows

$$\left\| \begin{bmatrix} \Delta F \\ \delta F \end{bmatrix} \right\|_F \leq O(mn+n^3)\varepsilon\|F\|_F \leq O(mn+n^3)(1+\eta_F)\|A\|_F.$$

To get relation (3.7), we collect the perturbations from both the Householder implicit tridiagonalization and the Gram-Schmidt orthogonalization,

$$\begin{aligned} \begin{bmatrix} \tilde{B} \\ 0 \end{bmatrix} &= \hat{P}^T\left\{\begin{bmatrix} 0 \\ F \end{bmatrix} + \begin{bmatrix} \Delta F \\ \delta F \end{bmatrix}\right\} = \hat{P}^T\left\{\begin{bmatrix} 0 \\ A+\delta_1 A \end{bmatrix}\hat{V} + \begin{bmatrix} \Delta F \\ \delta F \end{bmatrix}\right\} \\ &= \hat{P}^T\left\{\begin{bmatrix} 0 \\ A+\delta_1 A \end{bmatrix} + \begin{bmatrix} \Delta F \\ \delta F \end{bmatrix}\hat{V}^T\right\}\hat{V} = \\ &= \hat{P}^T\begin{bmatrix} \Delta A \\ A+\delta A \end{bmatrix}\hat{V}. \end{aligned}$$

Step 4: The final result

Finally, by using $\hat{P}_{11} = \hat{P}(1:n, 1:n)$, $\hat{P}_{21} = \hat{P}(n+1:n+m, 1:n)$, we have

$$\begin{bmatrix} \Delta A \\ A + \delta A \end{bmatrix} = \begin{bmatrix} \hat{P}_{11} \\ \hat{P}_{21} \end{bmatrix} \tilde{B}\hat{V}^T, \quad \hat{P}_{11}^T\hat{P}_{11} + \hat{P}_{21}^T\hat{P}_{21} = I,$$

and relation (3.8) follows by Lemma 3.1 from [6, p. 181]. The proof that the relation (3.8) holds is also given in Theorem 3.18 [2].

Let us consider the CS decomposition [35, p. 77], [89, pp. 37–40] of the matrix $\hat{P}$

$$\hat{P} = \begin{bmatrix} \hat{P}_{11} & \hat{P}_{12} \\ \hat{P}_{21} & \hat{P}_{22} \end{bmatrix}, \qquad P_{11} \in \mathbb{R}^{n\times n}, P_{12} \in \mathbb{R}^{n\times m}, P_{21} \in \mathbb{R}^{m\times n}, P_{22} \in \mathbb{R}^{m\times m},$$

then there exist $(m+n)\times(m+n)$ orthogonal matrices

$$\hat{W} = \begin{bmatrix} \hat{W}_{11} & 0 & 0 \\ 0 & \hat{W}_{22} & \hat{W}_{23} \\ 0 & \hat{W}_{32} & \hat{W}_{33} \end{bmatrix}, \text{ and } \hat{Z} = \begin{bmatrix} \hat{Z}_{11} & 0 & 0 \\ 0 & \hat{Z}_{22} & \hat{Z}_{23} \\ 0 & \hat{Z}_{32} & \hat{Z}_{33} \end{bmatrix},$$

such that $\hat{W}_{11}, \hat{W}_{22}, \hat{Z}_{11}, \hat{Z}_{22} \in \mathbb{R}^{n\times n}$, and

$$\begin{bmatrix} \hat{P}_{11} & \hat{P}_{12} \\ \hat{P}_{21} & \hat{P}_{22} \end{bmatrix} = \begin{bmatrix} \hat{W}_{11} & 0 & 0 \\ 0 & \hat{W}_{22} & \hat{W}_{23} \\ 0 & \hat{W}_{32} & \hat{W}_{33} \end{bmatrix} \begin{bmatrix} C & -S & 0 \\ S & C & 0 \\ 0 & 0 & I_{m-n} \end{bmatrix} \begin{bmatrix} \hat{Z}_{11}^T & 0 & 0 \\ 0 & \hat{Z}_{22}^T & \hat{Z}_{32}^T \\ 0 & \hat{Z}_{23}^T & \hat{Z}_{33}^T \end{bmatrix}, \tag{3.21}$$

where

$$\begin{aligned} C &= \operatorname{diag}(c_1, \ldots, c_n), \qquad c_i \geq 0,\ i = 1, \ldots, n, \\ S &= \operatorname{diag}(s_1, \ldots, s_n), \qquad s_i \geq 0,\ i = 1, \ldots, n, \\ C^2 + S^2 &= I_n. \end{aligned}$$

Further, let us define the following matrices:

$$\hat{W}_1 = \hat{W}_{11}, \quad \hat{Z}_1 = \hat{Z}_{11}, \quad \hat{W}_2 = \begin{bmatrix} \hat{W}_{22} \\ \hat{W}_{32} \end{bmatrix},$$

where $\hat{W}_1 \in \mathbb{R}^{n\times n}$ and $\hat{Z}_1 \in \mathbb{R}^{n\times n}$ are orthogonal matrices, and $\hat{W}_2 \in \mathbb{R}^{m\times n}$ is an orthonormal matrix, then from (3.21) it follows that

$$\hat{P}_{11} = \hat{W}_1 C \hat{Z}_1^T, \qquad \hat{P}_{12} = \hat{W}_2 S \hat{Z}_1^T.$$

Finally, we define $\hat{U} = \hat{W}_2\hat{Z}_1$, as the closest orthonormal matrix to $\hat{P}_{21}$ in any unitarily invariant norm. Since $(I+S)(I-S) = C^2$, we have

$$\begin{aligned} \hat{U} - \hat{P}_{21} &= \hat{W}_2(I-S)\hat{Z}_1^T = \hat{W}_2(I+S)^\dagger \hat{Z}_1^T \hat{Z}_1 C \hat{W}_1^T \hat{W}_1 C \hat{Z}_1^T = \\ &= \hat{W}_2(I+S)^\dagger \hat{Z}_1^T \hat{P}_{11}^T \hat{P}_{11}, \\ (\hat{U} - \hat{P}_{21})\tilde{B}\hat{V}^T &= \hat{W}_2(I+S)^\dagger \hat{Z}_1^T \hat{P}_{11}^T \Delta A, \\ \delta\hat{A} &= \hat{U}\tilde{B}\hat{V}^T - A = (\hat{U} - \hat{P}_{21})\tilde{B}\hat{V}^T + \delta A, \end{aligned}$$

thus

$$\begin{aligned}\|\delta\hat{A}\|_F &\leq \|\hat{W}_2(I+S)^\dagger \hat{Z}_1^T \hat{P}_{11}^T\|_2 \|\Delta A\|_F + \|\delta A\|_F \leq \|\Delta A\|_F + \|\delta A\|_F \leq \\ &\leq \sqrt{2}\sqrt{\|\Delta A\|_F^2 + \|\delta A\|_F^2} \leq \sqrt{2}\xi\|A\|_F.\end{aligned}$$

□

Remark 3.2.7. *Barlow in [2] suggested an alternative algorithm for bidiagonalization. It is almost identical to Algorithm 3.1.1, except in the way the scalar ϕ_{k+1} is computed. This time it is*

$$\phi_{k+1}e_1 = V_k z_k, \qquad V_k = I - v_k v_k^T,$$

which is equivalent to Algorithm 3.1.1 in exact arithmetic, but not in finite precision arithmetic. Let us denote

$$\begin{aligned}\bar{\phi}_{k+1} &= e_1^T \,\mathrm{fl}(\tilde{V}_k \tilde{z}_k) \\ \bar{s}_{k+1} &= \mathrm{fl}(f_{k+1} - \bar{\phi}_{k+1}\tilde{u}_k)\end{aligned}$$

where $\tilde{V}_k = I - \tilde{v}_k\tilde{v}_k^T$, $\hat{V}_k = I - \hat{v}_k\hat{v}_k^T$ and $\tilde{v}_k = \hat{v}_k + \delta\hat{v}_k$. Then it follows

$$\begin{aligned}\bar{w}_k &= \mathrm{fl}(\tilde{V}_k\tilde{z}_k) = \hat{V}_k\tilde{z}_k + \delta_1\bar{w}_k = \hat{V}_k(\tilde{A}_{k-1}(:,k+1:n)^T\tilde{u}_k + \delta\tilde{z}_k) + \delta_1\bar{w}_k = \\ &= \hat{V}_k\tilde{A}_{k-1}(:,k+1:n)^T(\hat{u}_k + \delta\hat{u}_k) + \hat{V}_k\delta\tilde{z}_k + \delta_1\bar{w}_k = \\ &= \tilde{A}_k(:,k+1:n)^T\hat{u}_k - E_k^T\hat{u}_k + \hat{V}_k\tilde{A}_{k-1}(:,k+1:n)^T\delta\hat{u}_k + \\ &\quad + \hat{V}_k\delta\tilde{z}_k + \delta_1\bar{w}_k = \\ &= \tilde{A}_k(:,k+1:n)^T\hat{u}_k + \delta\bar{w}_k,\end{aligned}$$

where Lemma 3.2.3 implies the bound on $\|\delta_1\bar{w}_k\|_2$

$$\|\delta_1\bar{w}_k\|_2 \leq O(n-k)\varepsilon\|\tilde{z}_k\|_2 \leq O(n-k)\varepsilon\|\tilde{A}_{k-1}(:,k+1:n)\|_F \leq O(n-k)\varepsilon\|F\|_F,$$

from (3.14) in the proof of Theorem 3.2.6 it follows

$$\|\delta\tilde{z}_k\|_2 \leq O(m)\varepsilon\|\tilde{A}_{k-1}(:,k+1:n)\|_F \leq O(m)\varepsilon\|F\|_F,$$

according to (3.11) in the proof of Theorem 3.2.6 we have

$$\|\delta\hat{u}_k\|_2 \leq O(m)\varepsilon,$$

and by (3.17) we can estimate the application of a Householder reflector:

$$\|E_k\|_F \leq O(n-k)\varepsilon\|\tilde{A}_{k-1}(:,k+1:n)\|_F \leq O(n-k)\varepsilon\|F\|_F.$$

So, finally we can conclude

$$\bar{\phi}_{k+1} = f_{k+1}^T\hat{u}_k + \delta\bar{\phi}_{k+1}, \qquad |\bar{\phi}_{k+1}| \leq O(m)\varepsilon\|F\|_F.$$

On the other hand, for $\hat{\phi}_{k+1} = f_{k+1}^T\hat{u}_k$, like in the proof of Theorem 3.2.6 we can write

$$\bar{s}_{k+1} = f_{k+1} - \hat{\phi}_{k+1}\hat{u}_k + \delta\bar{s}_{k+1}, \qquad \|\delta\bar{s}_{k+1}\|_2 \leq O(m)\varepsilon\|F\|_F.$$

Further, this influences the following bounds:

$$\left\| \begin{bmatrix} \Delta f_{k+1} \\ \delta f_{k+1} \end{bmatrix} \right\|_2 \leq O(m)\varepsilon\|F\|_F$$

$$\left\| \begin{bmatrix} \Delta F \\ \delta F \end{bmatrix} \right\|_F \leq O(mn^{\frac{1}{2}})\varepsilon\|F\|_F + \sqrt{2}\sum_{j=1}^{n-2}(\|\tilde{u}_j^T \tilde{A}_{n-2}(:, j+2:n)\|_2 + \|\delta\hat{u}_j\|_2\|F(:, j+2:n)\|_F)$$

which will not change the final bound in Theorem 3.2.6.

Instead of Householder reflectors we can use the Givens rotations in Algorithm 3.1.1. Recall that Givens rotation is a plane rotation

$$G_{i,j}(\phi) = \begin{bmatrix}
1 & & & \vdots & & & & \vdots & & & & \\
& \ddots & & \vdots & & & & \vdots & & 0 & & \\
& & 1 & \vdots & & & & \vdots & & & & \\
\cdots & \cdots & \cdots & \cos(\phi) & \cdots & \cdots & \cdots & \sin(\phi) & \cdots & \cdots & \cdots & i \\
& & & \vdots & 1 & & & \vdots & & & & \\
& & & \vdots & & \ddots & \vdots & & & & & \\
& & & \vdots & & & 1 & \vdots & & & & \\
\cdots & \cdots & \cdots & -\sin(\phi) & \cdots & \cdots & \cdots & \cos(\phi) & \cdots & \cdots & \cdots & j \\
& & & \vdots & & & & \vdots & 1 & & & \\
& 0 & & \vdots & & & & \vdots & & \ddots & & \\
& & & \vdots & & & & \vdots & & & 1 & \\
& & & i & & & & j & & & &
\end{bmatrix}.$$

We can also prove that this variant of one–sided bidiagonalization is numerically stable, but we need the following auxiliary results.

Lemma 3.2.8 ([47, p. 373]). *Let a Givens rotation $G_{i,j}(\phi)$ be constructed according to*

$$c = \cos(\phi) = \frac{x_i}{\sqrt{x_i^2 + x_j^2}}, \quad s = \sin(\phi) = \frac{x_j}{\sqrt{x_i^2 + x_j^2}}.$$

The computed $\tilde{c}$ and $\tilde{s}$ satisfy

$$\tilde{c} = c(1 + \theta_4), \quad \tilde{s} = s(1 + \theta_4'), \tag{3.22}$$

where $|\theta_4|, |\theta_4'| \leq 4\varepsilon + O(\varepsilon^2)$.

Lemma 3.2.9 ([47, p. 373]). *Let $x \in \mathbb{R}^m$ and consider the computation of $y = \tilde{G}_{i,j}x$, where $\tilde{G}_{i,j}$ is a computed Givens rotation in the (i, j) plane for which $\tilde{c}$ and $\tilde{s}$ satisfy (3.22). The computed $\tilde{y}$ satisfies*

$$\tilde{y} = (\hat{G}_{i,j} + \delta\hat{G}_{i,j})x, \qquad \|\delta\hat{G}_{i,j}\|_F \leq 6\sqrt{2}\varepsilon + O(\varepsilon^2),$$

where $\hat{G}_{i,j}$ is an exact Givens rotation based on $\hat{c}$ and $\hat{s}$ defined in Lemma 3.2.8. All the rows of $\delta\hat{G}_{i,j}$ except the i-th and j-th are zero.

Lemma 3.2.10 ([47, pp. 374–375]). *Consider the sequence of transformations*

$$A_{k+1} = W_k A_k, \qquad k = 1, \ldots, r,$$

where $A_1 = A \in \mathbb{R}^{m\times n}$ and each W_k is a product of disjoint Givens rotations. Assume that the individual Givens rotations are performed using computed sine and cosine values related to the exact values defining W_k by (3.22). Then the computed matrix $\tilde{A}_{r+1}$ satisfies

$$\tilde{A}_{r+1} = \hat{Q}^T(A + \delta A),$$

where $\hat{Q}^T = \hat{W}_r\hat{W}_{r-1}\cdots\hat{W}_1$ and δA satisfies the normwise and componentwise bounds

$$\begin{aligned} \|\delta A\|_F &\leq O(r)\varepsilon\|A\|_2, \\ |\delta A| &\leq O(rm)\varepsilon G|A|, \qquad \|G\|_F = 1. \end{aligned}$$

(In fact we can take $G = m^{-1}ee^T$, where $e = [1, 1, \ldots, 1]^T$.) In the special case $n = 1$, so that $A = a$, we have $\tilde{a}_{r+1} = (\hat{Q} + \delta\hat{Q})^T a$ with $\|\delta\hat{Q}\|_F \leq O(r)\varepsilon$.

Theorem 3.2.11 ([47, p. 375]). *Let $\tilde{R} \in \mathbb{R}^{m\times n}$ be the computed upper trapezoidal QR factor of $A \in \mathbb{R}^{m\times n}$ $(m \geq n)$ obtained via the Givens QR algorithm, with any standard choice and ordering of rotations. Then there exists an orthogonal $\hat{Q} \in \mathbb{R}^{m\times m}$ such that*

$$A + \delta A = \hat{Q}\tilde{R},$$

with $\|\delta A\|_F \leq O(m+n)\varepsilon\|A\|_F$ and $|\delta A| \leq O(m+n)m\varepsilon G|A|$, $\|G\|_F = 1$. (The matrix $\hat{Q}$ is a product of Givens rotations, the k-th of which corresponds to the exact application of the k-th step of the algorithm to $\tilde{A}_k$.)

The numerical analysis result on the Givens Barlow bidiagonalization can be found in the following corollary.

Corollary 3.2.12. *If in Algorithm 3.1.1 Householder reflectors are replaced by Givens rotations, then the results of Theorem 3.2.6 hold for $0 \leq \xi \leq O(mn + n^2)\varepsilon$.*

Proof. We just have to go through the proof of Theorem 3.2.6, and change the statements concerning Householder reflectors.

Step 1:

In finite precision arithmetic, the computed matrix F is obtained as

$$F = \tilde{A}_{n-2} = \mathrm{fl}((\cdots((A \cdot \tilde{\mathbf{G}}_1) \cdot \tilde{\mathbf{G}}_2)\cdots) \cdot \tilde{\mathbf{G}}_r),$$

where each $\tilde{\mathbf{G}}_k$, $k = 1, \ldots, r$ is a product of disjoint computed Givens rotations and $r = O(n)$. By Lemma 3.2.10 there exists an exactly orthogonal matrix $\hat{V} = \hat{\mathbf{G}}_1 \cdots \hat{\mathbf{G}}_r$, such that

$$F = (A + \delta_1 A)\hat{V}, \qquad \|\delta_1 A\|_F \leq \eta_F\|A\|_F, \ \eta_F \leq O(n)\varepsilon. \tag{3.23}$$

Step 2:

Remains unchanged.

Step 3:

Let $\hat{V}_j$ be the product of $n-j-1$ nondisjoint exact Givens rotations such that (3.15) holds. Then by Lemma 3.2.10 equation (3.16) holds with

$$\|E_j\|_F \leq O(n-j-1)\varepsilon\|\tilde{A}_{j-1}(:, j+1:n)\|_F.$$

Further, the equation (3.19) also holds, but the matrices $\hat{\mathbf{V}}_{j+1}^{(j)}, \ldots, \hat{\mathbf{V}}_{j+\ell}^{(j)}$ represent $\ell \leq r$ different products of remaining disjoint Givens rotations, which are restricted to the bottom-right $(n-j-1)\times(n-j-1)$ block. Thus

$$\|E_{j,\ell}\|_F \leq O(n)\varepsilon\|\tilde{A}_j(:, j+2:n)\|_F.$$

Finally, it follows that

$$\|\tilde{u}_j^T \tilde{A}_{j+\ell}(:, j+2:n)\|_2 \leq O(m+n)\varepsilon\|\tilde{A}_{n-2}(:, j+1:n)\|_F \leq O(m+n)\varepsilon\|F\|_F,$$

and

$$\left\| \begin{bmatrix} \Delta F \\ \delta F \end{bmatrix} \right\|_F \leq O(mn+n^2)\varepsilon\|F\|_F \leq O(mn+n^2)(1+\eta_F)\|A\|_F.$$

The final result is straightforward.

Step 4:

Remains unchanged.

□

The results of numerical analysis stated so far are dealing with the backward error of Algorithm 3.1.1. That means that we can estimate a perturbation δA of the matrix A, so that computed matrix $\tilde{B}$ is the exact bidiagonal factor of the matrix $A+\delta A$. Now we will examine the forward error, which comprises the distance between the singular value of A and the corresponding singular value of $\tilde{B}$. We will use a standard perturbation result from Theorem 2.3.1 to estimate the error.

Corollary 3.2.13. *If $\sigma_1 \geq \cdots \geq \sigma_n$ are the singular values of A, then the singular values $\tilde{\sigma}_1 \geq \cdots \geq \tilde{\sigma}_n$ of $\tilde{B}$ from Theorem 3.2.6 satisfy*

$$\max_i \frac{|\tilde{\sigma}_i - \sigma_i|}{\|A\|_F} \leq \sqrt{2}\xi.$$

Proof. If we use the fact that $\|\delta\hat{A}\|_2 \leq \|\delta\hat{A}\|_F$, and combine it with (3.8) from Theorem 3.2.6 and Theorem 2.3.1, we will obtain the result. □

The computed $\tilde{U}$ is not guaranteed to be numerically orthogonal, as was the case with the Ralha bidiagonalization. We can prove a similar result as Björck and Paige in [6] for the modified Gram–Schmidt orthogonalization.

Corollary 3.2.14. *If $\tilde{U}$ is computed by Algorithm 3.1.1 in finite precision arithmetic, and if the matrices $A \in \mathbb{R}^{m\times n}$ and $\tilde{B} \in \mathbb{R}^{n\times n}$ have full rank, then the following estimations hold*

$$\begin{aligned} \|\tilde{U} - \hat{U}\|_F &\leq O(mn+n^3)\varepsilon\kappa_F(\tilde{B}) = O(mn+n^3)\varepsilon\kappa_F(A) \\ \|\tilde{U}^T\tilde{U} - I\|_F &\leq O(mn+n^3)\varepsilon\kappa_F(\tilde{B}) = O(mn+n^3)\varepsilon\kappa_F(A), \end{aligned}$$

where $\kappa_F(A) = \|A\|_F\|A^\dagger\|_F$, and $\hat{U}$ is orthonormal matrix from Theorem 3.2.6.

Proof. First we will follow the steps of the algorithm for obtaining $\tilde{u}_{k+1}$. From [47] we have

$$\begin{aligned} \tilde{\phi}_{k+1} &= \tilde{u}_k^T f_{k+1} + \delta\tilde{\phi}_{k+1}, \quad |\delta\tilde{\phi}_{k+1}| \leq O(m)\varepsilon\|f_{k+1}\|_2 \\ \tilde{s}_{k+1} &= f_{k+1} - \tilde{\phi}_{k+1}\tilde{u}_k + \delta\tilde{s}_{k+1}, \quad \|\delta\tilde{s}_{k+1}\|_2 \leq O(1)\varepsilon\|f_{k+1}\|_2 \\ \tilde{\psi}_{k+1} &= \|\tilde{s}_{k+1}\|_2 + \delta\tilde{\psi}_{k+1}, \quad |\delta\tilde{\psi}_{k+1}| \leq O(m)\varepsilon\|f_{k+1}\|_2 \\ \tilde{u}_{k+1} &= \frac{f_{k+1} - \tilde{\phi}_{k+1}\tilde{u}_k + \delta\tilde{s}_{k+1}}{\tilde{\psi}_{k+1}} + \delta\tilde{u}_{k+1}, \quad \|\delta\tilde{u}_{k+1}\|_2 \leq O(1)\varepsilon \end{aligned}$$

which implies

$$\tilde{\phi}_{k+1}\tilde{u}_k + \tilde{\psi}_{k+1}\tilde{u}_{k+1} = f_{k+1} + \delta\tilde{s}_{k+1} + \tilde{\psi}_{k+1}\delta\tilde{u}_{k+1} = f_{k+1} + \delta\tilde{f}_{k+1},$$

with

$$\|\delta\tilde{f}_{k+1}\|_2 \leq O(1)\varepsilon\|f_{k+1}\|_2.$$

If we define $\delta\tilde{F} = [\delta\tilde{f}_1, \ldots, \delta\tilde{f}_n]$, than we can write

$$\tilde{U}\tilde{B} = F + \delta\tilde{F}, \qquad \|\delta\tilde{F}\|_F \leq O(1)\varepsilon\|F\|_F \leq O(1)\varepsilon\|A\|_F,$$

and by (3.10) it is

$$\tilde{U}\tilde{B} = (A + \delta_1 A)\hat{V} + \delta\tilde{F}.$$

If the matrix $\tilde{B}$ is nonsingular then

$$\tilde{U} = (A + \delta_1 A + \delta\tilde{F}\hat{V}^T)\hat{V}\tilde{B}^{-1}.$$

On the other hand, from (3.8) it follows that

$$\hat{U} = (A + \delta\hat{A})\hat{V}\tilde{B}^{-1}.$$

Putting all this together we obtain

$$\tilde{U} - \hat{U} = (\delta_1 A + \delta\tilde{F}\hat{V}^T - \delta\hat{A})\hat{V}\tilde{B}^{-1},$$

with

$$\|\tilde{U} - \hat{U}\|_F \leq O(mn+n^3)\varepsilon\kappa_F(\tilde{B}).$$

The second bound follows immediately by

$$\tilde{U}^T\tilde{U} - I = \tilde{U}^T\tilde{U} - \hat{U}^T\hat{U} = \hat{U}^T(\tilde{U} - \hat{U}) + (\tilde{U} - \hat{U})^T\hat{U} + (\tilde{U} - \hat{U})^T(\tilde{U} - \hat{U}),$$

and thus

$$\|\tilde{U}^T\tilde{U} - I\|_F \leq O(mn+n^3)\varepsilon\kappa_F(\tilde{B}).$$

□

Remark 3.2.15. *The result of Corollary 3.2.14 also holds for the alternative choice* $\phi_{k+1} = \gamma_k$.

The result of Corollary 3.2.14 shows that if A is ill conditioned, $\tilde{U}$ can be far from an orthonormal matrix.

Example 3.2.16. *Let* $A \in \mathbb{R}^{4\times 4}$ *be the matrix with* $\sigma_1 = 10^{10}$, $\sigma_2 = 1$, $\sigma_3 = 10^{-10}$ *and* $\sigma_4 = 10^{-20}$. *The matrix* A *is obtained as a product* $A = U\Sigma V^T$, *where* $\Sigma = \mathrm{diag}(\sigma_1, \sigma_2, \sigma_3, \sigma_4)$, *and* U *and* V *are computed as the QR factors of random matrices. The* MATLAB *program was used for the computation. The Barlow algorithm produced the following result:*

$$\tilde{B} = \begin{bmatrix} 5.7114 \cdot 10^{9} & -8.2086 \cdot 10^{9} & 0 & 0 \\ 0 & 5.3389 \cdot 10^{-1} & 9.5238 \cdot 10^{-1} & 0 \\ 0 & 0 & 5.1614 \cdot 10^{-6} & 2.2030 \cdot 10^{-7} \\ 0 & 0 & 0 & 2.2629 \cdot 10^{-7} \end{bmatrix},$$

$$\tilde{U} = \begin{bmatrix} 7.6056 \cdot 10^{-1} & -1.3535 \cdot 10^{-1} & -5.4709 \cdot 10^{-1} & -7.3738 \cdot 10^{-1} \\ 1.8502 \cdot 10^{-1} & -8.1511 \cdot 10^{-1} & -3.2165 \cdot 10^{-1} & 4.6023 \cdot 10^{-1} \\ 4.8577 \cdot 10^{-1} & 7.5365 \cdot 10^{-2} & -5.6628 \cdot 10^{-1} & -8.1180 \cdot 10^{-3} \\ 3.8902 \cdot 10^{-1} & 5.5820 \cdot 10^{-1} & -5.2589 \cdot 10^{-1} & 4.9436 \cdot 10^{-1} \end{bmatrix},$$

$$\tilde{V} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -6.8197 \cdot 10^{-1} & -2.0485 \cdot 10^{-2} & -7.3109 \cdot 10^{-1} \\ 0 & -3.0515 \cdot 10^{-1} & -9.0048 \cdot 10^{-1} & 3.0988 \cdot 10^{-1} \\ 0 & -6.6468 \cdot 10^{-1} & 4.3442 \cdot 10^{-1} & 6.0785 \cdot 10^{-1} \end{bmatrix}.$$

The singular values of the matrix $\tilde{B}$ *are*

$$\tilde{\Sigma}_B = \mathrm{diag}(10^{10}, 1.000000036943297, 1.589504011049195 \cdot 10^{-6}, 2.240629962266331 \cdot 10^{-7}),$$

which is consistent with Theorem 3.2.6 and Corollary 3.2.13, since

$$(mn + n^3)\varepsilon \|A\|_F = 1.776356839400251 \cdot 10^{-4}.$$

On the other hand

$$\|\tilde{U}^T\tilde{U} - I\|_F = 9.975416565064253 \cdot 10^{-1},$$

which shows that $\tilde{U}$ *is far from being orthogonal. It is also consistent with Corollary 3.2.14, since*

$$(mn + n^3)\varepsilon\kappa_F(A) = 1.322740457878272 \cdot 10^{4}.$$

It is still possible to recover a nearby orthonormal basis for some subspace of $\mathrm{span}\{\tilde{U}\}$, as it was shown in Corollary 3.20 in [2]. Let us assume that the singular value decomposition of the matrix $\tilde{B}$ is obtained in finite precision arithmetic, and that the decomposition satisfies

$$\tilde{B} + \delta\tilde{B} = \tilde{U}_B\tilde{\Sigma}\tilde{V}_B^T, \quad \|\delta\tilde{B}\|_F \leq g(n)\varepsilon\|B\|_F, \tag{3.24}$$

for some modestly growing function $g(n)$. Note that $\tilde{U}_B$ and $\tilde{V}_B$ are not exactly orthogonal, but they are numerically orthogonal:

$$\|\tilde{U}_B^T\tilde{U}_B - I\|_2, \|\tilde{V}_B^T\tilde{V}_B - I\|_2 = O(\varepsilon).$$

This departure from orthogonality has no qualitative effect on the analysis in the next corollary, so we will assume that $\tilde{U}_B = \hat{U}_B$ and $\tilde{V}_B = \hat{V}_B$ are orthogonal.

Next, we define $\tilde{Y} = \tilde{U}\tilde{U}_B$ and $\hat{Y} = \hat{U}\tilde{U}_B$, and then we take a partition of $\tilde{\Sigma}$, $\tilde{Y}$, $\hat{Y}$, and $\tilde{V}_B$ as follows

$$\tilde{\Sigma} = \begin{array}{c} \\ k \\ n-k \end{array} \begin{array}{c} \begin{array}{cc} k & n-k \end{array} \\ \begin{bmatrix} \tilde{\Sigma}_1 & 0 \\ 0 & \tilde{\Sigma}_2 \end{bmatrix} \end{array},$$

$$\tilde{Y} = \underset{k}{[\ \tilde{Y}_1} \quad \underset{n-k}{\tilde{Y}_2\ ]}, \quad \hat{Y} = \underset{k}{[\ \hat{Y}_1} \quad \underset{n-k}{\hat{Y}_2\ ]}, \quad \tilde{V}_B = \underset{k}{[\ \tilde{V}_{B,1}} \quad \underset{n-k}{\tilde{V}_{B,2}\ ]} \tag{3.25}$$

where $\tilde{\Sigma}_1$ is well conditioned. Here $\tilde{Y}_1$ is computed and $\hat{Y}_1$ is the exact orthonormal basis for a subspace that approximates the left singular subspace associated with the leading singular values.

Corollary 3.2.17. *If $\tilde{U}$ is computed by Algorithm 3.1.1 in finite precision arithmetic, and if $\tilde{Y}$ and $\hat{Y}$ are defined as above, then*

$$\|\tilde{Y}_1^T\tilde{Y}_1 - I\|_F \leq O(mn^{3/2} + n^{5/2})\varepsilon\frac{\tilde{\sigma}_1}{\tilde{\sigma}_k} + g(n)O(mn^{3/2} + n^{5/2})\varepsilon^2\kappa_F(A)\frac{\tilde{\sigma}_1}{\tilde{\sigma}_k}. \tag{3.26}$$

Proof. From the proof of Corollary 3.2.14 and (3.24) we have

$$\begin{aligned} \tilde{U}\tilde{B} &= (A + \delta_1 A)\hat{V} + \delta\tilde{F} \\ \tilde{U}(\tilde{U}_B\tilde{\Sigma}\tilde{V}_B^T - \delta\tilde{B}) &= (A + \delta_1 A)\hat{V} + \delta\tilde{F} \\ \tilde{Y}\tilde{\Sigma}\tilde{V}_B^T &= (A + \delta_1 A)\hat{V} + \delta\tilde{F} + \tilde{U}\delta\tilde{B} \end{aligned} \tag{3.27}$$

and

$$\begin{aligned} \hat{U}\tilde{B} &= (A + \delta\hat{A})\hat{V} \\ \hat{U}(\tilde{U}_B\tilde{\Sigma}\tilde{V}_B^T - \delta\tilde{B}) &= (A + \delta\hat{A})\hat{V} \\ \hat{Y}\tilde{\Sigma}\tilde{V}_B^T &= (A + \delta\hat{A})\hat{V} + \hat{U}\delta\tilde{B} \end{aligned} \tag{3.28}$$

Subtracting (3.28) from (3.27) we obtain

$$(\tilde{Y} - \hat{Y})\tilde{\Sigma}\tilde{V}_B^T = (\delta_1 A - \delta\hat{A})\hat{V} + \delta\tilde{F} + (\tilde{U} - \hat{U})\delta\tilde{B} \tag{3.29}$$

and by multiplying both sides of (3.29) with $\tilde{V}_{B,1}\tilde{\Sigma}_1^{-1}$ we get

$$\tilde{Y}_1 - \hat{Y}_1 = (\delta_1 A - \delta\hat{A})\hat{V}\tilde{V}_{B,1}\tilde{\Sigma}_1^{-1} + \delta\tilde{F}\tilde{V}_{B,1}\tilde{\Sigma}_1^{-1} + (\tilde{U} - \hat{U})\delta\tilde{B}\tilde{V}_{B,1}\tilde{\Sigma}_1^{-1}.$$

Hence, from proofs of Corollary 3.2.14 and Theorem 3.2.6 it follows

$$\begin{aligned}\|\tilde{Y}_1 - \hat{Y}_1\|_F &\leq [O(n^2) + O(mn+n^3)]\varepsilon\|A\|_F\tilde{\sigma}_k^{-1} + O(1)\varepsilon\|A\|_F\tilde{\sigma}_k^{-1} + \\ &\quad + g(n)O(mn+n^3)\varepsilon^2\kappa_F(A)\|B\|_F\tilde{\sigma}_k^{-1} \leq \\ &\leq O(mn^{3/2}+n^{5/2})\varepsilon\frac{\tilde{\sigma}_1}{\tilde{\sigma}_k} + g(n)O(mn^{3/2}+n^{5/2})\varepsilon^2\kappa_F(A)\frac{\tilde{\sigma}_1}{\tilde{\sigma}_k}\end{aligned}$$

The proof of (3.26) follows in the same way as the derivation of the bound on $\|\tilde{U}^T\tilde{U} - I\|_F$ in the proof of Corollary 3.2.14. □

So, we can conclude that a reasonable algorithm for computing the bidiagonal SVD of the matrix $\tilde{B}$ will lead to the construction of a numerically orthogonal basis for the left singular subspace associated with the leading singular values of A.

In spite of the possible loss of orthogonality of the matrix $\tilde{U}$, we can still prove the following proposition.

Proposition 3.2.18. *If $F = fl(AV)$ is obtained from Algorithm 3.1.1, then*

1. *F^TF is almost tridiagonal, with*

$$\begin{aligned}G_F &= F^TF - \mathrm{diag}(F^TF) - \mathrm{diag}(F^TF,-1) - \mathrm{diag}(F^TF,1), \\ \|G_F\|_F &\leq O(mn^{1/2}+n^{5/2})\varepsilon\|F\|_F^2 \qquad (3.30)\end{aligned}$$

   where $\mathrm{diag}(F^TF)$ *denotes the main diagonal of* F^TF, $\mathrm{diag}(F^TF,-1)$ *the subdiagonal, and* $\mathrm{diag}(F^TF,1)$ *the superdiagonal.*

2. *If $F = QR$ is the QR factorization of $F = [f_1 \ \dots \ f_n]$ and $R = [r_{ij}]$, then R is almost bidiagonal, with*

$$\begin{aligned}|r_{ij}| &\leq O(m+n^2)\varepsilon\zeta_i(F)\|F\|_F \qquad (3.31) \\ \zeta_i(F) &= \frac{\|f_2\|_2\cdots\|f_i\|_2}{r_{22}\cdots r_{ii}}, \quad i \leq j-2.\end{aligned}$$

   If the QR factorization is performed after column permutation, such that the most linear independent columns are brought to the first positions, then $\zeta_i(F)$ would be minimized.

Proof. First we will observe the size of $f_i^Tf_j$ where $i \leq j-2$. From Algorithm 3.1.1 and the proof of Theorem 3.2.6 we can describe a process of computing $\tilde{u}_i$, and use this information for estimating $f_i^Tf_j$.

$$\begin{aligned}\tilde{u}_i &= \mathrm{fl}\left(\frac{\tilde{s}_i}{\|\tilde{s}_i\|_2}\right) = \frac{\tilde{s}_i}{\|\tilde{s}_i\|_2} + \delta\hat{u}_i = \\ &= \frac{f_i - (f_i^T\hat{u}_{i-1})\hat{u}_{i-1} + \delta\tilde{s}_i}{\|f_i - (f_i^T\hat{u}_{i-1})\hat{u}_{i-1} + \delta\tilde{s}_i\|_2} + \delta\hat{u}_i,\end{aligned}$$

where

$$\begin{aligned} \hat{u}_i &= \frac{\tilde{s}_i}{\|\tilde{s}_i\|_2} \\ \|\delta\hat{u}_i\|_2 &\leq O(m)\varepsilon \\ \|\delta\tilde{s}_i\|_2 &\leq O(m)\varepsilon\|f_i\|_2 \end{aligned}$$

Now we can conclude that

$$\begin{aligned} f_i &= (f_i^T\hat{u}_{i-1})\hat{u}_{i-1} + \|f_i - (f_i^T\hat{u}_{i-1})\hat{u}_{i-1} + \delta\tilde{s}_i\|_2\tilde{u}_i - \delta\tilde{s}_i - \\ &\quad -\|f_i - (f_i^T\hat{u}_{i-1})\hat{u}_{i-1} + \delta\tilde{s}_i\|_2\delta\hat{u}_i \\ f_j^T f_i &= (f_i^T\hat{u}_{i-1})f_j^T\hat{u}_{i-1} + \|f_i - (f_i^T\hat{u}_{i-1})\hat{u}_{i-1} + \delta\tilde{s}_i\|_2 f_j^T\tilde{u}_i - f_j^T\delta\tilde{s}_i - \\ &\quad -\|f_i - (f_i^T\hat{u}_{i-1})\hat{u}_{i-1} + \delta\tilde{s}_i\|_2 f_j^T\delta\hat{u}_i. \end{aligned}$$

By the estimation

$$\|f_i - (f_i^T\hat{u}_{i-1})\hat{u}_{i-1} + \delta\tilde{s}_i\|_2 \leq (2 + O(m)\varepsilon)\|f_i\|_2 = O(1)\|f_i\|_2,$$

and relation (3.20) from the proof of Theorem 3.2.6, which reads

$$|f_j^T\tilde{u}_i|, |f_j^T\hat{u}_{i-1}| \leq O(m+n^2)\varepsilon\|F\|_F,$$

we can give a bound on $|f_j^T f_i|$, for $i \leq j-2$

$$\begin{aligned} |f_j^T f_i| &\leq |f_j^T\hat{u}_{i-1}|\|f_i\|_2 + O(1)|f_j^T\tilde{u}_i|\|f_i\|_2 + O(m)\varepsilon\|f_j\|_2\|f_i\|_2 + \\ &\quad +O(m)\varepsilon\|f_j\|_2\|f_i\|_2 \leq \\ &\leq [O(m+n^2)\varepsilon\|F\|_F + O(m)\varepsilon\|f_j\|_2]\|f_i\|_2 \leq \\ &\leq O(m+n^2)\varepsilon\|F\|_F\|f_i\|_2 \end{aligned}$$

Now we can derive the bound in (3.30).

$$\begin{aligned} \|G_F\|_F &= \sqrt{\sum_{i=1}^{n}\sum_{\substack{j=1 \\ |j-i|\geq 2}}^{n} |f_i^T f_j|^2} \leq \sqrt{\sum_{i=1}^{n}\sum_{\substack{j=1 \\ |j-i|\geq 2}}^{n} O(m+n^2)^2\varepsilon^2\|F\|_F^2\|f_i\|_2^2} \leq \\ &\leq O(m+n^2)\varepsilon\|F\|_F\sqrt{\sum_{i=1}^{n}\sum_{\substack{j=1 \\ |j-i|\geq 2}}^{n}\|f_i\|_2^2} \leq O(mn^{1/2}+n^{5/2})\varepsilon\|F\|_F^2. \end{aligned}$$

Suppose that $F = QR$ is an exact QR factorization of the matrix F, obtained by The Gram–Schmidt orthogonalization. Then, $r_{ij} = q_i^T f_j$ for $i \leq j$. We are interested in $|r_{ij}|$ for $i \leq j-2$, to give an estimation on how far the matrix R is from being bidiagonal. The analysis will be conducted by using mathematical induction. Let us start with the first row, $i = 1$. Then for Algorithm 3.1.1 we have

$$|r_{1j}| = |q_1^T f_j| = \frac{|f_1^T f_j|}{\|f_1\|_2} \leq O(m+n^2)\varepsilon\|F\|_F, \quad j = 3, \ldots, n.$$

To get an idea what is happening in other rows, we will perform the same analysis for the second row, $i = 2$.

$$\begin{aligned} |r_{2j}| &= |q_2^T f_j| = \frac{|f_2^T f_j - (q_1^T f_2) q_1^T f_j|}{\|f_2 - (q_1^T f_2) q_1\|_2} \leq \frac{|f_2^T f_j| + |q_1^T f_2||q_1^T f_j|}{\|f_2 - (q_1^T f_2) q_1\|_2} \leq \\ &\leq O(m+n^2)\varepsilon \|F\|_F \frac{\|f_2\|_2}{r_{22}}, \qquad j = 4, \ldots, n. \end{aligned}$$

As an assumption of induction we will take that

$$|r_{kj}| = |q_k^T f_j| \leq O(m+n^2)\varepsilon \|F\|_F \frac{\|f_2\|_2 \cdots \|f_k\|_2}{r_{22} \cdots r_{kk}}, \quad k = 2, \ldots, i-1, \ j = k+2, \ldots, n,$$

and we will prove that this assumption is valid for $k = i$. We have

$$\begin{aligned} |r_{ij}| &= |q_i^T f_j| = \frac{|f_i^T f_j - (q_{i-1}^T f_i) q_{i-1}^T f_j - \cdots - (q_1^T f_i) q_1^T f_j|}{\|f_i - (q_{i-1}^T f_i) q_{i-1} - \cdots - (q_1^T f_i) q_1\|_2} \leq \\ &\leq \frac{|f_i^T f_j| + |q_{i-1}^T f_j| \|f_i\|_2 + \cdots + |q_1^T f_j| \|f_i\|_2}{r_{ii}} \leq \\ &\leq O(m+n^2)\varepsilon \|F\|_F \left(\frac{\|f_i\|_2}{r_{ii}} + \frac{\|f_2\|_2 \cdots \|f_{i-1}\|_2}{r_{22} \cdots r_{i-1,i-1}} \frac{\|f_i\|_2}{r_{ii}} + \cdots + \frac{\|f_i\|_2}{r_{ii}} \right) \leq \\ &\leq O(m+n^2)\varepsilon \|F\|_F \frac{\|f_2\|_2 \cdots \|f_i\|_2}{r_{22} \cdots r_{ii}}, \quad j = i+2, \ldots, n, \end{aligned}$$

because $r_{kk} \leq \|f_k\|_2$. □

The scalar $\zeta_i(F)$ represents a condition number for producing a bidiagonal matrix out of the computed matrix F, where $F^T F$ is almost tridiagonal. Thus, if the matrix F has a small condition number, and if we compute the Householder QR factorization of F and replace $\tilde{U}$ with $\tilde{Q}$, and nontrivial elements of $\tilde{B}$ with corresponding diagonal and superdiagonal elements of $\tilde{R}$, then we could make an error comparable to the backward error of the bidiagonalization. Moreover, the computed matrix $\tilde{Q}$ would be numerically orthogonal.

3.3 Applications of the One–sided Algorithm

The matrix $\tilde{U} = [\tilde{u}_1, \ldots, \tilde{u}_n]$ may not be numerically orthogonal in case of the ill conditioned matrix A. As shown in Corollary 3.2.14, in Example 3.2.16 and in [2], $\tilde{U}$ can be far from orthogonal, it can be even numerically singular. If we want to use computed matrices $\tilde{U}$, $\tilde{\Sigma}$ and $\tilde{V}$ as SVD factors, then by Theorem 3.2.6 and Corollary 3.2.13 $\tilde{V}$ is acceptable as a matrix of right singular vectors, and the diagonal of $\tilde{\Sigma}$ is acceptable as singular values. On the other hand, large departure from orthogonality of $\tilde{U}$ cannot be tolerated, and columns of $\tilde{U}$ may not represent left singular vectors very well. Thus, the explicit usage of $\tilde{U}$ is not advisable. There are two possibilities how we can circumvent the problem of nonorthogonality. One way is to apply some sort of reorthogonalization,

where $\tilde{U}$ is transformed into a numerically orthogonal matrix. This would fix the numerical problem, but on the other hand, the algorithm would be less efficient. The main advantage of the one–sided bidiagonalization is that it has less floating point operations than the standard Householder bidiagonalization, and this would be undermined by the reorthogonalization. The other way is a modification of the problem we want to solve. It is seldom in practice that the SVD is the final target of the computation. More often, the SVD is a computational tool used to analyze and solve some other problem.

In this thesis, it will be shown that the possible loss of orthogonality is not always damaging as it may be expected, if the particular application of the SVD is properly formulated. Instead of trying to fix the loss of orthogonality, we can try to make it irrelevant by proper modification of its use in a given situation. This idea is inspired by the same approach that Björck and Paige used in [6] for $\tilde{Q}$, which is the QR factor obtained in finite precision arithmetic by the modified Gram–Schmidt orthogonalization. $\tilde{Q}$ can also be far from being numerically orthogonal. Björck and Paige modified the computation of the matrix–vector product $\tilde{Q}^T y$ in such a way, that $\mathrm{fl}(\tilde{Q}^T y)$ became numerically stable. Thus, we can retain good efficiency of the Barlow bidiagonalization and its potential in parallel computing, and still fix the problem with numerics.

The following subsections will describe several problems in numerical linear algebra and their modifications, so that they can be accurately solved by Barlow bidiagonalization without reorthogonalization (see [8]).

3.3.1 The Symmetric Definite Eigenvalue Problem

Suppose we need to compute the spectral decomposition of a symmetric positive definite matrix $H = A^T A$, where A is the computed Cholesky factor, or any other full column rank factor obtained from the application that generates H. Here we note that in some important applications, for example in finite element computation in structural mechanics, the numerically most important step is not to assemble H, but to formulate the problem in terms of A and functions of A. Hence, the spectral decomposition of H can be obtained from the singular value decomposition of A, as shown in the following algorithm.

Algorithm 3.3.1. *This algorithm finds the spectral decomposition of a symmetric positive definite matrix H.*

1. *Factor H as $H = A^T A$, where A is the Cholesky or any other full column rank factor of A.*

2. *Compute the singular value matrix $\Sigma = \mathrm{diag}(\sigma_i)$ and the right singular vectors V of A using bidiagonalization of Algorithm 3.1.1 and some state of the art bidiagonal SVD.*

3. *The spectral factorization of H is $H = V \Lambda V^T$, $\Lambda = \mathrm{diag}(\lambda_i)$, $\lambda_i = \sigma_i^2$.*

The idea of the above algorithm is not new. Its parallelization by one–sided tridiagonalization also appears in [44] and [45] where the final stage is not done by the SVD, but

by the symmetric tridiagonal spectral decomposition. Nevertheless, this is not wise from the numerical point of view. Our formulation is numerically correct, as the following analysis shows.

Theorem 3.3.2. *If in Algorithm 3.3.1 the matrix A is given, and H is implicitly defined as $H = A^TA$, then Algorithm 3.3.1 is backward stable. If $H \in \mathbb{R}^{n\times n}$ is given, and $\|H_s^{-1}\|_2 < \frac{1-2(n+1)\varepsilon}{n(n+1)\varepsilon}$, where ε is the machine roundoff, then Algorithm 3.3.1 will successfully compute the Cholesky factorization and the overall computation is backward stable. Here H_s is the matrix $(H_s)_{ij} = H_{ij}/\sqrt{H_{ii}H_{jj}}$, $i,j = 1,\ldots,n$.*

Proof. Let $\tilde{A}$ be the computed factor of H. Then $\tilde{A}^T\tilde{A} = H + \delta H$, where $\|\delta H\|_2 \leq f(n)\varepsilon\|H\|_2$, and the moderate function $f(n)$ depends on the details of the factorization. In fact, $|\delta H_{ij}| \leq f_1(n)\varepsilon\sqrt{H_{ii}H_{jj}}$, if $\tilde{A}$ is computed by Cholesky factorization, with $f_1(n) = O(n)$ and $f(n) = O(n^2)$ [47, pp. 206–207], [13]. If $\tilde{A}$ is the computed Cholesky factor, then our assumption guarantees that H is numerically definite and that the Cholesky factorization does not break down. This is a consequence of Theorem 10.7 from [47, pp. 208–209]. If A is given as input, then $\tilde{A} = A$, $\delta H = 0$.

If $\tilde{B}$ is a bidiagonal matrix computed by an application of Algorithm 3.1.1 to $\tilde{A}$, then by Theorem 3.2.6 there exist orthogonal matrices $\hat{P}$, $\hat{V}$, perturbations $\Delta\tilde{A}$, $\delta\tilde{A}$ such that

$$\begin{bmatrix} \tilde{B} \\ 0 \end{bmatrix} = \hat{P}^T \begin{bmatrix} \Delta\tilde{A} \\ \tilde{A}+\delta\tilde{A} \end{bmatrix} \hat{V}, \quad \left\| \begin{bmatrix} \Delta\tilde{A} \\ \delta\tilde{A} \end{bmatrix} \right\|_F \leq \tilde{\xi}\|\tilde{A}\|_F, \quad \tilde{\xi} \leq O(mn+n^3)\varepsilon. \tag{3.32}$$

The computed $\tilde{V}$ satisfies $\|\tilde{V} - \hat{V}\|_F \leq O(n^2)\varepsilon$. Note that

$$\begin{aligned}
\tilde{B}^T\tilde{B} &= \hat{V}^T(\tilde{A}^T\tilde{A} + \Delta\tilde{A}^T\Delta\tilde{A} + \delta\tilde{A}^T\delta\tilde{A} + \tilde{A}^T\delta\tilde{A} + \delta\tilde{A}^T\tilde{A})\hat{V} = \\
&= \hat{V}^T(H + \delta H + \Delta H)\hat{V}, \quad \text{where} \\
\|\Delta H\|_2 &\leq \|\Delta\tilde{A}\|_F^2 + \|\delta\tilde{A}\|_F^2 + 2\|\tilde{A}\|_2\tilde{\xi}\|\tilde{A}\|_F \leq \\
&\leq (\tilde{\xi}^2 n + 2\sqrt{n}\tilde{\xi})\|\tilde{A}\|_2^2 \leq \tilde{\xi}(2\sqrt{n} + n\tilde{\xi})(\|H\|_2 + \|\delta H\|_2) \leq \\
&\leq \tilde{\xi}\|H\|_2(2\sqrt{n} + n\tilde{\xi})(1 + f(n)\varepsilon) \leq \\
&\leq O(mn^{\frac{3}{2}} + n^{\frac{7}{2}})\varepsilon\|H\|_2.
\end{aligned}$$

This corresponds to backward stable tridiagonalization of H, where the tridiagonal matrix is implicitly defined by the bidiagonal $\tilde{B}$ as $\tilde{B}^T\tilde{B}$. It is not recommended to compute it explicitly. Here we can note that ΔH is comparable with δH.

Let $\tilde{\Sigma}$, $\tilde{U}_B$, $\tilde{V}_B$ be the computed elements of the SVD $\tilde{B} \approx \tilde{U}_B\tilde{\Sigma}\tilde{V}_B^T$. Then there exist orthogonal matrices $\hat{U}_B$, $\hat{V}_B$ and a backward perturbation $\delta\tilde{B}$ such that

$$\tilde{B} + \delta\tilde{B} = \hat{U}_B\tilde{\Sigma}\hat{V}_B^T \tag{3.33}$$

and $\|\delta\tilde{B}\|_F \leq g(n)\varepsilon\|\tilde{B}\|_F$. Furthermore, $\|\tilde{U}_B - \hat{U}_B\|_F$ and $\|\tilde{V}_B - \hat{V}_B\|_F$ are small. Now we can put all the elements of the Algorithm 3.3.1 together to obtain

$$\begin{aligned}
\begin{bmatrix} \tilde{\Sigma} \\ 0 \end{bmatrix} &= \begin{bmatrix} \hat{U}_B^T & 0 \\ 0 & I \end{bmatrix} \hat{P}^T \left\{ \begin{bmatrix} \Delta\tilde{A} \\ \tilde{A}+\delta\tilde{A} \end{bmatrix} + \hat{P} \begin{bmatrix} \delta\tilde{B} \\ 0 \end{bmatrix} \hat{V}^T \right\} \hat{V}\hat{V}_B = \\
&= \begin{bmatrix} \hat{U}_B^T & 0 \\ 0 & I \end{bmatrix} \hat{P}^T \left\{ \begin{bmatrix} 0 \\ \tilde{A} \end{bmatrix} + \begin{bmatrix} \Delta'\tilde{A} \\ \delta'\tilde{A} \end{bmatrix} \right\} \hat{V}\hat{V}_B, \quad \text{that is,} \\
\tilde{\Sigma}^2 &= (\hat{V}\hat{V}_B)^T(\tilde{A}^T\tilde{A} + \Delta' H)(\hat{V}\hat{V}_B),
\end{aligned}$$

where $\Delta' H$ has a similar bound as ΔH, and $\mathrm{fl}(\tilde{V}\tilde{V}_B)$ is close to $\hat{V}\hat{V}_B$. Namely,

$$\begin{aligned}
\|\Delta' H\|_2 &\leq \|\Delta'\tilde{A}\|_F^2 + \|\delta'\tilde{A}\|_F^2 + 2\|\tilde{A}\|_2\|\delta'\tilde{A}\|_F \leq \\
&\leq (\tilde{\xi}\|\tilde{A}\|_F + \|\delta\tilde{B}\|_F)^2 + 2\|\tilde{A}\|_2(\tilde{\xi}\|\tilde{A}\|_F + \|\delta\tilde{B}\|_F) \leq \\
&\leq (\tilde{\xi}\|\tilde{A}\|_F + g(n)\varepsilon\|\tilde{B}\|_F)^2 + 2\|\tilde{A}\|_2(\tilde{\xi}\|\tilde{A}\|_F + g(n)\varepsilon\|\tilde{B}\|_F) \leq \\
&\leq [n(\tilde{\xi} + g(n)\varepsilon(1+\tilde{\xi}))^2 + 2\sqrt{n}(\tilde{\xi} + g(n)\varepsilon(1+\tilde{\xi}))]\|\tilde{A}\|_2^2 \leq \\
&\leq [\tilde{\xi} + g(n)\varepsilon(1+\tilde{\xi})][2\sqrt{n} + n\tilde{\xi} + ng(n)\varepsilon(1+\tilde{\xi})](1+f(n)\varepsilon)\|H\|_2 \leq \\
&\leq O(mn^{\frac{3}{2}} + n^{\frac{7}{2}} + n^{\frac{1}{2}}g(n))\varepsilon\|H\|_2
\end{aligned}$$

which implies

$$H + \delta H + \Delta' H = (\hat{V}\hat{V}_B)\tilde{\Sigma}^2(\hat{V}\hat{V}_B)^T,$$

and $\tilde{\Sigma}^2$ is the exact diagonal eigenfactor of $H + \delta H + \Delta' H$, where

$$\begin{aligned}
\|\delta H + \Delta' H\|_2 &\leq [f(n)\varepsilon + 2\sqrt{n}(\tilde{\xi} + g(n)\varepsilon)]\|H\|_2 + O(\varepsilon^2) \leq \\
&\leq O(mn^{\frac{3}{2}} + n^{\frac{7}{2}} + n^{\frac{1}{2}}g(n))\varepsilon\|H\|_2
\end{aligned}$$

□

Since Algorithm 3.3.1 computes the eigenvalues as squares of the singular values, the forward error in the computed eigenvalues of $\tilde{A}^T\tilde{A}$ is governed by the condition number of $\tilde{A}$, which is approximately equal to the square root of the condition number of H. The overall error is described in the following corollary.

Corollary 3.3.3. *Let $\tilde{\lambda}_1 \geq \cdots \geq \tilde{\lambda}_n$ be the approximations of the eigenvalues of H, computed by Algorithm 3.3.1. Then, using the notation and the assumptions of Theorem 3.3.2, for all i it holds that*

$$\tilde{\lambda}_i = (1+\varrho_i)\lambda_i, \quad |\varrho_i| \leq 2\sqrt{2}\sqrt{n}\tilde{\xi}\sqrt{\kappa_2(H)} + O(n^2)\varepsilon\|H_s^{-1}\|_2 + 2\tau(n)\varepsilon + O(\varepsilon^2), \quad (3.34)$$

where $\tau(n)\varepsilon$ is described bellow, and denotes the bound on relative forward error for the SVD performed on $\tilde{B}$. Further, if all eigenvalues are simple with $Hv_i = \lambda_i v_i$, $V = [v_1, \ldots, v_n]$, $V^TV = I$, then the columns $\tilde{v}_i$ of the computed $\tilde{V}$ satisfy

$$\sin\measuredangle(\tilde{v}_i, v_i) \leq \left[\frac{O(mn^{\frac{3}{2}} + n^{\frac{7}{2}} + n^{\frac{1}{2}}g(n))}{\min_{\substack{j=1,\ldots,n \\ j\neq i}} |\tilde{\tilde{\sigma}}_i^2 - \lambda_j|} + O(n^2)\right]\varepsilon + O(\varepsilon^2), \quad (3.35)$$

where $\tilde{\tilde{\sigma}}_i$ are the computed singular values of $\tilde{A}$. The similar result can be obtained for the multiple eigenvalues, using the results presented by R.-C. Li in [67].

Proof. First we compare the computed eigenvalues with the eigenvalues of $\tilde{A}^T\tilde{A}$. From Corollary 3.2.13, it holds that the singular values $\hat{\tilde{\sigma}}_1 \geq \cdots \geq \hat{\tilde{\sigma}}_n$ of $\tilde{B}$ approximate the singular values $\tilde{\sigma}_1 \geq \cdots \geq \tilde{\sigma}_n$ of $\tilde{A}$ with an error bound

$$\max_i |\hat{\tilde{\sigma}}_i - \tilde{\sigma}_i| \leq \sqrt{2}\sqrt{n}\tilde{\xi}\tilde{\sigma}_1, \quad \text{i.e. } \hat{\tilde{\sigma}}_i = \tilde{\sigma}_i(1+\eta_i), \quad |\eta_i| \leq \sqrt{2}\sqrt{n}\tilde{\xi}\frac{\tilde{\sigma}_1}{\tilde{\sigma}_i}.$$

A state of the art SVD of $\tilde{B}$ computes $\tilde{\tilde{\sigma}}_i = \hat{\tilde{\sigma}}_i(1+\theta_i)$, where $|\theta_i| \leq \tau(n)\varepsilon$ for all i. Hence, our computed approximations have the following form

$$\tilde{\lambda}_i = \tilde{\tilde{\sigma}}_i^2 = \tilde{\sigma}_i^2(1+\theta_i)^2(1+\eta_i)^2 = \lambda_i(1+\tau_i)(1+\theta_i)^2(1+\eta_i)^2 = \lambda_i(1+\varrho_i),$$

and

$$|\varrho_i| \leq |\tau_i| + 2|\theta_i| + 2|\eta_i| + O(\varepsilon^2) \leq f_2(n)\varepsilon\|H_s^{-1}\|_2 + 2\tau(n)\varepsilon + 2\sqrt{2}\sqrt{n}\tilde{\xi}\kappa_2(\tilde{A}) + O(\varepsilon^2),$$

where $\max_i |\tau_i| \leq f_2(n)\varepsilon\|H_s^{-1}\|_2$, with $f_2(n) = nf_1(n) = O(n^2)$. This follows from the fact that $\tilde{A}^T\tilde{A} = H + \delta H$ is the computed Cholesky factorization, from a consequence of the Ostrowsky theorem 4.3.4, and from Lemma 2.2 and Theorem 2.3 in [69, pp. 963–964]. (If A is given on input, $\tau_i = 0$ for all i.)

For the other part of the corollary we use the matrices obtained from the SVD performed in finite precision arithmetic. We have

$$\tilde{V} = \mathrm{fl}(\tilde{V}_{\tilde{A}}\tilde{V}_{\tilde{B}}) = \hat{W} + \delta\tilde{V}, \qquad \hat{W} = \hat{V}_{\tilde{A}}\hat{V}_{\tilde{B}}, \tag{3.36}$$

where $\hat{V}_{\tilde{A}}$ is equal to $\hat{V}$ in (3.32) and $\hat{V}_{\tilde{B}}$ is defined by (3.33). They both are exact orthogonal matrices and

$$\|\delta\tilde{V}\|_F \leq \|\tilde{V}_{\tilde{A}} - \hat{V}_{\tilde{A}}\|_F + \|\tilde{V}_{\tilde{B}} - \hat{V}_{\tilde{B}}\|_F + O(\varepsilon^2) \leq O(n^2)\varepsilon + O(\varepsilon^2).$$

On the other hand, from the proof of Theorem 3.3.2, we have

$$H + \delta H + \Delta' H = \hat{W}\tilde{\tilde{\Sigma}}^2\hat{W}^T.$$

Here $\tilde{\Sigma}$ in the proof of Theorem 3.3.2 is replaced by $\tilde{\tilde{\Sigma}}$. In order to use the perturbation theory from [26], we define three orthogonal projections. They are required for error analysis of eigenvectors, and represent eigenprojections:

$$\begin{aligned} P_{v_i} &= v_i v_i^T \\ P_{\hat{w}_i} &= \hat{w}_i \hat{w}_i^T \\ P_{\tilde{v}_i} &= \frac{\tilde{v}_i \tilde{v}_i^T}{\|\tilde{v}_i\|_2^2} \end{aligned}$$

According to Eisenstat and Ipsen [26], we can write

$$\sin\measuredangle(v_i, \tilde{v}_i) = \|P_{v_i} - P_{\tilde{v}_i}\|_2 \leq \|P_{v_i} - P_{\hat{w}_i}\|_2 + \|P_{\hat{w}_i} - P_{\tilde{v}_i}\|_2. \tag{3.37}$$

For the first term in (3.37) we obtain the following estimation:

$$\begin{aligned} \|P_{v_i} - P_{\hat{w}_i}\|_2 &= \sin\measuredangle(v_i, \hat{w}_i) \leq \frac{\|\delta H + \Delta' H\|_2}{\min_{\substack{j=1,\ldots,n \\ j\neq i}} |\tilde{\tilde{\sigma}}_i^2 - \lambda_j|} \\ &\leq \frac{[f(n)\varepsilon + 2\sqrt{n}(\tilde{\xi} + g(n)\varepsilon)]\|H\|}{\min_{\substack{j=1,\ldots,n \\ j\neq i}} |\tilde{\tilde{\sigma}}_i^2 - \lambda_j|} + O(\varepsilon^2) \\ &\leq \frac{O(mn^{\frac{3}{2}} + n^{\frac{7}{2}} + n^{\frac{1}{2}}g(n))\varepsilon}{\min_{\substack{j=1,\ldots,n \\ j\neq i}} |\tilde{\tilde{\sigma}}_i^2 - \lambda_j|} + O(\varepsilon^2). \end{aligned}$$

Furthermore, by relation (3.36) we can compute the second term in (3.37),

$$\begin{aligned} P_{\tilde{v}_i} - P_{\hat{w}_i} &= \frac{(\hat{w}_i + \delta\tilde{v}_i)(\hat{w}_i + \delta\tilde{v}_i)^T}{\|\tilde{v}_i\|_2^2} - \hat{w}_i\hat{w}_i^T \\ &= \frac{(1 - \|\tilde{v}_i\|_2^2)\hat{w}_i\hat{w}_i^T + \delta\tilde{v}_i\hat{w}_i^T + \hat{w}_i\delta\tilde{v}_i^T + \delta\tilde{v}_i\delta\tilde{v}_i^T}{\|\tilde{v}_i\|_2^T}, \end{aligned}$$

thus

$$\begin{aligned} \|P_{\tilde{v}_i} - P_{\hat{w}_i}\|_2 &\leq \frac{1 - \|\tilde{v}_i\|_2^T + O(n^2)\varepsilon}{\|\tilde{v}_i\|_2} + O(\varepsilon^2) \\ &\leq \frac{1 - (1 - \|\delta\tilde{v}_i\|_2)^2 + O(n^2)\varepsilon}{(1 - \|\delta\tilde{v}_i\|_2)^2} + O(\varepsilon^2) \\ &\leq \frac{O(n^2)\varepsilon}{1 - O(n^2)\varepsilon} + O(\varepsilon^2) = O(n^2)\varepsilon + O(\varepsilon^2). \end{aligned}$$

The final result is now straightforward. □

By Corollary 3.3.3 Algorithm 3.3.1 produces computed eigenvalues $\tilde{\lambda}_i = \tilde{\tilde{\sigma}}_i^2$ of H with small relative error, and the accuracy of computed eigenvectors $\tilde{v}_i$ depends on the gap between $\tilde{\tilde{\sigma}}_i^2$ and the rest of the exact spectrum of H. Let us verify the result of Corollary 3.3.3 with an example.

Example 3.3.4. *We generated a symmetric positive definite matrix* $H \in \mathbb{R}^{10\times 10}$ *with fixed eigenvalues* $\lambda(H) = \{10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3, 10^4, 10^5, 10^6\}$ *as*

$$H = V\Lambda V^T,$$

where V *is a random orthogonal matrix, and* $\Lambda = \mathrm{diag}(10^{-3}, 10^{-2}, \ldots, 10^5, 10^6)$. *First we check the condition from Theorem 3.3.2. We have*

$$\|H_s^{-1}\|_2 = 1.0012 \cdot 10^8 < 8.1884 \cdot 10^{13} = \frac{1 - 2(n+1)\varepsilon}{n(n+1)\varepsilon},$$

and the Cholesky factorization will be computed successfully. Next, we execute Algorithm 3.3.1, and we obtain computed $\tilde{\Lambda}$ *and* $\tilde{V}$, *where computed eigenvalues are equal to*

$$\begin{aligned} &9.999999999999998 \cdot 10^5 \\ &9.999999999999987 \cdot 10^4 \\ &9.999999999999964 \cdot 10^3 \\ &9.999999999999948 \cdot 10^2 \\ &9.999999999998458 \cdot 10^1 \\ &9.999999999997730 \cdot 10^0 \\ &9.999999999960223 \cdot 10^{-1} \\ &9.999999998392710 \cdot 10^{-2} \\ &9.999999990792137 \cdot 10^{-3} \\ &9.999999713117659 \cdot 10^{-4}. \end{aligned}$$

The errors and the error bounds are presented in Figure 3.2 and 3.3. The eigenvalues are sorted in nonincreasing order. From [15], it follows that $\tau(n) \approx O(n^2)$ *for the bidiagonal SVD algorithm implemented in LAPACK and MATLAB, and this implies that* $g(n) \approx O(n^2)$.

We can see in Figure 3.3 that for $v_5, \ldots, v_{10}$ *the error is greater than the bound. It happened because after computing* $\tilde{H} = \mathrm{fl}(V\Lambda V^T)$ *in finite precision arithmetic, columns of* V *are not the exact eigenvectors of* $\tilde{H}$ *any more. Furthermore, computation of sine of an angle is numerically sensitive to errors. Nevertheless, shapes of both curves are similar, and the trend of the bound follows the trend of the computed error.*

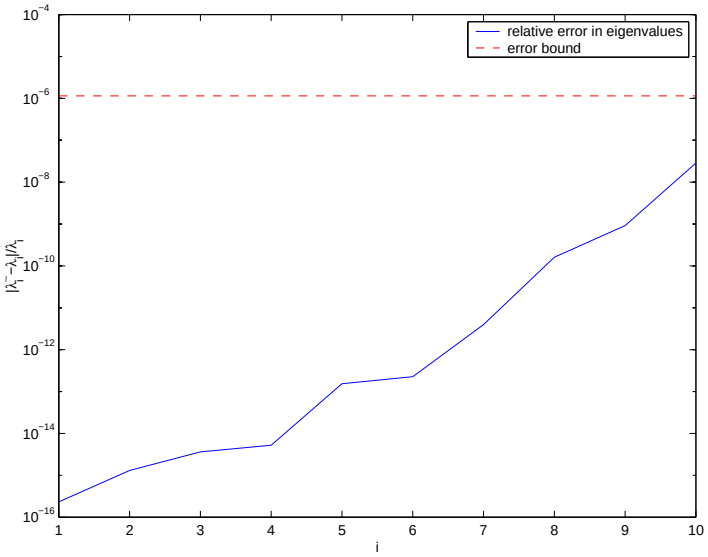


Figure 3.2: Relative error in eigenvalues and the bound (3.34), for Example 3.3.4.

3.3.2 Intersection of Null Spaces

Let $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{p\times n}$ be given, and consider the problem of finding an orthonormal basis for $\mathrm{null}(A) \cap \mathrm{null}(B)$. When the SVD is used to compute the orthonormal bases we obtain the following procedure as a consequence of Corollary 2.1.4 (see Section 2.2 and [35, pp. 583–584])

Algorithm 3.3.5. *Given* $A \in \mathbb{R}^{m\times n}$ *and* $B \in \mathbb{R}^{p\times n}$, *the following algorithm computes an integer* s *and a matrix* $Y = [y_1, \ldots, y_s]$ *having orthonormal columns which span* $\mathrm{null}(A) \cap \mathrm{null}(B)$. *If the intersection is trivial then* $s = 0$.

Compute the SVD $U_A^T A V_A = diag(\sigma_i)$. *Save* V_A *and set*
  $r = rank(A)$;
if $r < n$
  $C = BV_A(:, r+1:n)$;
  Compute the SVD $U_C^T C V_C = diag(\gamma_i)$. *Save* V_C *and set*

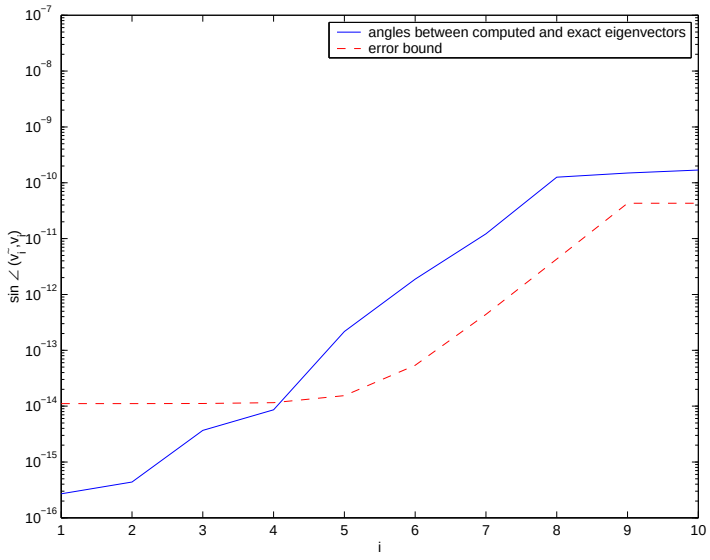


Figure 3.3: Sine of the angles between the computed and the exact eigenvectors, and the bound (3.35), for Example 3.3.4.

$q = rank(C);$
if $q < n - r$
$s = n - r - q;$
$Y = V_A(:, r + 1 : n)V_C(:, q + 1 : n - r);$
else
$s = 0;$
end
else
$s = 0;$
end

As we can see, Algorithm 3.3.5 uses only right singular vectors for computing the orthonormal basis for null spaces. Matrices U_A and U_C are never used, so their departure from orthogonality is not important. The following theorem shows mixed error stability of the algorithm.

Theorem 3.3.6. *Let $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{p\times n}$, and let $\tilde{Y}$ be computed in finite precision arithmetic, using Algorithm 3.3.5 and Algorithm 3.1.1 for obtaining the SVD. Then*

$$\|\tilde{Y} - \hat{Y}\|_F \leq (O(n^2) + h(n))\varepsilon,$$

where $\hat{Y}$ is a matrix consisting of orthonormal vectors which span close approximation of $\mathrm{null}(A + \delta_1 A) \cap \mathrm{null}(B + \delta_1 B)$, *and*

$$\begin{aligned} \|\delta_1 A\|_F &\leq (O(mn + n^3) + g(n))\varepsilon\|A\|_F, \\ \|\delta_1 B\|_F &\leq (O(p(n - r) + (n - r)^3 + n^2) + g(n - r) + h(n))\varepsilon\|B\|_F. \end{aligned}$$

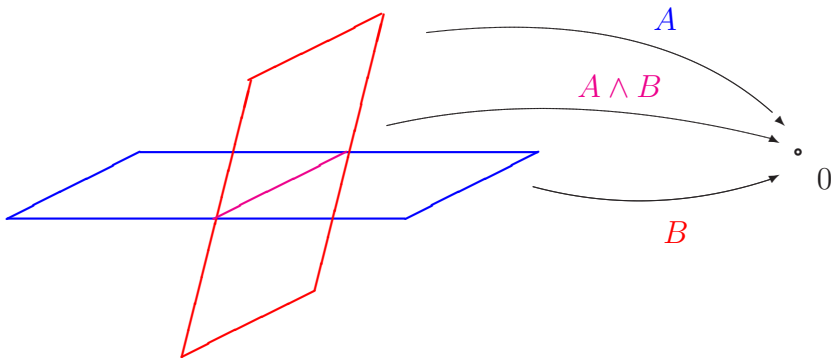


Figure 3.4: Intersection of null spaces of the operators A and B.

$g(n)\varepsilon$ is the bound on normwise backward error obtained in computing the SVD of an $n \times n$ bidiagonal matrix, and $h(n)\varepsilon$ is the bound on departure from orthogonality of the computed right singular vectors of the bidiagonal matrix.

Proof. If the SVD is computed using Barlow's bidiagonalization, then by Theorem 3.2.6 in finite precision arithmetic we have the following situation:

$$A + \delta_1 A = \hat{U}_A \tilde{\Sigma}_A \hat{V}_A^T,$$

where $\hat{U}_A \in \mathbb{R}^{m\times n}$ is orthonormal, $\tilde{\Sigma}_A \in \mathbb{R}^{n\times n}$ is computed diagonal, $\hat{V}_A \in \mathbb{R}^{n\times n}$ is orthogonal, and

$$\|\delta_1 A\|_F \leq (O(mn + n^3) + g(n))\varepsilon \|A\|_F.$$

For the computed matrix $\tilde{V}_A$ we can write

$$\tilde{V}_A = \hat{V}_A + \delta\hat{V}_A, \qquad \|\delta\hat{V}_A\|_F \leq (O(n^2) + h(n))\varepsilon.$$

Further, from [47, pp. 76–78] it follows

$$\tilde{C} = \mathrm{fl}(B\tilde{V}_A(:, r+1:n)) = B\hat{V}_A(:, r+1:n) + \delta C,$$

where

$$\|\delta C\|_F \leq (O(n^2) + h(n))\varepsilon \|B\|_F,$$

and, on the other hand

$$\tilde{C} + \delta_1 C = \hat{U}_C \tilde{\Sigma}_C \hat{V}_C^T,$$

where $\hat{U}_C \in \mathbb{R}^{p\times(n-r)}$ is orthonormal, $\tilde{\Sigma}_C \in \mathbb{R}^{(n-r)\times(n-r)}$ is computed diagonal, $\hat{V}_C \in \mathbb{R}^{(n-r)\times(n-r)}$ is orthogonal, and

$$\|\delta_1 C\|_F \leq (O(p(n-r) + (n-r)^3) + g(n-r))\varepsilon \|B\|_F.$$

For the computed matrix $\tilde{V}_C$ we can write

$$\tilde{V}_C = \hat{V}_C + \delta\hat{V}_C, \qquad \|\delta\hat{V}_C\|_F \leq (O((n-r)^2) + h(n-r))\varepsilon.$$

Finally, by [47] again we have

$$\begin{aligned}\tilde{Y} &= \mathrm{fl}(\tilde{V}_A(:, r+1:n)\tilde{V}_C(:, q+1:n-r)) \\ &= \hat{V}_A(:, r+1:n)\hat{V}_C(:, q+1:n-r) + \delta\hat{Y} \\ &= \hat{Y} + \delta\hat{Y},\end{aligned}$$

where

$$\hat{Y} = \hat{V}_A(:, r+1:n)\hat{V}_C(:, q+1:n-r), \qquad \|\delta\hat{Y}\|_F \leq (O(n^2) + h(n))\varepsilon.$$

If we put all this together, we obtain

$$\begin{aligned}\tilde{C} + \delta_1 C &= B\hat{V}_A(:, r+1:n) + \delta C + \delta_1 C = \\ &= (B + \delta C\hat{V}_A(:, r+1:n)^T + \delta_1 C\hat{V}_A(:, r+1:n)^T)\hat{V}_A(:, r+1:n) = \\ &= (B + \delta_1 B)\hat{V}_A(:, r+1:n),\end{aligned}$$

with

$$\|\delta_1 B\|_F \leq (O(p(n-r) + (n-r)^3 + n^2) + g(n-r) + h(n))\varepsilon\|B\|_F.$$

This means that

$$\begin{aligned}A + \delta_1 A &= \hat{U}_A\tilde{\Sigma}_A\hat{V}_A^T \\ (B + \delta_1 B)\hat{V}_A(:, r+1:n) &= \hat{U}_C\tilde{\Sigma}_C\hat{V}_C^T \\ \tilde{Y} &= \hat{Y} + \delta Y\end{aligned}$$

which implies that $\tilde{Y}$ is not very far from the exact solution of the same problem posed for matrices $A + \delta_1 A$ and $B + \delta_1 B$. Here should be noted that the last $n - r$ singular values in $\tilde{\Sigma}_A$ and the last $s = n - r - q$ singular values in $\tilde{\Sigma}_C$ are probably not equal to zero. Indeed, $A + \delta_1 A$ and $\tilde{C} + \delta_1 C$ could be nonsingular matrices having $n - r$ and s very small singular values, respectively. This means that $\hat{Y}$ represents an intersection of the subspaces which are very close to null spaces. □

3.3.3 The Linear Least Squares Problems

Consider now the $m \times n$ least squares problem

$$\min_{x \in \mathbb{R}^n} \|Ax - b\|_2.$$

If $A = U\Sigma V^T$ is the SVD of A with $m \times n$ orthonormal U and $n \times n$ orthogonal V, then by writing

$$\|Ax - b\|_2^2 = \|U\Sigma V^T x - UU^T b - (I - UU^T)b\|_2^2 = \|\Sigma V^T x - U^T b\|_2^2 + \|b - U(U^T b)\|_2^2,$$

we obtain the minimal norm solution $x = V\Sigma^\dagger U^T b$. Note that mutual orthogonality of columns of U was important in splitting b (and $Ax - b$), as well as using orthogonal

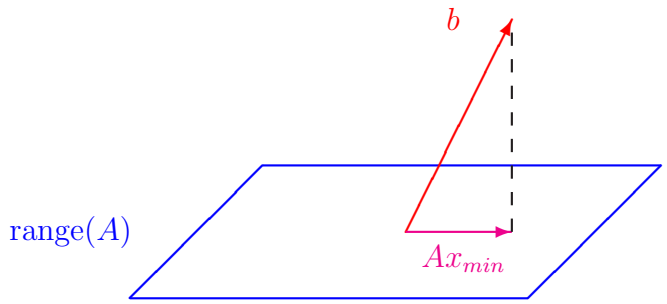


Figure 3.5: Finding the solution x_{min} of the linear least squares problems, for the matrix A and the vector b.

invariance of the Euclidean norm. Now, suppose we have computed the SVD, where the computed $\tilde{U}$, $\tilde{V}$, $\tilde{\Sigma}$ satisfy

$$A + \delta A = \tilde{U}\tilde{\Sigma}\tilde{V}^T, \qquad \|\delta A\|_F \leq \zeta \|A\|_F.$$

Since $\tilde{U}$ is not orthonormal and $\tilde{V}$ is not orthogonal, proper formulation of the solution procedure cannot be reduced to putting tildes to the matrices in the exact formulas above. However, the solution vector is computed as $\tilde{x} = \mathrm{fl}(\tilde{V}\tilde{\Sigma}^{\dagger}\tilde{U}^T b)$. To justify this formula, we need the fact that there exist orthonormal

$$\hat{U} = \tilde{U} - \delta\tilde{U}, \quad \text{such that } \|\delta\tilde{U}\|_F \leq \zeta$$

and orthogonal

$$\hat{V} = \tilde{V} - \delta\tilde{V}, \quad \text{such that } \|\delta\tilde{V}\|_F \leq \zeta.$$

Then

$$\mathrm{fl}(\tilde{U}^T b) = \hat{U}^T(b + \delta b), \quad \text{where } \|\delta b\|_2 \leq (\zeta + O(m)\varepsilon)\|b\|_2.$$

Further,

$$\mathrm{fl}(\tilde{\Sigma}^{\dagger}(\hat{U}^T(b + \delta b)) = (\tilde{\Sigma} + \delta\tilde{\Sigma})^{\dagger}\hat{U}^T(b + \delta b), \tag{3.38}$$

where $\delta\tilde{\Sigma}$ is diagonal with $|\delta\tilde{\Sigma}_{ii}| \leq \varepsilon\tilde{\Sigma}_{ii}$ for all i. Finally,

$$\begin{aligned} \tilde{x} &= \mathrm{fl}(\tilde{V}((\tilde{\Sigma} + \delta\tilde{\Sigma})^{\dagger}\hat{U}^T(b + \delta b))) = \tilde{V}(\tilde{\Sigma} + \delta\tilde{\Sigma})^{\dagger}\hat{U}^T(b + \delta b) + \delta\tilde{x} \\ &= \hat{V}(\tilde{\Sigma} + \delta\tilde{\Sigma})^{\dagger}\hat{U}^T(b + \delta b) + \delta\tilde{V}(\tilde{\Sigma} + \delta\tilde{\Sigma})^{\dagger}\hat{U}^T(b + \delta b) + \delta\tilde{x}, \end{aligned} \tag{3.39}$$

where

$$\|\delta\tilde{x}\|_2 \leq O(n)\varepsilon\|\tilde{V}\|_F\|(\tilde{\Sigma} + \delta\tilde{\Sigma})^{\dagger}\hat{U}^T(b + \delta b)\|_F \leq O(n^{\frac{3}{2}})\varepsilon\|(\tilde{\Sigma} + \delta\tilde{\Sigma})^{\dagger}\hat{U}^T(b + \delta b)\|_F.$$

Now, set $\hat{x} = \hat{V}(\tilde{\Sigma} + \delta\tilde{\Sigma})^{\dagger}\hat{U}^T(b + \delta b)$, $\delta\hat{x} = \delta\tilde{x} + \delta\tilde{V}\hat{V}^T\hat{x}$, and note that $\hat{x}$ solves a nearby problem

$$\min_{x \in \mathbb{R}^n} \|(A + \Delta A)x - (b + \delta b)\|_2$$

where $A + \Delta A = \hat{U}(\tilde{\Sigma} + \delta\tilde{\Sigma})\hat{V}^T$ approximates A equally well as $A + \delta A$, because

$$A + \Delta A = \tilde{U}\tilde{\Sigma}\tilde{V}^T + \hat{U}\delta\tilde{\Sigma}\hat{V}^T - \tilde{U}\tilde{\Sigma}\delta\tilde{V}^T - \delta\tilde{U}\tilde{\Sigma}\tilde{V}^T + \delta\tilde{U}\tilde{\Sigma}\delta\tilde{V}^T.$$

and

$$\|\Delta A\|_F \leq (O(1)\varepsilon + O(1)\zeta)\|A\|_F.$$

On the other hand, $\tilde{x} = \hat{x} + \delta\hat{x}$, where

$$\|\delta\hat{x}\|_2 \leq (O(n^{\frac{3}{2}})\varepsilon + O(1)\zeta)\|\hat{x}\|_2.$$

The conclusion is that the backward stable SVD with numerically orthogonal singular vectors produces a vector close to the exact solution of a nearby problem. Strictly speaking, this mixed stability is the best we can hope to achieve in the SVD solution to the least squares problem. Pushing all errors into the backward perturbation of the data introduces dependence on the size of the condition number. Applications of $\tilde{U}^T$ and $\tilde{\Sigma}^\dagger$ to b in (3.38) represent backward stable operations. On the other hand, application of $\tilde{V}$ in (3.39) undermines backward stability. If we assume that $\hat{V}(\tilde{\Sigma} + \delta\tilde{\Sigma})^\dagger\hat{U}^T$ is a full rank decomposition, then further computation in (3.39) will result with

$$\begin{aligned}\tilde{x} &= \hat{V}(\tilde{\Sigma} + \delta\tilde{\Sigma})^\dagger\hat{U}^T[b + \delta b + \hat{U}(\tilde{\Sigma} + \delta\tilde{\Sigma})\hat{V}^T\delta\tilde{V}(\tilde{\Sigma} + \delta\tilde{\Sigma})^\dagger\hat{U}^T(b + \delta b) + \hat{U}(\tilde{\Sigma} + \delta\tilde{\Sigma})\hat{V}^T\delta\tilde{x}] \\ &= \hat{V}(\tilde{\Sigma} + \delta\tilde{\Sigma})^\dagger\hat{U}^T(b + \Delta b),\end{aligned}$$

where

$$\|\Delta b\|_2 \leq O(\kappa(A))\varepsilon\|b\|_2.$$

Note that this is not the case for the least squares solution using QR factorization, which is a backward stable LS solution, see Theorem 19.3 in [47].

Using SVD with One–sided Bidiagonalization

Consider now the above solution procedure, but with the SVD computed using one–sided bidiagonalization. Since for the computed bidiagonalization $A \approx \tilde{U}\tilde{B}\tilde{V}^T$ we cannot guarantee that the computed $\tilde{U}$ is numerically orthogonal, the above analysis does not yield the conclusion we wanted. Recall that certain numerical orthogonality is ensured in the augmented matrix formulation (see the proof of Theorem 3.2.6) and we therefore write the least squares problem in the following obviously equivalent form

$$\min_{x\in\mathbb{R}^n} \left\| \begin{bmatrix} 0 \\ A \end{bmatrix} x - \begin{bmatrix} 0 \\ b \end{bmatrix} \right\|_2 .$$

The augmented bidiagonalization and the SVD read

$$\begin{bmatrix} 0 \\ A \end{bmatrix} = P \begin{bmatrix} B \\ 0 \end{bmatrix} V^T = P \begin{bmatrix} U_B & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \Sigma \\ 0 \end{bmatrix} (VV_B)^T,$$

where $B = U_B\Sigma V_B^T$ is the SVD of the bidiagonal B. Using this SVD, we obtain

$$\begin{aligned}\left\| \begin{bmatrix} 0 \\ A \end{bmatrix} x - \begin{bmatrix} 0 \\ b \end{bmatrix} \right\|_2 &= \left\| \begin{bmatrix} \Sigma \\ 0 \end{bmatrix} (VV_B)^T x - \begin{bmatrix} U_B^T & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} P_{11}^T & P_{21}^T \\ P_{12}^T & P_{22}^T \end{bmatrix} \begin{bmatrix} 0 \\ b \end{bmatrix} \right\|_2 = \\ &= \left\| \begin{bmatrix} \Sigma(VV_B)^T x \\ 0 \end{bmatrix} - \begin{bmatrix} U_B^T P_{21}^T b \\ P_{22}^T b \end{bmatrix} \right\|_2 ,\end{aligned}$$

thus

$$\|Ax-b\|_2^2 = \|\Sigma(VV_B)^T x - U_B^T P_{21}^T b\|_2^2 + \|P_{22}^T b\|_2^2,$$

where

$$P = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}, \quad P_{11} \in \mathbb{R}^{n\times n}, P_{21} \in \mathbb{R}^{m\times n},$$

and the minimal norm solution is

$$x = VV_B\Sigma^\dagger U_B^T P_{21}^T b.$$

In exact arithmetic, according to Björck and Paige [6], P is of the form

$$P = \begin{bmatrix} 0 & U^T \\ U & I - UU^T \end{bmatrix},$$

so that $P_{11} = 0$, $P_{21} = U$, $A = (P_{21}U_B)\Sigma(VV_B)$ is the SVD of A and the bidiagonalization is $A = P_{21}BV^T$. Recall that premultiplication by P expresses (theoretically and numerically) the modified Gram–Schmidt procedure applied to A, and note that $P_{21}^T b$ is the upper $n \times 1$ part of $P^T \begin{bmatrix} 0 \\ b \end{bmatrix}$. Instead of P_{21} we have the computed matrix $\tilde{U}$, but using it in $\tilde{U}^T b$ or for explicit forming of the left singular vector matrix $\tilde{U}\tilde{U}_B$ may introduce unacceptable numerical error. Still, there is a way presented in [6] that allows us to apply $\tilde{U}$ on b without producing a big error.

Lemma 3.3.7. *Let the computed approximation $\tilde{y} = [\tilde{\gamma}_1, \ldots, \tilde{\gamma}_n]^T$ of $y = \tilde{U}^T b$ be computed using the formulae*

(1) $\tilde{c}_1 = b$;

(2) **for** $i = 1 : n$
$\tilde{\gamma}_i = \mathrm{fl}(\tilde{u}_i^T \tilde{c}_i)$;
$\tilde{c}_{i+1} = \mathrm{fl}(\tilde{c}_i - \tilde{\gamma}_i \tilde{u}_i)$;
end

where $\tilde{U} = [\tilde{u}_1, \ldots, \tilde{u}_n]$ is computed as described in Algorithm 3.1.1 and Theorem 3.2.6. Then there exist perturbations $\delta_0 b$, $\Delta_0 b$ such that

$$\begin{bmatrix} \tilde{y} \\ \tilde{c}_{n+1} \end{bmatrix} = \hat{P}^T \begin{bmatrix} \Delta_0 b \\ b + \delta_0 b \end{bmatrix}$$

where $\hat{P}$ is the orthogonal matrix from Theorem 3.2.6, and

$$\left\| \begin{bmatrix} \Delta_0 b \\ \delta_0 b \end{bmatrix} \right\|_2 \leq O(mn)\varepsilon \|b\|_2.$$

Proof. First, we should note that the procedure presented in Lemma 3.3.7 is the same as performing a modified Gram–Schmidt orthogonalization of the vector b against vectors $\tilde{u}_1, \ldots, \tilde{u}_n$, but without normalization. Let

$$\tilde{P}_i = I - \begin{bmatrix} -e_i \\ \tilde{u}_i \end{bmatrix} \begin{bmatrix} -e_i^T & \tilde{u}_i^T \end{bmatrix}, \quad \hat{P}_i = I - \begin{bmatrix} -e_i \\ \hat{u}_i \end{bmatrix} \begin{bmatrix} -e_i^T & \hat{u}_i^T \end{bmatrix}, \quad i = 1, \ldots, n,$$

as defined in the proof of Theorem 3.2.6, and

$$\hat{\gamma}_i = \tilde{u}_i^T \tilde{c}_i, \qquad \hat{c}_i = \tilde{c}_{i-1} - \hat{\gamma}_{i-1}\tilde{u}_{i-1}, \quad i = 1, \ldots, n, \qquad \hat{y} = [\hat{\gamma}_1, \ldots, \hat{\gamma}_n]^T.$$

From the relations

$$\begin{aligned}
\tilde{P}_1 \begin{bmatrix} 0 \\ b \end{bmatrix} &= \begin{bmatrix} (\tilde{u}_1^T b)e_1 \\ b - (\tilde{u}_1^T b)\tilde{u}_1 \end{bmatrix} = \begin{bmatrix} \hat{\gamma}_1 e_1 \\ \hat{c}_2 \end{bmatrix} \\
\tilde{P}_2\tilde{P}_1 \begin{bmatrix} 0 \\ b \end{bmatrix} &= \begin{bmatrix} (\tilde{u}_1^T b)e_1 + (\tilde{u}_2^T \hat{c}_2)e_2 \\ \hat{c}_2 - (\tilde{u}_2^T \hat{c}_2)\tilde{u}_2 \end{bmatrix} = \begin{bmatrix} \hat{\gamma}_1 e_1 + \hat{\gamma}_2 e_2 \\ \hat{c}_3 \end{bmatrix} \\
&\vdots \\
\tilde{P}_n \cdots \tilde{P}_2\tilde{P}_1 \begin{bmatrix} 0 \\ b \end{bmatrix} &= \begin{bmatrix} (\tilde{u}_1^T b)e_1 + \cdots + (\tilde{u}_n^T \hat{c}_n)e_n \\ \hat{c}_n - (\tilde{u}_n^T \hat{c}_n)\tilde{u}_n \end{bmatrix} = \begin{bmatrix} \hat{y} \\ \hat{c}_{n+1} \end{bmatrix}
\end{aligned}$$

we inductively see that the computation of $\tilde{y}$ above is numerically equivalent to n applications of Householder reflectors with vectors $\tilde{u}_i$. The statement of Lemma 3.3.7 follows then from Lemma 3.2.4, for $\hat{P} = \hat{P}_1 \cdots \hat{P}_n$. □

Remark 3.3.8. *Recall that one of the techniques to obtain an acceptable numerical solution for the discretized Fredholm integral equation in subsection 2.2.10 is truncating the vector $U^T b$. Here we used a similar idea. Instead of dealing with the matrix, we changed the way vector $U^T b$ is computed in order to obtain a numerically stable algorithm. In exact arithmetic this is equivalent to ordinary matrix–vector multiplication with an orthonormal matrix.*

Remark 3.3.9. *If the procedure defined in Lemma 3.3.7 is performed in exact arithmetic, then $c_{n+1} = 0$, because c_{n+1} is a result of elimination of all the components of b. The components are defined in the orthonormal basis $U = [u_1, \ldots, u_n]$. Thus, if $\tilde{U}$ is not very far from being an orthonormal matrix we can expect $\tilde{c}_{n+1}$ to be very small and negligible. On the other hand, if $\tilde{U}$ is far from being orthonormal, then $\tilde{c}_{n+1}$ has to be taken into account. This will be done in the following lemma.*

Lemma 3.3.10. *Let A, $\tilde{B}$, $\hat{V}$ be as in Theorem 3.2.6, and let $\tilde{y}$ be defined as in Lemma 3.3.7. There exist an $m \times n$ orthonormal matrix $\hat{Q}$ and perturbations $\delta_1 A$, $\delta_1 b$ such that*

$$\tilde{B} = \hat{Q}^T(A + \delta_1 A)\hat{V}, \qquad \tilde{y} = \hat{Q}^T(b + \delta_1 b),$$

and

$$\|\delta_1 A\|_F \leq O(mn + n^3)\varepsilon\|A\|_F, \qquad \|\delta_1 b\|_2 \leq O(mn)\varepsilon\|b\|_2.$$

Proof. Let us define $\hat{P}_i^{(n+1)}$ and $\tilde{P}_i^{(n+1)}$ in the same way as $\hat{P}_i$ and $\tilde{P}_i$ were defined in Theorem 3.2.6, except that vectors e_i are now of dimension $n+1$. Hence, $\hat{P}_i^{(n+1)}$ is a $(m+n+1) \times (m+n+1)$ Householder reflector for $i = 1, \ldots, n$, and $\tilde{P}_i$ is a computed $(m+n+1) \times (m+n+1)$ Householder reflector. First we note that

$$\hat{P}_n^{(n+1)} \cdots \hat{P}_1^{(n+1)} = \begin{bmatrix} \hat{P}_{11}^T & 0 & \hat{P}_{21}^T \\ 0 & 1 & 0 \\ \hat{P}_{12}^T & 0 & \hat{P}_{22}^T \end{bmatrix}, \qquad \tilde{P}_n^{(n+1)} \cdots \tilde{P}_1^{(n+1)} = \begin{bmatrix} \tilde{P}_{11}^T & 0 & \tilde{P}_{21}^T \\ 0 & 1 & 0 \\ \tilde{P}_{12}^T & 0 & \tilde{P}_{22}^T \end{bmatrix}$$

From the Theorem 3.2.6 and Lemma 3.3.7 it follows that

$$\begin{matrix} n \\ 1 \\ m \end{matrix} \left[\begin{array}{c} \tilde{B} \\ 0 \\ 0 \end{array} \right] = \hat{P}_n^{(n+1)} \cdots \hat{P}_1^{(n+1)} \left[\begin{array}{c} \Delta A \\ 0 \\ A + \delta A \end{array} \right] \hat{V}, \qquad \left\| \left[\begin{array}{c} \Delta A \\ \delta A \end{array} \right] \right\|_F \leq O(mn + n^3)\varepsilon \|A\|_F,$$

and

$$\begin{matrix} n \\ 1 \\ m \end{matrix} \left[\begin{array}{c} \tilde{y} \\ 0 \\ \tilde{c}_{n+1} \end{array} \right] = \hat{P}_n^{(n+1)} \cdots \hat{P}_1^{(n+1)} \left[\begin{array}{c} \Delta_0 b \\ 0 \\ b + \delta_0 b \end{array} \right], \qquad \left\| \left[\begin{array}{c} \Delta_0 b \\ \delta_0 b \end{array} \right] \right\|_2 \leq O(mn)\varepsilon \|b\|_2.$$

Next, we define

$$\hat{u}_{n+1} = \frac{\tilde{c}_{n+1}}{\|\tilde{c}_{n+1}\|_2},$$

$$\hat{\gamma}_{n+1} = \hat{u}_{n+1}^T \tilde{c}_{n+1} = \|\tilde{c}_{n+1}\|_2,$$

$$\hat{P}_{n+1}^{(n+1)} = I - \left[\begin{array}{c} -e_{n+1} \\ \hat{u}_{n+1} \end{array} \right] \left[\begin{array}{cc} -e_{n+1}^T & \hat{u}_{n+1}^T \end{array} \right].$$

The Householder reflector $\hat{P}_{n+1}^{(n+1)}$ is chosen so that in exact arithmetic

$$\hat{P}_{n+1}^{(n+1)} \hat{P}_n^{(n+1)} \cdots \hat{P}_1^{(n+1)} \left[\begin{array}{c} \Delta_0 b \\ 0 \\ b + \delta_0 b \end{array} \right] = \hat{P}_{n+1}^{(n+1)} \left[\begin{array}{c} \tilde{y} \\ 0 \\ \tilde{c}_{n+1} \end{array} \right] =$$

$$= \left[\begin{array}{c} \tilde{y} \\ 0 \\ \tilde{c}_{n+1} \end{array} \right] - \hat{\gamma}_{n+1} \left[\begin{array}{c} 0 \\ -1 \\ \hat{u}_{n+1} \end{array} \right] = \left[\begin{array}{c} \tilde{y} \\ \hat{\gamma}_{n+1} \\ 0 \end{array} \right]$$

On the other hand, we can write

$$\left[\begin{array}{c} \tilde{y} \\ \hat{\gamma}_{n+1} \\ 0 \end{array} \right] = \hat{P}_{n+1}^{(n+1)} \hat{P}_n^{(n+1)} \cdots \hat{P}_1^{(n+1)} \left[\begin{array}{c} \Delta^{(n+1)} b \\ b + \delta^{(n+1)} b \end{array} \right], \tag{3.40}$$

where $\Delta^{(n+1)} b = \left[\begin{array}{c} \Delta_0 b \\ 0 \end{array} \right] \in \mathbb{R}^{n+1}$ and $\delta^{(n+1)} b = \delta_0 b$.

The matrix $\left[\begin{array}{c} \tilde{B} \\ 0 \\ 0 \end{array} \right]$ remains unchanged by multiplication with $\hat{P}_{n+1}^{(n+1)}$, so that

$$\left[\begin{array}{c} \tilde{B} \\ 0 \\ 0 \end{array} \right] = \hat{P}_{n+1}^{(n+1)} \left[\begin{array}{c} \tilde{B} \\ 0 \\ 0 \end{array} \right] = \hat{P}_{n+1}^{(n+1)} \hat{P}_n^{(n+1)} \cdots \hat{P}_1^{(n+1)} \left[\begin{array}{c} \Delta A \\ 0 \\ A + \delta A \end{array} \right] \hat{V}. \tag{3.41}$$

If we now define $\hat{P}^{(n+1)} = \hat{P}_1^{(n+1)} \cdots \hat{P}_{n+1}^{(n+1)}$, $\Delta^{(n+1)} A = \left[\begin{array}{c} \Delta A \\ 0 \end{array} \right] \in \mathbb{R}^{(n+1) \times n}$ and $\delta^{(n+1)} A = \delta A$, then we can combine the last two results (3.40) and (3.41) to obtain

$$\left[\begin{array}{cc} \Delta^{(n+1)} A & \Delta^{(n+1)} b \\ A + \delta^{(n+1)} A & b + \delta^{(n+1)} b \end{array} \right] = \hat{P}^{(n+1)} \left[\begin{array}{cc} \tilde{B} \hat{V}^T & \tilde{y} \\ 0 & \hat{\gamma}_{n+1} \\ 0 & 0 \end{array} \right],$$

where

$$\left\| \begin{bmatrix} \Delta^{(n+1)}A \\ \delta^{(n+1)}A \end{bmatrix} \right\|_F \leq O(mn+n^3)\varepsilon\|A\|_F, \qquad \left\| \begin{bmatrix} \Delta^{(n+1)}b \\ \delta^{(n+1)}b \end{bmatrix} \right\|_2 \leq O(mn)\varepsilon\|b\|_2.$$

Again, we have to partition the matrix $\hat{P}^{(n+1)}$ as follows

$$\hat{P}^{(n+1)} = \begin{bmatrix} \hat{P}_{11}^{(n+1)} & \hat{P}_{12}^{(n+1)} \\ \hat{P}_{21}^{(n+1)} & \hat{P}_{22}^{(n+1)} \end{bmatrix},$$

where

$$\hat{P}_{11}^{(n+1)} \in \mathbb{R}^{(n+1)\times(n+1)},\ \hat{P}_{12}^{(n+1)} \in \mathbb{R}^{(n+1)\times m},\ \hat{P}_{21}^{(n+1)} \in \mathbb{R}^{m\times(n+1)},\ \hat{P}_{22}^{(n+1)} \in \mathbb{R}^{m\times m},$$

and then

$$\begin{bmatrix} \Delta^{(n+1)}A & \Delta^{(n+1)}b \\ A+\delta^{(n+1)}A & b+\delta^{(n+1)}b \end{bmatrix} = \begin{bmatrix} \hat{P}_{11}^{(n+1)} \\ \hat{P}_{21}^{(n+1)} \end{bmatrix} \begin{bmatrix} \tilde{B}\hat{V}^T & \tilde{y} \\ 0 & \hat{\gamma}_{n+1} \end{bmatrix}.$$

Again we use Lemma 3.1 from [6] and Step 4 in the proof of Theorem 3.2.6, to conclude that there exist orthogonal matrices $W_1 \in \mathbb{R}^{(n+1)\times(n+1)}$ and $Z_1 \in \mathbb{R}^{(n+1)\times(n+1)}$, an orthonormal matrix $W_2 \in \mathbb{R}^{m\times(n+1)}$, and diagonal matrices $C = \mathrm{diag}(c_1,\ldots,c_{n+1})$ and $S = \mathrm{diag}(s_1,\ldots,s_{n+1})$, such that

$$\begin{aligned} C^2+S^2 &= I_{n+1}, \\ \hat{P}_{11}^{(n+1)} &= \hat{W}_1 C \hat{Z}_1^T, \\ \hat{P}_{12}^{(n+1)} &= \hat{W}_2 S \hat{Z}_1^T. \end{aligned}$$

Further, let us define an orthonormal matrix $\hat{Q}^{(n+1)} = \hat{W}_2\hat{Z}_1 \in \mathbb{R}^{m\times(n+1)}$, then the following holds

$$\begin{bmatrix} A+\delta_1 A & b+\delta_1 b \end{bmatrix} = \hat{Q}^{(n+1)} \begin{bmatrix} \tilde{B}\hat{V}^T & \tilde{y} \\ 0 & \hat{\gamma}_{n+1} \end{bmatrix},$$

where

$$\begin{bmatrix} \delta_1 A & \delta_1 b \end{bmatrix} = \hat{W}_2(I+S)^\dagger \hat{Z}_1^T (\hat{P}_{11}^{(n+1)})^T \begin{bmatrix} \Delta^{(n+1)}A & \Delta^{(n+1)}b \end{bmatrix} + \begin{bmatrix} \delta^{(n+1)}A & \delta^{(n+1)}b \end{bmatrix}$$

Finally, we can conclude that

$$\begin{aligned} \delta_1 A &= \hat{W}_2(I+S)^\dagger \hat{Z}_1^T (\hat{P}_{11}^{(n+1)})^T \Delta^{(n+1)}A + \delta^{(n+1)}A, \\ \delta_1 b &= \hat{W}_2(I+S)^\dagger \hat{Z}_1^T (\hat{P}_{11}^{(n+1)})^T \Delta^{(n+1)}b + \delta^{(n+1)}b, \end{aligned}$$

where

$$\begin{aligned} \|\delta_1 A\|_F &\leq \sqrt{2}\sqrt{\|\Delta^{(n+1)}A\|_F^2 + \|\delta^{(n+1)}A\|_F^2} \leq O(mn+n^3)\varepsilon\|A\|_F, \\ \|\delta_1 b\|_2 &\leq \sqrt{2}\sqrt{\|\Delta^{(n+1)}b\|_2^2 + \|\delta^{(n+1)}b\|_2^2} \leq O(mn)\varepsilon\|b\|_2, \end{aligned}$$

and

$$\begin{aligned} A + \delta_1 A &= \hat{Q}^{(n+1)} \begin{bmatrix} \tilde{B}\hat{V}^T \\ 0 \end{bmatrix}, \qquad \|\delta_1 A\|_F \leq O(mn + n^3)\varepsilon\|A\|_F, \\ b + \delta_1 b &= \hat{Q}^{(n+1)} \begin{bmatrix} \tilde{y} \\ \hat{\gamma}_{n+1} \end{bmatrix}, \qquad \|\delta_1 b\|_2 \leq O(mn)\varepsilon\|b\|_2. \end{aligned}$$

Taking $\hat{Q} = \hat{Q}^{(n+1)}(:, 1:n)$ completes the proof. □

Theorem 3.3.11. *Let*

$$\tilde{x} = \mathrm{fl}(\tilde{V}\tilde{V}_B\tilde{\Sigma}^{\dagger}\tilde{U}_B^T\tilde{y}),$$

where $\tilde{B}$ is the bidiagonal matrix obtained by the Barlow bidiagonalization, $\tilde{U}_B$, $\tilde{V}_B$ and $\tilde{\Sigma}$ are computed SVD factors of $\tilde{B} \approx \tilde{U}_B\tilde{\Sigma}\tilde{V}_B^T$, and $\tilde{y}$ is defined in Lemma 3.3.7. If the bidiagonal SVD is computed by a backward stable algorithm, then there exist orthogonal matrices $\hat{U}_B$ and $\hat{V}_B$, and a backward perturbation $\delta\tilde{B}$ such that

$$\begin{aligned} \tilde{U}_B &= \hat{U}_B + \delta\hat{U}_B, & \|\delta\hat{U}_B\|_F &\leq h(n)\varepsilon \\ \tilde{V}_B &= \hat{V}_B + \delta\hat{V}_B, & \|\delta\hat{V}_B\|_F &\leq h(n)\varepsilon \\ \tilde{B} + \delta\tilde{B} &= \hat{U}_B\tilde{\Sigma}\hat{V}_B^T, & \|\delta\tilde{B}\|_F &\leq g(n)\varepsilon\|A\|_F, \end{aligned}$$

In this case the vector $\tilde{x}$ satisfies

$$\|\tilde{x} - \hat{x}\|_2 \leq (h(n) + O(n^2))\varepsilon\|\hat{x}\|_2,$$

where $\hat{x}$ denotes the minimal norm solution of the problem

$$\min_{x\in\mathbb{R}^n} \|(A + \delta A)x - (b + \delta b)\|_2,$$

where

$$\|\delta A\|_F \leq (g(n) + O(mn + n^3))\varepsilon\|A\|_F, \qquad \|\delta b\|_2 \leq (h(n) + O(mn))\varepsilon\|b\|_2.$$

Proof. First, we should note that by Theorem 3.2.6, Lemma 3.2.4 and numerical analysis of matrix–vector product there exist perturbations $\Delta\hat{U}_B$, $\Delta\hat{V}_B$ and $\Delta\hat{V}$, such that for any vector $z \in \mathbb{R}^n$ the following relations hold.

$$\begin{aligned} \mathrm{fl}(\tilde{U}_B z) &= (\hat{U}_B + \Delta\hat{U}_B)z, & \text{with } \|\delta\hat{U}_B\|_F &\leq (h(n) + O(n^{3/2}))\varepsilon, \\ \mathrm{fl}(\tilde{V}_B z) &= (\hat{V}_B + \Delta\hat{V}_B)z, & \text{with } \|\delta\hat{V}_B\|_F &\leq (h(n) + O(n^{3/2}))\varepsilon, \\ \mathrm{fl}(\tilde{V} z) &= (\hat{V} + \Delta\hat{V})z, & \text{with } \|\delta\hat{V}\|_F &\leq O(n^2)\varepsilon. \end{aligned}$$

Since by Lemma 3.3.10 $\tilde{y} = \hat{Q}^T(b + \delta_1 b)$, we have

$$\tilde{x}^{(1)} = \mathrm{fl}(\tilde{U}_B^T\tilde{y}) = (\hat{U}_B + \Delta\hat{U}_B)^T\hat{Q}^T(b + \delta_1 b) = \hat{U}_B^T\hat{Q}^T(b + \delta b),$$

with

$$\|\delta b\|_2 \leq (h(n) + O(mn))\varepsilon\|b\|_2.$$

Next we have

$$\tilde{x}^{(2)} = \mathrm{fl}(\tilde{\Sigma}^{\dagger}\tilde{x}^{(1)}) = (\tilde{\Sigma} + \delta\tilde{\Sigma})^{\dagger}\hat{U}_B^T\hat{Q}^T(b + \delta b),$$

where

$$|\delta\tilde{\Sigma}| \leq \varepsilon|\tilde{\Sigma}|,$$

followed by

$$\begin{aligned} \tilde{x} &= \mathrm{fl}(\tilde{V}\tilde{V}_B\tilde{x}^{(2)}) = (\hat{V} + \Delta\hat{V})(\hat{V}_B + \Delta\hat{V}_B)(\tilde{\Sigma} + \delta\tilde{\Sigma})^{\dagger}\hat{U}_B^T\hat{Q}^T(b + \delta b) = \\ &= \hat{V}\hat{V}_B(\tilde{\Sigma} + \delta\tilde{\Sigma})^{\dagger}\hat{U}_B^T\hat{Q}^T(b + \delta b) + \delta\hat{x}, \end{aligned}$$

In order to complete the proof we have to define

$$\hat{x} = \hat{V}\hat{V}_B(\tilde{\Sigma} + \delta\tilde{\Sigma})^{\dagger}\hat{U}_B^T\hat{Q}^T(b + \delta b),$$

so that

$$\begin{aligned} \delta\hat{x} &= (\Delta\hat{V}\hat{V}^T + \hat{V}\Delta\hat{V}_B\hat{V}_B^T\hat{V}^T + \Delta\hat{V}\Delta\hat{V}_B\hat{V}_B^T\hat{V}^T)\hat{x}, \\ \|\delta\hat{x}\|_2 &\leq (h(n) + O(n^2))\varepsilon\|\hat{x}\|_2. \end{aligned}$$

On the other hand we can note that $\hat{x}$ is the exact solution of the linear least squares problem with matrix $\bar{A}$ and vector $\bar{b}$, where

$$\bar{A} = A + \delta A = \hat{Q}\hat{U}_B(\tilde{\Sigma} + \delta\tilde{\Sigma})\hat{V}_B^T\hat{V}^T, \quad \text{and} \quad \bar{b} = b + \delta b.$$

By Lemma 3.3.10 and the assumption of this theorem, we can write

$$A + \delta_1 A + \hat{Q}\delta\tilde{B}\hat{V}^T = \hat{Q}\hat{U}_B\tilde{\Sigma}\hat{V}_B^T\hat{V}^T,$$

so that

$$A + \delta_1 A + \hat{Q}\delta\tilde{B}\hat{V}^T + \hat{Q}\hat{U}_B\delta\tilde{\Sigma}\hat{V}_B^T\hat{V}^T = \hat{Q}\hat{U}_B(\tilde{\Sigma} + \delta\tilde{\Sigma})\hat{V}_B^T\hat{V}^T.$$

Thus

$$\begin{aligned} \delta A &= \delta_1 A + \hat{Q}\delta\tilde{B}\hat{V}^T + \hat{Q}\hat{U}_B\delta\tilde{\Sigma}\hat{V}_B^T\hat{V}^T, \quad \text{and} \\ \|\delta A\|_F &\leq (g(n) + O(mn + n^3))\varepsilon\|A\|_F, \end{aligned}$$

which ends the proof. □

To compute the bound on $\|\tilde{x}-x\|_2/\|x\|_2$, where x is the exact solution of $\min_{x\in\mathbb{R}^n}\|Ax-b\|_2$, we need the perturbation theory for the least squares problem. This is illustrated in the following example.

Example 3.3.12. *Let us define the matrix $A \in \mathbb{R}^{10\times 5}$ and the vector $b \in \mathbb{R}^{10}$ as*

$$A = \begin{bmatrix} 2 & 3 & 4 & -1 & 6 \\ 2 & 3 & 4 & -1 & 6 \\ -5 & -5 & 13 & -1 & 8 \\ -7 & -8 & 9 & 0 & 2 \\ -6 & -6 & 11 & -3 & -2 \\ 1 & 2 & 2 & -3 & -4 \\ 6 & -4 & 4 & 5 & -8 \\ 5 & -6 & 2 & 8 & -4 \\ 8 & -9 & 6 & 10 & 10^{10} \\ 3 & -3 & 4 & 2 & 10^{10} \end{bmatrix}, \qquad b = \begin{bmatrix} 15 \\ 13 \\ 10 \\ -4 \\ -6 \\ -2 \\ 3 \\ 5 \\ 10000000011 \\ 10000000006 \end{bmatrix}.$$

The vector b is obtained as $b = Ax + b_1$, where $x = [\ 1\ \ 1\ \ 1\ \ 1\ \ 1\ ]^T$ is the exact solution of the least squares problem with A and b, and $b_1 = [\ 1\ \ -1\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ ]^T$ is such that $b_1^T A = [\ 0\ \ 0\ \ 0\ \ 0\ \ 0\ ]$, that is $b_1 \perp \mathrm{range}(A)$. We compute the solution $\tilde{x}$ in finite precision arithmetic, as described in Lemma 3.3.7 and Theorem 3.3.11 and obtain

$$\tilde{x} = \begin{bmatrix} 1.000000059100661 \\ 9.999999743534302 \cdot 10^{-1} \\ 9.999999537719037 \cdot 10^{-1} \\ 9.999999939451119 \cdot 10^{-1} \\ 9.999999999999998 \cdot 10^{-1} \end{bmatrix}, \qquad \frac{\|\tilde{x} - x\|_2}{\|x\|_2} = 3.556496368936655 \cdot 10^{-8}.$$

For the error bounds we use again the results from [15] and [47], to produce the estimations $g(n) \approx O(n^2)$ and $h(n) \approx O(n^2)$. Theorem 3.3.11 gives the bound on relative error between the computed solution $\tilde{x}$ and the exact solution $\hat{x}$ of a nearby problem:

$$\frac{\|\tilde{x} - \hat{x}\|_2}{\|\hat{x}\|_2} \leq (h(n) + O(n^2))\varepsilon = c_1 \approx 2.775557561562891 \cdot 10^{-15}.$$

On the other hand, Theorem 19.1 in [47] implies that for $\eta = (n^{1/2} g(n) + O(mn^{3/2} + n^{7/2}))\varepsilon$, $\kappa_2(A)\eta < 1$ and $r = b - Ax$

$$\frac{\|\hat{x} - x\|_2}{\|x\|_2} \leq \frac{\kappa_2(A)\eta}{1 - \kappa_2(A)\eta} \left(2 + (\kappa_2(A) + 1) \frac{\|r\|_2}{\|A\|_2 \|x\|_2} \right) = c_2.$$

In our case it is

$$c_2 \approx 7.586838533644896 \cdot 10^{-4}.$$

So, if we put all this together we obtain the final bound

$$3.556496368936655 \cdot 10^{-8} = \frac{\|\tilde{x} - x\|_2}{\|x\|_2} \leq c_1 \frac{\|\hat{x}\|_2}{\|x\|_2} + c_2 \approx 7.586838533672672 \cdot 10^{-4}.$$

3.3.4 The Total Least Squares Problem

Consider the problem

$$\min_{b + r \in \mathrm{range}(A+E)} \|D[\ E\ \ r\ ]T\|_F, \qquad A, E \in \mathbb{R}^{m \times n},\ b, r \in \mathbb{R}^m,\ x \in \mathbb{R}^n, \tag{3.42}$$

where $D = \mathrm{diag}(d_1, \ldots, d_m)$ and $T = \mathrm{diag}(t_1, \ldots, t_{n+1})$ are nonsingular. In exact arithmetic, this problem is solved by the following algorithm (see [35, p. 579]).

Algorithm 3.3.13. *Given $A \in \mathbb{R}^{m \times n}$ $(m \geq n)$, $b \in \mathbb{R}^m$, and nonsingular $D = diag(d_1, \ldots, d_m)$ and $T = diag(t_1, \ldots, t_{n+1})$, the following algorithm computes (if possible) a vector $x \in \mathbb{R}^n$ such that $(A + E_{min})x = (b + r_{min})$, where $\|D[\ E_{min}\ \ r_{min}\ ]T\|_F$ and $\|x\|_\tau = \|T(1:n, 1:n)^{-1} x\|_2$ are minimal.*

Compute the SVD $U^T(D[\ A \quad b\ ]T)W = diag(\sigma_1, \ldots, \sigma_{n+1})$. *Save* W.
Determine p *such that* $\sigma_1 \geq \cdots \geq \sigma_{n-p} > \sigma_{n-p+1} = \cdots = \sigma_{n+1}$.
Compute a Householder matrix P *such that for* $V = WP$, $V(n+1, n-p+1:n) = 0$.

if $V(n+1, n+1) \neq 0$
  for $i = 1 : n$

$$x(i) = -\frac{t_i V(i, n+1)}{t_{n+1} V(n+1, n+1)};$$

  end
end

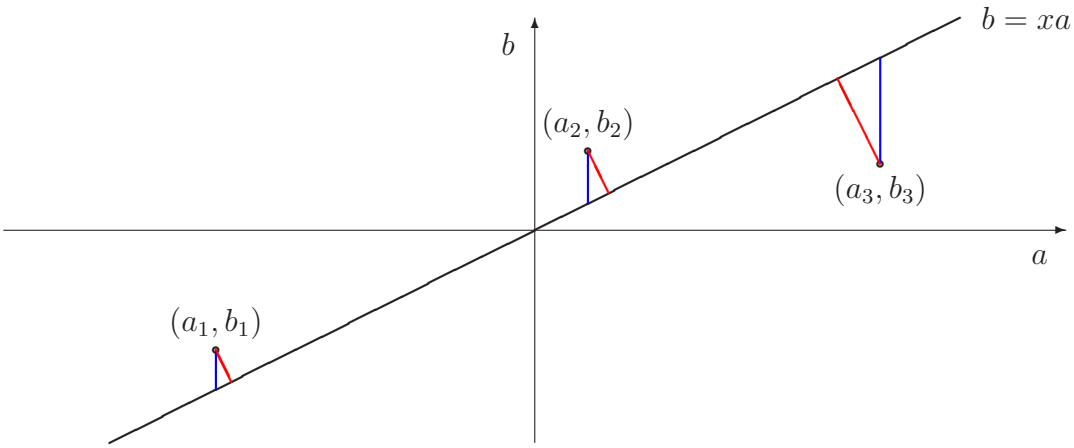


Figure 3.6: Least squares versus total least squares for $n = 1$ and $T = I$. Points $(a_i, b_i) \in \mathbb{R}^2$, $i = 1, \ldots, m$ are fitted by a line $b = xa$ through the origin. Then $A = [a_1, \ldots, a_m]^T \in \mathbb{R}^{m\times 1}$ and $b = [b_1, \ldots, b_m]^T \in \mathbb{R}^m$. In the least squares problem the vertical distance between the points and the line is minimized (—), while in the total least squares problem the perpendicular distance is minimized (—).

Algorithm 3.3.13 uses only the matrix V, and the matrix U is ignored. Again, loss of orthogonality is not an issue here. We have the following numerical results.

Theorem 3.3.14. *Let* $\tilde{x}$ *be a solution of the problem (3.42) computed in finite precision arithmetic, using Algorithm 3.3.13 and Algorithm 3.1.1 for computing the SVD. Then* $\tilde{x}$ *is the exact solution of the problem*

$$\min_{b+\delta b+r \in \ \mathrm{range}(A+\delta A+E)} \|D[\ E \quad r\ ]T\|,$$

where

$$\begin{aligned} \|D[\ \delta A \quad \delta b\ ]T\|_F \ \leq \ & [O(m(n+1) + (n+1)^3) + g(n+1) + \\ & + O(n^{\frac{1}{2}})h(n+1)]\varepsilon\|C\|_F, \end{aligned}$$

and $g(n)\varepsilon$ is the bound on normwise backward error obtained in computing SVD of the $n \times n$ bidiagonal matrix B, and $h(n)\varepsilon$ is the normwise bound on departure from orthogonality of the computed right singular vectors of B.

Proof. Let us define
$$C = D[\ A \quad b\ ]T,$$
so, when the SVD factorization of C is computed in finite precision arithmetic by Barlow's bidiagonalization, we obtain

$$C + \delta_0 C = \hat{U}\tilde{\Sigma}\hat{W}^T, \tag{3.43}$$

where $\hat{U}^T\hat{U} = I$, $\hat{W}^T\hat{W} = I$, $\tilde{\Sigma} = \mathrm{diag}(\tilde{\sigma}_1, \ldots, \tilde{\sigma}_{n+1})$, and

$$\|\delta_0 C\|_F \le (O(m(n+1) + (n+1)^3) + g(n+1))\varepsilon\|C\|_F,$$

If $\tilde{W}$ is the computed matrix, then from Theorem 3.2.6 it follows that

$$\tilde{W} = \hat{W} + \delta\hat{W}, \qquad \|\delta\hat{W}\|_F \le (O((n+1)^2) + h(n+1))\varepsilon.$$

Further, by Lemma 3.2.4

$$\tilde{V} = \mathrm{fl}(\tilde{W}\tilde{P}) = (\tilde{W} + \delta\tilde{W})\hat{P}, \qquad \|\delta\tilde{W}\|_F \le O((n+1)^{\frac{3}{2}})\varepsilon, \quad \hat{P}^2 = I,$$

and if we define
$$\hat{V} = \hat{W}\hat{P}, \qquad \hat{V}^T\hat{V} = I,$$

then we have

$$\begin{aligned} \tilde{V} &= \hat{V} + \delta\hat{V}, \\ \delta\hat{V} &= (\delta\hat{W} + \delta\tilde{W})\hat{P}, \\ \|\delta\hat{V}\|_F &= (O((n+1)^2) + h(n+1))\varepsilon \end{aligned} \tag{3.44}$$

We should note that the matrix $\hat{V} = [\ \hat{v}_1 \quad \hat{v}_2 \quad \ldots \quad \hat{v}_{n+1}\ ]$ is also the matrix of right singular vectors of the matrix $C + \delta_0 C$, and $\hat{U}\hat{P}$ is the matrix of left singular vectors in that case. To illustrate this claim let $\tilde{\sigma}_{n-2} = \tilde{\sigma}_{n-1} = \tilde{\sigma}_n = \tilde{\sigma}_{n+1}$, then Algorithm 3.3.13 performed in finite precision arithmetic will find a Householder reflector $\tilde{P}$, whose exact version $\hat{P}$ is such that

$$\tilde{W}\hat{P} = \begin{array}{|ccccccccc|} \hline \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & 0 & 0 & 0 & \bullet \\ \hline \end{array}.$$

We can partition the matrix $\tilde{W}$ in the following way

$$\tilde{W} = [\tilde{W}_1 \ \tilde{W}_2], \qquad \tilde{W}_2 \in \mathbb{R}^{m\times(p+1)},$$

and we define $\hat{P}$ as

$$\hat{P} = \begin{bmatrix} I_{n-p} & 0 \\ 0 & \hat{P}_2 \end{bmatrix},$$

where $\hat{P}_2 \in \mathbb{R}^{(p+1)\times(p+1)}$ is a Householder reflector such that

$$\tilde{W}_2(n+1, n-p+1:n+1)\hat{P}_2 = [\ 0 \quad \cdots \quad 0 \quad \bullet\ ].$$

On the other hand, the corresponding partition of $\tilde{\Sigma}$ reads

$$\tilde{\Sigma} = \begin{bmatrix} \tilde{\Sigma}_1 & 0 \\ 0 & \tilde{\sigma}_{n+1} I_{p+1} \end{bmatrix}, \qquad \tilde{\Sigma}_1 = \mathrm{diag}(\tilde{\sigma}_1, \ldots, \tilde{\sigma}_{n-p}).$$

Then, from (3.43) it follows that

$$C + \delta_0 C = \hat{U}\tilde{\Sigma}\hat{P}^2\hat{W}^T = \hat{U}\begin{bmatrix} \tilde{\Sigma}_1 & 0 \\ 0 & \tilde{\sigma}_{n+1}\hat{P}_2 \end{bmatrix}\hat{P}\hat{W}^T = \hat{U}\hat{P}\tilde{\Sigma}\hat{V}^T,$$

is a valid singular value decomposition.

For the computed $\tilde{x}$ we have

$$\tilde{x}(i) = \mathrm{fl}\left(-\frac{t_i\tilde{V}(i,n+1)}{t_{n+1}\tilde{V}(n+1,n+1)}\right) = -\frac{t_i\tilde{V}(i,n+1)}{t_{n+1}\tilde{V}(n+1,n+1)}(1+\theta_i),$$

where

$$|\theta_i| \le 3\varepsilon + O(\varepsilon^2).$$

If we define the vector $\bar{v}_{n+1}$ as

$$\bar{v}_{n+1} = \frac{(I+\Theta)\tilde{v}_{n+1}}{\|(I+\Theta)\tilde{v}_{n+1}\|_2}, \tag{3.45}$$

with $\Theta = \mathrm{diag}(\theta_1, \theta_2, \ldots, \theta_n, 0)$ and $\tilde{v}_{n+1}$ being the $(n+1)$-th column of $\tilde{V}$, then we can easily see that

$$\bar{v}_{n+1} = \hat{v}_{n+1} + \delta\bar{v}_{n+1},$$

$$\delta\bar{v}_{n+1} = \delta\hat{v}_{n+1} + \frac{(I+\Theta)\tilde{v}_{n+1} - \|(I+\Theta)\tilde{v}_{n+1}\|_2\tilde{v}_{n+1}}{\|(I+\Theta)\tilde{v}_{n+1}\|_2},$$

where by (3.44)

$$\tilde{v}_{n+1} = \hat{v}_{n+1} + \delta\hat{v}_{n+1}, \qquad \|\delta\hat{v}_{n+1}\|_2 \le (O((n+1)^2) + h(n+1))\varepsilon.$$

To find a bound on $\|\delta\bar{v}_{n+1}\|_2$ we have to perform the following analysis.

$$\begin{aligned}
\delta\bar{v}_{n+1} &= \delta\hat{v}_{n+1} + \frac{[(1-\|(I+\Theta)\tilde{v}_{n+1}\|_2)I+\Theta]\tilde{v}_{n+1}}{\|(I+\Theta)\tilde{v}_{n+1}\|_2} \\
1-\|(I+\Theta)\tilde{v}_{n+1}\|_2 &= \frac{1-(\hat{v}_{n+1}+\delta\hat{v}_{n+1})^T(I+\Theta)^2(\hat{v}_{n+1}+\delta\hat{v}_{n+1})}{1+\|(I+\Theta)\tilde{v}_{n+1}\|_2} \leq \\
&\leq -2\delta\hat{v}_{n+1}^T\hat{v}_{n+1} - \|\delta\hat{v}_{n+1}\|_2^2 - \\
&\quad -(\hat{v}_{n+1}+\delta\hat{v}_{n+1})^T(2\Theta+\Theta^2)(\hat{v}_{n+1}+\delta\hat{v}_{n+1}) \leq \\
&\leq O(1)\|\delta\hat{v}_{n+1}\|_2 + O(1)\varepsilon \\
\|\delta\bar{v}_{n+1}\|_2 &\leq \|\delta\hat{v}_{n+1}\|_2 + \frac{(|1-\|(I+\Theta)\tilde{v}_{n+1}\|_2| + \|\Theta\|_2)(1+\|\delta\hat{v}_{n+1}\|_2)}{1-\|\delta\hat{v}_{n+1}+\Theta(\hat{v}_{n+1}+\delta\hat{v}_{n+1})\|_2} \leq \\
&\leq O(1)\|\delta\hat{v}_{n+1}\|_2 + O(1)\varepsilon,
\end{aligned}$$

thus

$$\|\delta\bar{v}_{n+1}\|_2 \leq (O((n+1)^2) + h(n+1))\varepsilon.$$

After this analysis we can conclude that

$$\tilde{x}(i) = -\frac{t_i e_i^T[(I+\Theta)\tilde{v}_{n+1}]}{t_{n+1}e_{n+1}^T[(I+\Theta)\tilde{v}_{n+1}]} = -\frac{t_i e_i^T\bar{v}_{n+1}}{t_{n+1}e_{n+1}^T\bar{v}_{n+1}} = -\frac{t_i\bar{v}_{n+1}(i)}{t_{n+1}\bar{v}_{n+1}(n+1)}.$$

This means that the computed solution $\tilde{x}$ is an exact solution for some $D[\ A_2 \quad b_2\ ]T$, whose the $(n+1)$-th right singular vector is equal to $\bar{v}_{n+1}$. It remains to determine A_2 and b_2.

Let us define the matrix $\bar{V} = [\ \bar{v}_1 \quad \bar{v}_2 \quad \ldots \quad \bar{v}_{n+1}\ ]$ with

$$\begin{aligned}
\bar{v}_i &= \hat{v}_i, \qquad i = 1,\ldots,n \\
\bar{v}_{n+1} &= \text{as defined above in (3.45)}
\end{aligned}$$

This matrix is almost orthogonal, and close to the matrix of right singular vectors $\hat{V}$ in the SVD factorization of the matrix $C+\delta_0 C$. The next step is finding a QL factorization of the matrix $\bar{V}$, and this can be done by performing the Gram-Schmidt orthogonalization on the matrix $[\ \bar{v}_{n+1} \quad \bar{v}_n \quad \ldots \quad \bar{v}_1\ ]$. Hence, we have to find the elements of an orthogonal matrix $\hat{Z} = [\hat{z}_1,\ldots,\hat{z}_{n+1}]$ and a lower triangular matrix $\hat{L}$ such that $\bar{V} = \hat{Z}\hat{L}$.

We have:

$$\begin{aligned}
\hat{z}_{n+1} &= \bar{v}_{n+1} \\
\hat{L}(n+1,n+1) &= 1 \\
\hat{z}_n &= \hat{v}_n - (\hat{v}_n^T \delta\bar{v}_{n+1})\hat{z}_{n+1} \\
\hat{L}(n,n) &= 1 + O(\varepsilon^2) \\
\hat{L}(n+1,n) &= \hat{v}_n^T \delta\bar{v}_{n+1} \\
\hat{z}_{n-1} &= \hat{v}_{n-1} - O(\varepsilon^2)\hat{z}_n - (\hat{v}_{n-1}^T \delta\bar{v}_{n+1})\hat{z}_{n+1} \\
\hat{L}(n-1,n-1) &= 1 + O(\varepsilon^2) \\
\hat{L}(n,n-1) &= O(\varepsilon^2) \\
\hat{L}(n+1,n-1) &= \hat{v}_{n-1}^T \delta\bar{v}_{n+1} \\
&\vdots \\
\hat{z}_1 &= \hat{v}_1 - O(\varepsilon^2)\hat{z}_2 - \cdots - O(\varepsilon^2)\hat{z}_n - (\hat{v}_1^T \delta\bar{v}_{n+1})\hat{z}_{n+1} \\
\hat{L}(1,1) &= 1 + O(\varepsilon^2) \\
\hat{L}(2,1) &= O(\varepsilon^2) \\
&\vdots \\
\hat{L}(n,1) &= O(\varepsilon^2) \\
\hat{L}(n+1,1) &= \hat{v}_1^T \delta\bar{v}_{n+1}
\end{aligned}$$

hence

$$\begin{bmatrix} \hat{v}_1 & \dots & \hat{v}_n & \bar{v}_{n+1} \end{bmatrix} = \begin{bmatrix} \hat{z}_1 & \dots & \hat{z}_n & \bar{v}_{n+1} \end{bmatrix} \begin{bmatrix} 1+O(\varepsilon^2) & & & \\ & \ddots & & 0 \\ O(\varepsilon^2) & & \ddots & \\ & & & 1+O(\varepsilon^2) \\ \hat{v}_1^T \delta\bar{v}_{n+1} & \dots & \dots & \hat{v}_n^T \delta\bar{v}_{n+1} \quad 1 \end{bmatrix}.$$

Now consider the matrix $\bar{C}$ defined as

$$\bar{C} = \hat{U}\hat{P}\tilde{\Sigma}\bar{V}^T = \hat{U}\hat{P}\tilde{\Sigma}(\hat{V} + \begin{bmatrix} 0 & \delta\bar{v}_{n+1} \end{bmatrix})^T = C + \delta_0 C + \hat{U}\hat{P}\tilde{\Sigma} \begin{bmatrix} 0 \\ \delta\bar{v}_{n+1}^T \end{bmatrix} = C + \delta_0 C + \delta_1 \bar{C},$$

where

$$\|\delta_1 \bar{C}\|_F \leq (O((n+1)^2) + h(n+1))\varepsilon \|C\|_F.$$

On the other hand, we have

$$\begin{aligned}
\bar{C} &= \hat{U}\hat{P}\tilde{\Sigma}\hat{L}^T\hat{Z}^T = \\
&= \hat{U}\hat{P}\tilde{\Sigma}\begin{bmatrix} 1 & & & & \\ & \ddots & & 0 & \\ & & \ddots & & \\ & 0 & & 1 & \\ & & & & 1 \end{bmatrix}\hat{Z}^T + \\
&\quad +\hat{U}\hat{P}\tilde{\Sigma}\begin{bmatrix} O(\varepsilon^2) & & & & \hat{v}_1^T\delta\bar{v}_{n+1} \\ & \ddots & & O(\varepsilon^2) & \vdots \\ 0 & & \ddots & & \vdots \\ & & & O(\varepsilon^2) & \hat{v}_n^T\delta\bar{v}_{n+1} \\ & & & & 0 \end{bmatrix}\hat{Z}^T = \\
&= \hat{U}\hat{P}\tilde{\Sigma}\hat{Z}^T + \delta_2\bar{C},
\end{aligned}$$

where

$$\|\delta_2\bar{C}\|_F \leq (O(n^{\frac{1}{2}}(n+1)^2) + n^{\frac{1}{2}}h(n+1))\varepsilon\|C\|_F.$$

Finally we can write

$$C + \delta C = C + \delta_0 C + \delta_1\bar{C} - \delta_2\bar{C} = \hat{U}\hat{P}\tilde{\Sigma}\hat{Z}^T,$$

with

$$\delta C = D[\ \delta A \quad \delta b\ ]T,$$

and

$$\|\delta C\|_F \leq [O(m(n+1) + (n+1)^3) + g(n+1) + O(n^{\frac{1}{2}})h(n+1)]\varepsilon\|C\|_F.$$

Here we can conclude that $\tilde{x}$ is the exact solution for $C + \delta C$, because this matrix has the $(n+1)$-th right singular vector equal to $\bar{v}_{n+1}$, and its singular values are identical to $\tilde{\sigma}_i$, $i = 1, \ldots n+1$. Thus, $\bar{v}_{n+1}$ will correspond to the smallest singular value. Finally, we can write

$$\delta A = D^{-1}\delta C T^{-1}(:, 1:n), \qquad \delta b = D^{-1}\delta C T^{-1}(:, n+1).$$

□

Remark 3.3.15. *In case when $p > 0$, $C + \delta C$ has a multiple minimal singular value, and the computed solution $\tilde{x}$ does not have to be its exact total least squares solution with minimal $\| \ \|_\tau$ norm, where $\|y\|_\tau = \|T_1^{-1}y\|_2$ and $T_1 = diag(t_1, \ldots, t_n)$. This means, that $\hat{Z}(n+1, n-p+1:n)$ does not have to be equal to 0. By the comment in [35, p. 579] Algorithm 3.3.13 should produce a solution with such a property in exact arithmetic. Therefore we will find $Y = \hat{Z}Q$ as it would be calculated in Algorithm 3.3.13 in exact arithmetic, such that $Y(n+1, n-p+1:n) = 0$.*

First we need to partition the matrix $\hat{Z}$ as

$$\hat{Z} = \begin{bmatrix} \hat{Z}_{11} & \hat{Z}_{12} & \hat{Z}_{13} \\ \hat{Z}_{21} & \hat{Z}_{22} & \hat{Z}_{23} \end{bmatrix} \begin{matrix} n \\ 1 \end{matrix}$$
$$\begin{matrix} n-p & p & 1 \end{matrix}$$

From the definition of the Householder reflector $Q = I - qq^T$ and $Y = \hat{Z}Q$ in Algorithm 3.3.13, we can come to the following conclusion:

$$\begin{aligned}
\begin{bmatrix} Y_{22} & Y_{23} \end{bmatrix} &= \begin{bmatrix} 0 & Y_{23} \end{bmatrix}, \quad \text{where } Y_{23} = \sqrt{\|\hat{Z}_{22}\|_2^2 + Z_{23}^2} \\
q &= \frac{1}{\sqrt{Y_{23}(Y_{23} - \hat{Z}_{23})}} \begin{bmatrix} 0 \\ \hat{Z}_{22}^T \\ \hat{Z}_{23} - Y_{23} \end{bmatrix} \\
Y_{13} &= \frac{\hat{Z}_{12}\hat{Z}_{22} + \hat{Z}_{13}\hat{Z}_{23}}{\sqrt{\|\hat{Z}_{22}\|_2^2 + Z_{23}^2}}
\end{aligned}$$

Now, it is easy to see that Algorithm 3.3.13 applied to the matrix $C + \delta C$ in exact arithmetic will produce the solution

$$\bar{x} = -\frac{T_1 Y_{13}}{t_{n+1} Y_{23}} = -\frac{T_1(\hat{Z}_{12}\hat{Z}_{22}^T + \hat{Z}_{13}\hat{Z}_{23})}{t_{n+1}(\|\hat{Z}_{22}\|_2^2 + \hat{Z}_{23}^2)}.$$

Next we have to estimate how far is this exact solution $\bar{x}$ from our computed solution $\tilde{x}$, where once again

$$\tilde{x} = -\frac{T_1 \hat{Z}_{13}}{t_{n+1}\hat{Z}_{23}}.$$

We have

$$\bar{x} - \tilde{x} = \frac{-\hat{Z}_{23}T_1\hat{Z}_{12}\hat{Z}_{22}^T + \|\hat{Z}_{22}\|_2^2 T_1 \hat{Z}_{13}}{t_{n+1}\hat{Z}_{23}(\|\hat{Z}_{22}\|_2^2 + \hat{Z}_{23}^2)},$$

and

$$\|\bar{x} - \tilde{x}\|_\tau \le \frac{\hat{Z}_{23}\|\hat{Z}_{22}\|_2 + \|\hat{Z}_{22}\|_2^2}{t_{n+1}\hat{Z}_{23}(\|\hat{Z}_{22}\|_2^2 + \hat{Z}_{23}^2)}. \tag{3.46}$$

To find a bound on the right side of relation (3.46), we have to find the bound on $\|\hat{Z}_{22}\|_2$, $\hat{Z}_{22} = [\hat{z}_{n-p+1}(n+1), \ldots, \hat{z}_n(n+1)]$.

From the QL decomposition in the proof of Theorem 3.3.14, it follows that

$$\hat{Z} = \begin{bmatrix} \hat{v}_1 & \ldots & \hat{v}_n & \bar{v}_{n+1} \end{bmatrix} \hat{L}^{-1},$$

where $\hat{L}^{-1}$ is of the form

$$\hat{L}^{-1} = \begin{bmatrix} 1 + O(\varepsilon^2) & & & \\ & \ddots & & 0 \\ O(\varepsilon^2) & & \ddots & \\ & & & 1 + O(\varepsilon^2) \\ O(\eta) & \ldots & \ldots & O(\eta) \quad 1 \end{bmatrix},$$

with $O(\eta) = (O((n+1)^2) + h(n+1))\varepsilon$. Hence

$$\hat{z}_i(n+1) = \hat{v}_i(n+1) + O(\eta), \qquad i = n - p + 1, \ldots, n.$$

On the other hand, by (3.44) we have

$$\begin{aligned}\hat{V}(n+1, n-p+1:n) &= [\hat{v}_{n-p+1}(n+1), \ldots, \hat{v}_n(n+1)] \\ \hat{V}(n+1, n-p+1:n) &= \tilde{V}(n+1, n-p+1:n) - \delta\hat{V}(n+1, n-p+1:n), \\ \|\delta\hat{V}(n+1, n-p+1:n)\|_2 &\leq O(\eta).\end{aligned}$$

So, even when the Algorithm 3.3.13 is performed in finite precision arithmetic, $\tilde{V}(n+1, n-p+1:n)$ *is forced to zero, thus*

$$\begin{aligned}\hat{Z}_{22} &= -\delta\hat{V}(n+1, n-p+1:n) + [O(\eta), \ldots, O(\eta)] \\ \|\hat{Z}_{22}\|_2 &\leq (O(p^{\frac{1}{2}}(n+1)^2) + O(p^{\frac{1}{2}})h(n+1))\varepsilon.\end{aligned}$$

This implies

$$\begin{aligned}\|\bar{x} - \tilde{x}\|_\tau &\leq \frac{\|\hat{Z}_{22}\|_2}{t_{n+1}(\|\hat{Z}_{22}\|_2^2 + \hat{Z}_{23}^2)} + \frac{\|\hat{Z}_{22}\|_2^2}{t_{n+1}\hat{Z}_{23}(\|\hat{Z}_{22}\|_2^2 + \hat{Z}_{23}^2)} \leq \\ &\leq \frac{\|\hat{Z}_{22}\|_2}{t_{n+1}(\hat{Z}_{23}^2 - O(\varepsilon^2))} + O(\varepsilon^2) \leq \\ &\leq \frac{(O(p^{\frac{1}{2}}(n+1)^2) + O(p^{\frac{1}{2}})h(n+1))\varepsilon}{t_{n+1}\hat{Z}_{23}^2}.\end{aligned}$$

Finally, we can be sure that the total least squares solution computed in finite precision arithmetic will produce the solution which is not very far from the exact solution with minimal norm of a slightly perturbed problem.

We will illustrate the result of Theorem 3.3.14 in the following example.

Example 3.3.16. *We generated a matrix* $C \in \mathbb{R}^{10\times 5}$ *with fixed singular values as*

$$C = U\Sigma V^T,$$

where U *is random orthonormal and* V *is random orthogonal, and* $\Sigma = \mathrm{diag}(5, 4, 3, 2, 1)$. *We take* $D = I_{10}$ *and* $T = I_5$, *so that* $A = C(:, 1:4)$ *and* $b = C(:, 5)$. *From Algorithm 3.3.13, the exact solution of the total least squares with* A *and* b *is equal to*

$$x = -\frac{T(1:4, 1:4)V(1:4, 5)}{T(5, 5)V(5, 5)} = \begin{bmatrix} 8.897274820898000 \cdot 10^{-1} \\ 1.271784911649302 \\ -2.256435337898306 \\ -1.222585615762902 \cdot 10^{-1} \end{bmatrix}.$$

We compute the solution $\tilde{x}$ *in finite precision arithmetic using the Barlow bidiagonalization, where the condition* $\sigma_i = \sigma_{n+1}$ *is replaced by* $|\sigma_i - \sigma_{n+1}| \leq \varepsilon\sigma_1$, *and obtain*

$$\tilde{x} = \begin{bmatrix} 8.897274820897999 \cdot 10^{-1} \\ 1.271784911649303 \\ -2.256435337898308 \\ -1.222585615762904 \cdot 10^{-1} \end{bmatrix}, \qquad \frac{\|\tilde{x} - x\|_2}{\|x\|_2} = 7.694579786030378 \cdot 10^{-16}.$$

Again, we use the estimations $g(n) \approx O(n^2)$ *and* $h(n) \approx O(n^2)$. *Theorem 3.3.14 gives the backward error bound*

$$\|D[\ \delta A \quad \delta b\ ]T\|_F \leq \eta,$$

where

$$\eta = [O(m(n+1)+(n+1)^3)+g(n+1)+O(n^{\frac{1}{2}})h(n+1)]\varepsilon\|C\|_F \approx 1.984305870050888\cdot 10^{-13}.$$

On the other hand, Theorem 4.4 in [34] implies that in case when $\sigma_n(DAT_1)-\sigma_{n+1} > 0$ *and* $\eta \leq (\sigma_n(DAT_1)-\sigma_{n+1})/6$, *the following forward bound holds*

$$\begin{aligned}\frac{\|\tilde{x}-x\|_2}{\|x\|_2} &\leq \frac{9\eta\sigma_1}{\sigma_n-\sigma_{n+1}}\left(1+\frac{t_{n+1}\|Db\|_2}{\sigma_n(DAT_1)-\sigma_{n+1}}\right)\frac{1}{t_{n+1}\|Db\|_2-\sigma_{n+1}}\\ &\approx 5.264733890890990\cdot 10^{-11}.\end{aligned}$$

3.4 Efficiency

Another important characteristic of an algorithm is its efficiency, and the best way to evaluate efficiency is through execution time. The execution time of a numerical algorithm depends on two things: floating point operation count and time spent on communication between different levels of memory. We were concerned with the efficiency of full singular value decomposition algorithms which include bidiagonalization, and that means that they compute all of the SVD factors: Y, W and $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_n)$ such that $A = Y\Sigma W^T$.

Let N_B be the floating point operation count required to compute the SVD of the bidiagonal matrix B, then the floating point operation counts for the full SVD algorithms are presented in the following table (see [2]).

LAPACK `sgesvd()` routine		SVD with Algorithm 3.1.1
without QR[1]	with QR[2]	
$8mn^2+\frac{4}{3}n^3+N_B$	$6mn^2+\frac{20}{3}n^3+N_B$	$\mathbf{5mn^2+\frac{10}{3}n^3}+N_B$

Table 3.1: Floating point operation count for SVD algorithms with bidiagonalization.

Table 3.1 shows that Algorithm 3.1.1 requires less operations for computing matrices U, B and V, than the corresponding LAPACK bidiagonalization routine.

Extensive numerical tests were performed to test the efficiency of the SVD algorithms. The computations where performed in the "Advanced Computing Laboratory" of the Department of Mathematics, University of Zagreb. The laboratory consists of 20 computers, connected in a local 1Gb network. The specifications of the computers are shown in Table 3.2.

[1] The Householder algorithm
[2] The Lawson–Hanson–Chan algorithm

2 processors	Athlon Mp 1800+
Frequency	1533MHz
L1 Cache	64Kb
L2 Cache	256Kb
RAM	1Gb

Table 3.2: The specifications of the computers in "Advanced Computing Laboratory".

The computers are working under a Debian GNU/Linux operating system. The tests were written in FORTRAN 77 programming language and GNU (v0.5.24) compiler without optimization was used to obtain executable files. LAPACK and BLAS routines were called in those programs, and single precision was used throughout the tests. Matrices in the tests were generated as product $A = U\Sigma V^T$, where Σ is a diagonal matrix with fixed singular values $\{1, 2, \ldots, n\}$, and U and V are random orthogonal matrices.

The Table 3.3 gives the average execution times for the full SVD algorithms, expressed in seconds.

m	$\times$	n	t_1	t_L	$p_{1,L} = 100(t_L - t_1)/t_L$
100	$\times$	100	0.01	0.01	0.00%
200	$\times$	200	**0.14**	0.15	6.67%
500	$\times$	50	0.01	0.01	0.00%
500	$\times$	100	0.05	**0.04**	-25.00%
500	$\times$	500	3.87	**3.47**	-11.53%
1000	$\times$	100	0.14	**0.09**	-55.56%
1000	$\times$	500	6.19	**4.43**	-39.73%
1000	$\times$	1000	39.19	**36.95**	-6.06%
2000	$\times$	200	1.46	**0.61**	-139.34%
2000	$\times$	1000	55.25	**41.63**	-32,72
2000	$\times$	2000	359.05	**326,75**	-9.89%
3000	$\times$	3000	1514.46	**1300.94**	-16.41%

Table 3.3: Average execution times for full SVD algorithms.

The meaning of the headers in Table 3.3 are as follows:

- t_1 — the SVD with Algorithm 3.1.1 for bidiagonalization. The LAPACK routine `sbdsqr()` is used for the SVD of a bidiagonal matrix, which implements the bidiagonal QR algorithm.
- t_L — the LAPACK `sgesvd()` routine.
- $p_{1,L}$ — the percentage of time decrease, when the SVD with Algorithm 3.1.1 is compared to the LAPACK routine.

Despite the fact that the SVD solver with Algorithm 3.1.1 requires fewer floating point operations, the execution time is longer than the execution time of the LAPACK

[1] routine `sgesvd()`[21]. This happens because the LAPACK routine has optimized fast cache memory usage, while Algorithm 3.1.1 does not. In order to decrease cache communication time, we will develop a block version of Barlow's bidiagonalization in the next section.

3.5 Block Version

First of all, the block version is designed to increase efficiency of the Barlow algorithm, thus it implements the choice $\phi_{k+1} = \gamma_k$ which reduces the floating point operation count. Second, the new block version of Algorithm 3.1.1 improves the usage of fast cache memory. The main idea is to perform as many operations as possible on the data that is currently stored in the cache. In order to do that one has to transform the original algorithm. The first modification of the algorithm is that transformations by Householder reflectors are aggregated, where WY representation is used for a product of Householder reflectors [5]. This means that the matrix A is updated after every b steps, where $m \times b$ is the block dimension. Most of the operations in Algorithm 3.1.1 are matrix-vector operations, coded as BLAS 2 operations [20]. Memory hierarchy is utilized more efficiently if such algorithms are in terms of matrix-matrix operations, coded as BLAS 3 operations [18], [19], or grouped matrix-vector operations, called BLAS 2.5 operations [14], [50]. Employing the WY representation of products of Householder transformations results in more BLAS 3 operations; using the BLAS 2.5 approach of Howell et al. [50] leads to further improvement. Operations on the same data but performed in different places in Algorithm 3.1.1, are now performed simultaneously. These operations are:

$$\boxed{\begin{array}{rcl} x & \leftarrow & x + A^T y \\ w & \leftarrow & Ax \end{array}} \quad \text{or} \quad \boxed{\begin{array}{rcl} A & \leftarrow & A + uv^T \\ x & \leftarrow & A^T y \\ w & \leftarrow & Ax \end{array}} . \tag{3.47}$$

Now we discuss the modifications of Algorithm 3.1.1. As an input to the algorithm we will take the matrix $A \in \mathbb{R}^{m \times n}$, and partition it into block columns. Let $n = b \cdot g + r$, $r \leq b + 1$, where b is a given block column dimension and $g = \lfloor (n-2)/b \rfloor$ is the number of blocks of dimension $m \times b$. We choose the last two columns to be outside of the block partition, because the last two steps of the one–sided bidiagonalization (corresponding to the last two columns) do not involve computation of a Householder reflector. The g blocks will be updated by means of aggregated Householder transformations and BLAS 2.5 transformations related to the first group of transformations in (3.47). The remaining $r = n - b \cdot g$ columns will be updated with non-aggregated Householder transformations and the second group of BLAS 2.5 transformations in (3.47). As each block consists of b columns, the steps of the algorithm will be organized in two loops: the outer loop going through g blocks, and the inner loop going through b columns of the block. Thus we will denote by $A_{j,k}$ the matrix A after the first $j-1$ blocks and the first k columns in the j-th block have been updated.

A block partition for $g = 4$ and $b = 4$, is visualized in Figure 3.7.

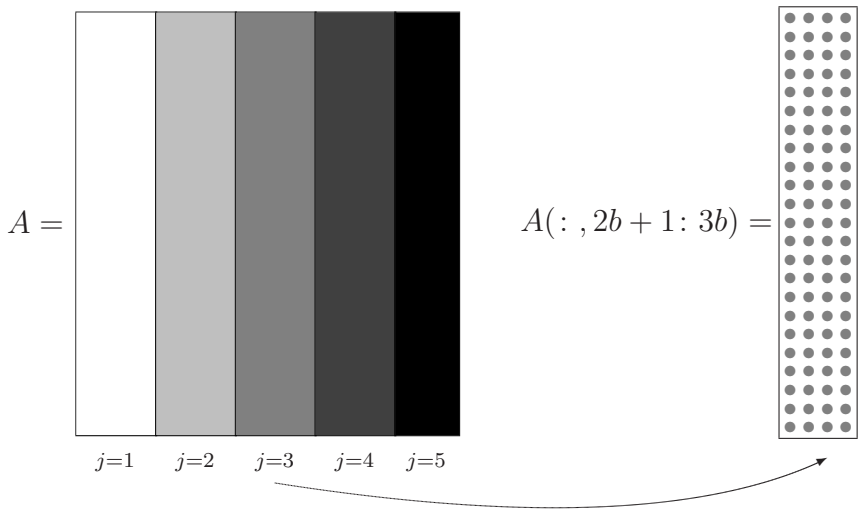


Figure 3.7: Block partition of the matrix A.

The main difference between Barlow's bidiagonalization and its block version is the way Householder reflectors are computed and applied to the matrix A. In the k-th step of Algorithm 3.1.1 columns $k+1$ through n of the matrix A are updated with the Householder reflector $\mathbf{V}_k$. After this step, the $(k+1)$-th column is not changed anymore and is consequently equal to the $(k+1)$-th column of F defined in (3.3).

In the block version of Barlow's bidiagonalization, updates with Householder reflectors are done block-wise. This means, only when all the columns in one block are updated and assigned to F (they will not be modified in the next steps), the rest of the matrix will then be updated with b Householder reflectors in aggregated form, that correspond to b steps of Algorithm 3.1.1. Until then, only the current column is updated. Let us assume that we have computed the first $(j-1)$ blocks of the matrix F obtaining the matrix $A_{j,0}$, and that we are observing the operations in the j-th step of the outer loop. Then for $k = 1, \ldots, b$ only the $((j-1)b+k)$-th column is updated by Householder reflectors from the steps $1, \ldots, k-1$ of the same block, obtaining the matrix $A_{j,k}$, and a new Householder reflector $\mathbf{V}_{(j-1)b+k}$ is computed. $\mathbf{V}_{(j-1)b+k}$ will effect columns $((j-1)b+k+1)$ through n, but no updates are done. The matrix $A_{j,1}$ is equal to $A_{j,0}$ because the $((j-1)b+1)$-th column is already updated, only the Householder reflector $\mathbf{V}_{(j-1)b+1}$ is computed. We use the WY form for a product of Householder reflectors described in [5] to write

$$\mathbf{V}_{(j-1)b+1} \cdots \mathbf{V}_{(j-1)b+k-1} = I - Y_j(:, 1 : k-1) W_j(:, 1 : k-1)^T. \tag{3.48}$$

After the (jb)-th column has been updated, columns $jb+1$ through n are updated with the product $\mathbf{V}_{(j-1)b+1}, \ldots, \mathbf{V}_{jb}$ in WY form (3.48). This process is illustrated in Figure 3.8. The $(g+1)$-th block is updated with Householder reflectors in the usual way, as it is done in Algorithm 3.1.1.

This is the same approach as in the LAPACK routine `sgebrd()` [21], where the routine `slabrd()` is called first, followed by the routine `sgebd2()`. `slabrd()` performs the

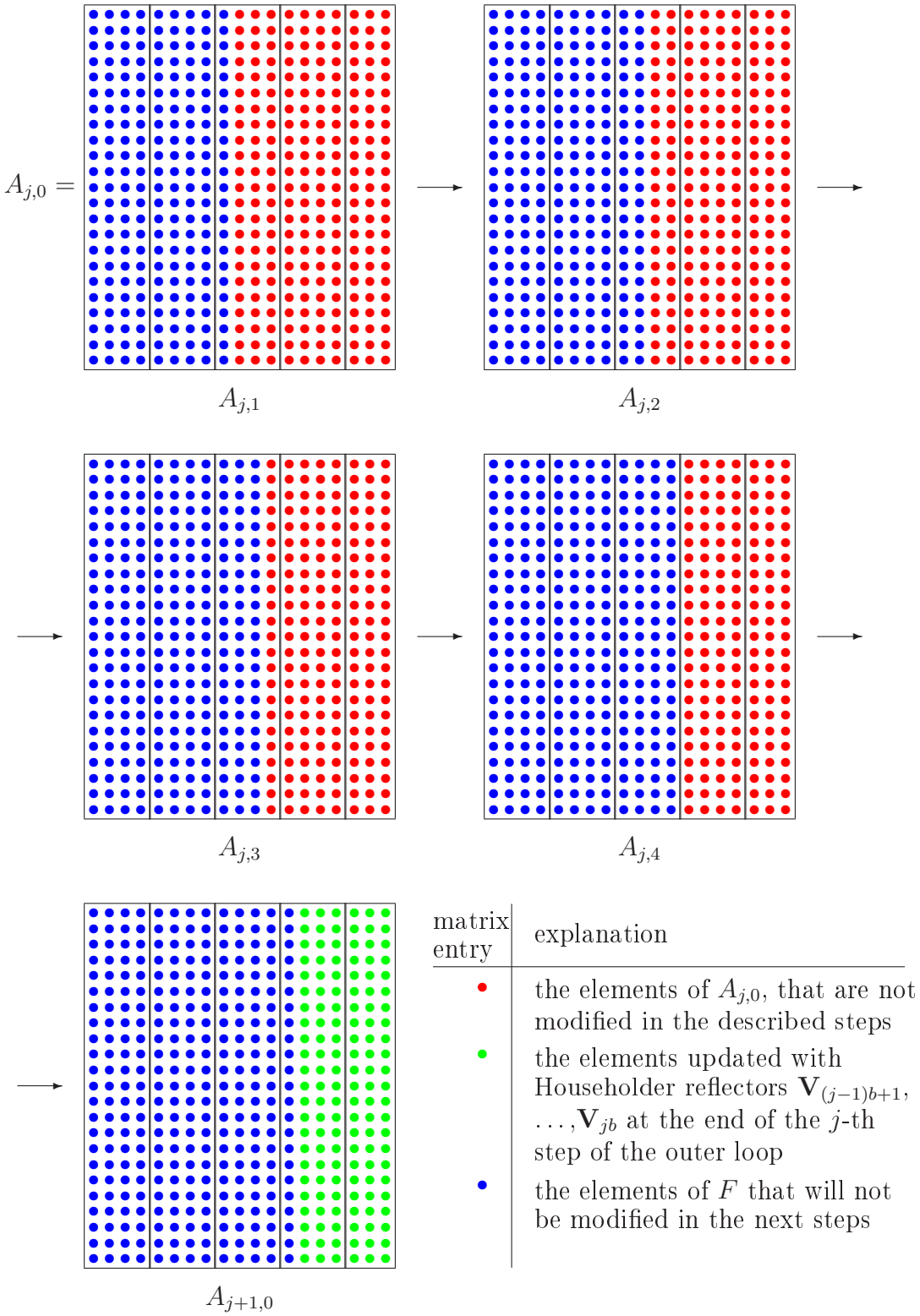


Figure 3.8: Column update in the j-th block of the matrix A.

two–sided aggregated Householder transformation over the first g blocks, and `sgebd2()` performs the unblocked transformations. The only difference is that in the block version of Algorithm 3.1.1 the one–sided Householder transformations are performed, and the dimension of the last block is computed differently.

Aggregated Householder transformations represent only one modification of Algorithm 3.1.1. The other modification is achieved by using the ideas described in [50]. Let us define the following correspondence:

$$\begin{array}{lcll} \ell & = & (j-1)b+k, & \text{current } \ell\text{-th column is the } k\text{-th column in the } j\text{-th block,} \\ \ell & \leftrightarrow & (j,k) & \text{the indices with } \ell \text{ are replaced by } (j,k). \end{array}$$

This correspondence is introduced only for notational convenience. Now we will investigate lines **(4)**, **(5)** and **(7)** in Algorithm 3.1.1, but with the index k replaced by ℓ. In all these statements the vector $z_\ell \rightarrow z_{j,k}$ is directly or indirectly used. In line **(4)** u_ℓ is multiplied by $A_{\ell-1}(:,\ell+1\colon n)^T \rightarrow A_{j,k-1}(:,\ell+1\colon n)^T$ in order to obtain $z_{j,k}$. On the other hand in line **(7)** the vector $v_\ell \rightarrow v_{j,k}$ is multiplied by $A_{j,k-1}(:,\ell+1\colon n)$, and $v_{j,k}$ is realized from $z_{j,k}$ through line **(5)** and the function **householder**(). From the definition of the function **householder**() we have

$$\begin{aligned} z_{j,k} &= A_{j,k-1}(:,\ell+1,n)^T u_\ell \\ v_{j,k} &= \frac{\sqrt{2}(z_{j,k}-\phi_{\ell+1}e_1)}{\|z_{j,k}-\phi_{\ell+1}e_1\|_2}, \quad \text{thus} \\ A_{j,k-1}(:,\ell+1\colon n)v_{j,k} &= \frac{\sqrt{2}[A_{j,k-1}(:,\ell+1\colon n)z_{j,k}-\phi_{\ell+1}A_{j,k-1}(:,\ell+1)]}{\|z_{j,k}-\phi_{\ell+1}e_1\|_2} \end{aligned} \tag{3.49}$$

From the previous observations concerning the update of the matrix $A_{j,0}$ with Householder reflectors, in the ℓ-th step (which in the block version will correspond to the j-th step of the outer loop and the k-th step of the inner loop) columns $\ell+1,\ldots,n$ are not yet updated. $A_{j,k-1}$ should be equal to $A_{j,0}\mathbf{V}_{j,1}\cdots\mathbf{V}_{j,k-1}$, hence from (3.48) and (3.49) it follows

$$\begin{aligned} z_{j,k} &= A_{j,0}(:,\ell+1,n)^T u_\ell - \\ &\quad -W_j(\ell+1\colon n,1\colon k-1)Y_j(:,1\colon k-1)^T A_{j,0}^T u_\ell \qquad (3.50) \\ A_{j,k-1}(:,\ell+1\colon n)v_{j,k} &= A_{j,0}(:,\ell+1\colon n)v_{j,k} - \\ &\quad -A_{j,0}Y_j(:,1\colon k-1)W_j(\ell+1\colon n,1\colon k-1)^T v_{j,k} = \\ &= \frac{\sqrt{2}[A_{j,0}(:,\ell+1\colon n)z_{j,k}-\phi_{\ell+1}A_{j,0}(:,\ell+1)]}{\|z_{j,k}-\phi_{\ell+1}e_1\|_2} - \\ &\quad -A_{j,0}Y_j(:,1\colon k-1)W_j(\ell+1\colon n,1\colon k-1)^T v_{j,k} \qquad (3.51) \end{aligned}$$

If we define

$$z_{j,k}^{(1)} = -W_j(\ell+1\colon n,1\colon k-1)Y_j(:,1\colon k-1)^T A_{j,0}^T u_\ell,$$

as the first phase in the computation of $z_{j,k}$, and

$$x_{j,k}^{(1)} = A_{j,0}(:,\ell+1\colon n)z_{j,k}$$

as the first phase in the computation of the vector $x_{j,k}^{(4)} = A_{j,k-1}(:,\ell+1:n)v_{j,k}$, then

$$\boxed{\begin{array}{rcll} z_{j,k} & = & z_{j,k}^{(1)} + A_{j,0}(:,\ell+1:n)^T u_\ell & \text{from (3.50)} \\ x_{j,k}^{(1)} & = & A_{j,0}(:,\ell+1:n) z_{j,k} & \text{from (3.51)} \end{array}}$$

will be computed simultaneously and they comprise the first group of BLAS 2.5 transformations in (3.47). By simultaneous computation we mean that as soon as one component of $z_{j,k}$ is computed, $x_{j,k}^{(1)}$ is updated with this new data by the BLAS 1 `saxpy` operation. The components of $z_{j,k}$ can be partitioned in blocks of dimension c, so that BLAS 2 `segmv` is used in the simultaneous computation instead of BLAS 1 operations. This would improve the cache memory usage even more.

In the k-th step of the inner loop for the last $(g+1)$-th block update with $\mathbf{V}_{g+1,k-1}$, the computation of $z_{g+1,k}$ and $x_{g+1,k}^{(1)}$ will be done simultaneously. Let again $\ell = gb+k$. First, we have

$$\begin{aligned} A_{g+1,k-1}(:,\ell+1:n) &= A_{g+1,k-2}(:,\ell+1:n) - \\ &\quad - A_{g+1,k-2}(:,\ell:n) v_{g+1,k-1} v_{g+1,k-1}(2:n-\ell+1)^T, \\ &= A_{g+1,k-2}(:,\ell+1:n) - x_{g+1,k-1}^{(3)} v_{g+1,k-1}(2:n-\ell+1)^T \end{aligned}$$

where $x_{g+1,k}^{(3)} = A_{g+1,k-1}(:,\ell+1:n)v_{g+1,k}$, and from (3.49) it follows

$$\begin{aligned} &A_{g+1,k-1}(:,\ell+1:n)v_{g+1,k} = \\ &= \frac{\sqrt{2}[A_{g+1,k-1}(:,\ell+1:n)z_{g+1,k} - \phi_{\ell+1} A_{g+1,k-1}(:,\ell+1)]}{\|z_{g+1,k} - \phi_{\ell+1} e_1\|_2}. \end{aligned} \tag{3.52}$$

Again, if we define

$$x_{g+1,k}^{(1)} = A_{g+1,k-1}(:,\ell+1:n) z_{g+1,k}$$

as the first phase in the computation of the vector $x_{g+1,k}^{(3)}$, then

$$\boxed{\begin{array}{rcl} A_{g+1,k-1}(:,\ell+1:n) & = & A_{g+1,k-2}(:,\ell+1:n) - \\ & & - x_{g+1,k-1}^{(3)} v_{g+1,k-1}(2:n-\ell+1)^T \\ z_{g+1,k} & = & A_{g+1,k-1}(:,\ell+1,n)^T u_\ell \\ x_{g+1,k}^{(1)} & = & A_{g+1,k-1}(:,\ell+1:n) z_{g+1,k} \end{array}}$$

comprises the second group of BLAS 2.5 transformations in (3.47).

The reason why these operations are performed simultaneously is that the same parts of the matrix A are involved, as well as the same parts of the vector $z_{j,k}$. So, when a particular block of the matrix and the vector is stored in the fast cache memory, all the operations can be done without transferring blocks from slower memory to cache. This will save some time spent on memory transfer in Algorithm 3.1.1.

Details of the block algorithm

The following operations are performed on each block $j = 1, \ldots, g$:

- In step $k = 1, \ldots, b$, only the ℓ-th column of $A_{j,k-1}$ is updated with aggregated Householder transformations computed in steps $i = 1, \ldots, k-1$ of the inner loop:

$$\begin{aligned} A_{j,k}(:,\ell) &= A_{j,k-1}(:,\ell) = [A_{j,0}\mathbf{V}_{j,1}\cdots\mathbf{V}_{j,k-1}](:,\ell) = \\ &= A_{j,0}(:,\ell) - A_{j,0}Y_j(:,1\colon k-1)W_j(k,1\colon k-1)^T, \end{aligned}$$

where $\ell = (j-1)b+k$. Here $\mathbf{V}_{j,i} = I - \mathbf{v}_{j,i}\mathbf{v}_{j,i}^T$ is the Householder reflector obtained in the i-th inner loop step within the j-th outer loop step, where $\mathbf{v}_{j,i} = [0,\ v_{j,i}]$, $v_{j,i} \in \mathbb{R}^{n-(j-1)b-i}$. We can note that $\mathbf{V}_{j,i} = I_{(j-1)b+i} \oplus V_{j,i}$ and $V_{j,i} = I - v_{j,i}v_{j,i}^T$. For the aggregated Householder transformations the following statements hold (see [5]).

$$\begin{aligned} \mathbf{V}_{j,1}\cdots\mathbf{V}_{j,k-1} &= I - Y_j(:,1\colon k-1)W_j(:,1\colon k-1)^T, \\ A_{j,k-1} &= A_{j,0}\mathbf{V}_{j,1}\cdots\mathbf{V}_{j,k-1} = A_{j,0} - A_{j,0}Y_j(:,1\colon k-1)W_j(:,1\colon k-1)^T, \end{aligned}$$

The term $A_{j,0}Y_j(:,1\colon k-1)$ is also occurring in relations (3.50) and (3.51), hence we define $X_j = A_{j,0}Y_j$. From the definition of the matrices Y_j and W_j in [5], W_j, Y_j and X_j satisfy the following recurrences

$$\begin{aligned} W_j(:,1) &= \mathbf{v}_{j,1}, \\ Y_j(:,1) &= \mathbf{v}_{j,1}, \\ X_j(:,1) &= A_{j,0}Y_j(:,1) = A_{j,0}\mathbf{v}_{j,1}, \\ W_j(:,1\colon k) &= [W_j(:,1\colon k-1),\ \mathbf{v}_{j,k}], \\ Y_j(:,1\colon k) &= [Y_j(:,1\colon k-1),\ \mathbf{V}_{j,1}\cdots\mathbf{V}_{j,k-1}\mathbf{v}_{j,k}] = \\ &= [Y_j(:,1\colon k-1),\ \mathbf{v}_{j,k} - Y_j(:,1\colon k-1)W_j(:,j\colon k-1)^T\mathbf{v}_{j,k}], \\ X_j(:,1\colon k) &= A_{j,0}Y_j(:,1\colon k) = \\ &= [X_j(:,1\colon k-1),\ A_{j,0}\mathbf{v}_{j,k} - X_j(:,1\colon k-1)W_j(:,1\colon k-1)^T\mathbf{v}_{j,k}]. \end{aligned} \tag{3.53}$$

- u_ℓ is produced from orthogonalization of $A_{j,k}(:,\ell)$ against $u_{\ell-1}$, and normalization (a Gram–Schmidt step).

- $W_j(:,k) = v_{j,k}$ and $X_j(:,k)$ are computed using BLAS 2.5 as follows. The vector $z_{j,k}$ and the vector $x_{j,k}^{(1)}$ are computed simultaneously, where $x_{j,k}^{(1)}$ is the first step in obtaining $X_j(:,k)$. First we have to set

$$z_{j,k}^{(1)} = -W_j(\ell+1\colon n, 1\colon k-1)X_j(:,1\colon k-1)^T u_\ell, \qquad x_{j,k}^{(1)} = 0,$$

and then for $i = \ell+1, \ldots, n$ we can compute

$$\begin{aligned} z_{j,k}(i-\ell) &= z_{j,k}^{(1)}(i-\ell) + A_{j,0}(:,i)^T u_\ell \\ x_{j,k}^{(1)} &= x_{j,k}^{(1)} + A_{j,0}(:,i)z_{j,k}(i-\ell). \end{aligned}$$

The components of $z_{j,k}$ are partitioned in blocks of dimension c, and as soon as one block of $z_{j,k}$ is computed, $x_{j,k}^{(1)}$ is updated with this new data. When $z_{j,k}$ is

finished $v_{j,k}$ is chosen in the same way as in Algorithm 3.1.1, and immediately after that $x^{(2)}_{j,k}$ and $x^{(3)}_{j,k}$ are computed so that

$$x^{(3)}_{j,k} = A_{j,0}(:\,, \ell+1\colon n)v_{j,k}.$$

$x^{(3)}_{j,k}$ is obtained from $x^{(1)}_{j,k}$ and $\phi_{\ell+1} = \pm\|z_{j,k}\|_2$ in the same way as $v_{j,k}$ is obtained from $z_{j,k}$ and $\phi_{\ell+1}$. Finally (see (3.53)),

$$X_j(:\,,k) = x^{(4)}_{j,k} = x^{(3)}_{j,k} - X_j(:\,,1\colon k-1)W_j(\ell+1\colon n, 1\colon k-1)^T v_{j,k}.$$

- After b steps, the rest of the matrix $A_{j,b}$ is updated with $\mathbf{V}_{j,1}\cdots\mathbf{V}_{j,b}$:

$$\begin{aligned} A_{j+1,0}(:\,,1\colon jb) &= A_{j,b}(:\,,1\colon jb), \\ A_{j+1,0}(:\,,jb+1\colon n) &= A_{j,b}(:\,,jb+1\colon n) - X_j(:\,,1\colon b)W_j(jb+1\colon n, 1\colon b)^T. \end{aligned}$$

For the last $(g+1)$-th $m \times r$ block we use a similar technique, except that in each step the whole matrix is updated. The following operations are performed on the last block of dimension $m \times r$:

- For steps $k = 1, \ldots, r$, the ℓ-th column of $A_{g+1,k-1}$ is updated with $\mathbf{V}_{g+1,k-1}$:

$$\begin{aligned} A_{g+1,k}(:\,,\ell) &= A_{g+1,k-1}(:\,,\ell) = \\ &= A_{g+1,k-2}(:\,,\ell) - v_{g+1,k-1}(1)x^{(3)}_{g+1,k-1}, \end{aligned}$$

  where $\ell = gb + k$, and

$$x^{(3)}_{g+1,k-1} = A_{g+1,k-2}(:\,,\ell\colon n)v_{g+1,k-1},$$

  and is computed in the previous step.

- u_ℓ is produced from orthogonalization of $A_{g+1,k}(:\,,\ell)$ against $u_{\ell-1}$, and normalization (a Gram–Schmidt step).

- The update of the rest of the matrix $A_{g+1,k-1}(:\,,\ell+1\colon n)$, and the computations of $v_{g+1,k}$ and $x^{(3)}_{g+1,k} = A_{g+1,k-1}(:\,,\ell+1\colon n)v_{g+1,k}$ are performed using BLAS 2.5. Namely, $A_{g+1,k-1}(:\,,\ell+1\colon n)$, $z_{g+1,k}$ and $x^{(1)}_{g+1,k}$ are computed simultaneously in the following way: $x^{(1)}_{g+1,k} = 0$, then for $i = \ell+1, \ldots, n$ do

$$\begin{aligned} A_{g+1,k-1}(:\,,i) &= A_{g+1,k-2}(:\,,i) - v_{g+1,k-1}(i-\ell+1)x^{(3)}_{g+1,k-1} \\ z_{g+1,k}(i-\ell) &= A_{g+1,k-1}(:\,,i)^T u_\ell \\ x^{(1)}_{g+1,k} &= x^{(1)}_{g+1,k} + A_{g+1,k-1}(:\,,i)z_{g+1,k}(i-\ell) \end{aligned}$$

  As soon as the i-th column of $A_{g+1,k-1}$ is computed, the proper component of $z_{g+1,k}$ is computed, and $x^{(1)}_{g+1,k}$ is updated with this new data. When $z_{g+1,k}$ is finished $v_{g+1,k}$ is chosen in the same way as in Algorithm 3.1.1, and $x^{(3)}_{g+1,k}$ is computed.

Now we can state the complete algorithm.

Algorithm 3.5.1 (The block Barlow one–sided bidiagonalization). *For* $A \in \mathbf{R}^{m\times n}$, $\mathrm{rank}(A) = n > 2$, *this algorithm computes orthonormal* U, *bidiagonal* B *and orthogonal* V *such that* $A = UBV^T$.

Initialize:
the block dimension for aggregated Householder transformations b;
the block dimension for BLAS 2.5 transformations c;
$A_{1,0} = A$;
$s_1 = A_{1,0}(:\,,1)$;
$g = \lfloor (n-2)/b \rfloor$;
for $j = 1\colon g$

{*Update the* j*-th block of the matrix* A *with aggregated Householder transformations and the first group of BLAS 2.5 transformation from (3.47).*}
$X_j = 0_{m\times b}$; $W_j = 0_{n\times b}$;
for $k = 1\colon b$
$\ell = (j-1)b + k$;
$A_{j,k}(:\,,1\colon \ell-1) = A_{j,k-1}(:\,,1\colon \ell-1)$;
if $k > 1$
$A_{j,k}(:\,,\ell) = A_{j,0}(:\,,\ell) - X_j(:\,,1\colon k-1)W_j(\ell,1\colon k-1)^T$;
$s_\ell = A_{j,k}(:\,,\ell) - \phi_\ell u_{\ell-1}$;
else
$A_{j,k}(:\,,\ell) = A_{j,k-1}(:\,,\ell)$;
end;
$\psi_\ell = \|s_\ell\|_2$;
$u_\ell = s_\ell/\psi_\ell$;
if $k > 1$
$z_{j,k}^{(1)} = -W_j(\ell+1\colon n, 1\colon k-1)X_j(:\,,1\colon k-1)^T u_\ell$;
else
$z_{j,k}^{(1)} = 0_{(n-\ell)\times 1}$;
end;
$x_{j,k}^{(1)} = 0_{m\times 1}$;
for $i = \ell+1\colon c\colon n$
$d = \min(c, n-i+1)$;
$z_{j,k}(i-\ell\colon i-\ell+d-1) = z_{j,k}^{(1)}(i-\ell\colon i-\ell+d-1) + A_{j,0}(:\,,i\colon i+d-1)^T u_\ell$;
$x_{j,k}^{(1)} = x_{j,k}^{(1)} + A_{j,0}(:\,,i\colon i+d-1)z_{j,k}(i-\ell\colon i-\ell+d-1)$;
end;
$[\phi_{\ell+1}, v_{j,k}, x_{j,k}^{(3)}] =$ **householder** $2(z_{j,k}, x_{j,k}^{(1)}, A_{j,0}(:\,,\ell+1))$;
$W_j(\ell+1\colon n, k) = v_{j,k}$;
$x_{j,k}^{(4)} = x_{j,k}^{(3)} - X_j(:\,,1\colon k-1)W_j(\ell+1\colon n, 1\colon k-1)^T v_{j,k}$;
$X_j(:\,,k) = x_{j,k}^{(4)}$;

---

end;

{*Update the rest of the matrix A with aggregated Householder transformations from the j-th block.*}
$A_{j+1,0}(:,1:jb) = A_{j,b}(:,1:jb)$;
$A_{j+1,0}(:,jb+1:n) = A_{j,b}(:,jb+1:n) - X_j W_j(jb+1:n,:)^T$;
$s_{jb+1} = A_{j+1,0}(:,jb+1) - \phi_{jb+1} u_{jb}$;

---

end;
$r = n - gb$;

{*Update the last block of the matrix A via the second group of BLAS 2.5 transformation from (3.47).*}
for $k = 1:r-1$
  $\ell = gb + k$;
  if $k > 1$
    $A_{g+1,k}(:,1:\ell-1) = A_{g+1,k-1}(:,1:\ell-1)$;
    $A_{g+1,k}(:,\ell) = A_{g+1,k-1}(:,\ell) - v_{g+1,k-1}(1) x^{(3)}_{gb+1,k-1}$;
    $s_\ell = A_{g+1,k}(:,\ell) - \phi_\ell u_{\ell-1}$;
  else
    $A_{g+1,k}(:,1:\ell) = A_{g+1,k-1}(:,1:\ell)$;
  end;
  $\psi_\ell = \|s_\ell\|_2$;
  $u_\ell = s_\ell / \psi_\ell$;
  $x^{(1)}_{g+1,k} = 0_{m\times 1}$;
  for $i = \ell+1:n$
    if $k > 1$
      $A_{g+1,k-1}(:,i) = A_{g+1,k-2}(:,i) - v_{g+1,k-1}(i-\ell+1) x^{(3)}_{g+1,k-1}$;
    end;
    if $\ell < n-1$
      $z_{g+1,k}(i-\ell) = A_{g+1,k-1}(:,i)^T u_\ell$;
      $x^{(1)}_{g+1,k} = x^{(1)}_{g+1,k} + z_{g+1,k}(i-\ell) A_{g+1,k-1}(:,i)$;
    end;
  end;
  if $\ell < n-1$
    $[\phi_{\ell+1}, v_{g+1,k}, x^{(3)}_{g+1,k}] =$ **householder** $2(z_{g+1,k}, x^{(1)}_{g+1,k}, A_{g+1,k-1}(:,\ell+1))$;
  end ;
end;
$\phi_n = u_{n-1}^T A_{g+1,r-1}(:,n)$;
$s_n = A_{g+1,r-1}(:,n) - \phi_n u_{n-1}$;
$\psi_n = \|s_n\|_2$;
$u_n = s_n / \psi_n$;

$V^T =$ **householder_product**$(v_{1,1}, \ldots, v_{g+1,n-2})$;

The auxiliary function **householder** $2()$ is defined as follows.

function $[\phi, v, y]$=**householder** $2(z, x, b)$
{*The function* **householder** $2()$ *computes* ϕ, v *and* y *such that* $Vz = \phi e_1$ *and* $y = Bv$, *where* $V = I - vv^T$ *is a Householder reflector,* $x = Bz$ *and* $b = Be_1$.}

$n = \text{length}(z)$;
$\phi = \|z\|_2$;
if $\phi > 0$
  $\phi = -\text{sign}(z(1))\phi$
  $t(1) = z(1) - \phi$;
  $t(2\colon n) = z(2\colon n)$;
  $v = \sqrt{2}t/\|t\|_2$;
  $w = x - \phi b$;
  $y = \sqrt{2}w/\|t\|_2$;
else
  $v = 0$;
end

Remark 3.5.2. *The vector* w *in the function* **householder** $2()$ *stands for* $x^{(2)}_{j,k}$.

Remark 3.5.3. *The choice of the block dimensions* b *and* c *depends on the computer which executes Algorithm 3.5.1. Their sizes are chosen to obtain optimal efficiency. In LAPACK routines, the function* `ilaenv()` *is used to determine the optimal block size for block algorithms. The section* Determining the Block Size for Block Algorithms *of [1] explains how* `ilaenv()` *works: "The version of* `ilaenv()` *supplied with the package contains default values that led to good behavior over a reasonable number of the test machines, but to achieve optimal performance, it may be beneficial to tune* `ilaenv()` *for the particular machine environment." Our optimal block dimensions were obtained through tests.*

3.6 Numerical Stability of the Block Version

Algorithm 3.1.1 is numerically backward stable, but what about Algorithm 3.5.1? The answer to this question is given by Theorem 3.6.4. Before stating a proof of Theorem 3.6.4 we will need results of three technical lemmas. The lemmas are based on the numerical analysis of basic numerical algorithms given by Higham [47], and the analysis of the modified Gram–Schmidt algorithm given by Björck and Paige [6]. In our numerical analysis we will use the following notation once again: tildes ($\tilde{}$) will mark computed quantities, and hats ($\hat{}$) will denote vectors and matrices that correspond to certain exact relations and exist only as theoretical entities, not actually computed.

Lemma 3.6.1. *When Algorithm 3.5.1 is executed in finite precision arithmetic with the*

unit roundoff error ε then the computed values are of the following form:

$$\begin{aligned}
\tilde{v}_{j,k} &= \hat{v}_{j,k} + \delta\hat{v}_{j,k}, \qquad \|\delta\hat{v}_{j,k}\|_2 \leq O(n-l)\varepsilon \\
\tilde{W}_j(:,1\colon k) &= \hat{W}_j(:,1\colon k) + \delta\hat{W}_j(:,1\colon k), \qquad \|\delta\hat{W}_j(:,1\colon k)\|_F \leq O(\sqrt{k}n)\varepsilon \\
\tilde{X}_j(:,k) &= \tilde{A}_{j,0}\hat{Y}_j(:,k) + \delta\hat{X}_j(:,k), \qquad \|\delta\hat{X}_j(:,k)\|_2 \leq O(kn)\varepsilon\|\tilde{A}_{j,0}\|_F,
\end{aligned}$$

where $\hat{v}_{j,k}$ define exact Householder reflectors $\hat{V}_{j,k}$, and $\hat{W}_j$, $\hat{Y}_j$ and $\hat{X}_j$ are exact matrices that are related to Householder vectors $\hat{v}_{j,k}$ as described in (3.53). Further, the exact values can be estimated as

$$\begin{aligned}
\|\hat{W}_j(:,1\colon k)\|_F &\leq \sqrt{2}\sqrt{k} \\
\|\hat{X}_j(:,1\colon k)\|_F &= \|\tilde{A}_{j,0}\hat{Y}_j(:,1\colon k)\|_F \leq 2\sqrt{2}\sqrt{k}\|\tilde{A}_{j,0}\|_F
\end{aligned}$$

Proof. The proof follows the execution of Algorithm 3.5.1.

1. First, we will start with the computation of $A_{j,0}(:,\ell+1\colon n)v_{j,k}$ as shown in Algorithm 3.5.1:

$$\begin{aligned}
x^{(3)}_{j,k} &= A_{j,0}(:,\ell+1\colon n)v_{j,k} = \\
&= \sqrt{2}[A_{j,0}(:,\ell+1\colon n)z_{j,k} + \mathrm{sign}(z_{j,k}(1))\|z_{j,k}\|_2 A_{j,0}(:,\ell+1)]/\|t_{j,k}\|_2,
\end{aligned}$$

   where $\ell = (j-1)b + k$. Here we can note that the computation of $A_{j,0}(:,\ell+1\colon n)v_{j,k}$ is parallel to the computation of $v_{j,k}$ (see function **householder** 2()), and not performed as an application of the submatrix of $A_{j,0}$ on already computed $v_{j,k}$. In our FORTRAN code which implements Algorithm 3.5.1, we used the routine `slarfg2()`, a modification of the LAPACK routine `slarfg()`. The routine `slarfg()` generates a Householder vector $v_{j,k}$, and `slarfg2()` generates both: $v_{j,k}$ and $A_{j,0}(:,\ell+1\colon n)v_{j,k}$. This routine obtains a slightly different result from that shown in Algorithm 3.5.1, but the error analysis in both cases is the same. We will present the analysis of the routine `slarfg2()`, where operations in exact arithmetic are performed as follows:

$$\begin{aligned}
\hat{\beta}_{j,k} &= -\mathrm{sign}(\tilde{z}_{j,k}(1))\|\tilde{z}_{j,k}\|_2 \\
\hat{v}_{j,k}(1) &= 1 \\
\hat{v}_{j,k}(2\colon n-\ell) &= \tilde{z}_{j,k}(2\colon n-\ell)/(\tilde{z}_{j,k}(1) - \hat{\beta}_{j,k}) \\
\hat{\tau}_{j,k} &= [\hat{\beta}_{j,k} - \tilde{z}_{j,k}(1)]/\hat{\beta}_{j,k} \\
\tilde{A}_{j,0}(:,\ell+1\colon n)\hat{v}_{j,k} &= [\tilde{A}_{j,0}(:,\ell+1\colon n)\tilde{z}_{j,k} - \hat{\beta}_{j,k}\tilde{A}_{j,0}(:,\ell+1)]/[\tilde{z}_{j,k}(1) - \hat{\beta}_{j,k}] \\
&\quad \text{computed as} \\
\hat{x}^{(1)}_{j,k} &= \tilde{A}_{j,0}(:,\ell+1\colon n)\tilde{z}_{j,k} \\
\hat{x}^{(3)}_{j,k} &= [\hat{x}^{(1)}_{j,k} - \hat{\beta}_{j,k}\tilde{A}_{j,0}(:,\ell+1)]/[\tilde{z}_{j,k}(1) - \hat{\beta}_{j,k}]
\end{aligned}$$

   and it is obvious that

$$|\hat{\beta}_{j,k}| = \|\tilde{z}_{j,k}\|_2, \qquad \|\hat{v}_{j,k}\|_2 = \sqrt{1 + \frac{\|\tilde{z}_{j,k}(2\colon n-\ell)\|_2^2}{(|\tilde{z}_{j,k}(1)| + \|\tilde{z}_{j,k}\|_2)^2}} \leq \sqrt{2},$$

$$1 \leq |\hat{\tau}_{j,k}| = \frac{\|\tilde{z}_{j,k}\|_2 + |\tilde{z}_{j,k}(1)|}{\|\tilde{z}_{j,k}\|_2} \leq 2.$$

Then, a Householder reflector has the form $\hat{V}_{j,k} = I - \hat{\tau}_{j,k}\hat{v}_{j,k}\hat{v}_{j,k}^T$, and the scalar $\hat{\tau}_{j,k}$ will be assigned to $\hat{x}_{j,k}^{(4)}$ later. In floating point arithmetic we have the following situation, which is straightforward.

a)

$$\tilde{\beta}_{j,k} = \hat{\beta}_{j,k}(1+\delta_1), \qquad |\delta_1| \leq O(n-\ell)\varepsilon$$

b)

$$\begin{aligned}
\tilde{v}_{j,k}(2\colon n-\ell) &= \text{fl}\left(\frac{\tilde{z}_{j,k}(2\colon n-\ell)}{\text{fl}(\tilde{z}_{j,k}(1) - \tilde{\beta}_{j,k})}\right) = \\
&= (I+\Delta_1)\frac{\tilde{z}_{j,k}(2\colon n-\ell)}{(1+\delta_2)[\tilde{z}_{j,k}(1) - \hat{\beta}_{j,k}(1+\delta_1)]} = \\
&= \frac{1}{(1+\delta_2)\left(1 - \frac{\hat{\beta}_{j,k}\delta_1}{\tilde{z}_{j,k}(1)-\hat{\beta}_{j,k}}\right)}(I+\Delta_1)\frac{\tilde{z}_{j,k}(2\colon n-\ell)}{\tilde{z}_{j,k}(1) - \hat{\beta}_{j,k}} = \\
&= (I+\Delta_2)\hat{v}_{j,k}(2\colon n-\ell) = \hat{v}_{j,k}(2\colon n-\ell) + \delta\hat{v}_{j,k}(2\colon n-\ell)
\end{aligned} \tag{3.54}$$

where

$$|\Delta_1| \leq \varepsilon I, \qquad |\delta_2| \leq \varepsilon,$$

$$\left|\frac{\hat{\beta}_{j,k}\delta_1}{\tilde{z}_{j,k}(1) - \hat{\beta}_{j,k}}\right| \leq \frac{\|\tilde{z}_{j,k}\|_2}{|\tilde{z}_{j,k}(1)| + \|\tilde{z}_{j,k}\|_2} O(n-\ell)\varepsilon \leq O(n-\ell)\varepsilon,$$

which implies that

$$|\Delta_2| \leq O(n-\ell)\varepsilon I, \qquad \|\delta\hat{v}_{j,k}\|_2 \leq O(n-\ell)\varepsilon\|\hat{v}_{j,k}\|_2 \leq O(n-\ell)\varepsilon. \tag{3.55}$$

where $\delta\hat{v}_{j,k}(1) = 0$.

c) Finally

$$\begin{aligned}
\tilde{\tau}_{j,k} &= \text{fl}\left(\frac{\text{fl}(\tilde{\beta}_{j,k} - \tilde{z}_{j,k}(1))}{\tilde{\beta}_{j,k}}\right) = (1+\delta_3)\frac{(1+\delta_4)[\hat{\beta}_{j,k}(1+\delta_1) - \tilde{z}_{j,k}(1)]}{\hat{\beta}_{j,k}(1+\delta_1)} = \\
&= \frac{(1+\delta_3)(1+\delta_4)}{1+\delta_1}\left(1 + \delta_1\frac{\hat{\beta}_{j,k}}{\hat{\beta}_{j,k} - \tilde{z}_{j,k}(1)}\right)\frac{\hat{\beta}_{j,k} - \tilde{z}_{j,k}(1)}{\hat{\beta}_{j,k}} = \\
&= (1+\delta_5)\hat{\tau}_{j,k}
\end{aligned} \tag{3.56}$$

where

$$|\delta_3| \leq \varepsilon, \quad |\delta_4| \leq \varepsilon, \quad \Longrightarrow \quad |\delta_5| \leq O(n-\ell)\varepsilon. \tag{3.57}$$

d) Further we have

$$\begin{aligned}\tilde{x}_{j,k}^{(1)} &= \mathrm{fl}(\tilde{A}_{j,0}(:,\ell+1\colon n)\tilde{z}_{j,k}) = \tilde{A}_{j,0}(:,\ell+1\colon n)\tilde{z}_{j,k} + \delta\hat{x}_{j,k}^{(1)},\\ \|\delta\hat{x}_{j,k}^{(1)}\|_2 &\leq O(n-\ell)\varepsilon\|\tilde{A}_{j,0}\|_F\|\tilde{z}_{j,k}\|_2,\end{aligned}$$

e)

$$\begin{aligned}\tilde{x}_{j,k}^{(2)} &= \mathrm{fl}(\tilde{x}_{j,k}^{(1)} - \mathrm{fl}(\tilde{\beta}_{j,k}\tilde{A}_{j,0}(:,\ell+1)) =\\ &= (I+\Delta_3)[\tilde{A}_{j,0}(:,\ell+1\colon n)\tilde{z}_{j,k} + \delta\hat{x}_{j,k}^{(1)} -\\ &\quad -(I+\Delta_4)\hat{\beta}_{j,k}(1+\delta_1)\tilde{A}_{j,0}(:,\ell+1)] =\\ &= \tilde{A}_{j,0}(:,\ell+1\colon n)\tilde{z}_{j,k} - \hat{\beta}_{j,k}\tilde{A}_{j,0}(:,\ell+1) + \delta\hat{x}_{j,k}^{(2)}\end{aligned}$$

where

$$|\Delta_3| \leq \varepsilon I, \quad |\Delta_4| \leq \varepsilon I, \quad \Longrightarrow \quad \|\delta\hat{x}_{j,k}^{(2)}\|_2 \leq O(n-\ell)\varepsilon\|\tilde{A}_{j,0}\|_F\|\tilde{z}_{j,k}\|_2.$$

f)

$$\begin{aligned}\tilde{x}_{j,k}^{(3)} &= \mathrm{fl}\left(\frac{\tilde{x}_{j,k}^{(2)}}{\mathrm{fl}(\tilde{z}_{j,k}(1) - \tilde{\beta}_{j,k})}\right) =\\ &= (I+\Delta_5)\frac{\tilde{A}_{j,0}(:,\ell+1\colon n)\tilde{z}_{j,k} - \hat{\beta}_{j,k}\tilde{A}_{j,0}(:,\ell+1) + \delta\hat{x}_{j,k}^{(2)}}{(1+\delta_6)[\tilde{z}_{j,k}(1) - \hat{\beta}_{j,k}(1+\delta_1)]} =\\ &= \frac{\tilde{A}_{j,0}(:,\ell+1\colon n)\tilde{z}_{j,k} - \hat{\beta}_{j,k}\tilde{A}_{j,0}(:,\ell+1)}{\tilde{z}_{j,k}(1) - \hat{\beta}_{j,k}} + \delta\hat{x}_{j,k}^{(3)} =\\ &= \tilde{A}_{j,0}(:,\ell+1\colon n)\hat{v}_{j,k} + \delta\hat{x}_{j,k}^{(3)}, \qquad (3.58)\end{aligned}$$

where

$$|\Delta_5| \leq \varepsilon I, \qquad |\delta_6| \leq \varepsilon,$$

$$\begin{aligned}\|\delta\hat{x}_{j,k}^{(3)}\|_2 &\leq O(n-\ell)\varepsilon\left\|\frac{\tilde{A}_{j,0}(:,\ell+1\colon n)\tilde{z}_{j,k} - \hat{\beta}_{j,k}\tilde{A}_{j,0}(:,\ell+1)}{\tilde{z}_{j,k}(1) - \hat{\beta}_{j,k}}\right\|_2 +\\ &\quad +O(n-\ell)\varepsilon\frac{\|\tilde{A}_{j,0}\|_F\|\tilde{z}_{j,k}\|_2}{|\tilde{z}_{j,k}(1)| + \|\tilde{z}_{j,k}\|_2} \leq\\ &\leq O(n-\ell)\varepsilon\|\tilde{A}_{j,0}(:,\ell+1\colon n)\hat{v}_{j,k}\|_2 + O(n-\ell)\varepsilon\|\tilde{A}_{j,0}\|_F \leq\\ &\leq O(n-\ell)\varepsilon\|\tilde{A}_{j,0}\|_F. \qquad (3.59)\end{aligned}$$

2. Now we turn to the computed columns of the matrix $\tilde{X}_j$. Let us remind that we use the aggregated form of Householder transformations, where in exact arithmetic we have

$$\begin{aligned}\mathbf{V}_{j,i}\mathbf{V}_{j,i-1}\cdots\mathbf{V}_{j,1} &= (I - \mathbf{v}_{j,i}\mathbf{v}_{j,i}^T)(I - \mathbf{v}_{j,i-1}\mathbf{v}_{j,i-1}^T)\cdots(I - \mathbf{v}_{j,1}\mathbf{v}_{j,1}^T) =\\ &= I - W_j(:,1\colon i)Y_j(:,1\colon i)^T,\end{aligned}$$

$$\mathbf{V}_{j,1},\ldots,\mathbf{V}_{j,i}\in\mathbb{R}^{n\times n},\qquad W_j(:,1\colon i),Y_j(:,1\colon i)\in\mathbb{R}^{n\times i},$$

and W_j and Y_j are computed through a recurrence

$$W_j(:,1)=\mathbf{v}_{j,1},\qquad Y_j(:,1)=\mathbf{v}_{j,1},$$

$$\begin{aligned} W_j(:,1\colon i) &= [W_j(:,1\colon i-1),\ \mathbf{v}_{j,i}],\\ Y_j(:,1\colon i) &= [Y_j(:,1\colon i-1),\ \mathbf{V}_{j,1}\cdots\mathbf{V}_{j,i-1}\mathbf{v}_{j,i}]. \end{aligned}$$

In our case, it holds

$$\hat{X}_j=\tilde{A}_{j,0}\hat{Y}_j,\qquad \hat{X}_j=\hat{X}_j(:,1\colon b),\ \hat{Y}_j=\hat{Y}_j(:,1\colon b),$$

and for $\ell=(j-1)b+k$

$$\begin{aligned} &\hat{W}_j(1\colon \ell,k)=0_{\ell\times 1},\qquad \hat{W}_j(\ell+1\colon n,k)=\hat{v}_{j,k},\\ &\hat{X}_j(:,k)=\tilde{A}_{j,0}(:,\ell+1\colon n)\hat{v}_{j,k}-\tilde{A}_{j,0}\hat{Y}_j(:,1\colon k-1)\hat{W}_j(\ell+1\colon n,1\colon k-1)^T\hat{v}_{j,k}. \end{aligned}$$

Because of the choice of $v_{j,k}$ and $\tau_{j,k}$ in `slarfg2()`, in floating point arithmetic $\tilde{X}_j(:,k)$ is computed through the following steps:

$$\begin{aligned} \tilde{x}_{j,k}^{(4)} &= \mathrm{fl}(\tilde{x}_{j,k}^{(3)}-\mathrm{fl}(\tilde{X}_j(:,1\colon k-1)\mathrm{fl}(\tilde{W}_j(\ell+1\colon n,1\colon k-1)^T\tilde{v}_{j,k})))\\ \tilde{X}_j(:,k) &= \mathrm{fl}(\tilde{\tau}_{j,k}\tilde{x}_{j,k}^{(4)}), \end{aligned}\tag{3.60}$$

and only for $k=1$, holds that $\tilde{X}_j(:,1)=\mathrm{fl}(\tilde{\tau}_{j,1}\tilde{x}_{j,1}^{(3)})$.

First of all, we can find the error estimation for $\tilde{W}_j(:,1\colon k)$ immediately from (3.54) and (3.55).

$$\|\hat{W}_j(:,1\colon k)\|_F\le\sqrt{2}\sqrt{k},\qquad \delta\hat{W}_j(:,1\colon k)=[\delta\hat{\mathbf{v}}_{j,1}\ \ldots\ \delta\hat{\mathbf{v}}_{j,k}],$$

$$\tilde{W}_j(:,1\colon k)=\hat{W}_j(:,1\colon k)+\delta\hat{W}_j(:,1\colon k),\qquad \|\delta\hat{W}_j(:,1\colon k)\|_F\le O(\sqrt{kn})\varepsilon\tag{3.61}$$

for $k=1,\ldots,b$, where $\delta\hat{\mathbf{v}}_{j,k}=[0\ \delta\hat{v}_{j,k}]^T$.

For the exact value $\|\hat{X}_j(:,1\colon k)\|_F$ we know that

$$\|\hat{X}_j(:,1\colon k)\|_F=\|\tilde{A}_{j,0}\hat{Y}_j(:,1\colon k)\|_F\le 2\sqrt{2}\sqrt{k}\|\tilde{A}_{j,0}\|_F,$$

where

$$\begin{aligned} \hat{Y}_j(:,k)&=\hat{\tau}_{j,k}\mathbf{V}_{j,1}\cdots\mathbf{V}_{j,k-1}\hat{v}_{j,k},\\ \hat{X}_j(:,1\colon k)&=\tilde{A}_{j,0}\hat{Y}_j(:,1\colon k). \end{aligned}$$

3. The error analysis for $\tilde{X}_j(:,1\colon k)$ will be conducted through mathematical induction on k.

a) For $k = 1$, from (3.56), (3.57), (3.58) and (3.59) it follows

$$\begin{aligned}\tilde{X}_j(:,1) &= \mathrm{fl}(\tilde{\tau}_{j,1}\tilde{x}^{(3)}_{j,1}) = \\ &= (I+\Delta_6)(1+\delta_5)\hat{\tau}_{j,1}(\tilde{A}_{j,0}(:,\ell+1\colon n)\hat{v}_{j,1} + \delta\hat{x}^{(3)}_{j,1}) \\ &= \hat{\tau}_{j,1}\tilde{A}_{j,0}(:,\ell+1\colon n)\hat{v}_{j,1} + \delta\hat{X}_j(:,1) = \tilde{A}_{j,0}\hat{Y}_j(:,1) + \delta\hat{X}_j(:,1),\end{aligned}$$

where

$$|\Delta_6| \le \varepsilon I \qquad \Longrightarrow \qquad \|\delta\hat{X}_j(:,1)\|_2 \le O(n-\ell)\varepsilon\|\tilde{A}_{j,0}\|_F.$$

b) Now, let us assume that for $k-1$, $k>1$ and $\ell = (j-1)b+k$, we obtain

$$\begin{aligned}\tilde{x}^{(4)}_{j,k-1} &= \tilde{A}_{j,0}(:,\ell\colon n)\hat{v}_{j,k-1} - \\ &\quad -\tilde{A}_{j,0}\hat{Y}_j(:,1\colon k-2)\hat{W}_j(\ell\colon n,1\colon k-2)^T\hat{v}_{j,k-1} + \delta\hat{x}^{(4)}_{j,k-1},\end{aligned} \tag{3.62}$$

$$\|\delta\hat{x}^{(4)}_{j,k-1}\|_2 \le O((k-1)n)\varepsilon\|\tilde{A}_{j,0}\|_F. \tag{3.63}$$

We can get the same result if we took any $h \in \mathbb{R}^{n-\ell+1}$, with $\|h\|_2 \le \sqrt{2}$, and start the computation of $\tilde{x}^{(4)}_{j,k-1}$ with it. The next step will show that

$$\begin{aligned}\tilde{x}^{(4)}_{j,k} &= \tilde{A}_{j,0}(:,\ell+1\colon n)\hat{v}_{j,k} - \\ &\quad -\tilde{A}_{j,0}\hat{Y}_j(:,1\colon k-1)\hat{W}_j(\ell+1\colon n,1\colon k-1)^T\hat{v}_{j,k} + \delta\hat{x}^{(4)}_{j,k},\end{aligned}$$

where

$$\|\delta\hat{x}^{(4)}_{j,k}\|_2 \le O(kn)\varepsilon\|\tilde{A}_{j,0}\|_F.$$

4. We have

$$\begin{aligned}\tilde{x}^{(4)}_{j,k} &= \mathrm{fl}(\tilde{x}^{(3)}_{j,k} - \mathrm{fl}(\tilde{X}_j(:,1\colon k-1)\mathrm{fl}(\tilde{W}_j(\ell+1\colon n,1\colon k-1)^T\tilde{v}_{j,k}))) = \\ &= (I+\Delta_7)[\tilde{x}^{(3)}_{j,k} - \mathrm{fl}(\tilde{X}_j(:,1\colon k-1)\mathrm{fl}(\tilde{W}_j(\ell+1\colon n,1\colon k-1)^T\tilde{v}_{j,k}))] = \\ &= (I+\Delta_7)[\tilde{x}^{(3)}_{j,k} - \mathrm{fl}(\tilde{X}_j(:,1\colon k-2)\mathrm{fl}(\tilde{W}_j(\ell+1\colon n,1\colon k-2)^T\tilde{v}_{j,k}) + \\ &\quad +\mathrm{fl}(\tilde{W}_j(\ell+1\colon n,k-1)^T\tilde{v}_{j,k})\tilde{X}_j(:,k-1))] = \\ &= (I+\Delta_7)\{\tilde{x}^{(3)}_{j,k} - (I+\Delta_8)[\mathrm{fl}(\tilde{X}_j(:,1\colon k-2)\mathrm{fl}(\tilde{W}_j(\ell+1\colon n,1\colon k-2)^T\tilde{v}_{j,k})) + \\ &\quad +\mathrm{fl}(\mathrm{fl}(\tilde{v}_{j,k-1}(2\colon n-\ell+1)^T\tilde{v}_{j,k})\tilde{X}_j(:,k-1))]\} = \\ &= (I+\Delta_7)\{\tilde{x}^{(3)}_{j,k} - \mathrm{fl}(\tilde{X}_j(:,1\colon k-2)\mathrm{fl}(\tilde{W}_j(\ell+1\colon n,1\colon k-2)^T\tilde{v}_{j,k})) + \\ &\quad +\mathrm{fl}(\mathrm{fl}(\tilde{v}_{j,k-1}(2\colon n-\ell+1)^T\tilde{v}_{j,k})\tilde{X}_j(:,k-1)) + \\ &\quad +\Delta_8[\mathrm{fl}(\tilde{X}_j(:,1\colon k-2)\mathrm{fl}(\tilde{W}_j(\ell+1\colon n,1\colon k-2)^T\tilde{v}_{j,k})) + \\ &\quad +\mathrm{fl}(\mathrm{fl}(\tilde{v}_{j,k-1}(2\colon n-\ell+1)^T\tilde{v}_{j,k})\tilde{X}_j(:,k-1))]\}\end{aligned} \tag{3.64}$$

where

a)

$$|\Delta_7| \leq \varepsilon I, \qquad |\Delta_8| \leq \varepsilon I, \tag{3.65}$$

and

$$\begin{aligned}
&\tilde{x}_{j,k}^{(3)} - \mathrm{fl}(\tilde{X}_j(:,1\colon k-2)\mathrm{fl}(\tilde{W}_j(\ell+1\colon n,1\colon k-2)^T\tilde{v}_{j,k})) = \\
&\quad = (I+\Delta_9)\mathrm{fl}(\tilde{x}_{j,k}^{(3)} - \mathrm{fl}(\tilde{X}_j(:,1\colon k-2)\mathrm{fl}(\tilde{W}_j(\ell+1\colon n,1\colon k-2)^T\tilde{v}_{j,k}))) = \\
&\quad = (I+\Delta_9)\tilde{x}_{j,k-1,k}^{(4)}
\end{aligned}$$

with

$$|\Delta_9| \leq \varepsilon I.$$

The notation $\tilde{x}_{j,k-1,k}^{(4)}$ describes the vector which is computed in the same way as $\tilde{x}_{j,k-1}^{(4)}$, but in this case its computation started with $\begin{bmatrix} 0 \\ \tilde{v}_{j,k} \end{bmatrix} \in \mathbb{R}^{n-\ell+1}$ instead of $\tilde{v}_{j,k-1}$.

b) Now we can apply the induction assumption here and state

$$\begin{aligned}
\tilde{x}_{j,k-1,k}^{(4)} &= \tilde{A}_{j,0}(:,\ell\colon n)\begin{bmatrix} 0 \\ \hat{v}_{j,k} \end{bmatrix} - \\
&\quad -\tilde{A}_{j,0}\hat{Y}_j(:,1\colon k-2)\hat{W}_j(\ell\colon n,1\colon k-2)^T\begin{bmatrix} 0 \\ \hat{v}_{j,k} \end{bmatrix} + \delta\hat{x}_{j,k-1,k}^{(4)} = \\
&= \tilde{A}_{j-1}(:,\ell+1\colon n)\hat{v}_{j,k} - \\
&\quad -\tilde{A}_{j-1}\hat{Y}_j(:,1\colon k-2)\hat{W}_j(\ell+1\colon n,1\colon k-2)^T\hat{v}_{j,k} + \delta\hat{x}_{j,k-1,k}^{(4)},
\end{aligned}$$

where

$$\|\delta\hat{x}_{j,k-1,k}^{(4)}\|_2 \leq O((k-1)n)\varepsilon\|\tilde{A}_{j,0}\|_F.$$

So

$$\begin{aligned}
&\tilde{x}_{j,k}^{(3)} - \mathrm{fl}(\tilde{X}_j(:,1\colon k-2)\mathrm{fl}(\tilde{W}_j(\ell+1\colon n,1\colon k-2)^T\tilde{v}_{j,k})) = \\
&\quad = \tilde{A}_{j,0}(:,\ell+1\colon n)\hat{v}_{j,k} - \tilde{A}_{j,0}\hat{Y}_j(:,1\colon k-2)\hat{W}_j(\ell+1\colon n,1\colon k-2)^T\hat{v}_{j,k} + \\
&\quad +\delta_1\hat{x}_{j,k}^{(4)}
\end{aligned} \tag{3.66}$$

with

$$\|\delta_1\hat{x}_{j,k}^{(4)}\|_2 \leq O((k-1)n)\varepsilon\|\tilde{A}_{j,0}\|_F. \tag{3.67}$$

c) Next, we have

$$\begin{aligned}
&\mathrm{fl}(\mathrm{fl}(\tilde{v}_{j,k-1}(2\colon n-\ell+1)^T\tilde{v}_{j,k})\tilde{X}_j(:,k-1)) = \\
&\quad = (I+\Delta_{10})(\tilde{v}_{j,k-1}(2\colon n-\ell+1)^T\tilde{v}_{j,k}+\delta_7)(I+\Delta_{11})\tilde{\tau}_{j,k-1}\tilde{x}_{j,k-1}^{(4)} = \\
&\quad = \tilde{v}_{j,k-1}(2\colon n-\ell+1)^T\tilde{v}_{j,k}\tilde{\tau}_{j,k-1}\tilde{x}_{j,k-1}^{(4)} + \delta_2\hat{x}_{j,k}^{(4)}
\end{aligned}$$

with

$$|\Delta_{10}| \leq \varepsilon I, \quad |\Delta_{11}| \leq \varepsilon I, \quad |\delta_7| \leq O(n-\ell)\varepsilon,$$

$$\|\tilde{x}^{(4)}_{j,k-1}\|_2 \leq \sqrt{2}\|\tilde{A}_{j,0}\|_F + O(\varepsilon),$$

which implies that

$$\|\delta_2\hat{x}^{(4)}_{j,k}\|_2 \leq O(n-\ell)\varepsilon\|\tilde{A}_{j,0}\|_F.$$

Further, from (3.54), (3.55), (3.60), (3.56), (3.57), (3.62) and (3.63) it follows that

$$\begin{aligned}
&\mathrm{fl}(\mathrm{fl}(\tilde{v}_{j,k-1}(2\colon n-\ell+1)^T\tilde{v}_{j,k})\tilde{X}_j(\colon,k-1)) = \\
&= (\hat{v}_{j,k-1}(2\colon n-\ell+1)^T + \delta\hat{v}_{j,k-1}(2\colon n-\ell+1)^T)(\hat{v}_{j,k}+\delta\hat{v}_{j,k})(1+\delta_5)\hat{\tau}_{j,k-1}\cdot \\
&\quad\cdot[\tilde{A}_{j,0}(\colon,\ell\colon n)\hat{v}_{j,k-1} - \tilde{A}_{j,0}\hat{Y}_j(\colon,1\colon k-2)\hat{W}_j(\ell\colon n,1\colon k-2)^T\hat{v}_{j,k-1} + \\
&\quad+\delta\hat{x}^{(4)}_{j,k-1}] + \delta_2\hat{x}^{(4)}_{j,k} = \\
&= (\hat{v}_{j,k-1}(2\colon n-\ell+1)^T + \delta\hat{v}_{j,k-1}(2\colon n-\ell+1)^T)(\hat{v}_{j,k}+\delta\hat{v}_{j,k})(1+\delta_5)\cdot \\
&\quad\cdot[\tilde{A}_{j,0}\hat{Y}_j(\colon,k-1) + \hat{\tau}_{j,k-1}\delta\hat{x}^{(4)}_{j,k-1}] + \delta_2\hat{x}^{(4)}_{j,k} = \\
&= \hat{v}_{j,k-1}(2\colon n-\ell+1)^T\hat{v}_{j,k}\tilde{A}_{j,0}\hat{Y}_j(\colon,k-1) + \delta_3\hat{x}^{(4)}_{j,k}, \qquad (3.68)
\end{aligned}$$

where

$$\|\delta_3\hat{x}^{(4)}_{j,k}\|_2 \leq [O(n)+O((k-1)n)]\varepsilon\|\tilde{A}_{j,0}\|_F. \qquad (3.69)$$

d) The last thing we have to check is:

$$\begin{aligned}
&\Delta_8[\mathrm{fl}(\tilde{X}_j(\colon,1\colon k-2)\mathrm{fl}(\tilde{W}_j(\ell+1\colon n,1\colon k-2)^T\tilde{v}_{j,k}))+ \\
&+\mathrm{fl}(\mathrm{fl}(\tilde{v}_{j,k-1}(2\colon n-\ell+1)^T\tilde{v}_{j,k})\tilde{X}_j(\colon,k-1))] = \\
&\quad= \Delta_8[\hat{X}_j(\colon,1\colon k-1)\hat{W}_j(\ell+1\colon n,1\colon k-1)^T\hat{v}_{j,k}+O(\varepsilon)] = \\
&\quad= \delta_4\hat{x}^{(4)}_{j,k}, \qquad (3.70)
\end{aligned}$$

where, from the induction assumption, (3.56), (3.54) and (3.61) it follows that

$$\|\delta_4\hat{x}^{(4)}_{j,k}\|_2 \leq O(k-1)\varepsilon\|\tilde{A}_{j,0}\|_F. \qquad (3.71)$$

Putting everything together, from (3.64), (3.65), (3.66), (3.67), (3.68), (3.69), (3.70) and (3.71) we obtain

$$\begin{aligned}
\tilde{x}^{(4)}_{j,k} &= (I+\Delta_7)[\tilde{A}_{j,0}(\colon,\ell+1\colon n)\hat{v}_{j,k} - \\
&\quad-\tilde{A}_{j,0}\hat{Y}_j(\colon,1\colon k-1)\hat{W}_j(\ell+1\colon n,1\colon k-1)^T\hat{v}_{j,k} + \\
&\quad+\delta_1\hat{x}^{(4)}_{j,k}+\delta_3\hat{x}^{(4)}_{j,k}+\delta_4\hat{x}^{(4)}_{j,k}] = \\
&= \tilde{A}_{j,0}(\colon,\ell+1\colon n)\hat{v}_{j,k} - \tilde{A}_{j,0}\hat{Y}_j(\colon,1\colon k-1)\hat{W}_j(\ell+1\colon n,1\colon k-1)^T\hat{v}_{j,k} + \\
&\quad+\delta\hat{x}^{(4)}_{j,k}, \qquad (3.72)
\end{aligned}$$

with

$$\|\delta\hat{x}^{(4)}_{j,k}\|_2 \leq [O(n)+O((k-1)n)]\varepsilon\|\tilde{A}_{j,0}\|_F \leq O(kn)\varepsilon\|\tilde{A}_{j,0}\|_F, \qquad (3.73)$$

and from (3.56), (3.57), (3.72) and (3.73) we obtain

$$\begin{aligned}
\tilde{X}_j(:,k) &= \mathrm{fl}(\tilde{\tau}_{j,k}\tilde{x}^{(4)}_{j,k}) = \\
&= (I+\Delta_{12})(1+\delta_5)\hat{\tau}_{j,k}[\tilde{A}_{j,0}(:,\ell+1\colon n)\hat{v}_{j,k} - \\
&\quad -\tilde{A}_{j,0}\hat{Y}_j(:,1\colon k-1)\hat{W}_j(\ell+1\colon n,1\colon k-1)^T\hat{v}_{j,k} + \delta\hat{x}^{(4)}_{j,k}] \\
&= \hat{\tau}_{j,k}[\tilde{A}_{j,0}(:,\ell+1\colon n)\hat{v}_{j,k} - \tilde{A}_{j,0}\hat{Y}_j(:,1\colon k-1)\cdot \\
&\quad \cdot\hat{W}_j(\ell+1\colon n,1\colon k-1)^T\hat{v}_{j,k}] + \delta\hat{X}_j(:,k) = \\
&= \tilde{A}_{j,0}\hat{Y}_j(:,k) + \delta\hat{X}_j(:,k),
\end{aligned}$$

with

$$|\Delta_{12}| \le \varepsilon I, \quad \Longrightarrow \quad \|\delta\hat{X}_j(:,k)\|_2 \le O(kn)\varepsilon\|\tilde{A}_{j,0}\|_F.$$

5. Finally, we get the result:

$$\tilde{X}_j(:,1\colon k) = \hat{X}_j(:,1\colon k) + \delta\hat{X}_j(:,1\colon k) = \tilde{A}_{j,0}\hat{Y}_j(:,1\colon k) + \delta\hat{X}_j(:,1\colon k),$$

where

$$\|\delta\hat{X}_j(:,1\colon k)\|_F = \sqrt{\sum_{i=1}^{k}\|\delta\hat{X}_j(:,i)\|_2^2} \le O(k^{\frac{3}{2}}n)\varepsilon\|\tilde{A}_{j,0}\|_F.$$

□

The next Lemma is stated on account of the completeness of the proof. It is a combination of **Step 2** in the proof of Theorem 3.2.6 and Remark 3.2.7.

Lemma 3.6.2. *Computed elements of the matrix $\tilde{B}$ from Algorithm 3.5.1 satisfy the following relations:*

$$\begin{aligned}
\begin{bmatrix} \tilde{\phi}_{k+1}e_k + \tilde{\psi}_{k+1}e_{k+1} \\ 0 \end{bmatrix} &= \hat{P}_{k+1}\hat{P}_k\left(\begin{bmatrix} 0 \\ f_{k+1} \end{bmatrix} + \begin{bmatrix} \Delta f_{k+1} \\ \delta f_{k+1} \end{bmatrix}\right), \\
\left\|\begin{bmatrix} \Delta f_{k+1} \\ \delta f_{k+1} \end{bmatrix}\right\|_2 &\le O(bm)\varepsilon\|F\|_F,
\end{aligned}$$

where $\hat{P}_k$, $k = 1,\ldots,n$ are exact $(m+n)\times(m+n)$ Householder reflectors defined in [6].

Proof. Consider the computation of the k–th column of $\tilde{B}$. An application of the results on floating point computation from [47, Chapters 2 and 3] reveals that

$$\begin{aligned}
\tilde{\psi}_1 &= \mathrm{fl}(\|f_1\|_2) = \|f_1\|_2 - \delta\tilde{\psi}_1, \\
|\delta\tilde{\psi}_1| &\le O(m)\varepsilon\|f_1\|_2, \\
\tilde{u}_1 &= \mathrm{fl}\left(\frac{f_1}{\tilde{\psi}_1}\right) = \hat{u}_1 + \delta\hat{u}_1, \\
\hat{u}_1 &= \frac{f_1}{\|f_1\|_2}, \\
\|\delta\hat{u}_1\|_2 &\le O(m)\varepsilon.
\end{aligned}$$

Furthermore, since the current column of A is updated with at most b Householder reflectors at once, the following bounds look like the bounds in Remark 3.2.7, multiplied by b. For $k = 1, 2, \ldots$ we have ([47, Chapter 19]):

$$\begin{aligned} \tilde{\phi}_{k+1} &= e_1^T \mathrm{fl}(\tilde{V}_k \tilde{z}_k) = \hat{u}_k^T f_{k+1} + \delta\hat{\phi}_{k+1}, \quad |\delta\hat{\phi}_{k+1}| \leq O(bm)\varepsilon\|F\|_F, \\ \tilde{s}_{k+1} &\equiv \mathrm{fl}(f_{k+1} - \tilde{\phi}_{k+1}\tilde{u}_k) = f_{k+1} - \hat{\phi}_{k+1}\hat{u}_k + \delta\hat{s}_{k+1}, \end{aligned}$$

where

$$\begin{aligned} \|\delta\hat{s}_{k+1}\|_2 &\leq O(bm)\varepsilon\|F\|_F, \\ \hat{\phi}_{k+1} &= \hat{u}_k^T f_{k+1}, \qquad |\hat{\phi}_{k+1}| \leq \|F\|_F. \end{aligned}$$

According to the ideas of Björck and Paige [6], we can write this computation as

$$\begin{aligned} \begin{bmatrix} \tilde{\phi}_{k+1}e_k \\ \tilde{s}_{k+1} \end{bmatrix} &= \begin{bmatrix} \hat{\phi}_{k+1}e_k \\ f_{k+1} - \hat{\phi}_{k+1}\hat{u}_k \end{bmatrix} + \begin{bmatrix} \delta\hat{\phi}_{k+1}e_k \\ \delta\hat{s}_{k+1} \end{bmatrix} = \\ &= \hat{P}_k \left\{ \begin{bmatrix} 0 \\ f_{k+1} \end{bmatrix} + \hat{P}_k \begin{bmatrix} \delta\hat{\phi}_{k+1}e_k \\ \delta\hat{s}_{k+1} \end{bmatrix} \right\}, \\ \hat{P}_k &= I_{m+n} - \begin{bmatrix} -e_k \\ \hat{u}_k \end{bmatrix} \begin{bmatrix} -e_k^T & \hat{u}_k^T \end{bmatrix}, \end{aligned}$$

where e_k denotes the kth column of the identity matrix I_n. Note that $\hat{P}_k^2 = I_{m+n}$. Further, the values $\tilde{\psi}_{k+1} = \mathrm{fl}(\|\tilde{s}_{k+1}\|_2)$, $\tilde{u}_{k+1} = \mathrm{fl}(\tilde{s}_{k+1}/\tilde{\psi}_{k+1})$ satisfy

$$\begin{aligned} \tilde{\psi}_{k+1} &= \|\tilde{s}_{k+1}\|_2 - \delta\tilde{\psi}_{k+1}, \quad |\delta\tilde{\psi}_{k+1}| \leq O(m)\varepsilon\|F\|_F, \\ \tilde{u}_{k+1} &= \hat{u}_{k+1} + \delta\hat{u}_{k+1}, \quad \hat{u}_{k+1} = \tilde{s}_{k+1}/\|\tilde{s}_{k+1}\|_2, \quad \|\delta\hat{u}_{k+1}\|_2 \leq O(m)\varepsilon. \end{aligned}$$

Thus, the computation of the $(k+1)$-th column of $\tilde{B}$ can be written as

$$\begin{aligned} &\begin{bmatrix} \tilde{\phi}_{k+1}e_k + \tilde{\psi}_{k+1}e_{k+1} \\ 0 \end{bmatrix} = \begin{bmatrix} \tilde{\phi}_{k+1}e_k + \|\tilde{s}_{k+1}\|_2 e_{k+1} \\ 0 \end{bmatrix} - \begin{bmatrix} \delta\tilde{\psi}_{k+1}e_{k+1} \\ 0 \end{bmatrix} = \\ &= \hat{P}_{k+1} \left\{ \begin{bmatrix} \tilde{\phi}_{k+1}e_k \\ \tilde{s}_{k+1} \end{bmatrix} - \hat{P}_{k+1} \begin{bmatrix} \delta\tilde{\psi}_{k+1}e_{k+1} \\ 0 \end{bmatrix} \right\} = \\ &= \hat{P}_{k+1} \left\{ \hat{P}_k \begin{bmatrix} 0 \\ f_{k+1} \end{bmatrix} + \begin{bmatrix} \delta\hat{\phi}_{k+1}e_k \\ \delta\hat{s}_{k+1} \end{bmatrix} - \hat{P}_{k+1} \begin{bmatrix} \delta\tilde{\psi}_{k+1}e_{k+1} \\ 0 \end{bmatrix} \right\} = \\ &= \hat{P}_{k+1}\hat{P}_k \left\{ \begin{bmatrix} 0 \\ f_{k+1} \end{bmatrix} + \begin{bmatrix} \Delta f_{k+1} \\ \delta f_{k+1} \end{bmatrix} \right\}, \quad \left\| \begin{bmatrix} \Delta f_{k+1} \\ \delta f_{k+1} \end{bmatrix} \right\|_2 \leq O(bm)\varepsilon\|F\|_F. \end{aligned}$$

In case when $k = 0$ and when we compute the first column of $\tilde{B}$, we can write $\tilde{\phi}_1 = 0$, $\tilde{s}_1 = f_1$, $\hat{P}_0 = I_{m+n}$. Hence, we can conclude that

$$| \, \|\tilde{B}(:, k+1)\|_2 - \|f_{k+1}\|_2 \, | \leq O(bm)\varepsilon\|F\|_F.$$

□

Lemma 3.6.3. *The computed vector $\tilde{u}_\ell$ and the computed submatrix $\tilde{A}_{g+1,r-1}(:,\ell+2:n)$ from Algorithm 3.5.1 satisfy the following relation:*

$$\|\tilde{u}_\ell^T \tilde{A}_{g+1,r-1}(:,\ell+2:n)\|_2 \leq O(b^2n + bm + bn^2)\varepsilon\|A\|_F.$$

Proof. As mentioned before, the choice of the vector $z_{j,k}$ plays an important role in backward stability of Algorithm 3.5.1. The computed vector $\tilde{z}_{j,k} \in \mathbb{R}^{n-\ell}$, satisfies the following relations.

$$\begin{aligned}
\tilde{z}_{j,k}^{(1)} &= -\mathrm{fl}(\tilde{W}(\ell+1:n,1:k-1)\mathrm{fl}(\tilde{X}(:,1:k-1)^T\tilde{u}_\ell)) = \\
&= -\tilde{W}(\ell+1:n,1:k-1)[\tilde{X}(:,1:k-1)^T\tilde{u}_\ell + \delta_1\hat{z}_{j,k}^{(1)}] + \delta_2\hat{z}_{j,k}^{(1)} = \\
&= -\tilde{W}(\ell+1:n,1:k-1)\tilde{X}(:,1:k-1)^T\tilde{u}_\ell + \delta\hat{z}_{j,k}^{(1)},
\end{aligned}$$

where $\ell = (j-1)b+k$, and

$$\|\delta_1\hat{z}_{j,k}^{(1)}\|_2 \leq O(\sqrt{k-1}m)\varepsilon\|\tilde{A}_{j,0}\|_F, \qquad \|\delta_2\hat{z}_{j,k}^{(1)}\|_2 \leq O((k-1)^2)\|\tilde{A}_{j,0}\|_F.$$

This implies

$$\|\delta\hat{z}_{j,k}^{(1)}\|_2 \leq O((k-1)m + (k-1)^2)\varepsilon\|\tilde{A}_{j,0}\|_F.$$

Further we have

$$\begin{aligned}
\tilde{z}_{j,k} &= \mathrm{fl}(\mathrm{fl}(\tilde{A}_{j,0}(:,\ell+1:n)^T\tilde{u}_\ell) + \tilde{z}_{j,k}^{(1)}) = \\
&= (I+\Delta_{13})[\tilde{A}_{j,0}(:,\ell+1:n)^T\tilde{u}_\ell + \delta_1\hat{z}_{j,k} - \\
&\quad -\tilde{W}_j(\ell+1:n,1:k-1)\tilde{X}_j(:,1:k-1)^T\tilde{u}_\ell + \delta\hat{z}_{j,k}^{(1)}] = \\
&= [\tilde{A}_{j,0}(:,\ell+1:n) - \tilde{X}_j(:,1:k-1)\tilde{W}_j(l+1:n,1:k-1)^T]^T\tilde{u}_\ell + \delta_2\hat{z}_{j,k},
\end{aligned}$$

with

$$|\Delta_{13}| \leq \varepsilon I, \qquad \|\delta_1\hat{z}_{j,k}\|_2 \leq O(m)\varepsilon\|\tilde{A}_{j,0}\|_F,$$

so

$$\|\delta_2\hat{z}_{j,k}\|_2 \leq O(km+k^2)\varepsilon\|\tilde{A}_{j,0}\|_F,$$

and from Lemma 3.6.1 it follows that

$$\begin{aligned}
\tilde{z}_{j,k} &= [\tilde{A}_{j,0}(:,\ell+1:n) - (\hat{X}_j(:,1:k-1) + \delta\hat{X}_j(:,1:k-1))\cdot \\
&\quad \cdot(\hat{W}_j(\ell+1:n,1:k-1)^T + \delta\hat{W}_j(\ell+1:n,1:k-1)^T)]^T\tilde{u}_\ell + \delta_2\hat{z}_{j,k} = \\
&= [\tilde{A}_{j,0}(:,\ell+1:n) - \hat{X}_j(:,1:k-1)\hat{W}_j(\ell+1:n,1:k-1)^T]^T\tilde{u}_\ell + \\
&\quad +\delta\hat{z}_{j,k}, \qquad (3.74)
\end{aligned}$$

where

$$\|\delta\hat{z}_{j,k}\|_2 \leq O(k^2n+km+k^2)\varepsilon\|\tilde{A}_{j,0}\|_F \leq O(k^2n+km)\varepsilon\|A\|_F. \qquad (3.75)$$

First, for $\ell = (j-1)b+k$, $k = 1,\ldots,b$, we define the matrices $\hat{V}_{j,k}, \tilde{V}_{j,k} \in \mathbb{R}^{(n-\ell)\times(n-\ell)}$ as (see proof of Lemma 3.6.1)

$$\hat{V}_{j,k} = I - \hat{\tau}_{j,k}\hat{v}_{j,k}\hat{v}_{j,k}^T, \quad \tilde{V}_{j,k} = I - \tilde{\tau}_{j,k}\tilde{v}_{j,k}\tilde{v}_{j,k}^T,$$

and $\hat{Q}_j, \tilde{Q}_j \in \mathbb{R}^{n\times n}$ as

$$\hat{Q}_j = \begin{bmatrix} I_{jb} & 0 \\ 0 & \hat{V}_{j,b} \end{bmatrix} \cdots \begin{bmatrix} I_{(j-1)b+2} & 0 \\ 0 & \hat{V}_{j,2} \end{bmatrix} \begin{bmatrix} I_{(j-1)b+1} & 0 \\ 0 & \hat{V}_{j,1} \end{bmatrix},$$

$$\tilde{Q}_j = \begin{bmatrix} I_{jb} & 0 \\ 0 & \tilde{V}_{j,b} \end{bmatrix} \cdots \begin{bmatrix} I_{(j-1)b+2} & 0 \\ 0 & \tilde{V}_{j,2} \end{bmatrix} \begin{bmatrix} I_{(j-1)b+1} & 0 \\ 0 & \tilde{V}_{j,1} \end{bmatrix},$$

where $\hat{Q}_j = I - \hat{W}_j \hat{Y}_j^T$ and $\tilde{Q}_j = I - \tilde{W}_j \tilde{Y}_j^T$. The exact Householder reflector $\hat{V}_{j,k} = I - \hat{\tau}_{j,k}\hat{v}_{j,k}\hat{v}_{j,k}^T \in \mathbb{R}^{(n-\ell)\times(n-\ell)}$ is chosen so that

$$(I - \hat{\tau}_{j,k}\hat{v}_{j,k}\hat{v}_{j,k}^T)\tilde{z}_{j,k} = \begin{bmatrix} \|\tilde{z}_{j,k}\|_2 \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{3.76}$$

Then, for $\hat{X}_j, \tilde{X}_j \in \mathbb{R}^{m\times b}$ and $\hat{W}_j, \tilde{W}_j \in \mathbb{R}^{n\times b}$, from Lemma 3.6.1 it follows that

$$\begin{aligned} \tilde{A}_{j+1,0} &= \mathrm{fl}(\tilde{A}_{j,0}\tilde{Q}_j^T) = \mathrm{fl}(\tilde{A}_{j,0} - \mathrm{fl}(\tilde{X}_j\tilde{W}_j^T)) = \\ &= \tilde{A}_{j,0} - [(\hat{X}_j + \delta\hat{X}_j)(\hat{W}_j^T + \delta\hat{W}_j^T) + \delta_1 A_{j+1,0}] + \delta_2 A_{j+1,0} = \\ &= \tilde{A}_{j,0} - \hat{X}_j\hat{W}_j^T + \delta A_{j+1,0} = \tilde{A}_{j,0} - \tilde{A}_{j,0}\hat{Y}_j\hat{W}_j^T + \delta A_{j+1,0} = \\ &= \tilde{A}_{j,0}\hat{Q}_j^T + \delta A_{j+1,0}, \end{aligned}$$

where

$$\|\delta_1 A_{j+1,0}\|_F \le O(b^2)\varepsilon\|\tilde{A}_{j,0}\|_F, \qquad \|\delta_2 A_{j+1,0}\|_F \le O(b)\varepsilon\|\tilde{A}_{j,0}\|_F,$$

which implies

$$\|\delta A_{j+1,0}\|_F \le O(b^2 n)\varepsilon\|\tilde{A}_{j,0}\|_F.$$

On the other hand, for $\hat{\mathbf{V}}_{j,k} = I_\ell \oplus \hat{V}_{j,k}$, $\hat{Q}_j = \hat{\mathbf{V}}_{j,b}\cdots\hat{\mathbf{V}}_{j,1}$ and $\tilde{Q}_j$, we can write once more

$$\begin{aligned} \tilde{A}_{j+1,0} &= \mathrm{fl}(\tilde{A}_{j,0}\tilde{Q}_j^T) = \mathrm{fl}(\tilde{A}_{j,0} - \mathrm{fl}(\tilde{X}_j\tilde{W}_j^T)) = \\ &= \tilde{A}_{j,0} - \hat{X}_j\hat{W}_j^T + \delta A_{j+1,0} = \\ &= \tilde{A}_{j,0} - \hat{X}_j(:,1\colon k-1)\hat{W}_j(:,1\colon k-1)^T - \hat{X}_j(:,k\colon b)\hat{W}_j(:,k\colon b)^T + \\ &\quad + \delta A_{j+1,0} = \\ &= [\tilde{A}_{j,0} - \hat{X}_j(:,1\colon k-1)\hat{W}_j(:,1\colon k-1)^T]\hat{\mathbf{V}}_{j,k}\cdots\hat{\mathbf{V}}_{j,b} + \delta A_{j+1,0}, \end{aligned}$$

and

$$\begin{aligned} F &= ((\cdots(((\tilde{A}_{j,0} - \hat{X}_j(:,1\colon k-1)\hat{W}_j(:,1\colon k-1)^T)\hat{\mathbf{V}}_{j,k}\cdots\hat{\mathbf{V}}_{j,b} + \delta A_{j+1,0})\cdot \\ &\quad \cdot\hat{Q}_{j+1}^T + \delta A_{j+2,0})\cdots)\cdot\hat{Q}_{n_u-1}^T + \delta A_{g+1,r-2})\hat{Q}_{n_u}^T + \delta A_{g+1,r-1} = \\ &= (\tilde{A}_{j,0} - \hat{X}_j(:,1\colon k-1)\hat{W}_j(:,1\colon k-1)^T)\hat{\mathbf{V}}_{j,k}\cdots\hat{\mathbf{V}}_{j,b}\hat{Q}_{j+1}^T\cdots\hat{Q}_{n_u}^T + \\ &\quad + \sum_{i=j+1}^{g}\delta A_{i+1,0}\hat{Q}_{i+1}^T\cdots\hat{Q}_{n_u}^T + \sum_{i=1}^{r-1}\delta A_{g+1,i}\hat{Q}_{g+i+1}^T\cdots\hat{Q}_{n_u}^T = \\ &= (\tilde{A}_{j,0} - \hat{X}_j(:,1\colon k-1)\hat{W}_j(:,1\colon k-1)^T)\hat{\mathbf{V}}_{j,k}\cdots\hat{\mathbf{V}}_{j,b}\hat{Q}_{j+1}^T\cdots\hat{Q}_{n_u}^T + \delta_1 F^{(j)}, \end{aligned} \tag{3.77}$$

where $n_u = g + r - 1$ is the total number of update steps, and

$$\|\delta_1 F^{(j)}\|_F \leq [O((g-j)\cdot b^2 \cdot n) + O((n-gb)n)]\varepsilon\|A\|_F \leq O(b\cdot n^2)\varepsilon\|A\|_F. \tag{3.78}$$

If we put all this together, for $\hat{\mathbf{V}}_{j,i}^{(\ell)} = I_{(i-k)} \oplus \hat{V}_{j,i} \in \mathbb{R}^{(n-\ell)\times(n-\ell)}$ and $\hat{Q}_i^{(\ell)} = \hat{Q}_i(\ell+1\colon n, \ell+1\colon n)$, where $i = k+1, \ldots, b$, from (3.77), (3.78), (3.74), (3.75) and (3.76) we will obtain

$$\begin{aligned}
&\tilde{u}_\ell^T \tilde{A}_{g+1,r-1}(:\,, \ell+1\colon n) = \tilde{u}_\ell^T F(:\,, \ell+1\colon n) = \\
&= \tilde{u}_\ell^T[(\tilde{A}_{j,0}(:\,, \ell+1\colon n) - \hat{X}_j(:\,, 1\colon k-1)\hat{W}_j(\ell+1\colon n, 1\colon k-1)^T)\cdot \\
&\quad \hat{V}_{j,k}\hat{\mathbf{V}}_{j,k+1}^{(\ell)}\cdots\hat{\mathbf{V}}_{j,b}^{(\ell)}(\hat{Q}_{j+1}^{(\ell)})^T\cdots(\hat{Q}_{n_u}^{(\ell)})^T(:\,, \ell+1\colon n) + \delta_1 F^{(j)}(:\,, \ell+1\colon n)] = \\
&= (\tilde{z}_{j,k}^T - \delta\hat{z}_{j,k}^T)\hat{V}_{j,k}\hat{\mathbf{V}}_{j,k+1}^{(\ell)}\cdots\hat{\mathbf{V}}_{j,b}^{(\ell)}(\hat{Q}_{j+1}^{(\ell)})^T\cdots(\hat{Q}_{n_u}^{(\ell)})^T(:\,, \ell+1\colon n) + \delta_3\hat{z}_{j,k}^T = \\
&= [\|\tilde{z}_{j,k}\|_2\ 0\ \cdots\ 0]\hat{\mathbf{V}}_{j,k+1}^{(\ell)}\cdots\hat{\mathbf{V}}_{j,b}^{(\ell)}(\hat{Q}_{j+1}^{(\ell)})^T\cdots(\hat{Q}_{n_u}^{(\ell)})^T(:\,, \ell+1\colon n) + \delta_4\hat{z}_{j,k}^T,
\end{aligned}$$

where

$$\|\delta_3\hat{z}_{j,k}\|_2 \leq O(b\cdot n^2)\varepsilon\|A\|_F,$$

so

$$\|\delta_4\hat{z}_\ell\|_2 \leq O(b^2 n + bm + bn^2)\varepsilon\|A\|_F.$$

Because $\hat{\mathbf{V}}_{j,k+1}^{(\ell)}, \ldots, \hat{\mathbf{V}}_{j,b}^{(\ell)}$, $\hat{Q}_{j+1}^{(\ell)}, \ldots, \hat{Q}_{n_u}^{(\ell)}$ does not have any effect on the ℓ-th coordinate (the 1-st coordinate of $\tilde{z}_{j,k}$) and they do not mix it with other coordinates, we can conclude that

$$\tilde{u}_\ell^T \tilde{A}_{g+1,r-1}(:\,, \ell+1\colon n) = [\|\tilde{z}_{j,k}\|_2\ 0\ \cdots\ 0] + \delta_4\hat{z}_{j,k}^T,$$

and

$$\tilde{u}_\ell^T \tilde{A}_{g+1,r-1}(:\,, \ell+2\colon n) = \delta_4\hat{z}_{j,k}(2\colon n-\ell)^T.$$

Finally we obtain the result

$$\|\tilde{u}_\ell^T \tilde{A}_{g+1,r-1}(:\,, \ell+2\colon n)\|_2 \leq O(b^2 n + bm + bn^2)\varepsilon\|A\|_F.$$

□

Now we can finally state the main theorem.

Theorem 3.6.4. *If $\tilde{B}$ is the bidiagonal matrix computed by Algorithm 3.5.1, then there exist an orthogonal $(m+n)\times(m+n)$ matrix $\hat{\mathcal{P}}$, an orthogonal $n\times n$ matrix $\hat{V}$ and backward perturbations ΔA, δA such that*

$$\begin{bmatrix} \tilde{B} \\ 0 \end{bmatrix} = \hat{\mathcal{P}}^T \begin{bmatrix} \Delta A \\ A + \delta A \end{bmatrix} \hat{V}, \quad \left\| \begin{bmatrix} \Delta A \\ \delta A \end{bmatrix} \right\|_F \leq \xi\|A\|_F, \tag{3.79}$$

where $0 \leq \xi \leq O(b(mn+n^3))\varepsilon$. The computed approximation $\tilde{V}$ of the matrix $\hat{V}$ satisfies $\|\tilde{V} - \hat{V}\|_F \leq O(n^2)\varepsilon$. Further, there exist an orthonormal $\hat{U}$ and a perturbation $\delta\hat{A}$ such that

$$A + \delta\hat{A} = \hat{U}\tilde{B}\hat{V}^T, \quad \|\delta\hat{A}\|_F \leq \sqrt{2}\xi\|A\|_F. \tag{3.80}$$

Proof. The proof is rather technical and we will divide it into three steps.

1. Step: The Householder transformations

We will set $F = \mathrm{fl}(AV) = \tilde{A}_{g+1,r-1}$, $\ell = (j-1)b+k$ and $r = n-gb$, where $\tilde{A}_{g+1,r-1}$ is the result of Algorithm 3.5.1 performed in finite precision arithmetic. Thus, in floating point computation we can use $f_\ell = F(:,\ell)$ instead of

$$f_\ell = \begin{cases} \tilde{A}_{j,k}(:,\ell) & \text{for } j = 1,\dots g,\ k = 1,\dots,b,\ \ell = 1,\dots,gb \\ \tilde{A}_{g+1,k}(:,\ell) & \text{for } k = 1,\dots,r-1,\ \ell = gb+1,\dots,n-1 \\ \tilde{A}_{g+1,r-1}(:,n) & \text{for } \ell = n, \end{cases}$$

because the denoted column will not be modified in successive steps of the algorithm (see Figure 3.8).

In this step of the proof we will analyze the application of Householder reflectors to the matrix A, in floating point arithmetic. This application is divided into g steps, where b columns of F are computed in each step, and in r remaining steps, where only one column of F is computed per step. First, we are investigating the computations performed in one block $j \in \{1,2,\dots,g\}$.

Lemma 3.6.1 gives the following error estimation:

$$\tilde{X}_j(:,1\colon k) = \tilde{A}_{j,0}\hat{Y}_j(:,1\colon k) + \delta\hat{X}_j(:,1\colon k),$$

where

$$\|\delta\hat{X}_j(:,1\colon k)\|_F = \sqrt{\sum_{i=1}^{k}\|\delta\hat{X}_j(:,i)\|_2^2} \le O(k^{\frac{3}{2}}n)\varepsilon\|\tilde{A}_{j,0}\|_F,$$

and

$$\tilde{W}_j(:,1\colon k) = \hat{W}_j(:,1\colon k) + \delta\hat{W}_j(:,1\colon k),$$

where

$$\|\delta\hat{W}_j(:,1\colon k)\|_F \le O(\sqrt{k}n)\varepsilon.$$

The only thing that remains to be checked is the error in the application of Householder reflectors to the matrix A. For $\hat{X}_j, \tilde{X}_j \in \mathbb{R}^{m\times b}$ and $\hat{W}_j, \tilde{W}_j \in \mathbb{R}^{n\times b}$, from the proof of Lemma 3.6.3 it follows that

$$\tilde{A}_{j+1,0} = \tilde{A}_{j,0}\hat{Q}_j^T + \delta A_{j+1,0},$$

where $\hat{Q}_j = I - \hat{W}_j\hat{Y}_j^T$, $\hat{X}_j = \tilde{A}_{j,0}\hat{Y}_j$, and

$$\|\delta A_{j+1,0}\|_F \le O(b^2n)\varepsilon\|\tilde{A}_{j,0}\|_F.$$

Finally, we obtain the result for $F = \tilde{A}_{g+1,r-1}$, where the first g updates are performed as shown above, and the last $r-1 = n-gb-1$ updates can be considered in the same framework but with $b = 1$. Let us denote $n_u = g+r-1$ as the total number of update steps. First we note that for $j = 1,\dots,g$

$$\begin{aligned} \|\tilde{A}_{j+1,0}\|_F &\le \|\tilde{A}_{j,0}\|_F + O(\varepsilon) \le \|\tilde{A}_{j-1,0}\|_F + O(\varepsilon) \le \cdots \le \|\tilde{A}_{2,0}\|_F + O(\varepsilon) \le \\ &\le \|A\|_F + O(\varepsilon). \end{aligned}$$

The same applies to the rest of the updates

$$\begin{aligned}\|F\|_F &\leq \|\tilde{A}_{g+1,r-2}\|_F + O(\varepsilon) \leq \cdots \leq \|\tilde{A}_{g+1,1}\|_F + O(\varepsilon) \leq \|\tilde{A}_{g+1,0}\|_F + O(\varepsilon) \leq \\ &\leq \|A\|_F + O(\varepsilon).\end{aligned}$$

Then, by induction we have

$$\begin{aligned}F &= ((\cdots((A\hat{Q}_1^T + \delta A_{2,0})\hat{Q}_2^T + \delta A_{3,0})\cdots)\hat{Q}_{n_u-1}^T + \delta A_{g+1,r-2})\hat{Q}_{n_u}^T + \delta A_{g+1,r-1} = \\ &= A\hat{Q}_1^T\hat{Q}_2^T\cdots\hat{Q}_{n_u}^T + \sum_{j=1}^{g}\delta A_{j+1,0}\hat{Q}_{j+1}^T\cdots\hat{Q}_{n_u}^T + \sum_{k=1}^{r-1}\delta A_{g+1,k}\hat{Q}_{g+k+1}^T\cdots\hat{Q}_{n_u}^T = \\ &= A\hat{V} + \delta_1 F\end{aligned}$$

where

$$\|\delta_1 F\|_F \leq [O(g\cdot b^2\cdot n) + O((n-gb)n)]\varepsilon\|A\|_F \leq O(bn^2)\varepsilon\|A\|_F.$$

At the end of this step of the proof, for $\hat{V} = \hat{Q}_1^T\hat{Q}_2^T\cdots\hat{Q}_{n_u}^T$ we can state that

$$F = (A + \delta_1 A)\hat{V}, \qquad \|\delta_1 A\|_F \leq \eta_F\|A\|_F, \quad \eta_F \leq O(bn^2)\varepsilon,$$

where $\delta_1 A = \delta_1 F\cdot\hat{V}^T$.

Step 2: The Gram–Schmidt orthogonalization and estimation of the backward error

This step is equivalent to **Step 2** in the proof of Theorem 3.2.6. Since the computation of $\tilde{B}$ from $F = [f_1, \ldots, f_n]$ corresponds to the modified Gram–Schmidt algorithm, we can use the results from [6] and represent the computation in an equivalent form, as the Householder QR factorization of the augmented matrix

$$\begin{bmatrix} 0 \\ F \end{bmatrix} = \begin{bmatrix} 0 \\ A + \delta_1 A \end{bmatrix}\hat{V}.$$

By Lemma 3.6.2, the following relations hold

$$\begin{aligned}\begin{bmatrix} \tilde{\phi}_{k+1}e_k + \tilde{\psi}_{k+1}e_{k+1} \\ 0 \end{bmatrix} &= \hat{P}_{k+1}\hat{P}_k\left\{\begin{bmatrix} 0 \\ f_{k+1}\end{bmatrix} + \begin{bmatrix} \Delta f_{k+1} \\ \delta f_{k+1}\end{bmatrix}\right\}, \\ \left\|\begin{bmatrix} \Delta f_{k+1} \\ \delta f_{k+1}\end{bmatrix}\right\|_2 &\leq O(bm)\varepsilon\|F\|_F,\end{aligned}$$

where

$$\hat{P}_k = I_{m+n} - \begin{bmatrix} -e_k \\ \hat{u}_k \end{bmatrix}\begin{bmatrix} -e_k^T & \hat{u}_k^T \end{bmatrix},$$

and $\hat{u}_k = \tilde{s}_k/\|\tilde{s}_k\|_2$ is the exact vector with $\|\hat{u}_k\|_2 = 1$. Putting all columns of $\tilde{B}$ together, we get

$$\begin{aligned}\begin{bmatrix} \tilde{B} \\ 0 \end{bmatrix} &= \left[\begin{bmatrix} \tilde{\psi}_1 e_1 \\ 0 \end{bmatrix}, \begin{bmatrix} \tilde{\phi}_2 e_1 + \tilde{\psi}_2 e_2 \\ 0 \end{bmatrix}, \ldots, \begin{bmatrix} \tilde{\phi}_n e_{n-1} + \tilde{\psi}_n e_n \\ 0 \end{bmatrix}\right] = \\ &= \left[\hat{P}_1\begin{bmatrix} \Delta f_1 \\ f_1 + \delta f_1 \end{bmatrix}, \hat{P}_2\hat{P}_1\begin{bmatrix} \Delta f_2 \\ f_2 + \delta f_2 \end{bmatrix}, \ldots, \hat{P}_n\hat{P}_{n-1}\begin{bmatrix} \Delta f_n \\ f_n + \delta f_n \end{bmatrix}\right],\end{aligned}$$

and by using the fact that

$$\hat{P}_i \begin{bmatrix} \tilde{B}(:,j) \\ 0 \end{bmatrix} = \begin{bmatrix} \tilde{\phi}_j e_{j-1} + \tilde{\psi}_j e_j \\ 0 \end{bmatrix} = \begin{bmatrix} \tilde{B}(:,j) \\ 0 \end{bmatrix}, \text{ for all } i \neq j, j-1,$$

we obtain

$$\begin{aligned} \begin{bmatrix} \tilde{B} \\ 0 \end{bmatrix} = & \left[\hat{P}_n \hat{P}_{n-1} \cdots \hat{P}_2 \hat{P}_1 \begin{bmatrix} \Delta f_1 \\ f_1 + \delta f_1 \end{bmatrix}, \hat{P}_n \hat{P}_{n-1} \cdots \hat{P}_2 \hat{P}_1 \begin{bmatrix} \Delta f_2 \\ f_2 + \delta f_2 \end{bmatrix}, \right. \\ & \hat{P}_n \hat{P}_{n-1} \cdots \hat{P}_3 \hat{P}_2 \begin{bmatrix} \Delta f_3 \\ f_3 + \delta f_3 \end{bmatrix}, \hat{P}_n \hat{P}_{n-1} \cdots \hat{P}_4 \hat{P}_3 \begin{bmatrix} \Delta f_4 \\ f_4 + \delta f_4 \end{bmatrix}, \\ & \left. \cdots, \hat{P}_n \hat{P}_{n-1} \hat{P}_{n-2} \begin{bmatrix} \Delta f_{n-1} \\ f_{n-1} + \delta f_{n-1} \end{bmatrix}, \hat{P}_n \hat{P}_{n-1} \begin{bmatrix} \Delta f_n \\ f_n + \delta f_n \end{bmatrix} \right]. \end{aligned}$$

The k–th column of the computed bidiagonal matrix is of the form

$$\hat{P}_n \hat{P}_{n-1} \cdots \hat{P}_k \hat{P}_{k-1} \begin{bmatrix} \Delta f_k \\ f_k + \delta f_k \end{bmatrix},$$

and the desired form is

$$\hat{P}_n \hat{P}_{n-1} \cdots \hat{P}_2 \hat{P}_1 \begin{bmatrix} \hat{\Delta} f_k \\ f_k + \hat{\delta} f_k \end{bmatrix} = \hat{\mathcal{P}}^T \begin{bmatrix} \hat{\Delta} f_k \\ f_k + \hat{\delta} f_k \end{bmatrix}, \quad \mathcal{P} = \hat{P}_1 \hat{P}_2 \cdots \hat{P}_{n-1} \hat{P}_n.$$

The first two columns ($k = 1, 2$) are already in the desired form and $\hat{\Delta} f_k = \Delta f_k$, $\hat{\delta} f_k = \delta f_k$. For $k \geq 3$ we write

$$\begin{bmatrix} \tilde{B}(:,k) \\ 0 \end{bmatrix} = (\hat{P}_n \hat{P}_{n-1} \cdots \hat{P}_k \hat{P}_{k-1} \overbrace{\hat{P}_{k-2} \cdots \hat{P}_2 \hat{P}_1)(\hat{P}_1 \hat{P}_2 \cdots \hat{P}_{k-2})}^{I} \begin{bmatrix} \Delta f_k \\ f_k + \delta f_k \end{bmatrix},$$

and then

$$\begin{aligned} \hat{P}_1 \hat{P}_2 \cdots \hat{P}_{k-2} \begin{bmatrix} \Delta f_k \\ f_k + \delta f_k \end{bmatrix} = & \begin{bmatrix} 0 \\ f_k \end{bmatrix} + \begin{bmatrix} \Delta_1 f_k \\ \delta_1 f_k \end{bmatrix} + \hat{P}_1 \begin{bmatrix} \Delta_2 f_k \\ \delta_2 f_k \end{bmatrix} + \hat{P}_1 \hat{P}_2 \begin{bmatrix} \Delta_3 f_k \\ \delta_3 f_k \end{bmatrix} + \cdots \\ & \cdots + \hat{P}_1 \cdots \hat{P}_{k-3} \begin{bmatrix} \Delta_{k-2} f_k \\ \delta_{k-2} f_k \end{bmatrix} + \hat{P}_1 \cdots \hat{P}_{k-2} \begin{bmatrix} \Delta f_k \\ \delta f_k \end{bmatrix} = \\ = & \begin{bmatrix} 0 \\ f_k \end{bmatrix} + \begin{bmatrix} \hat{\Delta} f_k \\ \hat{\delta} f_k \end{bmatrix}, \end{aligned}$$

where

$$\begin{bmatrix} \Delta_j f_k \\ \delta_j f_k \end{bmatrix} = \begin{bmatrix} e_j \\ -\hat{u}_j \end{bmatrix} (\hat{u}_j^T f_k), \quad j = 1, \ldots, k-2,$$

and with

$$\begin{bmatrix} \hat{\Delta} f_k \\ \hat{\delta} f_k \end{bmatrix} = \hat{P}_1 \cdots \hat{P}_{k-2} \begin{bmatrix} \Delta f_k \\ \delta f_k \end{bmatrix} + \begin{bmatrix} \Delta_1 f_k \\ \delta_1 f_k \end{bmatrix} + \sum_{j=2}^{k-2} \hat{P}_1 \cdots \hat{P}_{j-1} \begin{bmatrix} \Delta_j f_k \\ \delta_j f_k \end{bmatrix}.$$

Hence,

$$\begin{bmatrix} \tilde{B} \\ 0 \end{bmatrix} = \hat{\mathcal{P}}^T \left[\begin{bmatrix} \hat{\Delta} f_1 \\ f_1 + \hat{\delta} f_1 \end{bmatrix}, \ldots, \begin{bmatrix} \hat{\Delta} f_k \\ f_k + \hat{\delta} f_k \end{bmatrix}, \ldots, \begin{bmatrix} \hat{\Delta} f_n \\ f_n + \hat{\delta} f_n \end{bmatrix} \right] = \hat{\mathcal{P}}^T \left\{ \begin{bmatrix} 0 \\ F \end{bmatrix} + \begin{bmatrix} \Delta F \\ \delta F \end{bmatrix} \right\},$$

where, after suitable reordering of the entries in the sums,

$$\begin{bmatrix} \Delta F \\ \delta F \end{bmatrix} = \left[\begin{bmatrix} \Delta f_1 \\ \delta f_1 \end{bmatrix}, \begin{bmatrix} \Delta f_2 \\ \delta f_2 \end{bmatrix}, \ldots, \hat{P}_1 \cdots \hat{P}_{k-2} \begin{bmatrix} \Delta f_k \\ \delta f_k \end{bmatrix}, \ldots, \hat{P}_1 \cdots \hat{P}_{n-2} \begin{bmatrix} \Delta f_n \\ \delta f_n \end{bmatrix} \right] + + \sum_{j=1}^{n-2} \hat{P}_1 \cdots \hat{P}_{j-1} \left[\underbrace{0, \ \ldots \ , 0}_{j+1}, \ \begin{bmatrix} \Delta_j f_{j+2} \\ \delta_j f_{j+2} \end{bmatrix}, \ldots, \begin{bmatrix} \Delta_j f_n \\ \delta_j f_n \end{bmatrix} \right]$$

Taking norms, we obtain

$$\begin{aligned} \left\| \begin{bmatrix} \Delta F \\ \delta F \end{bmatrix} \right\|_F &\leq O(bm\sqrt{n})\varepsilon \|F\|_F + \sqrt{2} \sum_{j=1}^{n-2} \|\hat{u}_j^T \begin{bmatrix} f_{j+2} & f_{j+3} & \cdots & f_n \end{bmatrix} \|_2 \leq \\ &\leq O(bm\sqrt{n})\varepsilon \|F\|_F + \sqrt{2} \sum_{j=1}^{n-2} (\|\tilde{u}_j^T \tilde{A}_{g+1,r-1}(:, j+2\colon n)\|_2 + \\ &\quad + \|\delta \hat{u}_j\|_2 \|F(:, j+2\colon n)\|_F). \end{aligned}$$

It remains to estimate the products $\tilde{u}_i^T f_\ell$ for $\ell = 3, \ldots, n$ and $i = 1, \ldots, \ell - 2$, where $\ell = (j-1)b + k$, $k = 1, \ldots, b$. For this estimate, the important role plays the choice of the vector $z_{j,k}$. From Lemma 3.6.3 it follows that

$$\|\tilde{u}_\ell^T \tilde{A}_{g+1,r-1}(:, \ell+2\colon n)\|_2 \leq O(b^2 n + bm + bn^2)\varepsilon \|A\|_F.$$

Then,

$$\left\| \begin{bmatrix} \Delta F \\ \delta F \end{bmatrix} \right\|_F \leq O(b(mn + n^3))\varepsilon \|F\|_F \leq O(b(mn + n^3))(1 + \eta_F)\|A\|_F.$$

To get the relation (3.79), we collect the perturbations from both implicit tridiagonalization and the Gram-Schmidt computation,

$$\begin{aligned} \begin{bmatrix} \tilde{B} \\ 0 \end{bmatrix} &= \hat{\mathcal{P}}^T \left\{ \begin{bmatrix} 0 \\ F \end{bmatrix} + \begin{bmatrix} \Delta F \\ \delta F \end{bmatrix} \right\} = \hat{\mathcal{P}}^T \left\{ \begin{bmatrix} 0 \\ A + \delta_1 A \end{bmatrix} \hat{V} + \begin{bmatrix} \Delta F \\ \delta F \end{bmatrix} \right\} \\ &= \hat{\mathcal{P}}^T \left\{ \begin{bmatrix} 0 \\ A + \delta_1 A \end{bmatrix} + \begin{bmatrix} \Delta F \\ \delta F \end{bmatrix} \hat{V}^T \right\} \hat{V}. \end{aligned}$$

Step 3: The final result

Finally, using $\mathcal{P}_{11} = \mathcal{P}(1\colon n, 1\colon n)$, $\mathcal{P}_{21} = \mathcal{P}(n+1\colon n+m, 1\colon n)$, we have

$$\begin{bmatrix} \Delta A \\ A + \delta A \end{bmatrix} \hat{V} = \begin{bmatrix} \mathcal{P}_{11} \\ \mathcal{P}_{21} \end{bmatrix} \tilde{B}, \quad \mathcal{P}_{11}^T \mathcal{P}_{11} + \mathcal{P}_{21}^T \mathcal{P}_{21} = I,$$

and relation (3.80) follows by an application of [6, Lemma 3.1]. The proof that relation (3.80) holds for the non-block version of the algorithm is given in Theorem 3.2.6 and in Theorem 3.18 [2]. The same arguments can be applied to the block version.

□

In our numerical experiments, the optimal choice for the block dimension b was mostly 16, so the result of Theorem 3.6.4 is close to the result of Theorem 3.2.6.

Example 3.6.5. *Let $A = [a_{ij}]$ be the $n \times n$ Kahan matrix as in [2], with*

$$a_{ij} = \begin{cases} \alpha^{i-1} & i = j \\ -\alpha^{j-1}\beta & i > j \end{cases},$$

where $\alpha^2 + \beta^2 = 1$ and $\alpha, \beta > 0$. For our tests we chose $\alpha = \sin(1.2)$ and $n = 50, 60, \ldots, 200$. In this case the matrices are ill-conditioned, whose first $n-1$ singular values gradually decay and are bounded away from zero. On the other hand, the smallest singular value decays rapidly with n.

We compare the accuracy of Algorithm 3.5.1 with Algorithm 3.1.1 and Ralha's one–sided bidiagonalization, by measuring the Wielandt–Hoffman measure

$$\frac{\sqrt{\sum_{k=1}^{n}(\sigma_k(A) - \sigma(B))^2}}{\|A\|_F}. \tag{3.81}$$

The singular values $\sigma_k(A)$ of the matrix A are computed by the MATLAB command `svd()`*. The results are shown in Figure 3.9.*

We can note that Algorithm 3.5.1 sometimes produces the bidiagonal matrix B with slightly less accurate singular values, than Algorithm 3.1.1. Theorem 3.6.4 asserts that the bound on (3.81) for Algorithm 3.5.1 is b times larger than the corresponding bound for Algorithm 3.1.1, where b is the block dimension. In our case we took $b = 16$. If we compare the computed errors measured by (3.81), we can see that the largest difference is obtained for $n = 180$, where the error of Algorithm 3.5.1 is 1.67 times larger than the error of Algorithm 3.1.1. In this case, the estimation of the error bounds on (3.81) from Theorem 3.2.6 and Theorem 3.6.4 are :

Algorithm 3.1.1	*Algorithm 3.5.1*
$(n^2 + n^3)\varepsilon = 6.51 \cdot 10^{-10}$	$b(n^2 + n^3)\varepsilon = 1.04 \cdot 10^{-8}$

Hence, our computed errors satisfy both of the theorems.

Remark 3.6.6. *The statements of Corollary 3.2.13, Corollary 3.2.14, Corollary 3.2.17 and Proposition 3.2.18 still hold for Algorithm 3.5.1, only the bounds are multiplied by the block dimension b.*

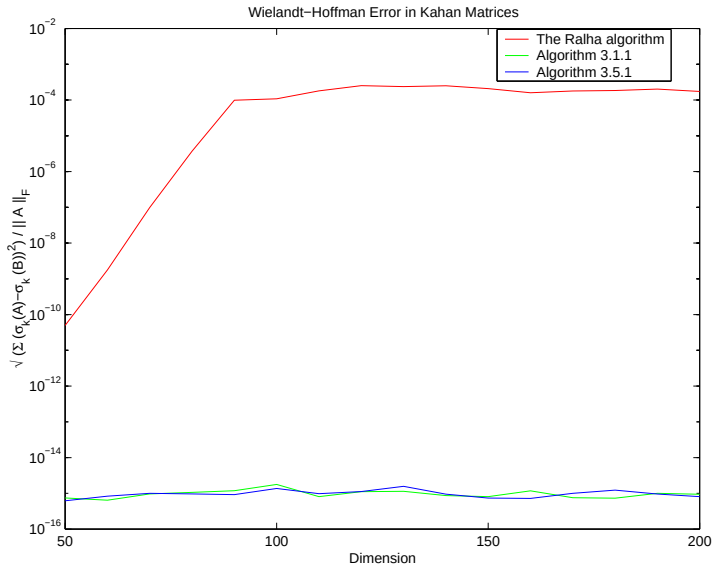


Figure 3.9: Error in singular values from Example 3.6.5.

3.7 Efficiency of the Block Version

For the block version of Barlow's one–sided bidiagonalization, extensive testing was carried out, too. Computations where performed in the same laboratory as before, and with the same type of matrices.

The block dimension b in the tests was chosen to obtain the best execution time. In our case it turned out to be $b = 16$. The influence of the parameter c on the execution time was negligible. In our tests we took $c = 8$. Table 3.4 gives average execution times for full SVD algorithms, expressed in seconds.

m	×	n	t_1	t_2	t_L	$p_{2,L}$	$p_{2,1}$
100	×	100	0.01	0.01	0.01	0.00%	0.00%
200	×	200	0.14	**0.13**	0.15	13.33%	7.14%
500	×	50	0.01	*0.01	0.01	0.00%	0.00%
500	×	100	0.05	*0.04	0.04	0.00%	20.00%
500	×	500	3.87	**3.40**	3.63	6.34%	16.02%
1000	×	100	0.14	*0.09	0.09	0.00%	35.71%
1000	×	500	6.19	***4.04**	4.45	9.21%	35.06%
1000	×	1000	39.19	**34.55**	37.43	7.69%	12.96%
2000	×	200	1.46	***0.58**	0.60	3.33%	46.30%
2000	×	1000	55.25	***39.22**	41.20	4.81%	28.27%
2000	×	2000	359.05	**324.59**	336.49	3.54%	12.22%
3000	×	3000	1514.46	**1261.34**	1318.24	4.32%	17.81%

Table 3.4: Average execution times for full SVD algorithms.

The meaning of the headers in Table 3.4 are as follows:

t_1	—	the SVD with Algorithm 3.1.1 for bidiagonalization. The LAPACK routine `sbdsqr()` is used for the SVD of a bidiagonal matrix, which implements the bidiagonal QR algorithm.
t_2	—	the SVD with Algorithm 3.5.1 for bidiagonalization. The LAPACK routine `sbdsqr()` is used for the SVD of a bidiagonal matrix, which implements the bidiagonal QR algorithm.
t_L	—	the LAPACK `sgesvd()` routine.
$p_{2,L} = 100(t_L - t_2)/t_L$	—	the percentage of time decrease, when the SVD with Algorithm 3.5.1 is compared to the LAPACK routine.
$p_{2,1} = 100(t_1 - t_2)/t_1$	—	the percentage of time decrease, when the SVD with Algorithm 3.5.1 is compared to the SVD with Algorithm 3.1.1.
*	—	QR factorization is performed before the SVD as it is in `sgesvd()`.

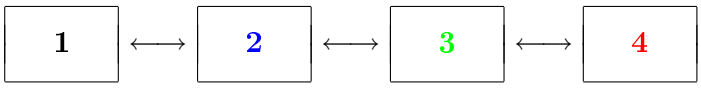
We can conclude that the block version of the one–sided bidiagonalization algorithm did decrease the execution time of Algorithm 3.1.1, as expected. Compared to the SVD with Algorithm 3.1.1 the most significant time decrease is 46.30% for matrix dimensions 2000×200. The SVD routine with Algorithm 3.5.1 produces a code that is not slower than the LAPACK `sgesvd()` routine in case when all of the SVD factors are required, although this varies with the dimensions of the matrix. In many cases we observed some gains in speed. If the matrix U is not needed then the advantage of the one–sided bidiagonalization over the LAPACK routine might be lost. That happens because U is always computed, whether it is needed or not (see [2, Table 1]). When solving the problems described in Section 3.3 ([8]), our algorithm would be preferable.

3.8 Parallel Version

The parallel bidiagonalization algorithm is performed on several processors simultaneously. Each matrix is distributed over the memories of processors, and this distribution is balanced. This means that the dimensions of the submatrices assigned to each processor are almost the same. The communication between processors is optimized, because interprocessor communication is most time consuming.

In our case we used following propositions:

- the processors were organized in linear order,



- we used ScaLAPACK [7] for the computation,
- we used MPI [38] for interprocessor communication.

The matrix distribution over the process is performed row-wise, because the algorithm is one–sided and column oriented.

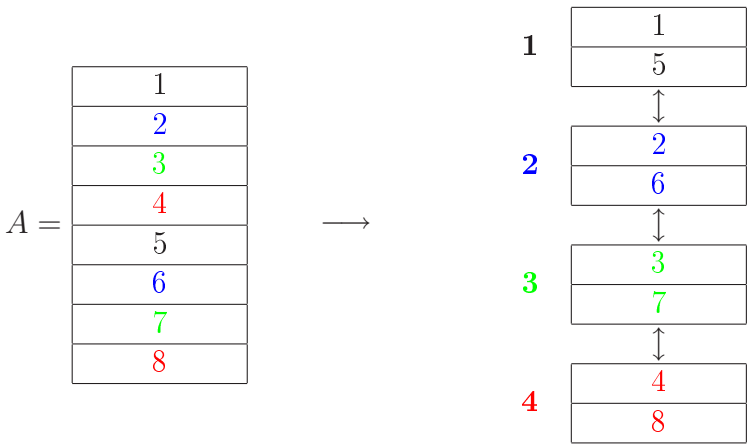


Figure 3.10: The block distribution of the matrix A

The most important features of the parallel version of the Barlow bidiagonalization algorithm are the following:

1. The matrix layout is one-dimensional block-cyclic row distribution. Each $m \times n$ matrix is divided in $m_b \times n$ blocks of continuous rows, where m_b is block row dimension. Then, the blocks are distributed across the processors in cyclic order, which guarantees good load balancing (see [7]).

2. The algorithm is performed in the same way as Algorithm 3.1.1 for $\phi_{k+1} = \gamma_k$, with extra interprocessor communication. Interprocessor communication is required for:

   - computation of z_k as matrix–vector multiplication,
   - broadcasting Householder vector v_k to all processors,
   - broadcasting ϕ_{k+1},
   - computing scalar products.

   The rest of the computations consists of BLAS 1 operations (operations with vectors), as well as computation and application of Householder reflectors, which need no communication.

The complete parallel algorithm with explanations is listed in Algorithm 3.8.1.

Algorithm 3.8.1 (The parallel Barlow one-sided bidiagonalization). *For $A \in \mathbb{R}^{m\times n}$, $\operatorname{rank}(A) = n > 2$, this algorithm computes in parallel an orthonormal $U = [u_1, \ldots, u_n]$, a bidiagonal B and an orthogonal V such that $A = UBV^T$.*

(1) Distribute $\Psi = [\ \psi_1 \quad \dots \quad \psi_n\ ]^T$ over the processors;
(2) Distribute $\Phi = [\ \phi_1 \quad \dots \quad \phi_{n-1}\ ]^T$ over the processors;
(3) $A_0 = A$;
(4) $f_1 = A(:,1);\ \psi_1 = \|f_1\|_2$; *Parallel dot product;*
(5) $u_1 = f_1/\psi_1$; *BLAS 1 operation without communication;*

for $k = 1\colon n-2$
(6) $z_k = A_{k-1}(:,k+1\colon n)^T u_k$; *Parallel matrix-vector product the resulting vector is stored in one processor;*
(7) $[\phi_{k+1}, v_k] =$ **householder**(z_k); *Computation performed on one processor;*
(8) $A_k(:,1\colon k) = A_{k-1}(:,1\colon k)$;
(9) $A_k(:,k+1\colon n) = A_{k-1}(:,k+1\colon n) - A_{k-1}(:,k+1\colon n)v_k v_k^T$; *$v_k$ broadcasted to all processors, parallel update;*
(10) $f_{k+1} = A_k(:,k+1)$;
(11) Broadcast ϕ_{k+1} to all processors;
(12) $s_{k+1} = f_{k+1} - \phi_{k+1}u_k$; *BLAS 1 operation without communication;*
(13) $\psi_{k+1} = \|s_{k+1}\|_2$; *Parallel dot product;*
(14) $u_{k+1} = s_{k+1}/\psi_{k+1}$; *BLAS 1 operation without communication;*

end;
(15) $f_n = A_{n-2}(:,n);\ \phi_n = u_{n-1}^T f_n$; *Parallel dot product;*
(16) $s_n = f_n - \phi_n u_{n-1}$; *BLAS 1 operation without communication;*
(17) $\psi_n = \|s_n\|_2$; *Parallel dot product;*
(18) $u_n = s_n/\psi_n$; *BLAS 1 operation without communication;*
(19) $V^T =$ **householder_product**$(v_1, \dots, v_{n-2})$ *Parallel computation;*
end.

3.9 Numerical Stability of the Parallel Version

The parallel version of Barlow's bidiagonalization algorithm performs the same operations as the serial non-block version. Preliminary numerical experiments showed that a parallel block version has a large overhead on our computers, thus it was almost always slower than the ScaLAPACK routine. The results of Theorem 3.2.6 hold for this version as well.

3.10 Efficiency of the Parallel Version

The tests for the parallel version of the Barlow bidiagonalization algorithm were done over a large variety of matrix dimensions. The computations were performed in the same laboratory as before and matrices were generated as described in Section 3.4. QR factorization was not performed before bidiagonalization, because Algorithm 3.8.1 is suitable for the parallel computing just the way it is. The QR factorization would just increase the interprocessor communication. The linear layout of the processors may not always be optimal for the ScaLAPACK routine, so we performed our test with all possible layouts for the fixed processor number and we chose the best execution time. Table 3.5 gives the average execution times expressed in seconds for full SVD algorithms when computed on p processors. In fact, this time represents the worst time on all p processors.

$m \times n$	p	t_3	$p_m \times p_n$	t_S	$p_{3,S}$	$\eta_{3,p}$	$\eta_{S,p}$
1000 × 100	4	**0.26**	4×1	0.60	56.67%	0.0865	0.0375
1000 × 500	4	**3.11**	4×1	4.95	37.17%	0.3248	0.2247
1000 × 1000	4	**12.08**	4×1	16.66	27.49%	0.7150	0.5617
	8	**9.37**	4×2	18.25	48.66%	0.4609	0.2564
	16	**8.08**	16×1	19.52	58.61%	0.2672	0.1198
2000 × 200	4	**0.80**	4×1	1.65	51.51%	0.1812	0.0909
2000 × 1000	4	**16.09**	4×1	19.25	16.42%	0.6094	0.5351
	8	**11.73**	4×2	20.36	42.39%	0.4179	0.2529
2000 × 2000	4	**98.47**	4×1	109.95	10.44%	0.8241	0.7651
	8	**48.98**	4×2	66.90	26.79%	0.8284	0.6287
	16	**30.93**	16×1	58.90	47.49%	0.6559	0.3571
4000 × 200	8	**1.10**	8×1	2.69	59.11%	—	—
4000 × 1000	8	**15.66**	8×1	21.64	27.63%	—	—
	16	**11.47**	16×1	22.11	48.12%	—	—
4000 × 4000	8	**420.96**	8×1	448.84	6.21%	—	—
	16	**178.91**	4×4	236.47	24.34%	—	—
5000 × 100	8	**0.45**	8×1	1.76	74.43%	—	—
5000 × 1000	16	**12.38**	16×1	22.68	45.41%	—	—
5000 × 5000	16	**362.16**	4×4	435.27	16.80%	—	—
8000 × 1000	16	**16.47**	16×1	25.10	34.38%	—	—
8000 × 8000	16	**2335.30**	4×4	2445.79	4.52%	—	—
10000 × 1000	16	**18.26**	16×1	26.20	30.31%	—	—
10000 × 10000	16	**3324.75**	4×4	3395.03	2.07%	—	—

Table 3.5: Average execution times for full parallel SVD algorithms.

The meaning of the headers in Table 3.5 are as follows:

t_3	—	the parallel SVD with Algorithm 3.8.1.
$p_m \times p_n$	—	processor layout with the best execution time of the ScaLAPACK routine.
t_S	—	the ScaLAPACK `psgesvd()` routine.
$p_{3,S} = 100(t_S - t_3)/t_S$	—	the percentage of time decrease, when the parallel SVD with Algorithm 3.8.1 is compared to the ScaLAPACK routine.
$\eta_{3,p} = (t_2/t_3)/p$	—	the efficiency of the parallel SVD with Algorithm 3.8.1 on p processors.
$\eta_{S,p} = (t_L/t_S)/p$	—	the efficiency of the ScaLAPACK routine on p processors.

As we can see from Table 3.5, we accomplished a considerable decrease in execution time. The SVD with the described parallel version of the one–sided bidiagonalization algorithm is much faster than the ScaLAPACK routine `psgesvd()`. Compared to the ScaLAPACK routine, the most significant time decrease is 74.43% for matrix dimensions 5000×100 and for 8 processors.

Another important feature of parallel algorithms is the efficiency. In ideal situation an algorithm executed on p processors should be p times faster than the same algorithm executed on only one processor. The efficiency measures departure from the ideal execution time. Table 3.5 shows the efficiency for both SVD algorithms applied to matrices with small dimensions. In case of larger dimensions we were not able to apply the algorithms on a single processor due to memory limitation, and therefore the efficiency is not computed. We can see that the parallel SVD with Algorithm 3.8.1 has better efficiency than the ScaLAPACK routine `psgesvd()`. The new algorithm has also better scalability than the ScaLAPACK routine, which is illustrated in Figure 3.11. The y axis in Figure 3.11 represents the reduction factor in execution time when the number of processors is doubled and the matrix dimensions are fixed. The labels on the x axis denote matrix dimensions and ratios p_1/p_2, which indicate that the number of processors is increased from p_1 to p_2. We can conclude that in all observed cases the parallel SVD algorithm with the one–sided bidiagonalization reduces the execution time by larger factor than the corresponding ScaLAPACK routine. In ideal situation this factor should be equal to 2, and in our test we obtained the optimal factors 2.35 for the SVD algorithm with the one–sided bidiagonalization, and 1.90 for the ScaLAPACK `psgesvd()` routine. The both optimal factors were obtained for a 4000×4000 matrix, when going from 8 to 16 processors. We can also observe that the efficiency degrades more rapidly for the ScaLAPACK routine than for Algorithm 3.8.1 when the number of processors is increasing.

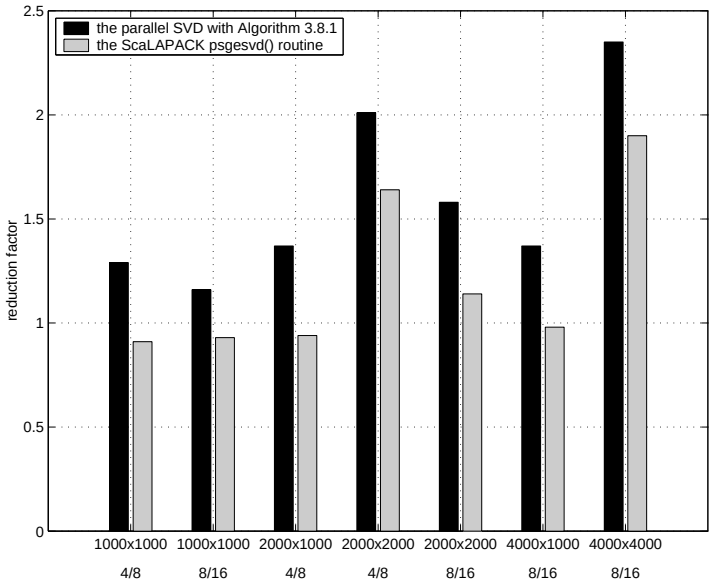


Figure 3.11: Reduction in execution time in case when the number of processors is doubled.

Chapter 4

The Symmetric Eigenvalue Problem

4.1 Definitions and Properties

Many mathematical models originating from physics and engineering reduce to eigenvalue problems, like solving the Helmholtz equation concerning electro-magnetic waves, determining vibration frequencies in structural mechanics, or partitioning a set of objects into different groups. The discretization of a differential equation in a physical model, or matrix interpretation of a problem in graph theory, result with a matrix eigenvalue problem. Matrices obtained in this way are usually very large and structured. Modern computers allow us to solve such large problems, with specially designed methods for large and structured matrices. Most of these methods are iterative, so special attention must be paid to their accuracy and efficiency.

Let us start with a definition.

Definition 4.1.1. *Let $A \in \mathbb{C}^{n\times n}$. Scalar $\lambda \in \mathbb{C}$ is called* ***eigenvalue*** *of A, if there exists a vector $u \in \mathbb{C}^n$, $u \neq 0$, such that*

$$Au = \lambda u.$$

Vector u is called the ***eigenvector*** *belonging to the eigenvalue λ.*

We will be mostly concerned with real symmetric matrices, that is

- $A \in \mathbb{R}^{n\times n}$,
- $A^T = A$,

and they possess very convenient properties, regarding eigenvalues and eigenvectors.

Theorem 4.1.2 ([78, p. 7]). *All eigenvalues of real symmetric matrices are real.*

As a result of Theorem 4.1.2, we may label eigenvalues of a symmetric matrix in increasing order

$$\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n. \tag{4.1}$$

If $u_1, \ldots, u_n$ are the corresponding eigenvectors, then they are orthogonal.

Theorem 4.1.3 ([78, p. 7]). *If $\lambda_k \neq \lambda_j$ then $u_k^T u_j = 0$.*

Note that Theorem 4.1.3 claims that only the eigenvectors belonging to different eigenvalues are orthogonal. When an eigenvalue is multiple, then specific subspaces must be introduced. The null space of $A - \lambda I$

$$\mathcal{N}_\lambda = \{x \in \mathbb{R}^n : (A - \lambda I)x = 0\}$$

is called the **eigenspace** belonging to λ, and represents an invariant subspace. The multiplicity of λ is the dimension of $\mathcal{N}_\lambda$. The consequence of the next theorem is that all invariant subspaces are spanned by eigenvectors.

Theorem 4.1.4 (The spectral theorem [78, p. 7]). *Let $A \in \mathbb{R}^{n\times n}$ be a symmetric matrix, then there exist an orthogonal matrix $U = [u_1, \ldots, u_n] \in \mathbb{R}^{n\times n}$ and a diagonal matrix $\Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_n) \in \mathbb{R}^{n\times n}$ such that*

$$A = U\Lambda U^T = \sum_{i=1}^{n} \lambda_i u_i u_i^T.$$

Scalars $\lambda_1, \ldots, \lambda_n$ are eigenvalues of A, and vectors $u_1, \ldots, u_n$ are orthonormal eigenvectors of A.

Matrices that originate from physical models are often symmetric matrices with one more property.

Definition 4.1.5.

- *Symmetric matrix $A \in \mathbb{R}^{n\times n}$ is called **positive definite** if $x^T Ax > 0$ for all $x \in \mathbb{R}^n$, $x \neq 0$.*
- *Symmetric matrix $A \in \mathbb{R}^{n\times n}$ is called **positive semidefinite** if $x^T Ax \geq 0$ for all $x \in \mathbb{R}^n$.*

Such matrices have specific eigenvalues:

Corollary 4.1.6.

- *All eigenvalues of symmetric positive definite matrices are positive.*
- *All eigenvalues of symmetric positive semidefinite matrices are nonnegative.*

Proof. Let $u_i \in \mathbb{R}^{n\times n}$ be an eigenvector from Theorem 4.1.4. Then

- $0 < u_i^T A u_i = \lambda_i$
- $0 \leq u_i^T A u_i = \lambda_i$

□

Recall that, by Theorem 2.1.6 and Theorem 2.1.7, squares of singular values of a rectangular matrix A correspond to eigenvalues of symmetric positive semidefinite matrices A^TA and AA^T, and singular values of A correspond to eigenvalues of the Jordan–Wielandt matrix up to the sign.

There are many important properties of eigenvalues of a symmetric matrix. First of them claims that these eigenvalues satisfy the following "minimax" characterization.

Theorem 4.1.7 (Courant–Fischer Minimax Theorem [35, p. 411]). *If $A \in \mathbb{R}^{n\times n}$ is symmetric, with eigenvalues ordered as in (4.1), then*

$$\lambda_k = \min_{\substack{\mathcal{S}\subset\mathbb{R}^n \\ \dim(\mathcal{S})=k}} \max_{\substack{x\in\mathcal{S} \\ \|x\|_2=1}} x^TAx, \quad k = 1,\dots,n.$$

Some other properties follow from the previous theorem.

Corollary 4.1.8 ([35, p. 411]). *If $A \in \mathbb{R}^{n\times n}$ and $A + E \in \mathbb{R}^{n\times n}$ are symmetric matrices, then*

$$\lambda_i(A) + \lambda_1(E) \le \lambda_i(A + E) \le \lambda_i(A) + \lambda_n(E), \quad i = 1,\dots,n,$$

where $\lambda_i(M)$, $i = 1,\dots,n$ are eigenvalues of the matrix M, ordered as in (4.1).

Corollary 4.1.9 (Interlacing Property [35, p. 411]). *If A_k denotes a $k \times k$ principal submatrix of a symmetric matrix $A \in \mathbb{R}^{n\times n}$, then for $k = 1,\dots,n$ the following interlacing property holds:*

$$\lambda_i(A) \le \lambda_i(A_k) \le \lambda_{i+n-k}(A), \quad i = 1,\dots,k,$$

where $\lambda_i(M)$, $i = 1,\dots,$ are eigenvalues of the matrix M, ordered as in (4.1).

Theorem 4.1.10 (Ky–Fan Minimum Property [49, p. 229]). *If $A \in \mathbb{R}^{n\times n}$ is symmetric, with eigenvalues ordered as in (4.1), then for $k = 1,\dots,n$ the following equation holds*

$$\sum_{i=1}^{k} \lambda_i = \min_{\substack{X\in\mathbb{R}^{n\times k} \\ X^TX=I_k}} \operatorname{trace}(X^TAX).$$

Moreover,

$$\sum_{i=1}^{k} \lambda_i = \min_{\substack{X\in\mathbb{R}^{n\times k} \\ \operatorname{rank}(X)=k}} \operatorname{trace}((X^TX)^{-1}(X^TAX)).$$

4.2 Applications of Eigenvalues and Eigenvectors

Many problems in physics are approximated by an discretized eigenvalue problem in a finite dimensional space. This means that the solution approximation is obtained by finding some subset of eigenvalues and corresponding eigenvectors of a symmetric matrix. In this subsection several examples will be presented, which illustrate application of eigenvalues and eigenvectors in other fields of science.

4.2.1 Propagation of Electro-Magnetic Waves (Helmholtz Equation)

In this example we will try to determine propagating modes in an optical structure, which are invariant with respect to one spatial direction. Under certain model simplifications, the problem will be transformed into an eigenvalue problem of a scalar Helmholtz equation, and then it will be discretized by means of finite elements into a matrix eigenvalue problem (see [30]).

The propagation of light waves in optic components is described through the Maxwell equations for nonmagnetic and free of charge media. If we assume that electric and magnetic fields have harmonic time dependency, then the time stationary *Maxwell equations* are obtained, with the following form

$$\begin{aligned} \nabla \times \vec{E} &= -i\omega\mu\vec{H} \\ \nabla \times \vec{H} &= (i\omega\epsilon + \sigma)\vec{E} \\ \nabla \cdot \epsilon\vec{E} &= 0 \\ \nabla \cdot \vec{H} &= 0 \end{aligned} \tag{4.2}$$

where $\vec{E}$ denotes an electric and $\vec{H}$ a magnetic vector field in a specific position, and parameter ω is the angular frequency. The dielectric number ϵ and the conductivity σ of the medium depend on the spatial position and angular frequency, and the permeability μ is the spatial constant. The elimination of the electric field from (4.2) will produce a stationary vector wave equation for the magnetic field

$$-\Delta\vec{H} - \omega^2\tilde{\epsilon}\mu\vec{H} = \nabla \log\tilde{\epsilon} \times \nabla \times \vec{H}, \tag{4.3}$$

where $\tilde{\epsilon} = \epsilon - i\omega^{-1}\sigma$ is the complex dielectric number. The electric field $\vec{E}$ can be computed from the second Maxwell equation in (4.2).

Since the dielectric number depends very weakly on the position in the observed optical structure, the right side of equation (4.3) can be neglected. Then, the vector *Helmholtz equation* is obtained

$$-\Delta\vec{H} - \omega^2\tilde{\epsilon}\mu\vec{H} = 0, \tag{4.4}$$

which is the basis for the simulation of the most important optical components. The equation (4.4) holds for each component of the magnetic field. So, there is no difference among them and it is sufficient to observe only the scalar Helmholtz equation

$$-\Delta H - \omega^2\tilde{\epsilon}\mu H = 0. \tag{4.5}$$

Further characteristic of the observed optical components is their invariance in one spatial direction. This means that the geometry of the structure changes very little or not at all in one direction (see Figure 4.1). Next, we will choose a coordinate system, so that this particular direction coincides with the z-coordinate direction. Then the dielectric number $\tilde{\epsilon}$ depends only on coordinates x and y, and angular frequency ω. To

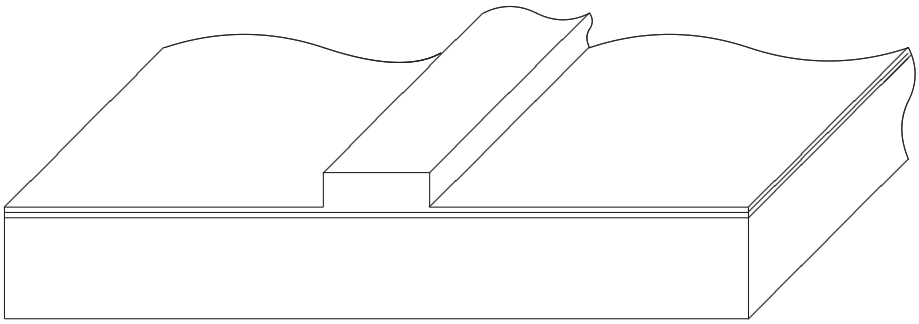


Figure 4.1: A typical structure in optics is a ribbed wave conductor.

determine propagating modes, the magnetic field component H will take the following appropriate form

$$H(x, y, z) = Au(x, y)e^{-ikz}, \tag{4.6}$$

where the real part of k denotes phase velocity, and the imaginary part of k gives information about damping and amplification in the propagation direction. If we put (4.6) in the equation (4.5) we will obtain an eigenvalue problem of the scalar Helmholtz equation

$$-\Delta u - \omega^2 \tilde{\epsilon}\mu u = -k^2 u. \tag{4.7}$$

The problem (4.7) is usually solved on a bounded two-dimensional subset Ω, with an appropriate boundary condition. We will take the Dirichlet boundary condition $u = 0$ on $\partial\Omega$. All together we obtained the following problem: to find functions $u \neq 0$ and numbers λ, which are solutions of the eigenvalue problem

$$\begin{aligned} -\Delta u(x, y) - f(x, y)u(x, y) &= \lambda u(x, y), && (x, y) \in \Omega \\ u(x, y) &= 0, && (x, y) \in \partial\Omega, \end{aligned} \tag{4.8}$$

where $f(x, y) = \omega^2 \tilde{\epsilon}(x, y, \omega)\mu$ and $\lambda = -k^2$. Since the dielectric number $\tilde{\epsilon}$ is complex, the function f accepts complex values, too, so in general the eigenfunctions u and the eigenvalues λ will also be complex.

For the scalar product in $L^2(\Omega)$

$$\langle v, u\rangle = \langle v, u\rangle_{L^2(\Omega)} = \int_\Omega \overline{v(x, y)}u(x, y)d(x, y),$$

we can introduce an equivalent variational formulation of the equation (4.8)

$$\langle \nabla w, \nabla u\rangle - \langle w, fu\rangle = \lambda\langle w, u\rangle, \qquad \forall w \in C_0^\infty(\Omega). \tag{4.9}$$

Two times differentiable functions u and numbers λ, which are solutions of (4.8), are also solutions of (4.9), and vice versa. Now we introduce a sesquilinear form

$$a(w,u) = \langle \nabla w, \nabla u \rangle - \langle w, fu \rangle, \tag{4.10}$$

so that the variational formulation of the eigenvalue problem reads as follows: $u \in H_0^1(\Omega) \setminus \{0\}$ and $\lambda \in \mathbb{C}$ are sought, such that

$$a(w,u) = \lambda \langle w, u \rangle, \qquad \forall w \in H_0^1(\Omega), \tag{4.11}$$

because $C_0^\infty(\Omega) \subset H_0^1(\Omega)$ is dense in $H_0^1(\Omega)$. The solution of the variational problem is called the *weak solution*, u is the right eigenfunction and λ is the eigenvalue.

The variational problem (4.11) can be associated with an adjoint problem: for $\lambda \in \mathbb{C}$, $v \in H_0^1(\Omega) \setminus \{0\}$ is sought, such that

$$a(v,w) = \lambda \langle v, w \rangle, \qquad \forall w \in H_0^1(\Omega), \tag{4.12}$$

where v is the left eigenfunction. This will lead to equivalent formulation of the adjoint problem: for $\lambda \in \mathbb{C}$, $v \in H_0^1(\Omega) \setminus \{0\}$ is sought, such that

$$a^*(w,v) = \overline{\lambda} \langle w, v \rangle, \qquad \forall w \in H_0^1(\Omega), \tag{4.13}$$

where by

$$a^*(w,v) = \overline{a(v,w)} = \langle \nabla w, \nabla v \rangle - \langle w, \overline{f} v \rangle$$

an adjoint sesquilinear form is defined. In case when $\overline{f} = f$ and the function f admits only real values on Ω, the eigenvalue problem is selfadjoint, which implies

$$a^*(w,v) = a(w,v), \qquad \forall v, w \in H_0^1(\Omega).$$

For this case the following theorem can be proven.

Theorem 4.2.1. *Let $\Omega \subset \mathbb{R}^2$ be bounded and let the function $f : \Omega \to \mathbb{R}$ be an element of function space $L^\infty(\Omega)$. Then there exists an orthonormal basis $\{u_j\}_{j=1}^\infty$ of $L^2(\Omega)$, such that the basis functions $u_j \in H_0^1(\Omega)$ satisfy the following relation*

$$a(w,u_j) = \lambda_j \langle w, u_j \rangle, \qquad \forall w \in H_0^1(\Omega), \tag{4.14}$$

for all $j \in \mathbb{N}$. All eigenfunctions u_j are real, and all eigenvalues λ_j are also real and ordered as $\lambda_1 \leq \lambda_2 \leq \cdots \to \infty$.

From now on we will assume that $a(w,v)$ is selfadjoint.

In optics, only a few smallest eigenvalues are of interest, and they are not computed exactly. The variational problem is used to formulate an approximate problem by means of finite elements. The domain Ω is divided into finite number of triangles t_j, as it is done for the ribbed wave conductor in Figure 4.2. The set of all triangles $t = \{t_1, \ldots, t_{N_t}\}$ is called *triangulation* of Ω. The triangles are disjoint and $\overline{\Omega} = \bigcup_{j=1}^{N_t} \overline{t}_j$. The parameter h denotes the maximal side length of all the triangles from t. A point from $\overline{\Omega}$ is called a *node* if it is a corner of a triangle from t. The set off all nodes $p = \{p_1, \ldots, p_{N_p}\}$

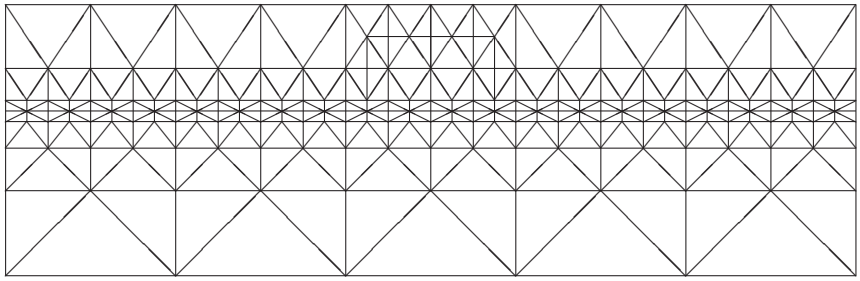


Figure 4.2: Triangulation of the domain Ω.

where $p_i = (x_i, y_i)$, includes N inner nodes which are not in $\partial\Omega$. A function $w_i(x, y)$ is associated with each inner node p_i, such that it is linear on each triangle, continuous on the whole $\overline{\Omega}$ and satisfies

$$w_i(x_j, y_j) = \delta_{ij}, \qquad j = 1, \ldots, N_p.$$

The support of $w_i(x, y)$ consists of all the triangles which have the point p_i as a corner. The space $W_N = \text{span}\{w_i\}_{i=1}^N =: W_h$ is a N-dimensional linear subspace of $H_0^1(\Omega)$ and is called the *finite element space*. It consists of piecewise linear functions, which are continuous on $\overline{\Omega}$ and disappear outside Ω. Each $u_h \in W_h$ has its unique representation as

$$u_h = \sum_{i=1}^{N} u_h^{(i)} w_i,$$

where $u_h^{(i)} = u_h(x_i, y_i)$.

Suppose we want to find approximations $\lambda_{j,h} \in \mathbb{R}$ to eigenvalues λ_j and $u_{j,h} \in W_h \subset H_0^1(\Omega)$ to eigenvectors u_j, for $j = 1, \ldots, q$. Then we will consider functions $u_{j,h} \neq 0$ and numbers $\lambda_{j,h}$ which are solutions of the discrete problem

$$a(w_h, u_{j,h}) = \lambda_{j,h} \langle w_h, u_{j,h} \rangle, \qquad \forall w_h \in W_h. \tag{4.15}$$

Further, we define

$$u_{j,h} = \sum_{i=1}^{N} u_{j,h}^{(i)} w_i = \sum_{i=1}^{N} u_{ij,h} w_i, \qquad j = 1, \ldots, q,$$

and obtain equations that are equivalent to (4.15)

$$\sum_{i=1}^{N} a(w_k, w_i) u_{ij,h} = \lambda_{j,h} \sum_{i=1}^{N} \langle w_k, w_i \rangle u_{ij,h}, \qquad k = 1, \ldots, N. \tag{4.16}$$

Now we define matrices

$$A_h = [a(w_k, w_i)]_{\substack{k=1,\ldots,N \\ i=1,\ldots,N}}, \qquad U_h = [u_{ij,h}]_{\substack{i=1,\ldots,N \\ j=1,\ldots,q}},$$

$$B_h = [\langle w_k, w_i \rangle]_{\substack{k=1,\ldots,N \\ i=1,\ldots,N}}, \qquad \Lambda_h = \text{diag}(\lambda_{j,h})_{j=1,\ldots,q},$$

so that (4.16) can be written in a compact form as a matrix eigenvalue problem

$$A_h U_h = B_h U_h \Lambda_h. \tag{4.17}$$

The matrices A_h and B_h are sparse, because of the small supports of functions w_i. A_h is Hermitian and is called the *system matrix*. B_h is a Gram matrix of the basis functions w_i and hence it is Hermitian and positive definite. It is called the *mass matrix*. Since (4.17) represents a general eigenvalue problem, we will use the Cholesky factorization of the mass matrix $B_h = R_h^* R_h$, and introduce an $N \times q$ matrix of unknowns $X_h = R_h U_h$. Finally we will obtain an equivalent partial eigenvalue problem

$$C_h X_h = X_h \Lambda_h, \qquad C_h = R_h^{-*} A_h R_h^{-1}. \tag{4.18}$$

Now we see that there is justification for taking $\lambda_{j,h} \in \mathbb{R}$, and X_h to be orthonormal. Then, $U_h = R_h^{-1} X_h$, so that $U_h^* B_h U_h = I$, and $u_{1,h}, \ldots, u_{q,h} \in W_h$ generate an orthonormal vector set, where the scalar product is determined by the matrix B_h.

Remark 4.2.2. *Since the elements of the Gram matrix are obtained as the scalar products $\langle w_k, w_i \rangle$, the Gram matrix is usually represented in a factorized form $B_h = F_h^* F_h$, so that for the partition $F = [\ f_1 \ \cdots \ f_N\ ]$, $\langle w_k, w_i \rangle = f_k^T f_i$. The matrix B_h is then never assembled, and the Cholesky factorization is computed through the QR factorization of F_h, so that $F_h = Q_h R_h$ and $B_h = R_h^* Q_h^* Q_h R_h = R_h^* R_h$.*

4.2.2 Vibration Frequencies of a Mass System with Springs

In stability analysis of mechanical structures, the method of concentrated mass is often used. Behavior of a complex mechanical system is approximated by the behavior of a system that consists only of the concentrated masses connected with elastic springs. The problem is to find free oscillation of this system.

Assume that we are observing a system with masses and springs, as described in Figure 4.3.

Here m_i denotes the i-th mass, and k_j denotes the stiffness of the j-th spring. Further we define the following matrices:

$$M = \begin{bmatrix} m_1 & 0 & 0 & 0 \\ 0 & m_2 & 0 & 0 \\ 0 & 0 & m_3 & 0 \\ 0 & 0 & 0 & m_4 \end{bmatrix},$$

$$K = \begin{bmatrix} k_1 + k_2 + k_6 & -k_2 & -k_6 & 0 \\ -k_2 & k_2 + k_3 + k_8 & -k_3 & -k_8 \\ -k_6 & -k_3 & k_3 + k_4 + k_6 + k_7 & -k_4 \\ 0 & -k_8 & -k_4 & k_4 + k_5 + k_8 \end{bmatrix},$$

where the matrix K represents the interaction between masses. The i-th row corresponds to the i-th mass, and its j-th column corresponds to the relation between the i-th and the j-th mass. On the diagonal, in the position (i, i), there is a sum of the stiffnesses of all

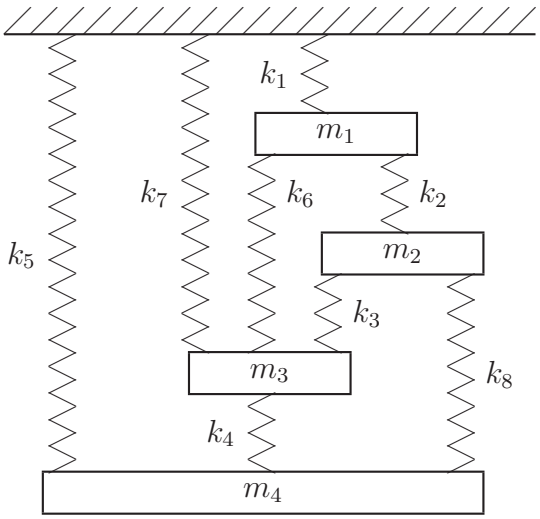


Figure 4.3: System of masses that are connected with springs.

springs that are connected to the i-th mass. The number $-k_\ell$ is placed in the position (i, j) if the ℓ-th spring connects the i-th and the j-th mass, otherwise $K(i, j) = 0$. Finally, we define a vector

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix},$$

where x_i represents the vertical shift of the i-th mass from the steady state. From the law of momentum conservation, the mass position can be described by a system of differential equations

$$\ddot{x} = -M^{-1}Kx. \tag{4.19}$$

If we assume that the solution of the system has the following form

$$x = x_0 e^{i\phi t},$$

which is the standard procedure when solving a system of linear differential equations, then from (4.19) we obtain

$$\ddot{x} = -\phi^2 x_0 e^{i\phi t} = -M^{-1}Kx_0 e^{i\phi t}.$$

By elimination of the term $e^{i\phi t}$ we reduce our problem to the eigenvalue problem for the matrix $M^{-1}K$:

$$M^{-1}Kx_0 = \phi^2 x_0.$$

In this case the eigenvalue is equal to ϕ^2, which presents the square of the oscillation frequency. We can also note that the matrix K is symmetric, and that $M^{-1}K$ is not. This can be improved by noting that the matrix M is diagonal with positive diagonal

elements, so we can define $M^{\frac{1}{2}} = \text{diag}(m_1^{\frac{1}{2}}, \ldots, m_4^{\frac{1}{2}})$. Premultiplication of the matrix $M^{-1}K$ with $M^{\frac{1}{2}}$ and postmultiplication with $M^{-\frac{1}{2}}$, produce a matrix similar to $M^{-1}K$ which is symmetric and of the form

$$A = M^{\frac{1}{2}}(M^{-1}K)M^{-\frac{1}{2}} = M^{-\frac{1}{2}}KM^{-\frac{1}{2}}. \tag{4.20}$$

So, if λ and u are the eigenvalue and the corresponding eigenvector of the matrix A from (4.20), then the solution of the problem (4.19) is equal to

$$x = M^{-\frac{1}{2}}ue^{i\sqrt{\lambda}t}.$$

4.2.3 Graph Partitioning

Let us consider the following problem: we want to divide a set of objects into groups which contain objects with similar properties. First, the problem of partitioning the set of objects will be replaced by a problem of partitioning a set of graph vertices. This will be done under the condition that weights of edges which connect vertices from different groups are minimized. Second, the graph partitioning problem will be reduced to an eigenvalue problem.

So, we will start with some definitions and results taken from [74], which are necessary for stating the graph problem.

Definition 4.2.3. *The* graph *is an ordered pair $G = (V, E)$, where $\emptyset \neq V = V(G)$ is the set of vertices, $E = E(G)$ is the set of edges disjoint with V, and each edge $e \in E$ connects two vertices $u, v \in V$ which we call the ends of e. The vertices u and v are then incident, and we can write $e = \{u, v\}$.*

- *The graph G is* finite *if the sets V and E are finite.*
- *An edge whose ends coincide is called a* loop.
- *Two or more edges with the same pair of ends are called* multiple edges.
- *The graph is* simple *if it contains no loops and no multiple edges.*
- *A simple graph in which each pair of vertices is connected by an edge is called a* complete *graph.*
- *Let w be a function $w : E(G) \to F$, where F can be $\mathbb{R}$, $\mathbb{R}^+$, $\mathbb{Z}_m$,.... The ordered pair (G, w) consisting of the graph G and the weight function w is called the* weighted *graph.*

Let a simple finite weighted graph (G, w) be given, where $G = (V, E)$, $\emptyset \neq V = \{1, 2, \ldots, n\}$ is the set of vertices and E is the set of edges $\{i, j\}$ $i, j \in V$, with weights $w(\{i, j\}) \in \mathbb{R}^+$. We want to divide V into two subsets V_1 and V_2, and we call this procedure *bipartitioning*. The bipartition of the set V can be described by the relation

$$V = V_1 \cup V_2, \qquad V_1 \cap V_2 = \emptyset.$$

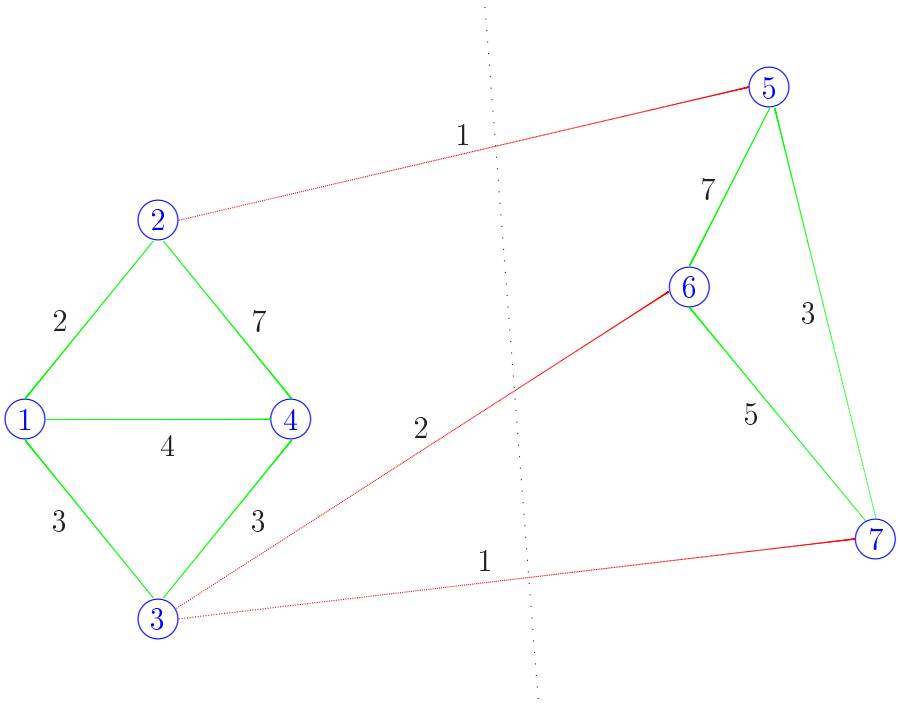


Figure 4.4: The bipartition of the graph G.

The question now arises, how to produce a meaningful bipartition so that V_1 and V_2 present groups with some common property?

First, we will need the following definition.

Definition 4.2.4. *The* adjacency matrix of the graph G *is the* $n \times n$ *matrix* $W = [w_{ij}]$, *where*

$$w_{ij} = \begin{cases} w(\{i,j\}), & \text{if } \{i,j\} \in E, \\ 0, & \text{otherwise.} \end{cases}$$

The matrix W is a symmetric matrix whose elements are nonnegative real numbers. Since the graph is simple, diagonal elements of W are equal to 0.

So, let us partition the set V into two subsets V_1 and V_2. A dissimilarity between these two subspaces can be computed as the total weight of all edges which connect sets V_1 and V_2. It is called the *cut* of the partition, and it is defined by

$$\text{cut}(V_1, V_2) = \sum_{i \in V_1, j \in V_2} w_{ij} \quad \text{for } V_1, V_2 \subset V.$$

Next we will generalize the meaning of the weight function: let

$$w(i) = \sum_{j=1}^{n} w_{ij},$$

be a sum of weights of edges incident with the vertex i, then $w(i)$ is called the *weight of the vertex* $i \in V$, and let

$$w(V_k) = \sum_{i \in V_k} w(i) = \mathrm{cut}(V_k, V) = \sum_{i \in V_k, j \in V} w_{ij}$$

be a sum of weights of all the vertices from V_k, then $w(V_k)$ is called the *weight of the set* $V_k \subseteq V$.

For example, if the set of vertices $V = \{1, 2, 3, 4, 5, 6, 7\}$ of the graph G from Figure 4.4 is partitioned into the subsets $V_1 = \{1, 2, 3, 4\}$ and $V_2 = \{5, 6, 7\}$, then

$$\begin{aligned} \mathrm{cut}(V_1, V_2) &= 4 \\ w(V_1) &= w(1) + w(2) + w(3) + w(4) = 42 \\ w(V_2) &= w(5) + w(6) + w(7) = 34 \end{aligned}$$

The most simple way to partition a graph is by minimization of the cut, and there exist efficient algorithms which find the partition with the minimal cut. But, in such partitioning blocks with small number of vertices are very often isolated, and sometimes this is not satisfactory.

We need some additional conditions on the graph partition to avoid such small blocks. If we want our subsets to have balanced weights, we have to minimize another objective function

$$\mathrm{cut}_N(V_1, V_2) = \frac{\mathrm{cut}(V_1, V_2)}{w(V_1)} + \frac{\mathrm{cut}(V_2, V_1)}{w(V_2)},$$

which is called a *normalized cut*. For the bipartition in Figure 4.4 we have

$$\mathrm{cut}_N(V_1, V_2) = \frac{4}{42} + \frac{4}{32} = 0.2202.$$

Unfortunately, the problem of finding the exact minimal normalized cut belongs to the NP class. This means that this problem is most likely not solvable by a deterministic algorithm in polynomial time. The execution time of an algorithm that solves the problem grows exponentially with the size of input set.

Nevertheless, we will show that this problem can be solved approximately by placing the problem in a real domain.

We start again with the set of vertices $V = \{1, 2, \ldots, n\}$. The partition $V = V_1 \cup V_2$ can be represented by the vector $x = [x_i]$ defined as

$$x_i = \begin{cases} 1, & i \in V_1 \\ -1, & i \in V_2 \end{cases}, \qquad i = 1, \ldots, n.$$

It can be easily shown that for the matrix $D = \mathrm{diag}(w(1), \ldots, w(n))$ the following holds

$$\begin{aligned} \mathrm{cut}(V_1, V_2) &= \frac{1}{4} x^T (D - W) x, , \\ \mathrm{cut}_N(V_1, V_2) &= \frac{z^T (D - W) z}{z^T D z}, \end{aligned}$$

where

$$z_i = \begin{cases} 1, & i \in V_1 \\ -q, & i \in V_2 \end{cases}, \qquad i = 1, \ldots, n, \quad q = \frac{w(V_1)}{w(V_2)}.$$

Matrix $L = D - W$ is called the *Laplace matrix of the graph* G and has many useful properties. The matrix $L = [\ell_{ij}]$ is a $n \times n$ matrix whose each row and column corresponds to one vertex, so that

$$\ell_{ij} = \begin{cases} \sum_{k=1}^{n} w_{ik}, & i = j \\ -w_{ij}, & i \neq j,\ \{i,j\} \in E \\ 0, & \text{otherwise.} \end{cases}$$

L is a symmetric positive semidefinite matrix, so all its eigenvalues are real and nonnegative. Further

$$Le = 0, \qquad \text{for} \qquad e = [\ 1 \quad \ldots \quad 1\ ]^T,$$

which implies that 0 is the smallest eigenvalue of L, and e is the corresponding eigenvector.

Now we can reformulate our problem. We started with two discrete minimization problems

$$\min_{\substack{V_1 \cup V_2 = V \\ V_1 \cap V_2 = \emptyset}} \operatorname{cut}(V_1, V_2) = \min_{\substack{x_i \in \{-1,1\} \\ x^T e = 0}} x^T L x \tag{4.21}$$

$$\min_{\substack{V_1 \cup V_2 = V \\ V_1 \cap V_2 = \emptyset}} \operatorname{cut}_N(V_1, V_2) = \min_{\substack{z_i \in \{-q,1\} \\ z^T De = 0}} \frac{z^T L z}{z^T D z}, \tag{4.22}$$

where in case when n is even, the condition $x^T e = 0$ means that $\sum_{i=1}^{n} x_i = 0$ and that subspaces V_1 and V_2 contain the same number of vertices. This is done to avoid a trivial solution and to balance the number of vertices in subspaces. The condition $z^T De = 0$ means that

$$0 = \sum_{i=1}^{n} z_i w(i) = \sum_{i \in V_1} w(i) - q \sum_{i \in V_2} w(i) = w(V_1) - q w(V_2),$$

which gives the definition of the number q.

The discrete problems (4.21) and (4.22) will be now replaced by continuous minimization problems

$$\min_{\substack{\|x\|_2 = 1 \\ x^T e = 1}} x^T L x \tag{4.23}$$

$$\min_{z^T De = 0} \frac{z^T L z}{z^T D z} = \min_{\substack{\|y\|_2 = 1 \\ y^T D^{\frac{1}{2}} e = 0}} y D^{-\frac{1}{2}} L D^{-\frac{1}{2}} y \tag{4.24}$$

where $y = D^{\frac{1}{2}} z$. The matrix $L_N = D^{-\frac{1}{2}} L D^{-\frac{1}{2}}$ is called the *normalized Laplace matrix of the graph* G, and is also symmetric positive semidefinite, with the smallest eigenvalue equal to 0 and with the corresponding eigenvalue equal to $D^{\frac{1}{2}} e$.

By the consequence of Theorem 4.1.7, the solution of problem (4.23) is equal to the minimal eigenvalue of the matrix which is a compression of L to $\mathcal{U}_2$, where $\mathcal{U}_2$ is an $(n-1)$-dimensional subspace spanned by all eigenvectors of L except e. That means that the minimum is obtained for $u_2 = [u_i^{(2)}]$, the second smallest eigenvalue of L. This vector is called the *Fiedler vector*.

Equivalently, the solution of the problem (4.23) is equal to $u_{N,2} = [u_i^{(N,2)}]$, which is the second smallest eigenvalue of L_N. This vector is called the *normalized Fiedler vector*.

So, as an approximative optimal bipartition we can take

for minimization of cut

$$V_1 = \{i : u_i^{(2)} \geq 0\}, \qquad V_2 = \{i : u_i^{(2)} < 0\}$$

for minimization of cut_N

$$V_1 = \{i : D^{-\frac{1}{2}} u_i^{(N,2)} \geq 0\}, \qquad V_2 = \{i : D^{-\frac{1}{2}} u_i^{(N,2)} < 0\}.$$

4.3 Perturbation Theory

This section is similar to section 2.3. All the perturbation results for the symmetric eigenvalue problem can be generalized to a singular value problem of generalized matrices. That means that the results in section 2.3 are derived from the results described in this section. So, when the spectral decomposition of a symmetric matrix $A \in \mathbb{R}^{n\times n}$ is computed in finite precision arithmetic, instead of exact factors $U \in \mathbb{R}^{n\times n}$ and $\Lambda \in \mathbb{R}^{n\times n}$, matrices $\tilde{U}$ and $\tilde{\Lambda}$ will be computed. Numerical analysis of the method used for computing eigenvalues and eigenvectors results with a matrix $\tilde{A}$, such that computed eigenvectors in $\tilde{U}$ and computed eigenvalues in $\tilde{\Lambda}$ are exact for that matrix. The relation between exact and computed factors is then given by perturbation theory, which compares the matrices A and $\tilde{A}$. Basically, the perturbation theory will produce bounds on the errors in computed eigenvalues and eigenvectors. The error bounds can be divided in two different categories:

1. eigenvalue error bounds

2. eigenspace error bounds

4.3.1 Eigenvalue Error Bounds

First, let us take a look at additive perturbations of a Hermitian matrix.

Theorem 4.3.1 (Mirsky–Lidskii-Wielandt [66, p. 21]). *Let $A \in \mathbb{C}^{n\times n}$ be a Hermitian matrix, and suppose that $\tilde{A} \in \mathbb{C}^{n\times n}$ is also Hermitian, and that their eigenvalues are ordered as in (4.1). Then, for any unitarily invariant norm $\|\cdot\|$, we have*

$$\|\Lambda - \tilde{\Lambda}\| \leq \|A - \tilde{A}\|.$$

Specially, for $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$ *and* $\tilde{\Lambda} = \mathrm{diag}(\tilde{\lambda}_1, \ldots, \tilde{\lambda}_n)$*, we have*

$$\max_{i=1,\ldots,n} |\lambda_i - \tilde{\lambda}_i| \leq \|A - \tilde{A}\|_2,$$
$$\sqrt{\sum_{i=1}^{n} (\lambda_i - \tilde{\lambda}_i)^2} \leq \|A - \tilde{A}\|_F.$$

Theorem 4.3.1 claims that the norm of backward absolute error is the upper bound for absolute error in eigenvalues. But, the absolute errors are not always the best way of measuring errors as we saw in section 2.3. The next step would be to look at relative errors in eigenvalues.

Theorem 4.3.2 ([52, p. 161]). *Let* A, $\tilde{A} \in \mathbb{C}^{n\times n}$ *be Hermitian, and let* A *also be positive definite, then*

$$\max_{i=1,\ldots,n} \frac{|\lambda_i - \tilde{\lambda}_i|}{|\lambda_i|} \leq \|A^{-\frac{1}{2}}(\tilde{A} - A)A^{-\frac{1}{2}}\|_2.$$

Multiplicative perturbations are much more suitable for this case, so the next results will deal with such perturbations. Let us start again with the Ostrowsky-type bounds.

Theorem 4.3.3 ([52, p. 187]). *Let* $A \in \mathbb{C}^{n\times n}$ *be Hermitian, and let* $\tilde{A} = DAD^*$ *also be Hermitian, where* D *is nonsingular. Then*

$$\frac{|\lambda_i|}{\|(D^*D)^{-1}\|_2} \leq |\tilde{\lambda}_i| \leq |\lambda_i| \|D^*D\|_2.$$

Theorem 4.3.4 ([52, p. 188]). *Let* $A \in \mathbb{C}^{n\times n}$ *be Hermitian, and let* $\tilde{A} = DAD^*$ *also be Hermitian, where* D *is nonsingular. Then*

$$\max_{i=1,\ldots,n} \frac{|\lambda_i - \tilde{\lambda}_i|}{|\lambda_i|} \leq \|I - DD^*\|_2.$$

Thus, relative error in singular values of $\tilde{A}$ is small if D is close to a unitary matrix.

We are also going to give a result for the χ relative distance between exact and computed eigenvalues.

Theorem 4.3.5 ([64, p. 395]). *Let* $A \in \mathbb{C}^{n\times n}$ *and* $\tilde{A} = D^*AD$ *be Hermitian matrices, where* D *is nonsingular. Then,*

$$\max_{i=1,\ldots,n} \chi(\lambda_i, \tilde{\lambda}_i) \leq \|D^* - D^{-1}\|_2,$$
$$\sqrt{\sum_{i=1}^{n} \chi^2(\lambda_i, \tilde{\lambda}_i)} \leq \|D^* - D^{-1}\|_F.$$

4.3.2 Eigenspace Error Bounds

We are comparing subspaces again through the angle matrix $\Theta(X, Y)$.

Theorem 4.3.6 (Davis–Kahan [67, p. 4]). *Let $A \in \mathbb{C}^{n\times n}$ and $\tilde{A} \in \mathbb{C}^{n\times n}$ be two Hermitian matrices with the following spectral decompositions*

$$A = U\Lambda U^* = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Lambda_1 & 0 \\ 0 & \Lambda_2 \end{bmatrix} \begin{bmatrix} U_1^* \\ U_2^* \end{bmatrix}, \tag{4.25}$$

$$\tilde{A} = \tilde{U}\tilde{\Lambda}\tilde{U}^* = \begin{bmatrix} \tilde{U}_1 & \tilde{U}_2 \end{bmatrix} \begin{bmatrix} \tilde{\Lambda}_1 & 0 \\ 0 & \tilde{\Lambda}_2 \end{bmatrix} \begin{bmatrix} \tilde{U}_1^* \\ \tilde{U}_2^* \end{bmatrix}, \tag{4.26}$$

where $U, \tilde{U} \in \mathbb{C}^{n\times n}$ are unitary, $U_1, \tilde{U}_1 \in \mathbb{C}^{n\times k}$ and

$$\Lambda_1 = \operatorname{diag}(\lambda_1, \ldots, \lambda_k), \qquad \Lambda_2 = \operatorname{diag}(\lambda_{k+1}, \ldots, \lambda_n), \tag{4.27}$$

$$\tilde{\Lambda}_1 = \operatorname{diag}(\tilde{\lambda}_1, \ldots, \tilde{\lambda}_k), \qquad \tilde{\Lambda}_2 = \operatorname{diag}(\tilde{\lambda}_{k+1}, \ldots, \tilde{\lambda}_n), \tag{4.28}$$

with $1 \le k < n$. Let us define the residual

$$R = \tilde{A}U_1 - U_1\Lambda_1 = (\tilde{A} - A)U_1.$$

If

$$\delta = \min_{i=1,\ldots,k,\ j=1,\ldots,n-k} |\lambda_i - \tilde{\lambda}_{k+j}| > 0,$$

then

$$\|\sin\Theta(U_1, \tilde{U}_1)\|_F \le \frac{\|R\|_F}{\delta}.$$

The scalar δ represents the absolute gap between the eigenvalues of Λ_1 and $\tilde{\Lambda}_2$. The next theorem involves the p relative distance defined in subsection 2.3.2.

Theorem 4.3.7 ([67, p. 6]). *Let $A \in \mathbb{C}^{n\times n}$, and $\tilde{A} = D^*AD$ be two Hermitian matrices with spectral decomposition (4.25), (4.26), (4.27) and (4.28), where D is non-singular. Let $\eta_2 = \min_{i=1,\ldots,k,\ j=1,\ldots,n-k} \rho_2(\lambda_i, \tilde{\lambda}_{k+j}) > 0$, then*

$$\|\sin\Theta(U_1, \tilde{U}_1)\|_F \le \frac{\sqrt{\|(I - D^*)U_1\|_F^2 + \|(I - D^{-1})U_1\|_F^2}}{\eta_2}.$$

4.4 Subspace Methods for the Partial Eigenvalue Problem

The task of solving the partial eigenvalue problem for a symmetric matrix $A \in \mathbb{R}^{n\times n}$ is to find $U \in \mathbb{R}^{n\times k}$, $U^TU = I_k$ for $1 \le k \ll n$ and $\Lambda = \operatorname{diag}(\lambda_{i_1}, \ldots, \lambda_{i_k}) \in \mathbb{R}^{k\times k}$, such that

$$AU = U\Lambda, \tag{4.29}$$

where λ_{i_j}, $j = 1, \ldots, k$ represent some choice of eigenvalues of A. In most cases we will observe a situation when

$$\lambda_{i_j} = \lambda_j, \tag{4.30}$$

where λ_j are ordered as in (4.1). This task is equivalent to finding a specific k dimensional invariant subspace which corresponds to the k smallest eigenvalues.

4.4.1 The Rayleigh–Ritz method

The Rayleigh–Ritz method is a basic method for computing eigenvalue and eigenvector approximations of a symmetric matrix A, from a given subspace $\mathcal{X}$. Most of the iterative subspace methods use this method for obtaining approximations in the current iteration. The Rayleigh–Ritz method computes eigenvalues and eigenvectors of A's compression to the subspace $\mathcal{X}$, and it turns out that they have some optimal properties.

Let the subspace $\mathcal{X}$ be represented by an orthonormal basis $X = [x_1, \ldots, x_k]$. Then, the algorithm runs as it is described in [78, Chapter 11].

Algorithm 4.4.1 (The Rayleigh–Ritz method). *For a symmetric matrix $A \in \mathbb{R}^{n\times n}$, and for a given orthonormal matrix $X = [x_1, \ldots, x_m]$ this algorithm computes an orthonormal matrix $Y = [y_1, \ldots, y_k]$, where $k \leq m$, such that $y_1, \ldots, y_k$ represent good approximations of k eigenvectors of A.*

$H = X^T AX$;
Compute k eigenpairs of H which are of interest: $Hs_i = \theta_i s_i$ $i = 1, \ldots, k$;
for $i = 1 : k$
  $y_i = Xs_i$;
end
$Y = [\ y_1 \ \ \ldots \ \ y_k\ ]$;
for $i = 1, \ldots, k$
  $r_i = Ay_i - \theta_i y_i$;
  Each interval $[\theta_i - \|r_i\|_2, \theta_i + \|r_i\|_2]$ contains an eigenvalue of A;
end

The scalars θ_i are called the *Ritz values*, and the vectors y_i are called the *Ritz vectors*. The full set $\{(\theta_i, y_i), i = 1, \ldots, m\}$ is the best set of the eigenpair approximations of A which can be derived from the m-dimensional subspace $\mathcal{X}$. The following theorems describe the optimality of the Ritz values and the Ritz vectors.

Theorem 4.4.2 ([78, p. 215]). *Let $A \in \mathbb{R}^{n\times n}$ be symmetric, with eigenvalues ordered as in (4.1), and let $\mathcal{X}$ be a k-dimensional subspace. Then*

$$\theta_i = \min_{\substack{\mathcal{S}\subset\mathcal{X} \\ \dim(\mathcal{S})=i}} \max_{\substack{x\in\mathcal{S} \\ \|x\|_2=1}} x^T Ax, \quad i = 1, \ldots, k,$$

where $\theta_1 \leq \theta_2 \leq \cdots \leq \theta_k$.

Theorem 4.4.3 ([78, p. 216]). *Let $A \in \mathbb{R}^{n\times n}$ be symmetric, with eigenvalues ordered as in (4.1), and let $\mathcal{X}$ be a k-dimensional subspace represented by an orthonormal basis X. Then for $H = X^T AX$*

$$\|AX - XH\|_2 \leq \|AX - XG\|_2, \qquad \forall G \in \mathbb{R}^{k\times k}.$$

Theorem 4.4.4 ([78, p. 219]). *Let X be any orthonormal $n \times k$ matrix. Associated with it are $H = X^T AX$ and $R = AX - XH$. There are k of A's eigenvalues $\{\lambda_{j_i}, i = 1, \ldots, k\}$ which can be put in one–one correspondence with the eigenvalues θ_i of H in such a way that*

$$|\theta_i - \lambda_{j_i}| \leq \|R\|_2, \qquad i = 1, \ldots, k.$$

A tighter bound is stated in the next theorem.

Theorem 4.4.5 ([70, p. 3]). *Let $A \in \mathbb{C}^{n\times n}$ be Hermitian and let A have eigenvalues $\lambda_1 \leq \cdots \leq \lambda_n$. Let $\mathcal{X}$ be a k-dimensional subspace of $\mathbb{C}^n$ given as the range of orthonormal $X \in \mathbb{C}^{n\times k}$. Further, let $M = X^*AX \in \mathbb{C}^{k\times k}$ be the Rayleigh quotient, and let the residual be defined as*

$$R = AX - XM = (I - XX^*)AX$$

We define additive perturbation $\delta A = RX^* + XR^*$, and change A to $\tilde{A} = A - \delta A$. Then the eigenvalues $\tilde{\lambda}_1 \leq \cdots \leq \tilde{\lambda}_n$ of $\tilde{A}$ satisfy

$$\max_{i=1,\ldots,n} |\lambda_i - \tilde{\lambda}_i| \leq \frac{\|R\|_2^2}{\min\limits_{j,k} |\mu_j - \nu_k|},$$

where μ_j, are the eigenvalues of M, ν_k are the eigenvalues of $N = X_\perp^* A X_\perp$, and $X_\perp$ is the orthonormal basis of $\mathcal{X}^\perp$. Further, $\lambda(\tilde{A}) = \lambda(M) \cup \lambda(N)$.

4.4.2 Simple Subspace Iteration

Subspace iteration is a straightforward generalization of both the power method and the inverse iteration [78, Chapter 14]. Let $A \in \mathbb{R}^{n\times n}$, and let $\mathcal{X} \subset \mathbb{R}^n$ be a k-dimensional subspace. Then we define a new subspace

$$A\mathcal{X} = \{Ax : x \in \mathcal{X}\}.$$

The next step is the definition of the **block Krylov subspace**

$$\mathcal{K}_\ell(A, \mathcal{X}) = \operatorname{span}\{\mathcal{X}, A\mathcal{X}, A^2\mathcal{X}, \ldots, A^{\ell-1}\mathcal{X}\},$$

which plays an important role in subspace iterations. In practice, $\mathcal{X}$ is represented by an orthonormal basis $X = [x_1 \ \ldots x_k]$. Let u_s be the eigenvector belonging to the eigenvalue λ_s with the largest absolute value. Since the power method converges to u_s for every starting vector not orthogonal to u_s if $|\lambda_s| > |\lambda_i|$ for $i \neq s$, the basis of A^jX should be orthonormalized in each step of the computation of Krylov subspace. Without the reorthogonalization all of the columns of the matrix A^jX would converge to the same vector. This way we will obtain a basis for the leading k-dimensional invariant subspace, consisting of k eigenvectors that belong to k eigenvalues with the largest absolute values.

Algorithm 4.4.6 (Simple subspace iteration). *For $A \in \mathbb{R}^{n\times n}$, and for a given orthonormal matrix $X_0 = [x_1^{(0)}, \ldots, x_k^{(0)}]$ this algorithm computes an orthonormal matrix $X = [x_1, \ldots, x_k]$, such that $\mathcal{X} = \operatorname{span}\{X\}$ represents a good approximation of $\operatorname{span}\{u_{n-k+1}, \ldots, u_n\}$.*

for $j = 1, 2, \ldots$
  $Y_j = AX_{j-1}$;
  Compute QR factorization $Y_j = X_jR_j$;
  Test X_j for convergence. If the convergence condition is satisfied then stop.
end
$X = X_j$

Another possibility is to use an inverse of the matrix $A - \lambda I$, i.e. to solve systems with the matrix $A - \lambda I$. Then, subspace iteration can be modified to use $(A - \lambda I)^{-1}$ for obtaining k eigenvalues closest to λ, together with their eigenvectors. This is the most common way in which this technique is used.

Algorithm 4.4.7 (Inverse subspace iteration). *For $A \in \mathbb{R}^{n\times n}$, for a given orthonormal matrix $X_0 = [x_1^{(0)}, \ldots, x_k^{(0)}]$, and for $\lambda \in \mathbb{R}$ this algorithm computes an orthonormal matrix $X = [x_1, \ldots, x_k]$, such that $\mathcal{X} = \operatorname{span}\{X\}$ represents a good approximation of $\operatorname{span}\{u_{i_1}, \ldots, u_{i_k}\}$, where $u_{i_1}, \ldots, u_{i_k}$ are eigenvectors belonging to k eigenvalues closest to λ.*

for $j = 1, 2, \ldots$
  Solve $X_{j-1} = (A - \lambda I)Y_j$;
  Compute QR factorization $Y_j = X_j R_j$;
  Test X_j for convergence. If the convergence condition is satisfied then stop.
end
$X = X_j$

For $\lambda = 0$ inverse subspace iteration will converge to $\operatorname{span}\{u_1, \ldots, u_k\}$. The following theorem describes this convergence.

Theorem 4.4.8 ([78, p. 297]). *Let $A \in \mathbb{R}^{n\times n}$ be a symmetric matrix with $0 < \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_k < \lambda_{k+1} \leq \cdots \leq \lambda_n$, and let $U(:, 1:k)^T X_0$ be invertible. Then*

$$\tan \angle(u_i, A^{-j}\mathcal{X}_0) \leq \left(\frac{\lambda_i}{\lambda_{k+1}}\right)^j \tan \angle(\mathcal{U}, \mathcal{X}_0),$$

where

$$\mathcal{U} = \operatorname{span}\{u_1, \ldots, u_k\}, \qquad \mathcal{X}_0 = \operatorname{span}\{X_0\}.$$

From Theorem 4.4.8 it follows that inverse subspace iteration might converge very slowly if λ_{k+1} is close to λ_k.

4.4.3 The Block Rayleigh Quotient Iteration

The Rayleigh quotient iteration (RQI) is a very well known method for computing an eigenpair of a symmetric matrix [78]. For a matrix $A \in \mathbb{R}^{n\times n}$, the **Rayleigh quotient** is defined as

$$\rho(x) = \frac{x^T A x}{x^T x}, \qquad \text{where } x \in \mathbb{R}^n, x \neq 0,$$

or in matrix form

$$H(X) = (X^T X)^{-1} X^T A X, \qquad \text{where } X \in \mathbb{R}^{n\times k}, k < n, \ X \text{ has full column rank.}$$

There were several attempts to generalize RQI into a subspace method, but the most convenient is the Block Rayleigh Quotient Iteration (BRQI) described in [28].

Algorithm 4.4.9 (The Block Rayleigh Quotient Iteration). *For a symmetric matrix $A \in \mathbb{R}^{n\times n}$, and for a given orthonormal matrix $X_0 = [x_1^{(0)}, \ldots, x_k^{(0)}]$ this algorithm computes an orthonormal matrix $X = [x_1, \ldots, x_k]$, such that $\mathcal{X} = \mathrm{span}\{X\}$ represents a good approximation of $\mathrm{span}\{u_{i_1}, \ldots, u_{i_k}\}$, where $1 \le i_1 < i_2 < \cdots < i_k \le n$.*

for $j = 1, 2, \ldots$
  Compute the Ritz values $\theta_i^{(j-1)}$ and the Ritz vectors $y_i^{(j-1)}$, $i = 1, \ldots, k$
  for A and X_{j-1}, so that:
    $X_{j-1}^T A X_{j-1} = Z_{j-1}\Theta_{j-1}Z_{j-1}^T, \quad \Theta_{j-1} = \mathrm{diag}(\theta_1^{(j-1)}, \ldots, \theta_k^{(j-1)}),$
    $Z_{j-1} \in \mathbb{R}^{k\times k}, \ Z_{j-1}^T Z_{j-1} = I, \quad Y_{j-1} = [y_1^{(j-1)}, \ldots, y_k^{(j-1)}] = X_{j-1}Z_{j-1}.$
  for $i = 1 : k$
    Let m_i and n_i be given integers, $0 \le m_i < i$, $0 \le n_i \le k - i$, define:
      $X_{i,j-1} = [x_{i-m_i}^{(j-1)}, \ldots, x_{i+n_i}^{(j-1)}], \quad Q_{i,j-1} = I - X_{i,j-1}X_{i,j-1}^T;$
    Compute $w_i^{(j-1)}$ such that
      $Q_{i,j-1}(A - \theta_i^{(j-1)}I)(y_i^{(j-1)} + w_i^{(j-1)}) = 0, \quad (w_i^{(j-1)})^T X_{i,j-1} = 0;$
  end
  Compute QR factorization $[y_1^{(j-1)} + w_1^{(j-1)}, \ldots, y_k^{(j-1)} + w_k^{(j-1)}] = X_j R_j$;
  Test X_j for convergence. If the convergence condition is satisfied then stop.
end
$X = X_j$

The parameters m_i and n_i are integers chosen so that $Q_{i,j-1}(A-\theta_i^{(j-1)}I)\Big|_{X_{i,j-1}^{\perp}}$ is well-conditioned. In case when $m_i = n_i = 0$, the vector $y_i^{(j-1)} + w_i^{(j-1)}$ after normalization is equal to $x_i^{(j)}$ updated by a classical RQI iteration. This special case can be written in a more simple form than Algorithm 4.4.9, as follows.

Algorithm 4.4.10 (The Block Rayleigh Quotient Iteration (Classical)). *For a symmetric matrix $A \in \mathbb{R}^{n\times n}$, and for a given orthonormal matrix $X_0 = [x_1^{(0)}, \ldots, x_k^{(0)}]$ this algorithm computes an orthonormal matrix $X = [x_1, \ldots, x_k]$, such that $\mathcal{X} = \mathrm{span}\{X\}$ represents a good approximation of $\mathrm{span}\{u_{i_1}, \ldots, u_{i_k}\}$, where $1 \le i_1 < i_2 < \cdots < i_k \le n$.*

for $j = 1, 2, \ldots$
  Compute the Ritz values $\theta_i^{(j-1)}$ and the Ritz vectors $y_i^{(j-1)}$, $i = 1, \ldots, k$
  for A and X_{j-1}, so that:
    $X_{j-1}^T A X_{j-1} = Z_{j-1}\Theta_{j-1}Z_{j-1}^T, \quad \Theta_{j-1} = \mathrm{diag}(\theta_1^{(j-1)}, \ldots, \theta_k^{(j-1)}),$
    $Z_{j-1} \in \mathbb{R}^{k\times k}, \ Z_{j-1}^T Z_{j-1} = I, \quad Y_{j-1} = [y_1^{(j-1)}, \ldots, y_k^{(j-1)}] = X_{j-1}Z_{j-1}.$
  for $i = 1 : k$
    Solve $(A - \theta_i^{(j-1)}I)w_i^{(j)} = y_i^{(j-1)}$;
  end
  Compute QR factorization $[w_1^{(j)}, \ldots, w_k^{(j)}] = X_j R_j$;
  Test X_j for convergence. If the convergence condition is satisfied then stop.
end
$X = X_j$

BRQI has local quadratic convergence, which is described in the following theorem.

Theorem 4.4.11 ([28, p. 68]). *There exist constants $\epsilon_0 > 0$, ρ, $\chi < 1$, $C_\theta > 0$ such that if $\operatorname{dist}(\mathcal{U}, \mathcal{X}_0) < \epsilon_0$, Algorithm 4.4.9 is well defined and the following properties hold for $j = 0, 1, 2, \ldots$,*

$$\begin{aligned} \dim(\mathcal{X}_j) &= k, \\ \operatorname{dist}(\mathcal{U}, \mathcal{X}_j) &\leq \rho(\operatorname{dist}(\mathcal{U}, \mathcal{X}_{j-1}))^2, \\ \operatorname{dist}(\mathcal{U}, \mathcal{X}_j) &\leq \chi^{\operatorname{dist}(\mathcal{U}, \mathcal{X}_{j-1})}, \\ |\theta_i^{(j)} - \lambda_i| &\leq C_\theta \operatorname{dist}(\mathcal{U}, \mathcal{X}_j), \quad i = 1, \ldots, k, \end{aligned}$$

where

$$\mathcal{U} = \operatorname{span}\{u_1, \ldots, u_k\}, \quad \mathcal{X}_0 = \operatorname{span}\{X_0\}, \quad \mathcal{X}_j = \operatorname{span}\{X_j\},$$

and $\operatorname{dist}(\mathcal{X}, \mathcal{Y})$ is defined in section 2.2.

Moreover, the algorithm converges, that is:

$$\lim_{j \to \infty} \operatorname{dist}(\mathcal{U}, \mathcal{X}_j) = 0.$$

4.4.4 The Lanczos Method

The Lanczos method was introduced in 1950 as a method for the reduction of a symmetric matrix to a tridiagonal form. Twenty years later Paige showed that despite its sensitivity to roundoff, the simple Lanczos algorithm is an effective tool for computing some eigenvalues and their eigenvectors.

The Lanczos algorithm is the Rayleigh–Ritz procedure implemented on the sequence of Krylov subspaces

$$\mathcal{K}_j(A, x) = \operatorname{span}\{x, Ax, A^2x, \ldots, A^{j-1}x\}, \quad j = 1, 2, \ldots.$$

At each step the subspace dimension grows by one, but the costly Rayleigh–Ritz procedure is dramatically simplified. Let $Q_j = [q_1, \ldots, q_j]$ be the orthonormal basis for $\mathcal{K}_j(A, x)$ such that $q_1 = x/\|x\|_2$, then in this basis A's compression $Q_j^T A Q_j$ to $\mathcal{K}_j(A, x)$, is represented by a tridiagonal matrix T_j

$$T_j = \begin{bmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \beta_2 & \ddots & \ddots & \\ & & \ddots & \ddots & \beta_{j-1} \\ & & & \beta_{j-1} & \alpha_j \end{bmatrix},$$

see [78, Chapters 12 and 13]. The algorithm is summarized by two equations,

$$AQ_j - Q_j T_j = r_j e_j^T, \quad r_j = q_{j+1}\beta_j,$$

and

$$I - Q_j^T Q_j = 0.$$

Algorithm 4.4.12 (The Lanczos method). *For a symmetric matrix $A \in \mathbb{R}^{n\times n}$, and for a given vector $x_0 \in \mathbb{R}^n$ this algorithm computes approximations to some eigenvalues and their eigenvectors.*

$r_0 = x_0$; $\beta_0 = \|r_0\|_2 \neq 0$ $q_0 = 0$;
for $j = 1, 2, \ldots$
$q_j = \dfrac{r_{j-1}}{\beta_{j-1}}$;
$u_j = Aq_j$;
$r_j = u_j - q_{j-1}\beta_{j-1}$;
$\alpha_j = q_j^T r_j$;
$r_j = r_j - q_j\alpha_j$
$\beta_j = \|r_j\|_2$;
Compute eigenvalues (the Ritz values) μ_i and eigenvectors s_i of T_j, and the Ritz vectors $v_i = Q_j s_i$;
If satisfied stop.
end

The following results describe the convergence of the Lanczos method.

Theorem 4.4.13 ([85, p. 689]). *Let $A \in \mathbb{R}^{n\times n}$ be a symmetric matrix with eigenvalues $\lambda_1 < \lambda_2 < \cdots < \lambda_k < \lambda_{k+1} \leq \cdots \leq \lambda_n$ and eigenvectors $u_1, \ldots, u_n$. Let P_i be an eigenprojection associated with λ_i, $i \leq k$, and let x_0 the be the starting vector for Algorithm 4.4.12. Let us assume that $P_i x_0 \neq 0$, and consider $u_i = P_i x_0 / \|P_i x_0\|_2$. Set*

$$\gamma_i = 1 - 2\frac{\lambda_n - \lambda_i}{\lambda_n - \lambda_{i+1}}, \quad \text{and} \quad \begin{cases} K_i = \displaystyle\prod_{j=1}^{i-1} \frac{\lambda_n - \lambda_j}{\lambda_i - \lambda_j}, & \text{if } i \neq 1, \\ K_1 = 1. \end{cases}$$

Then

$$\tan\angle(u_i, \mathcal{K}_m(A, x_0)) \leq \frac{K_i}{|T_{m-i}(\gamma_i)|} \tan\angle(u_i, x_0),$$

where $T_\ell(x)$ is a Chebyshev polynomial of the first kind of degree ℓ,

$$T_\ell(x) = \frac{1}{2}\left[\left(x + \sqrt{x^2-1}\right)^\ell + \left(x - \sqrt{x^2-1}\right)^\ell\right], \quad \text{for } |x| > 1.$$

Corollary 4.4.14 ([85, p. 692]). *Let the symmetric matrix $A \in \mathbb{R}^{n\times n}$ and a vector $x_0 \in \mathbb{R}^n$ satisfy the assumptions of Theorem 4.4.13. Let $\mu_1 \leq \cdots \leq \mu_m$ be the eigenvalues of T_m, and assume that $\mu_{i-1} < \lambda_i$. Let γ_i be defined as in Theorem 4.4.13, and let*

$$\begin{cases} N_i = \displaystyle\prod_{j=1}^{i-1} \frac{\lambda_n - \mu_j}{\lambda_i - \mu_j}, & \text{if } i \neq 1, \\ N_1 = 1. \end{cases}$$

Then

$$0 \leq \mu_i - \lambda_i \leq (\lambda_n - \lambda_i)\left(\frac{N_i}{T_{m-i}(\gamma_i)} \tan\angle(u_i, x_0)\right)^2.$$

Corollary 4.4.15 ([85, p. 694]). *Let λ_i be the i-th eigenvalue of A with an associated eigenvector u_i, $\|u_i\|_2 = 1$. Let μ_i be the eigenvalues and v_i the eigenvectors of T_m, and $d_i = \min_{j \neq i} |\lambda_i - \mu_j|$, $r_m = \|(I - \pi_m)A\pi_m\|_2$, where π_m is an orthogonal projection on $\mathcal{K}_m(A, x_0)$. Then*

$$\sin \angle(u_i, v_i) \leq \left(1 + \frac{r_m^2}{d_i^2}\right)^{\frac{1}{2}} \sin \angle(u_i, \mathcal{K}_m(A, x_0)) \leq \left(1 + \frac{r_m^2}{d_i^2}\right)^{\frac{1}{2}} \frac{K_i}{|T_{m-i}(\gamma_i)|} \tan \angle(u_i, x_0).$$

When the Lanczos algorithm is executed in floating point arithmetic, the matrix Q_j might lose its orthogonality. Nevertheless, the following holds [78, p. 270]:

- Orthogonality among q_i, $i = 1, \ldots, j$, is well maintained until one of the Ritz vectors begins to converge.
- Each new Lanczos vector q_{j+1} and each "bad Ritz vector" has a significant component in the direction of each "good Ritz vector".
- The emergence of almost duplicate copies of previously converged Ritz pairs is possible.

4.4.5 Locally Optimal Block Preconditioned Conjugate Gradient Method

Locally Optimal Block Preconditioned Conjugate Gradient Method (LOBPCG) is a quite fast iterative method for computing invariant subspace of a symmetric matrix, based on a local optimization of a three–term recurrence. It was introduced by Knyazev in 2001 [57]. The algorithm combines the preconditioned steepest descent method for the eigenvalue problem and the three–term recurrence of the preconditioned block Lanczos algorithm. The unpreconditioned versions of both methods will be described in more details in Section 5.1.2.

We will start the description of the algorithm by an observation: if λ is an eigenvalue of A, then zero is an eigenvalue of the matrix $A - \lambda I$ with the same eigenvector, and we will show that it is also an eigenvalue of the matrix $T(A - \lambda I)$, where T is a symmetric positive definite preconditioner which approximates A^{-1}. T is chosen so that it accelerates the convergence of the method. So, if λ is the smallest eigenvalue of A, zero is the smallest eigenvalue of $A - \lambda I$. Since $A - \lambda I$ is congruent to $T^{\frac{1}{2}}(A - \lambda I)T^{\frac{1}{2}}$, Sylvester's inertia theorem implies that zero is also the smallest eigenvalue of $T^{\frac{1}{2}}(A - \lambda I)T^{\frac{1}{2}}$. Further, $T^{\frac{1}{2}}(A - \lambda I)T^{\frac{1}{2}} \sim T^{\frac{1}{2}}[T^{\frac{1}{2}}(A - \lambda I)T^{\frac{1}{2}}]T^{-\frac{1}{2}} = T(A - \lambda I)$ which implies that the smallest eigenvalue of $T(A - \lambda I)$ is also equal to zero and all other eigenvalues are positive. On the other hand, if λ is the largest eigenvalue of A, then zero is also the largest eigenvalue of $T(A - \lambda I)$ and all other eigenvalues are negative. In both cases zero is well separated from the rest of the spectrum of $T(A - \lambda I)$. The best choice of λ is to take the Rayleigh quotient $\rho(x_j)$ of the current eigenvector approximation x_j.

It makes sense to employ the Lanczos method on $T(A - \rho(x_j)I)$ to compute its extreme singular vectors. The three–term recurrence of the preconditioned Lanczos

algorithm applied to $T(A-\rho(x_j)I)$ reads

$$r_j = T(A-\rho(x_j)I)x_j - \alpha_j x_j - \beta_{j-1}x_{j-1}, \tag{4.31}$$

where x_j is an eigenvector approximation in the j-th step of the Lanczos algorithm (a Ritz vector). x_{j+1} is obtained by normalization of r_j. So, the new approximation can be obtained from the recurrence of the type

$$x_{j+1} = \alpha_j w_j + \tau_j x_j + \gamma_j x_{j-1}, \quad w_j = T(A-\rho(x_j)I)x_j, \tag{4.32}$$

where the parameters α_j, τ_j and γ_j are chosen using an idea of local optimality: select the parameters that maximize or minimize the Rayleigh quotient

$$\rho(x_{j+1}) = \frac{x_{j+1}^T A x_{j+1}}{x_{j+1}^T x_{j+1}}.$$

If $\rho(x_{j+1})$ is maximized, then we have

$$\max_{\alpha_j,\tau_j,\gamma_j} \rho(x_{j+1}) = \max_{y_{j+1}\in\mathbb{R}^3} \frac{y_{j+1}^T W_{j+1}^T A W_{j+1} y_{j+1}}{y_{j+1}^T W_{j+1}^T W_{j+1} y_{j+1}}, \tag{4.33}$$

where

$$y_{j+1} = \begin{bmatrix} \alpha_j \\ \tau_j \\ \gamma_j \end{bmatrix}, \quad W_{j+1} = [\ w_j \quad x_j \quad x_{j-1}\ ].$$

We assume that W_{j+1} has full column rank. On the other hand if $W_{j+1} = V_{j+1}R_{j+1}$ is the QR factorization of W_{j+1}, where V_{j+1} is orthonormal, then (4.33) implies

$$\begin{aligned} \max_{\alpha_j,\tau_j,\gamma_j} \rho(x_{j+1}) &= \max_{y_{j+1}\in\mathbb{R}^3} \frac{y_{j+1}^T R_{j+1}^T V_{j+1}^T A V_{j+1} R_{j+1} y_{j+1}}{y_{j+1}^T R_{j+1}^T V_{j+1}^T V_{j+1} R_{j+1} y_{j+1}} = \\ &= \max_{z_{j+1}\in\mathbb{R}^3} \frac{z_{j+1}^T V_{j+1}^T A V_{j+1} z_{j+1}}{z_{j+1}^T z_{j+1}}, \end{aligned} \tag{4.34}$$

where $z_{j+1} = R_{j+1}y_{j+1}$. By Theorem 4.1.7 (4.34) implies that finding the optimal x_{j+1} is equivalent to finding the maximum eigenvalue of $V_{j+1}^T A V_{j+1}$, where V_{j+1} is the orthonormal basis for $\mathrm{span}\{w_j, x_j, x_{j-1}\}$. Thus, the Rayleigh–Ritz method can be applied to A and $\mathrm{span}\{w_j, x_j, x_{j-1}\}$, and x_{j+1} can be chosen as the Ritz vector $V_{j+1}z_{j+1}$ corresponding to the maximum Ritz value. Hereby we described a basis of a preconditioned Conjugate Gradient method for invariant subspaces.

As the current eigenvector approximation x_j and the previous eigenvector approximation x_{j-1} are getting closer to each other in the process of iterations, W_{j+1} will be badly conditioned. So, instead of w_j, x_j and x_{j-1} in the three-term recurrence (4.32), the recurrence will involve w_j, x_j and p_j, where p_j is the implicitly computed difference between x_j and x_{j-1}. Now, we obtain the following recurrences:

$$\begin{aligned} x_{j+1} &= \alpha_j w_j + \tau_j x_j + \gamma_j p_j, \quad w_j = T(Ax_j - \rho(x_j)x_j), \\ p_{j+1} &= \alpha_j w_j + \gamma_j p_j, \quad p_0 = 0, \end{aligned} \tag{4.35}$$

with parameters α_j, τ_j and γ_j chosen using the idea of local optimality. We see that

$$p_{j+1} = x_{j+1} - \tau_j x_j,$$

thus

$$x_{j+1} \in \text{span}\{w_j, x_j, p_j\} = \text{span}\{w_j, x_j, x_{j-1}\},$$

therefore (4.35) is equivalent to (4.32), and this defines the main step of the Locally Optimal Preconditioned Conjugate Gradient method.

In case when an invariant subspace is required, which corresponds to the k largest eigenvalues, a simple generalization of the described process will produce a block algorithm. A block version of the Locally Optimal Preconditioned Conjugate Gradient method determines the i-th eigenvector approximation $x_i^{(j+1)}$ as

$$x_i^{(j+1)} \in \text{span}\left\{x_1^{(j-1)}, x_1^{(j)}, T(A-\rho(x_1^{(j)})I)x_1^{(j)}, \ldots, x_k^{(j-1)}, x_k^{(j)}, T(A-\rho(x_k^{(j)})I)x_k^{(j)}\right\},$$

where $x_i^{(j+1)}$ is computed as the i-th Ritz vector. Thus

$$x_i^{(j+1)} = \sum_{\ell=1}^{k} \alpha_{i,\ell}^{(j)} w_\ell^{(j)} + \tau_{i,\ell}^{(j)} x_\ell^{(j)} + \gamma_{i,\ell}^{(j)} p_\ell^{(j)},$$

where $w_\ell^{(j)} = T(A-\rho(x_\ell^{(j)})I)x_\ell^{(j)}$, and

$$p_i^{(j+1)} = x_i^{(j+1)} - \sum_{\ell=1}^{k} \tau_{i,\ell}^{(j)} x_\ell^{(j)}.$$

Again, this implies that

$$\begin{aligned} x_i^{(j+1)} &\in \text{span}\left\{w_1^{(j)}, \ldots, w_k^{(j)}, x_1^{(j)}, \ldots, x_k^{(j)}, p_1^{(j)}, \ldots, p_k^{(j)}\right\} = \\ &= \text{span}\left\{w_1^{(j)}, \ldots, w_k^{(j)}, x_1^{(j)}, \ldots, x_k^{(j)}, x_1^{(j-1)}, \ldots, x_k^{(j-1)}\right\}. \end{aligned}$$

Finally, we can write the whole algorithm.

Algorithm 4.4.16 (LOBPCG). *For a symmetric matrix $A \in \mathbb{R}^{n\times n}$, and for a given orthonormal matrix $X_0 = [x_1^{(0)}, \ldots, x_k^{(0)}]$ this algorithm computes an orthonormal matrix $X = [x_1, \ldots, x_k]$, such that $\mathcal{X} = \text{span}\{X\}$ represents a good approximation to* $\text{span}\{u_{n-k+1}, \ldots, u_n\}$.

for $i = 1 : k$
  $p_i^{(0)} = 0$;
end
for $j = 1, 2, \ldots$
  for $i = 1 : k$
    $\rho(x_i^{(j-1)}) = (x_i^{(j-1)})^T A x_i^{(j-1)}$;
    $r_i^{(j-1)} = A x_i^{(j-1)} - \rho(x_i^{(j-1)}) x_i^{(j-1)}$;

$w_i^{(j-1)} = Tr_i^{(j-1)}$;
Use the Rayleigh–Ritz method for A on the trial subspace
$\operatorname{span}\left\{w_1^{(j-1)}, \ldots, w_k^{(j-1)}, x_1^{(j-1)}, \ldots, x_k^{(j-1)}, p_1^{(j-1)}, \ldots, p_k^{(j-1)}\right\}$;

$x_i^{(j)} = \sum_{\ell=1}^{k} \alpha_{i,\ell}^{(j-1)} w_\ell^{(j-1)} + \tau_{i,\ell}^{(j-1)} x_\ell^{(j-1)} + \gamma_{i,\ell}^{(j-1)} p_\ell^{(j-1)}$,
where $x_i^{(j)}$ is the i-th Ritz vector corresponding to the i-th largest Ritz value;

$p_i^{(j)} = \sum_{\ell=1}^{k} \alpha_{i,\ell}^{(j-1)} w_\ell^{(j-1)} + \gamma_{i,\ell}^{(j-1)} p_\ell^{(j-1)}$;

end
Test X_j for convergence. If the convergence condition is satisfied then stop.
end
$X = X_j$

The following theorem gives a convergence estimate for Algorithm 4.4.16, when an invariant subspace is required, which corresponds to the k smallest eigenvalues.

Theorem 4.4.17 ([59, p. 44]). *The preconditioner T is assumed to satisfy*

$$\delta_0(x^T T x) \le x^T A^{-1} x \le \delta_1(x^T T x), \quad \forall x \in \mathbb{R}^n,\ 0 < \delta_0 \le \delta_1,$$

in every iteration step, where $\kappa(TA) = \delta_1/\delta_0$. For a fixed index $i \in [1,k]$, if $\rho(x_i^{(j)}) \in [\lambda_{\ell_i}, \lambda_{\ell_i+1}\rangle$ then it holds for the Ritz value $\rho(x_i^{(j+1)})$ computed by Algorithm 4.4.16, that either $\rho(x_i^{(j+1)}) < \lambda_{\ell_i}$ (unless $\ell_i = i$), or $\rho(x_i^{(j+1)}) \in [\lambda_{\ell_i}, \rho(x_i^{(j)})\rangle$.

In the latter case

$$\frac{\rho(x_i^{(j+1)}) - \lambda_{\ell_i}}{\lambda_{\ell_i+1} - \rho(x_i^{(j+1)})} \le (q(\kappa(TA), \lambda_{\ell_i}, \lambda_{\ell_i+1}))^2 \frac{\rho(x_i^{(j)}) - \lambda_{\ell_i}}{\lambda_{\ell_i+1} - \rho(x_i^{(j)})},$$

where

$$q(\kappa(TA), \lambda_{\ell_i}, \lambda_{\ell_i+1}) = 1 - \left(1 - \frac{\kappa(TA) - 1}{\kappa(TA) + 1}\right)\left(1 - \frac{\lambda_{\ell_i}}{\lambda_{\ell_i+1}}\right).$$

4.4.6 The Jacobi–Davidson Method

The Jacobi–Davidson method is an iterative method for computing a few of the extreme eigenvalues of a symmetric matrix and corresponding eigenvectors. This method is based on a combination of an old method of Jacobi and of the Davidson method, as described by Sleijpen and van der Vorst in [88]. The main idea is to expand the trial subspace and to apply the Rayleigh–Ritz method on that subspace.

Suppose we are given an eigenvector approximation z_j in the j-th iteration, and we want to find a correction to that approximation that is orthogonal to z_j. Therefore we are interested in seeing what happens in the subspace $\operatorname{span}\{z_j\}^\perp$. The compression of A to that space is given by

$$B = (I - z_j z_j^T) A (I - z_j z_j^T),$$

where z_j is a normalized vector. It follows that

$$A = B + Az_jz_j^T + z_jz_j^TA - \theta_j z_j z_j^T, \qquad \theta_j = z_j^TAz_j. \tag{4.36}$$

When we want to find an eigenvalue λ of A close to θ_j, then we need a correction $t_j \perp z_j$ to z_j such that

$$A(z_j + t_j) = \lambda(z_j + t_j). \tag{4.37}$$

After inserting (4.36) into (4.37), and by using the fact that $Bz_j = 0$ we obtain

$$(B - \lambda I)t_j = -r_j + (\lambda - \theta_j - z_j^TAt_j)z_j, \qquad r_j = Az_j - \theta_j z_j. \tag{4.38}$$

Since the left side of (4.38) and r_j have no component in z_j, it follows that the factor for z_j must vanish, and hence t_j should satisfy

$$(B - \lambda I)t_j = -r_j. \tag{4.39}$$

If we replace λ by the current approximation θ_j, we will obtain the final form of the correction equation

$$(B - \theta_j I)t_j = -r_j, \qquad t_j \perp z_j. \tag{4.40}$$

This equation is usually not solved exactly. Its solution approximation is computed instead, usually by an iterative method. The vector t_j will be used to expand the trial subspace.

The algorithm for finding the invariant subspace which corresponds to the k largest eigenvalues is presented bellow.

Algorithm 4.4.18 (The Jacobi–Davidson method). *For a symmetric matrix $A \in \mathbb{R}^{n\times n}$, and for a given vector x this algorithm computes an orthonormal matrix $X = [x_1, \ldots, x_k]$, such that $\mathcal{X} = \operatorname{span}\{X\}$ represents a good approximation of $\operatorname{span}\{u_{n-k+1}, \ldots, u_n\}$.*

$x_1 = \frac{x}{\|x\|_2}$; $\quad y_1 = Ax_1$; $\quad h_{11} = x_1^Ty_1$;

Set $X_1 = [x_1]$, $\quad Y_1 = [y_1]$, $\quad H_1 = X_1^TAX_1 = [h_{11}]$;

$z_1 = x_1$; $\quad \theta_1 = h_{11}$; $\quad r_1 = y_1 - \theta_1 z_1$;

until *convergence* **do**

for $j = 1 : m - 1$

Solve (approximately)
$(I - z_jz_j^T)(A - \theta_j I)(I - z_jz_j^T)t_j = -r_j \qquad$ *for* $t_j \perp z_j$;

Orthogonalize t_j against X_j using modified Gram–Schmidt, to obtain x_{j+1}

$X_{j+1} = [\ X_j \quad x_{j+1}\ ]$;

$y_{j+1} = Ax_{j+1}$;

$Y_{j+1} = [\ Y_j \quad y_{j+1}\ ]$;

$$H_{j+1} = \begin{bmatrix} H_j & X_j^Ty_{j+1} \\ y_{j+1}^TX_j & x_{j+1}^Ty_{j+1} \end{bmatrix};$$

Compute the largest $\min\{k, m\}$ *eigenpairs of* H_{j+1},
let (θ_{j+1}, s_{j+1}) *be the eigenpair such that* θ_{j+1} *is the largest eigenvalue;*

$z_{j+1} = X_{j+1} s_{j+1}$;

$\bar{z}_{j+1} = A z_{j+1}$;

$r_{j+1} = \bar{z}_{j+1} - \theta_{j+1} z_{j+1}$;

Test for convergence and stop if satisfied;

end

Set $X_1 = [z_{j+1}], \quad Y_1 = [\bar{z}_{j+1}], \quad H_1 = [\theta_{j+1}]$;

end

$X = X_{j+1}(:, 1:k)$;

Chapter 5

Multispace

In case when a matrix A is symmetric positive definite then all its eigenvalues are positive, and inverse subspace iteration will produce an approximation to invariant subspace U, which corresponds to the k smallest eigenvalues. On the other hand, in case when λ_{k+1} is close to λ_k inverse subspace iteration might converge very slowly (see Theorem 4.4.8).

The inverse iterations are usually used to compute an invariant subspace, when good approximations to eigenvalues are known. In that case sometimes only few iterations are needed to obtain accurate solution. The only problem arises when the desired subspace corresponds to the eigenvalues that are not well separated from the rest of the spectrum. The slow convergence in that case does not mean that all eigenvectors converge slowly. The eigenvectors, whose eigenvalues are far from the rest of the spectrum will converge faster. If we have some information about the spectrum, then we should start with a subspace with larger dimension than desired, which will guarantee faster convergence. On the other hand, the Lanczos method produces a sequence of Krylov subspaces with increasing dimensions, and the accuracy of the eigenvector approximations is increasing with dimension. The inverse iterations are dealing with the subspaces of the same dimension and the subspaces from the previous iterations are not involved in the current iteration. The current subspace of the Lanczos method includes all subspaces from the previous steps, and thus improves eigenvector approximations.

The idea of the multispace method is to speed up the slow convergence of the inverse subspace iteration by a technique similar to the multigrid method for solving linear systems. The multigrid method uses a simple iteration method which usually stagnates after a couple of iterations, and then transfers the whole problem to the smaller dimension. This transfer to the smaller dimension improves the convergence of the whole process.

5.1 The Algorithms

5.1.1 Multigrid Algorithms

Let us first start with a simple description of the multigrid algorithms. Originally, multigrid methods were developed for solving boundary value problems defined on certain spatial domains. Such problems were discretized over a set of nodes in the domain, which were organized in a grid. The resulting discretized problem becomes a problem of solving a linear system, whose matrix has a very specific structure, and unknowns correspond to the solution value in the nodes. The main idea of multigrid is solving this system on coarser and coarser grids with an iterative method, thus making the dimension of the problem smaller and smaller. After reaching the coarsest grid, the solution approximation is interpolated back to the finer grids. It turns out that this principle has good convergence properties. Such methods can be applied to a wide class of problems, even to the problems that are not associated with a physical grid. This was the basis for the development of the algebraic multigrid.

Algorithm 5.1.1 (Multigrid V cycle). *Let $A^{(1)} = A$, $A^{(k)} \in \mathbb{R}^{n_k \times n_k}$, $b^{(1)} = b$, $b^{(k)} \in \mathbb{R}^{n_k}$, where $n_1 > n_2 > \cdots > n_s$, and let us denote the restriction operator from finer to coarser grid by $\mathbf{I}_{(n_{k+1},n_k)} \in \mathbb{R}^{n_{k+1} \times n_k}$ and the interpolation operator from coarser to finer grid by $\mathbf{I}_{(n_k,n_{k+1})} \in \mathbb{R}^{n_k \times n_{k+1}}$. Let $y^{(k)}$ be the initial approximation, then the following algorithm computes the solution of the system $A^{(k)}x^{(k)} = b^{(k)}$.*

1. *Perform p iterations of the iterative method obtaining a new approximation $y^{(k)}$*
2. *Compute $r^{(k)} = b^{(k)} - A^{(k)}y^{(k)}$ and restrict it to the smaller dimension by $r^{(k+1)} = \mathbf{I}_{(n_{k+1},n_k)}r^{(k)}$*
3. *Compute $A^{(k+1)} = \mathbf{I}_{(n_{k+1},n_k)}A^{(k)}\mathbf{I}_{(n_k,n_{k+1})}$ and approximately solve $A^{(k+1)}e^{(k+1)} = r^{(k+1)}$ on the smaller dimension*
4. *Interpolate $e^{(k)}$ to the original dimension by $e^{(k)} = \mathbf{I}_{(n_k,n_{k+1})}e^{(k+1)}$ and compute $y^{(k)} = y^{(k)} + e^{(k)}$*
5. *Perform p iterations of the iterative method obtaining the final approximation $y^{(k)}$*

Step 3. denotes a recurrent call to the multigrid routine.

As we can see, the iterative method for solving linear systems is first applied to the original matrix, then again to the restricted matrix with smaller dimension, and so on, until the smallest dimension is reached. On the smallest dimension the problem is solved exactly, i.e. with a direct method, and the solution is interpolated back to the larger dimensions.

The matrix of the discretized system, the restriction and the interpolation operator depend on three things:

- differential operator of the original problem

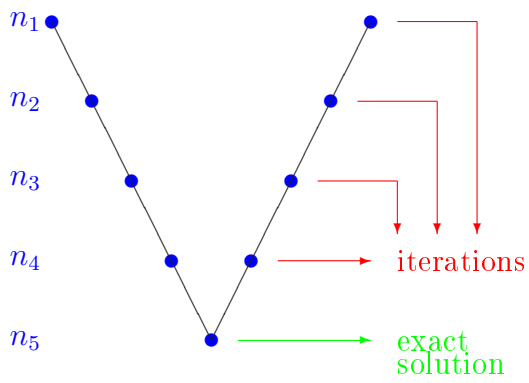


Figure 5.1: A multigrid V cycle.

- grid
- discretization of derivations

For example, a discretized Laplace operator on a squared domain will produce a banded symmetric positive definite matrix. The condition number of this matrix increases with n, and the Jacobi and the Gauss–Seidel iterations stagnate after a modest number of iterations. In this case the restriction and the interpolation operator are chosen so that $\mathbf{I}^T_{(m,n)} = c\mathbf{I}_{(n,m)}$, where c is a constant, and the transfer to a smaller dimension and back improves the convergence. It eliminates directions in the solution approximation that caused slow convergence.

The idea of using multigrid for solving an eigenvalue problem is not new. Until now, many eigensolvers for elliptic eigenvalue problems with multigrid efficiency were developed, see [76]. According to [76], multigrid eigensolvers can be classified in the following three categories:

The Rayleigh quotient multigrid minimization (RQMG)

When the eigenvalue problem for a self-adjoint elliptic partial differential operator is considered, then the discrete matrix eigenproblem can be treated as an optimization problem for the Rayleigh quotient

$$\rho(x) = \frac{x^T A x}{x^T x}.$$

By the Courant–Fischer principle (Theorem 4.1.7) the minimum of $\rho(x)$ equals to the smallest eigenvalue of A, and is taken at the corresponding eigenvector. Hence the iterative minimization of $\rho(x)$ can serve as an eigensolver.

This minimization can be realized by means of a multigrid procedure. A coordinate relaxation scheme is applied, i.e. for each coordinate direction d_i^k (which is associated with the i-th finite element function on a certain grid level k) the minimum

$$\rho(x + \gamma d_i^k) = \min_{\tau \in \mathbb{R}} \frac{(x + \tau d_i^k)^T A (x + \tau d_i^k)}{(x + \tau d_i^k)^T (x + \tau d_i^k)}$$

is computed, which is at the same time the smallest Ritz value of A in the 2D space $\text{span}\{x, d_i^k\}$. The new iterate is $x + \gamma d_i^k$. A multigrid cycle of RQMG consists in a successive minimization of $\rho(x)$ for all finite element functions on all grid levels. It is interesting to note that

$$\nabla\rho(x) = \frac{2}{x^T x}(Ax - \rho(x)x).$$

For more information see [10], [68] and [71].

Direct multigrid eigensolvers

Direct multigrid eigensolvers and the third class of eigensolvers are related to approximate variants of inverse iteration and the Rayleigh quotient iteration. Inverse iterations are usually dealing with almost singular matrices $A - \lambda I$ when λ is close to an eigenvalue of A. So, solving a linear system with such a matrix can be quite difficult. Alternatively, one can solve a non-singular coarse grid correction equation within the orthogonal complement of the actual eigenvector approximation. This approach provides the basis for the direct multigrid eigensolver.

The resulting two-grid method maps a given iterate x having the Rayleigh quotient $\rho(x)$ to the new eigenvector approximations x'. It is given by

$\tilde{x} = Sx$	Smoothing step
$d_c = R(A - \rho(x)I)\tilde{x}$	Coarse grid projection of the residual
$d_c^\perp = Q_c d_c$	Orthogonal projection
$y_c = (A_c - \rho(x)I_c)^{-1} d_c^\perp$	Solution of correction equation
$x' = x - PQ_c y_c$	Prolongation and correction

Here, the index c denotes coarse grid quantities. R is a restriction operator, P is an interpolation operator and Q_c is the orthogonal projection operator to the orthogonal complement of the actual eigenvector approximation. For more information see [9], [41], [42] and [46].

Eigensolvers using multigrid as a linear solver

Another way to avoid solving a near singular linear system in inverse iteration, is to apply multigrid preconditioning for A in order to determine an approximate solution of the linear system

$$Ax^{(i+1)} = \rho(x^{(i)})x^{(i)}, \quad i = 0, 1, 2, \ldots. \tag{5.1}$$

A scaling constant $\rho(x^{(i)})$ is introduced in order to achieve its stationarity in eigenvectors.

The multigrid preconditioner B^{-1} is an approximate inverse of A, which is assumed to be a symmetric positive definite operator, such that

$$\|I - B^{-1}A\|_A \leq \gamma, \tag{5.2}$$

for a constant $\gamma \in [0, 1\rangle$, where $\|\cdot\|_A$ denotes the operator norm induced by A. The best preconditioners satisfy (5.2) where γ is bounded away from 1, and is

independent of the mesh size and of the number of unknowns. The approximate solution of (5.1) for $x = x^{(i)}$ and by using B^{-1} as a preconditioner yields a new iterate x' approximating $x^{(i+1)}$

$$x' = x - B^{-1}(Ax - \rho(x)x). \tag{5.3}$$

The iteration (5.3) can be considered as the most simple eigensolver embodying the idea of multigrid as a linear solver. It can be interpreted as a (multigrid) *preconditioned variant of inverse iteration* (PINVIT). For more information see [58] and [75].

Unfortunately, the cubic convergence of the Rayleigh quotient iteration cannot be transferred to the preconditioned multigrid case. A significant acceleration of (5.3) can be achieved by using multigrid preconditioners in LOBPCG [59].

The results of all these eigensolvers still represent a point in vector space $\mathbb{R}^n$, while on the other hand the solution of an eigenvalue problem is the whole subspace. So, a better approach would be to combine a multigrid technique on subspaces with decreasing dimensions, rather then grids. This is the foundation of the *multispace* method, described in the next subsection.

5.1.2 Multispace Algorithm

Let us assume that $A \in \mathbb{R}^{n\times n}$ is a symmetric and positive definite matrix. This is the only condition imposed on A. The structure of the matrix A and its origin are not important. In this case, inverse subspace iteration will converge to the k smallest eigenvalues if we take the shift to be zero, which is according to Ky–Fan minimum principle (Theorem 4.1.10) equivalent to minimizing trace functional

$$\rho(X) = \mathrm{trace}((X^TX)^{-1}(X^TAX)), \tag{5.4}$$

where $X \in \mathbb{R}^{n\times k}$ has full column rank. The subspace that minimizes $\rho(X)$ is spanned by the first k eigenvectors of A. To see that, we denote the orthogonal projection on the range(X) with π, where $\pi = XX^\dagger = X(X^TX)^{-1}X^T$. Then it is easy to see that

$$\rho(X) = \mathrm{trace}((X^TX)^{-1}X^TAX) = \mathrm{trace}(A\pi) = \mathrm{trace}(\pi A) = \mathrm{trace}(\pi A\pi),$$

which follows from the fact that $\pi^2 = \pi$, and $\mathrm{trace}(AB) = \mathrm{trace}(BA)$ for any matrices A and B. This implies that, finding the gradient of ρ will only involve finding the gradient of π. Further, if $\bar{X}$ spans an orthonormal basis for range(X), then we can use it instead of X, and the expression for $\rho(\bar{X})$ is now simplified. So, from the Ky–Fan minimum principle and the Cauchy's interlacing theorem (Corollary 4.1.9) now it follows

$$\rho(\bar{X}) = \mathrm{trace}(\bar{X}^TA\bar{X}) = \sum_{i=1}^{k}\mu_i \geq \sum_i \lambda_i = \mathrm{trace}(U^TAU),$$

where μ_i are eigenvalues of $\bar{X}^TA\bar{X}$, and $U = [\ u_1 \ \cdots \ u_k\ ]$ with $Au_i = \lambda_i u_i$.

From now on we will use the following notation: for each matrix $S \in \mathbb{R}^{n\times k}$ a k-dimensional subspace spanned by the columns of S will be denoted by $\mathcal{S} = \text{range}\{S\}$.

Now, we want to use the multigrid idea, in its two-grid correction form. Let $X_0 \in \mathbb{R}^{n\times k}$ be an orthonormal matrix such that $\mathcal{X}_0$ represents initial approximative subspace. Further, let $W^{(n,m)} \in \mathbb{R}^{n\times m}$ be an orthonormal matrix which spans an m-dimensional subspace $\mathcal{W}^{(n,m)} \subset \mathbb{R}^n$, where $k \leq m < n$. Then, the basis of the new algorithm should be:

1. Perform p steps of inverse iteration, starting with X_0 in order to obtain X_p.
2. Compute $A^{(m)} = (W^{(n,m)})^T A W^{(n,m)}$, and restrict X_p to the smaller dimension by $Y_0 = (W^{(n,m)})^T X_p$.
3. Find an orthonormal basis $Y \in \mathbb{R}^{m\times k}$ belonging to the k smallest eigenvalues of $A^{(m)}$, by inverse iteration starting with Y_0.
4. Transfer Y to the original dimension by $\bar{X}_p = W^{(n,m)} Y$.
5. Perform p more steps of inverse iteration, starting with $\bar{X}_p$ in order to obtain X_{2p}.

Step 3. denotes a recurrent call to the multispace routine. Here $W^{(n,m)}$ stands for the interpolation operator and $(W^{(n,m)})^T$ for the restriction operator. Now, one question still remains: how to choose an appropriate subspace $\mathcal{W}^{(n,m)}$? The answer relies on the equation (5.4).

The first assumption should be that $\mathcal{X}_p = \text{span}\{X_p\} \subset \mathcal{W}^{(n,m)}$. So, if we define $Z \in \mathbb{R}^{n\times m}$ as a trial subspace basis for $\mathcal{W}^{(n,m)}$, we can put

$$Z(1:n, 1:k) = X_p.$$

Now we have to find remaining $m-k$ vectors in the basis Z, which we denote by $P \in \mathbb{R}^{n\times(m-k)}$

$$Z(1:n, k+1:m) = P.$$

Once we construct Z we can take QR factorization

$$Z = W^{(n,m)} R, \quad \text{where} \quad W^{(n,m)}(1:n, 1:k) = X_p,\ Z(1:n, k+1:m) = \bar{P},$$

and take $W^{(n,m)}$ as the desired orthonormal basis, so that $Y_0 = \begin{bmatrix} I_k \\ 0 \end{bmatrix}$. The good choice for P would be such that $\mathcal{W}^{(n,m)}$ contains directions in which ρ would be minimized even better. This means that we want

$$\min_{\substack{Y\in\mathbb{R}^{m\times k},\\ Y^TY=I_k}} \text{trace}(Y^T A^{(m)} Y) = \min_{\substack{Y\in\mathbb{R}^{m\times k},\\ Y^TY=I_k}} \text{trace}((W^{(n,m)}Y)^T A W^{(n,m)} Y) \ll \text{trace}(X_p^T A X_p), \tag{5.5}$$

and we want this trace reduction to be as large as possible. Let us take a better look at this trace minimization. If we make a partition

$$W^{(n,m)} = [\ X_p \quad \bar{P}\ ], \qquad Y = \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix},$$

where $Y_1 \in \mathbb{R}^{k\times k}$ and $Y_2 \in \mathbb{R}^{(m-k)\times k}$, then, as in [30]

$$\begin{aligned}\min_{\substack{Y\in\mathbb{R}^{m\times k}\\ Y^TY=I_k}} \operatorname{trace}((W^{(n,m)}Y)^T A W^{(n,m)}Y) &= \min_{\substack{Y\in\mathbb{R}^{m\times k}\\ \operatorname{rank}(Y)=k}} \rho(W^{(n,m)}Y)\\ &= \min_{\substack{Y\in\mathbb{R}^{m\times k}\\ \operatorname{rank}(Y)=k}} \rho(X_pY_1 + \bar{P}Y_2).\end{aligned}$$

It turns out that $\bar{P}$ spans an $(m-k)$-dimensional search direction $\mathcal{P} \perp \mathcal{X}_p$ for the minimization of the functional ρ, taking $\mathcal{X}_p$ as starting point. So, we must choose $\mathcal{P}$ in such a way, that it contains directions of rapid functional descent.

In case of the minimization of a functional defined on a vector space, the direction of its fastest descend is equal to minus functional gradient [3, Lemma 8.6.1]. Thus, the iterative steepest descent method uses functional gradient as a search direction. Since we want to minimize the functional ρ, we will also observe its steepest descend direction which is constructed in [73] as

$$G(X) = \nabla\rho(X) = 2(AX - XH_X)(X^TX)^{-1}, \quad H_X = (X^TX)^{-1}(X^TAX),$$

where $\operatorname{rank}(X) = k$. The problem is that $G(X) \in \mathbb{R}^{n\times k}$, and usually $m \geq k$, so we are still missing more directions in subspace $\mathcal{W}^{(n,m)}$. The first step of the steepest descent method would find S_1, such that

$$\min_{\substack{Y\in\mathbb{R}^{2k\times k}\\ \operatorname{rank}(Y)=k}} \rho(X_pY_1 - G(X_p)Y_2) = \rho(S_1). \tag{5.6}$$

where minimum is obtained for $\bar{Y}_1$ and $\bar{Y}_2$, and

$$\begin{aligned} S_1 &= X_p\bar{Y}_1 - G(X_p)\bar{Y}_2 = X_p\cdot(\bar{Y}_1 + 2H_{X_p}\bar{Y}_2) - AX_p\cdot 2\bar{Y}_2 \quad \text{with}\\ G(X_p) &= 2(AX_p - X_pH_{X_p}).\end{aligned}$$

This implies that

$$S_1 \subset \operatorname{span}\{X_p, AX_p\} = \mathcal{K}_2(A, X_p),$$

where $\mathcal{K}_j(A, X_p) = \mathcal{K}_j(A, \mathcal{X}_p)$ is a block Krylov subspace.

Here we have to be careful of how we construct the matrix S_1. The following relations hold

$$X_pY_1 - G(X_p)Y_2 = \begin{bmatrix} X_p & -G(X_p)\end{bmatrix}\begin{bmatrix} Y_1\\ Y_2\end{bmatrix},$$

and

$$\rho(X_pY_1 - G(X_p)Y_2) = \tilde{\rho}\left(\begin{bmatrix} Y_1\\ Y_2\end{bmatrix}\right),$$

where

$$\begin{aligned}\tilde{\rho}\left(\begin{bmatrix} Y_1\\ Y_2\end{bmatrix}\right) &= \operatorname{trace}\left(\left(\begin{bmatrix} Y_1\\ Y_2\end{bmatrix}^T\tilde{B}\begin{bmatrix} Y_1\\ Y_2\end{bmatrix}\right)^{-1}\left(\begin{bmatrix} Y_1\\ Y_2\end{bmatrix}^T\tilde{A}\begin{bmatrix} Y_1\\ Y_2\end{bmatrix}\right)\right),\\ \tilde{A} &= \begin{bmatrix} X_p & -G(X_p)\end{bmatrix}^TA\begin{bmatrix} X_p & -G(X_p)\end{bmatrix},\\ \tilde{B} &= \begin{bmatrix} X_p & -G(X_p)\end{bmatrix}^T\begin{bmatrix} X_p & -G(X_p)\end{bmatrix}.\end{aligned}$$

So, if $[\ X_p \quad -G(X_p)\ ]$ has full column rank, then the minimization in (5.6) is equivalent to finding the k smallest eigenvalues with corresponding eigenvectors of the $2k \times 2k$ generalized eigenvalue problem

$$\tilde{A}y = \mu \tilde{B}y.$$

Since, $X_p^T G(X_p) = 0$, $[\ X_p \quad -G(X_p)\ ]$ will have full column rank if $G(X_p)$ has full column rank. Let $g_p = \text{rank}(G(X_p))$, and let $\bar{G}_p \in \mathbb{R}^{n \times g_p}$ be the orthonormal basis for range$(G(X_p))$. If $g_p < k$, that means that $\mathcal{X}_p$ contains a subspace which is A invariant. In that case we can take $\bar{G}_p$ as a new search direction, and define

$$\bar{A} = [\ X_p \quad -\bar{G}_p\ ]^T A [\ X_p \quad -\bar{G}_p\ ] \in \mathbb{R}^{(k+g_p)\times(k+g_p)}.$$

In this way (5.6) is equivalent to finding the k smallest eigenvalues with corresponding eigenvectors of the matrix $\bar{A}$.

If we assume that the $(j-1)$-th step of the steepest descent produces S_{j-1}, such that $S_{j-1} \subset \mathcal{K}_j(A, X_p)$, then the j-th step will produce

$$S_j = S_{j-1}\bar{Y}_1 - G(S_{j-1})\bar{Y}_2 = S_{j-1}\cdot(\bar{Y}_1 + 2H_{S_{j-1}}(S_{j-1}^T S_{j-1})^{-1}\bar{Y}_2) - AS_{j-1}\cdot 2(S_{j-1}^T S_{j-1})^{-1}\bar{Y}_2,$$

and

$$S_j \subset \mathcal{K}_j(A, X_p) + A\mathcal{K}_j(A, X_p) = \mathcal{K}_{j+1}(A, X_p).$$

We can conclude that for $S_0 = X_p$

$$S_j \subset \mathcal{K}_{j+1}(A, X_p), \qquad j = 0, \ldots, \ell - 1,\ \ell = \left\lceil \frac{m}{k} \right\rceil.$$

Again, in case when $G(S_{j-1})$ is rank deficient, we should choose the orthonormal basis for range$(G(S_{j-1}))$ as a new search direction.

Finally, if we chose

$$P = [\ AX_p \quad A^2X_p \quad \cdots \quad A^{\ell-1}X_p\ ] \in \mathbb{R}^{n\times(\ell-1)k},$$

then $\mathcal{W}^{(n,m)} = \mathcal{K}_\ell(A, X_p)$ will contain the solution approximation from $\ell - 1$ consecutive steepest descent steps. So, if we started with X_p, any orthonormal basis $W^{(n,m)}$ of $\mathcal{W}^{(n,m)}$ will satisfy condition (5.5), and the functional reduction will be satisfactory, because we chose directions of steepest descent. A concrete implementation of an algorithm generating $W^{(n,m)}$ should detect any rank deficiency in $[\ X \quad P\ ]$, and it should contain only the columns that span range$([\ X \quad P\ ])$. In case when $X_0 = U$ is the exact solution, that is $AU = U\Lambda$, the inverse iteration will not change $\mathcal{U} = \text{range}(U)$, since $A^{-1}U = U\Lambda^{-1}$. Further, rank$([\ U \quad P\ ]) = k$ and the space $\mathcal{W}^{(n,m)}$ will coincide with $\mathcal{X}_0$ and $A^{(m)} = \Lambda$. Hence, we are done and there is no need for further iterations.

Here we can use the block Lanczos algorithm to produce $W^{(n,m)}$ and the matrix $A^{(m)}$. The block version of the Lanczos algorithm works the same way as the standard version except that it starts with an $n \times k$ matrix instead of a vector. Hence, for the orthonormal basis $Q^{(j)} = [Q_1, \ldots, Q_j]$ of $\mathcal{K}_j(A, X_0)$, the matrix $T_j = (Q^{(j)})^T A Q^{(j)} \in \mathbb{R}^{jk\times jk}$ is banded,

and has the following form

$$T_j = \begin{bmatrix} A_1 & B_1^T & & & \\ B_1 & A_2 & B_2^T & & \\ & B_2 & \ddots & \ddots & \\ & & \ddots & \ddots & B_{j-1}^T \\ & & & B_{j-1} & A_j \end{bmatrix}, \quad A_i \in \mathbb{R}^{k\times k}$$

see [78, Chapter 13]. Here is the block Lanczos algorithm.

Algorithm 5.1.2 (The block Lanczos method). *For a symmetric matrix $A \in \mathbb{R}^{n\times n}$, and a given orthonormal matrix $X_0 \in \mathbb{R}^{n\times k}$, this algorithm computes an orthonormal basis W for $\mathcal{K}_{\lceil \frac{m}{k} \rceil}(A, X_0)$ and $T = W^T A W$.*

function $[W, T] =$ **block_ lanczos**(A, X_0, m)

$\ell = \lceil \frac{m}{k} \rceil$;
$R_0 = X_0$; $Q_0 = 0$;
for $j = 1 : \ell$
  Compute QR factorization $R_{j-1} = Q_j B_{j-1}$;
  $R_j = AQ_j - Q_{j-1}B_{j-1}^T$;
  $A_j = Q_j^T R_j$;
  $R_j = R_j - Q_j A_j$
end
$W = [\ Q_1 \ \cdots \ Q_\ell\ ]$, $T = T_\ell$
end

Since the matrix $A^{(m)}$ in the next multispace level is banded, it makes matrix–vector multiplication cheap. Instead of solving the problem (5.5) for the matrix $A^{(m)}$, we can apply again the same procedure described above, by performing several steps of inverse iteration. Hence, the algorithm switches to subspaces with smaller and smaller dimension until it reaches the smallest dimension. On the smallest dimension the problem is solved by a direct method. We take $m_{max} = \ell_{max} \cdot k$ to be the largest acceptable dimension of $\mathcal{W}^{(n,m)}$, where $\ell_{max} = \dim(\mathcal{W}^{(n,m)})/\dim(\mathcal{X})$ is the largest dimension increasing factor.

Now we can derive the whole algorithm.

Algorithm 5.1.3 (Multispace V cycle). *For a symmetric positive definite matrix $A \in \mathbb{R}^{n\times n}$, and a given orthonormal matrix $X_0 = [x_1^{(0)}, \ldots, x_k^{(0)}]$ this algorithm computes an orthonormal matrix $X = [x_1, \ldots, x_k]$, such that $\mathcal{X} = \mathrm{range}\{X\}$ represents a good approximation to $\mathrm{span}\{u_1, \ldots, u_k\}$.*

function $X =$ **multispace**(A, X_0)

if $(n \le \min\{2k, n_{min}\})$
  Exactly solve the problem (4.29) by a direct method;
else

$\ell = \min\{\lceil \frac{n}{4k} \rceil, \ell_{max}\}$; $m = \ell \cdot k$;
$X_p = \textbf{inverse\_iteration}(A, X_0, p)$;
$[W^{(n,m)}, A^{(m)}] = \textbf{block\_lanczos}(A, X_p, m)$;
$Y_0 = I_m(:, 1:k)$;
$Y = \textbf{multispace}(A^{(m)}, Y_0)$;
$\bar{X}_p = W^{(n,m)} Y$;
$X_{2p} = \textbf{inverse\_iteration}(A, \bar{X}_p, p)$;
end
end

function $X = \textbf{inverse\_iteration}(A, X_0, p)$
{*Implements the inverse subspace iteration (Algorithm 4.4.7) for the symmetric matrix A and the starting k-dimensional approximation X_0. It performs p iterations.*}

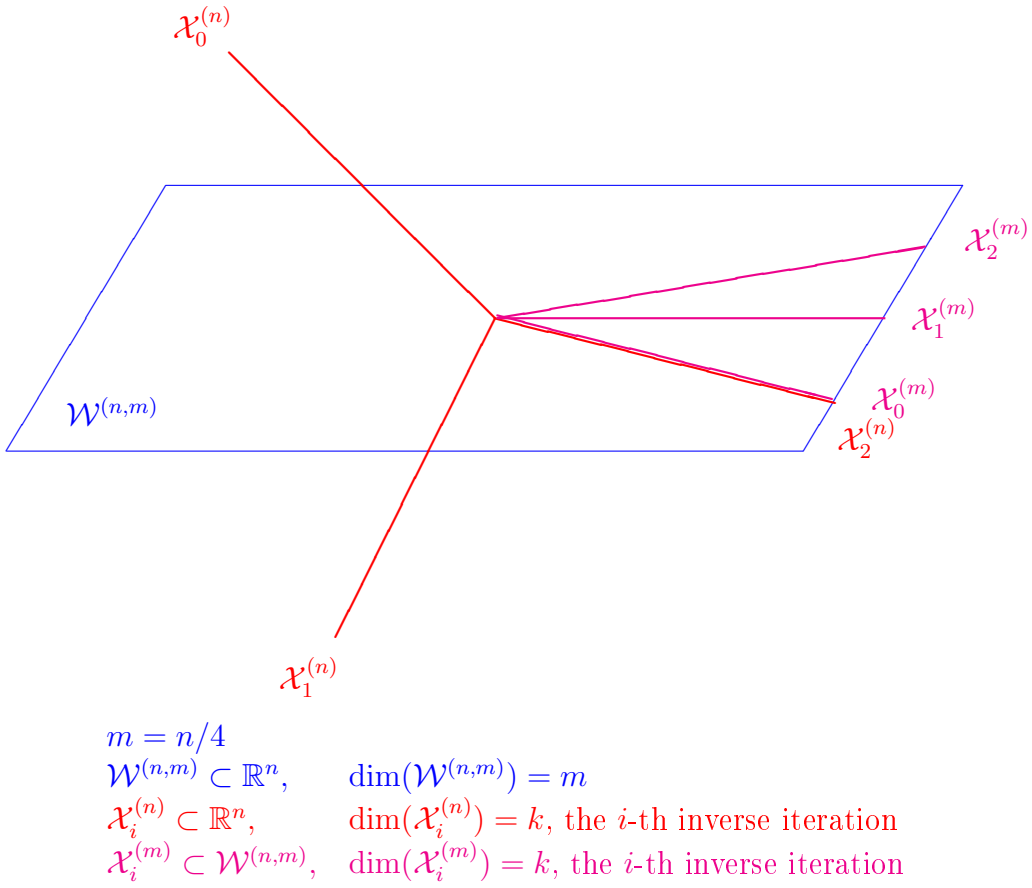


Figure 5.2: Inverse iteration and transfer to the subspace in multispace method

In our preliminary implementation, which is made only to study the convergence properties, the linear systems in the inverse iteration are solved by the Cholesky factorization. The factorization is computed only once, and applied p times. In the inverse

iteration implementation there exist many possibilities for improvements, such as: to employ the sparsity structure of the matrix, use the data from the factorization of the previous multispace level with larger dimension, or to apply some iterative method for solving linear systems. The bottleneck of the block Lanczos algorithm are QR factorizations, which we will try to reduce in the future work. The important thing here is, that we are still far from an efficient implementation, but we are offering a new approach to iterative methods for partial eigenvalue problem. When the inverse iteration stagnate, we are searching for the new direction in the subspace of larger dimension. The dimension of the subspace is successively decreasing similarly to the multigrid idea.

Let us take a look at the floating point operation count of the multispace method. First, let us define the following notations:

$C_{MS}(n)$	=	operation count for multispace in dimension n,
$C_{II}(n,p)$	=	operation count for inverse iteration in dimension n, when p iterations are performed,
$C_{BL}(n,m)$	=	operation count for the block Lanczos method when subspace transfer is performed from dimension n to dimension m.
$C_{MP}(n,m,k)$	=	operation count for matrix product $n \times m$ times $m \times k$.
$C_0(n)$	=	operation count for exact eigensolver in dimension n.

Let us take $m = n/2^q$, $\dim(\mathcal{X}_0) = k$, and $m = \ell \cdot k$, then we have

$$C_{MS}(n) = 2C_{II}(n,p) + C_{BL}\left(n, \frac{n}{2^q}\right) + C_{MP}\left(n, \frac{n}{2^q}, k\right) + C_{MS}\left(\frac{n}{2^q}\right).$$

Let s be the total number of multispace levels, that means that $n/2^{sq}$ is the smallest dimension where the problem is solved directly, and $n/2^{sq} \approx 2k$. Than we can obtain

$$C_{MS}(n) = 2\sum_{i=0}^{s-1} C_{II}\left(\frac{n}{2^{iq}}, p\right) + \sum_{i=0}^{s-1} C_{BL}\left(\frac{n}{2^{iq}}, \frac{n}{2^{(i+1)q}}\right) + \sum_{i=0}^{s-1} C_{MP}\left(\frac{n}{2^{iq}}, \frac{n}{2^{(i+1)q}}, k\right) + C_0\left(\frac{n}{2^{sq}}\right).$$

For example, for the concrete implementation in dimension n we can have the following values:

Matrix product

operation	operation count
$n \times m$ times $m \times k$	$2nmk$

Inverse iteration *($2p$ **iterations**)*

operation	operation count
The Cholesky factorization	$n^3/3$
solving 2 triangular systems	$4pn^2k$
QR factorizations	$8pn^2k$
total	$n^3/3 + 12pn^2k$

The block Lanczos method

operation	operation count
QR factorizations	$\ell \cdot 4n^2k = 4n^3/2^q$
matrix products	$\ell \cdot (2n^2k + 6nk^2) = 2n^3/2^q + 6n^2k/2^q$
total	$6n^3/2^q + 6n^2k/2^q$

Direct eigensolver

operation	operation count
solver in dimension $n/2^{sq}$	$O(k^3)$

This will result with

$$C_{MS}(n) = \left(\frac{1}{3} + \frac{6}{2^q}\right) \frac{1 - \frac{1}{2^{3sq}}}{1 - \frac{1}{2^{3q}}} n^3 + \left(12p + \frac{8}{2^q}\right) \frac{1 - \frac{1}{2^{2sq}}}{1 - \frac{1}{2^{2q}}} n^2 k + O(k^3).$$

To keep the balance between convergence and efficiency, we chose $q = 2$ in most of our numerical tests. We will observe two cases: $p = 2$ and $p = 5$, because $p > 5$ does not pay off. Then

$p = 2$

$$\begin{aligned} C_{MS}(n) &= 1.86\left(1 - \frac{1}{64^s}\right) n^3 + 27.73\left(1 - \frac{1}{16^s}\right) n^2 k + O(k^3) \\ C_{II}(n, 2p) &= 0.33n^3 + 24n^2 k \end{aligned}$$

$p = 5$

$$\begin{aligned} C_{MS}(n) &= 1.86\left(1 - \frac{1}{64^s}\right) n^3 + 66.13\left(1 - \frac{1}{16^s}\right) n^2 k + O(k^3) \\ C_{II}(n, 2p) &= 0.33n^3 + 60n^2 k \end{aligned}$$

Now we can conclude that if the inverse iteration converges very slowly, than it pays off to add more floating point operations in order to increase the rate of convergence. But, does multispace justify these extra expenses? The answer to this question is given in the next two sections.

5.2 Convergence

The rates of convergence of the multispace method are based on the results of Saad [85]. We start with two-space analysis, where only one transfer to the subspace is performed, and the problem is solved exactly in the subspace. Multispace can be regarded as a perturbed two-space method.

For the main result we need the following lemmas.

Lemma 5.2.1. *Let π_0 be an orthogonal projection on $\mathcal{X}_0$, which is the starting subspace in the multispace algorithm. Let us assume that $\mathcal{X}_0$ is such that the vectors $\pi_0 u_1$, $\pi_0 u_2$, ..., $\pi_0 u_k$ are independent. Then there exists in X_0 a unique vector $\bar{x}_i$ such that*

$$u_j^T \bar{x}_i = \delta_{ij}, \quad \text{for } j = 1, \ldots, k. \tag{5.7}$$

The vector $\bar{x}_i$ is the vector of $\mathcal{X}_0$ whose orthogonal projection on $\mathcal{U} = \text{span}\{u_1, \ldots, u_k\}$ is exactly u_i.

Proof. See the proof of Lemma 4. in [85, p. 699]. □

Lemma 5.2.2. *Let $Q = [q_1, \ldots, q_k] \in \mathbb{R}^{n\times k}$ and $P = [p_1, \ldots, p_k] \in \mathbb{R}^{n\times k}$ be two orthonormal matrices, which span two k dimensional subspaces $\mathcal{Q} = \mathrm{span}\{Q\}$ and $\mathcal{P} = \mathrm{span}\{P\}$. Then*

$$\sin\angle(\mathcal{Q},\mathcal{P}) \le \sqrt{k}\max_{i=1,\ldots,k}\sin\angle(q_i,\mathcal{P}).$$

Proof. Let $\bar{q}$ be such that (see section 2.2 and [90])

$$\sin\angle(\bar{q},\mathcal{P}) = \max_{q\in\mathcal{Q}}\sin\angle(q,\mathcal{P}) = \sin\angle(\mathcal{Q},\mathcal{P}),$$

and let $\|\bar{q}\|_2 = 1$ with $\bar{q} = \sum_{j=1}^{k}\beta_j q_j$. Then for the orthogonal projection $P_{\mathcal{P}}$ on $\mathcal{P}$ we have

$$\begin{aligned}
\sin\angle(\bar{q},\mathcal{P}) &= \|(I-P_{\mathcal{P}})\bar{q}\|_2 = \|\sum_{j=1}^{k}\beta_j(I-P_{\mathcal{P}})q_j\|_2 \le \\
&\le \sum_{j=1}^{k}|\beta_j|\|(I-P_{\mathcal{P}})q_j\|_2 \le \max_{i=1,\ldots,k}\|(I-P_{\mathcal{P}})q_i\|_2\sum_{j=1}^{k}|\beta_j| \le \\
&\le \max_{i=1,\ldots,k}\|(I-P_{\mathcal{P}})q_i\|_2\sqrt{k}\sqrt{\sum_{j=1}^{k}\beta_j^2} = \\
&= \sqrt{k}\max_{i=1,\ldots,k}\sin\angle(q_i,\mathcal{P}).
\end{aligned}$$

□

Theorem 5.2.3. *Let $A \in \mathbb{R}^{n\times n}$ be a symmetric positive definite matrix with eigenvalues $\lambda_1 \le \cdots \le \lambda_n$ and eigenvectors $u_1, \ldots, u_n$, and let the eigenvalues of $A^{(m)} = (W^{(n,m)})^T A W^{(n,m)}$ be denoted by μ_j. Let $X_0 \in \mathbb{R}^{n\times k}$, $X_0^T X_0 = I_k$ be a starting approximation for multispace and $U = [u_1, \ldots, u_k]$. Let π_ℓ be an orthogonal projection on $\mathcal{W}^{(n,m)} = \mathcal{K}_\ell(A, X_p)$, $m = k\cdot\ell$. Let us assume that the vectors $\pi_0 u_1$, $\pi_0 u_2$, $\ldots$, $\pi_0 u_k$ are independent and $\lambda_k < \lambda_{k+1}$. Set*

$$\gamma_i = 1 - 2\frac{\lambda_n - \lambda_i}{\lambda_n - \lambda_{k+1}},$$

and let $T_{\ell-1}(x)$ be a Chebyshev polynomial of the first kind of degree $\ell - 1$. Then, for $i = 1, \ldots, k$

(a) when all of $2p$ inverse iterations are performed before the transfer to the subspace

$$\tan\angle(u_i,\mathcal{K}_\ell(A,X_{2p})) \le \frac{1}{|T_{\ell-1}(\gamma_i)|}\left(\frac{\lambda_i}{\lambda_{k+1}}\right)^{2p}\tan\angle(\mathcal{U},\mathcal{X}_0), \tag{5.8}$$

$$\sin\angle(u_i,\bar{\mathcal{X}}_{2p}) \le \frac{\left(1+\frac{r_\ell^2}{d_i^2}\right)^{\frac{1}{2}}}{|T_{\ell-1}(\gamma_i)|}\left(\frac{\lambda_i}{\lambda_{k+1}}\right)^{2p}\tan\angle(\mathcal{U},\mathcal{X}_0), \tag{5.9}$$

where $d_i = \min_{j\neq i,\ldots,i+n_i-1}|\lambda_i - \mu_j|$ if λ_i is of multiplicity n_i, and $r_\ell = \|(I - \pi_\ell)A\pi_\ell\|_2$.

(b) when p inverse iterations are performed before and p after the transfer to the subspace (as in Algorithm 5.1.3)

$$\tan\angle(u_i, \mathcal{X}_{2p}) \leq C \left(\frac{\lambda_i}{\lambda_{k+1}}\right)^p \left(\frac{\lambda_k}{\lambda_{k+1}}\right)^p \tan\angle(\mathcal{U}, \mathcal{X}_0), \tag{5.10}$$

where

$$C = \frac{\max_{i=1,\ldots,k} \left\{ \frac{\sqrt{k}\left(1+\frac{r_\ell^2}{d_i^2}\right)^{\frac{1}{2}}}{|T_{\ell-1}(\gamma_i)|} \right\}}{\sqrt{1 - \max_{i=1,\ldots,k} \left\{ \frac{k\left(1+\frac{r_\ell^2}{d_i^2}\right)}{T_{\ell-1}^2(\gamma_i)} \left(\frac{\lambda_i}{\lambda_{k+1}}\right)^{2p} \right\} \tan^2\angle(\mathcal{U}, \mathcal{X}_0)}},$$

if the denominator of C is greater than zero.

Proof. In this proof we use the result on angles between subspaces from Wedin [90]. Let $\bar{x}_i$ be the vector from Lemma 5.2.1, and let

$$\bar{x}_i = \sum_{j=1}^{n} \alpha_j u_j,$$

be its coordinates in the eigenbasis. Since condition (5.7) must be satisfied, it follows

$$\bar{x}_i = u_i + \sum_{j=k+1}^{n} \alpha_j u_j.$$

(a) The process is following:

$$\mathcal{X}_0 \xrightarrow{\text{inverse iteration}} \mathcal{X}_{2p} = A^{-2p}\mathcal{X}_0 \xrightarrow{\text{block Lanczos}} \mathcal{K}_\ell(A, X_{2p}) = A^{-2p}\mathcal{K}_\ell(A, X_0)$$
$$\xrightarrow{\text{solution on subspace}} \bar{\mathcal{X}}_{2p},$$

so, in this case it is $\mathcal{W}^{(n,m)} = \mathcal{K}_\ell(A, X_{2p})$. Let us consider an element $x \in A^{-2p}\mathcal{K}_\ell(A, X_0)$ of the form

$$x = A^{-2p} q(A) \bar{x}_i, \quad q \in \mathbb{P}_{\ell-1},$$

where $\mathbb{P}_{\ell-1}$ is a set of polynomials of degree not exceeding $\ell - 1$. Then

$$x = \lambda_i^{-2p} q(\lambda_i) u_i + \sum_{j=k+1}^{n} \alpha_j \lambda_j^{-2p} q(\lambda_j) u_j.$$

Let P_i denote the eigenprojection associated with λ_i. Then

$$\frac{\|(I - P_i)x\|_2^2}{\|P_i x\|_2^2} = \frac{\sum_{j=k+1}^{n} \lambda_j^{-4p} q^2(\lambda_j)\alpha_j^2}{\lambda_i^{-4p} q^2(\lambda_i)} = \sum_{j=k+1}^{n} \left(\frac{\lambda_i}{\lambda_j}\right)^{4p} \frac{q^2(\lambda_j)\alpha_j^2}{q^2(\lambda_i)}. \tag{5.11}$$

Let $\bar{q} \in \mathbb{P}_{\ell-1}$ be the polynomial for which the right-hand side of (5.11) reaches its minimum, and let $\bar{x} \in A^{-2p}\mathcal{K}_\ell(A, X_0)$ be the corresponding vector, such that $\bar{x} = A^{-2p}\bar{q}(A)\bar{x}_i$. Let $T_{\ell-1} \in \mathbb{P}_{\ell-1}$ be a Chebyshev polynomial of the first kind, then $T_{\ell-1}$ satisfies the following conditions [78, p.332]:

- $T_{\ell-1}(\lambda)/2^{\ell-2}$ has the smallest infinity norm on $[-1,1]$ of all monic polynomials of degree $\ell-1$, where infinity norm is defined by $\|q\|_{\infty,[-1,1]} = \max_{\lambda\in[-1,1]} |q(\lambda)|$.
- $\|T_{\ell-1}\|_{\infty,[-1,1]} = 1$.
- Of all polynomials q of degree $\leq \ell-1$ which satisfy $q(\gamma) = \delta$ for some $|\gamma| > 1$ the polynomials with the smallest infinity norm on $[-1,1]$ is

$$\hat{q}(\lambda) = \frac{\delta}{T_{\ell-1}(\gamma)} T_{\ell-1}(\lambda),$$

and

$$\|\hat{q}\|_{\infty,[-1,1]} = \frac{|\delta|}{|T_{\ell-1}(\gamma)|}.$$

Since the observed properties of the Chebyshev polynomials hold for $[-1,1]$, and the polynomials q in (5.11) acts on λ_j for $j = k+1,\ldots,n$, we have to map $[\lambda_{k+1},\lambda_n]$ onto $[-1,1]$. We can do that by an affine function $f : [\lambda_{k+1},\lambda_n] \to [-1,1]$ defined by $f(x) = ax - b$, where

$$a = \frac{2}{\lambda_n - \lambda_{k+1}}, \qquad b = \frac{\lambda_{k+1}+\lambda_n}{\lambda_n - \lambda_{k+1}}, \tag{5.12}$$

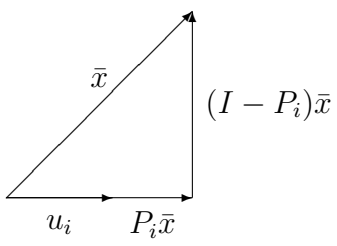
and then we can observe the polynomial $T_{\ell-1}(a\lambda - b)$. From the minimum property of $\bar{q}$, and the fact that $|T_{\ell-1}(\lambda)| \leq 1$ for $\lambda \in [-1,1]$ it follows that

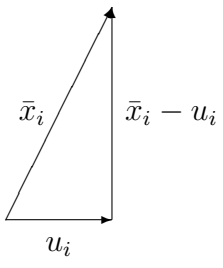
$$\begin{aligned}
\frac{\|(I-P_i)\bar{x}\|_2^2}{\|P_i\bar{x}\|_2^2} &= \sum_{j=k+1}^{n} \left(\frac{\lambda_i}{\lambda_j}\right)^{4p} \frac{\bar{q}^2(\lambda_j)\alpha_j^2}{\bar{q}^2(\lambda_i)} \leq \\
&\leq \left(\frac{\lambda_i}{\lambda_{k+1}}\right)^{4p} \sum_{j=k+1}^{n} \frac{T_{\ell-1}^2(a\lambda_j - b)\alpha_j^2}{T_{\ell-1}^2(a\lambda_i - b)} \leq \\
&\leq \frac{1}{T_{\ell-1}^2(\gamma_i)} \left(\frac{\lambda_i}{\lambda_{k+1}}\right)^{4p} \sum_{j=k+1}^{n} \alpha_j^2 = \\
&= \frac{1}{T_{\ell-1}^2(\gamma_i)} \left(\frac{\lambda_i}{\lambda_{k+1}}\right)^{4p} \|\bar{x}_i - u_i\|_2^2,
\end{aligned}$$

where

$$\gamma_i = a\lambda_i - b = 1 - 2\frac{\lambda_n - \lambda_i}{\lambda_n - \lambda_{k+1}}.$$

Now we have the following situation:





So, we can conclude that

$$\begin{aligned}\frac{\|(I-P_i)\bar{x}\|_2^2}{\|P_i\bar{x}\|_2^2} &= \tan\angle(u_i,\bar{x}) \geq \tan\angle(u_i,\mathcal{K}_\ell(A,X_{2p}))\\ \|\bar{x}_i-u_i\|_2 &= \tan\angle(\bar{x}_i,u_i) = \tan\angle(\bar{x}_i,\mathcal{U}) \leq \tan\angle(\mathcal{U},\mathcal{X}_0),\end{aligned} \tag{5.13}$$

and this proves (5.8).

Proof of (5.9) follows directly from Theorem 3 in [85, p. 694] and from the comment in [85, p. 703].

Let us assume that the eigenvalue λ_i is of multiplicity n_i, so that $i+n_i-1 \leq k$, and let P_{μ_j}, $j=1,\ldots,k$ denote eigenprojections whose range are the Ritz vectors associated with μ_j. We also assume that $\mu_1,\ldots,\mu_s$ are all distinct eigenvalues of $A^{(m)}$. Then we first want to prove the inequality

$$\left\|\left(\pi_\ell - \sum_{j=i}^{i+n_i-1} P_{\mu_j}\right)u_i\right\|_2 \leq \frac{r_\ell}{d_i}\|(I-\pi_\ell)u_i\|_2, \tag{5.14}$$

where u_i is any eigenvector associated with λ_i, and π_ℓ is an orthogonal projection on $\mathcal{W}^{(n,m)} = \mathcal{K}_\ell(A,X_{2p})$. The projections P_{μ_j} satisfy the following condition

$$\begin{aligned}&P_{\mu_j}P_{\mu_i} = \delta_{ij}P_{\mu_j},\\ &\sum_{t=1}^{s} P_{\mu_t} = \pi_\ell,\end{aligned} \tag{5.15}$$

and hence

$$(\pi_\ell A-\lambda_i I)\pi_\ell u_i = (\pi_\ell A-\lambda_i I)\sum_{t=1}^{s} P_{\mu_t}u_i = \sum_{t=1}^{s}(\mu_t-\lambda_i)P_{\mu_t}u_i.$$

Multiplying the two sides by $I-\sum_{j=i}^{i+n_i-1} P_{\mu_j}$ we obtain

$$\begin{aligned}\left(I-\sum_{j=i}^{i+n_i-1}P_{\mu_j}\right)(\pi_\ell A-\lambda_i I)\pi_\ell u_i &= \sum_{t=1}^{s}(\mu_t-\lambda_i)\left(I-\sum_{j=i}^{i+n_i-1}P_{\mu_j}\right)P_{\mu_t}u_i =\\ &= \sum_{t\neq i,\ldots,i+n_i-1}(\mu_t-\lambda_i)P_{\mu_t}u_i.\end{aligned} \tag{5.16}$$

Taking the norms of the two sides of equation (5.16) gives

$$\begin{aligned}\left\|\left(I-\sum_{j=i}^{i+n_i-1}P_{\mu_j}\right)(\pi_\ell A-\lambda_i I)\pi_\ell u_i\right\|_2^2 &= \sum_{t\neq i,\ldots,i+n_i-1}(\mu_t-\lambda_i)^2\|P_{\mu_t}u_i\|_2^2 \geq\\ &\geq d_i^2\sum_{t\neq i,\ldots,i+n_i-1}\|P_{\mu_t}u_i\|_2^2 =\\ &= d_i^2\left\|\left(\pi_\ell-\sum_{j=i}^{i+n_i-1}P_{\mu_j}\right)u_i\right\|_2^2,\end{aligned} \tag{5.17}$$

where $d_i = \min_{t \neq i,\ldots,i+n_i-1} |\lambda_i - \mu_t|$, and the last inequality follows from (5.15). For the left side of (5.16) we get

$$\begin{aligned}
\left\| \left(I - \sum_{j=i}^{i+n_i-1} P_{\mu_j} \right) (\pi_\ell A - \lambda_i I)\pi_\ell u_i \right\|_2^2 &\leq \left\| I - \sum_{j=i}^{i+n_i-1} P_{\mu_j} \right\|_2^2 \|\pi_\ell (A - \lambda_i I)\pi_\ell u_i\|_2^2 = \\
&= \|\pi_\ell (A - \lambda_i I)[u_i - (I - \pi_\ell)u_i]\|_2^2 = \\
&= \|\pi_\ell (A - \lambda_i I)(I - \pi_\ell)(I - \pi_\ell)u_i\|_2^2 \leq \\
&\leq \|\pi_\ell (A - \lambda_i I)(I - \pi_\ell)\|_2^2 \|(I - \pi_\ell)u_i\|_2^2 = \\
&= r_\ell^2 \|(I - \pi_\ell)u_i\|_2^2. \qquad (5.18)
\end{aligned}$$

Now (5.14) follows from (5.17) and (5.18). Next, we observe the decomposition

$$\left(I - \sum_{j=i}^{i+n_i-1} P_{\mu_j} \right) u_i = (I - \pi_\ell)u_i + \left(\pi_\ell - \sum_{j=i}^{i+n_i-1} P_{\mu_j} \right) u_i,$$

where the two vectors in the right side are orthogonal. Thus, from (5.14) it follows that

$$\begin{aligned}
\left\| \left(I - \sum_{j=i}^{i+n_i-1} P_{\mu_j} \right) u_i \right\|_2 &= \left(\|(I - \pi_\ell)u_i\|_2^2 + \left\| \left(\pi_\ell - \sum_{j=i}^{i+n_i-1} P_{\mu_j} \right) u_i \right\|_2^2 \right)^{\frac{1}{2}} \leq \\
&\leq \left(1 + \frac{r_\ell^2}{d_i^2} \right)^{\frac{1}{2}} \|(I - \pi_\ell)u_i\|_2. \qquad (5.19)
\end{aligned}$$

Let $y_i = \sum_{j=i}^{i+n_i-1} P_{\mu_j} u_i$ be a Ritz vector, then

$$\begin{aligned}
\left\| \left(I - \sum_{j=i}^{i+n_i-1} P_{\mu_j} \right) u_i \right\|_2 &= \sin\angle(u_i, y_i) \geq \sin\angle(u_i, \bar{\mathcal{X}}_{2p}), \\
\|(I - \pi_\ell)u_i\|_2 &= \sin\angle(u_i, \mathcal{K}_\ell(A, X_{2p})) \leq \tan\angle(u_i, \mathcal{K}_\ell(A, X_{2p})),
\end{aligned}$$

and (5.9) follows from (5.19) and (5.8).

(b) In this case, the process is as follows:

$$\mathcal{X}_0 \xrightarrow{\text{inverse iteration}} \mathcal{X}_p = A^{-p}\mathcal{X}_0 \xrightarrow{\text{block Lanczos}} \mathcal{K}_\ell(A, X_p) = A^{-p}\mathcal{K}_\ell(A, X_0)$$

$$\xrightarrow{\text{solution on subspace}} \bar{\mathcal{X}}_p \xrightarrow{\text{inverse iteration}} \mathcal{X}_{2p} = A^{-p}\bar{\mathcal{X}}_p.$$

From part (a) of this proof it follows

$$\sin\angle(u_i, \bar{\mathcal{X}}_p) \leq \frac{\left(1 + \frac{r_\ell^2}{d_i^2}\right)^{\frac{1}{2}}}{|T_{\ell-1}(\gamma_i)|} \left(\frac{\lambda_i}{\lambda_{k+1}} \right)^p \tan\angle(\mathcal{U}, \mathcal{X}_0).$$

On the other hand from Theorem 4.4.8 it follows that

$$\tan\angle(u_i, A^{-p}\bar{\mathcal{X}}_p) \leq \left(\frac{\lambda_i}{\lambda_{k+1}}\right)^p \tan\angle(\mathcal{U}, \bar{\mathcal{X}}_p).$$

So we have bounds on $\sin\angle(u_i, \bar{\mathcal{X}}_p)$, and we need a bound on $\tan\angle(\mathcal{U}, \mathcal{X}_0)$. For that, we will use the result of Lemma 5.2.2, to obtain

$$\sin\angle(U, \bar{\mathcal{X}}_p) \leq \max_{i=1,\ldots,k} \left[\frac{\sqrt{k}\left(1+\frac{r_\ell^2}{d_i^2}\right)^{\frac{1}{2}}}{|T_{\ell-1}(\gamma_i)|}\left(\frac{\lambda_i}{\lambda_{k+1}}\right)^p\right] \tan\angle(\mathcal{U}, \mathcal{X}_0).$$

Further, it follows that

$$\begin{aligned}
\tan\angle(u_i, A^{-p}\bar{\mathcal{X}}_p) &\leq \left(\frac{\lambda_i}{\lambda_{k+1}}\right)^p \frac{\sin\angle(\mathcal{U}, \bar{\mathcal{X}}_p)}{\sqrt{1-\sin^2\angle(\mathcal{U}, \bar{\mathcal{X}}_p)}} \leq \\
&\leq \left(\frac{\lambda_i}{\lambda_{k+1}}\right)^p \frac{\max_{i=1,\ldots,k}\left\{\frac{\sqrt{k}\left(1+\frac{r_\ell^2}{d_i^2}\right)^{\frac{1}{2}}}{|T_{\ell-1}(\gamma_i)|}\left(\frac{\lambda_i}{\lambda_{k+1}}\right)^p\right\}\tan\angle(\mathcal{U}, \mathcal{X}_0)}{\sqrt{1-\max_{i=1,\ldots,k}\left\{\frac{k\left(1+\frac{r_\ell^2}{d_i^2}\right)}{T_{\ell-1}^2(\gamma_i)}\left(\frac{\lambda_i}{\lambda_{k+1}}\right)^{2p}\right\}\tan^2\angle(\mathcal{U}, \mathcal{X}_0)}} \leq \\
&\leq \frac{\max_{i=1,\ldots,k}\left\{\frac{\sqrt{k}\left(1+\frac{r_\ell^2}{d_i^2}\right)^{\frac{1}{2}}}{|T_{\ell-1}(\gamma_i)|}\right\}}{\sqrt{1-\max_{i=1,\ldots,k}\left\{\frac{k\left(1+\frac{r_\ell^2}{d_i^2}\right)}{T_{\ell-1}^2(\gamma_i)}\left(\frac{\lambda_i}{\lambda_{k+1}}\right)^{2p}\right\}\tan^2\angle(\mathcal{U}, \mathcal{X}_0)}} \cdot \\
&\quad \cdot \left(\frac{\lambda_i}{\lambda_{k+1}}\right)^p \left(\frac{\lambda_k}{\lambda_{k+1}}\right)^p \tan\angle(\mathcal{U}, \mathcal{X}_0).
\end{aligned}$$

□

Remark 5.2.4. *The process described in Algorithm 5.1.3 has weaker convergence bound than the process described in part (a) of Theorem 5.2.3. On the other hand, when multispace is performed in floating point arithmetic, case (a) can produce a solution which is far from being orthonormal, due to the Lanczos method. Case (b) will guarantee numerical orthonormality of the solution. The difference between the convergence of these two cases turned out to be negligible in many examples, and hence case (b) is preferable.*

It remains only to bound eigenvalue errors.

Corollary 5.2.5. *Let the assumptions of Theorem 5.2.3 be satisfied, and let* $A^{(m)} = (W^{(n,m)})^T A W^{(n,m)}$ *have eigenvalues* $\mu_1 \leq \cdots \leq \mu_m$. *Then, for* $i = 1, \ldots, k$, *it holds that*

(a) when all of 2p inverse iterations are performed before the transfer to the subspace

$$0 \leq \mu_i - \lambda_i \leq (\lambda_n - \lambda_i) \left[\frac{K_i}{|T_{\ell-i}(\gamma_i)|} \left(\frac{\lambda_i}{\lambda_{k+1}} \right)^{2p} \tan \angle(\mathcal{U}, \mathcal{X}_0) \right]^2, \tag{5.20}$$

where

$$K_i = \prod_{s=1}^{i-1} \frac{(\lambda_n - \lambda_s)}{(\lambda_i - \mu_s)},$$

and

$$\max_{i=1,\ldots,k} |\mu_i - \lambda_i| \leq (\lambda_n - \lambda_1)\sqrt{k} \max_{i=1,\ldots,k} \left(\frac{1 + \frac{r_\ell^2}{d_i^2}}{T_{\ell-1}^2(\gamma_i)} \right) \left[\left(\frac{\lambda_k}{\lambda_{k+1}} \right)^{2p} \tan \angle(\mathcal{U}, \mathcal{X}_0) \right]^2, \tag{5.21}$$

(b) when p inverse iterations are performed before, and p after the transfer to the subspace (as in Algorithm 5.1.3), we define $\{\nu_1, \ldots, \nu_k\}$ to be eigenvalues of $X_{2p}^T A X_{2p}$, then

$$\max_{i=1,\ldots,k} |\nu_i - \lambda_i| \leq (\lambda_n - \lambda_1)\sqrt{k} \left[C \left(\frac{\lambda_k}{\lambda_{k+1}} \right)^{2p} \tan \angle(\mathcal{U}, \mathcal{X}_0) \right]^2, \tag{5.22}$$

where C is defined in Theorem 5.2.3.

Proof. The proof of this corollary is based on Theorem 6 in [85, p. 702]. Let $T = [t_1, \ldots, t_m]$ be the eigenvectors of $A^{(m)}$ and let $V = [v_1, \ldots, v_n] = W^{(n,m)}T$ be the corresponding Ritz vectors. A theorem from [78, p. 190], which is a consequence of the Courant–Fischer Minimax Theorem 4.1.7, states that

$$\begin{aligned} \mu_i &= \min_{\substack{y \in \mathbb{R}^m \\ y \perp t_j,\ j=1,\ldots,i-1}} \frac{y^T A^{(m)} y}{y^T y} = \min_{\substack{y \in \mathbb{R}^m \\ y \perp t_j,\ j=1,\ldots,i-1}} \frac{(W^{(n,m)}y)^T A W^{(n,m)} y}{(W^{(n,m)}y)^T W^{(n,m)} y} = \\ &= \min_{\substack{w \in \mathcal{W}^{(n,m)} \\ w \perp v_j,\ j=1,\ldots,i-1}} \frac{w^T A w}{w^T w}. \end{aligned}$$

Since for case (a) $\mathcal{W}^{(n,m)} = \mathcal{K}_\ell(A, X_{2p}) = A^{-2p}\mathcal{K}_\ell(A, X_0)$, any $w \in \mathcal{W}^{(n,m)}$ can be represented as $w = A^{-2p}q(A)x$ where $q \in \mathbb{P}_{\ell-1}$, and $x \in \mathcal{X}_0$. Let us take

$$\begin{aligned} q_i(x) &= \prod_{s=1}^{i-1} (x - \mu_s) T_{\ell-i}(ax - b), \\ w_i &= A^{-2p} q_i(A) \bar{x}_i = \lambda_i^{-2p} q_i(\lambda_i) u_i + \sum_{j=k+1}^{n} \alpha_j \lambda_j^{-2p} q_i(\lambda_j) u_j, \end{aligned}$$

where a, b and $T_{\ell-i}$ are defined in the proof of Theorem 5.2.3, and $\bar{x}_i$ is defined in Lemma 5.2.1. Then,

$$w_i = (A-\mu_s I)z_s \quad \text{for} \quad z_s = A^{-2p}\prod_{\substack{t=1\\t\neq s}}^{i-1}(A-\mu_t I)T_{\ell-i}(aA-bI)\bar{x}_i \in \mathcal{W}^{(n,m)},\ s=1,\ldots,i-1,$$

and it implies that

$$w_i^T v_s = z_s^T(A-\mu_s I)v_s = 0, \qquad s=1,\ldots,i-1,$$

since $(A-\mu_s I)v_s \perp \mathcal{W}^{(n,m)}$. Hence

$$\mu_i \leq \frac{w_i^T A w_i}{w_i^T w_i},$$

and for $i=1,\ldots,k$

$$\begin{aligned}
0 \leq \mu_i - \lambda_i &\leq \frac{w_i^T(Aw_i-\lambda_i w_i)}{w_i^T w_i} = \\
&= \frac{(\lambda_i^{-2p}q_i(\lambda_i)u_i + \sum_{j=k+1}^n \alpha_j\lambda_j^{-2p}q_i(\lambda_j)u_j)^T\left(\sum_{j=k+1}^n \alpha_j(\lambda_j-\lambda_i)\lambda_j^{-2p}q_i(\lambda_j)u_j\right)}{\lambda_i^{-4p}q_i^2(\lambda_i) + \sum_{j=k+1}^n \alpha_j^2\lambda_j^{-4p}q_i^2(\lambda_j)} \leq \\
&\leq \frac{\sum_{j=k+1}^n(\lambda_j-\lambda_i)\alpha_j^2\lambda_j^{-4p}q_i^2(\lambda_j)}{\lambda_i^{-4p}q_i^2(\lambda_i)} \leq (\lambda_n-\lambda_i)\frac{\sum_{j=k+1}^n \alpha_j^2\lambda_j^{-4p}q_i^2(\lambda_j)}{\lambda_i^{-4p}q_i^2(\lambda_i)}.
\end{aligned}$$

By inserting the definition of q_i we obtain

$$\begin{aligned}
\mu_i - \lambda_i &\leq (\lambda_n-\lambda_i)\frac{\sum_{j=k+1}^n \alpha_j^2\lambda_j^{-4p}\prod_{s=1}^{i-1}(\lambda_j-\mu_s)^2T_{\ell-i}^2(a\lambda_j-b)}{\lambda_i^{-4p}\prod_{s=1}^{i-1}(\lambda_i-\mu_s)^2T_{\ell-i}^2(a\lambda_i-b)} \leq \\
&\leq (\lambda_n-\lambda_i)\left(\frac{\lambda_i}{\lambda_{k+1}}\right)^{4p}\prod_{s=1}^{i-1}\frac{(\lambda_n-\lambda_s)^2}{(\lambda_i-\mu_s)^2}\sum_{j=k+1}^n\frac{\alpha_j^2T_{\ell-i}^2(a\lambda_j-b)}{T_{\ell-i}^2(a\lambda_i-b)} \leq \\
&\leq (\lambda_n-\lambda_i)\left(\frac{\lambda_i}{\lambda_{k+1}}\right)^{4p}\prod_{s=1}^{i-1}\frac{(\lambda_n-\lambda_s)^2}{(\lambda_i-\mu_s)^2}\frac{1}{T_{\ell-i}^2(\gamma_i)}\sum_{j=k+1}^n\alpha_j^2 = \\
&= (\lambda_n-\lambda_i)\frac{K_i^2}{T_{\ell-i}^2(\gamma_i)}\left(\frac{\lambda_i}{\lambda_{k+1}}\right)^{4p}\|\bar{x}_i-u_i\|_2^2
\end{aligned}$$

The proof of (5.20) follows from (5.13).

(5.21) and (5.22) follow from Theorem 3 [65, pp 254], which claims that

(a) $\max_{i=1,\ldots,k}|\mu_i-\lambda_i| \leq (\lambda_n-\lambda_1)\sin^2\angle(\mathcal{U},\bar{\mathcal{X}}_{2p})$,

(b) $\max_{i=1,\ldots,k}|\nu_i-\lambda_i| \leq (\lambda_n-\lambda_1)\sin^2\angle(\mathcal{U},\mathcal{X}_{2p})$.

Final equations are straightforward consequences of Theorem 5.2.3 and Lemma 5.2.2. □

5.3 Numerical Examples

In these examples we are testing the functionality of the multispace approach. We want to illustrate the convergence of multispace, and to demonstrate the examples where multispace do speed up the inverse iteration.

Example 5.3.1. *Let us first consider the difference between case (a) and case (b) in Theorem 5.2.3. We will observe a two-space process, where the partial eigenvalue problem is solved exactly on the subspace. Moreover, we reorthogonalize the basis returned from the block Lanczos algorithm, in order to simulate the situation in exact arithmetic. We will take a symmetric matrix* $A \in \mathbb{R}^{100\times 100}$ *to have fixed eigenvalues* $\{1, 2, \ldots, 100\}$*. Since we are minimizing (5.4), we can note that the minimal value of* $\rho(X)$ *is*

$$\min_{\substack{X\in\mathbb{R}^{n\times k}\\ X^TX=I_k}} \rho(X) = 1+2+3+4+5 = 15,$$

for $k = 5$*. We performed the following tests:*

method	$\rho(X_{2p})$
multispace with 5 inverse iterations before and 5 inverse iterations after the subspace transfer	15.00001854142704
multispace with 10 inverse iterations before subspace transfer	15.00000000041437
10 inverse iterations	15.15549917

From these results we can see that multispace increased the accuracy of inverse iteration, and that case (a) is more accurate than case (b) when the basis is reorthogonalized. In floating point arithmetic, without the reorthogonalization of the Lanczos basis, the result of case (a) was very inaccurate.

Example 5.3.2. *In this example A is taken to be the block tridiagonal matrix of order 1024 resulting from discretizing Poisson's equation with the 5-point operator on an* 32×32 *mesh. We are searching for the 6 smallest eigenvalues, where the eigenvalues are approximatively equal to*

$$\begin{aligned}
\lambda_1 &\approx 1.811230970764231 \cdot 10^{-2}\\
\lambda_2 &\approx 4.519876032840046 \cdot 10^{-2}\\
\lambda_3 &\approx 4.519876032844214 \cdot 10^{-2}\\
\lambda_4 &\approx 7.228521094917532 \cdot 10^{-2}\\
\lambda_5 &\approx 9.007020762483668 \cdot 10^{-2}\\
\lambda_6 &\approx 9.007020762485157 \cdot 10^{-2}\\
\lambda_7 &\approx 1.171566582456000 \cdot 10^{-1}
\end{aligned}$$

The convergence of the inverse iteration depends on $\lambda_6/\lambda_7 = 7.688014405125310 \cdot 10^{-1}$*, and the minimal trace is* $\sum_1^6 \lambda_i = 3.609354565633485 \cdot 10^{-1}$*. In Table 5.1 we compare* $\rho(X)$ *for the multispace and the inverse iteration, where in each multispace level 2 inverse iterations are performed before the subspace transfer and 2 after.*

Multispace:

number of V cycles	$\rho(X)$
0	$2.4 \cdot 10^{1}$
1	$3.933010161085040 \cdot 10^{-1}$
2	$3.610192107781051 \cdot 10^{-1}$
3	$3.609354751996430 \cdot 10^{-1}$
4	$3.609354565674766 \cdot 10^{-1}$

Inverse iteration:

number of iterations	$\rho(X)$
0	$2.4 \cdot 10^{1}$
4	$7.751530085839884 \cdot 10^{0}$
8	$1.413509702629963 \cdot 10^{0}$
12	$8.926254300321311 \cdot 10^{-1}$
16	$5.859145267902786 \cdot 10^{-1}$
78	$3.609354565675780 \cdot 10^{-1}$

Table 5.1: Reduction of $\rho(X)$.

From the last row in Table 5.1, we can see that the inverse iteration achieved approximatively the same value for $\rho(X)$ after 78 iterations, as the multispace for 4 V cycles, with a total of 16 inverse iterations on the first level.

Example 5.3.3. *The tests are performed with a matrix $A \in \mathbb{R}^{500\times 500}$ such that*

$$\begin{array}{lll} \lambda_1 = \sin\left(\frac{\pi}{10}\right) & \lambda_2 = \sin\left(\frac{2\pi}{10}\right) & \lambda_3 = \sin\left(\frac{3\pi}{10}\right) \\ \lambda_4 = \sin\left(\frac{4\pi}{10}\right) & \lambda_5 = \sin\left(\frac{4\pi}{10}\right) & \lambda_6 = 1 \\ \lambda_7 = 2 & \dots & \lambda_{500} = 495 \end{array}$$

The task was to determine the 5 smallest eigenvalues and corresponding eigenvectors. On the other hand, there is a possible source of problems for inverse iteration, because

$$\frac{\lambda_5}{\lambda_6} = 0.95105651629515.$$

We took trace error to be $|\rho(X_{2p}) - \sum_{i=1}^{5} \lambda_i|$. When multispace results are plotted versus the number of inverse iterations, this means that a total of $2p$ inverse iterations are performed on each level (as in the case of Algorithm 5.1.3). The tests were performed on the same computers as before, and the results are illustrated in the following figures.

For example from Figure 5.4 we can see that multispace with 4 inverse iterations on each level achieves the trace error equal to $2.4381 \cdot 10^{-7}$, while inverse iterations alone could not achieve that accuracy even after 100 steps. On the time scale, from Figure 5.9 we can see that this trace error was achieved after 0.13 seconds, while 100 inverse iterations had execution time equal to 0.31 seconds.

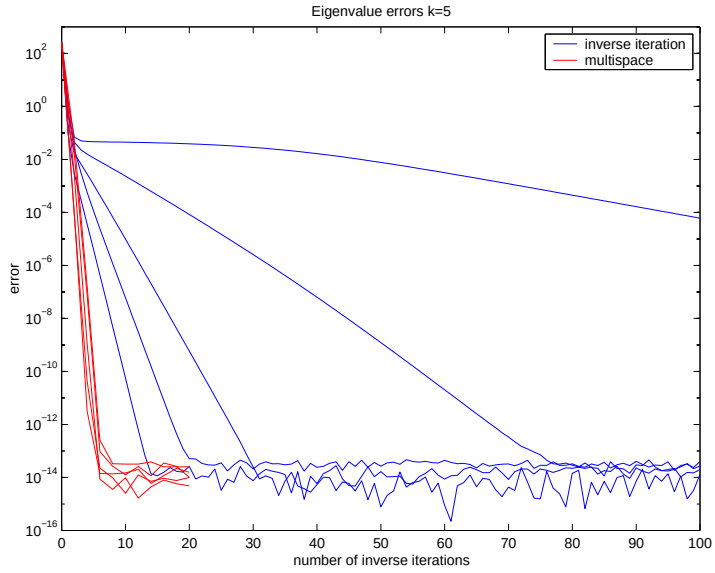


Figure 5.3: Eigenvalue errors versus the number of inverse iterations.

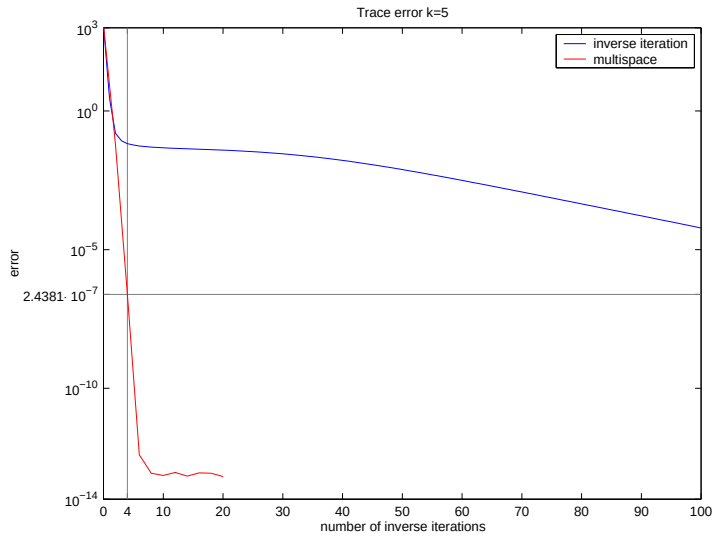


Figure 5.4: Trace errors versus the number of inverse iterations.

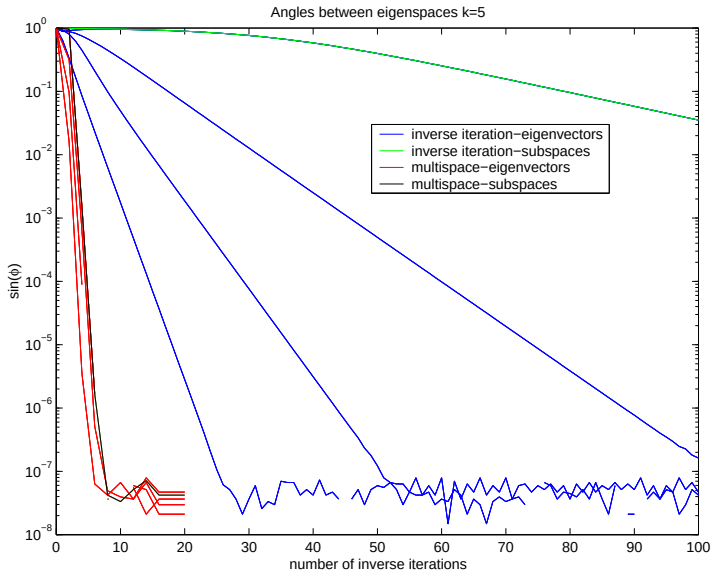


Figure 5.5: Angles between individual eigenvectors and their approximation, and angles between subspaces $\mathcal{U}$ and $\mathcal{X}_{2p}$, versus the number of inverse iterations.

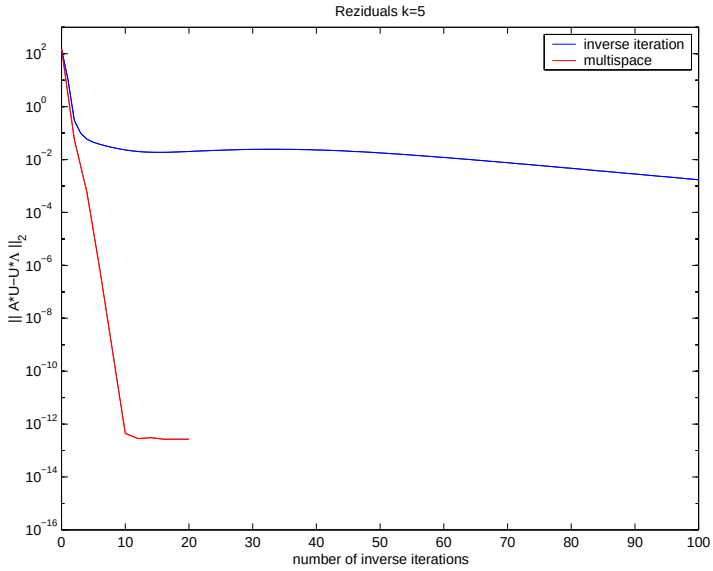


Figure 5.6: Residual norms versus the number of inverse iterations.

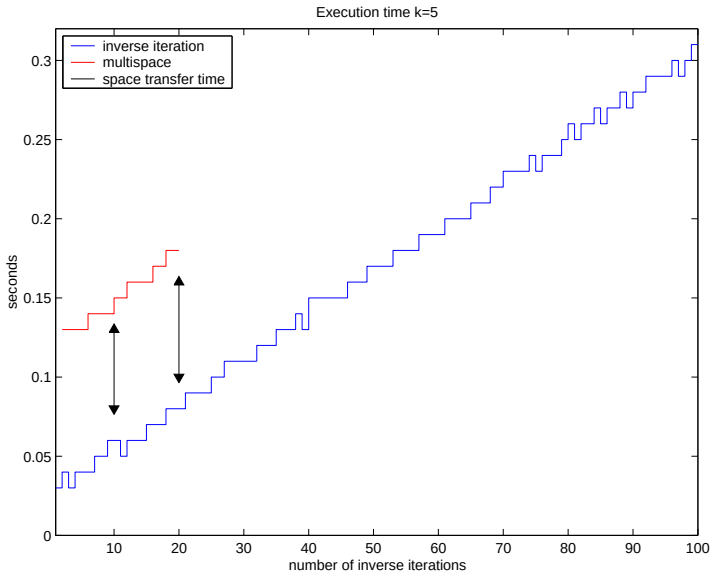


Figure 5.7: Execution times

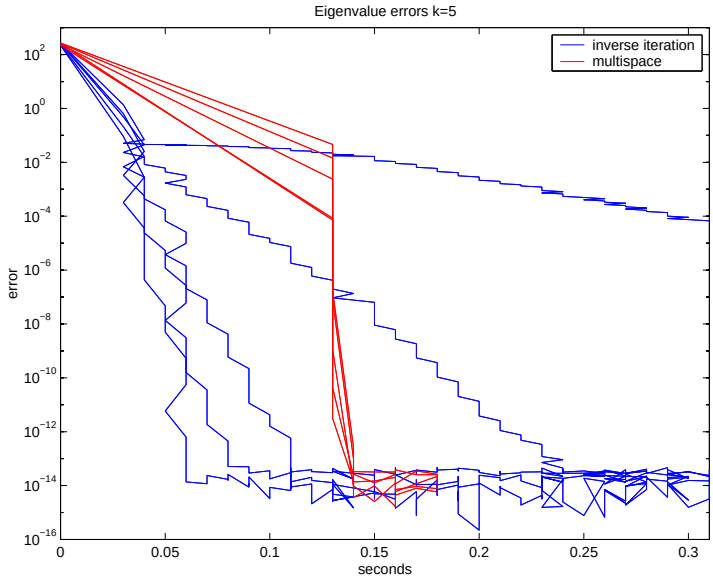


Figure 5.8: Eigenvalue errors versus time.

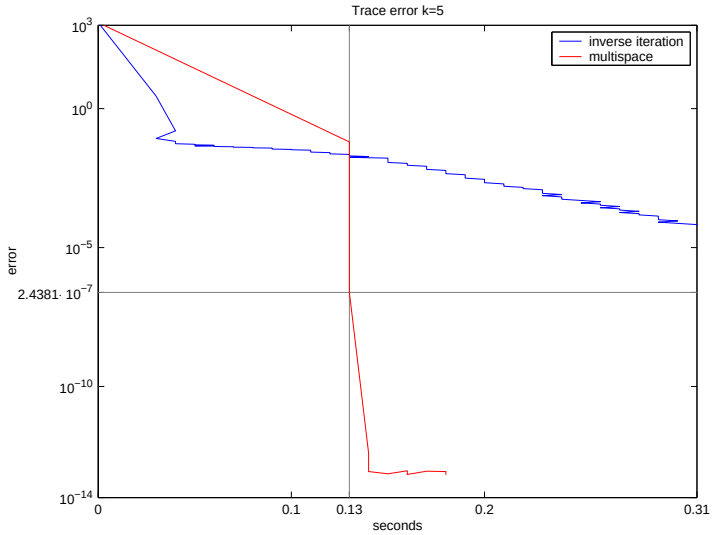


Figure 5.9: Trace errors versus time.

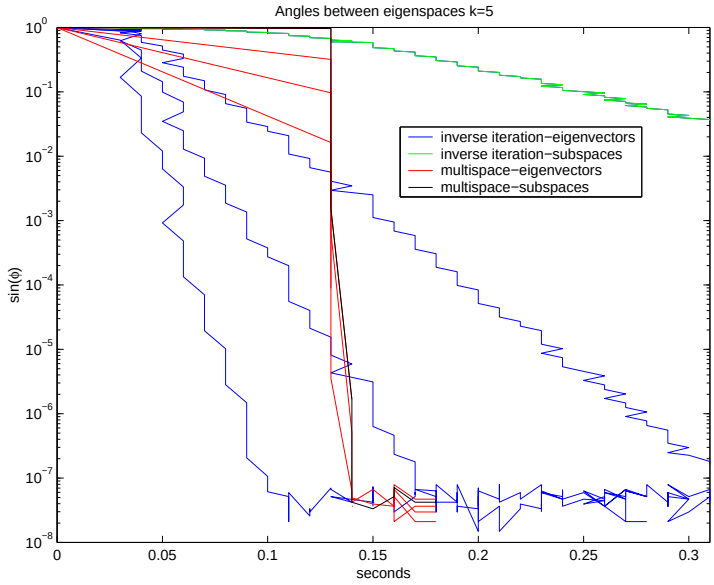


Figure 5.10: Angles between individual eigenvectors and their approximation, and angles between subspaces $\mathcal{U}$ and $\mathcal{X}_{2p}$, versus time.

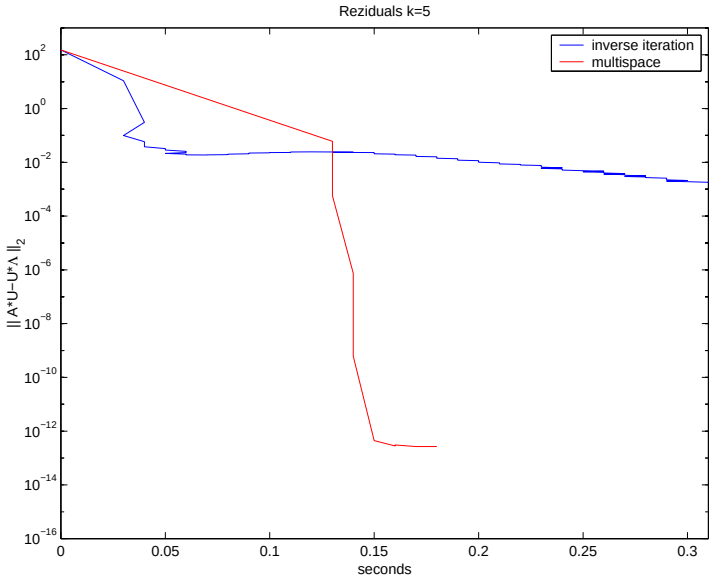


Figure 5.11: Residual norms versus time.

Chapter 6

New Bounds on Singular Value Approximations from Subspaces

In this chapter the new results on singular value approximations from subspaces are presented. We are considering singular value decomposition of a compression of the operator to low dimensional subspaces. Our main task is to find bounds on the relative error between singular values of the compression and the original operator. Since we are dealing with subspaces, we derive convenient bounds that involve angles between some specific subspaces, like it was presented by Drmač in [22], and by Drmač and Hari in [23] for the eigenvalues of a Hermitian matrix.

6.1 Classical Results

The classical bound on the absolute eigenvalue error of a Hermitian matrix is given by Kahan in the following theorem [56].

Theorem 6.1.1. *Let H be an $n \times n$ Hermitian matrix with eigenvalues $\lambda_1 \leq \cdots \leq \lambda_n$ and let X be an $n \times \ell$ matrix of full column rank. If M is any $\ell \times \ell$ Hermitian matrix, with eigenvalues $\mu_1 \leq \cdots \leq \mu_\ell$, then there are eigenvalues $\lambda_{i_1} \leq \cdots \leq \lambda_{i_\ell}$ of H such that*

$$\max_{j=1,\ldots,\ell} |\lambda_{i_j} - \mu_j| \leq \frac{\|R(M)\|_2}{\sigma_{\min}(X)}, \quad R(M) = HX - XM. \tag{6.1}$$

In (6.1) the absolute error in eigenvalues is bounded by the residual norm. The residual norm represents a measure of deviation of the subspace spanned by the columns of X from an invariant subspace. In practice, M is the Rayleigh quotient matrix $M = X^*HX$, where $X^*X = I_\ell$.

The above theorem can be generalized to the singular values of a rectangular $m \times n$ matrix (see Theorem 4.5 in [48]).

Theorem 6.1.2. *Let A be an $m \times n$ rectangular matrix, where $m \geq n$, with singular values $\sigma_1 \geq \cdots \geq \sigma_n$. Let X be an $m \times \ell$ orthonormal matrix, and let Y be an $n \times \ell$ orthonormal matrix. If G is any $\ell \times \ell$ matrix, with singular values $\gamma_1 \geq \cdots \geq \gamma_\ell$, then*

there are singular values $\sigma_{i_1} \geq \cdots \geq \sigma_{i_\ell}$ *of* A *such that*

$$\max_{j=1,\ldots,\ell} |\sigma_{i_j} - \gamma_j| \leq \max\{\|R_R(G)\|_2, \|R_L(G)\|_2\}, \tag{6.2}$$

where

$$R_R(G) = AY - XG, \qquad R_L(G) = A^*X - YG^*.$$

Proof. Let $[X\ X_\perp]$ represent an unitary basis for $\mathbb{C}^m$, and let $[Y\ Y_\perp]$ represent an unitary basis for $\mathbb{C}^n$. The representation of A in these two bases can be written as

$$A = \begin{bmatrix} X & X_\perp \end{bmatrix} \begin{bmatrix} M & L \\ K & N \end{bmatrix} \begin{bmatrix} Y^* \\ Y_\perp^* \end{bmatrix},$$

where

$$M = X^*AY, \quad L = X^*AY_\perp, \quad K = X_\perp^*AY, \quad N = X_\perp^*AY_\perp.$$

Further, $R_R(G) \in \mathbb{C}^{m\times\ell}$ and $R_L(G) \in \mathbb{C}^{n\times\ell}$, so they can be written as

$$R_R(G) = \begin{bmatrix} X & X_\perp \end{bmatrix} \begin{bmatrix} M-G \\ K \end{bmatrix}, \quad R_L(G) = \begin{bmatrix} Y & Y_\perp \end{bmatrix} \begin{bmatrix} M^*-G^* \\ L^* \end{bmatrix},$$

and

$$\left\| \begin{bmatrix} M-G \\ K \end{bmatrix} \right\|_2 = \|R_R(G)\|_2 \leq \nu, \quad \left\| \begin{bmatrix} M-G & L \end{bmatrix} \right\|_2 = \|R_L(G)\|_2 \leq \nu,$$

where

$$\nu = \max\{\|R_R(G)\|_2, \|R_L(G)\|_2\}.$$

The dilatation theorem from [12] states that there exists an $(m-\ell)\times(n-\ell)$ matrix T, such that

$$\left\| \begin{bmatrix} M-G & L \\ K & N-T \end{bmatrix} \right\|_2 \leq \nu, \tag{6.3}$$

with

$$N - T = -S(M-G)^*P + \nu(I-SS^*)^{\frac{1}{2}}C(I-P^*P)^{\frac{1}{2}},$$

where

$$S = K[\nu^2 I - (M-G)^*(M-G)]^{\dagger\frac{1}{2}}, \qquad P = [\nu^2 I - (M-G)(M-G)^*]^{\dagger\frac{1}{2}}L,$$

and C is an arbitrary contraction, i.e. a matrix satisfying $\|C\|_2 \leq 1$.

Now, if we define the matrix $\tilde{A}$ as

$$\tilde{A} = \begin{bmatrix} X & X_\perp \end{bmatrix} \begin{bmatrix} G & 0 \\ 0 & T \end{bmatrix} \begin{bmatrix} Y^* \\ Y_\perp^* \end{bmatrix},$$

then the singular values $\tilde{\sigma}_i$, $i = 1,\ldots n$ of $\tilde{A}$ include singular values of G, and by the Weyl inequalities (Theorem 2.3.1) and (6.3), the following holds

$$\max_{i=1,\ldots,n} |\sigma_i - \tilde{\sigma}_i| \leq \|A - \tilde{A}\|_2 = \left\| \begin{bmatrix} M-G & L \\ K & N-T \end{bmatrix} \right\|_2 \leq \nu.$$

□

The consequence of Theorem 6.1.2 with a tighter bound is stated in the following corollary.

Corollary 6.1.3. *Let A, X, Y, $R_R(G)$ and $R_L(G)$ be defined as in Theorem 6.1.2 but with $G = M = X^*AY$. Let us assume that $\sigma(M) = \{\mu_1, \dots, \mu_\ell\}$ and $\sigma(N) = \{\nu_1, \dots, \nu_{n-\ell}\}$ satisfy condition $\sigma(M) \cap \sigma(N) = \emptyset$, where μ_j and ν_k are the singular values of M and $N = X_\perp^* A Y_\perp$ respectively. Then, the following bound holds*

$$\max_{i=1,\dots,n} |\sigma_i - \tilde{\sigma}_i| \leq \frac{\max\{\|R_R(M)\|_2^2, \|R_L(M)\|_2^2\}}{\min\limits_{j=1,\dots,\ell \ ,k=1,\dots,n-\ell} |\mu_j - \nu_k|}, \tag{6.4}$$

where $\{\tilde{\sigma}_1, \dots, \tilde{\sigma}_n\} = \sigma(M) \cup \sigma(N)$.

Proof. $\tilde{\sigma}_i$, $i = 1, \dots, n$ are the singular values of the matrix $\tilde{A} = \begin{bmatrix} M & 0 \\ 0 & N \end{bmatrix}$, and from [70, p. 548], it follows that

$$\max_{i=1,\dots,n} |\sigma_i - \tilde{\sigma}_i| \leq \frac{\max\{\|L\|_2^2, \|K\|_2^2\}}{\min\limits_{j=1,\dots,\ell \ ,k=1,\dots,n-\ell} |\mu_j - \nu_k|}.$$

From the proof of Theorem 6.1.2 we can conclude that $\|R_R(M)\|_2 = \|K\|_2$ and $\|R_L(M)\|_2 = \|L\|_2$. □

Instead of the absolute error bound, we want to derive bound on relative error in the singular values. This usually involves multiplicative perturbations as we saw in subsection 2.3.1, so the technique that is used to obtain such a bound is different than the technique used in Kahan's theorem. In the next section the new result is presented, which includes relative error bounds between singular values of a rectangular matrix and the compression of the matrix to a low dimensional subspaces. The bounds are functions of angles between suitably chosen subspaces. In section 6.3 more tight quadratic bounds are introduced, which are again expressed in terms of the angles between the same subspaces.

6.2 A New Subspace Bound

To find the error bound between the singular values of a rectangular matrix $A \in \mathbb{C}^{m\times n}$ and its compression $M \in \mathbb{R}^{\ell\times\ell}$, first we have to construct an artificial additive perturbation to obtain a matrix $\tilde{A} \in \mathbb{C}^{m\times n}$ whose singular values include singular values of the compression. On the other hand, for the relative error bounds we need a multiplicative perturbation. So, we will find suitable bases for the spaces $\mathbb{C}^m$ and $\mathbb{C}^n$, and we will express the matrix A as a multiplicative perturbation of $\tilde{A}$. The singular value error bounds are then derived by means of the relative perturbation theory, see section 2.3 and [52].

From now on we will consider $m \times n$ rectangular matrices, where $m \geq n$, and we will denote $\mathcal{R}(B) = \text{range}(B)$ for any matrix B. Let $\mathcal{X}$ be an ℓ-dimensional subspace of $\mathbb{C}^m$

given as the range of orthonormal $X \in \mathbb{C}^{m\times\ell}$, such that $\mathcal{X} \cap \mathcal{R}(A)^\perp = \{0\}$, and let $\mathcal{Y}$ be an ℓ-dimensional subspace of $\mathbb{C}^n$ given as the range of orthonormal $Y \in \mathbb{C}^{n\times\ell}$. For matrices $A \in \mathbb{C}^{m\times n}$ and $M \in \mathbb{C}^{\ell\times\ell}$ we define two types of residuals as in Theorem 6.1.2

$$R_R(M) = AY - XM, \qquad R_L(M) = A^*X - YM^*.$$

Further, the subspaces of interest in the following theorems will be

$$\mathcal{W} = A^*\mathcal{X} \quad \text{and} \quad \mathcal{Z} = A^\dagger\mathcal{X},$$

and together with $\mathcal{Y}$ they will be involved in error bounds. In fact, the bounds are expressed in terms of angles between these subspaces, where the angle between two equidimensional subspaces is defined in subsection 2.2.6.

Finally, we can state the main result.

Theorem 6.2.1. *Let $A \in \mathbb{C}^{m\times n}$, and let A have the singular values $\sigma_1 \geq \cdots \geq \sigma_n > 0$. Further, let $M = X^*AY \in \mathbb{C}^{\ell\times\ell}$ be the Rayleigh quotient, and let residuals be defined as*

$$R_R = R_R(M) = (I - XX^*)AY, \qquad R_L = R_L(M) = (I - YY^*)A^*X.$$

We define an additive perturbation $\delta A = R_RY^* + XR_L^*$, and change A to $\tilde{A} = A - \delta A$, so the singular values of M are the singular values of $\tilde{A}$. If $\theta = \max\{\measuredangle(\mathcal{W},\mathcal{Y}), \measuredangle(\mathcal{Z},\mathcal{Y})\} < \pi/2$, then $\tilde{A}$ has also full column rank, and its singular values $\tilde{\sigma}_1 \geq \cdots \geq \tilde{\sigma}_n > 0$ satisfy

$$\max_{i=1,\ldots,n} \frac{|\sigma_i - \tilde{\sigma}_i|}{\tilde{\sigma}_i} \leq 2\tan\theta + \tan^2\theta. \tag{6.5}$$

Proof. Without loss of generality we can assume that $\mathcal{X} \subset \mathcal{R}(A)$. If this is not the case, then let us observe the subspace $\mathcal{X}_A = AA^\dagger\mathcal{X}$. $\mathcal{X}_A$ is the orthogonal projection of $\mathcal{X}$ on $\mathcal{R}(A)$, and since $\mathcal{X} \cap \mathcal{R}(A)^\perp = \{0\}$, $\mathcal{X}_A$ is of the same dimension as $\mathcal{X}$. Further, by the Moore–Penrose conditions for the pseudo–inverse [35, p. 243], we have

$$\begin{aligned} A^*\mathcal{X}_A &= A^*AA^\dagger\mathcal{X} = A^*(AA^\dagger)^*\mathcal{X} = (AA^\dagger A)^*\mathcal{X} = A^*\mathcal{X} = \mathcal{W} \\ A^\dagger\mathcal{X}_A &= A^\dagger AA^\dagger\mathcal{X} = A^\dagger\mathcal{X} = \mathcal{Z}. \end{aligned}$$

So, from now on we will take $\mathcal{X}_A$ instead of $\mathcal{X}$, or we will just assume that $\mathcal{X} \subset \mathcal{R}(A)$.

Let $X_\perp = [X_{1,\perp}, X_{2,\perp}]$ be an orthonormal basis for $\mathcal{X}^\perp$, where $X_{1,\perp} \in \mathbb{C}^{m\times(n-\ell)}$ and $X_{2,\perp} \in \mathbb{C}^{m\times(m-n)}$ are such that $\mathcal{R}(X_{1,\perp}) \subset \mathcal{R}(A)$ and $X_{2,\perp}^*A = 0$. Let $Y_\perp \in \mathbb{C}^{n\times(n-\ell)}$ be an orthonormal basis for $\mathcal{Y}^\perp$. Then, the matrix A can be written as

$$A = \begin{bmatrix} X & X_{1,\perp} & X_{2,\perp} \end{bmatrix} \begin{bmatrix} M & L \\ K & N \\ 0 & 0 \end{bmatrix} \begin{bmatrix} Y^* \\ Y_\perp^* \end{bmatrix},$$

where $L = X^*AY_\perp \in \mathbb{C}^{\ell\times(n-\ell)}$, $K = X_{1,\perp}^*AY \in \mathbb{C}^{(n-\ell)\times\ell}$ and $N = X_{1,\perp}^*AY_\perp \in \mathbb{C}^{(n-\ell)\times(n-\ell)}$. On the other hand, for $\tilde{A}$ we have

$$\begin{aligned} \tilde{A} &= A - (I - XX^*)AYY^* - XX^*A(I - YY^*) = \\ &= XX^*AYY^* + (I - XX^*)A(I - YY^*) = \\ &= XX^*AYY^* + X_\perp X_\perp^*AY_\perp Y_\perp^* = XX^*AYY^* + X_{1,\perp}X_{1,\perp}^*AY_\perp Y_\perp^* = \\ &= \begin{bmatrix} X & X_{1,\perp} & X_{2,\perp} \end{bmatrix} \begin{bmatrix} M & 0 \\ 0 & N \\ 0 & 0 \end{bmatrix} \begin{bmatrix} Y^* \\ Y_\perp^* \end{bmatrix}. \end{aligned}$$

So, A and $B = \begin{bmatrix} M & L \\ K & N \\ 0 & 0 \end{bmatrix}$ have the same singular values, and $\tilde{A}$ and $\tilde{B} = \begin{bmatrix} M & 0 \\ 0 & N \\ 0 & 0 \end{bmatrix}$ have also the same singular values. Hence, singular values of M and N are singular values of $\tilde{A}$. So far, we have been following standard approach, and Theorem 6.1.2 can be proved in this way for $G = M$.

If we prove that M and N are nonsingular, then we can introduce a multiplicative perturbation

$$\begin{bmatrix} M & L \\ K & N \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} M & 0 \\ 0 & N \\ 0 & 0 \end{bmatrix} \begin{bmatrix} I_\ell & M^{-1}L \\ N^{-1}K & I_{n-\ell} \end{bmatrix},$$

that is,

$$B = \tilde{B}D, \quad D = I + C, \quad C = \begin{bmatrix} 0_\ell & M^{-1}L \\ N^{-1}K & 0_{n-\ell} \end{bmatrix}. \tag{6.6}$$

Since B and $\tilde{B}$ have full column rank, D must be nonsingular, and from Theorem 2.3.5 it follows

$$\begin{aligned} \max_{i=1,\ldots,n} \frac{|\sigma_i - \tilde{\sigma}|}{\tilde{\sigma}_i} &\leq \|I - D^*D\|_2 = \|I - (I+C)^*(I+C)\|_2 = \|C + C^* + CC^*\|_2 \leq \\ &\leq 2\|C\|_2 + \|C\|_2^2, \end{aligned} \tag{6.7}$$

where $\|C\|_2 = \max\{\|M^{-1}L\|_2, \|N^{-1}K\|_2\}$.

So, it remains to check the nonsingularity of M and N, and to compute $\|M^{-1}L\|_2$, $\|N^{-1}K\|_2$. We will start with $M = (A^*X)^*Y$. A^*X has full column rank since $\mathcal{X} \cap \mathcal{R}(A)^\perp = \{0\}$, and let $A^*X = QR$ be a QR factorization of A^*X where $Q \in \mathbb{C}^{n\times\ell}$ is orthonormal and $R \in \mathbb{C}^{\ell\times\ell}$ is nonsingular. Then $M = R^*Q^*Y$, where $Q^*Y \in \mathbb{C}^{\ell\times\ell}$ has singular values equal to ones and cosines of acute principal angles between $\mathcal{W}$ and $\mathcal{Y}$. The question remains whether Q^*Y has full rank. If Q^*Y is singular, then by Wedin [90] it follows that there exist $y \in \mathcal{Y}$ such that $y \perp \mathcal{W}$ and $w \in \mathcal{W}$ such that $w \perp \mathcal{Y}$. By (2.15) and [90], that means that $\measuredangle(y, \mathcal{W}) = \pi/2$, $\measuredangle(w, \mathcal{Y}) = \pi/2$ and

$$\begin{aligned} \measuredangle(\mathcal{W}, \mathcal{Y}) &= \max_{w\in\mathcal{W}} \measuredangle(w, \mathcal{Y}) = \frac{\pi}{2} \\ &= \max_{y\in\mathcal{Y}} \measuredangle(y, \mathcal{W}) = \frac{\pi}{2}, \end{aligned}$$

which contradicts the assumption of the theorem. So, there do not exist $y \in \mathcal{Y}$ perpendicular to $\mathcal{W}$ and $w \in \mathcal{W}$ perpendicular to $\mathcal{Y}$. Q^*Y is nonsingular, and hence M is nonsingular.

Now, for $N = (A^*X_{1,\perp})^*Y_\perp$, $A^*X_{1,\perp}$ has full column rank since $\mathcal{R}(X_{1,\perp}) \subset \mathcal{R}(A)$, and

$$(A^\dagger X)^* A^* X_{1,\perp} = X^*(AA^\dagger)^* X_{1,\perp} = X^* X_{1,\perp} = 0. \tag{6.8}$$

$A^*\mathcal{R}(X_{1,\perp})$ and $\mathcal{Z} = A^\dagger\mathcal{X}$ are subspaces of $\mathbb{C}^n$, and $\dim(A^*\mathcal{R}(X_{1,\perp})) = n-\ell$, $\dim(\mathcal{Z}) = \ell$ and $\mathcal{Z} \perp A^*\mathcal{R}(X_{1,\perp})$. So, we can conclude that $\mathcal{Z} = (A^*\mathcal{R}(X_{1,\perp}))^\perp$ and $A^*\mathcal{R}(X_{1,\perp}) = \mathcal{Z}^\perp$. Let $A^*X_{1,\perp} = Q_\perp R_\perp$ be a QR factorization of $A^*X_{1,\perp}$, where $Q_\perp \in \mathbb{C}^{n\times(n-\ell)}$

is orthonormal and $R_\perp \in \mathbb{C}^{(n-\ell)\times(n-\ell)}$ is nonsingular. Then $N = R_\perp^* Q_\perp^* Y_\perp$, where $Q_\perp^* Y_\perp \in \mathbb{C}^{(n-\ell)\times(n-\ell)}$ has singular values equal to ones and cosines of acute principal angles between $\mathcal{Z}^\perp$ and $\mathcal{Y}^\perp$.

Let $P_\mathcal{Y}$, $P_\mathcal{W}$, and $P_\mathcal{Z}$ be orthogonal projections onto the subspaces $\mathcal{Y}$, $\mathcal{W}$ and $\mathcal{Z}$ respectively. Then, by Wedin [90] and the assumption of the theorem that $\measuredangle(\mathcal{Z},\mathcal{Y}) < \pi/2$, it follows

$$\|P_\mathcal{Z} - P_\mathcal{Y}\|_2 = \sin\measuredangle(\mathcal{Z},\mathcal{Y}) < 1. \tag{6.9}$$

On the other hand

$$\sin\measuredangle(\mathcal{Z}^\perp,\mathcal{Y}^\perp) = \|P_{\mathcal{Z}^\perp} - P_{\mathcal{Y}^\perp}\|_2 = \|(I-P_\mathcal{Z}) - (I-P_\mathcal{Y})\|_2 = \|P_\mathcal{Z} - P_\mathcal{Y}\|_2 < 1, \tag{6.10}$$

and $\measuredangle(\mathcal{Z}^\perp,\mathcal{Y}^\perp) < \pi/2$. So, by the same reasoning as before we can conclude that $Q_\perp^* Y_\perp$ is nonsingular, and hence N is nonsingular.

For computing $\|C\|_2$ we need some more theory on angles between subspaces. By [90] there exists an orthonormal basis in $\mathbb{C}^n$ such that with respect to this basis $P_\mathcal{Y}$ and $P_\mathcal{W}$ are represented by

$$P_\mathcal{Y} = \left[\begin{array}{c|c|c} I_k & & \\ \hline & \overset{\ell-k}{\underset{i=1}{\bigoplus}} J_i & \\ \hline & & 0_{n-2\ell+k} \end{array}\right], \quad P_\mathcal{W} = \left[\begin{array}{c|c|c} I_k & & \\ \hline & \overset{\ell-k}{\underset{i=1}{\bigoplus}} P_i & \\ \hline & & 0_{n-2\ell+k} \end{array}\right], \tag{6.11}$$

where

$$J_i = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad P_i = \begin{bmatrix} \cos\psi_i \\ \sin\psi_i \end{bmatrix} \begin{bmatrix} \cos\psi_i & \sin\psi_i \end{bmatrix}, \quad \psi_i \in \left\langle 0, \frac{\pi}{2} \right\rangle.$$

Similarly, there exists an orthonormal basis in $\mathbb{C}^n$ such that with respect to this basis $P_\mathcal{Y}$ and $P_\mathcal{Z}$ are represented by

$$P_\mathcal{Y} = \left[\begin{array}{c|c|c} I_{k'} & & \\ \hline & \overset{\ell-k'}{\underset{i=1}{\bigoplus}} J_i & \\ \hline & & 0_{n-2\ell+k'} \end{array}\right], \quad P_\mathcal{Z} = \left[\begin{array}{c|c|c} I_{k'} & & \\ \hline & \overset{\ell-k'}{\underset{i=1}{\bigoplus}} Q_i & \\ \hline & & 0_{n-2\ell+k'} \end{array}\right], \tag{6.12}$$

where

$$J_i = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad Q_i = \begin{bmatrix} \cos\phi_i \\ \sin\phi_i \end{bmatrix} \begin{bmatrix} \cos\phi_i & \sin\phi_i \end{bmatrix}, \quad \phi_i \in \left\langle 0, \frac{\pi}{2} \right\rangle.$$

From the QR factorization $A^*X = QR$ it follows that

$$M^{-1}L = (Q^*Y)^{-1}R^{-*}R^*Q^*Y_\perp = (Q^*Y)^{-1}Q^*Y_\perp. \tag{6.13}$$

Let us observe the following matrix

$$(P_\mathcal{W}P_\mathcal{Y})^\dagger P_\mathcal{W}P_{\mathcal{Y}^\perp} = (Y(Q^*Y)^{-1}Q^*)QQ^*Y_\perp Y_\perp^* = Y(Q^*Y)^{-1}Q^*Y_\perp Y_\perp^* = YM^{-1}LY_\perp^*,$$

where $P_{\mathcal{Y}} = YY^*$, $P_{\mathcal{Y}^\perp} = Y_\perp Y_\perp^*$ and $P_{\mathcal{W}} = QQ^*$. Hence, we can conclude that $M^{-1}L$ and $(P_{\mathcal{W}}P_{\mathcal{Y}})^\dagger P_{\mathcal{W}}P_{\mathcal{Y}^\perp} = (P_{\mathcal{W}}P_{\mathcal{Y}})^\dagger(P_{\mathcal{W}} - P_{\mathcal{W}}P_{\mathcal{Y}})$ have the same nontrivial singular values. From [90] and (6.11) it follows that in a suitably chosen basis $(P_{\mathcal{W}}P_{\mathcal{Y}})^\dagger P_{\mathcal{W}}P_{\mathcal{Y}^\perp}$ can be represented as

$$(P_{\mathcal{W}}P_{\mathcal{Y}})^\dagger P_{\mathcal{W}}P_{\mathcal{Y}^\perp} = \left[\begin{array}{c|c|c} 0_k & & \\ \hline & \overset{\ell-k}{\bigoplus_{i=1}} E_i & \\ \hline & & 0_{n-2\ell+k} \end{array}\right],$$

where

$$E_i = \left[\begin{array}{c} 1 \\ 0 \end{array}\right] \tan\psi_i \left[\begin{array}{cc} 0 & 1 \end{array}\right].$$

Finally,

$$\psi = \max_{i=1,\ldots,\ell-k} \psi_i = \measuredangle(\mathcal{W},\mathcal{Y}), \tag{6.14}$$

$$\|M^{-1}L\|_2 = \tan\psi. \tag{6.15}$$

From the QR factorization $A^*X_{1,\perp} = Q_\perp R_\perp$ it follows that

$$N^{-1}K = (Q_\perp^*Y_\perp)^{-1}R_\perp^{-*}R_\perp^*Q_\perp^*Y = (Q_\perp^*Y_\perp)^{-1}Q_\perp^*Y. \tag{6.16}$$

Now, we consider the matrix

$$(P_{\mathcal{Z}^\perp}P_{\mathcal{Y}^\perp})^\dagger P_{\mathcal{Z}^\perp}P_{\mathcal{Y}} = (Y_\perp(Q_\perp^*Y_\perp)^{-1}Q_\perp^*)Q_\perp Q_\perp^*YY^* = Y_\perp(Q_\perp^*Y_\perp)^{-1}Q_\perp^*YY^* = Y_\perp N^{-1}KY^*,$$

where $P_{\mathcal{Z}^\perp} = Q_\perp Q_\perp^*$. Again, we can conclude that $N^{-1}K$ and $(P_{\mathcal{Z}^\perp}P_{\mathcal{Y}^\perp})^\dagger P_{\mathcal{Z}^\perp}P_{\mathcal{Y}} = (I - P_{\mathcal{Z}} - P_{\mathcal{Y}} + P_{\mathcal{Z}}P_{\mathcal{Y}})^\dagger(P_{\mathcal{Y}} - P_{\mathcal{Z}}P_{\mathcal{Y}})$ have the same nontrivial singular values. From [90] and (6.12) it follows that in suitably chosen basis $(P_{\mathcal{Z}^\perp}P_{\mathcal{Y}^\perp})^\dagger P_{\mathcal{Z}^\perp}P_{\mathcal{Y}}$ can be represented as

$$(P_{\mathcal{Z}^\perp}P_{\mathcal{Y}^\perp})^\dagger P_{\mathcal{Z}^\perp}P_{\mathcal{Y}} = \left[\begin{array}{c|c|c} 0_{k'} & & \\ \hline & \overset{\ell-k}{\bigoplus_{i=1}} F_i & \\ \hline & & 0_{n-2\ell+k'} \end{array}\right],$$

where

$$F_i = \left[\begin{array}{c} 0 \\ -1 \end{array}\right] \tan\phi_i \left[\begin{array}{cc} 1 & 0 \end{array}\right].$$

Finally,

$$\phi = \max_{i=1,\ldots,\ell-k'} \phi_i = \measuredangle(\mathcal{Z},\mathcal{Y}), \tag{6.17}$$

$$\|N^{-1}K\|_2 = \tan\phi. \tag{6.18}$$

Thus,

$$\|C\|_2 = \max\{\|M^{-1}L\|_2, \|N^{-1}K\|_2\} = \max\{\tan\psi, \tan\phi\} = \tan\theta, \tag{6.19}$$

and we finished the proof. □

Corollary 6.2.2. *Under the same conditions as in Theorem 6.2.1, but with additional constraint $\theta < \pi/4$, the following holds*

$$\max_{i=1,\ldots,n} \frac{|\sigma_i - \tilde{\sigma}_i|}{\sqrt{\sigma_i \tilde{\sigma}_i}} \leq \tan\theta + \frac{\tan^2\theta}{2(1-\tan\theta)}. \tag{6.20}$$

Proof. By (6.6) and (6.19) it is $\|C\|_2 < 1$, and from Theorem 2.3.7 it follows

$$\begin{aligned}
\max_{i=1,\ldots,n} \frac{|\sigma_i - \tilde{\sigma}_i|}{\sqrt{\sigma_i \tilde{\sigma}_i}} &\leq \frac{1}{2}\|D^* - D^{-1}\|_2 = \frac{1}{2}\|I + C^* - I + \sum_{i=1}^{\infty}(-1)^{i-1}C^i\|_2 \leq \\
&\leq \|C\|_2 + \frac{1}{2}\sum_{i=2}^{\infty}\|C\|_2^i = \|C\|_2 + \frac{\|C\|_2^2}{2(1-\|C\|_2)} \leq \\
&\leq \tan\theta + \frac{\tan^2\theta}{2(1-\tan\theta)}.
\end{aligned}$$

□

The approximations $\tilde{\sigma}_i$ will be close to the exact singular values σ_i if the angle θ is small. On the other hand, subspaces close to singular subspaces can produce large θ and large relative error in singular values. For example, let us observe the following simple example. The matrix A is defined as

$$A = \begin{bmatrix} \alpha^{-2} & 0 \\ 0 & \alpha \end{bmatrix}, \qquad 0 < \alpha \ll 1,$$

and the vectors

$$x = y = \frac{1}{\sqrt{1+\alpha^2}}\begin{bmatrix} \alpha \\ 1 \end{bmatrix} \approx \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$

are very close to the singular vectors corresponding to the singular value $\sigma_2 = \alpha$. Further, the singular value approximation is given by

$$\tilde{\sigma}_2 = x^T A y = \frac{1}{1+\alpha^2}(1+\alpha) \approx 1 + \alpha,$$

whose relative error is large:

$$\frac{|\tilde{\sigma}_2 - \sigma_2|}{\sigma_2} \approx \alpha^{-1}.$$

The space $A^*\mathcal{X}$ is spanned by the vector

$$\frac{A^*x}{\|A^*x\|_2} = \frac{1}{\sqrt{\alpha^{-2}+\alpha^2}}\begin{bmatrix} \alpha^{-1} \\ \alpha \end{bmatrix} \approx \begin{bmatrix} 1 \\ \alpha^2 \end{bmatrix} \approx \begin{bmatrix} 1 \\ 0 \end{bmatrix},$$

and we can conclude that

$$\measuredangle(A^*\mathcal{X}, \mathcal{Y}) \approx \frac{\pi}{2}.$$

The bound in Theorem 6.2.1 is illustrated in the following numerical example.

Example 6.2.3. *Let $A \in \mathbb{R}^{100\times 20}$ have fixed singular values $\{1, 2, \ldots, 19, 100\}$, and suppose we are looking for the 5 largest singular values. We use the power method on A^*A, where $Y_0 \in \mathbb{R}^{20\times 5}$ is a random orthonormal matrix and $X_i = AY_i$, $i = 1, 2, \ldots, 120$. We obtain the results shown in Figure 6.1.*

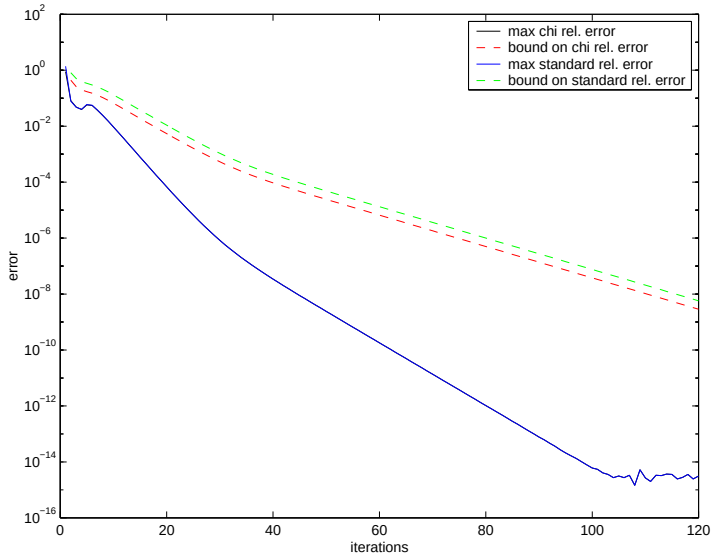


Figure 6.1: Relative errors in singular values, and the bounds (6.20) and (6.5).

6.3 Quadratic Residual Bound

As we can see in Example 6.2.3, the error bound given in Theorem 6.2.1 is not very tight. The next step is to find more accurate bound, involving the same angles and subspaces. This is done in the following theorem, which is mostly based on the results from [23].

Let us assume, once more, that the bases for $\mathcal{X}$, $\mathcal{X}^\perp$ and $\mathcal{Y}$ are chosen so that

$$A = \begin{bmatrix} \Sigma_M & L \\ K & \Sigma_N \\ 0 & 0 \end{bmatrix}, \tag{6.21}$$

where $\Sigma_M = \operatorname{diag}(\mu_1, \ldots, \mu_\ell)$ and $\Sigma_N = \operatorname{diag}(\nu_1, \ldots, \nu_{n-\ell})$ are diagonal matrices with the singular values of M and N, respectively. Since the last $m - n$ rows of A in (6.21) are equal to zero, without loss of generality we can assume that $A \in \mathbb{C}^{n\times n}$ is square and of the form

$$A = \begin{bmatrix} \Sigma_M & L \\ K & \Sigma_N \end{bmatrix}. \tag{6.22}$$

Next we will assume that the singular values of M and N are separated by the interval $\langle \alpha, \beta \rangle$, which means that for $\mu_1 \geq \cdots \geq \mu_\ell$ and $\nu_1 \geq \cdots \geq \nu_{n-\ell}$

$$\nu_1 \leq \alpha < \beta \leq \mu_\ell,$$

and we define relative gaps

$$\rho(\sigma_{\ell+i}, \Sigma_M) = \min_{j=1,\ldots,\ell} \frac{|\mu_j - \sigma_{\ell+i}|}{\mu_j}, \qquad \rho(\sigma_i, \Sigma_N) = \min_{j=1,\ldots,n-\ell} \frac{|\nu_j - \sigma_i|}{\nu_j}.$$

Theorem 6.3.1. *Let A, $\mathcal{X}$, $\mathcal{Y}$, and $\psi, \phi, \theta < \pi/4$ be as in Theorem 6.2.1 and relation (6.22). Let the singular values of M and N be separated by the interval $\langle \alpha, \beta \rangle$. Then*

1. *for each $i \in \{1, \dots, \ell\}$, if σ_i is not a singular value of $\begin{bmatrix} K & \Sigma_N \end{bmatrix}$ and if $\rho(\sigma_i, \Sigma_N) > \tan^2 \phi$, then*

$$\frac{|\mu_i - \sigma_i|}{\mu_i} \leq \min\left\{2\tan\theta + \tan^2\theta, \max\left\{\tan^2\phi, \tan^2\psi + \frac{(\tan\psi + \tan\phi)^2}{\rho(\sigma_i, \Sigma_N) - \tan^2\phi}\right\}\right\}, \tag{6.23}$$

   in case when $\tan^2\psi + \frac{(\tan\psi+\tan\phi)^2}{\rho(\sigma_i,\Sigma_N)-\tan^2\phi} < 1$;

2. *for each $i \in \{1, \dots, n-\ell\}$, if $\sigma_{\ell+i}$ is not a singular value of $\begin{bmatrix} \Sigma_M & L \end{bmatrix}$ and if $\rho(\sigma_{\ell+i}, \Sigma_M) > \tan^2\psi$, then*

$$\frac{|\nu_i - \sigma_{\ell+i}|}{\nu_i} \leq \min\left\{2\tan\theta + \tan^2\theta, \max\left\{\tan^2\psi, \tan^2\phi + \frac{(\tan\psi + \tan\phi)^2}{\rho(\sigma_{\ell+i}, \Sigma_M) - \tan^2\psi}\right\}\right\}, \tag{6.24}$$

   in case when $\tan^2\phi + \frac{(\tan\psi+\tan\phi)^2}{\rho(\sigma_{\ell+i},\Sigma_M)-\tan^2\psi} < 1$.

If we have no information about the distributions of the singular values, then for the singular values $\tilde{\sigma}_i$ of $\begin{bmatrix} \Sigma_M & 0 \\ 0 & \Sigma_N \end{bmatrix}$ we can write

$$\left|\frac{\tilde{\sigma}_i - \sigma_i}{\tilde{\sigma}_i}\right| \leq \min\left\{2\tan\theta + \tan^2\theta, \tan^2\theta\left(1 + \frac{4}{\max\{\rho(\sigma_i, \Sigma_M) - \tan^2\psi, \rho(\sigma_i, \Sigma_N) - \tan^2\phi\}}\right)\right\}, \tag{6.25}$$

where if $\tilde{\sigma}_i = \mu_{j_i}$ for some $1 \leq j_i \leq \ell$, then the conditions from item 1. have to be satisfied, and if $\tilde{\sigma}_i = \nu_{k_i}$ for some $1 \leq k_i \leq n - \ell$, then the conditions from item 2. have to be satisfied.

Proof. Let us observe the Schur factorization of $AA^* - \sigma_i^2 I$, for σ_i which is not a singular value of $\begin{bmatrix} K & \Sigma_N \end{bmatrix}$:

$$\begin{aligned} AA^* - \sigma_i^2 I &= \begin{bmatrix} \Sigma_M^2 + LL^* - \sigma_i^2 I & \Sigma_M K^* + L\Sigma_N \\ K\Sigma_M + \Sigma_N L^* & KK^* + \Sigma_N^2 - \sigma_i^2 I \end{bmatrix} = \\ &= \begin{bmatrix} I & (\Sigma_M K^* + L\Sigma_N)(KK^* + \Sigma_N^2 - \sigma_i^2 I)^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} N'(\sigma_i) & 0 \\ 0 & KK^* + \Sigma_N^2 - \sigma_i^2 I \end{bmatrix} \cdot \\ &\quad \cdot \begin{bmatrix} I & 0 \\ (KK^* + \Sigma_N^2 - \sigma_i^2 I)^{-1}(K\Sigma_M + \Sigma_N L^*) & I \end{bmatrix}, \end{aligned}$$

where $N'(\sigma_i)$ is the Schur complement

$$N'(\sigma_i) = \Sigma_M^2 + LL^* - \sigma_i^2 I - (\Sigma_M K^* + L\Sigma_N)(KK^* + \Sigma_N^2 - \sigma_i^2 I)^{-1}(K\Sigma_M + \Sigma_N L^*).$$

Then $AA^* - \sigma_i^2 I$ is congruent to

$$\begin{bmatrix} \Sigma_M^2 - \sigma_i^2 I & 0 \\ 0 & \Sigma_N^2 - \sigma_i^2 I \end{bmatrix} + \begin{bmatrix} LL^* - (\Sigma_M K^* + L\Sigma_N)(KK^* + \Sigma_N^2 - \sigma_i^2 I)^{-1}(K\Sigma_M + \Sigma_N L^*) & 0 \\ 0 & KK^* \end{bmatrix}. \tag{6.26}$$

Since the i-th eigenvalue of $AA^* - \sigma_i^2 I$ is equal to zero, Sylvester's inertia theorem implies that the matrix in (6.26) have zero as the i-th eigenvalue. It further follows that σ_i^2 is the i-th eigenvalue of the matrix

$$H(\sigma_i) = \begin{bmatrix} \Sigma_M^2 & 0 \\ 0 & \Sigma_N^2 \end{bmatrix} + \begin{bmatrix} LL^* - (\Sigma_M K^* + L\Sigma_N)(KK^* + \Sigma_N^2 - \sigma_i^2 I)^{-1}(K\Sigma_M + \Sigma_N L^*) & 0 \\ 0 & KK^* \end{bmatrix}.$$

So, the eigenvalues of $H(\sigma_i)$ are now compared to μ_j^2 and ν_j^2. We can write

$$H(\sigma_i) = \begin{bmatrix} \Sigma_M & 0 \\ 0 & \Sigma_N \end{bmatrix} \left\{ I + \begin{bmatrix} \Sigma_M^{-1} LL^* \Sigma_M^{-1} - (K^* + \Sigma_M^{-1} L\Sigma_N)(KK^* + \Sigma_N^2 - \sigma_i^2 I)^{-1}(K + \Sigma_N L^* \Sigma_M^{-1}) & 0 \\ 0 & \Sigma_N^{-1} KK^* \Sigma_N^{-1} \end{bmatrix} \right\} \begin{bmatrix} \Sigma_M & 0 \\ 0 & \Sigma_N \end{bmatrix},$$

Let us define the matrix C as

$$\begin{aligned} C &= \begin{bmatrix} \Sigma_M^{-1} LL^* \Sigma_M^{-1} - (K^* + \Sigma_M^{-1} L\Sigma_N)(KK^* + \Sigma_N^2 - \sigma_i^2 I)^{-1}(K + \Sigma_N L^* \Sigma_M^{-1}) & 0 \\ 0 & \Sigma_N^{-1} KK^* \Sigma_N^{-1} \end{bmatrix} = \\ &= \begin{bmatrix} \Sigma_M^{-1} LL^* \Sigma_M^{-1} - (K^* \Sigma_N^{-1} + \Sigma_M^{-1} L)(\Sigma_N^{-1} KK^* \Sigma_N^{-1} + I - \sigma_i^2 \Sigma_N^{-2})^{-1}(\Sigma_N^{-1} K + L^* \Sigma_M^{-1}) & 0 \\ 0 & \Sigma_N^{-1} KK^* \Sigma_N^{-1} \end{bmatrix} = \\ &= \begin{bmatrix} EE^* - (F^* + E)(FF^* + I - \sigma_i^2 \Sigma_N^{-2})^{-1}(F + E^*) & 0 \\ 0 & FF^* \end{bmatrix}, \end{aligned}$$

where by the proof of Theorem 6.2.1

$$E = \Sigma_M^{-1} L, \qquad \|E\|_2 = \tan\psi, \tag{6.27}$$

$$F = \Sigma_N^{-1} K, \qquad \|F\|_2 = \tan\phi. \tag{6.28}$$

So, in case when

$$\|C\|_2 = \max\{\|EE^* - (F^* + E)(FF^* + I - \sigma_i^2 \Sigma_N^{-2})^{-1}(F + E^*)\|_2, \|FF^*\|_2\} < 1,$$

$I + C$ is positive definite. Since from the condition of the theorem $\|FF^*\|_2 = \tan^2\phi < 1$, we need to explore the other term in $\|C\|_2$:

$$\begin{aligned} &\|EE^* - (F^* + E)(FF^* + I - \sigma_i^2 \Sigma_N^{-2})^{-1}(F + E^*)\|_2 \leq \\ &\quad \leq \tan^2\psi + (\tan\psi + \tan\phi)^2 \|(FF^* + I - \sigma_i^2 \Sigma_N^{-2})^{-1}\|_2 \leq \\ &\quad \leq \tan^2\psi + \frac{(\tan\psi + \tan\phi)^2}{\min_{j=1,\ldots,n-\ell} \frac{|\sigma_i^2 - \nu_j^2|}{\nu_j^2} - \tan^2\phi} \leq \\ &\quad \leq \tan^2\psi + \frac{(\tan\psi + \tan\phi)^2}{\rho(\sigma_i, \Sigma_N) - \tan^2\phi}, \end{aligned}$$

under the condition that $\rho(\sigma_i, \Sigma_N) > \tan^2\phi$. So, in addition if we demand that $\tan^2\psi + \frac{(\tan\psi + \tan\phi)^2}{\rho(\sigma_i, \Sigma_N) - \tan^2\phi} < 1$ then $\|C\|_2 < 1$ and again $I + C$ is positive definite. Hence, we can conclude that

$$H(\sigma_i) \sim (I + C)^{\frac{1}{2}} \begin{bmatrix} \Sigma_M^2 & 0 \\ 0 & \Sigma_N^2 \end{bmatrix} (I + C)^{\frac{1}{2}} = (I + C)^{\frac{1}{2}} \tilde{A}\tilde{A}^* (I + C)^{\frac{1}{2}}.$$

By Theorem 4.3.4 it follows that

$$\begin{aligned} \frac{|\mu_i - \sigma_i|}{\mu_i} &\leq \frac{|\mu_i^2 - \sigma_i^2|}{\mu_i^2} \leq \|C\|_2 \leq \\ &\leq \max\left\{ \tan^2\phi, \tan^2\psi + \frac{(\tan\psi + \tan\phi)^2}{\rho(\sigma_i, \Sigma_N) - \tan^2\phi} \right\}. \end{aligned}$$

This proves (6.23).

Relation (6.24) can be proven in the same way, in case when $\sigma_{\ell+i}$ is not a singular value of $\begin{bmatrix} \Sigma_M & L \end{bmatrix}$ and if we factorize

$$AA^* - \sigma_{\ell+i}^2 I = \begin{bmatrix} I & 0 \\ (K\Sigma_M + \Sigma_N L^*)(\Sigma_M^2 + LL^* - \sigma_{\ell+i}^2 I)^{-1} & I \end{bmatrix} \begin{bmatrix} \Sigma_M^2 + LL^* - \sigma_{\ell+i}^2 I & 0 \\ 0 & M'(\sigma_{\ell+i}) \end{bmatrix} \cdot \cdot \begin{bmatrix} I & (\Sigma_M^2 + LL^* - \sigma_{\ell+i}^2 I)^{-1}(\Sigma_M K^* + L\Sigma_N) \\ 0 & I \end{bmatrix},$$

where $M'(\sigma_{\ell+i})$ is the Schur complement

$$M'(\sigma_{\ell+i}) = KK^* + \Sigma_N^2 - \sigma_{\ell+i}^2 I - (K\Sigma_M + \Sigma_N L^*)(\Sigma_M^2 + LL^* - \sigma_{\ell+i}^2 I)^{-1}(\Sigma_M K^* + L\Sigma_N).$$

□

Example 6.3.2. *Let, again, $A \in \mathbb{R}^{100\times 20}$ be defined as in Example 6.2.3, and suppose the same iteration are performed as in Example 6.2.3. We obtain the results shown in Figure 6.2.*

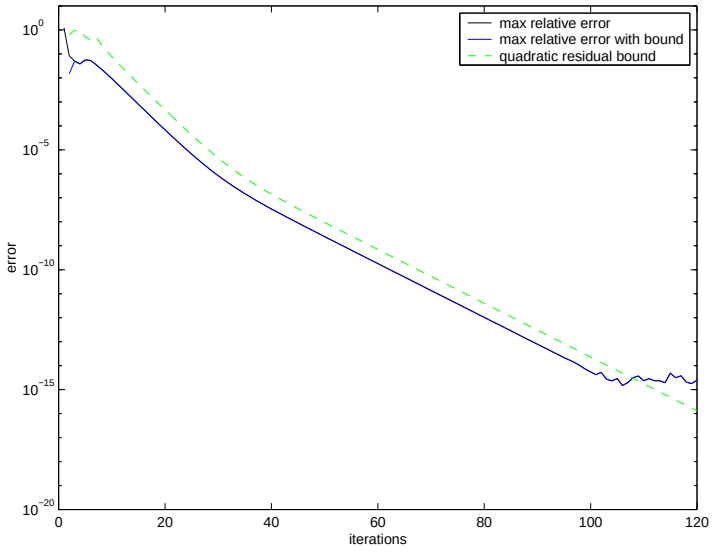


Figure 6.2: Relative errors in singular values, and the quadratic bound (6.25). The blue line denotes the maximum relative error for the singular values that satisfy the conditions of Theorem 6.3.1. The conditions are not met only at the beginning of iterations.

The example confirms that the new quadratic bound for relative errors in singular values is tight, as it can be expected. After the 100-th iteration we can note that the bound becomes smaller than the computed relative error. This happens when the computed relative error reaches the order of machine precision and remains on that level, while the bound is computed more accurately. In the exact arithmetic the relative error would continue to decrease.

Example 6.3.3. *In this example we compare the bounds in Theorem 6.3.1 and Corollary 6.1.3. The important difference between these two bounds is that the bound in Corollary 6.1.3 involves the absolute gap, and the bound in Theorem 6.3.1 involves the relative gap. So, we generate a matrix* $A \in \mathbb{R}^{100\times 20}$ *with fixed singular values* $\{0.001, 0.002, \ldots, 0.015, 0.015000001, 0.0151,\ 0.0152, 0.0153, 0.0154\}$*, and we search for the 5 largest singular values whose absolute distance from the remaining singular values is smaller than the relative distance. The same iteration are performed as in Example 6.2.3, and the results are shown in Figure 6.3.*

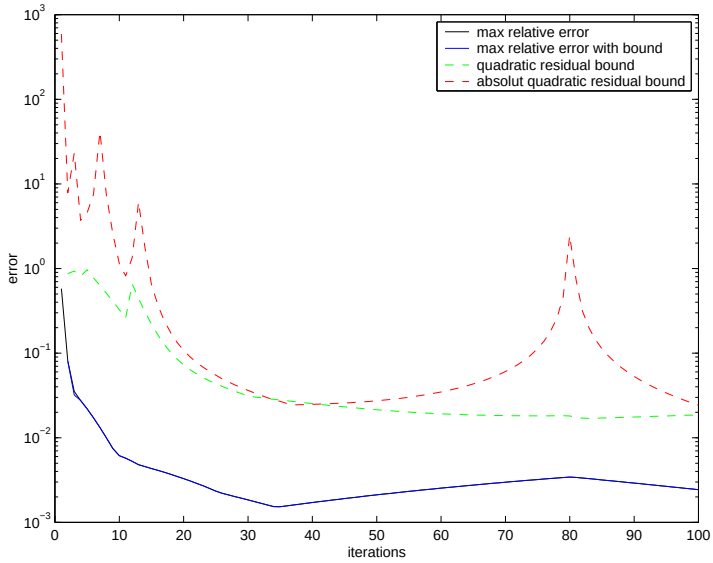


Figure 6.3: Relative errors in singular values, and the quadratic bounds (6.25) and (6.4). The red dashed line denotes the relative error bound obtained from Corollary 6.1.3, for the singular value approximation with the largest relative error.

Figure 6.3 confirms that the relative error bound is in the most cases better than the absolute error bound applied to the relative error.

Example 6.3.4. *We will perform one more test for our bounds, by using more sophisticated method for computing a few extremal singular values: a Jacobi–Davidson type SVD method (*`jdsvd`*) proposed by Hochstenbach in [48]. We are searching for the* 5 *largest singular values of* $A \in \mathbb{R}^{1000\times 250}$*, whose singular values are equal to* $i^2/100$*,* $i = 1, \ldots, 250$*. The relative tolerance of the outer iteration is taken to be equal to* 10^{-12}*. We obtain the results shown in Figure 6.4.*

The `jdsvd` method computes one pair of left and right singular vectors at the time, hence we can notice 5 peaks and 5 valleys in Figure 6.4. The peaks represent a starting point when the initial search directions for a new pair is chosen, and the valleys represent the iterations when the pair reached maximum accuracy. We can notice that both of the residual bounds (linear and quadratic) are following the shape of the relative error curve, but the quadratic bound is more tight when the approximations are more accurate.

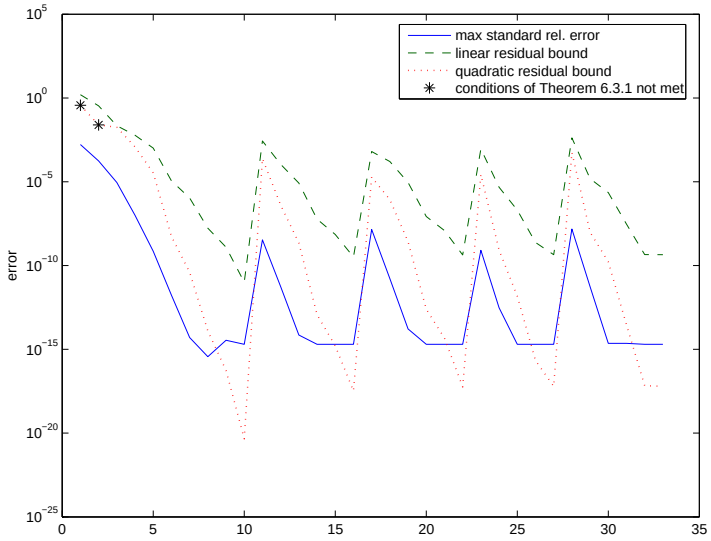


Figure 6.4: Relative errors in singular values, the linear bound and the quadratic bound for `jdsvd`.

Here we can also observe the phenomenon of the quadratic bound being smaller than the actual relative error when the error reaches the order of machine precision.

6.4 Rank Deficient Case

The statements of Theorem 6.2.1 and Theorem 6.3.1 can be easily generalized to the rank deficient case. We will go through the proof of Theorem 6.2.1 once more, under the assumption that $\mathrm{rank}(A) = r < n$. In that case we define an orthonormal basis $X_\perp = [X_{1,\perp}, X_{2,\perp}]$ for $\mathcal{X}^\perp$, where $X_{1,\perp} \in \mathbb{C}^{m\times(r-\ell)}$ and $X_{2,\perp} \in \mathbb{C}^{m\times(m-r)}$ are such that $\mathcal{R}(X_{1,\perp}) \subset \mathcal{R}(A)$ and $X_{2,\perp}^* A = 0$. Let $Y_\perp \in \mathbb{C}^{n\times(n-\ell)}$ be an orthonormal basis for $\mathcal{Y}^\perp$. Then, the matrix A can be written as

$$A = \begin{bmatrix} X & X_{1,\perp} & X_{2,\perp} \end{bmatrix} \begin{bmatrix} M & L \\ K & N \\ 0 & 0 \end{bmatrix} \begin{bmatrix} Y^* \\ Y_\perp^* \end{bmatrix},$$

where $L = X^*AY_\perp \in \mathbb{C}^{\ell\times(n-\ell)}$, $K = X_{1,\perp}^*AY \in \mathbb{C}^{(r-\ell)\times\ell}$ and $N = X_{1,\perp}^*AY_\perp \in \mathbb{C}^{(r-\ell)\times(n-\ell)}$.

As in case with the full rank matrix, we can conclude that M is nonsingular. But, N is not square any more, and now we have to prove that $\mathrm{rank}(N) = r - \ell$. By (6.8), we have again that

$$A^*\mathcal{R}(X_{1,\perp}) \subset \mathcal{R}(A^*) \quad \text{and} \quad \mathcal{Z} = A^\dagger\mathcal{X} \subset \mathcal{R}(A^\dagger) = \mathcal{R}(A^*),$$

with

$$\dim(A^*\mathcal{R}(X_{1,\perp})) = r - \ell, \quad \dim(\mathcal{Z}) = \ell, \quad \text{and} \quad \mathcal{Z} \perp A^*\mathcal{R}(X_{1,\perp}),$$

so we can conclude that

$$\mathcal{R}(A^*) = \mathcal{Z} \oplus A^*\mathcal{R}(X_{1,\perp}).$$

Since $\mathbb{C}^n = \mathcal{R}(A^*) \oplus \mathcal{N}(A)$, where $\mathcal{N}(A)$ is the kernel of A, we have

$$\mathcal{Z} = (A^*\mathcal{R}(X_{1,\perp}) \oplus \mathcal{N}(A))^{\perp}.$$

Let $S_\perp$ be the orthonormal basis for $\mathcal{N}(A)$, then from (6.9) and (6.10) it follows that $[\ Q_\perp \ \ S_\perp\ ]^*Y_\perp$ is nonsingular, where $[\ Q_\perp \ \ S_\perp\ ]$ is an orthonormal basis for $\mathcal{Z}^\perp$, and $Q_\perp$ is an orthonormal basis for $A^*\mathcal{R}(X_{1,\perp})$, such that $A^*X_{1,\perp} = Q_\perp R_\perp$, $N = R_\perp^* Q_\perp^* Y_\perp$ and $R_\perp$ is nonsingular. The matrix $Q_\perp^* Y_\perp$ is the upper $(n-r)\times(n-\ell)$ block of the $(n-\ell)\times(n-\ell)$ matrix $[\ Q_\perp \ \ S_\perp\ ]^*Y_\perp$, and by the interlacing property (Corollary 2.1.14), the following holds

$$\sigma_{r-\ell}(Q_\perp^* Y_\perp) \geq \sigma_{n-\ell}([\ Q_\perp \ \ S_\perp\ ]^*Y_\perp) > 0.$$

Thus, $Q_\perp^* Y_\perp$ and N have full row rank, and $NN^\dagger = I_{r-\ell}$. Now, we can write

$$\begin{bmatrix} M & L \\ K & N \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} M & 0 \\ 0 & N \\ 0 & 0 \end{bmatrix} \begin{bmatrix} I_\ell & M^{-1}L \\ N^\dagger K & I_{n-\ell} \end{bmatrix},$$

where A and B are similar, and $\tilde{A}$ and $\tilde{B}$ are similar, and

$$B = \tilde{B}D, \quad D = I + C, \quad C = \begin{bmatrix} 0_\ell & M^{-1}L \\ N^\dagger K & 0_{n-\ell} \end{bmatrix}. \tag{6.29}$$

Since $\|C\|_2 = \max\{\|M^{-1}L\|_2, \|N^\dagger K\|_2\}$ and $\|M^{-1}L\|_2 = \tan\psi$ as in the full rank case, we have to compute $\|N^\dagger K\|_2$. We also have to prove that D is nonsingular in order to apply perturbation theory.

First we want to show that D is nonsingular. Let us consider the following matrix product

$$_r\left\{ \begin{bmatrix} M & L \\ \underbrace{K}_{n} & \underbrace{N} \end{bmatrix} = {}_r\left\{ \begin{bmatrix} M - LN^\dagger K & L \\ \underbrace{0}_{n} & \underbrace{N} \end{bmatrix} \begin{bmatrix} I_\ell & 0 \\ \underbrace{N^\dagger K}_{n} & \underbrace{I_{n-\ell}} \end{bmatrix} \right\}_n.$$

The matrix

$$\begin{matrix} \ell \\ r-\ell \end{matrix} \begin{bmatrix} M & L \\ K & N \end{bmatrix} \begin{matrix} \\ \\ \ell \end{matrix} \quad \begin{matrix} \\ \\ n-\ell \end{matrix}$$

has full row rank, and its rank is equal to r, since $\mathrm{rank}(A) = r$. This means that this matrix has r linearly independent columns. Since M is nonsingular, that implies that its first ℓ columns are linearly independent and that $\begin{bmatrix} L \\ N \end{bmatrix}$ has remaining $r-\ell$ linearly independent columns. On the other hand, the matrix $\begin{bmatrix} I_\ell & 0 \\ N^\dagger K & I_{n-\ell} \end{bmatrix}$ is nonsingular, which implies that

$$\begin{matrix} \ell \\ r-\ell \end{matrix} \begin{bmatrix} M - LN^\dagger K & L \\ 0 & N \end{bmatrix} \begin{matrix} \\ \\ \ell \end{matrix} \quad \begin{matrix} \\ \\ n-\ell \end{matrix}$$

must have full row rank. Again, this means that this matrix has r linearly independent columns, and among the last $n-\ell$ columns, $r-\ell$ of them are linearly independent. Hence, the first ℓ columns must be linearly independent, which implies that $M - LN^\dagger K$ is nonsingular.

Now, if we assume that D is singular, then there exist $x \in \mathbb{C}^\ell$ and $y \in \mathbb{C}^{n-\ell}$ such that

$$\begin{aligned} x + M^{-1}Ly &= 0 \\ N^\dagger Kx + y &= 0. \end{aligned}$$

Extracting y from the second equation, and introducing it to the first equation produces the following relation

$$x - M^{-1}LN^\dagger Kx = 0,$$

that is $(M - LN^\dagger K)x = 0$. This implies that the matrix $M - LN^\dagger K$ is singular, which is a contradiction with the previous analysis. So, D is nonsingular and

$$\max_{i=1,\ldots,n} \frac{|\sigma_i - \tilde{\sigma}|}{\tilde{\sigma}_i} \leq 2\|C\|_2 + \|C\|_2^2.$$

Finally, we have to estimate $\|C\|_2$. As in the proof of Theorem 6.2.1 (see (6.16)), we can write

$$N^\dagger K = (Q_\perp^* Y_\perp)^\dagger Q_\perp^* Y.$$

The matrices $[\ Q_\perp \quad S_\perp\ ]^* Y_\perp$ and $P_{\mathcal{Z}^\perp} P_{\mathcal{Y}^\perp} = [\ Q_\perp \quad S_\perp\ ][\ Q_\perp \quad S_\perp\ ]^* Y_\perp Y_\perp^*$ have the same singular values, as well as the matrices $[\ Q_\perp \quad S_\perp\ ]^* Y$ and $P_{\mathcal{Z}^\perp} P_{\mathcal{Y}} = [\ Q_\perp \quad S_\perp\ ] \cdot$ $\cdot [\ Q_\perp \quad S_\perp\ ]^* Y Y^*$. From [90] and (6.12) it follows that $P_{\mathcal{Z}^\perp} P_{\mathcal{Y}^\perp}$ can be represented in a suitably chosen basis as

$$P_{\mathcal{Z}^\perp} P_{\mathcal{Y}^\perp} = \left[\begin{array}{c|c|c} 0_{k'} & & \\ \hline & \bigoplus_{i=1}^{\ell-k} G_i & \\ \hline & & I_{n-2\ell+k'} \end{array}\right],$$

where

$$G_i = \left[\begin{array}{c} -\sin\phi_i \\ \cos\phi_i \end{array}\right] \cos\phi_i [\ 0 \quad 1\ ],$$

and $P_{\mathcal{Z}^\perp} P_{\mathcal{Y}}$ can be represented as

$$P_{\mathcal{Z}^\perp} P_{\mathcal{Y}} = \left[\begin{array}{c|c|c} 0_{k'} & & \\ \hline & \bigoplus_{i=1}^{\ell-k} H_i & \\ \hline & & 0_{n-2\ell+k'} \end{array}\right],$$

where

$$H_i = \left[\begin{array}{c} \sin\phi_i \\ -\cos\phi_i \end{array}\right] \sin\phi_i [\ 1 \quad 0\ ].$$

Finally, by the interlacing property we can conclude that

$$\begin{aligned} \|(Q_\perp^* Y_\perp)^\dagger\|_2 &= \frac{1}{\sigma_{r-\ell}(Q_\perp^* Y_\perp)} \leq \frac{1}{\sigma_{n-\ell}([\ Q_\perp \quad S_\perp\ ]^* Y_\perp)} = \frac{1}{\cos\phi}, \\ \|Q_\perp^* Y\|_2 &= \sigma_1(Q_\perp^* Y) \leq \sigma_1([\ Q_\perp \quad S_\perp\ ]^* Y_\perp) = \sin\phi, \\ \|N^\dagger K\|_2 &\leq \tan\phi, \end{aligned}$$

where ϕ is defined in (6.17). This implies that

$$\|C\|_2 \leq \max\{\tan\psi, \tan\phi\},$$

and we obtained the same result as in Theorem 6.2.1.

Bibliography

[1] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. W. Demmel, J. J. Dongarra, J. J. Du Croz, A. Greenbaum, S. J. Hammarling, A. McKenney, and D. C. Sorensen, *LAPACK Users' Guide*, SIAM, Philadelphia, third ed., 1999.

[2] J. L. Barlow, N. Bosner, and Z. Drmač, *A New Stable Bidiagonal Reduction Algorithm*, Linear Algebra Appl., 397 (2005), pp. 35–84.

[3] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear Programming : Theory and Algorithms*, John Wiley & Sons, Inc., New York, Chichester, Brisbane, Toronto, Singapore, second ed., 1983.

[4] E. Beltrami, *Sulle Funzioni Bilineari*, Giornale di Matematiche ud uso Degli Studenti Delle Universita, 11 (1873), pp. 98–106.

[5] C. Bischof and C. F. V. Loan, *The WY Representation for Products of Householder Matrices*, SIAM J. Sci. Stat. Comput., 8 (1987), pp. s2–s13.

[6] Å. Björck and C. C. Paige, *Loss and Recapture of Orthogonality in the Modified Gram-Schmidt Algorithm*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 176–190.

[7] L. S. Blackford, J. Choi, A. Cleary, E. D'Azevedo, J. W. Demmel, I. S. Dhillon, J. J. Dongarra, S. J. Hammarling, G. Henry, A. Petitet, K. Stanley, D. Walker, and R. C. Whaley, *ScaLAPACK Users' Guide*, SIAM, Philadelphia, 1997.

[8] N. Bosner and Z. Drmač, *On Accuracy Properties of One-Sided Bidiagonalization Algorithm and Its Applications*, in Proceedings of the Conference on Applied Mathematics and Scientific Computing, Z. Drmač, M. Marušić, and Z. Tutek, eds., Dordrecht, 2005, Springer, pp. 141–150.

[9] A. Brandt, S. F. McCormick, and J. Ruge, *Multigrid Methods for Differential Eigenproblems*, SIAM J. Sci. Statist. Comput., 4 (1983), pp. 244–260.

[10] Z. Cai, J. Mandel, and S. F. McCormick, *Multigrid Methods for Nearly Singular Linear Equations and Eigenvalue Problems*, SIAM J. Numer. Anal., 34 (1997), pp. 178–200.

[11] T. F. Chan, *An Improved Algorithm for Computing the Singular Value Decomposition*, ACM Trans. Math. Softw., 8 (1982), pp. 72–83.

[12] C. Davis, W. M. Kahan, and H. F. Weinberger, *Norm-Preserving Dilations and Their Applications to Optimal Error Bounds*, SIAM J. Numer. Anal., 19 (1982), pp. 445–469.

[13] J. W. Demmel, *On Floating Point Errors in Cholesky*, LAPACK Working Note #14, 1989.

[14] J. W. Demmel, S. C. Eisenstat, J. R. Gilbert, X. S. Li, and J. W. H. Liu, *A Supernodal Approach to Sparse Partial Pivoting*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 720–755.

[15] J. W. Demmel and W. M. Kahan, *Accurate Singular Values of Bidiagonal Matrices*, SIAM J. Sci. Stat. Comput., 11 (1990), pp. 873–912.

[16] J. W. Demmel and K. Veselić, *Jacobi's Method is More Accurate than QR*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 1204–1245.

[17] I. S. Dhillon and B. N. Parlett, *Multiple Representations to Compute Orthogonal Eigenvectors of Symmetric Tridiagonal Matrices*, Linear Algebra Appl., 387 (2004), pp. 1–28.

[18] J. J. Dongarra, J. J. Du Croz, S. J. Hammarling, and I. S. Duff, *A Set of Level 3 Basic Linear Algebra Subprograms*, ACM Trans. Math. Softw., 16 (1990), pp. 1–17.

[19] ——, *Algorithm 679; A Set of Level 3 Basic Linear Algebra Subprogram: Model Implementation and Test Programs*, ACM Trans. Math. Softw., 16 (1990), pp. 18–28.

[20] J. J. Dongarra, J. J. Du Croz, S. J. Hammarling, and R. J. Hansen, *An Extended Set of Fortran Basic Linear Algebra Subroutines*, ACM Trans. Math. Softw., 14 (1988), pp. 1–17.

[21] J. J. Dongarra, S. J. Hammarling, and D. C. Sorensen, *Block Reduction of Matrices to Condensed Forms for Eigenvalue Computations*, J. Comput. Appl. Math., 27 (1989), pp. 215–227.

[22] Z. Drmač, *On Relative Residual Bounds for the Eigenvalues of a Hermitian Matrix*, Linear Algebra Appl., 244 (1996), pp. 155–163.

[23] Z. Drmač and V. Hari, *Relative Residual Bounds for the Eigenvalues of a Hermitian Semidefinite Matrix*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 21–29.

[24] Z. Drmač and K. Veselić, *New Fast and Accurate Jacobi SVD Algorithm: I*, LAPACK Working Note #169, 2005.

[25] ——, *New Fast and Accurate Jacobi SVD Algorithm: II*, LAPACK Working Note #170, 2005.

[26] S. C. Eisenstat and I. C. F. Ipsen, *Relative Perturbation Bounds for Eigenspaces and Singular Vector Subspaces*, in Applied Linear Algebra. Proceedings of the 5th SIAM Conference, J. G. Lewis, ed., held in Snowbird, USA, 1994, SIAM.

[27] L. Elden and H. Park, *A Procrustes Problem on the Stiefel Manifold*, Numer. Math., 82 (1999), pp. 599–619.

[28] J.-L. Fattebert, *A Block Rayleigh Quotient Iteration with Local Quadratic Convergence*, ETNA, Electron. Trans. Numer. Anal, 7 (1998), pp. 56–74.

[29] K. V. Fernando and B. N. Parlett, *Accurate Singular Values and Differential* qd *Algorithms*, Numer. Math., 67 (1994), pp. 191–229.

[30] T. Friese, *Eine Mehrgitter-Methode zur Lösung des Eigenwertproblems der komplexen Helmholtzgleichung*, PhD thesis, Fachbereich Mathematik u. Informatik, Freie Universität Berlin, 1999. http://www.diss.fu-berlin.de/1999/13/.

[31] A. A. Goldin, *Software for Particle Size Distribution Analysis in Photon Correlation Spectroscopy*, software documentation, Alango Science Division, http://www.alango.com/science/WhitePapers/dynals1/dynals100.htm, 2002.

[32] G. H. Golub, A. Hoffman, and G. W. Stewart, *A Generalization of the Eckart-Young-Mirsky Matrix Approximation Theorem*, Linear Algebra Appl., 88/89 (1987), pp. 317–327.

[33] G. H. Golub and W. M. Kahan, *Calculating the Singular Values and Pseudo-inverse of a Matrix*, SIAM J. Num. Anal. Ser. B, 2 (1965), pp. 205–224.

[34] G. H. Golub and C. F. V. Loan, *An Analysis of the Total Least Squares Problem*, SIAM J. Numer. Anal., 17 (1980), pp. 883–893.

[35] ——, *Matrix Computations*, The John Hopkins University Press, Baltimore and London, second ed., 1989.

[36] G. H. Golub and H. Zha, *The Canonical Correlations of Matrix Pairs and Their Numerical Computation*, in Linear algebra for signal processing. Papers based on lectures presented at the IMA workshop, held at IMA, University of Minnesota, Minneapolis, MN, USA, April 6-10, 1992, A. Bojanczyk, ed., vol. 69 of IMA Vol. Math. Appl., New York, 1995, Springer-Verlag, pp. 27–49.

[37] C. W. Groetsch, *Inverse Problems in the Mathematical Sciences*, Braunschweig, Vieweg, 1993.

[38] W. Gropp, E. Lusk, and A. Skjellum, *Using MPI: Portable Parallel Programming with the Message Passing Interface*, Scientific and Engineering Computation Series, The MIT Press, Cambridge, London, second ed., 1999.

[39] B. Grosser and B. Lang, *An $\mathcal{O}(n^2)$ Algorithm for the Bidiagonal SVD*, Linear Algebra Appl., 358 (2003), pp. 45–70.

[40] L. Grubišić, *Numerical Computation of the Partial Eigenvalue Problem for Symmetric Matrices (Numeričko računanje parcijalnog problema vlastitih vrijednosti za simetrične matrice)*, master's thesis, Department of Mathematics, University of Zagreb, 2001.

[41] W. Hackbusch, *On The Computation of Approximate Eigenvalues and Eigenfunctions of Elliptic Operators by Means of a Multi-grid Method*, SIAM J. Numer. Anal., 16 (1979), pp. 201–215.

[42] ——, *Multi-grid Methods and Applications*, vol. 4 of Springer series in computational mathematics, Springer, Berlin, 1985.

[43] V. Hari and K. Veselić, *On Jacobi Methods for Singular Value Decompositions*, SIAM J. Sci. Stat., 8 (1987), pp. 741–754.

[44] M. Hegland, *A Distribution Independent Algorithm for the Reduction to Tridiagonal Form Using One-Sided Rotations*, in Proceedings of the IEEE First International Conference on Algorithms And Architectures for Parallel Processing, vol. 1, held in Brisbane, Australia, 1995, pp. 286–289.

[45] M. Hegland, M. Kahn, and M. Osborne, *A Parallel Algorithm for the Reduction to Tridiagonal Form for Eigendecomposition*, SIAM J. Sci. Comput., 21 (1999), pp. 987–1005.

[46] V. Heuveline and C. Bertsch, *On Multigrid Methods for the Eigenvalue Computation of Nonselfadjoint Elliptic Operators*, East-West J. Numer. Math., 8 (2000), pp. 275–297.

[47] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia, 1996.

[48] M. E. Hochstenbach, *A Jacobi-Davidson Type SVD Method*, Siam J. Sci. Comput., 23 (2001), pp. 606–628.

[49] R. A. Horn and C. R. Johnson, *Matrix analysis. (Matrichnyj analiz)*, Mir, Moskva, 1989.

[50] G. Howell, C. Fulton, J. W. Demmel, S. J. Hammarling, and K. Marmol, *Cache Efficient Bidiagonalization Using BLAS 2.5 Operators*, LAPACK Working Note #174, 2006.

[51] S. V. Huffel and J. Vandewalle, *The Total Least Squares Problem: Computational Aspects and Analysis*, SIAM Publications, Philadelphia, 1991.

[52] I. C. F. Ipsen, *Relative Perturbation Bounds for Matrix Eigenvalues and Singular Values*, vol. 7 of Acta Numerica, Cambridge University Press, Cambridge, 1998, pp. 151–201.

[53] C. G. J. Jacobi, *Über ein Leichtes Verfahren Die in der Theorie der Sacularstorungen Vorkommendern Gleichungen Numerisch Aufzulosen*, Crelle's J., 30 (1846), pp. 51–94.

[54] C. Jordan, *Mémoire sur les formes bilinéaires*, Journal de Mathéantiques Purse et Appliquées, Deuxiéme Sérei, 19 (1874), pp. 35–54.

[55] Y. M. Kadah and X. Hu, *Algebraic Reconstruction for Magnetic Resonance Imaging Under B_0 Inhomogeneity*, IEEE Transactions on Medical Imaging, 17 (1998), pp. 362–370.

[56] W. M. Kahan, *Inclusion Theorems for Clusters of Eigenvalues of Hermitian Matrices*, Tech. Report No. CS42, Computer Science Dept., University of Toronto, 1967.

[57] A. V. Knyazev, *Toward the Optimal Preconditioned Eigensolver: Locally Optimal Block Preconditioned Conjugate Gradient Method*, SIAM J. Sci. Comput., 23 (2001), pp. 517–541.

[58] A. V. Knyazev and K. Neymeyr, *A Geometric Theory for Preconditioned Inverse Iteration. III: A Short and Sharp Convergence Estimate for Generalized Eigenvalue Problems*, Linear Algebra Appl., 358 (2003), pp. 95–114.

[59] ——, *Efficient Solution of Symmetric Eigenvalue Problems Using Multigrid Preconditioners in the Locally Optimal Block Conjugate Gradient Method*, ETNA, Electron. Trans. Numer. Anal, 15 (2003), pp. 38–55.

[60] E. Kogbetliantz, *Diagonalization of General Complex Matrices as a New Method for Solution of Linear Equations*, in Proc. Intern. Congr. Math. Amsterdam 2, 1954, pp. 356–357.

[61] ——, *Solutions of Linear Equations by Diagonalization of Coefficient Matrices*, Quart. Appl. Math, 13 (1955), pp. 123–132.

[62] S. Kurepa, *Functional Analysis: Elements of Operator Theory (Funkcionalna analiza: elementi teorije operatora)*, Školska knjiga, Zagreb, second ed., 1990.

[63] C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems*, Prentice-Hall Series in Automatic Computation, Prentice-Hall, 1974.

[64] C.-K. Li and R. Mathias, *The Lidskii-Mirsky-Wielandt Theorem - Additive and Multiplicative Versions*, Numer. Math., 81 (1999), pp. 377–413.

[65] R. C. Li, *On Eigenvalues of a Rayleigh Quotient Matrix*, Linear Algebra Appl., 169 (1992), pp. 249–255.

[66] ——, *Relative Perturbation Theory: (I) Eigenvalue and Singular Value Variations*, LAPACK Working Note #84, 1994.

[67] ——, *Relative Perturbation Theory: (II) Eigenspace and Singular Subspace Variations*, LAPACK Working Note #85, 1994.

[68] J. Mandel and S. F. McCormick, *A Multilevel Variational Method for $Au = \lambda Bu$ on Composite Grids*, J. Comput. Phys., 80 (1989), pp. 442–452.

[69] R. Mathias, *Spectral Perturbation Bounds for Positive Definite Matrices*, SIAM J. Matrix. Anal. Appl., 18 (1997), pp. 959–980.

[70] ——, *Quadratic Residual Bounds for the Hermitian Eigenvalue Problem*, SIAM J. Matrix Anal. Appl., 19 (1998), pp. 541–550.

[71] S. F. McCormick, *Multilevel Adaptive Methods for Elliptic Eigenproblems: a Two-Level Convergence Theory*, SIAM J. Numer. Anal., 31 (1994), pp. 1731–1745.

[72] C. S. McGoldrick, W. J. Dowling, and A. Bury, *Image Coding Using the Singular Value Decomposition and Vector Quantization*, in Proceedings of the Fifth International Conference on Image Processing and its Applications, vol. 410 of IEE Conference Publication, IEEE, 1995, pp. 296–300.

[73] A. Meyer, B. Döhler, and L. Skurt, *Simultane Algorithmen für großdimensionierte Eigenwertprobleme und ihre Anwendung auf das Schwingungsproblem*, Wissenschaftliche Schriftenreihe der Technische Hochschule Karl-Marx-Stadt 8/1983, Technische Hochschule Karl-Marx-Stadt, 1983.

[74] I. Mirošević, *Spectral Graph Partitioning and Application to Knowledge Extraction (Spektralno particioniranje grafa i primjena na ekstrakciju znanja)*, master's thesis, Department of Mathematics, University of Zagreb, 2005.

[75] K. Neymeyr, *A Geometric Theory for Preconditioned Inverse Iteration. I: Extrema of the Rayleigh Quotient*, Linear Algebra Appl., 322 (2001), pp. 61–85.

[76] ——, *Solving Mesh Eigenproblems with Multigrid Efficiency*, in Numerical Methods for Scientific Computing. Variational problems and applications, E. Heikkola, Y. Kuznetsov, P. Neittaanmäki, and O. Pironneau, eds., Barcelona, 2003, CIMNE.

[77] C. C. Paige, *Bidiagonalization of Matrices and Solution of Linear Equations*, SIAM J. Numer. Anal., 11 (1974), pp. 197–209.

[78] B. N. Parlett, *The Symmetric Eigenvalue Problem*, Prentice-Hall Series in Computational Mathematics, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1980.

[79] R. Penrose, *A Generalized Inverse for Matrices*, Proc. Camb. Philos. Soc., 51 (1955), pp. 406–413.

[80] R. Ralha, *A New Algorithm for Singular Value Decompositions*, in Proceedings of the 3rd Euromicro Workshop on Parallel and Distributed Processing, IEEE Computer Society Press, 1994, pp. 240–244.

[81] ——, *One-sided Reduction to Bidiagonal Form*, Linear Algebra Appl., 358 (2003), pp. 219–238 .

[82] R. Ralha and A. Mackiewicz, *An Efficient Algorithm for the Computation of Singular Values*, in Proceedings of the III International Congress on Numerical Methods in Engineering, M. Doblaré, J. M. Correas, L. Gaverte, and M. Pastor, eds., SEMNI (Sociedad Española de Métodos Numéricos en Ingenieria), 1996, pp. 1371–1380.

[83] B. W. Rust, *Truncating the Singular Value Decomposition for Ill-Posed Problems*, Tech. Report NISTIR 6131, National Institute of Standards and Technology, 1998.

[84] ——, *Parameter Selection for Constrained Solutions to Ill-Posed Problems*, in Modeling the Earth's Systems: Physical to Infrastructural, E. J. Wegman and Y. M. Martinez, eds., vol. 32 of Computing Science and Statistics, 2000.

[85] Y. Saad, *On the Rates of Convergence of the Lanczos and the Block-Lanczos Methods*, SIAM J. Numer. Anal., 17 (1980), pp. 687–706.

[86] T. Schneider, *Homework 4.* http://www.gps.caltech.edu/∼tapio/acm118/Hw/hw4.pdf. Homework task for the course "Methods in Applied Statistics and Data Analysis".

[87] ——, *Solutions to Homework 4.* http://www.gps.caltech.edu/∼tapio/acm118/Hw/hw4sol.pdf. Solutions to the homework task for the course "Methods in Applied Statistics and Data Analysis".

[88] G. L. G. Sleijpen and H. A. van der Vorst, *A Jacobi-Davidson Iteration Method for Linear Eigenvalue Problems*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 401–425.

[89] G. W. Stewart and J. G. Sun, *Matrix Perturbation Theory*, Computer Science and Scientific Computing, Academic Press, Inc., Boston etc., 1990.

[90] P. A. Wedin, *On Angles Between Subspaces of a Finite Dimensional Inner Product Space*, in Matrix Pencils, vol. 973 of Lect. Notes Math., Proc. Conf. Pite Havsbad/Swed. 1982, 1983, pp. 263–285.

[91] P. R. Willems, B. Lang, and C. Vömel, *Computing the Bidiagonal SVD Using Multiple Relatively Robust Representations*, LAPACK Working Note #166, Computer Science Division UC Berkeley, 2005.

[92] G. M. Wing, *A Primer on Integral Equations of the First Kind*, SIAM Publications, Philadelphia, PA, 1991.

[93] P. Xiao and R. E. Imhof, *Inverse Method Analysis in Opto-Thermal Skin Measurements*, Proceedings of SPIE, 3601 (1999), pp. 340–347.

[94] J.-F. Yang and C.-L. Lu, *Combined Techniques of Singular Value Decomposition and Vector Quantization for Image Coding*, IEEE Trans. Image Process., 4 (1995), pp. 1141–1146.

[95] M. K. S. Yeung, J. Tegnér, and J. J. Collins, *Reverse Engineering Gene Networks Using Singular Value Decomposition and Robust Regression*, Proceedings of the National Academy of Science of the United States of America, 99 (2002), pp. 6163–6168.

[96] L. Zhang, *Singular Value Decomposition with Application on Network Traffic Modeling*, tech. report, Department of Statistics and Operation Research University of North Carolina, Chapel Hill, 2004.

Appendix A

Summary

This thesis is dealing with two major topics in numerical linear algebra: the singular value decomposition (SVD) and the eigenvalue problem. Two new algorithms are proposed: one for finding the singular value decomposition, and one for solving the partial eigenvalue problem of a symmetric positive definite matrix. The one–sided bidiagonalization algorithm proposed by Barlow is analyzed, and is proven to be numerically stable. The bidiagonalization constitutes the first step in computing the SVD. Next, the block version of the one–sided bidiagonalization is proposed, which increases its efficiency and retains numerical stability. The parallel version of the same algorithm was also studied and numerical tests show that it is faster than the parallel algorithm implemented in the ScaLAPACK software package. One–sided bidiagonalization turned out to be competitive to other standard bidiagonalization algorithms, and there are several applications where it can be successfully applied. Another algorithm presented in this thesis is the new subspace method for computing eigenvectors corresponding to the several smallest eigenvalues of a symmetric positive definite matrix. The name of the method is multispace, and it is a combination of multigrid approach and of two very well known subspace methods: inverse iteration and the block Lanczos method. The new multigrid approach is designed to speed up the convergence of slow converging inverse iteration. A convergence rate for multispace is also presented, proving that the whole process converges to an invariant subspace. In addition to the algorithms, a new perturbation result for singular value approximations from subspaces is presented. The new result represents a measure for relative errors in singular values expressed by terms involving angles of appropriate subspaces.

Appendix B

Sažetak

Ova disertacija bavi se dvjema glavnim temama numeričke linearne algebre: dekompozicijom singularnih vrijednosti (SVD) i svojstvenim problemom. Predstavljena su dva nova algoritma: jedan za računanje dekompozicije singularnih vrijednosti, i jedan za rješavanje parcijalnog svojstvenog problema za simetričnu pozitivno definitnu matricu. Jednostrana bidijagonalizacija koju je predložio Barlow analizirana je, i pokazano je da je ona numerički stabilna. Bidijagonalizacija predstavlja prvi korak u računanju SVD-a. Zatim je predložena blok verzija jednostrane bidijagonalizacije, koja povečava njenu efikasnost i zadržava numeričku stabilnost. Paralelna verzija istog algoritma je također proučavana, a numerički testovi pokazuju da je ona brža od paralelnog algoritma implementiranog u softverskom paketu ScaLAPACK. Ispostavilo se da je jednostrana bidijagonalizacija konkurentna ostalim standardnim bidijagonalizacijskim algoritmima, i postoji nekoliko primjena u kojima se može uspješno primijeniti. Drugi algoritam opisan u ovoj disertaciji je nova potprostorna metoda za računanje svojstvenih vektora koji pripadaju nekolicini najmanjih svojstvenih vrijednosti simetrične pozitivno definitne matrice. Metoda se zove multispace, i ona je kombinacija multigrid pristupa i dviju dobro poznatih potprostornih metoda: inverznih iteracija i blok Lanczos-ove metode. Novi multigrid pristup je dizajniran tako da ubrza konvergenciju sporo konvergirajućih inverznih iteracija. Brzina konvergencije multispace-a je također prezentirana, čime se dokazuje da cijeli proces konvergira ka invarijantnom potprostoru. Osim algoritama, prezentiran je i novi perturbacijski rezultat za aproksimacije singularnih vrijednosti iz potprostora. Novi rezultat predstavlja mjeru relativne greške u singularnim vrijednostima izraženu pomoću funkcije kuta između pogodno izabranih potprostora.

Appendix C

Curriculum Vitæ

I was born on July 8, 1973 in Zagreb, where I have finished elementary school. I continued my education in High School for Mathematics and Informatics in Zagreb, where I graduated in 1992. The same year I enrolled the studies for engineer of mathematics at Department of Mathematics, University of Zagreb. In the third year of the studies I chose courses in applied mathematics, and in the fourth year I won scholarship of the concern "Agrokor". In October 1997 I successfully graduated on the topic "Maxwell Equations", and started postgraduate studies of mathematics. I work as a teaching assistant at Department of Mathematics, University of Zagreb since February 1998. My work field is numerical linear algebra, and I am involved in the development of numerical algorithms, numerical analysis and perturbation theory. I am also writing software and testing parallel algorithms. Next to my scientific work, I am engaged in lectures for undergraduate students, mostly in subjects connected to the numerical mathematics and computer science. In December 2001 I successfully defended my master thesis on the topic "Iterative Methods for Solving Linear Systems". I also attended two summer schools for postgraduate students: summer school on the subject of "Function analysis, differential equations, numerical analysis" held in September 2002 in Budapest, and summer school on the subject of "Cell biology and mathematical modelling" held in June 2004 in Hvar. I also participated in 9 conferences in Croatia and abroad, and had a presentation in 6 of them, as well as in scientific meetings and seminars. Some of my results are published in 3 reviewed articles. Two other articles are in the reviewing process, and one of them is already accepted for publication.