# ADVANCED LANCZOS METHODS FOR LARGE-SCALE MATRIX PROBLEMS

SARAH WILLEMIEN GAAF

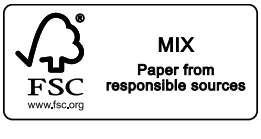
This work is financially supported by the Netherlands Organization for Scientific Research (NWO) through the Vidi grant 639.032.223 with title "Innovative methods for large matrix problems".


Nederlandse Organisatie
voor Wetenschappelijk Onderzoek



A catalogue record is available from the Eindhoven University of Technology Library.

ISBN: 978-90-386-4357-1




Printed by Gildeprint, Enschede, The Netherlands

# ADVANCED LANCZOS METHODS FOR LARGE-SCALE MATRIX PROBLEMS

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de Technische Universiteit Eindhoven, op gezag van de rector magnificus prof.dr.ir. F.P.T. Baaijens, voor een commissie aangewezen door het College voor Promoties, in het openbaar te verdedigen op maandag 16 oktober 2017 om 16:00 uur

door

SARAH WILLEMIEN GAAF

geboren te Leeuwarden

Dit proefschrift is goedgekeurd door de promotoren en de samenstelling van de promotiecommissie is als volgt:

| | |
|---|---|
| voorzitter: | prof.dr. J.J. Lukkien |
| promotor: | prof.dr.ir. B. Koren |
| copromotor: | dr. M.E. Hochstenbach |
| leden: | prof.dr. V. Simoncini (Università di Bologna) |
| | dr. E. Jarlebring (KTH Royal Institute of Technology) |
| | dr.ir. M. van Gijzen (Technische Universiteit Delft) |
| | prof.dr.ir. E.H. van Brummelen |
| | prof.dr. W.H.A. Schilders |

Het onderzoek of ontwerp dat in dit proefschrift wordt beschreven is uitgevoerd in overeenstemming met de TU/e Gedragscode Wetenschapsbeoefening.

# CONTENTS


1 INTRODUCTION 1
1.1 Eigenvalue and singular value problems . . . . . 1
1.1.1 The standard eigenvalue problem . . . . . 2
1.1.2 The singular value decomposition . . . . . 2
1.1.3 Non-linear eigenvalue problems . . . . . 3
1.2 Lanczos methods . . . . . 3
1.2.1 The standard Lanczos method . . . . . 4
1.2.2 The Lanczos bidiagonalization method . . . . . 5
1.2.3 The two-sided Lanczos method . . . . . 6
1.3 Outline . . . . . 7
1.3.1 Chapter 2 . . . . . 7
1.3.2 Chapter 3 . . . . . 8
1.3.3 Chapter 4 . . . . . 8
1.4 Notation . . . . . 9
1.5 Hardware and software . . . . . 9

2 PROBABILISTIC BOUNDS FOR THE MATRIX CONDITION NUMBER WITH EXTENDED LANCZOS BIDIAGONALIZATION 11
2.1 Introduction . . . . . 11
2.1.1 Contributions of this chapter to the problem . . . 12
2.1.2 Overview of the chapter . . . . . 13
2.2 Extended Lanczos bidiagonalization . . . . . 13
2.3 The special structure of the generated H and K matrices 17
2.4 Polynomials arising in extended Lanczos bidiagonalization 22
2.5 Probabilistic bounds for the condition number . . . . . 26
2.6 Other condition estimators . . . . . 30
2.6.1 Probabilistic condition estimators based on the 2-norm . . . . . 31
2.6.2 Condition estimators based on other norms . . . 33
2.7 Numerical experiments . . . . . 34
2.8 Final considerations . . . . . 41

3 APPROXIMATING LEADING SINGULAR TRIPLETS OF A MATRIX FUNCTION 43

3.1 Introduction . . . . . 43
3.1.1 Appearances of the norm of a matrix function . . 43
3.1.2 Contributions of this chapter to the problem . . . 44
3.1.3 Overview of the chapter . . . . . 45
3.2 Available techniques for estimating the norm . . . . . 46
3.3 Lanczos bidiagonalization . . . . . 48
3.4 Inexact Lanczos bidiagonalization . . . . . 49
3.4.1 A computable stopping criterion . . . . . 52
3.5 Approximation of $f(A)v$ and $f(A)^*u$ . . . . . 54
3.5.1 A computable inner stopping criterion . . . . . 55
3.6 Relaxing the inner solution accuracy . . . . . 56
3.6.1 Spectral properties of the approximate singular triplets . . . . . 57
3.6.2 Variable accuracy in the inner approximation . . 60
3.7 Practical implementation . . . . . 61
3.8 Numerical experiments . . . . . 64
3.8.1 Assessing the effectiveness of the inexact bidiagonalization . . . . . 66
3.8.2 Comparisons with the power method . . . . . 70
3.8.3 Numerical tests with the extended Krylov subspace 71
3.8.4 Numerical tests with variable accuracy . . . . . 73
3.9 Final considerations . . . . . 76

4 THE INFINITE BI-LANCZOS METHOD FOR NONLINEAR EIGENVALUE PROBLEMS 79
4.1 Introduction . . . . . 79
4.1.1 Contributions of this chapter to the problem . . . 80
4.1.2 Overview of the chapter . . . . . 81
4.2 Infinite-dimensional reformulation . . . . . 82
4.2.1 The nonlinear eigenvalue problem and the operator $\mathbf{A}$ . . . . . 82
4.2.2 Krylov subspace and infinite-dimensional vector representations . . . . . 86
4.2.3 Scalar products and matrix-vector products . . . 88
4.3 Derivation of the infinite bi-Lanczos method . . . . . 90
4.3.1 The bi-Lanczos method for standard eigenvalue problems . . . . . 90
4.3.2 The infinite bi-Lanczos method . . . . . 92
4.3.3 Computational representation of the infinite vectors 95

4.3.4 Complexity considerations and implementation . 97
4.4 Numerical experiments . . . . . 100
4.4.1 A second order delay-differential equation . . . . 100
4.4.2 A benchmark problem representing an electromagnetic cavity . . . . . 104
4.5 Final considerations . . . . . 105

5 CONCLUSION 109
5.1 Overview of the chapters . . . . . 110
5.1.1 Overview of chapter 2 . . . . . 110
5.1.2 Overview of chapter 3 . . . . . 110
5.1.3 Overview of chapter 4 . . . . . 111
5.2 Discussion and outlook . . . . . 111

BIBLIOGRAPHY 115

BIBLIOGRAPHY 115

SUMMARY 123

CURRICULUM VITAE 125

PUBLICATIONS 127

ACKNOWLEDGMENTS 129

# 1 INTRODUCTION

Linear algebra is present across all sciences as its elementary and elegant theory is often the basis of scientific calculations. Problems such as eigenvalue and singular value problems arise in many scientific and engineering areas. Examples of disciplines are physics, economics, chemistry, oceanography, control theory, dynamical systems, mechanics, signal and image processing, and so forth. The matrices at the core of these problems are frequently of large dimension and sparse, i.e., many of its elements equal zero. Since direct methods may not be feasible for large-scale problems, iterative methods are developed to solve the eigenvalue problems and sparse linear systems of large dimensions. Iterative methods have been extensively studied in the last decades, see for instance [1, 70]. Subspace methods, a widely used subclass of iterative methods, are an attractive approach considering the matrix-vector operations that can be computed efficiently. Krylov methods are a well-known and broadly used variant of subspace methods, projecting the problem onto a subspace of low dimension. While the computing time is reduced, and the memory usage is diminished, the solution can be approximated solving a small-scale problem. This dissertation presents three new versions of the Lanczos method, a specific type of Krylov subspace method, that are developed to tackle eigenvalue and singular value problems involving large-scale matrices.

## 1.1 EIGENVALUE AND SINGULAR VALUE PROBLEMS

It is important in many scientific problems to obtain specific knowledge about the matrices that are involved in the problem description. The eigenvalues and eigenvectors of a matrix are an example of important properties of a matrix. We start with a description of three matrix problems that are fundamental for this dissertation.

### 1.1.1 *The standard eigenvalue problem*

The standard eigenvalue problem corresponding to a matrix $A \in \mathbb{C}^{n\times n}$ is to find eigenvalues $\lambda \in \mathbb{C}$ and eigenvectors $x \in \mathbb{C}^n \backslash \{0\}$ such that

$$Ax = \lambda x.$$

If the matrix $A$ is diagonalizable, i.e., a matrix that is similar to a diagonal matrix, it can be represented in terms of its eigenvectors and eigenvalues also known as the eigendecomposition. Depending on the available properties of the matrix $A$, the problem can be formulated more specifically and the appropriate numerical methods to solve the problem may be chosen accordingly. For example, if $A$ is Hermitian ($A^* = A$, where $A^*$ is the conjugate transpose of $A$), then we are dealing with a Hermitian eigenvalue problem. In that case, the eigenvalues of $A$ are real, and the Lanczos method can be used to approximate its eigenvalues and vectors. For non-Hermitian $A$ the eigenvalues will be complex and the Arnoldi method is one of the standard methods to solve the eigenvalue problem.

### 1.1.2 *The singular value decomposition*

An other way to factorize a matrix is by the singular value decomposition. For any matrix $A \in \mathbb{C}^{m\times n}$ the singular value decomposition is defined as $A = X\Sigma Y^*$, where $X$ is an $m \times m$ unitary matrix ($X^{-1} = X^*$), $Y$ an $n \times n$ unitary matrix, and $\Sigma$ an $m \times n$ diagonal matrix containing non-negative real numbers (the singular values of $A$) on the diagonal. The orthonormal columns of $X$ and $Y$ contain the left and right singular vectors of $A$, respectively. As for the eigenvalue problem, for many applications it is important to find the singular values and corresponding singular vectors. However, for large matrices it may not be feasible or unattractive to compute the singular value decomposition itself. Iterative methods such as Lanczos bidiagonalization are then used to approximate the singular triplets (the singular values and the corresponding left and right singular vectors) of interest.

### 1.1.3 *Non-linear eigenvalue problems*

The standard eigenvalue problem can be defined in a more general way by writing it as finding eigenvalues $\lambda \in \mathbb{C}$ and eigenvectors $x \in \mathbb{C}^n \backslash \{0\}$ such that $M(\lambda)x := (A - \lambda I)x = 0$. Other choices for the operator $M$ lead to more general problems, for example $M(\lambda) = A - \lambda B$ corresponds to the generalized eigenvalue problem. We consider nonlinear eigenvalue problems of the form: find $\lambda \in \mathbb{C}$ and $x \in \mathbb{C}^n \backslash \{0\}$ such that $M(\lambda)x = 0$, for

$$M(\lambda) = g_0(\lambda)A_0 + \ldots + g_m(\lambda)A_m,$$

where $A_i \in \mathbb{C}^{n \times n}$ are constant coefficient matrices and $g_i$ are functions such as polynomials, rational functions, or the exponential function. Note that the nonlinearity is contained in $\lambda$; the eigenvector $x$ is only linearly involved in the problem description. As an example, the delay differential equation (DDE) gives rise to such a nonlinear eigenvalue problem. Consider therefore the linear time-invariant DDE with several discrete delays:

$$\dot{x}(t) = A_0 x(t) + \sum_{i=1}^{m} A_i x(t - \tau_i),$$

for $A_0 \ldots, A_m \in \mathbb{C}^{n \times n}$, and $\tau_1, \ldots, \tau_m \in \mathbb{R}$. The solution $x(t) = e^{-\lambda t} x_0$ gives rise to the nonlinear eigenvalue problem

$$M(\lambda)x = (\lambda I - A_0 - \sum_{i=1}^{m} A_i e^{-\tau_i \lambda})x = 0.$$

Various methods such as Newton's method, contour integration, or linearization are used to solve these types of problems. See [33] for a recent overview on nonlinear eigenvalue problems.

## 1.2 LANCZOS METHODS

To solve the large-scale problems as the ones presented in the previous section, iterative methods are used. In this dissertation we develop advanced Lanczos methods, a specific type of Krylov subspace methods. Cornelius Lanczos[1] (1893-1974) was a Jewish Hungarian mathematician and physicist. Apart from inventing the Lanczos method and the $\tau$

1 See the series of video tapes produced in 1972, in which Cornelius Lanczos tells about his life as mathematician, made available online: `guettel.com/lanczos/`.

method, he was the assistant of Albert Einstein, and he (re-)discovered the fast Fourier transform and the singular value decomposition. Below we describe three well-known Lanczos methods that form the basis of the methods that are developed in this dissertation. See for instance [1] for a more detailed description of the presented (and related) methods.

### 1.2.1 *The standard Lanczos method*

The aim of the Lanczos method is to approximate eigenvalues $\lambda \in \mathbb{R}$ and eigenvectors $x \in \mathbb{C}^n \backslash \{0\}$ of the Hermitian eigenvalue problem $Ax = \lambda x$, for the Hermitian matrix $A$. With a properly chosen starting vector $v_1 \in \mathbb{C}^n$, the method builds an orthogonal basis $V_j = [v_1, \cdots, v_j]$ of the Krylov subspace $\mathcal{K}_j(A, v_1) := \text{span}\{v_1, Av_1, \ldots, A^{j-1}v_1\}$. The procedure can be represented as follows

$$v_1 \xrightarrow{A} v_2 \xrightarrow{A} v_3 \xrightarrow{A} \ldots,$$

where the orthogonalization of each new vector with respect to the previously constructed vectors is not shown. After $j$ steps of the iteration we obtain the recursion relation

$$AV_j = V_j T_j + \beta_j v_{j+1} e_j^*,$$

where $V_j^* v_{j+1} = 0$. The matrix

$$T_j = \begin{bmatrix} \alpha_1 & \beta_1 & & \\ \beta_1 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \beta_{j-1} \\ & & \beta_{j-1} & \alpha_j \end{bmatrix}$$

is the representation of $A$ in the orthogonal basis $V_j$. The matrix $A$ is accessed only in the form of matrix-vector multiplications. Furthermore, the method can be performed using short (three-term) recurrences. The Lanczos method is closely related to the power method, where the main difference is that the eigenvalue approximation of the Lanczos method is based on all previously computed vectors instead of only the current vector. In fact, if $T_j s_i = \theta_i s_i$ is an eigensolution of $T_j$, then the Ritz value $\theta_i$ and the Ritz vector $x_i = V_j s_i$ are an approximation to an eigenpair of $A$. As soon as one eigenvalue converges the orthogonality of the

computed vectors is lost; a known drawback of the Lanczos method. One can either reorthogonalize (fully, selectively or locally), or discard spurious eigenvalues.

### 1.2.2 *The Lanczos bidiagonalization method*

The Golub–Kahan–Lanczos bidiagonalization, first described in [31] (and abbreviated as Lanczos bidiagonalization throughout the rest of this dissertation), is among the most used strategies for approximating selected singular triplets $(\sigma, u, v)$ for $\sigma \in \mathbb{R}$ and $u, v \in \mathbb{C}^n \backslash \{0\}$, of a given matrix $A$, such that $Av = \sigma u$ and $A^*u = \sigma v$. With a properly chosen starting vector $v_1 \in \mathbb{C}^n$ and defining $u_1 := Av_1$, the method builds two orthogonal bases, $V_j = \begin{bmatrix} v_1, v_2, \cdots, v_j \end{bmatrix}$ and $U_j = \begin{bmatrix} u_1, u_2, \cdots, u_j \end{bmatrix}$ of the Krylov subspaces $\mathcal{K}_j(A^*A, v_1)$ and $\mathcal{K}_j(AA^*, u_1)$, respectively. The procedure can be represented as follows

$$v_1 \xrightarrow{A} u_1 \xrightarrow{A^*} v_2 \xrightarrow{A} u_2 \xrightarrow{A^*} v_3 \ldots .$$

The orthogonal projection of the vectors (not shown in the above representation) is done in such a way that $V_j^* V_j = I$ and $U_j^* U_j = I$. After $j$ steps of the iteration we obtain the recursion relations

$$\begin{aligned} AV_j &= U_j B_j, \\ A^*U_j &= V_j B_j^* + \beta_j v_{j+1} e_j^*, \end{aligned}$$

where $V_j^* v_{j+1} = 0$, and $A$ is reduced to the bidiagonal matrix (nonzero main diagonal and first superdiagonal)

$$B_j = \begin{bmatrix} \alpha_1 & \beta_1 & & & \\ & \alpha_2 & \beta_2 & & \\ & & \ddots & \ddots & \\ & & & \alpha_{j-1} & \beta_{j-1} \\ & & & & \alpha_j \end{bmatrix}.$$

If $(\mu_i, c_i, d_i)$ is a singular triplet of $B_j$, then $(\mu_i, U_j c_i, V_j d_i)$ is an approximation to a singular triplet of $A$. As in the Lanczos method, the matrix $A$ is involved only via matrix-vector multiplications and the method is based on short recurrences. The Lanczos bidiagonalization method is

mathematically equivalent to the Lanczos method applied to the matrix $A^*A$ (or to the matrix $AA^*$) and also to the Lanczos method applied to

$$H(A) = \begin{bmatrix} 0 & A \\ A^* & 0 \end{bmatrix}.$$

Therefore, the same convergence properties hold as for the Lanczos method.

### 1.2.3 *The two-sided Lanczos method*

The two-sided Lanczos method, also called the non-Hermitian Lanczos method or the bi-Lanczos method, is an oblique projection method for non-Hermitian matrices. It tackles the non-Hermitian eigenvalue problem: find eigenvalues $\lambda \in \mathbb{C}$ and right and left eigenvectors, $x, y \in \mathbb{C}^n \backslash \{0\}$, respectively, such that

$$\begin{aligned} Ax &= \lambda x, \\ y^*A &= \lambda y^*. \end{aligned}$$

With two properly chosen starting vectors $q_1, p_1 \in \mathbb{C}^n$, the method builds two bi-orthogonal bases $Q_j = \begin{bmatrix} q_1, \cdots, q_j \end{bmatrix}$ and $P_j = \begin{bmatrix} p_1, \cdots, p_j \end{bmatrix}$ of the Krylov subspaces $\mathcal{K}_j(A, q_1)$ and $\mathcal{K}_j(A^*, p_1)$, respectively. The bases are constructed such that $Q_j$ and $P_j$ are bi-orthogonal, meaning that $P_j^* Q_j = I$ (note that $Q_j^* Q_j \neq I \neq P_j^* P_j$). This method allows for a freedom in scaling the vectors differently, such that $P_j^* Q_j$ is a diagonal matrix. The simultaneously built subspaces have the advantage to cover the approximation of both the left and right eigenvectors. The process can be represented as follows

$$\begin{aligned} q_1 &\xrightarrow{A} q_2 \xrightarrow{A} q_3 \xrightarrow{A} \dots, \\ p_1 &\xrightarrow{A^*} p_2 \xrightarrow{A^*} p_3 \xrightarrow{A^*} \dots, \end{aligned}$$

where the bi-orthogonalization of each new vector with respect to the previously constructed vectors of the other basis is not shown. It is clear from this representation that the method asks for the availability of the

matrix-vector operations $Aq$ and $A^*p$. After $j$ steps of the iteration we obtain the recursion relation

$$\begin{aligned} AQ_j &= Q_j T_j + \beta_j q_{j+1} e_j^*, \\ A^* P_j &= P_j T_j^* + \bar{\gamma}_{j+1} p_{j+1} e_j^*, \\ P_j^* Q_j &= I, \end{aligned}$$

where $Q_j^* p_{j+1} = 0$ and $P_j^* q_{j+1} = 0$. The non-Hermitian tridiagonal matrix

$$T_j = \begin{bmatrix} \alpha_1 & \gamma_2 & & \\ \beta_2 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \gamma_j \\ & & \beta_j & \alpha_j \end{bmatrix},$$

is the representation of $A$ in the Lanczos bases. If $T_j z_i = \theta_i z_i$ and $T_j^* w_i = \theta_i w_i$ denote an eigensolution of $T_j$, then the eigenvalues of $A$ are approximated by the eigenvalues $\theta_i$ of $T_j$, and the approximate eigenvectors are defined as $x_i = Q_j z_i$ (right) and $y_i = P_j w_i$ (left). As the Lanczos methods presented above, also this method can be performed using short recurrences, namely two three-term recurrences per iteration. However, since the method is not based on orthogonal transformations, the method suffers from numerical instability and there is a risk of breakdown.

## 1.3 OUTLINE

This dissertation contains three research chapters. In each of the chapters a novel Lanczos method is developed to solve a particular large-scale matrix problem. We give a short overview of the subjects covered in the following chapters.

### 1.3.1 *Chapter 2*

In chapter 2 we focus on the matrix condition number. Reliable estimates for the condition number of a large, sparse, real matrix $A$ are important in many applications. We propose to approximate the condition number $\kappa(A)$ by approximating both the smallest and the largest

singular value of $A$. The Lanczos bidiagonalization method, see section 1.2.2, provides satisfying approximations for the largest singular value, but it is usually unsuitable for finding a good approximation to the smallest singular value. Therefore, we develop a new extended Lanczos bidiagonalization method, which turns out to be ideal for the simultaneous approximation of both the smallest and largest singular value of a matrix, providing a lower bound for the condition number. Moreover, the method yields probabilistic upper bounds for $\kappa(A)$. The user can select the probability with which the upper bound holds, as well as the ratio of the probabilistic upper bound and the lower bound. This chapter is based on [26].

### 1.3.2 *Chapter 3*

In chapter 3 we treat the problem of approximating the norm of a large-scale matrix function. More precisely, given a large square matrix $A$ and a sufficiently regular function $f$ so that $f(A)$ is well defined, we are interested in the approximation of the leading singular values and corresponding left and right singular vectors of $f(A)$, and hence in particular in the approximation of $\|f(A)\|$, where $\|\cdot\|$ is the matrix norm induced by the Euclidean vector norm. We assume that neither $f(A)$ nor $f(A)v$ can be computed exactly, and thus the Lanczos bidiagonalization, see section 1.2.2, cannot be applied. Therefore, we introduce a new *inexact* Lanczos bidiagonalization procedure, where the inexactness is related to the inaccuracy of the operations $f(A)v$, $f(A)^*v$. The lack of a true residual requires particular outer and inner stopping criteria that will be devised. This chapter is based on [28].

### 1.3.3 *Chapter 4*

Chapter 4 deals with nonlinear eigenvalue problems (NEP) as presented in section 1.1.3. We propose a new two-sided Lanczos method, inspired on the two-sided Lanczos method presented in section 1.2.3. Compared to the standard two-sided method, this new method implicitly works with matrices and vectors with infinite size. Particular (starting) vectors are used, enabling the method to carry out all computations efficiently with finite matrices and vectors. We specifically introduce a new way to represent infinite vectors that span the subspace

corresponding to the conjugate transpose operation for approximating the left eigenvectors. This chapter is based on [27].

The concluding chapter, chapter 5, gives a brief overview of the issues that are discussed in this dissertation. Finally, various perspectives for further research are listed.

## 1.4 NOTATION

The following notation will be used throughout. Vectors are indicated by lowercase Roman letters, whereas capital Roman letters refer to matrices or operators. However, lowercase Roman letters will also be used to indicate sizes, dimensions, functions, and indices. Lowercase Greek letters represent scalars. Calligraphic uppercase letters (e.g., $\mathcal{B}$) refer to block-matrices. Letters in bold face correspond to matrices and vectors of infinite size. $\mathbb{R}$ and $\mathbb{C}$ stand for the real and complex numbers, respectively. The 2-norm of a matrix $A$ is denoted by $\|A\|$, where $\|\cdot\|$ is the matrix norm induced by the Euclidean vector norm, and it is defined as

$$\|A\| = \max_{0 \neq x \in \mathbb{C}^n} \frac{\|Ax\|}{\|x\|}, \tag{1.1}$$

where the Euclidean vector norm is defined as $\|x\| = (\sum_{i=1}^{n} |x_i|^2)^{\frac{1}{2}}$, for $x \in \mathbb{C}^n$. The vector $e_i$ indicates the $i$th column of the identity matrix of a given dimension. The conjugate transpose of a matrix $A$ will be denoted by $A^*$, and for symmetric matrices we use $A^T$ to indicate its transpose. We will use the MATLAB-like notation $[x; y]$ to denote the column vector

$$\begin{bmatrix} x \\ y \end{bmatrix}, \quad x \in \mathbb{C}^{n_x}, \quad y \in \mathbb{C}^{n_y}.$$

For $A \in \mathbb{C}^{n \times n}$, $\mathrm{spec}(A)$ denotes the set of its eigenvalues, $\mathrm{span}(A)$ refers to the space spanned by the columns of $A$, and $W(A) = \{z \in \mathbb{C} : z = (x^*Ax)/(x^*x), x \in \mathbb{C}^n \backslash \{0\}\}$ is its field of values.

## 1.5 HARDWARE AND SOFTWARE

The simulations were carried out with an implementation in MATLAB (R2012b and R2015b), and using a computer with an Intel Core i5-

3360M processor and 8 GB of RAM, unless mentioned otherwise. Various MATLAB codes that are used to produce the results in this dissertation are available online: `www.win.tue.nl/~hochsten/eigenvaluetools/` and `www.math.kth.se/~eliasj/src/infbilanczos/`.

# 2

# PROBABILISTIC BOUNDS FOR THE MATRIX CONDITION NUMBER WITH EXTENDED LANCZOS BIDIAGONALIZATION

*Adapted from [26]*

Reliable estimates for the condition number of a large, sparse, real matrix $A$ are important in many applications. To get an approximation for the condition number $\kappa(A)$, an approximation for the smallest singular value is needed. Standard Krylov subspaces are usually unsuitable for finding a good approximation to the smallest singular value. Therefore, we study extended Krylov subspaces which turn out to be ideal for the simultaneous approximation of both the smallest and largest singular value of a matrix. First, we develop a new extended Lanczos bidiagonalization method. With this method we obtain a lower bound for the condition number. Moreover, the method also yields probabilistic upper bounds for $\kappa(A)$. The user can select the probability with which the upper bound holds, as well as the ratio of the probabilistic upper bound and the lower bound.

## 2.1 INTRODUCTION

Let $A \in \mathbb{R}^{n\times n}$ be a large, nonsingular matrix. Let $A = X\Sigma Y^T$ be the singular value decomposition of $A$, as defined in section 1.1.2, where $X$ and $Y$ are $n \times n$ matrices with orthonormal columns containing the left and right singular vectors of $A$, respectively. Furthermore, $\Sigma$ is an $n \times n$ diagonal matrix with positive real entries containing the singular values of $A$ that are numbered in decreasing order: $\sigma_1 \geq \cdots \geq \sigma_n > 0$.

We are interested in the important problem of approximating the condition number of $A$,

$$\kappa(A) = \|A\| \, \|A^{-1}\| = \frac{\sigma_1}{\sigma_n},$$

where $\|\cdot\|$ stands for the 2-norm. The (Golub–Kahan–)Lanczos bidiagonalization method [31], as introduced in section 1.2.2, provides an approximation, a lower bound, for the maximum singular value $\sigma_1$ of $A$. In addition, an upper bound for the minimum singular value is ob-

tained, but this is usually a rather poor bound. To approximate the condition number, good approximations to $\sigma_n$ are needed.

### 2.1.1 *Contributions of this chapter to the problem*

This chapter has three contributions. First, we develop a new extended Lanczos bidiagonalization method. The method generates a basis for the extended Krylov subspace:

$$\mathcal{K}_{k+1,k+1}(A^TA, v) = \text{span}\{(A^TA)^{-k}v, \ldots, v, A^TAv, \ldots, (A^TA)^kv\}.$$

Extended Krylov subspace methods have been studied in the last 15 years by various authors [18, 49, 50, 56, 74]. The second contribution of this chapter is that we obtain simultaneously a lower bound for $\sigma_1$ and an upper bound for $\sigma_n$, which leads to a lower bound of good quality for $\kappa(A)$. Third, we obtain a probabilistic upper bound for the condition number. Probabilistic techniques have become increasingly popular, see, for instance, [16, 17, 38, 46, 59]. Whereas in [16, 38, 59] the power method is used, this paper is based on Krylov methods, as are the techniques in [17, 46, 59]. An important feature of the Lanczos bidiagonalization procedure is that the starting vector can be (and often is) chosen randomly. Therefore, the probability that this vector has a small component in the direction of the desired singular vector (relative to $1/\sqrt{n}$) is small. Another characteristic of the procedure is that during the bidiagonalization process polynomials implicitly arise. These two properties are exploited in [46] to obtain probabilistic upper bounds for $\sigma_1$.

In this chapter, we will expand the techniques from [46] to obtain both probabilistic lower bounds for $\sigma_n$ and probabilistic upper bounds for $\sigma_1$, leading to probabilistic upper bounds for $\kappa(A)$. These upper bounds hold with user-chosen probability: the user can select an $\varepsilon > 0$ such that the bounds hold with probability $1 - 2\varepsilon$, as well as a $\zeta > 1$ such that the ratio of the probabilistic upper bound and the lower bound for $\kappa(A)$ is less than $\zeta$. The method will adaptively perform a number of steps $k$ to accomplish this. Probabilistic condition estimators in [16] or [59] provide a ratio between the probabilistic upper bound and the lower bound, given a fixed $k$ and $\varepsilon$. The method presented here does not come with an analogous relation; however, the method we propose generally gives sharper bounds as is shown in section 2.7.

We stress the fact that the method developed in this chapter requires an (exact) LU decomposition. If this is unaffordable, there are alternative methods available that only need a preconditioner such as an inexact LU decomposition. The JDSVD method [44, 45] is one of these methods. However, because of the current state of both numerical methods and hardware, LU decompositions have increasingly become an option, sometimes even for rather large matrices.

The theory discussed in this chapter considers only real matrices. For general complex matrices the theory from this chapter to obtain probabilistic bounds needs to be adapted in a nontrivial way, and will be subject to future study.

### 2.1.2 *Overview of the chapter*

The rest of this chapter is organized as follows. In section 2.2 we introduce the extended Lanczos bidiagonalization method, and the special structure of the matrices obtained by this method are examined in section 2.3. Section 2.4 focuses on the Laurent polynomials arising in the procedure. In section 2.5 we elaborate on the computation of a probabilistic bound for the condition number. Section 2.6 discusses some comparisons with several other (probabilistic) condition number estimators. We end with some numerical experiments and conclusions in sections 2.7 and 2.8.

## 2.2 EXTENDED LANCZOS BIDIAGONALIZATION

The method we will develop starts with a random vector $v_0$ with unit norm. We express $v_0$ as linear combination of the right singular vectors $y_i$ of $A$

$$v_0 = \sum_{i=1}^{n} \gamma_i y_i. \tag{2.1}$$

Notice that both the $y_i$ and $\gamma_i$ are unknown. Given the matrix $A$ and the starting vector $v_0$, the extended Lanczos bidiagonalization method repeatedly performs matrix-vector operations with the matrices $A$, $A^T$, $A^{-T}$, and $A^{-1}$ (note that, apart from $A$, the aforementioned matrices are not constructed explicitly). In every step a generated vector is orthogonalized with respect to the previously constructed vectors, and

subsequently normalized. This procedure can be visualized as a string of operations working on vectors:

$$v_0 \xrightarrow{A} u_0 \xrightarrow{A^T} v_1 \xrightarrow{A^{-T}} u_{-1} \xrightarrow{A^{-1}} v_{-1} \xrightarrow{A} u_1 \xrightarrow{A^T} \dots$$

Note that in this visualization the orthonormalization of the vectors is not shown. In this scheme, applying the operation $A^{-T}$ after $A^T$ (and $A$ after $A^{-1}$) may seem contradictory, but since the vectors are orthogonalized in between this truly yields new vectors. Another way to represent this procedure is the table below:

| Step | Action | Generated | Action | Generated | Action | Generated | Action | Generated |
|---|---|---|---|---|---|---|---|---|
| 0 | $Av_0$ | $u_0$ | $A^Tu_0$ | $v_1$ | $A^{-T}v_1$ | $u_{-1}$ | $A^{-1}u_{-1}$ | $v_{-1}$ |
| 1 | $Av_{-1}$ | $u_1$ | $A^Tu_1$ | $v_2$ | $A^{-T}v_2$ | $u_{-2}$ | $A^{-1}u_{-2}$ | $v_{-2}$ |
| $\vdots$ | | | | | | | | |
| $k-1$ | $Av_{-k+1}$ | $u_{k-1}$ | $A^Tu_{k-1}$ | $v_k$ | $A^{-T}v_k$ | $u_{-k}$ | $A^{-1}u_{-k}$ | $v_{-k}$ |

During the procedure, the generated vectors $v_j$ are normalized after being orthogonalized with respect to all previous generated $v_i$, i.e., for $k \geq 1$

$$\begin{aligned} v_k &\perp \{v_0, v_1, v_{-1}, \dots, v_{k-1}, v_{-k+1}\}, \\ v_{-k} &\perp \{v_0, v_1, v_{-1}, \dots, v_{-k+1}, v_k\}. \end{aligned}$$

Similarly, all generated vectors $u_j$ have unit norm and

$$\begin{aligned} u_{k-1} &\perp \{u_0, u_{-1}, u_1, \dots, u_{k-2}, u_{-k+1}\}, \\ u_{-k} &\perp \{u_0, u_{-1}, u_1, \dots, u_{-k+1}, u_{k-1}\}. \end{aligned}$$

Define the matrices $V_1 = [v_0]$ and $U_1 = [u_0]$, and for $k \geq 1$

$$\begin{aligned} V_{2k} &= [V_{2k-1}, v_k], & U_{2k} &= [U_{2k-1}, u_{-k}], \\ V_{2k+1} &= [V_{2k}, v_{-k}], & U_{2k+1} &= [U_{2k}, u_k]. \end{aligned}$$

The columns of these matrices are orthonormal and span the corresponding subspaces $\mathcal{V}_{2k}$, $\mathcal{V}_{2k+1}$, $\mathcal{U}_{2k}$, and $\mathcal{U}_{2k+1}$, respectively. We assume for the moment that no breakdowns occur, so all spaces are of full di-

mension; how to handle a breakdown is discussed in section 2.7. After $k \geq 1$ steps the algorithm gives rise to the following matrix equations:

$$\begin{aligned} AV_{2k-1} &= U_{2k-1}H_{2k-1}, \\ A^TU_{2k-1} &= V_{2k-1}(H_{2k-1})^T + \beta_{k-1}\, v_k\, e_{2k-1}^T, \\ A^{-T}V_{2k} &= U_{2k}(K_{2k})^T, \\ A^{-1}U_{2k} &= V_{2k}K_{2k} + \delta_k\, v_{-k}\, e_{2k}^T, \end{aligned} \tag{2.2}$$

and furthermore,

$$\begin{aligned} AV_{2k} &= U_{2k}H_{2k} + \beta_{-k}\, u_k\, e_{2k}^T, \\ A^TU_{2k} &= V_{2k}(H_{2k})^T, \\ A^{-T}V_{2k+1} &= U_{2k+1}(K_{2k+1})^T + \delta_{-k}\, u_{-k-1}\, e_{2k+1}^T, \\ A^{-1}U_{2k+1} &= V_{2k+1}K_{2k+1}. \end{aligned} \tag{2.3}$$

Here, and throughout the chapter, $H_{m,p}$ is an $m \times p$ matrix. We will use only one subscript if the matrix is square, i.e., $H_m$ is an $m \times m$ matrix, and we will refer to the matrices $H_{m,p}$ and $K_{m,p}$ as $H$ and $K$ if the size is not of interest. Furthermore, $e_i$ is the $i$th unit vector and the coefficients $\beta_j$ and $\delta_j$ are entries of the matrices $H$ and $K$ which will be specified in section 2.3. More details on the recurrence relation between the vectors $u$ and $v$ will be given in the next section, where we show that orthogonalization can be done using 3-term recurrences. In particular, the pseudocode for the algorithm that will be introduced in section 2.7 shows that only 3 vectors of storage are needed.

Let $\theta_1^{(2k-1)} \geq \cdots \geq \theta_{2k-1}^{(2k-1)}$ be the singular values of $H_{2k-1}$, and let $\theta_1^{(2k)} \geq \cdots \geq \theta_{2k}^{(2k)}$ be the singular values of $H_{2k}$. Similarly, let $\xi_1^{(2k-1)} \geq \cdots \geq \xi_{2k-1}^{(2k-1)}$ be the singular values of $K_{2k-1}$, and let $\xi_1^{(2k)} \geq \cdots \geq \xi_{2k}^{(2k)}$ be the singular values of $K_{2k}$. These values are approximations of the singular values of $A$ and $A^{-1}$, respectively. We will avoid the use of superscripts if this is clear from the context. Further, let $c_j$ and $d_j$ indicate the corresponding right singular vectors of $H$ and $K$, respectively. We will now study the behavior of these values $\theta_j$ and $\xi_j$ to obtain bounds for the extreme singular values of $A$.

**Proposition 2.2.1** *For* $1 \leq j \leq 2k-1$,

*(a) the singular values of $H$ converge monotonically to the largest singular values of $A$:* $\theta_j^{(2k-1)} \leq \theta_j^{(2k)} \leq \sigma_j(A)$,

*(b) the inverse singular values of K converge monotonically to the smallest singular values of A:*

$$\sigma_{n-j+1}(A) = \left(\sigma_j(A^{-1})\right)^{-1} \leq \left(\xi_j^{(2k)}\right)^{-1} \leq \left(\xi_j^{(2k-1)}\right)^{-1}.$$

*Proof.* The matrix $H_{2k-1}$ can be seen as the matrix $H_{2k}$ from which the $2k$th row and column have been deleted. The same holds for the matrices $K_{2k-1}$ and $K_{2k}$. Now we apply [47, corollary 3.1.3] and obtain the first inequalities of both $(a)$ and $(b)$. The second inequalities hold because of [47, lemma 3.3.1] □

In the next section we will see that $H^{-1} = K$, and therefore the equality $\{\theta_1^{-1}, \ldots, \theta_{2k}^{-1}\} = \{\xi_1, \ldots, \xi_{2k}\}$ holds. Proposition 2.2.1 shows in particular that the largest singular value of the matrices $H$ converges monotonically to $\sigma_1$, and the inverse of the largest singular value of the matrices $K$ converges monotonically to $\sigma_n$. After the $k$th step of the procedure, we obtain the value $\theta_1^{(2k)}$, a lower bound for $\sigma_1$, and the value $(\xi_1^{(2k)})^{-1}$, an upper bound for $\sigma_n$.

**Corollary 2.2.2** *After the kth step of extended Lanczos bidiagonalization we obtain a lower bound for the condition number of A:*

$$\kappa_{low}(A) = \frac{\theta_1}{\xi_1^{-1}} \leq \frac{\sigma_1}{\sigma_n} = \kappa(A). \tag{2.4}$$

The experiments in section 2.7 show for different matrices that the lower bound achieved by extended Lanczos bidiagonalization may often be very good.

We can reformulate the expressions in (2.2) and (2.3) to see the similarities with the extended Lanczos method, see, e.g., [49], with starting vector $v_0$ and matrix $A^TA$, so that for $k \geq 1$:

$$\begin{aligned}
A^TAV_{2k-1} &= A^TU_{2k-1}H_{2k-1} \\
&= V_{2k-1}(H_{2k-1})^TH_{2k-1} + \beta_{k-1}\, v_k\, e_{2k-1}^T\, H_{2k-1} \\
(A^TA)^{-1}V_{2k} &= A^{-1}U_{2k}(K_{2k})^T \\
&= V_{2k}K_{2k}(K_{2k})^T + \alpha_k^{-1}\, \delta_k\, v_{-k}\, e_{2k}^T \\
AA^TU_{2k} &= AV_{2k}(H_{2k})^T \\
&= U_{2k}H_{2k}(H_{2k})^T + \alpha_k\, \beta_{-k}\, u_k\, e_{2k}^T \\
(AA^T)^{-1}U_{2k-1} &= A^{-T}V_{2k-1}K_{2k-1} \\
&= U_{2k-1}(K_{2k-1})^TK_{2k-1} + \delta_{-k+1}\, u_{-k+1}\, e_{2k-1}^TK_{2k-1}.
\end{aligned} \tag{2.5}$$

This way of representing the procedure will be convenient in the next sections where we will investigate the structure of the generated matrices and introduce Laurent polynomials.

## 2.3 THE SPECIAL STRUCTURE OF THE GENERATED H AND K MATRICES

In the previous section we introduced the extended Lanczos bidiagonalization method. The four leading submatrices arising in (2.2) and (2.3) are given by

$$\begin{aligned} H_{2k-1} &= U_{2k-1}^T A V_{2k-1}, & K_{2k-1} &= V_{2k-1}^T A^{-1} U_{2k-1}, \\ H_{2k} &= U_{2k}^T A V_{2k}, & K_{2k} &= V_{2k}^T A^{-1} U_{2k}. \end{aligned}$$

These matrices $H$ and $K$ turn out to be tridiagonal matrices with a special structure as we will show in the next proposition. Note that we assume for all $j \in \{-k, \ldots, k\}$ that the entries $\alpha_j$, $\beta_j$ and $\delta_j$ are nonzero.

**Proposition 2.3.1** *(a) The matrix $H$ is tridiagonal and of the form*

$$\begin{bmatrix} \alpha_0 & \beta_0 & & & & & & \\ & \alpha_1 & & & & & & \\ & \beta_{-1} & \alpha_{-1} & \beta_1 & & & & \\ & & & \alpha_2 & & & & \\ & & & \beta_{-2} & \alpha_{-2} & \beta_2 & & \\ & & & & & \alpha_3 & & \\ & & & & & & \ddots & \end{bmatrix}, \tag{2.6}$$

*where its entries satisfy*

$$\begin{aligned} h_{2j,2j} &= \alpha_j &&= \|A^{-T} v_j\|^{-1} = \|A^T u_{-j}\|, \\ h_{2j+1,2j} &= \beta_{-j} &&= u_j^T A v_j, \\ h_{2j+1,2j+1} &= \alpha_{-j} &&= u_j^T A v_{-j}, \\ h_{2j+1,2j+2} &= \beta_j &&= \|A^T u_j - (u_j^T A v_j) v_j - (u_j^T A v_{-j}) v_{-j}\|. \end{aligned}$$

*(b) The matrix K is tridiagonal and of the form*

$$\begin{bmatrix} \alpha_0^{-1} & \delta_0 & & & & & & \\ & \alpha_1^{-1} & & & & & & \\ & \delta_1 & \alpha_{-1}^{-1} & \delta_{-1} & & & & \\ & & & \alpha_2^{-1} & & & & \\ & & & \delta_2 & \alpha_{-2}^{-1} & \delta_{-2} & & \\ & & & & & \alpha_3^{-1} & & \\ & & & & & & \ddots & \end{bmatrix}, \tag{2.7}$$

*where its diagonal entries are defined in (a) and its off-diagonal entries satisfy*

$$\begin{aligned} k_{2j+1,2j} &= \delta_j = \|A^{-1}u_{-j} - (v_{-j+1}^T A^{-1}u_{-j})v_{-j+1} - (v_j^T A^{-1}u_{-j})v_j\|, \\ k_{2j+1,2j+2} &= \delta_{-j} = v_{-j}^T A^{-1}u_{-(j+1)}. \end{aligned}$$

*Proof.* We will focus first on the transposed matrix $H_{2k}^T$. Note therefore that $A^T U_{2k} = V_{2k} H_{2k}^T$. The $(2j+1)$st column of $A^T U_{2k}$ is $A^T u_j$, and thus the $(2j+1)$st column of $H_{2k}^T$ can be described using the step of the algorithm in which the vector $v_{j+1}$ is constructed

$$\beta_j v_{j+1} = A^T u_j - \sum_{i=-j}^{j} \gamma_i \, v_i,$$

where $\gamma_i = v_i^T A^T u_j = u_j^T A v_i$ and $\beta_j$ is a factor such that $v_{j+1}$ has unit norm. For all $i \in \{-j+1, \ldots, j-1\}$ we have

$$\begin{aligned} Av_i &\in \operatorname{span}\{(AA^T)^{-j+1}Av_0, \ldots, (AA^T)^{j-1}Av_0\} \\ &= \operatorname{span}\{u_0, u_{-1}, u_1, \ldots, u_{-j+1}, u_{j-1}\}, \end{aligned}$$

and therefore $\gamma_i = 0$ for all $i \in \{-j+1, \ldots, j-1\}$. We obtain the three-term recurrence relation

$$A^T u_j = (u_j^T A v_j)\, v_j + (u_j^T A v_{-j})\, v_{-j} + \beta_j v_{j+1},$$

which implies that the $(2j+1)$st column of $H_{2k}^T$ has only three nonzero entries. This gives us the three nonzero entries of the odd rows of $H$, $h_{2j+1,2j}$, $h_{2j+1,2j+1}$, and $h_{2j+1,2j+2}$.

For the description of the $(2j)$th column of $H_{2k}^T$ another step of the algorithm is used, namely

$$\alpha_j^{-1} u_{-j} = A^{-T} v_j - \sum_{i=-j+1}^{j-1} \gamma_i u_i,$$

where $\gamma_i = u_i^T A^{-T} v_j = v_j^T A^{-1} u_i$ and $\alpha_j^{-1}$ is a factor such that $u_{-j}$ has unit norm. For all $i \in \{-j+1, \ldots, j-1\}$ we have

$$\begin{aligned} A^{-1} u_i &\in \operatorname{span}\{(A^T A)^{-j+1} v_0, \ldots, (A^T A)^{j-1} v_0\} \\ &= \operatorname{span}\{v_0, v_1, v_{-1}, \ldots, v_{j-1}, v_{-j+1}\}, \end{aligned}$$

and therefore $\gamma_i = 0$ for all $i \in \{-j+1, \ldots, j-1\}$. We obtain the recurrence relation

$$A^{-T} v_j = \alpha_j^{-1} u_{-j}, \qquad \text{and therefore} \qquad A^T u_{-j} = \alpha_j v_j,$$

implying that the $(2j)$th column of $H_{2k}^T$ has only one nonzero entry. The entries of the matrix $K$ can be obtained by a similar reasoning. □

This description of the matrices $H$ and $K$ leads to the following recurrence relations:

$$\begin{array}{lcll} Av_{-k} &=& \alpha_{-k} u_k, & k \geq 0, \quad \star \\ Av_k &=& \beta_{k-1} u_{k-1} + \alpha_k u_{-k} + \beta_{-k} u_k, & k \geq 1, \\ A^T u_{-k} &=& \alpha_k v_k, & k \geq 1, \\ A^T u_k &=& \beta_{-k} v_k + \alpha_{-k} v_{-k} + \beta_k v_{k+1}, & k \geq 1, \quad \star \\ A^{-T} v_k &=& \alpha_k^{-1} u_{-k}, & k \geq 1, \quad \star \\ A^{-T} v_{-k} &=& \delta_k u_{-k} + \alpha_{-k}^{-1} u_k + \delta_{-k} u_{-(k+1)}, & k \geq 1, \\ A^{-1} u_k &=& \alpha_{-k}^{-1} v_{-k}, & k \geq 0, \\ A^{-1} u_{-k} &=& \delta_{-(k-1)} v_{-(k-1)} + \alpha_k^{-1} v_k + \delta_k v_{-k}, & k \geq 1, \quad \star \end{array} \tag{2.8}$$

and $A^T u_0 = \alpha_0 v_0 + \beta_0 v_1$, $A^{-T} v_0 = \alpha_0^{-1} u_0 + \delta_0 u_{-1}$. The relations indicated by a $\star$ correspond to the matrix vector multiplications that are done explicitly during the procedure, while the other lines are added to give a complete representation of the relations in (2.2) and (2.3). These relations suggest that this method requires at most 6 vectors of storage, and the algorithm presented in section 2.7 even shows only 3 vectors have to be stored. Furthermore, having found this explicit form of the two matrices, it can be seen that the matrices $H$ and $K$ are inverses.

**Proposition 2.3.2** *The leading submatrix of H of order j is the inverse of the leading submatrix of K of the same order, i.e., for* $1 \leq j < n$,

$$H_j K_j = K_j H_j = I_j.$$

*Proof.* If we would carry out $n$ steps of extended Lanczos bidiagonalization, we would obtain orthogonal matrices $V_n$ and $U_n$ satisfying

$$\begin{aligned} H_n K_n &= U_n^T A V_n V_n^T A^{-1} U_n = I_n, \\ K_n H_n &= V_n^T A^{-1} U_n U_n^T A V_n = I_n. \end{aligned}$$

Due to the special tridiagonal structure, it is easy to see that the statement of the proposition holds. □

The previous proposition implies that the singular values of $K$ are the inverses of the singular values of $H$, and therefore we can adjust corollary 2.2.2.

**Corollary 2.3.3** *After the kth step of extended Lanczos bidiagonalization we obtain a lower bound for the condition number of A:*

$$\kappa_{low}(A) = \frac{\theta_1}{\theta_{2k}} \leq \frac{\sigma_1}{\sigma_n} = \kappa(A). \tag{2.9}$$

The matrices in the reformulated expressions (2.5) also have a special structure, just as the matrices formed in the extended Lanczos method in [49]. The four symmetric matrices generated in this extended Lanczos process, for $k \geq 1$, are given by

$$\begin{array}{lcll} R_{2k-1} & = & (H_{2k-1})^T H_{2k-1} & = V_{2k-1}^T A^T A V_{2k-1}, \\ \widetilde{R}_{2k} & = & H_{2k} H_{2k}^T & = U_{2k}^T A A^T U_{2k}, \\ \widetilde{S}_{2k-1} & = & (K_{2k-1})^T K_{2k-1} & = U_{2k-1}^T (A A^T)^{-1} U_{2k-1}, \\ S_{2k} & = & K_{2k} K_{2k}^T & = V_{2k}^T (A^T A)^{-1} V_{2k}. \end{array} \tag{2.10}$$

They are all four the product of two tridiagonal matrices with a special structure, namely the matrices obtained from extended Lanczos bidiag-

onalization. The matrices $R_{2k-1}$ and $\widetilde{R}_{2k}$ are pentadiagonal and of the form

$$R_{2k-1} = \begin{bmatrix} \times & \times & & & & & \\ \times & \times & \times & \times & & & \\ & \times & \times & \times & & & \\ & \times & \times & \times & \times & \times & \\ & & & \times & \times & \times & \\ & & & \times & \times & \ddots & \end{bmatrix}, \quad \widetilde{R}_{2k} = \begin{bmatrix} \times & \times & \times & & & & \\ \times & \times & \times & & & & \\ \times & \times & \times & \times & \times & & \\ & & \times & \times & \times & & \\ & & \times & \times & \times & \times & \\ & & & & \times & \ddots & \end{bmatrix}.$$

The matrices $S_{2k}$ and $\widetilde{S}_{2k+1}$ have similar structures. The product of the matrices $R$ and $S$ is a rank-one modification of the identity. Again, if we would carry out $n$ steps of extended Lanczos bidiagonalization, we would obtain orthogonal matrices $V_n$ and $U_n$ with the following identities:

$$\begin{aligned} S_n R_n &= V_n^T (A^T A)^{-1} V_n V_n^T (A^T A) V_n = I_n = R_n S_n, \\ \widetilde{S}_n \widetilde{R}_n &= U_n^T (A A^T)^{-1} U_n U_n^T (A A^T) U_n = I_n = \widetilde{R}_n \widetilde{S}_n. \end{aligned}$$

Due to the special pentadiagonal structure of the matrices, for $1 \leq j < n$, the product is a rank-one modification of the identity, where we have to distinguish between the even and odd cases:

$$\begin{aligned} S_{2k} R_{2k} &= I_{2k} + \mathring{w}_{2k} e_{2k}^T, & S_{2k+1} R_{2k+1} &= I_{2k+1} + e_{2k+1} \mathring{w}_{2k+1}^T, \\ R_{2k} S_{2k} &= I_{2k} + e_{2k} \check{w}_{2k}^T, & R_{2k+1} S_{2k+1} &= I_{2k+1} + \check{w}_{2k+1} e_{2k+1}^T, \\ \widetilde{S}_{2k} \widetilde{R}_{2k} &= I_{2k} + \widetilde{w}_{2k} e_{2k}^T, & \widetilde{R}_{2k+1} \widetilde{S}_{2k+1} &= I_{2k+1} + e_{2k+1} \widetilde{w}_{2k+1}^T, \\ \widetilde{R}_{2k} \widetilde{S}_{2k} &= I_{2k} + e_{2k} \widehat{w}_{2k}^T, & \widetilde{S}_{2k+1} \widetilde{R}_{2k+1} &= I_{2k+1} + \widehat{w}_{2k+1} e_{2k+1}^T. \end{aligned}$$

Here, the various vectors $w_{2k} \in \mathbb{R}^{2k}$ and $w_{2k+1} \in \mathbb{R}^{2k+1}$ are such that only the last two entries are (possibly) nonvanishing.

The matrices $S$, $\widetilde{S}$, $R$, and $\widetilde{R}$ are used in the next section to give an explicit expression for the Laurent polynomials arising in extended Lanczos bidiagonalization.

## 2.4 POLYNOMIALS ARISING IN EXTENDED LANCZOS BIDIAGONALIZATION

In every step of the extended Lanczos bidiagonalization procedure four different vectors are generated. Since these vectors lie in an extended Krylov subspace, they can be expressed using polynomials:

$$\begin{array}{lcll} v_k & = & p_k(A^TA)v_0 & \in \mathcal{K}_{k,k+1}(A^TA, v_0), \\ u_{-k} & = & q_{-k}(AA^T)Av_0 & \in \mathcal{K}_{k+1,k}(AA^T, Av_0), \\ v_{-k} & = & p_{-k}(A^TA)v_0 & \in \mathcal{K}_{k+1,k+1}(A^TA, v_0), \\ u_k & = & q_k(AA^T)Av_0 & \in \mathcal{K}_{k+1,k+1}(AA^T, Av_0). \end{array} \tag{2.11}$$

The polynomials $p_k$ and $p_{-k}$ are Laurent polynomials of the form

$$p_k(t) = \sum_{j=-k+1}^{k} \mu_j^{(k)}\, t^j, \qquad p_{-k}(t) = \sum_{j=-k}^{k} \mu_j^{(-k)}\, t^j. \tag{2.12}$$

Similarly, $q_{-k}$ and $q_k$ are Laurent polynomials and are defined as

$$q_{-k}(t) = \sum_{j=-k}^{k-1} \nu_j^{(-k)}\, t^j, \qquad q_k(t) = \sum_{j=-k}^{k} \nu_j^{(k)}\, t^j. \tag{2.13}$$

The recurrence relations in (2.8) give rise to recurrence relations connecting the polynomials $p$ and $q$:

$$\begin{array}{lcll} p_{-k}(t) & = & \alpha_{-k} q_k(t), & k \geq 0, \\ p_k(t) & = & \beta_{k-1} q_{k-1}(t) + \alpha_k q_{-k}(t) + \beta_k q_k(t), & k \geq 1, \\ tq_{-k}(t) & = & \alpha_k p_k(t), & k \geq 1, \\ tq_k(t) & = & \beta_{-k} p_k(t) + \alpha_{-k} p_{-k}(t) + \beta_k p_{k+1}(t), & k \geq 1, \\ t^{-1}p_k(t) & = & \alpha_k^{-1} q_{-k}(t), & k \geq 1, \\ t^{-1}p_{-k}(t) & = & \delta_k q_{-k}(t) + \alpha_{-k}^{-1} q_k(t) + \delta_{-k} q_{-(k+1)}(t), & k \geq 1, \\ q_k(t) & = & \alpha_{-k}^{-1} p_{-k}(t), & k \geq 0, \\ q_{-k}(t) & = & \delta_{-(k-1)} p_{-(k-1)}(t) + \alpha_k^{-1} p_k(t) + \delta_k p_{-k}(t), & k \geq 1, \end{array}$$

and $tq_0(t) = \alpha_0 p_0(t) + \beta_0 p_1(t)$, $t^{-1}p_0(t) = \alpha_0^{-1} q_0(t) + \delta_0 q_{-1}(t)$.

Define the following two inner products

$$\langle f, g \rangle = v_0^T f(A^TA)\, g(A^TA) v_0, \tag{2.14}$$

$$[f, g] = v_0^T f(A^TA)\, A^TA\, g(A^TA) v_0. \tag{2.15}$$

**Lemma 2.4.1** *Let $i, j \in \{-k, \ldots, k\}$. The polynomials $p_i$ and $p_j$ are orthonormal with respect to the inner product* (2.14)*, whilst the polynomials $q_i$ and $q_j$ are orthonormal with respect to the inner product* (2.15)*.*

*Proof.* By construction of the $v_i$'s and $u_i$'s we have

$$\langle p_i, p_j \rangle = v_0^T p_i(A^T A)\, p_j(A^T A) v_0 = v_i^T v_j = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases},$$

and

$$[q_i, q_j] = v_0^T q_i(A^T A)\, A^T A\, q_j(A^T A) v_0 = u_i^T u_j = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}.$$

□

Recall that, for $1 < j \leq 2k$, $\theta_j$ is a singular value of $H$, $\xi_j$ is a singular value of $K$, and $c_j$ and $d_j$ indicate the corresponding right singular vectors of $H$ and $K$, respectively.

**Proposition 2.4.2**

*(a) The zeros of the polynomial $p_k$ are exactly $\theta_1^2, \ldots, \theta_{2k-1}^2$.*

*(b) The zeros of the polynomial $p_{-k}$ are exactly $\theta_1^2, \ldots, \theta_{2k}^2$.*

*(c) The zeros of the polynomial $q_{-k}$ are exactly $\theta_1^2, \ldots, \theta_{2k-1}^2$.*

*(d) The zeros of the polynomial $q_k$ are exactly $\theta_1^2, \ldots, \theta_{2k}^2$.*

*Proof.* The proof is similar for all the polynomials; we will give the details only for the first two. Starting with $p_k$, let $j \in \{1, \ldots, 2k-1\}$. Using (2.5) it can be easily seen that the Galerkin condition holds for the pair $(\theta_j^2, V_{2k-1} c_j)$:

$$A^T A V_{2k-1} c_j - \theta_j^2\, V_{2k-1} c_j \perp \mathcal{V}_{2k-1}.$$

Further, since $V_{2k-1} c_j \in \mathcal{V}_{2k-1}$ it follows that

$$(A^T A - \theta_j^2 I) V_{2k-1} c_j \in \operatorname{span}\{(A^T A)^{-k+1} v_0, \ldots, (A^T A)^k v_0\}.$$

For each $j = 1, \ldots, 2k-1$ we have that $(A^T A - \theta_j^2 I) V_{2k-1} c_j \in \mathcal{V}_{2k}$ but is orthogonal to $\mathcal{V}_{2k-1}$. This means that for all $j = 1, \ldots, 2k-1$ the vector

$(A^TA - \theta_j^2 I)V_{2k-1}c_j$ is a nonzero multiple of $v_k = p_k(A^TA)v_0$. Hence $p_k(t)$ contains all factors $t - \theta_j^2$, i.e., its zeros are exactly $\theta_1^2, \ldots, \theta_{2k-1}^2$.

Similarly for the polynomial $p_{-k}$, let $i \in \{1, \ldots, 2k\}$. Again, using (2.5), it can be easily seen that the Galerkin condition holds for the pair $(\xi_i^2, V_{2k}d_i)$. For each $i = 1, \ldots, 2k$ the vector $((A^TA)^{-1} - \xi_i^2 I)V_{2k}d_i$ is a nonzero multiple of $v_{-k} = p_{-k}(A^TA)v_0$, since it is orthogonal to $\mathcal{V}_{2k}$ but an element of $\mathcal{V}_{2k+1}$. Thus $p_{-k}$ contains all factors $(t^{-1} - \xi_j^2)$, and thus all the factors $(t^{-1} - \theta_j^{-2})$, since $H^{-1} = K$.

Similar proofs can be given for (c) and (d). Note that the proofs in [46, p. 467] and [67, p. 266–267] follow the same line of reasoning. □

We know from (2.12) that $p_k(t) = \sum_{j=-k+1}^{k} \mu_j^{(k)} t^j$, which, using the result of proposition 2.4.2, implies that $p_k$ is of the form

$$p_k(t) = \mu_k^{(k)} \cdot t^{-k+1} \cdot (t - \theta_1^2) \cdots (t - \theta_{2k-1}^2). \tag{2.16}$$

Similarly, $p_{-k}$, $q_k$, and $q_{-k}$ are of the form

$$\begin{aligned} p_{-k}(t) &= \mu_{-k}^{(-k)} \cdot t^k \cdot (t^{-1} - \theta_1^{-2}) \cdots (t^{-1} - \theta_{2k}^{-2}), \\ q_{-k}(t) &= \nu_{-k}^{(-k)} \cdot t^{k-1} \cdot (t^{-1} - \theta_1^{-2}) \cdots (t^{-1} - \theta_{2k-1}^{-2}), \\ q_k(t) &= \nu_k^{(k)} \cdot t^{-k+1} \cdot (t - \theta_1^2) \cdots (t - \theta_{2k}^2). \end{aligned} \tag{2.17}$$

It turns out that the coefficients $\mu_k^{(k)}$, $\mu_{-k}^{(-k)}$, $\nu_{-k}^{(-k)}$, and $\nu_k^{(k)}$ can be expressed as a product of certain entries of the matrices $H$ and $K$ introduced in (2.6) and (2.7), respectively.

**Lemma 2.4.3** *The coefficients $\mu_k^{(k)}$, $\mu_{-k}^{(-k)}$, $\nu_{-k}^{(-k)}$, and $\nu_k^{(k)}$ of the polynomials $p_k$, $p_{-k}$, $q_{-k}$ and $q_k$ can be expressed as the product of entries of the matrices $H$ and $K$ defined in (2.6) and (2.7), respectively:*

$$\begin{aligned} \nu_{-k}^{(-k)} &= (-1)^k \prod_{i=-k+1}^{k} \alpha_i \prod_{i=0}^{k-1} \beta_i^{-1} \prod_{i=1}^{k-1} \delta_i^{-1}, \quad \textit{and} \quad \mu_{-k}^{(-k)} = \delta_k^{-1} \nu_{-k}^{(-k)}, \\ \nu_k^{(k)} &= (-1)^k \prod_{i=-k}^{k} \alpha_i^{-1} \prod_{i=0}^{k-1} \beta_i^{-1} \prod_{i=1}^{k} \delta_i^{-1}, \quad \textit{and} \quad \mu_k^{(k)} = \beta_{k-1}^{-1} \nu_{k-1}^{(k-1)}. \end{aligned} \tag{2.18}$$

*Proof.* From the equations in (2.2), the expressions in (2.11), and from the form of the matrices $H$ and $K$ whose entries are described explicitly

in (2.6) and (2.7), respectively, the following recurrence relations for the polynomials can be derived:

$$\begin{aligned} q_{-k}(t) &= \alpha_k t^{-1} p_k(t), \\ p_{-k}(t) &= \delta_k^{-1}(q_{-k}(t) - \delta_{-k+1} p_{-k+1}(t) - \alpha_k^{-1} p_k(t)), \\ q_k(t) &= \alpha_{-k}^{-1} p_{-k}(t), \\ p_{k+1}(t) &= \beta_k^{-1}(t q_k(t) - \beta_k p_k(t) - \alpha_{-k} p_{-k}(t)). \end{aligned} \tag{2.19}$$

Manipulating these relations one obtains recurrence relations for the coefficients:

$$\begin{aligned} \nu_{-k}^{(-k)} &= (-1)^k \alpha_{-k+1} \alpha_k \beta_{k-1}^{-1} \delta_{k-1}^{-1} \nu_{-k+1}^{(-k+1)}, \\ \mu_{-k}^{(-k)} &= (-1)^k \alpha_{-k+1} \alpha_k \beta_{k-1}^{-1} \delta_k^{-1} \mu_{-k+1}^{(-k+1)}, \\ \nu_k^{(k)} &= (-1)^k \alpha_k^{-1} \alpha_{-k}^{-1} \beta_{k-1}^{-1} \delta_k^{-1} \nu_{k-1}^{(k-1)}, \\ \mu_{k+1}^{(k+1)} &= (-1)^k \alpha_k^{-1} \alpha_{-k}^{-1} \beta_k^{-1} \delta_k^{-1} \mu_k^{(k)}. \end{aligned} \tag{2.20}$$

From these relations, the expressions for the coefficients follow easily. □

The results of proposition 2.4.2 and lemma 2.4.3 lead to the following corollary.

**Corollary 2.4.4** *The polynomials $p_k$ and $p_{-k}$ can be expressed as*

$$\begin{aligned} p_k(t) &= \mu_k^{(k)} \cdot t^{-k+1} \cdot det(t I_{2k-1} - R_{2k-1}), \\ p_{-k}(t) &= \mu_{-k}^{(-k)} \cdot t^k \cdot det(t^{-1} I_{2k} - \widetilde{S}_{2k}), \end{aligned}$$

*where $\mu_k^{(k)}$ and $\mu_{-k}^{(-k)}$ are defined in (2.18), and $\widetilde{S}_{2k}$ is the leading submatrix of order $2k$ of $\widetilde{S}_{2k+1}$ defined in (2.10). The polynomials $q_k$ and $q_{-k}$ can be expressed analogously.*

We recall from proposition 2.2.1 that for increasing $k$ the largest singular value of $H_{2k-1}$ converges monotonically to $\sigma_1$, and the inverse of the largest singular value of $K_{2k}$ converges monotonically to $\sigma_n$. This implies that the largest zero of polynomial $p_k$ increases monotonically to $\sigma_1^2$. Likewise, the smallest zero of polynomial $p_{-k}$ decreases monotonically to $\sigma_n^2$. These polynomials are used in the next section to obtain probabilistic bounds for both the largest and smallest singular value of $A$.

## 2.5 PROBABILISTIC BOUNDS FOR THE CONDITION NUMBER

After step $k$, extended Lanczos bidiagonalization implicitly provides Laurent polynomials $p_k$ and $p_{-k}$. In the previous section we have seen that the zeros of $p_k$ and $p_{-k}$ are closely related to the singular values of the matrices $H$ and $K$ (proposition 2.4.2). Moreover, the polynomials $|p_k|$ and $|p_{-k}|$ are strictly increasing to the right of their largest zero and also to the left of their smallest zero, for $t \to 0$. These properties will lead to the derivation of a probabilistic upper bound for $\kappa(A)$. Therefore, we first observe the two equalities

$$1 = \|v_k\|^2 = \|p_k(A^TA)v_0\|^2 = \|\sum_{i=1}^{n} p_k(A^TA)\gamma_i y_i\|^2 = \sum_{i=1}^{n} \gamma_i^2 p_k(\sigma_i^2)^2,$$
$$1 = \|v_{-k}\|^2 = \|p_{-k}(A^TA)v_0\|^2 = \|\sum_{i=1}^{n} p_{-k}(A^TA)\gamma_i y_i\|^2 = \sum_{i=1}^{n} \gamma_i^2 p_{-k}(\sigma_i^2)^2.$$

Here we used, in view of (2.1), that $A^TAy_i = \sigma_i^2 y_i$ and the fact that the right singular vectors $y_i$ are orthonormal. Since the obtained sums only consist of nonnegative terms, we conclude that

$$|p_k(\sigma_1^2)| \leq \frac{1}{|\gamma_1|}, \qquad \text{and} \qquad |p_{-k}(\sigma_n^2)| \leq \frac{1}{|\gamma_n|}. \tag{2.21}$$

Similarly,

$$\begin{aligned} 1 = \|u_k\|^2 &= \|q_k(AA^T)Av_0\|^2 \\ &= \|\sum_{i=1}^{n} q_k(AA^T)\gamma_i\sigma_i x_i\|^2 = \sum_{i=1}^{n} \gamma_i^2\sigma_i^2 q_k(\sigma_i^2), \\ 1 = \|u_{-k}\|^2 &= \|q_{-k}(AA^T)Av_0\|^2 \\ &= \|\textstyle\sum_{i=1}^{n} q_{-k}(AA^T)\gamma_i\sigma_i x_i\|^2 = \displaystyle\sum_{i=1}^{n} \gamma_i^2\sigma_i^2 q_{-k}(\sigma_i^2). \end{aligned}$$

Here we used that $AA^Tx_i = \sigma_i^2 x_i$ and the fact that the left singular vectors $x_i$ are orthonormal. Again, the sum we obtain only contains nonnegative terms and thus $1 \geq \sigma_1 |\gamma_1| \, |q_k(\sigma_1^2)|$, which gives us the inequality

$$\sigma_1 \, |q_k(\sigma_1^2)| \leq \frac{1}{|\gamma_1|}, \qquad \sigma_n \, |q_{-k}(\sigma_n^2)| \leq \frac{1}{|\gamma_n|}. \tag{2.22}$$

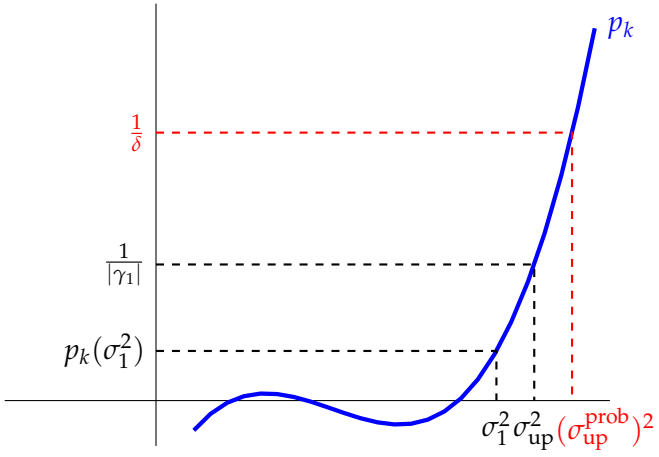



Figure 2.1: Schematic representation on how the probabilistic bound $(\sigma_{\text{up}}^{\text{prob}})^2$ is related to the polynomial $p_k$. Note that non of the values presented in black are known. Only in case the computed $\frac{1}{\delta}$ is smaller than $p_k(\sigma_1^2)$ (probability $< \varepsilon$), the computed probabilistic upper bound $(\sigma_{\text{up}}^{\text{prob}})^2$ is a lower bound for $\sigma_1^2$.

If $\gamma_1$ were known, the first estimates in (2.21) and (2.22) would provide an upper bound for $\|A\|^2 = \sigma_1^2$, namely the largest zero of the functions

$$f_1(t) = |p_k(t)| - \frac{1}{|\gamma_1|}, \qquad f_2(t) = t\,|q_k(t)| - \frac{1}{|\gamma_1|}.$$

Similarly, if $\gamma_n$ were known, the second estimates in (2.21) and (2.22) would both provide a lower bound for $\|A^{-1}\|^{-2} = \sigma_n^2$, namely the smallest zero of the functions

$$g_1(t) = |p_{-k}(t)| - \frac{1}{|\gamma_n|}, \qquad g_2(t) = t\,|q_{-k}(t)| - \frac{1}{|\gamma_n|}.$$

However, both $\gamma_1$ and $\gamma_n$ are unknown. Therefore we will compute a value $\delta$ that will be a lower bound for $|\gamma_1|$ and $|\gamma_n|$ with a user-chosen probability. Suppose that $|\gamma_1| < \delta$. Then the largest zero of $f_1^\delta(t) = |p_k(t)| - \delta^{-1}$ is smaller than the largest zero of $f_1^{\gamma_1}(t) = |p_k(t)| - |\gamma_1|^{-1}$ and thus may be less than $\sigma_1^2$. This means that $\delta$ may not give an upper bound for $\sigma_1$. We now compute the value $\delta$ such that the probability that $|\gamma_1| < \delta$ (or $|\gamma_n| < \delta$) is small, namely $\varepsilon$. We refer to figure 2.1 for a schematic representation of the polynomial $p_k$ and the upper bounds that we are interested in.

Let $S^{n-1}$ be the unit sphere in $\mathbb{R}^n$. We choose the starting vector $v_0$ randomly from a uniform distribution on $S^{n-1}$, see, e.g., [59, p. 1116], which (by an orthogonal transformation) implies that $(\gamma_1, \ldots, \gamma_n)$ is also random with respect to the uniform distribution on $S^{n-1}$, (MATLAB code: `v0=randn(n,1); v0=v0/norm(v0)`).

**Lemma 2.5.1** *Assume that the starting vector $v_0$ has been chosen randomly with respect to the uniform distribution over the unit sphere $S^{n-1}$ and let $\delta \in [0,1]$. Then*

$$\begin{aligned} P(|\gamma_1| \leq \delta) &= 2B(\tfrac{n-1}{2}, \tfrac{1}{2})^{-1} \int_0^{arcsin(\delta)} \cos^{n-2}(t)\, dt \\ &= B(\tfrac{n-1}{2}, \tfrac{1}{2})^{-1} \int_0^{\delta^2} t^{-\frac{1}{2}} (1-t)^{\frac{n-3}{2}}\, dt, \end{aligned}$$

*where $B$ denotes Euler's Beta function: $B(x,y) = \int_0^1 t^{x-1}(1-t)^{y-1}dt$, and $P$ stands for probability.*

*Proof.* For the first equality see [17, lemma 3.1], and for the second see [55, theorem 7.1]. □

The user selects the probability $\varepsilon = P(|\gamma_1| \leq \delta)$, i.e., the probability that the computed bound may not be an upper bound for the singular value $\sigma_1$. Given this user-chosen $\varepsilon$ we have to determine the $\delta$ for which

$$\varepsilon = \frac{B_{\text{inc}}(\frac{n-1}{2}, \frac{1}{2}, \delta^2)}{B_{\text{inc}}(\frac{n-1}{2}, \frac{1}{2}, 1)}, \tag{2.23}$$

where $B_{\text{inc}}(x,y,z) := \int_0^z t^{x-1}(1-t)^{y-1}dt$ is the incomplete Beta function. The $\delta$ can be computed using MATLAB's function `betaincinv`. With this $\delta$ we can compute two probabilistic bounds, i.e., the square root of the largest zero of the function $f_1^\delta$ and the square root of the smallest zero of the function $g_1^\delta$. Computing these values can be done with Newton's method or bisection. We use the value $\theta_1^2$, the largest root of $p_k$, as lower bound of the interval to search, to ensure we find the largest root of the function $f_1^\delta$ (similar reasoning holds for the function $g_1^\delta$). Note that one could equally choose to use the functions $f_2^\delta$ and $g_2^\delta$. We thus acquire a probabilistic upper bound for $\sigma_1$ and a probabilistic lower bound for $\sigma_n$:

$$\sigma_1 < \sigma_{\text{up}}^{\text{prob}} \qquad \text{and} \qquad \sigma_n > \sigma_{\text{low}}^{\text{prob}}.$$

Both inequalities are true with probability at least $1-\varepsilon$. Since the coefficients $\gamma_1$ and $\gamma_n$ are chosen independently, the probability that both inequalities hold is at least $1-2\varepsilon$. This proves the following theorem.

**Theorem 2.5.2** *Assume that the starting vector $v_0$ has been chosen randomly with respect to the uniform distribution over $S^{n-1}$. Let $\varepsilon \in (0,1)$ and let $\delta$ be given by (2.23). Then $\sigma_{up}^{prob}$, the square root of the largest zero of the polynomial*

$$f_1^{\delta}(t) = |p_k(t)| - \tfrac{1}{\delta}, \tag{2.24}$$

*is an upper bound for $\sigma_1$ with probability at least $1-\varepsilon$. Also, $\sigma_{low}^{prob}$, the square root of the smallest zero of the polynomial*

$$g_1^{\delta}(t) = |p_{-k}(t)| - \tfrac{1}{\delta}, \tag{2.25}$$

*is a lower bound for $\sigma_n$ with probability at least $1-\varepsilon$.*

Note that the implementation of the polynomial uses the recurrence relations in (2.19). Therefore, we approximate directly the singular values $\sigma_1$ and $\sigma_n$, and avoid taking squares or square roots. Combining these two bounds leads to a probabilistic upper bound for the condition number of $A$.

**Corollary 2.5.3** *The inequality*

$$\kappa(A) = \frac{\sigma_1}{\sigma_n} \leq \frac{\sigma_{up}^{prob}}{\sigma_{low}^{prob}} = \kappa_{up}(A). \tag{2.26}$$

*holds with probability at least $1-2\varepsilon$.*

The probabilistic upper bounds usually decrease monotonically as a function of $k$. The lemma below gives some intuition for this behavior.

**Lemma 2.5.4** *Let $t_1$ and $t_2$ be such that $|p_k(t_1)| = \frac{1}{\delta}$, $|p_{k+1}(t_2)| = \frac{1}{\delta}$ and define $M := \alpha_k \alpha_{-k} \beta_k \delta_k$. If $t_1 \geq \theta_1^2 + M^{-1}(1+\sqrt{M}\theta_2)$, then $t_2 \leq t_1$.*

*Proof.* We investigate when $|p_{k+1}(t_1)| \geq \frac{1}{\delta}$, since this implies $t_2 \leq t_1$. Denote by $\theta_1^2 \geq \cdots \geq \theta_{2k+1}^2$ the zeros of the polynomial $p_{k+1}(t)$, and by $\eta_1^2 \geq \cdots \geq \eta_{2k-1}^2$ the zeros of $p_k(t)$. Then

$$\left| \frac{p_{k+1}(t_1)}{p_k(t_1)} \right| = \left| \frac{\mu_{k+1}\, t_1^{-k}(t_1-\theta_1^2)\cdots(t_1-\theta_{2k+1}^2)}{\mu_k\, t_1^{-k+1}(t_1-\eta_1^2)\cdots(t_1-\eta_{2k-1}^2)} \right| = \delta\, |p_{k+1}(t_1)|.$$

The relations in (2.20) show that $\left|\frac{\mu_{k+1}}{\mu_k}\right| = (\alpha_k\alpha_{-k}\beta_k\delta_k)^{-1} =: M$. By the interlacing properties of singular values ($\eta_{2i-1} \geq \theta_{2i+1}$ for $i = 1, \ldots, k$) we obtain the inequality

$$\delta\,|p_{k+1}(t_1)| \geq M\,\frac{(t_1-\theta_1^2)(t_1-\theta_2^2)}{t_1}.$$

So we are interested in finding $t_1$ such that $M\,(t_1-\theta_1^2)(t_1-\theta_2^2) \geq t_1$, which is

$$Mt_1^2 - (M(\theta_1^2+\theta_2^2)+1)t_1 + M\theta_1^2\theta_2^2 \geq 0.$$

This holds for

$$\begin{aligned} t_1 &\geq \tfrac{1}{2M}\big(M(\theta_1^2+\theta_2^2)+1+\sqrt{(M(\theta_1^2+\theta_2^2)+1)^2-4M^2\theta_1^2\theta_2^2}\big) \\ &= \tfrac{1}{2}(\theta_1^2+\theta_2^2)+\tfrac{1}{2M}+\tfrac{1}{2M}\sqrt{(M(\theta_1^2-\theta_2^2)+1)^2+4M\theta_2^2}. \end{aligned}$$

Therefore $\delta\,|p_{k+1}(t_1)| \geq 1$ (and hence $t_2 < t_1$) holds for $t_1 \geq \theta_1^2 + M^{-1}(1+\sqrt{M}\theta_2)$. □

## 2.6 OTHER CONDITION ESTIMATORS

In this section we will first compare probabilistic results for $\kappa_2(A)$ obtained by Dixon [16] and Gudmundsson, Kenney, and Laub [34] with those of our method. Subsequently, we will briefly mention some condition number estimators for $\kappa_1(A)$ and $\kappa_F(A)$.
As for the method introduced in this paper, for all methods to approximate either $\kappa_1(A)$, or $\kappa_F(A)$, or $\kappa_2(A)$, discussed in this section, an LU decomposition is needed and $\mathcal{O}(1)$ vectors of storage are required (see for our method the recurrence relations (2.8) and the algorithm presented in section 2.7). Note that of the approaches discussed in this section only the block method by Higham and Tisseur [41] is also suitable for complex matrices.

### 2.6.1 *Probabilistic condition estimators based on the 2-norm*

**Theorem 2.6.1** *(Dixon [16, theorem 1][1]) Let $B$ be a symmetric positive definite (SPD) matrix with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_n$, and $\zeta > 1$. If $v$ is chosen randomly on the unit sphere, then*

$$v^T B v \leq \lambda_1 \leq \zeta \cdot v^T B v \tag{2.27}$$

*holds with probability at least $1 - 0.8\sqrt{n/\zeta}$.*

Note that the left inequality always holds; the probabilistic part only concerns the second inequality. Dixon [16, theorem 2] subsequently applies this result to both $B^k = (A^T A)^k$ and $B^{-k} = (A^T A)^{-k}$, which gives the following theorem.

**Theorem 2.6.2** *(Dixon [16, theorem 2]) Let $A$ be a real nonsingular $n \times n$ matrix and $k$ be a positive integer. For $v, w \in \mathbb{R}^n$, define*

$$\varphi_k(v, w) = (v^T (A^T A)^k v \cdot w^T (A^T A)^{-k} w)^{1/2k}.$$

*If $v$ and $w$ are selected randomly and independently on $S^{n-1}$ and $\zeta > 1$ then*

$$\varphi_k(v, w) \leq \kappa(A) \leq \zeta \cdot \varphi_k(v, w)$$

*holds with probability at least $1 - 1.6\sqrt{n/\zeta^k}$.*

Kuczyński and Woźniakowski [59] present several probabilistic bounds for quantities that are better estimates of the largest eigenvalue of an SPD matrix than the one considered by Dixon, with the same number of matrix-vector products. They appropriately call the method that leads to the quantity $(v^T B^k v)^{1/k}$ studied by Dixon the *modified* power method. The more common power method considers, with the same number $k$ of matrix-vector products, the Rayleigh quotient of $B^{k-1}v$, that is, the quantity $(B^{k-1}v)^T B B^{k-1} v = v^T B^{2k-1} v$. This generally results in a better approximation than the quantity considered by Dixon. In [59], the following results are given for the power method and the Lanczos method.

1 Note that [16, theorem 1] contains a typo: $k$ should be 1.

**Theorem 2.6.3** *(Kuczyński and Woźniakowski [59, theorem 4.1(a)]) With the same notations as in theorem 2.6.1, let $0 < \eta < 1$. Let $\theta^{pow}$ be the largest Ritz value obtained with $k \geq 2$ steps of the power method. Then the probability that*

$$\lambda_1 < (1-\eta)^{-1}\,\theta^{pow} \tag{2.28}$$

*holds is at least $1 - 0.824\,\sqrt{n}\,(1-\eta)^{k-\frac{1}{2}}$.*

**Theorem 2.6.4** *(Kuczyński and Woźniakowski [59, theorem 4.2(a)]) With the same notations as in theorem 2.6.3, let $\theta^{Lan}$ be the largest Ritz value obtained with k steps of Lanczos. Then the probability that*

$$\lambda_1 < (1-\eta)^{-1}\,\theta^{Lan} \tag{2.29}$$

*holds is at least $1 - 1.648\,\sqrt{n}\,e^{-\sqrt{\eta}\,(2k-1)}$.*

The proof of theorem 2.6.4 makes use of a Chebyshev polynomial, a well-known proof technique in the area of Krylov methods. Extended Lanczos bidiagonalization adaptively constructs a polynomial that is optimal in some sense for the given matrix and starting vector. Therefore, as we will see below, our probabilistic bounds are usually better than that of theorem 2.6.4.

We can now apply theorems 2.6.3 and 2.6.4 to the matrices $B = A^TA$ and $B = (A^TA)^{-1}$ as above. The following results are new, but follow directly from [59].

**Corollary 2.6.5** *Let $A$ be a real nonsingular $n \times n$ matrix and $k$ be a positive integer, and let $v$ and $w$ be random independent vectors on $S^{n-1}$.*
***(a) [Power method on $A^TA$ and $(A^TA)^{-1}$]*** *Let $\theta_{\max}^{pow} = \frac{v^T(A^TA)^{2k-1}v}{v^T(A^TA)^{2k-2}v}$ be the approximation to $\sigma_1^2$ obtained with k steps of the power method applied to $A^TA$ with starting vector $v$, and let $\theta_{\min}^{pow} = \frac{w^T(A^TA)^{-(2k-1)}w}{w^T(A^TA)^{-(2k-2)}w}$ be the approximation to $\sigma_n^{-2}$ obtained with k steps of the power method applied to $(A^TA)^{-1}$ with starting vector $w$. Then*

$$\kappa(A) \leq (1-\eta)^{-1}\,(\theta_{\max}^{pow} \cdot \theta_{\min}^{pow})^{1/2}$$

*holds with probability at least $1 - 1.648\,\sqrt{n}\,(1-\eta)^{k-\frac{1}{2}}$.*
***(b) [Lanczos on $A^TA$ and $(A^TA)^{-1}$]*** *Let $\theta_{\max}^{Lan}$ be the largest Ritz value obtained with k steps of Lanczos applied to $A^TA$ with starting vector $v$, and*

*let $\theta_{\min}^{Lan}$ be the largest Ritz value obtained with k steps of Lanczos applied to $(A^TA)^{-1}$ with starting vector w. Then*

$$\kappa(A) \leq (1-\eta)^{-1} \, (\theta_{\max}^{Lan} \cdot \theta_{\min}^{Lan})^{1/2}$$

*holds with probability at least* $1 - 3.296\sqrt{n}\, e^{-\sqrt{\eta}\,(2k-1)}$.

**Example 2.6.6** We now give an indicative numerical example for the diagonal matrix $A = \texttt{diag(linspace(1,1e12,n))}$ of size $n = 10^5$ and $\kappa(A) = 10^{12}$. In table 2.1, the probabilistic upper bounds by Dixon (the modified power method, theorem 2.6.2), Kuczyński and Woźniakowski (the power method and the Lanczos method, corollary 2.6.5), and the extended Lanczos bidiagonalization method are considered. We give the ratio $\kappa^{\text{up}}/\kappa^{\text{low}}(A)$, where $\kappa^{\text{up}}$ denotes the various probabilistic upper bounds, and the requirement is that each holds with probability at least 98%. As expected, the power method gives a smaller ratio than the modified power method (see also [59] for more details). The ratio generated by a Chebyshev polynomial is even better, taking into account the subspace effect of a Krylov method. However, the ratio obtained with the polynomial implicitly generated by the method of this paper is the best.

| $k$ | Dixon | K&W (power) | K&W (Lanczos) | Ext LBD |
|---|---|---|---|---|
| 10 | 7.60 | 2.92 | 1.49 | 1.16 |
| 20 | 2.76 | 1.68 | 1.08 | 1.04 |
| 30 | 1.97 | 1.41 | 1.04 | 1.02 |

Table 2.1: Ratios $\kappa^{\text{up}}(A)/\kappa^{\text{low}}(A)$ for $A = \texttt{diag(linspace(1,1e12,n))}$ of size $n = 10^5$, where $\kappa^{\text{up}}$ denotes the probabilistic upper bound provided by Dixon [16], Kuczyński and Woźniakowski (corollary 2.6.5), and the extended Lanczos bidiagonalization method (Ext LBD). We take $k = 10$, 20, and 30 steps, and require that all upper bounds $\kappa^{\text{up}}$ hold with at least 98% ($\varepsilon = 0.01$).

### 2.6.2 *Condition estimators based on other norms*

Next, we mention the successful block method by Higham and Tisseur [41] to estimate $\kappa_1(A) = \|A\|_1 \, \|A^{-1}\|_1$, the 1-norm condition num-

ber. Although $\kappa_2(A)$ and $\kappa_1(A)$ are *equivalent* norms in $\mathbb{R}^n$ in the sense that $\frac{1}{n}\kappa_1(A) \leq \kappa_2(A) \leq n\,\kappa_1(A)$, these bounds are much too crude to be useful for large matrices. Therefore, we may well view $\kappa_2(A)$ and $\kappa_1(A)$ as independent quantities in practice; which one is preferred may depend on the user and application.

Gudmundsson, Kenney, and Laub [34] present an estimator for the condition number based on the Frobenius norm. They select $k$ vectors from $S^{n-1}$, compute an orthonormal basis $Q$ for the span, and take $\sqrt{n/k}\,\|AQ\|_F\,\|A^{-1}Q\|_F$ as an estimate for $\kappa_F(A)$. Again, although $\kappa_2(A)$ and $\kappa_F(A)$ are related in the sense that $\kappa_2(A) \leq \kappa_F(A) \leq n\,\kappa_2(A)$, they can be seen as independent quantities in practice.

## 2.7 NUMERICAL EXPERIMENTS

We present the pseudocode for the extended Lanczos bidiagonalization method including the computation of a lower bound and a probabilistic upper bound for the condition number, see algorithm 2.1. This pseudocode shows that this method requires only 3 vectors of storage. Because of the modest number of steps needed to achieve the given ratio, it turns out that in our examples reorthogonalization with respect to more previous vectors is not needed.

**Experiment 2.7.1** First, we test the method on some well-known diagonal test matrices to get an impression of the performance of the method. In figure 2.2, we plot the convergence of the probabilistic upper bound $\kappa_{\text{up}}(A)$ and lower bound $\kappa_{\text{low}}(A)$ as a function of $k$ for the matrix $A = \texttt{diag(linspace(1,1e12,n))}$, for $n = 10^5$ (a) and for an "exponential diagonal" matrix of the form $A = \texttt{diag}(\rho.\hat{}\texttt{(0:1e5-1))}$ where $\rho$ is such that $\kappa(A) = 10^{12}$ (b). The plots suggest that the spectrum of the latter matrix is harder. Note that there seem to be two different rates of convergence in the dashed lines. A deeper understanding is needed to explain this behavior, and it may be subject to future research. The related question to define an a priori stopping criterion may be investigated as well.

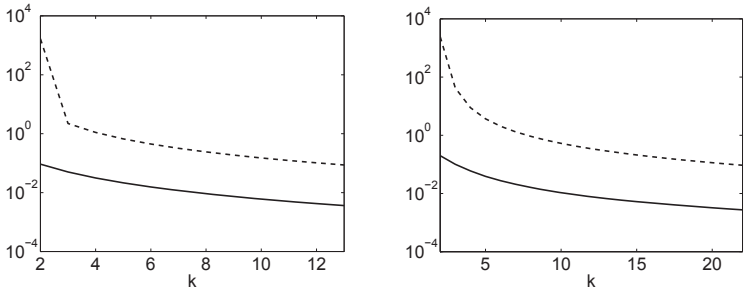
Next, for figure 2.3(a), we carry out $k = 5$ steps of the method for $A = \texttt{diag(linspace(1,1e12,n))}$, $n = 10^5$, and investigate the behavior of the ratio $\kappa_{\text{up}}(A)/\kappa_{\text{low}}(A)$, where $\kappa_{\text{up}}(A)$ is an upper bound with probability at least $1 - 2\varepsilon$, as a function of $\varepsilon$. In figure 2.3(b) we plot

**Algorithm 2.1:** Extended Lanczos bidiagonalization method

**Input:** Nonsingular $(n \times n)$ matrix $A$, random starting vector $w = v_0$, probability level $\varepsilon$, ratio $\zeta$, maximum extended Krylov dimension $2k$.

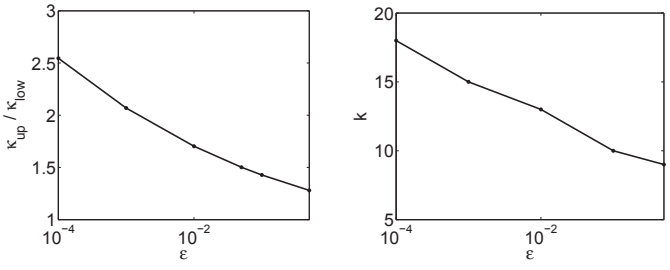
**Output:** A lower bound $\kappa_{\text{low}}(A)$ and a probabilistic upper bound $\kappa_{\text{up}}(A)$ for the condition number $\kappa(A)$ such that $\kappa_{\text{up}}/\kappa_{\text{low}} \leq \zeta$. The probability that $\kappa(A) \leq \kappa_{\text{up}}(A)$ holds is at least $1 - 2\varepsilon$.

Determine $\delta$ from $n$ and $\varepsilon$, see (2.23)
**for** $j = 0, \ldots, k-1$
  $u = Aw$
  $\alpha_{-j} = \|u\|$
  **if** $\alpha_{-j} = 0$, **abort, end**
  $u = u \,/\, \alpha_{-j}$
  $u = A^T u$
  **if** $j > 0$, $\beta_{-j} = v^T u$, $u = u - \beta_{-j} v$, **end**
  $u = u - \alpha_{-j} w$
  $\beta_j = \|u\|$
  **if** $\beta_j = 0$, **abort, end**
  $v = u \,/\, \beta_j$
  $u = A^{-T} v$
  **if** $\|u\| = 0$, **abort, end**
  $\alpha_{j+1} = \|u\|^{-1}$
  Create $H_{2(j+1)}$ using the obtained coefficients $\alpha$'s and $\beta$'s (see (2.6)
  Determine singular values $\theta_1$ and $\theta_{2(j+1)}$ of $H_{2(j+1)}$
  Compute $\kappa_{\text{low}}(A) = \theta_1/\theta_{2(j+1)}$ (see (2.9))
  Determine $\sigma_{\text{up}}^{\text{prob}}$ for $\sigma_1$ with probability $\geq 1 - \varepsilon$ using $f_1^\delta$ (see (2.24))
  Determine $\sigma_{\text{low}}^{\text{prob}}$ for $\sigma_n$ with probability $\geq 1 - \varepsilon$ using $g_1^\delta$ (see (2.25))
  Compute $\kappa_{\text{up}}(A) = \sigma_{\text{up}}^{\text{prob}} \,/\, \sigma_{\text{low}}^{\text{prob}}$ (see (2.26))
  **if** $\kappa_{\text{up}}/\kappa_{\text{low}} \leq \zeta$, **quit, end**
  $u = \alpha_{j+1} u$
  $u = A^{-1} u$
  $\delta_{-j} = w^T u$
  $u = u - \delta_{-j} w - \alpha_{j+1}^{-1} v$
  $\delta_{j+1} = \|u\|$
  **if** $\delta_{j+1} = 0$, **abort, end**
  $w = u \,/\, \delta_{j+1}$
**end**




(a) (b)

Figure 2.2: The relative errors $\kappa_{\text{up}}(A)/\kappa(A) - 1$ (dash) and $1 - \kappa_{\text{low}}(A)/\kappa(A)$ (solid) as function of $k$, for $A =$ `diag(linspace(1,1e12,n))`, $n = 10^5$ (a), and a matrix of the form $A =$ `diag(`$\rho$`.^(0:1e5-1))` with $\kappa(A) = 10^{12}$ (b). Here, $\kappa_{\text{low}}(A)$ is a lower bound and $\kappa_{\text{up}}(A)$ is an upper bound with probability at least 98%.




(a) (b)

Figure 2.3: For $A =$ `diag(linspace(1,1e12,n))`, $n = 10^5$, after $k = 5$ steps of the method: (a) the ratio $\kappa_{\text{up}}(A)/\kappa_{\text{low}}(A)$ where $\kappa_{\text{up}}(A)$ is an upper bound with probability at least $1 - 2\varepsilon$, as function of $\varepsilon$; (b) the iteration $k$ needed to ensure that $\kappa_{\text{up}}(A) \leq 1.1 \cdot \kappa_{\text{low}}(A)$, where $\kappa_{\text{up}}(A)$ is an upper bound with probability at least $1 - 2\varepsilon$, as a function of $\varepsilon$.

the iteration $k$ that is needed to ensure that $\kappa_{\text{up}}(A) \leq 1.1 \cdot \kappa_{\text{low}}(A)$, as a function of $\varepsilon$.

**Experiment 2.7.2** We test the method to estimate the condition number for some large matrices. The matrices we choose are real and non-symmetric. Most of these matrices can be found in the Matrix Market [80] or the University of Florida Sparse Matrix Collection [14, 81]. The starting vector $v_0$ is randomly chosen from a uniform distribution on $S^{n-1}$ as explained in section 2.5. For these experiments we choose $\varepsilon = 0.01$ which corresponds to a reliability of at least 98% for the bounds for the condition number to be true (see section 2.5). Also, we choose $\zeta = 2$ and $\zeta = 1.1$ such that the ratio of the probabilistic upper bound and the lower bound is $\leq \zeta$. To accomplish this, the method adaptively chooses the number of steps $k$. Note that $k$ steps correspond to $k$ operations with $A^TA$ and $k$ operations with $(A^TA)^{-1}$. We use MATLAB's `betaincinv` to compute $\delta$ and bisection to compute the largest and smallest zero of $f_1^\delta$ and $g_1^\delta$, respectively (see (2.24) and (2.25)).

In table 2.1 the results for $\zeta = 2$ are presented. The reason of the choice of $\zeta = 2$ is a comparison of our method to the block method by Higham and Tisseur [41] to estimate the 1-norm condition number $\kappa_1(A)$, which is reported to give almost always an estimate correct to within a factor 2. Although $\kappa_1$ and $\kappa_2$ are independent quantities (see section 2.6.2 for comments), the methods have both a storage of $\mathcal{O}(1)$ vectors and for both methods (only) one LU-factorization is computed which is needed for the inverse operations $A^{-1}$ and $A^{-T}$. The comparison is made to indicate that the running time of the two methods usually does not differ much (see table 2.1). As is shown in table 2.1, especially for the larger matrices, a large part of the computational time is spent on the LU-factorization. Therefore, for such matrices extended Lanczos bidiagonalization may be seen as a relatively cheap add-on. For $\zeta = 2$, usually only a modest number of steps $k$ are sufficient. Of course, choosing a larger $\zeta$ will decrease this number of steps even more. While decreasing $\zeta$ will make the method computationally more expensive, for many matrices this will be a relatively small increase in view of the costs of the LU decomposition. In table 2.2 the results for $\zeta = 1.1$, giving very sharp bounds, show that even for this small $\zeta$ the number of steps $k$ and the running time remain modest.

**Experiment 2.7.3** We compare the new method with the following alternative method to derive a lower bound for $\kappa_2(A)$. First, one applies $k$

| Matrix $A$ | Dim. | $\kappa$ | $\kappa_{\text{low}}$ | $\kappa_{\text{up}}$ | $k$ | CPU | LU | CPU[1] |
|---|---|---|---|---|---|---|---|---|
| `utm5940` | 5940 | $4.35 \cdot 10^8$ | $3.98 \cdot 10^8$ | $7.21 \cdot 10^8$ | 4 | 0.17 | 55 | 0.12 |
| `dw8192` | 8192 | $3.81 \cdot 10^6$ | $3.81 \cdot 10^6$ | $5.07 \cdot 10^6$ | 6 | 0.10 | 49 | 0.08 |
| `grcar10000` | 10000 | $3.63 \cdot 10^0$ | $3.59 \cdot 10^0$ | $5.80 \cdot 10^0$ | 6 | 0.07 | 30 | 0.05 |
| `memplus` | 17758 | $1.29 \cdot 10^5$ | $1.29 \cdot 10^5$ | $2.47 \cdot 10^5$ | 6 | 0.17 | 57 | 0.13 |
| `af23560` | 23560 | $1.99 \cdot 10^4$ | $1.93 \cdot 10^4$ | $2.82 \cdot 10^4$ | 6 | 0.93 | 73 | 0.88 |
| `rajat16` | 96294 | $\star$ | $5.63 \cdot 10^{12}$ | $5.69 \cdot 10^{12}$ | 5 | 9.29 | 97 | 9.19 |
| `torso1` | 116158 | $\star$ | $1.41 \cdot 10^{10}$ | $1.42 \cdot 10^{10}$ | 3 | 26.8 | 94 | 28.5 |
| `dc1` | 116835 | $\star$ | $2.39 \cdot 10^8$ | $4.59 \cdot 10^8$ | 5 | 5.87 | 94 | 5.57 |
| `twotone` | 120750 | $\star$ | $1.75 \cdot 10^9$ | $2.91 \cdot 10^9$ | 4 | 1.37 | 75 | 1.33 |
| `FEM-3D-thermal2` | 147900 | $\star$ | $3.05 \cdot 10^3$ | $5.15 \cdot 10^3$ | 7 | 13.1 | 80 | 12.7 |
| `xenon2` | 157464 | $\star$ | $4.29 \cdot 10^4$ | $8.14 \cdot 10^4$ | 7 | 19.4 | 83 | 19.6 |
| `crashbasis` | 160000 | $\star$ | $6.30 \cdot 10^2$ | $1.21 \cdot 10^3$ | 7 | 3.35 | 64 | 2.59 |
| `scircuit` | 170998 | $\star$ | $2.40 \cdot 10^9$ | $4.69 \cdot 10^9$ | 7 | 2.11 | 58 | 1.39 |
| `transient` | 178866 | $\star$ | $1.02 \cdot 10^{11}$ | $2.00 \cdot 10^{11}$ | 8 | 7.76 | 87 | 7.12 |
| `stomach` | 213360 | $\star$ | $4.62 \cdot 10^1$ | $9.02 \cdot 10^1$ | 6 | 13.7 | 80 | 13.7 |

Table 2.2: The approximations of the condition number $\kappa$ of different matrices using extended Lanczos bidiagonalization. The method gives a lower bound $\kappa_{\text{low}}$ for $\kappa$ and also a probabilistic upper bound $\kappa_{\text{up}}$ that holds with probability at least 98% ($\varepsilon = 0.01$). The method continues until the ratio $\kappa_{\text{up}}(A)/\kappa_{\text{low}}(A)$ is below the indicated level of $\zeta = 2$. The number of steps $k$ needed to obtain this ratio and the CPU-time in seconds are shown. Also the percentage of the time taken by the LU-decomposition is displayed. Lastly we give CPU[1] of `condest(A)`. The symbol $\star$ is used when the value is too expensive to compute.

| Matrix $A$ | Dim. | $\kappa$ | $\kappa_{\text{low}}$ | $\kappa_{\text{up}}$ | $k$ | CPU | LU |
|---|---|---|---|---|---|---|---|
| utm5940 | 5940 | $4.35 \cdot 10^8$ | $4.35 \cdot 10^8$ | $4.71 \cdot 10^8$ | 10 | 0.36 | 60 |
| dw8192 | 8192 | $3.81 \cdot 10^6$ | $3.81 \cdot 10^6$ | $4.06 \cdot 10^6$ | 9 | 0.13 | 35 |
| grcar10000 | 10000 | $3.63 \cdot 10^0$ | $3.62 \cdot 10^0$ | $3.97 \cdot 10^0$ | 13 | 0.13 | 20 |
| memplus | 17758 | $1.29 \cdot 10^5$ | $1.29 \cdot 10^5$ | $1.41 \cdot 10^5$ | 15 | 0.28 | 31 |
| af23560 | 23560 | $1.99 \cdot 10^4$ | $1.99 \cdot 10^4$ | $2.12 \cdot 10^4$ | 9 | 1.10 | 63 |
| rajat16 | 96294 | $\star$ | $5.63 \cdot 10^{12}$ | $5.69 \cdot 10^{12}$ | 5 | 9.39 | 97 |
| torso1 | 116158 | $\star$ | $1.41 \cdot 10^{10}$ | $1.42 \cdot 10^{10}$ | 3 | 26.9 | 94 |
| dc1 | 116835 | $\star$ | $2.39 \cdot 10^8$ | $2.45 \cdot 10^8$ | 8 | 6.01 | 92 |
| twotone | 120750 | $\star$ | $1.75 \cdot 10^9$ | $1.91 \cdot 10^9$ | 7 | 1.69 | 64 |
| FEM-3D-thermal2 | 147900 | $\star$ | $3.15 \cdot 10^3$ | $3.43 \cdot 10^3$ | 12 | 15.1 | 70 |
| xenon2 | 157464 | $\star$ | $4.32 \cdot 10^4$ | $4.67 \cdot 10^4$ | 14 | 22.5 | 71 |
| crashbasis | 160000 | $\star$ | $6.40 \cdot 10^2$ | $7.01 \cdot 10^2$ | 18 | 5.21 | 40 |
| scircuit | 170998 | $\star$ | $2.45 \cdot 10^9$ | $2.67 \cdot 10^9$ | 16 | 3.29 | 37 |
| transient | 178866 | $\star$ | $1.03 \cdot 10^{11}$ | $1.11 \cdot 10^{11}$ | 21 | 9.24 | 73 |
| stomach | 213360 | $\star$ | $4.82 \cdot 10^1$ | $5.24 \cdot 10^1$ | 14 | 17.3 | 64 |

Table 2.3: The approximations of the condition number $\kappa$ of different matrices using extended Lanczos bidiagonalization. The method gives a lower bound $\kappa_{\text{low}}$ for $\kappa$ and also a probabilistic upper bound $\kappa_{\text{up}}$ that holds with probability at least 98% ($\varepsilon = 0.01$). The method continues until the ratio $\kappa_{\text{up}}(A)/\kappa_{\text{low}}(A)$ is below the indicated level of $\zeta = 1.1$. The number of steps $k$ needed to obtain this ratio and the CPU-time in seconds are shown. Also the percentage of the time taken by the LU-decomposition is displayed. The symbol $\star$ is used when the value is too expensive to compute.

Lanczos iterations with $A^TA$ to a starting vector $v$, providing an approximation to $\sigma_1(A)$ from the standard Krylov subspace $\mathcal{K}_{k+1}(A^TA, v) = \mathcal{K}_{1,k+1}(A^TA, v)$. Subsequently, one applies $k$ Lanczos iterations with $(A^TA)^{-1}$ to the same starting vector $v$, giving an approximation to $\sigma_n(A)$ from $\mathcal{K}_{k+1}((A^TA)^{-1}, v) = \mathcal{K}_{k+1,1}(A^TA, v)$. Together these two values form a lower bound for $\kappa(A)$ as in (2.9). The lower bound of extended Lanczos bidiagonalization is at least as good as the lower bound obtained by the alternative approach, as the former approach considers subspaces of the extended space $\mathcal{K}_{k+1,k+1}(A^TA, v)$. Furthermore, since in the extended Lanczos bidiagonalization procedure we can control the ratio $\zeta$, a natural stopping criterion arises for this method, as well as a good measure of the quality of both upper and lower bound. For the other approach these features are both missing.

As an example, the lower bound of $\kappa(A)$ for the matrix $A =$ `af23560` (taken from [81]) using extended Lanczos bidiagonalization ($k = 6$) is $1.93 \cdot 10^4$ in 0.93 seconds. Using twice a Lanczos procedure ($k = 6$) gives the lower bound $1.87 \cdot 10^4$ in 0.99 seconds. For the same number of steps, the matrix `stomach` (taken from [81]) gives $4.62 \cdot 10^1$ for extended Lanczos bidiagonalization (13.7 seconds) and $4.54 \cdot 10^1$ for the alternative approach (14.5 seconds). Besides a better lower bound, an important advantage of extended Lanczos bidiagonalization is that, almost for free, a probabilistic upper bound is provided as well. Note that in this example the CPU time for extended Lanczos bidiagonalization includes the time for the computation of the probabilistic upper bounds.

**Experiment 2.7.4** Another alternative to approximate the condition number of $A$ is to use the `svds` command in MATLAB. We compared our method, with the parameters $\zeta = 1.1$ and $\varepsilon = 0.01$, to the outcome of the command `svds(A,1,'L')/svds(A,1,0)`. The results in table 2.4 show that our method significantly outperforms the `svds` approach concerning the running time (in these examples our method is 8 to 13 times faster), giving the same lower bound for $\kappa(A)$. Again, as stated in the previous experiment, our method also gives a probabilistic upper bound for the condition number, almost for free.

Finally some words on a breakdown. A breakdown takes place when the method has found an invariant subspace. This is a rare event; in exact arithmetic the probability that this happens for a $k \ll n$ is zero since we have selected a random vector (a $k$-dimensional subspace of

| Matrix | Ext LBD | CPU | svds | CPU |
|---|---|---|---|---|
| memplus | $1.294 \cdot 10^5$ | 0.28 | $1.294 \cdot 10^5$ | 2.30 |
| af23560 | $1.988 \cdot 10^4$ | 1.10 | $1.989 \cdot 10^4$ | 15.1 |
| rajat16 | $5.629 \cdot 10^{12}$ | 9.39 | $5.629 \cdot 10^{12}$ | 100.9 |

Table 2.4: For three matrices the lower bound given by the extended Lanczos bidiagonalization (Ext LBD), with the parameters $\zeta = 1.1$ and $\varepsilon = 0.01$, and the corresponding CPU time in seconds. Also the bound given by the procedure using svds and the corresponding CPU time in seconds is shown.

$\mathbb{R}^n$ has zero measure). A breakdown has not been encountered in our numerical experiments. However, it might happen in rare cases that in the algorithm $\alpha_{-j}$ (Step 5), $\beta_j$ (Step 13), $\|u\|$ (Step 15) or $\delta_{j+1}$ (Step 28) are zero or very small. In such a case, we can just stop the method, and return the lower and probabilistic bounds obtained before the breakdown. If these do not yet satisfy the requirements of the user, we can restart the method with a new random vector. An extra run of the extended Lanczos bidiagonalization method will not significantly increase the overall costs. With this adaptation we trust that the method can result in a robust implementation for the use in libraries.

## 2.8 FINAL CONSIDERATIONS

We have proposed a new extended Lanczos bidiagonalization method. This method leads to tridiagonal matrices with a special structure. The method provides a lower bound for $\kappa(A)$ of good quality and a probabilistic upper bound for $\kappa(A)$ that holds with a user-chosen probability $1 - 2\varepsilon$. Although we have not encountered any breakdown in the experiments, the algorithm may abort (in case of a division by zero) and not return any estimate. When choosing $k$ adaptively, given a user-selected $\varepsilon$ and desired ratio $\kappa_{\text{up}}(A)/\kappa_{\text{low}}(A) < \zeta$, the results show that generally this $k$ is fairly small, even for $\zeta = 1.1$. Only 3 vectors of storage are required. This method can be used whenever an LU-factorization is computable in a reasonable amount of time. When this is not an option, methods such as the one in [44, 45] can be used.

# 3

# APPROXIMATING LEADING SINGULAR TRIPLETS OF A MATRIX FUNCTION

*Adapted from [28]*

Given a large square matrix $A$ and a sufficiently regular function $f$ so that $f(A)$ is well defined, we are interested in the approximation of the leading singular values and corresponding left and right singular vectors of $f(A)$, and in particular in the approximation of $\|f(A)\|$, where $\|\cdot\|$ is the matrix norm induced by the Euclidean vector norm. Since neither $f(A)$ nor $f(A)v$ can be computed exactly, we introduce a new *inexact* Golub-Kahan-Lanczos bidiagonalization procedure, where the inexactness is related to the inaccuracy of the operations $f(A)v$, $f(A)^*v$. Particular outer and inner stopping criteria are devised so as to cope with the lack of a true residual. Numerical experiments with the new algorithm on typical application problems are reported.

## 3.1 INTRODUCTION

Given a large $n \times n$ complex matrix $A$ and a sufficiently regular function $f$ so that $f(A)$ is well defined, we are interested in approximating the largest singular values and corresponding left and right singular vectors of the matrix function $f(A)$; we shall refer to these three quantities as *triplet*. This computation will also give an approximation to the 2-norm $\|f(A)\|$, where $\|\cdot\|$ is the matrix norm induced by the Euclidean vector norm, as defined in (1.1). In this chapter we will mainly discuss this norm approximation because of its relevance for applications. However, we shall keep in mind that the considered procedure allows us to also determine singular triplets $(\sigma, u, v)$, and that a group of singular triplets can be determined simultaneously.

### 3.1.1 *Appearances of the norm of a matrix function*

The problem of approximating the norm of a matrix function arises in the solution of stiff linear initial value problems [29, 71], in the evaluation of derivatives and perturbations of matrix functions, which arise

for instance in electronic structure theory [30, 54, 68], and in monitoring the magnitude of the inverse of distance matrices [2]. In numerical linear algebra the norm of matrix polynomials may be used in the analysis of iterative procedures, and in the norm of rational matrix functions. In particular the transfer function may give information on the sensitivity of the matrix itself to perturbations; see, e.g., [11, 83] and their references. Some of the functions we considered in our numerical experiments are related to the explicit solution of time-dependent differential equations. Physical phenomena may call for moderately accurate approximations of $\|f(tA)\|$ to analyze the transient phase of the solution $u(t) = f(tA)u(0)$ for $A$ nonnormal; see, e.g., [83, chapter 14]. We also recall that the best rank $p$ 2-norm approximation to $f(A)$ is the one formed by the $p$ leading singular triplets; this approximation can be used for instance in model order reduction where the quantity $u^* f(A)v$ may have to be approximated for a sequence of vectors $u, v$; see, e.g., [62, section 3.9] and references therein.

### 3.1.2 *Contributions of this chapter to the problem*

If $A$ were normal, then the approximation could be stated in terms of an eigenvalue problem in $A$. Indeed, if $A = Q\Lambda Q^*$ is the eigendecomposition of $A$ with $Q$ unitary and $\Lambda$ diagonal, then $f(A) = Qf(\Lambda)Q^*$ [47], so that the leading singular values of $f(A)$ could be determined by a procedure that approximates the eigenvalues of $A$.

The problem is significantly more challenging if $A$ is large and nonnormal, since there is no relation between eigenvalues and singular values that can be readily exploited during the computation. Moreover, although $A$ may be sparse, in general $f(A)$ will be dense, and it cannot be computed explicitly. We are thus left with procedures that use $f$ and $A$ by means of the action of $f(A)$ to a vector $v$. As stated in chapter 1, Lanczos bidiagonalization is a commonly used strategy for approximating selected singular triplets of a given matrix. Given a matrix $F$, this procedure generates a sequence of orthonormal vectors $\{v_1, v_2, \ldots\}$ and $\{u_1, u_2, \ldots\}$ by alternating products of $Fv$ and $F^*u$. In our case, $F = f(A)$ with $A$ of large dimensions, therefore these matrix vector products cannot be computed exactly and the standard Lanczos bidiagonalization process fails.

We introduce a novel *inexact* Lanczos bidiagonalization process, where at each iteration the action of $f(A)v$ and $f(A)^*u$ is approximated with some loose tolerance by means of a projection method. The problem of approximating $f(A)v$ has seen a great interest growth in the past fifteen years, due to the emerging occurrence of this computation in many scientific and engineering applications; see, e.g., [6, 19, 25, 36, 39, 42, 43], and their references. For our purposes we shall use Krylov subspace methods for approximating $f(A)^*u$ and $f(A)v$ at each iteration, equipped with a cheap stopping criterion that may also be adapted to the outer current accuracy. We shall show that the inexactness in the Lanczos bidiagonalization causes the loss of the known symmetry structure of the process. Nonetheless, as is the case in finite precision analysis [60], orthogonality of the basis can be preserved, so that the recurrence maintains its effectiveness.

### 3.1.3 *Overview of the chapter*

If a rough approximation to $\|f(A)\|$ is the only quantity of interest, instead of a group of singular triplets, other approaches could be considered. For instance, $f$ could be approximated by some other more convenient functions, and then the resulting matrix function norm could be more easily estimated. As an alternative, equivalent definitions of $f(A)$ could be used, from which the norm could also be estimated. Or, the relation of $\|f(A)\|$ with other norms or with some other spectral tool could be used. Some of these approaches are briefly recalled in section 3.2. Methods in the mentioned classes, however, usually at most provide the order of magnitude of the actual norm and are thus inappropriate if more correct digits are needed.

In section 3.3 the standard Lanczos bidiagonalization is recalled and the general notation used in this chapter is introduced. Section 3.4 presents the inexact Lanczos bidiagonalization procedure, including the details on the stopping criteria in section 3.4.1. Section 3.5 discusses the approximation of the matrix function multiplication, and a stopping criterion for its accuracy, while in section 3.6 a stopping criterion in the case of an inner flexible strategy is analyzed. In section 3.6.1 we show how specific spectral properties allow us to make a variable accuracy for the inner iteration feasible, which is finalized in section 3.6.2. Section 3.7 focuses on the practical implementation and the numerical

results are presented in section 3.8. We will conclude with some discussion in section 3.9.

## 3.2 AVAILABLE TECHNIQUES FOR ESTIMATING THE NORM

The Lanczos bidiagonalization method is widely recognized as the method of choice for approximating selected singular triplets of a large matrix. However, if one is only interested in estimates of $\|f(A)\|_2$ with $A$ non-Hermitian, rather different procedures could also be used. A simple approach consists of roughly estimating $\|\cdot\|_2$ by using some other matrix norm. For instance,

$$\|f(A)\|_2 \leq \sqrt{n}\|f(A)\|_p, \quad p = 1, \infty,$$

or

$$\|f(A)\|_2 \leq \sqrt{\|f(A)\|_1 \, \|f(A)\|_\infty},$$

where $\|f(A)\|_p$ is once again an induced norm [48, p. 365]. These bounds are usually pessimistic, and they are clearly unsatisfactory for $n$ large. The fact that for $A$ large the entries of $f(A)$ are not all readily available provides an additional challenge.

In the following we describe a few approaches available in the literature that are tailored to the matrix function case. The first three initially determine an explicit upper bound for the norm, which only depends on scalar quantities. The core computation will then be to provide a good approximation to the theoretical upper bound. The quality of the final estimate of $\|f(A)\|_2$ will thus depend both on the sharpness of the initial upper bound and on the accuracy of the computation. For general nonnormal matrices the initial bound is often not very sharp, limiting the quality of the overall estimation. Finally, a purely computation-oriented estimation is obtained with the power method, which directly approximates $\|f(A)\|_2$ as the square root of the largest eigenvalue of $f(A)^* f(A)$. A more detailed list follows.

1. Let $r(A)$ be the numerical radius of $A$, that is $r(A) = \max\{|z| : z \in W(A)\}$, where $W(A)$ is the field of values of $A$. Since (see, e.g., [35, theorem 1.3-1])

$$r(A) \leq \|A\|_2 \leq 2\, r(A),$$

by applying the bounds to $f(A)$ instead of $A$, it is possible to estimate $\|f(A)\|_2$ by means of $r(f(A))$; see, e.g., [65, 86] for numerical methods to compute the numerical radius of a given matrix. A related special case is given by the exponential function, for which the bound

$$\|\exp(A)\| \leq \exp(\alpha) \tag{3.1}$$

holds, where $\alpha$ is the largest eigenvalue of the Hermitian part of $A$, that is of $\frac{1}{2}(A + A^*)$ [39, section 10.1].

2. If it is possible to find $K > 0$ and $\Omega \subset \mathbb{C}$ such that

   $$\|f(A)\|_2 \leq K\|f\|_\Omega,$$

   then it is sufficient to estimate $\|f\|_\Omega$; here $\|f\|_\Omega$ is the $L_\infty$-norm of $f$ on $\Omega$. This can be done for instance when $f$ is a polynomial, for which $K$ is known to be less than 11.08 and conjectured to be equal to 2, and $\Omega$ coincides with the field of values of $A$, see [12] for its derivation. We refer to the Ph.D. thesis of D. Choi [11], for a discussion on the use of this bound when $A$ is normal, or when $A$ is a contraction; see also [83] for a detailed analysis of this bound when using pseudospectral information. The computationally intensive task is given by the determination of $\Omega$. If $\Omega$ coincides with $W(A)$, then the cost of accurately approximating $\Omega$ may be higher than that of approximating the single quantity $\|f(A)\|_2$.

3. This approach is in the same spirit as the one above. For $\varepsilon > 0$, let $\sigma_\varepsilon(A) = \{z \in \text{spec}(A + E) : \|E\| < \varepsilon\}$ and assume that $f$ is analytic in $\sigma_\varepsilon(A)$. If $L_\varepsilon$ denotes the length of the boundary $\partial\sigma_\varepsilon(A) = \{z \in \mathbb{C} : \|(zI - A)^{-1}\| = \varepsilon^{-1}\}$, then by using the Cauchy integral expression for $f(A)$ we obtain (see, e.g., [83])

   $$\|f(A)\| \leq \frac{L_\varepsilon}{2\pi\varepsilon}\|f\|_{\partial\sigma_\varepsilon}.$$

   Although the involved quantities may be easier to compute than in the previous case, the dependence on $\varepsilon > 0$ remains not fully controllable.

4. Using the relation $\|f(A)\|_2^2 = \lambda_{\max}(f(A)^* f(A))$ a run of a few iterations of the power method can give an estimate to the value

$\lambda_{\max}(f(A)^* f(A))$; see, e.g., [39, algorithm 3.19] for an algorithm specifically designed for the largest singular triplet.

The power method is probably the most appealing approach among the ones listed above. If a rough approximation is required, typically to determine the order of magnitude, then the power method provides a satisfactory answer in a few iterations. However, if more than one digit of accuracy is required, then the process may need many iterations to converge. As far as $A$ is concerned, the stability in the computation may be highly influenced by the squaring; we refer to section 3.8 for an example of this well known phenomenon.

## 3.3 LANCZOS BIDIAGONALIZATION

We start by recalling the Lanczos bidiagonalization process, introduced in section 1.2.2, in the context of this chapter in terms of the matrix function $f(A)$. Then we will discuss how to actually obtain $f(A)$ times a vector. Let $u_0 = 0$ and $\beta_1 = 0$, and given the vector $v_1$ of unit norm, then for $j = 1, \dots, m$ the following recurrence relations define the algorithm for the Lanczos bidiagonalization

$$\begin{aligned} \beta_{2j} u_j &= f(A) v_j - \beta_{2j-1} u_{j-1}, \\ \beta_{2j+1} v_{j+1} &= f(A)^* u_j - \beta_{2j} v_j. \end{aligned} \tag{3.2}$$

The coefficients $\beta_{2j}$ and $\beta_{2j+1}$ are computed so that the corresponding vectors $u_j$ and $v_{j+1}$ have unit norm. By collecting the two sets of vectors as $U_m = [u_1, \dots, u_m]$, $V_m = [v_1, \dots, v_m]$, we observe that $U_m^* U_m = I$, $V_m^* V_m = I$, $V_m^* v_{m+1} = 0$. Moreover, the two recurrences can be compactly written as

$$\begin{aligned} f(A) V_m &= U_m B_m, \\ f(A)^* U_m &= V_m B_m^* + \beta_{2m+1} v_{m+1} e_m^*, \end{aligned} \tag{3.3}$$

where $B_m$ is the following bidiagonal matrix

$$B_m = \begin{bmatrix} \beta_2 & \beta_3 & & \\ & \beta_4 & \beta_5 & \\ & & \ddots & \ddots \end{bmatrix} \in \mathbb{R}^{m \times m}.$$

It can be shown that the columns of $V_m$ span the Krylov subspace $\mathcal{K}_m(f(A)^*f(A), v_1)$ and the columns of $U_m$ span the Krylov subspace $\mathcal{K}_m(f(A)f(A)^*, f(A)v_1)$. Define

$$\mathcal{B}_{2m} = \begin{bmatrix} 0 & B_m \\ B_m^* & 0 \end{bmatrix}, \qquad \text{and} \qquad \mathcal{W}_{2m} = \begin{bmatrix} U_m & 0 \\ 0 & V_m \end{bmatrix},$$

and

$$\mathcal{F} = \begin{bmatrix} 0 & f(A) \\ f(A)^* & 0 \end{bmatrix}.$$

Then the recursion (3.3) can be rewritten in the more compact matrix notation

$$\mathcal{F}\mathcal{W}_{2m} = \mathcal{W}_{2m}\mathcal{B}_{2m} + \beta_{2m+1} \begin{bmatrix} 0 \\ v_{m+1} \end{bmatrix} e_m^*, \quad e_m \in \mathbb{R}^{2m}. \tag{3.4}$$

Via a permutation this expression reflects its equivalence to the standard Lanczos method for the symmetric matrix $\mathcal{F}$, for which the equality $\text{span}(\mathcal{W}_{2m}) = \mathcal{K}_{2m}(\mathcal{F}, [0; v_1])$ holds, see also [13, p. 178–186], [32, p. 486]. The eigenvalues of $\mathcal{B}_{2m}$ occur in $\pm$ pairs. Within this setting, it is thus possible to approximate the singular values of $f(A)$ by the positive eigenvalues of $\mathcal{B}_{2m}$, or, equivalently, by the singular values of $B_m$. In particular, for the largest singular value it holds that (see [47, corollary 3.1.3, lemma 3.3.1.])

$$\sigma_1(B_{j-1}) \leq \sigma_1(B_j) \leq \sigma_1(f(A)), \qquad 2 \leq j \leq m.$$

There are several advantages of the Lanczos bidiagonalization over the simpler power method applied to $f(A)^*f(A)$, which are mainly related to the fact that the eigenvalue squaring in this latter problem may lead to severe loss of information in the case very small or very large singular values arise. In the inexact case the bidiagonal formulation also allows us to better trace the inexactness during the whole approximation process; this is discussed in the next section.

## 3.4 INEXACT LANCZOS BIDIAGONALIZATION

When neither the explicit computation of the matrix $f(A)$ nor the accurate operation $f(A)v$ (or $f(A)^*u$) are feasible, then approximate computations must be performed, resulting in an *inexact* Lanczos bidiagonalization procedure. As a consequence, the recurrence (3.3) needs to

be significantly revised so as to acknowledge for the quantities that are actually computed.

For a given $v$, the exact matrix-vector multiplication $f(A)v$ has to be replaced by an inner procedure that approximates the resulting vector up to a certain accuracy. The same holds for the operation $f(A)^*u$ for a given vector $u$. For the sake of the analysis, at each iteration $j$ we shall formalize this difference by writing, for some matrices $C_j$ and $D_j$,

$$\begin{aligned} \beta_{2j}u_j &= (f(A)v_j + C_jv_j) - \beta_{2j-1}u_{j-1}, \\ \beta_{2j+1}v_{j+1} &= (f(A)^*u_j + D_ju_j) - \beta_{2j}v_j, \end{aligned}$$

where $C_j, D_j$ implicitly represent the perturbation induced by the approximate computations. Since in general $f(A)^* + D_j$ is no longer the conjugate transpose of $f(A) + C_j$, orthogonality of a new vector $v_{m+1}$ has to be enforced by explicit orthogonalization against all previous vectors $v_j$, $1 \le j \le m$. The same holds for the vectors $u_j$, $j = 1, \ldots, m$. Therefore, instead of one bidiagonal matrix $B_m$ in the exact relation, we now obtain an upper triangular matrix $M_m$ and an upper Hessenberg matrix $T_m$. This leads to the following relations for the inexact (perturbed) Lanczos bidiagonalization:

$$\begin{aligned} (f(A) + \mathfrak{C}_m)V_m &= U_mM_m, \\ (f(A)^* + \mathfrak{D}_m)U_m &= V_mT_m + t_{m+1,m}v_{m+1}e_m^*, \end{aligned} \tag{3.5}$$

where $\mathfrak{C}_m = \sum_{j=1}^m C_jv_jv_j^*$ and $\mathfrak{D}_m = \sum_{j=1}^m D_ju_ju_j^*$. The matrices $V_m$ and $U_m$ are different from the matrices in the exact relation, but they still have orthonormal columns.

The inexact Lanczos bidiagonalization can also be described using the notation of (3.4). Define

$$\widetilde{\mathcal{B}}_{2m} = \begin{bmatrix} 0 & M_m \\ T_m & 0 \end{bmatrix}, \qquad \mathcal{W}_{2m} = \begin{bmatrix} U_m & 0 \\ 0 & V_m \end{bmatrix},$$

and the perturbation matrix

$$\mathcal{G}_{2m} = \begin{bmatrix} 0 & \mathfrak{C}_m \\ \mathfrak{D}_m & 0 \end{bmatrix} \mathcal{W}_{2m} =: \mathcal{E}_m\mathcal{W}_{2m}.$$

The perturbed relation thus becomes

$$\mathcal{F}\mathcal{W}_{2m} + \mathcal{G}_{2m} = \mathcal{W}_{2m}\widetilde{\mathcal{B}}_{2m} + t_{m+1,m}\begin{bmatrix} 0 \\ v_{m+1} \end{bmatrix} e_m^*, \tag{3.6}$$

where $e_m \in \mathbb{R}^{2m}$, and

$$\begin{aligned}\mathcal{F}\mathcal{W}_{2m} + \mathcal{G}_{2m} &= (\mathcal{F} + \mathcal{E}_m)\mathcal{W}_{2m} \\ &= \begin{bmatrix} 0 & f(A) + \mathfrak{C}_m \\ f(A)^* + \mathfrak{D}_m & 0 \end{bmatrix} \mathcal{W}_{2m} =: \widetilde{\mathcal{F}}_{2m}\mathcal{W}_{2m}.\end{aligned}$$

In contrast to the exact case, the space spanned by the columns of $\mathcal{W}_{2m}$ is not a Krylov subspace. However, when $t_{m+1,m}$ is small, this new space is close to an invariant subspace of the perturbed matrix $\widetilde{\mathcal{F}}_{2m}$, because then $\widetilde{\mathcal{F}}_{2m}\mathcal{W}_{2m} \approx \mathcal{W}_{2m}\widetilde{\mathcal{B}}_{2m}$. Notice the similarity of (3.6) with equation (3.1) in [73], which shows that with this formulation, the inexact projection problem amounts to solving a structured eigenvalue problem, where the original Hermitian matrix $\mathcal{F}$ has been perturbed by a structured non-Hermitian perturbation $\mathcal{E}_m$. The theory in [73] can then be used to analyze and monitor the inexact computations, although the general results in [73] should be carefully adapted to the new problem structure.

If $\mathcal{E}_m$ is small in norm, the eigenvalues of the *non-Hermitian* matrix $\widetilde{\mathcal{F}}_{2m}$ are small perturbations of the eigenvalues of the *Hermitian* matrix $\mathcal{F}$. Indeed, the eigenvalues of the perturbed matrix $\widetilde{\mathcal{F}}_{2m}$ lie in discs with radius $\|\mathcal{E}_m\|$ and center the (real) eigenvalues of $\mathcal{F}$ (see, e.g., [76, section IV, theorem 5.1]). Therefore, for small perturbations in the computations, the eigenvalues of the symmetric matrix $\mathcal{F}$ will be perturbed accordingly. On the other hand, in the following we shall consider the case when $\|\mathcal{E}_m\|$ is larger than usually allowed by a perturbation analysis argument. Therefore, different strategies need to be devised to ensure good approximations to the wanted eigenvalues of $\mathcal{F}$.

Following the standard procedure of the exact case, we should consider the matrix $\widetilde{\mathcal{B}}_{2m}$ to approximate the largest eigenpairs of $\widetilde{\mathcal{F}}_{2m}$, and according to the discussion above, of $\mathcal{F}$. Due to the non-Hermitian structure of $\widetilde{\mathcal{B}}_{2m}$, however, there are different matrices that can provide the sought after singular value information, namely the matrix $\widetilde{\mathcal{B}}_{2m}$ itself, and the two distinct matrices $T_m$ or $M_m$. The last two matrices yield approximations to the corresponding singular triplets of $f(A) + \mathfrak{C}_m$ and $f(A)^* + \mathfrak{D}_m$. The following bound between the largest eigenvalue of $\widetilde{\mathcal{B}}_{2m}$ and the largest singular values of $T_m$ and $M_m$ shows

that all these quantities can be easily related. Let $q = [x; y]$ and let $\theta$ be an eigenvalue of $\widetilde{\mathcal{B}}_{2m}$. Using $\|x\|\|y\| \leq \frac{1}{2}\left(\|x\|^2 + \|y\|^2\right)$ we obtain[1]

$$\begin{aligned}|\theta| &\leq \max_{\mathbf{q}\neq 0}\left|\frac{q^*\widetilde{\mathcal{B}}_{2m}q}{q^*q}\right| = \max_{[x;y]\neq 0}\left|\frac{x^*M_m y + y^*T_m x}{x^*x + y^*y}\right| \\ &\leq \max_{[x;y]\neq 0}\frac{\|x\|\|M_m y\|}{\|x\|^2+\|y\|^2} + \max_{[x;y]\neq 0}\frac{\|y\|\|T_m x\|}{\|x\|^2+\|y\|^2} \\ &\leq \tfrac{1}{2}(\sigma_1(M_m) + \sigma_1(T_m)).\end{aligned}$$

If the inexactness of the bidiagonalization is very large, $M_m$ and $T_m^*$ are very different from each other. In this case, the leading singular values of these two matrices - and thus their mean - may be significantly larger than the biggest (in modulus) eigenvalue of $\widetilde{\mathcal{B}}_{2m}$, since they are related to the numerical radius of $\widetilde{\mathcal{B}}_{2m}$, rather than to its spectrum. This motivated us to use the eigenvalues of $\widetilde{\mathcal{B}}_{2m}$ in the approximation, rather than the singular values of its blocks. Moreover, working with $\widetilde{\mathcal{B}}_{2m}$ made the analysis of the relaxed strategy particularly convenient, since known results on relaxed eigenvalue computation could be exploited.

### 3.4.1 *A computable stopping criterion*

In this section we analyze a strategy for monitoring the convergence of the inexact bidiagonal iteration. Some stopping criterion needs to be introduced to exit the process. This can be based for instance on the problem residual. As it is common to other inexact processes, the true problem residual is inaccessible as soon as inexactness takes place, so one could for example use an approximation to the true residual. As a result, however, the computed approximate residual may no longer be meaningful for the original problem, if the approximation was too coarse. We thus discuss a meaningful and computable stopping criterion.

Let $(\theta, q)$ be an eigenpair of $\widetilde{\mathcal{B}}_{2m}$, where $q$ is a unit vector. As the iterations proceed, $(\theta, \mathcal{W}_{2m}q)$ tends to approximate an eigenpair of $\widetilde{\mathcal{F}}_{2m}$. We would like to ensure that $(\theta, \mathcal{W}_{2m}q)$ also tends to an eigenpair of $\mathcal{F}$. To monitor the convergence of $\theta$ and to define a stopping criterion for the outer iteration, the residual is used. We call $\mathcal{F}\mathcal{W}_{2m}q - \theta\mathcal{W}_{2m}q$ the *true residual*, which is not available, since $\mathcal{F}$ cannot be applied exactly.

1 Another bound can be obtained for the geometric mean, that is $|\theta| \leq \sqrt{\sigma_1(M_m)\sigma_1(T_m)}$.

We thus introduce the *computed residual*, which is the residual of the actually computed quantities, namely (see (3.6))

$$r_{2m} := \widetilde{\mathcal{F}}_{2m}\mathcal{W}_{2m}q - \theta\mathcal{W}_{2m}q = t_{m+1,m}\begin{bmatrix}0\\ v_{m+1}\end{bmatrix}e_m^*q, \quad (e_m \in \mathbb{R}^{2m}).$$

In the sequel, we shall use the following obvious inequality to estimate the true residual norm:

$$\|\mathcal{F}\mathcal{W}_{2m}q - \theta\mathcal{W}_{2m}q\| \le \|r_{2m}\| + \|(\mathcal{F}\mathcal{W}_{2m}q - \theta\mathcal{W}_{2m}q) - r_{2m}\|, \quad (3.7)$$

where $\|(\mathcal{F}\mathcal{W}_{2m}q - \theta\mathcal{W}_{2m}q) - r_{2m}\|$ is the gap between the computed and the true residuals, in short the "residual gap". If this gap can be imposed to be small, then the computed residual will give an estimate for the true residual. In this case, convergence can be monitored by only using the (available) computed residual. More precisely, the following relative stopping criterion can be employed:

$$\text{if} \quad \frac{|t_{m+1,m}e_m^*q|}{|\theta|} < \varepsilon_{out} \quad \text{then} \quad \text{stop} \quad (3.8)$$

for some outer tolerance $\varepsilon_{out}$, where $\theta$ is the largest (in modulus) eigenvalue. Finally, as the computed residual norm goes to zero, the quantity $\|(\mathcal{F}\mathcal{W}_{2m}q - \theta\mathcal{W}_{2m}q) - r_{2m}\|$ will tend to dominate again, playing the role of the final attainable accuracy level.

To see how we can impose the residual gap to be small, and recalling the definition of $\mathcal{G}_{2m}$, we first consider a more convenient expression for $\mathcal{G}_{2m}q$, with $q = [x; y]$, that is

$$\mathcal{G}_{2m}q = \begin{bmatrix}\mathfrak{C}_m V_m y\\ \mathfrak{D}_m U_m x\end{bmatrix} =: \begin{bmatrix}G_m^{(1)}y\\ G_m^{(2)}x\end{bmatrix}.$$

Let $G_m^{(\ell)} = [g_1^{(\ell)}, \ldots, g_m^{(\ell)}]$, for $\ell = 1, 2$. Then

$$\begin{aligned}\|(\mathcal{F}\mathcal{W}_{2m}q - \theta\mathcal{W}_{2m}q) - r_{2m}\| = \|\mathcal{G}_{2m}q\| &= \left\|\begin{bmatrix}G_m^{(1)}y\\ G_m^{(2)}x\end{bmatrix}\right\| \\ &= \left\|\sum_{j=1}^{m}\begin{bmatrix}g_j^{(1)}e_j^*y\\ g_j^{(2)}e_j^*x\end{bmatrix}\right\| \le \sum_{j=1}^{m}\left\|\begin{bmatrix}g_j^{(1)}e_j^*y\\ g_j^{(2)}e_j^*x\end{bmatrix}\right\| \\ &= \sum_{j=1}^{m}(\|g_j^{(1)}\|^2\,|e_j^*y|^2 + \|g_j^{(2)}\|^2\,|e_j^*x|^2)^{\frac{1}{2}}. \quad (3.9)\end{aligned}$$

The vectors $g_j^{(\ell)}$, $\ell = 1, 2$, implicitly carry the error associated with the inexact computation of $f(A)v$ and $f(A)^*u$, respectively, in the inner iteration. If every term of this sum is small, the computed residual will be close to the true residual. The following lemma relates the inaccuracy in the matrix-vector products to the residual gap; its proof is an adaptation of the corresponding result in [73], where the structure is exploited to highlight the dependence on $m$ while the problem has size $2m$.

**Lemma 3.4.1** *Assume that $m$ iterations of the inexact Lanczos bidiagonalization process have been taken. If $\|g_j^{(1)}\|, \|g_j^{(2)}\| < \frac{1}{m}\varepsilon$ for $1 \leq j \leq m$, then $\|(\mathcal{F}\mathcal{W}_{2m}q - \theta\mathcal{W}_{2m}q) - r_{2m}\| < \varepsilon$.*

*Proof.* From $\|q\| = 1$ with $q = [x; y]$ it follows that $\|[e_j^* x; e_j^* y]\| \leq 1$. From (3.9) we obtain

$$\begin{aligned}\|(\mathcal{F}\mathcal{W}_{2m}q - \theta\mathcal{W}_{2m}q) - r_{2m}\| &\leq \sum_{j=1}^{m} (\|g_j^{(1)}\|^2 |e_j^* y|^2 + \|g_j^{(2)}\|^2 |e_j^* x|^2)^{\frac{1}{2}} \\ &< \sum_{j=1}^{m} \frac{1}{m}\varepsilon (|e_j^* y|^2 + |e_j^* x|^2)^{\frac{1}{2}} \\ &\leq \frac{1}{m}\varepsilon \sum_{j=1}^{m} 1 = \varepsilon.\end{aligned}$$

□

This result shows that if $\varepsilon$ is sufficiently small, then the residual gap will stay below the computed residual norm until convergence. In our experiments, $m$ will play the role of the maximum number of Lanczos bidiagonalization iterations, which is usually set to a number between 50 and 500.

## 3.5 APPROXIMATION OF $f(A)v$ AND $f(A)^*u$

The performance of the inexact Lanczos bidiagonalization process depends on the approximation accuracy of the matrix-vector products $f(A)v$ and $f(A)^*u$. Due to the size of $A$, we consider approximating these quantities by means of a projection-type iterative method as follows; we limit our discussion to $f(A)v$, and a corresponding procedure can be used for $f(A)^*u$. We also notice that in general, $f(A)v$

and $f(A)^*u$ require distinct approximations. This is also true for functions satisfying $f(A^*) = f(A)^*$. Starting with the unit vector $v$ and the matrix $A$, we construct a sequence of approximation subspaces $\mathcal{K}_i$ of $\mathbb{R}^n$, $i = 1, 2, \ldots$, and define the matrix $P_i = [p_1, p_2, p_3, \ldots, p_i] \in \mathbb{C}^{n\times i}$, whose orthonormal columns span the subspace, and $v = p_1 = P_i e_1$, in a way such that the spaces are nested, that is $\mathcal{K}_i \subseteq \mathcal{K}_{i+1}$. Typical choices are Krylov and rational Krylov subspaces [36, 39]. The desired approximation is then obtained as

$$f(A)v \approx P_i f(H_i)e_1, \qquad H_i = P_i^* A P_i.$$

For small $i$, the reduced non-Hermitian matrix $H_i$ has small size, so that $f(H_i)$ can be computed efficiently by various strategies such as decomposition-type methods [39].

### 3.5.1 *A computable inner stopping criterion*

Our stopping criterion of this approximation process is based on an estimation of the error norm, and it uses an approach previously introduced in [56, proposition 2.2]; see also [22] for an earlier application to the exponential.

**Proposition 3.5.1** [56, proposition 2.2] *Assume that $i + d$ inner iterations have been executed. Let $z_{i+d} = P_{i+d} f(H_{i+d})e_1$ be an approximation to $f(A)v$ and define $\omega_{i+d} = \|z_{i+d} - z_i\|/\|z_i\|$. If $\|f(A)v - z_{i+d}\| \ll \|f(A)v - z_i\|$ and $\|f(A)v\| \approx \|z_i\|$, then*

$$\|f(A)v - z_i\| \approx \frac{\omega_{i+d}}{1 - \omega_{i+d}} \|z_i\|. \tag{3.10}$$

Under the stated hypotheses, the result in (3.10) shows that after $i + d$ iterations it is possible to provide an estimate of the error norm at iteration $i$. The first hypothesis aims to ensure that after $d$ additional iterations the error has decreased enough to be able to disregard the error at step $i + d$ compared to that at step $i$. The second hypothesis ensures that after $i$ iterations the approximate quantity is close, in terms of Euclidean norm, to the exact one. Similar results are employed in the algebraic linear system literature; see, e.g., [77]. The proposition above provides a theoretical ground for the following more heuristic stopping criterion for the approximation of $f(A)v$:

$$\text{if} \quad \frac{\omega_{i+d}}{1 - \omega_{i+d}} \leq \varepsilon_{in} \quad \text{then} \quad \text{stop}$$

for some inner tolerance $\varepsilon_{in}$. In the numerical experiments presented in section 3.8 we have used $d = 4$, and we assumed that the conditions of the above proposition were satisfied. The accuracy of the inner iteration will influence the final accuracy of the inexact Lanczos bidiagonalization. In the notation of the previous section, if after $i$ inner iterations the stopping criterion is satisfied, we have thus derived the following estimate for the perturbation occurring at the $j$th step of Lanczos bidiagonalization,

$$\|f(A)v_j - z_i\| = \|C_j v_j\| \approx \varepsilon_{in}\|z_i\|.$$

We recall that the matrix $C_j$ is not explicitly determined, and it is used to express the inexactness at iteration $j$ in terms of a matrix-vector product with $v_j$. Note that here $\|C_j v_j\| = \|g_j^{(1)}\|$, with the notation in (3.9). An analogous relation holds with respect to $f(A)^* u_j$ and thus $\|g_j^{(2)}\|$. Since the approximation process changes at each iteration, the threshold for the quantity $\|C_j v_j\|$ may vary as the Lanczos bidiagonalization proceeds, so that $\varepsilon_{in} = \varepsilon_{in}^{(j)}$. As experienced with other eigenvalue and linear system problems, $\varepsilon_{in}^{(j)}$ may even be allowed to grow during the iteration, without significantly affecting the overall process. This is discussed in the next section.

## 3.6 RELAXING THE INNER SOLUTION ACCURACY

The bound in (3.9) on the residual gap suggests that the accuracy of the inner solution approximation can be relaxed as convergence takes place. Indeed, following similar strategies in [10, 75, 73], we observe that to ensure a small gap it is the product $\|g_j^{(1)}\|\,|e_j^* y|$ in (3.9) that needs to be small, and not each factor; a similar argument holds for $\|g_j^{(2)}\|\,|e_j^* x|$. Therefore, if $|e_j^* y|$ is sufficiently small, indicating that the $(m+j)$th component of the eigenvector $q$ is small, then $\|g_j^{(1)}\|$ is allowed to be larger, and the required accuracy $\varepsilon_{out}$ can still be achieved. This induces a variable accuracy in the inner iteration, which drives the size of $\|g_j^{(1)}\|$. In the following we shall first show that the quantities $|e_j^* y|$ and $|e_j^* x|$ do tend to decrease as the approximation improves. We then derive a computable expression for the stopping tolerance in the approximation of $f(A)v$ and $f(A)^* u$ at each iteration of the resulting "relaxed" Lanczos bidiagonalization process. This strategy may be convenient in case the

cost of approximating $f(A)v$ and $f(A)^*u$ is very high, as is the case for instance if an accurate approximation to the leading singular triplets is requested.

### 3.6.1 *Spectral properties of the approximate singular triplets*

To ensure that the magnitude of $\|g_j^{(\ell)}\|$, $\ell = 1, 2$, can be relaxed in the bound (3.9), we need to verify that $|e_j^* x|$ and $|e_j^* y|$ become small as convergence takes place. This fact has been verified in the eigenvalue setting in [73]. However, the peculiar structure of the Lanczos bidiagonal recurrence requires the ad-hoc modifications of the results in [73]. To this end, we first define the submatrix of $\widetilde{\mathcal{B}}_{2m}$ of size $2k$ as

$$\widetilde{\mathcal{B}}_{2k} = \begin{bmatrix} 0 & M_k \\ T_k & 0 \end{bmatrix},$$

where $M_k$, $T_k$ are the leading portions of the corresponding $m \times m$ matrices. Let $(\theta^{(2k)}, q^{(2k)})$ be an eigenpair of $\widetilde{\mathcal{B}}_{2k}$, where $q^{(2k)} = [x; y]$ has unit norm, and $x, y \in \mathbb{C}^k$. Further, let

$$\widetilde{q} = \begin{bmatrix} x \\ 0 \\ y \\ 0 \end{bmatrix}, \tag{3.11}$$

where the 0-vectors have length $m - k$. Define $\mathcal{X} = [\widetilde{q}, Y]$, where $Y$ is chosen such that $\mathcal{X}$ is unitary, and at last we define $\underline{\widetilde{\mathcal{B}}}_{2m} = Y^* \widetilde{\mathcal{B}}_{2m} Y \in \mathbb{C}^{(2m-1)\times(2m-1)}$. The following proposition shows that under certain hypotheses some of the components of the approximate eigenvectors do tend to zero as convergence takes place.

**Proposition 3.6.1** *Let $(\theta^{(2k)}, q^{(2k)})$ be an eigenpair of $\widetilde{\mathcal{B}}_{2k}$, and $\widetilde{q}$ be as defined in* (3.11). *Let $s_{2m}^* = \widetilde{q}^* \widetilde{\mathcal{B}}_{2m} - \theta^{(2k)} \widetilde{q}^*$, $\delta_{2m,2k} = \sigma_{min}(\underline{\widetilde{\mathcal{B}}}_{2m} - \theta^{(2k)} I) > 0$, and $r_{2k} = t_{k+1,k} \begin{bmatrix} 0 \\ v_{k+1} \end{bmatrix} e_k^* q^{(2k)}$. If*

$$\|r_{2k}\| < \frac{\delta_{2m,2k}^2}{4\,\|s_{2m}\|}, \tag{3.12}$$

*then there exists a unit norm eigenvector* $q = [x_1; x_2; y_1; y_2]$ *of* $\widetilde{\mathcal{B}}_{2m}$ *with* $x_1, y_1 \in \mathbb{C}^k$, $x_2, y_2 \in \mathbb{C}^{m-k}$, *such that*

$$\left\| \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} \right\| \leq \frac{\tau}{\sqrt{1+\tau^2}},$$

*with* $\tau \in \mathbb{R}$, $0 \leq \tau < 2\frac{\|r_{2k}\|}{\delta_{2m,2k}}$. *Moreover, if* $\theta$ *is the eigenvalue associated with* $q$, *we have*

$$|\theta - \theta^{(2k)}| \leq \|s_{2m}\|\tau. \tag{3.13}$$

*Proof.* Define the submatrix of $\widetilde{\mathcal{B}}_{2m}$ of size $2k$ as

$$\widetilde{\mathcal{B}}_{2k} = \begin{bmatrix} 0 & M_k \\ T_k & 0 \end{bmatrix}, \qquad \text{i.e.} \qquad \widetilde{\mathcal{B}}_{2m} = \begin{bmatrix} 0 & 0 & M_k & M_\star \\ 0 & 0 & 0 & \star \\ T_k & T_\star & 0 & 0 \\ t_{k+1,k} e_1 e_k^* & \star & 0 & 0 \end{bmatrix}.$$

Define the vector $\tilde{q} = \frac{1}{\sqrt{2}}[x; 0; y; 0]$, where the 0-vectors have length $m-k$. Let $\mathcal{X} = [\tilde{q}, Y]$ be such that $\mathcal{X}$ is unitary, where $Y = [Y_1; Y_2; Y_3; Y_4]$. This implies that $\frac{1}{\sqrt{2}}(Y_1^* x + Y_3^* y) = 0$, $Y_4 Y_4^* = I = Y_2 Y_2^*$ and $Y_2 Y_4^* = Y_4 Y_2^* = 0$. Now, write

$$\mathcal{X}^* \widetilde{\mathcal{B}}_{2m} \mathcal{X} = \begin{bmatrix} \tilde{q}^* \widetilde{\mathcal{B}}_{2m} \tilde{q} & \tilde{q}^* \widetilde{\mathcal{B}}_{2m} Y \\ Y^* \widetilde{\mathcal{B}}_{2m} \tilde{q} & Y^* \widetilde{\mathcal{B}}_{2m} Y \end{bmatrix} = \begin{bmatrix} \theta^{(2k)} & g_1^* \\ g_2 & \underline{\widetilde{\mathcal{B}}}_{2m} \end{bmatrix}.$$

Here

$$\begin{aligned} \|g_2\| = \|Y^* \widetilde{\mathcal{B}}_{2m} \tilde{q}\| &= \|\frac{1}{\sqrt{2}} Y^* \begin{bmatrix} M_k y \\ 0 \\ T_k x \\ t_{k+1,k} e_1 e_k^* x \end{bmatrix} \| \\ &= \|\frac{1}{\sqrt{2}} Y_4^* t_{k+1,k} e_1 e_k^* x\| = \|r_{2k}\|. \end{aligned}$$

Further, since $s_{2m}^* Y = \tilde{q}^* \widetilde{\mathcal{B}}_{2m} Y - \theta^{(2k)} \tilde{q}^* Y = \tilde{q}^* \widetilde{\mathcal{B}}_{2m} Y$, we have

$$\|g_1\| = \|\tilde{q}^* \widetilde{\mathcal{B}}_{2m} Y\| = \|s_{2m}^* Y\| \leq \|s_{2m}\|.$$

Now, by [76, theorem 2.1, p.230],

$$\text{if } \frac{\|r_{2k}\|\,\|s_{2m}\|}{\delta_{2m,2k}^2} < \frac{1}{4}, \quad \text{i.e.,} \quad \|r_{2k}\| < \frac{\delta_{2m,2k}^2}{4\|s_{2m}\|},$$

then there exists a vector $p \in \mathcal{C}^{2m-1}$ satisfying $\tau = \|p\| < 2\frac{\|r_{2k}\|}{\delta_{2m,2k}}$, such that the unit norm vector

$$q = \begin{bmatrix} x_1 \\ x_2 \\ y_1 \\ y_2 \end{bmatrix} = \frac{1}{\sqrt{1+\|p\|^2}} \left( \frac{1}{\sqrt{2}} \begin{bmatrix} x \\ 0 \\ y \\ 0 \end{bmatrix} + \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \end{bmatrix} p \right)$$

is an eigenvector of $\widetilde{\mathcal{B}}_{2m}$. Moreover,

$$\left\| \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} \right\| = \left\| \frac{1}{\sqrt{1+\tau^2}} \begin{bmatrix} Y_2 \\ Y_4 \end{bmatrix} p \right\| \leq \frac{\tau}{\sqrt{1+\tau^2}}.$$

This same theorem also states that $|\theta - \theta^{(2k)}| = \|g_1^* p\| \leq \|g_1\|\|p\| \leq \|s_{2m}\|\tau$. □

This proposition states that if after $k \leq m$ iterations of Lanczos bidiagonalization the computed residual $\|r_{2k}\|$ is sufficiently small, then there exists an eigenvector of $\widetilde{\mathcal{B}}_{2m}$ such that some of its components are bounded correspondingly. These are precisely the (small) components that allow us to relax the accuracy of the inner iteration. Note that $\delta_{2m,2k}$ gives an indication of the distance between the spectrum of $\underline{\widetilde{\mathcal{B}}}_{2m}$ and $\theta^{(2k)}$. It should be kept in mind that for nonnormal matrices, the value of $\delta_{2m,2k}$ may be much smaller than the separation between the spectrum of $\underline{\widetilde{\mathcal{B}}}_{2m}$ and $\theta^{(2k)}$ [76, example 2.4, p. 234]. On the other hand, since $\widetilde{\mathcal{B}}_{2m}$ is a perturbation to a Hermitian matrix, the quantity $s_{2m}$ is an approximate residual for $(\theta^{(2k)}, q^{(2k)})$ as an eigenpair of $\widetilde{\mathcal{B}}_{2m}$, and thus it will be small as $m$ grows. As a consequence, condition (3.12) in the proposition is likely to be satisfied. Moreover, since $s_{2m}$ is going to be small, the eigenvalue error (3.13) may be much smaller than $\tau$. In our practical implementation we assumed that condition (3.12) is satisfied after the first two iterations.

### 3.6.2 *Variable accuracy in the inner approximation*

In this section we show that relaxation in the inner accuracy at step $k \leq m$ is possible if there exists an eigenpair $(\theta^{2(k-1)}, q^{2(k-1)})$ of $\widetilde{\mathcal{B}}_{2(k-1)}$ such that

$$\|r_{2(k-1)}\| < \frac{\delta^2_{2m,2(k-1)}}{4\|s_{2m}\|}, \tag{3.14}$$

$$\forall \theta_j \in \Lambda(\widetilde{\mathcal{B}}_{2m}), \theta_j \neq \theta, |\theta_j - \theta^{2(k-1)}| > 2\frac{\|s_{2m}\|\|r_{2(k-1)}\|}{\delta_{2m,2(k-1)}}. \tag{3.15}$$

The first condition (3.14) ensures that there exists an eigenvector $q$ of $\widetilde{\mathcal{B}}_{2m}$ whose specified components are small, according to proposition 3.6.1. Let $\theta$ be the eigenvalue associated with this $q$. The second condition, (3.15), guarantees that the eigenvalue $\theta^{2(k-1)}$ of $\widetilde{\mathcal{B}}_{2(k-1)}$ is a perturbation of the eigenvalue $\theta$ of $\widetilde{\mathcal{B}}_{2m}$, which is the final approximation to the original problem. In particular, the two conditions ensure that $\theta^{2(k-1)}$ is closer to $\theta$ than to all other eigenvalues $\theta_j$ of $\widetilde{\mathcal{B}}_{2m}$. It is also interesting to observe that if (3.14) holds, then (3.15) can be replaced by the stricter but possibly more insightful condition $|\theta_j - \theta^{2(k-1)}| > \delta_{2m,2(k-1)}/2$.

The following theorem shows that a variable accuracy will still guarantee a small residual gap; hence a true residual can be obtained whose norm is bounded by the accuracy of the gap, in agreement with (3.7).

**Theorem 3.6.2** *Assume $m$ inexact Lanczos bidiagonalization iterations are carried out. Let $(\theta, q)$ be an eigenpair of $\widetilde{\mathcal{B}}_{2m}$, where $\theta$ is simple and $\|q\| = 1$. Given $0 < \varepsilon_{out} \in \mathbb{R}$, with the notation of (3.9) assume that for $k = 1, \ldots, m$, and $i = 1, 2$,*

$$\|g_k^{(i)}\| \leq \begin{cases} \frac{\delta_{2m,2(k-1)}}{2m\|r_{2(k-1)}\|}\varepsilon_{out} & \text{if } k > 1, \text{ and there exists } (q^{2(k-1)}, \theta^{2(k-1)}) \text{ of } \widetilde{\mathcal{B}}_{2(k-1)} \text{ satisfying (3.14) and (3.15),} \\ \frac{1}{m}\varepsilon_{out} & \text{otherwise.} \end{cases} \tag{3.16}$$

*Then $\|(\mathcal{F}\mathcal{W}_{2m}q - \theta\mathcal{W}_{2m}q) - r_{2m}\| \leq \varepsilon_{out}$.*

*Proof.* Although the strategy used for the proof is similar to that of theorem 3.1 in [73], the block structure of our problem requires specialized

technical details. Suppose that at the $(k-1)$th iteration there exists an eigenpair $(\theta^{2(k-1)}, q^{2(k-1)})$ of $\widetilde{\mathcal{B}}_{2(k-1)}$ satisfying the conditions (3.14) and (3.15). This implies that $\theta^{2(k-1)}$ is a perturbation of the considered eigenvalue $\theta$ of $\widetilde{\mathcal{B}}_{2m}$, since $\theta$ is the only eigenvalue of $\widetilde{\mathcal{B}}_{2m}$ such that

$$|\theta - \theta^{2(k-1)}| \leq 2\frac{\|s_{2m}\|\|r_{2(k-1)}\|}{\delta_{2m,2(k-1)}}.$$

Let $\mathcal{K} \subset \{1,\ldots,m\}$ be defined such that for each $k \in \mathcal{K}$ there exists an eigenpair $(q^{2(k-1)}, \theta^{2(k-1)})$ of $\widetilde{\mathcal{B}}_{2(k-1)}$ satisfying the conditions (3.14) and (3.15). Then, similar to the reasoning in the proof of lemma 3.4.1 and using (3.9),

$$\begin{aligned}
\|(\mathcal{F}\mathcal{W}_{2m}q - \theta\mathcal{W}_{2m}q) - r_{2m}\| &= \|\mathcal{G}_{2m}q\| \\
&\leq \sum_{k=1}^{m}(\|g_k^{(1)}\|^2\,|e_k^*y|^2 + \|g_k^{(2)}\|^2\,|e_k^*x|^2)^{\frac{1}{2}} \\
&\leq \sum_{k\in\mathcal{K}}(\|g_k^{(1)}\|^2\,|e_k^*y|^2 + \|g_k^{(2)}\|^2|e_k^*x|^2)^{\frac{1}{2}} \\
&\qquad + \sum_{\substack{k\notin\mathcal{K},\\ k\leq m}}(\|g_k^{(1)}\|^2\,|e_k^*y|^2 + \|g_k^{(2)}\|^2|e_k^*x|^2)^{\frac{1}{2}} \\
&\leq \sum_{k\in\mathcal{K}}\frac{\delta_{2m,2(k-1)}\varepsilon_{out}}{2m\|r_{2(k-1)}\|}(|e_k^*y|^2 + |e_k^*x|^2)^{\frac{1}{2}} \\
&\qquad + \sum_{\substack{k\notin\mathcal{K},\\ k\leq m}}\frac{\varepsilon_{out}}{m}(|e_k^*y|^2 + |e_k^*x|^2)^{\frac{1}{2}} \\
&\leq \sum_{k\in\mathcal{K}}\frac{\delta_{2m,2(k-1)}\varepsilon_{out}}{2m\|r_{2(k-1)}\|}2\frac{\|r_{2(k-1)}\|}{\delta_{2m,2(k-1)}} + \sum_{\substack{k\notin\mathcal{K},\\ k\leq m}}\frac{\varepsilon_{out}}{m} \\
&= \frac{|\mathcal{K}|}{m}\,\varepsilon_{out} + \frac{m-|\mathcal{K}|}{m}\,\varepsilon_{out} = \varepsilon_{out}.
\end{aligned}$$

□

## 3.7 PRACTICAL IMPLEMENTATION

Algorithm 3.1 implements the inexact Lanczos bidiagonalization to approximate $\|f(A)\|_2$ and the associated singular vectors. As in the previous chapter the function $\mathrm{rgs}(z, Z)$ represents the reorthogonalization of the vector $z$ with respect to the orthogonal columns of $Z$, and returns

**Algorithm 3.1:** Inexact Lanczos bidiagonalization

**Input:** $A \in \mathbb{C}^{n \times n}$ non-Hermitian, a function $f$, a maximum number of (outer) iterations $m$, an (outer) tolerance $\varepsilon_{out}$.

**Output:** An approximation to the leading singular triplet.

1: Choose $v_1$ with $\|v_1\| = 1$, and set $V = [v_1]$, $U = []$, $M = []$, $T = []$.
2: **for** $j = 1, \ldots, m$
3: $z \approx f(A)v_j$
4: $u_j, m_j \longleftarrow$ `rgs`$(z, U)$
5: Expand basis: $U = [U, u_j]$
6: Expand matrix: $M = [M, m_j]$ (the old $M$ is padded with a zero row)
7: $z \approx f(A)^* u_j$
8: $v_{j+1}, t_j \longleftarrow$ `rgs`$(z, V)$
9: Expand basis: $V = [V, v_{j+1}]$
10: Expand matrix: $T = [T, t_j]$ (the old $T$ is padded with a zero row)
11: $K =$ `[zeros(j,j), M ; T(1:j,:) , zeros(j,j)]`
12: $[Q, D] \longleftarrow$ `eig`$(K)$
13: $(\theta, q) \longleftarrow$ with $\theta = \max_i |D_{ii}|$ (extract $x, y$ from $q = [x; y]$ with $\|x\| = 1, \|y\| = 1$)
14: Convergence check: if $|T(j+1, j)q(j)|/\theta < \varepsilon_{out}$ then return $(\theta, Ux, Vy)$ and stop
15: If required: compute variable tolerance to be used in the next iteration
16: **end**

the orthogonalization coefficients. The same algorithm can be used to approximate more singular triplets.

At every iteration of the Lanczos bidiagonalization, two inner iterations (line 3 and line 7) approximate the corresponding matrix-vector multiplication $f(A)v$ and $f(A)^*u$, respectively. The inner iteration uses one of the algorithms for approximating the action of a matrix function to a vector, as discussed in section 3.5. In theory, any such algorithm could be used; in our experiments we employed both the standard and extended Krylov subspace methods.

If the variant with variable inner tolerance is employed, the next inner tolerance is computed at the end of every Lanczos bidiagonalization iteration. To be conservative, during the first two iterations the inner tolerance is $\frac{1}{m}\varepsilon_{out}$, so that $\|g_k\| \leq \frac{1}{m}\varepsilon_{out}$. Then, in subsequent iter-

ations we assume that (3.14) and (3.15) are always satisfied, and thus we require that the inner stopping criterion is such that

$$\max\{\|g_k^{(1)}\|, \|g_k^{(2)}\|\} \leq \frac{\delta_{2m,2(k-1)}}{2m\|r_{2(k-1)}\|}\varepsilon_{out}.$$

Note that a relative criterion is always used, that is, in practice the quantity to be checked is divided by the current approximation $\theta^{2(k-1)}$. This corresponds to using $\varepsilon_{out}^{(k)} = \theta^{2(k-1)}\varepsilon_{out}$ for some fixed value $\varepsilon_{out}$. Since $\delta_{2m,2(k-1)}$ is not available at iteration $k$, we consider the following approximation:

$$\delta^{2(k-1)} := \min_{\theta_j \in \Lambda(\widetilde{\mathcal{B}}_{2(k-1)}) \setminus \{\theta^{2(k-1)}\}} |\theta^{2(k-1)} - \theta_j|.$$

In fact, $\delta_{2m,2(k-1)}$ can be much smaller than the computed $\delta^{2(k-1)}$. However, it will not be overrated much when $\theta^{2(k-1)}$ is converging to the corresponding eigenvalue $\theta$ of $\widetilde{\mathcal{B}}_{2m}$, since it is related to the sensitivity of $\widetilde{\mathcal{B}}_{2m}$ and not of the matrix $\mathcal{F}$. If the $\delta^{2(k-1)}$ is very small, it constrains the inner accuracy to be very small too. This occurs when the largest eigenvalues of $\widetilde{\mathcal{B}}_{2m}$ are clustered. We refer to section 3.8.4 for a numerical illustration. We also remark that the computation of $\delta^{2(k-1)}$ does not significantly increase the computational costs, as all the eigenvalues of $\widetilde{\mathcal{B}}_{2(k-1)}$ are already required to obtain the current approximation.

For the approximation of more than one singular triplet some extra care is needed. Step 13 of algorithm 3.1 could be generalized to the selection of the first $\ell$, say, leading eigenvectors. Since these eigenvectors are not orthogonal in general, for stability reasons we propose to work with the partial Schur form of the matrix $\widetilde{\mathcal{B}}_{2m}$, keeping in mind that the same procedure can be applied to eigenvectors. Let

$$\widetilde{\mathcal{B}}_{2m} \begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} X \\ Y \end{bmatrix} L,$$

be the partial Schur decomposition of $\widetilde{\mathcal{B}}_{2m}$, such that the decomposition only contains the portion corresponding to the leading $\ell$ eigenvalues in $L$ with positive real part. Note that the columns of $[X; Y]$ are orthogonal, $L$ is upper triangular and $L$ shares eigenvalues with $\widetilde{\mathcal{B}}_{2m}$. If the data are real then we can use the real Schur decomposition with $L$ quasi-block

triangular. We incorporate this decomposition into the inexact Lanczos bidiagonalization, using the notation of (3.5), and obtain

$$\begin{aligned}(f(A)+\mathfrak{C}_m)V_mY &= U_mXL,\\ (f(A)^*+\mathfrak{D}_m)U_mX &= V_mYL + t_{m+1,m}v_{m+1}e_m^*X.\end{aligned}$$

The norm of the quantity $t_{m+1,m}v_{m+1}e_m^*X$ can be used in a stopping criterion as in the single vector case. Define now $\mathcal{V} = \operatorname{span}(V_mY)$ and $\mathcal{U} = \operatorname{span}(U_mX)$. In the above equations two (approximate) invariant singular subspaces can be recognized, $(f(A)+\mathfrak{C}_m)\mathcal{V} \subset \mathcal{U}$ and approximately $(f(A)^*+\mathfrak{D}_m)\mathcal{U} \subset \mathcal{V}$.

To provide the user with $\ell$ singular triplets, we first orthogonalize the columns of $Y$ and $X$ by means of the *reduced* QR decomposition, that is $Y = Q_YR_Y$ and $X = Q_XR_X$. Let also $U_L\Sigma_LV_L^* = R_XLR_Y^{-1}$ be the singular value decomposition of the right-hand side matrix. Then we can write

$$(f(A)+\mathfrak{C}_m)V_mQ_Y = U_mQ_X(R_XLR_Y^{-1}) = U_mQ_XU_L\Sigma_LV_L^*,$$

so that $(f(A)+\mathfrak{C}_m)(V_mQ_YV_L) = (U_mQ_XU_L)\Sigma_L$. We may thus conclude that the columns of $(V_mQ_YV_L)$, $(U_mQ_XU_L)$ and the diagonal elements of $\Sigma_L$ yield the desired triplets.

## 3.8 NUMERICAL EXPERIMENTS

In this section we report on our numerical experiments to evaluate the performance of the inexact Lanczos bidiagonalization for different combinations of matrices and functions. All experiments in this section were performed with MATLAB Version 7.13.0.564 (R2011b) on a Dell Latitude laptop running Ubuntu 14.04 with 4 CPUs at 2.10GHz. We are mainly interested in the first singular triplet of $f(A)$, so as to obtain $\|f(A)\|$. We considered five different matrices, summarized in table 3.1, all of dimension $n = 10,000$ except $A_4$. The spectrum of a sample of these matrices of smaller size, $n = 1000$, is reported in figure 3.1 (matrix $A_4$ is omitted having a (too large) fixed size). For $A_4$, the matrix originating from modeling a 2D fluid flow in a driven cavity, using the incompressible Navier Stokes equations, was shifted by $10I$, to ensure that its field of values is in $\mathbb{C}^+$ so that projection methods for $f(A)\mathbf{v}$ are applicable for all considered functions. We refer to the Matrix Market site for more information on this problem [80]. For $A_5$ a 5-point stencil

finite difference approximation was used, together with homogeneous Dirichlet boundary conditions. Note that only the spectrum of the matrix $A_1$ is non-symmetric with respect to the real axis. We considered the following functions for $x$,

$$\exp(x), \quad \exp(-x), \quad \sqrt{x}, \quad \frac{1}{\sqrt{x}}, \quad \frac{\exp(-\sqrt{x})-1}{x}.$$

We note that all these functions allow for an efficient computation when applied to small scale matrices, by means of specifically designed MATLAB functions; see [39]. The performance also critically depends on the choice of the inner method for approximating $f(A)v$ and $f(A)^*u$ at each iteration. We shall report our experience with the standard and extended Krylov methods. The Rational Krylov method could also be employed for this approximation.

| Matrix | Structure | Description |
|---|---|---|
| $A_1$ | tridiag$(0, \underline{\lambda_i}, 0.3)$ | $\lambda_i = (1+\rho_i^{(1)}) + i(\rho_i^{(2)} - 0.5)$ |
| $A_2$ | tridiag$(1.5, \underline{2}, -1)$ | |
| $A_3$ | Toeplitz | $i$-th row: $[4,0,0,0,0,-2,0,\underline{10},0,0,0,6]$ |
| $A_4$ | shifted E20R1000 | Driven cavity problem (Matrix Market) shifted as $A := A_{e20r1000} + 10I$ |
| $A_5$ | Sparse | Centered Finite Difference discretization of $\mathcal{L}(u) = -\nabla^2 u - 100u_x - 100u_y$ |

Table 3.1: Description of the selected matrices, all of size $n = 10,000$, except $A_4$, of size 4241. $\rho_i^{(j)}$ is a random entry taken from a uniform distribution in $(0,1)$.

A random vector is used to start the inexact Lanczos bidiagonalization process. Convergence is monitored by checking the computed residual norm with respect to the first singular triplet, and the inexact Lanczos bidiagonalization terminates as soon as (3.8) is satisfied; different values for $\varepsilon_{out}$ will be considered. In case more than one triplet is desired, inexactness can be tuned by using the norm of the *matrix* residual, as described in more detail in section 3.7.

In section 3.8.1 we explore the fixed inner tolerance method, and the dependence of its performance on all the other parameters, including the outer accuracy. Indeed, if only a rough approximation of $\|f(A)\|$ is

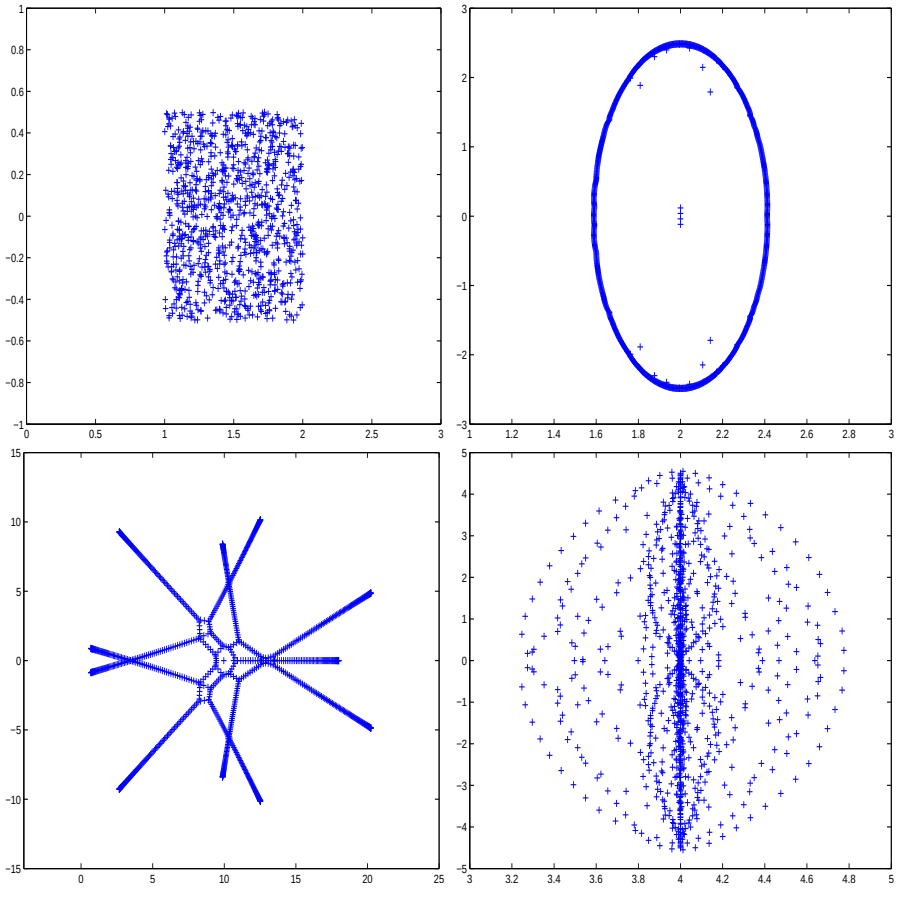


Figure 3.1: Spectrum of matrices $A_1, A_2, A_3, A_5$ (from left to right) in table 3.1, for a smaller size, $n = 1000$.

required, the computational efforts should be proportionally low. We subsequently compare our method to the power method, and present results for an implementation based on the extended Krylov subspace. In section 3.8.4 the influence of the variable (relaxed) inner tolerance described in section 3.6.2 is analyzed, thus a more stringent final accuracy is considered so as to exercise the variable inner threshold.

### 3.8.1 *Assessing the effectiveness of the inexact bidiagonalization*

We analyze the performance of the inexact method when approximating $\|f(A_i)\|$ together with the associated singular vectors. To this end,

| matr | function $f$ | $\tilde{\sigma}_1$ | $\frac{\tilde{\sigma}_1-\tilde{\sigma}_2}{\tilde{\sigma}_1}$ | tot # outer | tot # inner | average # inner | exec time |
|---|---|---|---|---|---|---|---|
| $A_1$ | $\exp(-x)$ | 0.463506 | 3.34e-02 | 14 | 308 | 11.0 | 0.39 |
| | $\sqrt{x}$ | 1.50143 | 1.29e-02 | 15 | 351 | 11.7 | 0.44 |
| | $\frac{\exp(-\sqrt{x})-1}{x}$ | 0.727351 | 2.11e-02 | 14 | 414 | 14.8 | 0.70 |
| | $\exp(x)$ | 9.19137 | 1.76e-02 | 16 | 352 | 11.0 | 0.59 |
| | $1/\sqrt{x}$ | 1.10350 | 2.33e-02 | 13 | 364 | 14.0 | 0.73 |
| $A_2$ | $\exp(-x)$ | 0.222621 | 1.78e-02 | 11 | 308 | 14.0 | 0.24 |
| | $\sqrt{x}$ | 1.7921 | 2.05e-02 | 8 | 252 | 15.8 | 0.28 |
| | $\frac{\exp(-\sqrt{x})-1}{x}$ | 0.469829 | 1.81e-02 | 10 | 429 | 21.4 | 0.68 |
| | $\exp(x)$ | 12.1783 | 9.01e-02 | 5 | 139 | 13.9 | 0.14 |
| | $1/\sqrt{x}$ | 0.814356 | 2.26e-02 | 8 | 324 | 20.2 | 0.44 |
| $A_3$ | $\exp(-x)$ | 0.508086 | 1.48e-02 | 13 | 709 | 27.3 | 0.61 |
| | $\sqrt{x}$ | 4.56086 | 1.85e-02 | 12 | 1036 | 43.2 | 3.11 |
| | $\frac{\exp(-\sqrt{x})-1}{x}$ | 0.615673 | 1.84e-02 | 12 | 1953 | 81.4 | 18.29 |
| | $\exp(x)$ | 6.75709e8 | 1.45e-02 | 13 | 694 | 26.7 | 0.83 |
| | $1/\sqrt{x}$ | 0.959018 | 1.70e-02 | 12 | 1852 | 77.2 | 14.56 |
| $A_4$ | $\exp(-x)$ | 0.000172183 | 2.42e-01 | 6 | 456 | 38.0 | 0.58 |
| | $\sqrt{x}$ | 6.09177 | 2.78e-02 | 14 | 454 | 16.2 | 0.81 |
| | $\frac{\exp(-\sqrt{x})-1}{x}$ | 0.118301 | 3.07e-02 | 10 | 486 | 24.3 | 1.15 |
| | $\exp(x)$ | 3.15141e10 | 1.35e-01 | 5 | 397 | 39.7 | 0.49 |
| | $1/\sqrt{x}$ | 0.354039 | 5.55e-02 | 9 | 394 | 21.9 | 0.81 |
| $A_5$ | $\exp(-x)$ | 0.99709 | 3.39e-02 | 7 | 223 | 15.9 | 0.34 |
| | $\sqrt{x}$ | 2.81987 | 1.16e-02 | 16 | 5145 | 160.8 | 193.35 |
| | $\frac{\exp(-\sqrt{x})-1}{x}$ | 6.93384 | 2.44e-01 | 4 | 1570 | 196.2 | 111.67 |
| | $\exp(x)$ | 2959.17 | 1.84e-02 | 14 | 450 | 16.1 | 0.65 |
| | $1/\sqrt{x}$ | 7.36692 | 2.31e-01 | 4 | 1564 | 195.5 | 112.22 |

Table 3.2: Inexact Lanczos bidiagonalization for approximating the leading singular triplet of $f(A)$; outer tolerance $\varepsilon = 10^{-2}$ .

we need to monitor the number of iterations of both the outer and the two inner iterations, together with the execution time required to reach the required tolerance. In particular, we show both the total and average number of inner iterations. We also display the distance between the final first approximate singular value, $\tilde{\sigma}_1$ and the second approximate singular value, $\tilde{\sigma}_2$: a small relative distance implies that the method will take more iterations to converge. Moreover, this dis-

tance cannot be easily predicted from the matrix $A$, although it significantly influences the computation. For instance, the largest (in modulus) eigenvalues of $\mathcal{F}$ associated with the matrix function $A_2^{\frac{1}{2}}$ are:

$$-1.7965100, 1.7965100, 1.7964169, -1.7964169, 1.7962429, -1.7962424.$$

Although this fact does not constitute a difficulty if just the order of magnitude of $\|A_2^{\frac{1}{2}}\|$ is sought, it indicates that requiring a more accurate approximation will lead to significantly more expensive computations. This problem can be readily observed by comparing the outer number of iterations in table 3.2 and table 3.3, where we report the results of our experiments for $\varepsilon_{out} = 10^{-2}$ and $\varepsilon_{out} = 10^{-4}$, respectively. In both cases, the inner tolerance was set to $\varepsilon_{in} = \varepsilon_{out}/(m_{\max})$, where $m_{\max} = 1000$, so that $\varepsilon_{in} = 10^{-7}$ for the more stringent outer tolerance. For all examples, the first six significant digits of $\tilde{\sigma}_1$ are reported.

Comparing the two tables also shows that the singular values are as accurate as the outer tolerance can predict: for smaller $\varepsilon_{out}$ already the third singular value digit changes, that is it still has to reach its final (exact) value. This is obviously also related to the relative distance from the second singular value, which is better captured for a smaller $\varepsilon_{out}$.

We also observe that the choice of $f$ strongly influences the overall performance: the bidiagonalization process may take the same number of (outer) iterations for two different selections of $f$, and yet the total computational cost may be significantly different (see $A_1$ and $A_3$ in table 3.2). As a consequence, the number of outer iterations is not a realistic measure of the algorithm complexity.

On a negative side, we observe that in both tables the method performs poorly on $A_5$ for $f(x) = \sqrt{x}$. For this particular matrix, the inner method takes very many iterations during the whole Lanczos bidiagonalization process, with a number of inner iterations that is close to the average throughout. We anticipate that this is not the case for the power method, where as the outer iterations proceed, drastically fewer iterations are required in the inner approximation. This phenomenon seems to be specific for this combination of function and matrix, since in all other cases the performance of the Lanczos bidiagonalization and power method is more similar. It may be further investigated in a future study.

Finally, for the exponential functions $\exp(x), \exp(-x)$ we computed the upper bound in (3.1) by using the MATLAB function `eigs` applied

| matr | function $f$ | $\tilde{\sigma}_1$ | $\frac{\tilde{\sigma}_1-\tilde{\sigma}_2}{\tilde{\sigma}_1}$ | tot # outer | tot # inner | average # inner | exec time |
|---|---|---|---|---|---|---|---|
| $A_1$ | $\exp(-x)$ | 0.463735 | 2.04e-02 | 24 | 624 | 13.0 | 0.87 |
| | $\sqrt{x}$ | 1.50496 | 8.76e-04 | 53 | 1775 | 16.7 | 3.27 |
| | $\frac{\exp(-\sqrt{x})-1}{x}$ | 0.728200 | 7.62e-03 | 29 | 1160 | 20.0 | 2.47 |
| | $\exp(x)$ | 9.19576 | 9.22e-03 | 32 | 832 | 13.0 | 1.21 |
| | $\frac{1}{\sqrt{x}}$ | 1.10504 | 5.52e-03 | 29 | 1156 | 19.9 | 2.21 |
| $A_2$ | $\exp(-x)$ | 0.223129 | 4.88e-05 | 209 | 7104 | 17.0 | 26.72 |
| | $\sqrt{x}$ | 1.79651 | 5.18e-05 | 162 | 8069 | 24.9 | 18.92 |
| | $\frac{\exp(-\sqrt{x})-1}{x}$ | 0.470776 | 3.85e-05 | 193 | 12320 | 31.9 | 42.03 |
| | $\exp(x)$ | 12.1825 | 8.39e-04 | 47 | 1596 | 17.0 | 1.41 |
| | $\frac{1}{\sqrt{x}}$ | 0.816492 | 5.90e-05 | 150 | 9210 | 30.7 | 29.43 |
| $A_3$ | $\exp(-x)$ | 0.509010 | 4.43e-05 | 224 | 14544 | 32.5 | 31.55 |
| | $\sqrt{x}$ | 4.57175 | 3.35e-05 | 250 | 41844 | 83.7 | 533.17 |
| | $\frac{\exp(-\sqrt{x})-1}{x}$ | 0.616989 | 1.20e-04 | 155 | 39968 | 128.9 | 1078.46 |
| | $\exp(x)$ | 6.77296e8 | 1.17e-04 | 183 | 11660 | 31.9 | 19.91 |
| | $\frac{1}{\sqrt{x}}$ | 0.960790 | 2.12e-05 | 312 | 77958 | 124.9 | 1827.25 |
| $A_4$ | $\exp(-x)$ | 0.000172195 | 2.32e-01 | 9 | 783 | 43.5 | 1.26 |
| | $\sqrt{x}$ | 6.09289 | 2.38e-02 | 22 | 1103 | 25.1 | 2.25 |
| | $\frac{\exp(-\sqrt{x})-1}{x}$ | 0.118347 | 2.44e-02 | 16 | 1109 | 34.7 | 3.16 |
| | $\exp(x)$ | 3.15148e10 | 1.34e-01 | 7 | 626 | 44.7 | 0.87 |
| | $\frac{1}{\sqrt{x}}$ | 0.354473 | 1.18e-02 | 19 | 1227 | 32.3 | 3.91 |
| $A_5$ | $\exp(-x)$ | 0.998062 | 7.74e-03 | 24 | 911 | 19.0 | 1.37 |
| | $\sqrt{x}$ | 2.82811 | 1.67e-04 | 185 | 70926 | 191.7 | 4059.40 |
| | $\frac{\exp(-\sqrt{x})-1}{x}$ | 6.93435 | 2.32e-01 | 7 | 2814 | 201.0 | 186.39 |
| | $\exp(x)$ | 2975.18 | 2.91e-03 | 55 | 2091 | 19.0 | 3.20 |
| | $\frac{1}{\sqrt{x}}$ | 7.36768 | 2.17e-01 | 7 | 2814 | 201.0 | 197.62 |

Table 3.3: Inexact Lanczos bidiagonalization for approximating the leading singular triplet of $f(A)$; outer tolerance $\varepsilon = 10^{-4}$ .

to $\frac{1}{2}(A + A^*)$. In all cases except for matrix $A_4$ the estimate is pretty sharp. On the other hand, for $\exp(A_4)$ the upper bound is $2 \cdot 10^{13}$, which is three orders of magnitude larger than the actual norm; for $\exp(-A_4)$ the upper bound[2] is 0.004737, which is more than one order of magnitude larger than the actual value, 0.000172. This example

2 This bound is obtained for $\exp(\hat{A})$ with $\hat{A} = -A_4$.

illustrates that, as discussed in section 3.2, the accuracy of this type of estimate cannot be easily monitored, especially in the case of nonnormal matrices.

We also experimented with the approximation of more than one triplet. We refer to the end of section 3.7 for a description on how to compute various singular triplets simultaneously and what stopping criterion to use. Our findings for $A_1$ and $f(x) = 1/\sqrt{x}$ are reported, where we use $\varepsilon_{out} = 10^{-6}$ and $m_{\max} = 100$. Table 3.4 shows the largest six singular values obtained with a fixed inner tolerance of $10^{-8}$ ($\tilde{\sigma}_j$, second column). As an a-posteriori test, we report in the last column the norm of the residual, i.e., $\|f(A)^*\tilde{u}_j - \tilde{\sigma}_j\tilde{v}_j\|$, for the approximated singular triplets ($\tilde{\sigma}_j$, $\tilde{u}_j$, $\tilde{v}_j$), where $j = 1, \ldots, 6$, and $f(A)^*\tilde{u}_j$ is computed with an accuracy of $10^{-11}$. The iteration of the inexact Lanczos bidiagonalization is stopped as soon as the outer stopping criterion is satisfied for the norm of the matrix residual based on the largest six singular values. As expected, the last singular triplet dominates the convergence process: the norm of the matrix residual becomes sufficiently small only when this last triplet has a residual norm of the order of $\varepsilon_{out}$.

| $j$ | $\tilde{\sigma}_j$ | $\|f(A)^*\tilde{u}_j - \tilde{\sigma}_j\tilde{v}_j\|$ |
|---|---|---|
| 1 | 1.117020718020110 | 4.1600e-11 |
| 2 | 1.107884805364245 | 8.4447e-11 |
| 3 | 1.098396522938209 | 7.9652e-11 |
| 4 | 1.098383564116149 | 4.5703e-11 |
| 5 | 1.088066190375196 | 4.0340e-08 |
| 6 | 1.086710736722748 | 8.2518e-07 |

Table 3.4: First six approximate singular values of $A_1^{-1/2}$ with fixed tolerance ($\varepsilon_{out} = 10^{-6}$).

### 3.8.2 *Comparisons with the power method*

We wish to compare the performance of the new method with that of the power method, as described in section 3.2. Since in most cases, the leading singular values are not well isolated, we expect that the power method will be slow if an accurate approximation is required. Therefore, we only report results for $\varepsilon = 10^{-2}$. Moreover, our experience is

that since the computation is inexact, the product $f(A)^*(f(A)v)$ may give complex values, since the computed actions of $f(A)$ and $f(A)^*$ are not the conjugate of each other. As a result, the approximate eigenvalue may be complex, though with a small imaginary part, and the quantity that is actually computed is given by

$$\lambda^{(k)} = \left| \frac{(v^{(k)})^* f(A)^* f(A) v^{(k)}}{(v^{(k)})^* v^{(k)}} \right|,$$

where $v^{(k)}$ is the power method direction after $k$ iterations. Consequently, at convergence we obtain $\tilde{\sigma}_1 \approx \sqrt{\lambda^{(k)}}$. The stopping criterion is based on the relative eigenvalue residual norm, that is

$$\|y^{(k)} - \lambda^{(k)} v^{(k)}\| / \lambda^{(k)} \leq \varepsilon_{out},$$

where $y^{(k)}$ is the result of the approximation of $f(A)^*(f(A)v^{(k)})$. Note that we kept the same tolerance as for the Lanczos bidiagonalization, although a more stringent tolerance may be required in practice. Table 3.5 collects the results for all test cases.

As expected, the power method is more expensive than the Lanczos bidiagonalization, on average four to five times more expensive, in all those cases when the first singular value is not well separated from the second one. Only for the cases of good separation, for instance with $A_5$ and the functions $(\exp(\sqrt{x}) - 1)/x$ and $1/\sqrt{x}$, convergence is reached in very few iterations, and the power method is competitive.

We also implemented the power method as described in [39, algorithm 3.19], using the relative singular value residual as stopping criterion. The performance, both in terms of inner and outer number of iterations, is comparable to that in table 3.5. Finally, we stress that in both implementations the stopping criterion involves inexact matrix-vector products. Therefore, the monitored quantity is not the true residual of the corresponding problem.

### 3.8.3 *Numerical tests with the extended Krylov subspace*

If for the final approximation a high accuracy is required, so that a more stringent outer tolerance is used, then the inner iteration also requires more computational effort, as its stopping tolerance is also decreased. In this case, it may be appropriate to use more effective methods. One

| matr | function | tot # outer | tot # inner | $\tilde{\sigma}_1$ | residual norm | exec time |
|---|---|---|---|---|---|---|
| $A_1$ | $\exp(-x)$ | 51 | 1071 | 0.46327 | 9.8648e-03 | 1.5 |
| | $\sqrt{x}$ | 93 | 2010 | 1.5028 | 9.8607e-03 | 2.8 |
| | $\frac{\exp(-\sqrt{x})-1}{x}$ | 61 | 1782 | 0.71879 | 9.8904e-03 | 3.5 |
| | $\exp(x)$ | 65 | 1409 | 9.1666 | 9.9964e-03 | 2.0 |
| | $1/\sqrt{x}$ | 69 | 1942 | 1.0938 | 9.8670e-03 | 3.3 |
| $A_2$ | $\exp(-x)$ | 36 | 899 | 0.22238 | 9.8636e-03 | 0.8 |
| | $\sqrt{x}$ | 37 | 979 | 1.7903 | 9.7995e-03 | 1.1 |
| | $\frac{\exp(-\sqrt{x})-1}{x}$ | 36 | 1216 | 0.46921 | 9.7553e-03 | 2.1 |
| | $\exp(x)$ | 9 | 232 | 12.176 | 9.4623e-03 | 0.2 |
| | $1/\sqrt{x}$ | 36 | 1215 | 0.81375 | 9.8715e-03 | 1.8 |
| $A_3$ | $\exp(-x)$ | 38 | 1605 | 0.50724 | 9.9024e-03 | 1.5 |
| | $\sqrt{x}$ | 41 | 1000 | 4.5564 | 9.8934e-03 | 1.3 |
| | $\frac{\exp(-\sqrt{x})-1}{x}$ | 34 | 4448 | 0.61486 | 9.9901e-03 | 39.1 |
| | $\exp(x)$ | 38 | 1699 | 6.7455e8 | 9.7718e-03 | 1.7 |
| | $1/\sqrt{x}$ | 36 | 4684 | 0.95774 | 9.7264e-03 | 36.3 |
| $A_4$ | $\exp(-x)$ | 11 | 825 | 0.00017219 | 5.8988e-03 | 1.0 |
| | $\sqrt{x}$ | 56 | 1710 | 6.0870 | 9.9037e-03 | 3.0 |
| | $\frac{\exp(-\sqrt{x})-1}{x}$ | 28 | 1309 | 0.11823 | 9.5839e-03 | 3.3 |
| | $\exp(x)$ | 10 | 775 | 3.1510e10 | 8.8031e-03 | 1.1 |
| | $1/\sqrt{x}$ | 33 | 1361 | 0.35405 | 9.9455e-03 | 2.8 |
| $A_5$ | $\exp(-x)$ | 15 | 376 | 0.99643 | 9.9168e-03 | 0.52 |
| | $\sqrt{x}$ | 52 | 1479 | 2.81329 | 9.9649e-03 | 18.47 |
| | $\frac{\exp(-\sqrt{x})-1}{x}$ | 7 | 2745 | 6.93355 | 9.6566e-03 | 189.19 |
| | $\exp(x)$ | 55 | 1363 | 2956.34 | 9.8499e-03 | 1.79 |
| | $1/\sqrt{x}$ | 8 | 3137 | 7.36721 | 6.9885e-03 | 202.71 |

Table 3.5: Power method for approximating the leading singular triplet of $f(A)$; outer tolerance $\varepsilon = 10^{-2}$.

such possibility is the extended Krylov subspace method (EKSM) [18, 56], which may be convenient in case the considered function requires a good approximation of the eigenvalues of $A$ closest to the origin. In table 3.6 we report on the computations for $\varepsilon_{out} = 10^{-4}$ when EKSM is used. These numbers should be compared with those in table 3.3. We notice that EKSM requires the solution of a system with $A$ (or $A^*$) at each iteration; to limit computational costs, a sparse LU factorization

of $A$ was performed and stored once and for all at the beginning of the Lanczos bidiagonalization, and used repeatedly in the inner iteration. This represents a tremendous saving with respect to more general rational approximations, where solves with $(A - \tau_j I)$ have to be performed at each inner iteration, with $\tau_j$ varying with the inner step.

In table 3.6 all cases where EKSM provides faster convergence, that is less computing time, are marked in boldface. It is clear that EKSM is beneficial when good approximations to both ends of the spectrum are required, as is the case for $x^\alpha$. The lack of improvement in the case of the exponential is expected, as it is known, at least in the Hermitian case, that only one extreme of the spectrum needs to be captured for a fast approximation of $\exp(A)v$.

We also remark that EKSM could also be employed as inner method in the case of the power iteration used in section 3.8.2.

### 3.8.4 *Numerical tests with variable accuracy*

In the previous sections, for $\varepsilon_{out} = 10^{-4}$ the inner tolerance was set to the fixed value $\varepsilon_{in} = 10^{-7}$. Here we explore the performance of the inexact computation when the inner tolerance is relaxed.

A relaxed inner accuracy is most convenient when the inner iteration is expensive, so as to profit from a lower number of inner iterations. Therefore, we report on our experience with the extended Krylov subspace as inner method, as the method requires one system solve with the coefficient matrix at each iteration. A more stringent outer tolerance is used, that is $\varepsilon_{out} = 10^{-7}$, than in previous experiments, to clearly see the relaxation in the inner tolerance; we also use $m_{\max} = 50$ as maximum number of iterations to balance the much smaller $\varepsilon_{out}$ for determining the initial inner tolerance.

Figure 3.2 shows the performance of the relaxation strategy for $A_5$ and $f(x) = 1/\sqrt{x}$. The plot shows the outer convergence history as the bidiagonalization proceeds, and the corresponding variable inner tolerance. The digits next to each iteration report the actual numbers of inner iterations by means of EKSM to reach the required inner accuracy for approximating $f(A)v_j$; similar numbers are observed for $f(A)^*u_j$.

Table 3.7 reports the values of $\delta_{2m,2(k-1)}$ and $\delta^{2(k-1)}$ during the iterations displayed in figure 3.2; see the discussion on these parameters at the end of section 3.7. For this specific example, the values of $\delta^{2(k-1)}$

| matr | function | $\tilde{\sigma}_1$ | $\frac{\tilde{\sigma}_1-\tilde{\sigma}_2}{\tilde{\sigma}_1}$ | tot # outer | tot # inner | average # inner | exec time |
|---|---|---|---|---|---|---|---|
| $A_1$ | $\exp(-x)$ | 0.463735 | 2.04e-02 | 24 | 480 | 10.0 | 3.49 |
| | $\sqrt{x}$ | 1.50496 | 8.76e-04 | 53 | 954 | 9.0 | 8.24 |
| | $\frac{\exp(-\sqrt{x})-1}{x}$ | 0.728200 | 7.62e-03 | 29 | 522 | 9.0 | 4.29 |
| | $\exp(x)$ | 9.19576 | 9.22e-03 | 32 | 704 | 11.0 | 5.82 |
| | $\frac{1}{\sqrt{x}}$ | 1.10504 | 5.52e-03 | 29 | 522 | 9.0 | 4.59 |
| $A_2$ | $\exp(-x)$ | 0.223129 | 4.88e-05 | 209 | 5434 | 13.0 | 36.71 |
| | $\sqrt{x}$ | 1.79651 | 5.18e-05 | 162 | 3564 | 11.0 | 20.03 |
| | $\frac{\exp(-\sqrt{x})-1}{x}$ | 0.470776 | 3.85e-05 | 193 | 4246 | 11.0 | **29.99** |
| | $\exp(x)$ | 12.1825 | 8.39e-04 | 47 | 1408 | 15.0 | 5.07 |
| | $\frac{1}{\sqrt{x}}$ | 0.816492 | 5.90e-05 | 150 | 3300 | 11.0 | **18.70** |
| $A_3$ | $\exp(-x)$ | 0.509010 | 4.43e-05 | 224 | 11827 | 26.4 | 106.31 |
| | $\sqrt{x}$ | 4.57175 | 3.35e-05 | 250 | 9402 | 18.8 | **84.26** |
| | $\frac{\exp(-\sqrt{x})-1}{x}$ | 0.616989 | 1.20e-04 | 155 | 5578 | 18.0 | **40.02** |
| | $\exp(x)$ | 6.7729e8 | 1.17e-04 | 183 | 11169 | 33 | 112.76 |
| | $\frac{1}{\sqrt{x}}$ | 0.960790 | 2.12e-05 | 312 | 11449 | 18.3 | **125.20** |
| $A_4$ | $\exp(-x)$ | 0.000172195 | 2.32e-01 | 9 | 376 | 20.9 | 4.20 |
| | $\sqrt{x}$ | 6.09289 | 2.38e-02 | 22 | 483 | 11.0 | 5.99 |
| | $\frac{\exp(-\sqrt{x})-1}{x}$ | 0.118347 | 2.44e-02 | 16 | 318 | 9.9 | 4.32 |
| | $\exp(x)$ | 3.15148e10 | 1.34e-01 | 7 | 527 | 37.6 | 4.66 |
| | $\frac{1}{\sqrt{x}}$ | 0.354473 | 1.18e-02 | 19 | 416 | 10.9 | 4.88 |
| $A_5$ | $\exp(-x)$ | 0.998062 | 7.74e-03 | 24 | 887 | 18.5 | 11.95 |
| | $\sqrt{x}$ | 2.82811 | 1.67e-04 | 185 | 8165 | 22.1 | **121.99** |
| | $\frac{\exp(-\sqrt{x})-1}{x}$ | 6.93435 | 2.32e-01 | 7 | 294 | 21.0 | **5.01** |
| | $\exp(x)$ | 2975.18 | 2.91e-03 | 55 | 2090 | 19.0 | 25.19 |
| | $\frac{1}{\sqrt{x}}$ | 7.36768 | 2.17e-01 | 7 | 294 | 21.0 | **4.32** |

Table 3.6: Inexact Lanczos bidiagonalization, outer tolerance $\varepsilon = 10^{-4}$, inner approximation: extended Krylov subspace method.

are a good estimate for the actual $\delta_{2m,2(k-1)}$ even at an early stage of the iteration (we recall here that no relaxed strategy is used in the first two iterations).

We also experimented with the approximation of more than one triplet. We report on our findings for $A_1$ and again $f(x) = 1/\sqrt{x}$ (similar accuracies were obtained for other functions for the same ma-

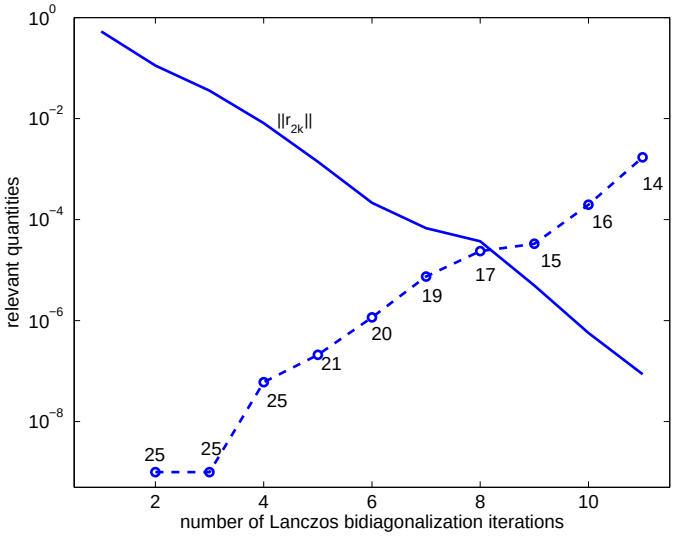



Figure 3.2: Relaxed inner iteration for variable stopping tolerance, to approximate $\|A_5^{-1/2}\|$, with $\varepsilon_{out} = 10^{-7}$.

| $k-1$ | $\delta^{2(k-1)}$ | $\delta_{2m,2(k-1)}$ |
|---|---|---|
| 1 | - | 2.3147e-01 |
| 2 | - | 7.7043e-01 |
| 3 | 2.1738e+00 | 9.5073e-01 |
| 4 | 1.7049e+00 | 9.6671e-01 |
| 5 | 1.6151e+00 | 9.6744e-01 |
| 6 | 1.6030e+00 | 9.6746e-01 |
| 7 | 1.6017e+00 | 9.6746e-01 |
| 8 | 1.3696e+00 | 9.6746e-01 |
| 9 | 9.7931e-01 | 9.6746e-01 |
| 10 | 9.6834e-01 | 9.6746e-01 |
| 11 | 9.6757e-01 | 9.6746e-01 |

Table 3.7: Values of $\delta^{2(k-1)}$ and $\delta_{2m,2(k-1)}$ as the relaxed iteration proceeds, with data as in figure 3.2.

trix); to explore the variable inner accuracy we used $\varepsilon_{out} = 10^{-9}$ and $m_{\max} = 100$. Table 3.8 shows the largest ten singular values obtained with a fixed inner tolerance of $10^{-11}$ ($\tilde{\sigma}_j$, second column), and with a

relaxed inner tolerance ($\tilde{\sigma}_j^{(fl)}$, third column), which a-posteriori we observed to go from $10^{-11}$ up to $10^{-5}$. The last column reports the relative error $|\tilde{\sigma}_j - \tilde{\sigma}_j^{(fl)}|/\tilde{\sigma}_j$. In both cases, the iteration of the inexact Lanczos bidiagonalization was stopped as soon as the outer stopping criterion was satisfied for the largest singular value. While in the fixed inner tolerance case the number of iterations varied between 28 and 30, in the flexible case a number of iterations as low as 15 was needed to satisfy the inner criterion at the last stage of the convergence process. After exiting the flexible procedure, however, the first ten approximate singular values are very close to those obtained with the fixed inner tolerance, much closer than warranted by the final inner accuracy of $10^{-5}$. This shows in particular that the flexible inner tolerance is conservative, and more accurate approximations are usually expected.

| $j$ | $\tilde{\sigma}_j$ | $\tilde{\sigma}_j^{(fl)}$ | $\lvert\tilde{\sigma}_j - \tilde{\sigma}_j^{(fl)}\rvert/\tilde{\sigma}_j$ |
|---|---|---|---|
| 1 | 1.117020718026223 | 1.117020718026212 | 9.93e-15 |
| 2 | 1.107884805324699 | 1.107884805324724 | 2.26e-14 |
| 3 | 1.098394607515649 | 1.098394607513931 | 1.56e-12 |
| 4 | 1.095557563655289 | 1.095557550135225 | 1.23e-08 |
| 5 | 1.087939266226247 | 1.087939266157844 | 6.28e-11 |
| 6 | 1.081739455193175 | 1.081739454786304 | 3.76e-10 |
| 7 | 1.077326677541678 | 1.077326677826174 | 2.64e-10 |
| 8 | 1.070641401649297 | 1.070641400153385 | 1.39e-09 |
| 9 | 1.064637797334345 | 1.064637795718615 | 1.51e-09 |
| 10 | 1.055679471834666 | 1.055679470916211 | 8.70e-10 |

Table 3.8: First ten approximate singular values of $A_1^{-1/2}$ with fixed tolerance ($\varepsilon_{out} = 10^{-9}$), and relaxed inner tolerance.

## 3.9 FINAL CONSIDERATIONS

We have explored the use and properties of an inexact Lanczos bidiagonalization method for approximating the leading singular triplet of a large matrix function, and in particular its 2-norm. Although several strategies are known to provide rough estimates of a matrix function 2-norm, more accurate approximations require a careful implementation

of available approaches, since neither $f(A)$ nor products of the type $f(A)v$ are available exactly. In particular, we showed that the inexact Lanczos bidiagonalization yields a non-Hermitian perturbation of the original Hermitian matrix, and the recurrence needs to be revisited.

Our numerical experiments showed that the computational complexity may vary significantly depending on the requested final accuracy, since the two inner iterations in which $f(A)v$ and $f(A)^*u$ are approximated may be very time and memory consuming. We showed that the relaxed strategy alleviates this problem whenever accurate approximations are required. However, for particular selections of matrices and functions, the approximation of $f(A)v$ can still be very expensive, and some other strategies could be exploited, such as restarting; see, e.g., [21, 24, 36] and references therein.

Finally, our approach could be used to estimate the norm of other matrix objects, such as the geometric mean [9], or the *derivatives* of matrix functions, such as the Fréchet derivative of the matrix exponential or of other functions [40]. We also mention that in the solution of time-dependent differential equations the evaluation of $\|f(tA)\|$ for $t > 0$ is of great interest to monitor the presence of transient behaviors for $A$ nonnormal. This problem requires ad-hoc analysis and specialized algorithmic strategies to limit computational costs, and it will be investigated in future research.

# 4

# THE INFINITE BI-LANCZOS METHOD FOR NONLINEAR EIGENVALUE PROBLEMS

*Adapted from [27]*

In this chapter we propose a two-sided Lanczos method for the nonlinear eigenvalue problem (NEP). This two-sided approach provides approximations to both the right and left eigenvectors of the eigenvalues of interest. The method implicitly works with matrices and vectors with infinite size, but because particular (starting) vectors are used, all computations can be carried out efficiently with finite matrices and vectors. We specifically introduce a new way to represent infinite vectors that span the subspace corresponding to the conjugate transpose operation for approximating the left eigenvectors. Furthermore, we show that also in this infinite-dimensional interpretation the short recurrences inherent to the Lanczos procedure offer an efficient algorithm regarding both the computational cost and the storage.

## 4.1 INTRODUCTION

Let $M : \mathbb{C} \to \mathbb{C}^{n\times n}$ be a matrix depending on a parameter with elements that are analytic in $\rho\bar{\mathbb{D}}$, where $\rho > 0$ is a constant, $\mathbb{D}$ is the open unit disk and $\bar{\mathbb{D}}$ its closure. We present a new method for the nonlinear eigenvalue problem: find $(\lambda, x, y) \in \rho\mathbb{D} \times \mathbb{C}^n \times \mathbb{C}^n$, where $x \neq 0$, $y \neq 0$, such that

$$M(\lambda)\, x = 0, \tag{4.1a}$$

$$M(\lambda)^* y = 0. \tag{4.1b}$$

We are interested in both the left and the right eigenvectors of the problem. The simultaneous approximation of both left and right eigenvectors is useful, e.g., in the estimation of the eigenvalue condition number and the vectors can be used as initial values for locally convergent two-sided iterative methods, e.g., those described in [72]. The NEP (4.1) has received considerable attention in the numerical linear algebra community, and there are several competitive numerical methods. There are for instance, so-called single vector methods such as Newton type

methods [20, 72, 78], which often can be improved with subspace acceleration, see [84], and Jacobi–Davidson methods [7]. These have been extended in a block sense [57]. There are methods specialized for symmetric problems that have an (easily computable) Rayleigh functional [79]. There is also a recent class of methods which can be interpreted as either dynamically extending an approximation or carrying out an infinite-dimensional algorithm, see for instance [4, 37, 53] and references therein. For recent developments see the summary papers [63, 69, 85] and the benchmark collection [8].

### 4.1.1 *Contributions of this chapter to the problem*

We propose a new method that is based on the two-sided Lanczos method for non-Hermitian problems. An intuitive derivation of the main idea of this chapter is the following. Suppose $(\lambda, x)$ is a solution to (4.1a). By adding trivial identities we obtain an equality between vectors of infinite length (cf. [53])

$$\begin{bmatrix} -M(0) & & & \\ & I & & \\ & & I & \\ & & & \ddots \end{bmatrix} \begin{bmatrix} \frac{\lambda^0}{0!}x \\ \frac{\lambda^1}{1!}x \\ \frac{\lambda^2}{2!}x \\ \vdots \end{bmatrix} = \lambda \begin{bmatrix} \frac{1}{1}M'(0) & \frac{1}{2}M''(0) & \frac{1}{3}M'''(0) & \cdots \\ \frac{1}{1}I & & & \\ & \frac{1}{2}I & & \\ & & \frac{1}{3}I & \\ & & & \ddots \end{bmatrix} \begin{bmatrix} \frac{\lambda^0}{0!}x \\ \frac{\lambda^1}{1!}x \\ \frac{\lambda^2}{2!}x \\ \vdots \end{bmatrix}. \tag{4.2}$$

Here, $I$ is the $n \times n$ identity matrix. One variant of the infinite Arnoldi method [53] is based on carrying out Arnoldi's method on the infinite-dimensional system (4.2). Our approach is based on two-sided Lanczos and requires analysis also of the transposed matrix. Throughout this chapter we assume that 0 is not an eigenvalue, so that $M(0)^{-1}$ exists. (This does not represent a loss of generality, as we can apply a shift in case 0 is an eigenvalue.) Let $\mathbf{N} \in \mathbb{C}^{n\times\infty}$ be defined by

$$\begin{aligned} \mathbf{N} &:= \begin{bmatrix} N_1 & N_2 & N_3 & \cdots \end{bmatrix} \\ &:= \begin{bmatrix} -M(0)^{-1}M'(0) & -\frac{1}{2}M(0)^{-1}M''(0) & -\frac{1}{3}M(0)^{-1}M'''(0) & \cdots \end{bmatrix} \end{aligned}$$

and define a vector of infinite length $\mathbf{v} := [v_j]_{j=1}^{\infty} = [\frac{\lambda^{(j-1)}}{(j-1)!}x]_{j=1}^{\infty}$, where $v_j \in \mathbb{C}^n$ for $j = 1, 2, \ldots$. Relation (4.2) can now be more compactly expressed as

$$\mathbf{v} = \lambda \left( \mathbf{e}_1 \otimes \mathbf{N} + \mathbf{S} \otimes I \right) \mathbf{v}, \quad \text{where} \quad \mathbf{S} := \begin{bmatrix} 0 & 0 & 0 & \cdots \\ \frac{1}{1} & & & \\ & \frac{1}{2} & & \\ & & \frac{1}{3} & \\ & & & \ddots \end{bmatrix}, \tag{4.3}$$

and $\mathbf{e}_1 = \begin{bmatrix} 1 & 0 & 0 & \cdots \end{bmatrix}^T$ is the first basis vector. Equations (4.2) and (4.3) may be viewed as a companion linearization for the nonlinear eigenvalue problem. Note that a solution $\lambda$ to (4.3) corresponds to a reciprocal eigenvalue of the infinite-dimensional matrix

$$\mathbf{A} := \mathbf{e}_1 \otimes \mathbf{N} + \mathbf{S} \otimes I. \tag{4.4}$$

The derivation of our new bi-Lanczos procedure is based on applying the Lanczos method (for non-Hermitian problems) to the infinite-dimensional matrix $\mathbf{A}$. The method builds two bi-orthogonal subspaces using short recurrences. One subspace serves the approximation of right eigenvectors, the other the approximation of the left eigenvectors.

### 4.1.2 *Overview of the chapter*

In section 4.2 we derive several results for infinite-dimensional matrices of the type (4.4) and associated infinite vectors (eigenvectors and elements of vectors in an associated Krylov subspace). In particular, analogous to companion linearizations for polynomial eigenvalue problems, relation (4.3) is equivalent to (4.1a), which has been used in [53]. For the approximation of solutions to (4.1b) we derive a new and more involved relationship for the left eigenvectors, also presented in section 4.2. This leads to a new way to represent infinite vectors that span the subspace corresponding to the conjugate transpose operation for approximating the left eigenvectors. With two particular types of (starting) vectors, we can carry out an algorithm for the infinite-dimensional operator $\mathbf{A}$ using only finite arithmetic. This is covered in the first four subsections of section 4.3. The second half of section 4.3 is dedicated

to various computational issues and complexity considerations. In section 4.4 we present a few examples to illustrate the performance of the new method, and we conclude with a short discussion.

Throughout this chapter we use bold symbols to indicate matrices or vectors of infinite dimensions, i.e., an infinite matrix is denoted by $\mathbf{A} \in \mathbb{C}^{\infty\times\infty}$, and an infinite-dimensional vector is denoted by $\mathbf{x} \in \mathbb{C}^{\infty}$. Unless otherwise stated, the length-$n$ blocks of a vector of infinite length are denoted with subscript, e.g., $\mathbf{w} = [w_1^T, w_2^T, \dots]^T$ where $w_j \in \mathbb{C}^n$ for $j \geq 1$.

## 4.2 INFINITE-DIMENSIONAL REFORMULATION

We reformulate the nonlinear eigenvalue problem as a linear eigenvalue problem by showing the equivalence between the two problems. The new representation involves infinite-dimensional matrices and vectors. The algorithm that will be introduced in the next section generates two Krylov subspaces, and for these subspaces we need to distinguish two ways to characterize infinite-dimensional vectors. In this section we show that both types of infinite-dimensional vectors can be represented by a finite number of vectors of length $n$. Furthermore, various operations with these infinite-dimensional matrices and vectors are explored.

### 4.2.1 *The nonlinear eigenvalue problem and the operator* $\mathbf{A}$

In our formalization of the operator $\mathbf{A}$ we first need to define its domain. This is necessary to prove equivalence between $(\lambda, x, y)$ which is a solution to (4.1) and the eigentriplet $(\mu, \mathbf{v}, \mathbf{w})$ of $\mathbf{A}$, where $\mu = \lambda^{-1}$. Let $\|\cdot\|$ denote the 2-norm. It turns out to be natural to define the operators on a weighted, mixed 1-norm and 2-norm space defined by

$$\ell_1(\rho) := \Big\{ \mathbf{w} = [w_j]_{j=1}^{\infty} \in \mathbb{C}^{\infty} : \sum_{j=1}^{\infty} \tfrac{\rho^j}{j!} \|w_j\| < \infty \Big\}. \tag{4.5}$$

Note that some vectors in $\ell_1(\rho)$ correspond to sequences of vectors that are unbounded, i.e., $\|w_j\| \to \infty$ as $j \to \infty$, but do not grow arbitrarily fast, since $\mathbf{w} \in \ell_1(\rho)$ implies that

$$\tfrac{\rho^j}{j!} \|w_j\| \to 0 \quad \text{as } j \to \infty. \tag{4.6}$$

In the proofs of the theorems and propositions below we need to allow the vectors to have this growth, to accommodate the fact that derivatives of analytic functions are not necessarily bounded. We choose $\rho$ to be the convergence radius of the power series expansion of the analytic function $M$, and set $\mathcal{D}(\mathbf{A}) = \mathcal{D}(\mathbf{A}^*) = \ell_1(\rho)$ as the domain of the operator. The following two theorems do not only show the equivalence between the nonlinear eigenvalue problem and the operator $\mathbf{A}$, but also reveal the structure of the left and right eigenvectors of $\mathbf{A}$. The first result is an adaption of [53, theorem 1] for our discrete operator and only assuming a finite convergence radius.

**Theorem 4.2.1 (Right eigenvectors of A)** *Suppose $M$ is analytic in $\lambda \in \rho\bar{\mathbb{D}}$ and let $\mathbf{A}$ be defined by (4.4).*

*(i) If $(\mu, \mathbf{v}) \in \mathbb{C} \times \mathcal{D}(\mathbf{A})\backslash\{0\}$ is an eigenpair of $\mathbf{A}$ and $\lambda = \mu^{-1} \in \rho\mathbb{D}$, then there exists a vector $x \in \mathbb{C}^n$ such that*

$$\mathbf{v} = \left[\tfrac{\lambda^{j-1}}{(j-1)!}\, x\right]_{j=1}^{\infty}. \tag{4.7}$$

*(ii) The pair $(\lambda, x) \in \rho\mathbb{D}\backslash\{0\} \times \mathbb{C}^n\backslash\{0\}$ is a solution to (4.1a) if and only if the pair $\left(\lambda^{-1}, \mathbf{v}\right) \in (\mathbb{C}\backslash\rho^{-1}\bar{\mathbb{D}}) \times \mathcal{D}(\mathbf{A})$ is an eigenpair of $\mathbf{A}$, where $\mathbf{v}$ is given by (4.7).*

*Proof.* To show (i), let $\mathbf{v} = [v_j]_{j=1}^{\infty}$, where $v_j \in \mathbb{C}^n$ are the blocks of $\mathbf{v}$. From the block rows $2, 3, \ldots$ of $\lambda\mathbf{A}\mathbf{v} = \mathbf{v}$, we have that $v_{j+1} = \frac{\lambda}{j}\, v_j$ for $j = 1, 2, \ldots$. It follows from induction that the blocks in the eigenvector satisfy

$$v_j = \tfrac{\lambda^{j-1}}{(j-1)!}\, v_1, \quad j = 1, 2, \ldots. \tag{4.8}$$

We also have that $\mathbf{v} \in \ell_1(\rho)$, since $\|v_j\| = \frac{|\lambda|^{j-1}}{(j-1)!}\, \|v_1\|$, such that $\mathbf{v} \in \ell_1 \subset \ell_1(\rho)$.

To show (ii), assume first that $(\lambda^{-1}, \mathbf{v})$ is an eigenpair of $\mathbf{A}$. From (i) we know that the blocks of $\mathbf{v}$ satisfy $v_j = \frac{\lambda^{j-1}}{(j-1)!}\, v_1$. The first block row of $\mathbf{v} = \lambda\mathbf{A}\mathbf{v}$ implies that

$$\begin{aligned} v_1 &= \lambda \sum_{j=1}^{\infty} N_j\, \tfrac{\lambda^{j-1}}{(j-1)!}\, v_1 = -\sum_{j=1}^{\infty} \tfrac{\lambda^j}{j!} M(0)^{-1}\, M^{(j)}(0)\, v_1 \\ &= -M(0)^{-1}\, (M(\lambda) - M(0))\, v_1. \end{aligned}$$

Therefore, since 0 is not an eigenvalue, $(\lambda, v_1)$ is a solution to (4.1a).

To show the converse, suppose that $(\lambda, x)$ is a solution to (4.1a). Let $\mathbf{v}$ be as in (4.7). The rest of the proof consists of showing that

$$\lambda \mathbf{A}\mathbf{v} = \mathbf{v}. \tag{4.9}$$

Similar to above, the first block row of $\lambda\mathbf{A}\mathbf{v}$ is

$$-\lambda \sum_{j=1}^{\infty} \tfrac{1}{j}\, M(0)^{-1}\, M^{(j)}(0)\, v_j = -M(0)^{-1} M(\lambda)\, x + x = x.$$

In the last step we used that $M(\lambda)x = 0$, since $(\lambda, x)$ is a solution to (4.1a). Hence, the equality in the first block row of (4.9) is proven. The equality in (4.9) corresponding to blocks $j > 1$ follows from the fact that $v_{j+1} = (\lambda/j)\, v_j$, for $j = 1, 2, \ldots$, by construction. □

We now study the equivalence between a left eigenpair of the nonlinear eigenvalue problem and a left eigenpair of $\mathbf{A}$. Also, the structure of the left eigenvectors of $\mathbf{A}$ will be concretized.

**Theorem 4.2.2 (Left eigenvectors of A)** *Suppose $M$ is analytic in $\lambda \in \rho\mathbb{D}$ and let $\mathbf{A}^*$ be defined by* (4.4)*.*

*(i) If $(\mu, \mathbf{w}) \in \mathbb{C} \times \mathcal{D}(\mathbf{A}^*)\backslash\{0\}$ is an eigenpair of $\mathbf{A}^*$ and $\lambda = \mu^{-1} \in \rho\mathbb{D}$, then there exists a vector $z \in \mathbb{C}^n$ such that*

$$\mathbf{w} = \sum_{j=1}^{\infty} (\mathbf{S}^T \otimes I)^{j-1} \mathbf{N}^* \lambda^j z. \tag{4.10}$$

*(ii) The pair $(\lambda, y) \in \rho\mathbb{D}\backslash\{0\} \times \mathbb{C}^n\backslash\{0\}$ is a solution to* (4.1b) *if and only if the pair $\left(\lambda^{-1}, \mathbf{w}\right) \in (\mathbb{C}\backslash\rho^{-1}\bar{\mathbb{D}}) \times \mathcal{D}(\mathbf{A}^*)$ is an eigenpair of $\mathbf{A}^*$, where $\mathbf{w}$ is given by* (4.10) *with $z = M(0)^* y$.*

*Proof.* Suppose $\lambda\mathbf{A}^*\mathbf{w} = \mathbf{w}$, where $\mathbf{w} \in \ell_1(\rho)$. We use induction to show that

$$w_1 = \sum_{j=1}^{k} \tfrac{\lambda^j}{(j-1)!} N_j^*\, w_1 + \tfrac{\lambda^k}{k!}\, w_{k+1} \tag{4.11}$$

for any $k$. Relation (4.11) is easy to see for $k = 1$. Suppose (4.11) is satisfied for $k - 1$, i.e.,

$$w_1 = \sum_{j=1}^{k-1} \tfrac{\lambda^j}{(j-1)!} N_j^*\, w_1 + \tfrac{\lambda^{k-1}}{(k-1)!}\, w_k. \tag{4.12}$$

Block row $k$ of $\lambda \mathbf{A}^* \mathbf{w} = \mathbf{w}$ reduces to

$$\lambda\, N_k^*\, w_1 + \tfrac{\lambda}{k}\, w_{k+1} = w_k. \tag{4.13}$$

The induction is completed by inserting (4.13) in relation (4.12), which yields (4.11). Due to the fact that $\mathbf{w} \in \ell_1(\rho)$, (4.6) holds, and since $|\lambda| < \rho$, we have $\|\frac{\lambda^k}{k!} w_{k+1}\| < \frac{\rho^k}{k!} \|w_{k+1}\| \to 0$ as $k \to \infty$. This implies that (4.11) holds also in the limit $k \to \infty$ and

$$w_1 = \sum_{j=1}^{\infty} \tfrac{\lambda^j}{(j-1)!} N_j^*\, w_1 = (\mathbf{e}_1^T \otimes I)\Big( \sum_{j=1}^{\infty} \lambda^j (\mathbf{S}^T \otimes I)^{j-1} \mathbf{N}^* w_1 \Big). \tag{4.14}$$

In the last equality in (4.14) we used that

$$\mathbf{S}^j \mathbf{e}_k = \tfrac{(k-1)!}{(j+k-1)!}\, \mathbf{e}_{k+j} \tag{4.15}$$

and therefore $(\mathbf{e}_k^T \otimes I)(\mathbf{S}^T \otimes I)^{j-1} = \mathbf{e}_k^T (\mathbf{S}^T)^{j-1} \otimes I = \frac{(k-1)!}{(j+k-2)!} \mathbf{e}_{k+j-1}^* \otimes I$ for any $k$, as well as $(\mathbf{e}_j^T \otimes I)\mathbf{N}^* = N_j^*$. We have proven the first block row of (4.10) with (4.14), by setting $z = w_1$. The proof of the other rows follows from induction, since assuming that $w_k = (\mathbf{e}_k^T \otimes I)\mathbf{w}$, where $\mathbf{w}$ is the right-hand side of (4.10), and using (4.13) we find that $w_{k+1} = (\mathbf{e}_{k+1}^* \otimes I)\mathbf{w}$.

To show (ii), first assume that $\mathbf{w} \in \ell_1(\rho)$ satisfies $\lambda \mathbf{A} \mathbf{w} = \mathbf{w}$. This is the same assumption as in (i) and therefore (4.14) is satisfied. By setting $y = M(0)^{-*} z = M(0)^{-*} w_1$, we have that $M(0)^* y = \sum_{j=1}^{\infty} M^{(j)}(0)^* y$, i.e., (4.1b) is satisfied. To show the backward implication in (ii), we now assume that $(\lambda, y)$ is a solution to (4.1b). Let $z = M(0)^* y$ and define a vector $\mathbf{w}$ as (4.10). Then

$$\begin{aligned}
\lambda \mathbf{A}^* \mathbf{w} &= \lambda \sum_{j=1}^{\infty} (\mathbf{e}_1^T \otimes \mathbf{N}^* + \mathbf{S}^T \otimes I)(\mathbf{S}^T \otimes I)^{j-1} \mathbf{N}^* \lambda^j z \\
&= \lambda \mathbf{N}^* \sum_{j=1}^{\infty} \tfrac{1}{(j-1)!} (\mathbf{e}_j^T \otimes I) \mathbf{N}^* \lambda^j z + \lambda \sum_{j=1}^{\infty} (\mathbf{S}^T \otimes I)^j \mathbf{N}^* \lambda^j z \\
&= \lambda \mathbf{N}^* \sum_{j=1}^{\infty} \tfrac{-1}{j!} M^{(j)}(0)^* y + \sum_{j=2}^{\infty} (\mathbf{S}^T \otimes I)^{j-1} \mathbf{N}^* \lambda^j z \\
&= \mathbf{N}^* \lambda z + \sum_{j=2}^{\infty} (\mathbf{S}^T \otimes I)^{j-1} \mathbf{N}^* \lambda^j z \\
&= \sum_{j=1}^{\infty} (\mathbf{S}^T \otimes I)^{j-1} \mathbf{N}^* \lambda^j z = \mathbf{w}.
\end{aligned}$$

To show that $\mathbf{w} \in \ell_1(\rho)$ we now study the weighted $\ell_1$-norm,

$$\begin{aligned}\sum_{k=1}^{\infty} \tfrac{\rho^k}{k!}\|w_k\| &\leq \sum_{k=1}^{\infty} \tfrac{\rho^k}{k!} \sum_{j=1}^{\infty} \tfrac{|\lambda|^j (k-1)!}{(j+k-2)!} \|M^{(k+j-1)}(0)^*\| \|\widehat{y}\| \\ &\leq \sum_{k=1}^{\infty} \tfrac{\rho^k}{k!} \sum_{j=1}^{\infty} M_\rho \tfrac{|\lambda|^j (k-1)!(k+j-1)!}{(j+k-2)!\rho^{j+k-1}} \|\widehat{y}\| \\ &= \tfrac{M_\rho \|\widehat{y}\|}{r} \sum_{j=1}^{\infty} \sum_{k=1}^{\infty} \tfrac{|\lambda|^j}{\rho^j} \tfrac{j+k-1}{k!}.\end{aligned} \tag{4.16}$$

Since $M$ is analytic, there exists a constant $M_\rho$ such that $\|M^{(j)}(0)\| \leq M_\rho \frac{j!}{\rho^j}$. Now note that Taylor expansion of $e^x$ gives the explicit expression $\sum_{k=1}^{\infty} \frac{j+k-1}{k!} = (j-1)(e-1)+e$. By combining this with (4.16) and $|\lambda| < \rho$ we find that the right-hand side of (4.16) is finite and therefore $\mathbf{w} \in \ell_1(\rho)$. □

### 4.2.2 *Krylov subspace and infinite-dimensional vector representations*

In the next section we will develop a Krylov method for the infinite-dimensional problem. The method is based on operations with the matrix $\mathbf{A}$ and vectors of infinite length: it builds infinite-dimensional Krylov subspaces, $\mathcal{K}_k(\mathbf{A}, \mathbf{x})$ and $\mathcal{K}_k(\mathbf{A}^*, \widetilde{\mathbf{y}})$, for some starting vectors $\mathbf{x}$ and $\widetilde{\mathbf{y}}$ of infinite length. Therefore, we have to address the issue of storing vectors with infinite length. By choosing the starting vectors carefully we will be able to store only a finite number of vectors of length $n$. The Krylov subspaces will contain approximate eigenvectors that are the eigenvectors of interest.

**Proposition 4.2.3** *Suppose* $\mathbf{x} = \mathbf{e}_1 \otimes x_1$ *and* $\widetilde{\mathbf{y}} = \mathbf{N}^* \widetilde{y}_1$, *where* $x_1, \widetilde{y}_1 \in \mathbb{C}^n$.

*(a) For any* $k \in \mathbb{N}$, $\mathbf{A}^k \mathbf{x} = \sum_{j=1}^{k+1} (\mathbf{e}_j \otimes z_{k-j+1})$, *where* $z_0 = \frac{1}{k!} x_1$ *and for* $i \in \{1, \ldots, k\}$ $z_i$ *is given by the recursion* $z_i = \sum_{\ell=1}^{i} \frac{(k-i+\ell)!}{(\ell-1)!(k-i)!} N_\ell z_{i-\ell}$.

*(b) For any* $k \in \mathbb{N}$, $(\mathbf{A}^*)^k \widetilde{\mathbf{y}} = \sum_{j=1}^{k+1} (\mathbf{S}^T \otimes I)^{j-1} \mathbf{N}^* \widetilde{z}_{k-j+1}$, *where* $\widetilde{z}_0 = \widetilde{y}_1$ *and for* $i \in \{1, \ldots, k\}$ $\widetilde{z}_i$ *is given by the recurrence relation* $\widetilde{z}_i = \sum_{\ell=1}^{i} \frac{1}{(\ell-1)!} N_\ell^* \widetilde{z}_{i-\ell}$.

*Proof.* (a) It is easily seen that the result holds for $k = 1$, when $z_0 = x_1$ and $z_1 = N_1 z_0$. Suppose the result holds for $k-1$, thus

$$\mathbf{A}^{k-1}\mathbf{x} = \sum_{j=1}^{k} (\mathbf{e}_j \otimes a_{k-j}),$$

where $a_0 = \frac{1}{(k-1)!} x_1$ and $a_i = \sum_{\ell=1}^{i} \frac{(k-i+\ell-1)!}{(\ell-1)!(k-i-1)!} N_\ell a_{i-\ell}$ for $i \in \{1, \ldots, k-1\}$. Then, using (4.15),

$$\begin{aligned}
\mathbf{A}^k\mathbf{x} &= \sum_{j=1}^{k} (\mathbf{e}_1 \otimes \mathbf{N})(\mathbf{e}_j \otimes a_{k-j}) + \sum_{j=1}^{k} (\mathbf{S} \otimes I)(\mathbf{e}_j \otimes a_{k-j}) \\
&= \sum_{j=1}^{k} (\mathbf{e}_1 \otimes N_j a_{k-j}) + \sum_{j=1}^{k} \tfrac{(j-1)!}{j!} (\mathbf{e}_{j+1} \otimes a_{k-j}) \\
&= (\mathbf{e}_1 \otimes \sum_{j=1}^{k} N_j a_{k-j}) + \sum_{j=2}^{k+1} (\mathbf{e}_j \otimes \tfrac{1}{(j-1)} a_{k-j+1}).
\end{aligned}$$

Defining $z_k = \sum_{j=1}^{k} N_j a_{k-j}$ and $z_{k-j+1} = \frac{1}{(j-1)} a_{k-j+1}$, it can be seen that all $z_i$ are as stated in (a). This shows (a) by induction.

(b) It is easily seen that for $k = 1$ the result holds, where $\widetilde{z}_0 = \widetilde{y}_1$ and $\widetilde{z}_1 = N_1 \widetilde{z}_0$. Suppose the proposition holds for $k-1$. Then

$$\begin{aligned}
(\mathbf{A}^*)^k \widetilde{\mathbf{y}} &= \sum_{j=1}^{k} (\mathbf{e}_1^T \otimes \mathbf{N}^*)(\mathbf{S}^T \otimes I)^{j-1} \mathbf{N}^* \widetilde{z}_{k-j} + \sum_{j=1}^{k} (\mathbf{S}^T \otimes I)^j \mathbf{N}^* \widetilde{z}_{k-j} \\
&= \sum_{j=1}^{k} \tfrac{1}{(j-1)!} (\mathbf{e}_j^T \otimes \mathbf{N}^*) \mathbf{N}^* \widetilde{z}_{k-j} + \sum_{j=2}^{k+1} (\mathbf{S}^T \otimes I)^{j-1} \mathbf{N}^* \widetilde{z}_{k-j+1} \\
&= \mathbf{N}^* \sum_{j=1}^{k} \tfrac{1}{(j-1)!} N_j^* \widetilde{z}_{k-j} + \sum_{j=2}^{k+1} (\mathbf{S}^T \otimes I)^{j-1} \mathbf{N}^* \widetilde{z}_{k-j+1} \\
&= \sum_{j=1}^{k+1} (\mathbf{S}^T \otimes I)^{j-1} \mathbf{N}^* \widetilde{z}_{k-j+1},
\end{aligned}$$

where $\widetilde{z}_0 = \widetilde{y}_1$, $\widetilde{z}_1$, …, $\widetilde{z}_m$ are as stated under (b). This proves (b) by induction. □

As we have seen in theorems 4.2.1 and 4.2.2, the right and left eigenvectors of interest have the form (4.7) and (4.10), respectively. Proposition 4.2.3 has shown that by choosing starting vectors $\mathbf{x} = \mathbf{e}_1 \otimes x_1$ and $\widetilde{\mathbf{y}} = \mathbf{N}^* \widetilde{y}_1$ the vectors that span the Krylov subspaces $\mathcal{K}_{k_a}(\mathbf{A}, \mathbf{x})$ and $\mathcal{K}_{k_{\widetilde{a}}}(\mathbf{A}^*, \widetilde{\mathbf{y}})$ are of the form

$$\mathbf{a} = \sum_{j=1}^{k_a} (\mathbf{e}_j \otimes a_j), \tag{4.17a}$$

$$\widetilde{\mathbf{a}} = \sum_{j=1}^{k_{\widetilde{a}}} (\mathbf{S}^T \otimes I)^{j-1} \mathbf{N}^* \widetilde{a}_j, \tag{4.17b}$$

respectively. Also linear combinations of vectors from the same Krylov subspaces will be of this form. These vectors can be seen as truncated versions of the vectors in (4.7) and (4.10). In fact, the approximate eigenvectors that are taken from these two Krylov subspaces, will be of this form. We will distinguish the two types of vectors (4.17a) and (4.17b) by a tilde. Vectors of the form (4.17a) have a finite number of nonzeros and therefore storing only the nonzero entries gives a finite representation of the vector of infinite length, i.e., by storing the vectors $a_j$, for $j = 1, \ldots, k_a$. The vectors of infinite length of type (4.17b) can also be stored with a finite number of vectors in $\mathbb{C}^n$, namely by storing the vectors $\widetilde{a}_j$, for $j = 1, \ldots, k_{\widetilde{a}}$.

### 4.2.3 *Scalar products and matrix-vector products*

The previously introduced types of infinite-dimensional vectors, (4.17a) and (4.17b), will be used in the algorithm for the infinite-dimensional problem. Various operations involving these types of vectors of infinite length, such as scalar products and matrix-vector products, have to be adapted to the infinite-dimensional case. First we introduce two different scalar products.

**Lemma 4.2.4** *Suppose* $\mathbf{a}, \mathbf{b} \in \mathbb{C}^\infty$ *are two vectors of type* (4.17a) *given by* $\mathbf{a} = \sum_{j=1}^{k_a} (\mathbf{e}_j \otimes a_j)$ *and* $\mathbf{b} = \sum_{j=1}^{k_b} (\mathbf{e}_j \otimes b_j)$. *Then,*

$$\mathbf{a}^* \mathbf{b} = \sum_{j=1}^{\min(k_a, k_b)} a_j^* b_j. \tag{4.18}$$

*Proof.* This follows straightforwardly from the definition of the vectors. □

Another scalar product used in the bi-Lanczos algorithm is a product of vectors of type (4.17a) and (4.17b). It can be computed efficiently in infinite dimensions as explained in the next proposition.

**Theorem 4.2.5** *Suppose* $\widetilde{\mathbf{a}}, \mathbf{b} \in \mathbb{C}^\infty$ *are of type* (4.17b) *and* (4.17a)*, respectively, given by* $\widetilde{\mathbf{a}} = \sum_{j=1}^{k_{\tilde{a}}} (\mathbf{S}^T \otimes I)^{j-1} \mathbf{N}^* \widetilde{a}_j$ *and* $\mathbf{b} = \sum_{\ell=1}^{k_b} (\mathbf{e}_\ell \otimes b_\ell)$*. Then,*

$$\widetilde{\mathbf{a}}^* \mathbf{b} = \sum_{j=1}^{k_{\tilde{a}}} \sum_{\ell=1}^{k_b} \tfrac{(\ell-1)!}{(j+\ell-2)!} \widetilde{a}_j^* N_{j+\ell-1} b_\ell. \tag{4.19}$$

*Proof.* This can be derived directly via the following equality

$$\begin{aligned} \widetilde{\mathbf{a}}^* \mathbf{b} &= \sum_{j=1}^{k_{\tilde{a}}} \sum_{\ell=1}^{k_b} \Big( (\mathbf{S}^T \otimes I)^{j-1} \mathbf{N}^* \widetilde{a}_j \Big)^* (\mathbf{e}_\ell \otimes b_\ell) \\ &= \sum_{j=1}^{k_{\tilde{a}}} \sum_{\ell=1}^{k_b} \widetilde{a}_j^* \mathbf{N} \Big( \mathbf{e}_{j+\ell-1} \otimes \tfrac{(\ell-1)!}{(j+\ell-2)!} b_\ell \Big). \end{aligned}$$

□

To translate the finite dimensional matrix-vector multiplication to the infinite-dimensional case two variants of matrix-vector products have to be investigated, one with the matrix $\mathbf{A}$ and a vector of type (4.17a), and one with the matrix $\mathbf{A}^*$ and a vector of type (4.17b).

**Theorem 4.2.6 (Action of A)** *Suppose* $\mathbf{a} \in \mathbb{C}^\infty$ *is of type* (4.17a) *given by* $\mathbf{a} = \sum_{j=1}^{k_a} (\mathbf{e}_j \otimes a_j)$*. Then,*

$$\mathbf{A}\mathbf{a} = \sum_{j=1}^{k_a+1} (\mathbf{e}_j \otimes b_j), \tag{4.20}$$

*where*

$$b_j = \tfrac{1}{j-1} a_{j-1} \text{ for } j = 2, \ldots, k_a + 1, \text{ and } b_1 = \sum_{j=1}^{k_a} N_j a_j. \tag{4.21}$$

*Proof.* This can be proven by induction. The computation is analogous to the one needed in the proof of Proposition 4.2.3(a). □

**Theorem 4.2.7 (Action of $\mathbf{A}^*$)** *Suppose* $\widetilde{\mathbf{a}} \in \mathbb{C}^\infty$ *is of type* (4.17b) *given by* $\widetilde{\mathbf{a}} = \sum_{j=1}^{k_{\widetilde{a}}} (\mathbf{S}^T \otimes I)^{j-1} \mathbf{N}^* \widetilde{a}_j$. *Then,*

$$\mathbf{A}^* \widetilde{\mathbf{a}} = \sum_{j=1}^{k_{\widetilde{a}}+1} (\mathbf{S}^T \otimes I)^{j-1} \mathbf{N}^* \widetilde{b}_j, \tag{4.22}$$

*where*

$$\widetilde{b}_j = \widetilde{a}_{j-1} \text{ for } j = 2, \ldots k_{\widetilde{a}} + 1, \text{ and } \widetilde{b}_1 = \sum_{j=1}^{k_{\widetilde{a}}} \tfrac{1}{(j-1)!} N_j^* \widetilde{a}_j. \tag{4.23}$$

*Proof.* Analogous to the computation in the proof of Proposition 4.2.3(b). □

## 4.3 DERIVATION OF THE INFINITE BI-LANCZOS METHOD

The algorithm proposed in this chapter is based on the Lanczos method for non-Hermitian eigenvalue problems specified in [1, section 7.8.1]. We first introduce the standard method and then adapt the algorithm in such a way that it can be used for the infinite-dimensional problem.

### 4.3.1 *The bi-Lanczos method for standard eigenvalue problems*

We now briefly summarize the bi-Lanczos method, already introduced in section 1.2.3, that we use in our derivation. The method, presented in algorithm 4.1, uses an oblique projection building two bi-orthogonal subspaces for the simultaneous approximation of left and right eigenvectors. The short recurrences that are typical for this method lead to far less storage requirements with respect to orthogonal projection methods for the same problem. However, as is well known, the method suffers from the loss of bi-orthogonality in finite precision arithmetic. One can either accept the loss and take more steps, or, if desired, one can re-biorthogonalize all vectors in each iteration. A compromise between these two options is to maintain semiduality as proposed in [15]. For information on various types of breakdowns, how to continue after a breakdown, and how to detect (near) breakdowns, we refer to [1, section 7.8.1], and [23].

**Algorithm 4.1:** Bi-Lanczos

**Input:** Vectors $q_1$, $\widetilde{q}_1$, with $\widetilde{q}_1^* q_1 = 1$, $\gamma_1 = \beta_1 = 0$, $q_0 = \widetilde{q}_0 = 0$.
**Output:** Approximate eigentriplets $(\theta_i^{(j)}, x_i^{(j)}, y_i^{(j)})$ of $A$.

**for** $j = 1, 2, \ldots$, until convergence
(1) $r = Aq_j$
(2) $s = A^* \widetilde{q}_j$
(3) $r := r - \gamma_j q_{j-1}$
(4) $s := s - \bar{\beta}_j \widetilde{q}_{j-1}$
(5) $\alpha_j = \widetilde{q}_j^* r$
(6) $r := r - \alpha_j q_j$
(7) $s := s - \bar{\alpha}_j \widetilde{q}_j$
(8) $\omega_j = r^* s$
(9) $\beta_{j+1} = |\omega_j|^{1/2}$
(10) $\gamma_{j+1} = \bar{\omega}_j / \beta_{j+1}$
(11) $q_{j+1} = r / \beta_{j+1}$
(12) $\widetilde{q}_{j+1} = s / \bar{\gamma}_{j+1}$
(13) Compute eigentriplets $(\theta_i^{(j)}, z_i^{(j)}, \widetilde{z}_i^{(j)})$ of $T_j$.
(14) Test for convergence.
(15) Rebiorthogonalize if necessary.
**end**
(16) Compute approximate eigenvectors $x_i^{(j)} = Q_j z_i^{(j)}$, $y_i^{(j)} = \widetilde{Q}_j \widetilde{z}_i^{(j)}$.

The algorithm takes as input two bi-orthogonal vectors, $q_1$ and $\widetilde{q}_1$, and builds the Krylov subspaces $\mathcal{K}_k(A, q_1)$ and $\mathcal{K}_k(A^*, \widetilde{q}_1)$ (lines (1)-(2)). The algorithm uses short recurrences for the bi-orthogonalization process (lines (3)-(7)). Although there are various choices for the scaling of the vectors that span the Krylov spaces, we specifically choose to scale the vectors in such a way that $\widetilde{q}_i^* q_j = \delta_{ij}$, where $\delta_{ij}$ is the Kronecker delta (lines (8)-(12)), as this will turn out to be necessary for the translation to the infinite-dimensional algorithm. After $k$ iterations we obtain the relations:

$$
\begin{aligned}
AQ_k &= Q_k T_k + \beta_{k+1} q_{k+1} e_k^T, \\
A^* \widetilde{Q}_k &= \widetilde{Q}_k T_k^* + \bar{\gamma}_{k+1} \widetilde{q}_{k+1} e_k^T, \\
\widetilde{Q}_k^* Q_k &= I_k,
\end{aligned}
$$

where for $i = 1, \ldots, k$ the columns of $Q_k$ are equal to the vectors $q_i$, and where $\widetilde{q}_i$ are the columns of $\widetilde{Q}_k$, $e_k$ is the $k$th unit vector, and where

$$T_k = \begin{bmatrix} \alpha_1 & \gamma_2 & & \\ \beta_2 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \gamma_k \\ & & \beta_k & \alpha_k \end{bmatrix},$$

where the coefficients $\alpha_j$, $\beta_j$ and $\gamma_j$, $j \in \{1, \ldots, k\}$, are computed in lines (5), (9) and (10). Furthermore, the relations $\widetilde{q}_{k+1}^* Q_k = 0$ and $\widetilde{Q}_k^* q_{k+1} = 0$ hold. After $k$ iterations, the eigentriplets $(\theta_i^{(k)}, z_i^{(k)}, \widetilde{z}_i^{(k)})$, $i = 1, 2, \ldots, k$, of $T_k$ (line (13)) can be computed. The Ritz values $\theta_i^{(k)}$ are the approximate eigenvalues of $A$, and the corresponding right and left Ritz vectors are $x_i^{(k)} = Q_k z_i^{(k)}$ and $y_i^{(k)} = \widetilde{Q}_k \widetilde{z}_i^{(k)}$, respectively (line (16)).

### 4.3.2 *The infinite bi-Lanczos method*

In algorithm 4.2 we present the bi-Lanczos algorithm for the infinite-dimensional problem. It is set up analogously to the algorithm of the standard bi-Lanczos method: all line numbers are corresponding. As we have seen, every vector of infinite length can be represented by a finite number of vectors of length $n$. In the algorithm these vectors of infinite length are denoted by matrices whose columns correspond to the length-$n$ vectors representing the infinite-dimensional vector. The index of the matrices in the new algorithm indicate the number of columns of the matrix, i.e., $R_k \in \mathbb{C}^{n \times k}$, and we denote the $\ell$th column of $R_k$ by $R_{k,\ell}$, i.e., $R_k = [R_{k,1}, \ldots, R_{k,k}]$.

We now describe certain steps of the algorithm illustrating that the algorithm can be implemented with matrices of finite size. This first description can be considerably improved by reducing the number of necessary linear solves as we shall explain in section 4.3.3. We refer to the previous subsection for details on steps (9)-(14).

(1)+(2) In the first two lines two matrix-vector multiplications are executed, the first with the infinite-dimensional matrix $\mathbf{A}$ and a vector of type (4.17a), the second with the infinite-dimensional matrix $\mathbf{A}^*$ and a vector of type (4.17b). In Theorems 4.2.6 and 4.2.7 it is shown how these actions are performed. A vector represented

by $k$ vectors of length $n$ results, after a multiplication with $\mathbf{A}$ or $\mathbf{A}^*$, in a vector represented by $k+1$ vectors of length $n$. More precisely, the actions (4.20) and (4.22) can be computed with the formulas

$$b_j = \tfrac{1}{j-1} a_{j-1} \text{ for } j = 2, \ldots, k_a + 1, \tag{4.24a}$$

$$b_1 = -M(0)^{-1} \sum_{j=1}^{k_a} \frac{1}{j} M^{(j)}(0) a_j, \tag{4.24b}$$

and

$$\widetilde{b}_j = \widetilde{a}_{j-1} \text{ for } j = 2, \ldots k_{\widetilde{a}} + 1, \tag{4.25a}$$

$$\widetilde{b}_1 = -\sum_{j=1}^{k_{\widetilde{a}}} M^{(j)}(0)^* \left( \tfrac{1}{j!} M(0)^{-*} \widetilde{a}_j \right). \tag{4.25b}$$

At first sight (4.25) appears to require an excessive number of linear solves. We show in section 4.3.3 how these linear solves can be avoided by using a particular implicit representation of $\widetilde{a}_j$.

(3)+(4) The new vectors are orthogonalized against a previous vector. Linear combinations of a vector of the form (4.17a) (or (4.17b)) are again of that form, and thus can be represented as such. The new vectors are represented by $k+1$ vectors of length $n$, while the previous vectors are represented by $k-1$ length-$n$ vectors. To enable the summation we add two zero columns to the $n \times (k-1)$-matrices representing the previous vectors.

(5)+(8) The coefficients computed in this step are needed for the orthogonalization of the vectors, and furthermore they are the entries of the tridiagonal matrix $T_k$. The computation of these coefficients involves an inner product between a vector of type (4.17a) and one of type (4.17b), and is executed as described in Theorem 4.2.5. More specifically, the theory in Theorem 4.2.5 is in our setting rearranged to the explicit formulas

$$\widetilde{\mathbf{a}}^* \mathbf{b} = -\sum_{j=1}^{k_{\widetilde{a}}} \widetilde{a}_j^* M(0)^{-1} \left( \sum_{\ell=1}^{k_b} M^{(j+\ell-1)}(0) \tfrac{(\ell-1)!}{(j+\ell-1)!} b_\ell \right) \tag{4.26a}$$

$$= -\sum_{j=1}^{k_{\widetilde{a}}} (M(0)^{-*} \widetilde{a}_j)^* \left( \sum_{\ell=1}^{k_b} M^{(j+\ell-1)}(0) \tfrac{(\ell-1)!}{(j+\ell-1)!} b_\ell \right). \tag{4.26b}$$

Similar to formula (4.25), we show how to reduce the number of linear solves in section 4.3.3.

(6)+(7) These orthogonalization steps are comparable to those in (3)+(4). Since the vectors are orthogonalized against the previous vector, one column of zeros is added to allow for the summation.

(15) An important property of this new method is that the computation of the approximate eigenvectors for the solution of (4.1) entails the storage of (only) $k$ vectors of length $n$ for each subspace. To clarify this, recall from section 4.3.1 that from the eigentriplet $(\theta_1^{(k)}, z_1^{(k)}, \widetilde{z}_1^{(k)})$ of $T_k$ we can deduce an approximate eigentriplet $(\theta_1^{(k)}, Q_k z_1^{(k)}, \widetilde{Q}_k \widetilde{z}_1^{(k)})$ for $\mathbf{A}$. The approximate right eigenpair $(\theta_1^{(k)}, Q_k z_1^{(k)})$ approximates thus a right eigenpair of $\mathbf{A}$ that has the form $(\lambda, \mathbf{v})$, where $\mathbf{v} = \left[\frac{\lambda^{j-1}}{(j-1)!}\, x\right]_{j=1}^{\infty}$ (see (4.7)). From this approximate pair of $\mathbf{A}$, we are able to extract an approximate solution to (4.1a). Note that the columns of $Q_k$ represent vectors of type (4.17a) and thus a linear combination of the columns is itself a representation of a vector of this type. Suppose $s_r$ stands for the first length-$n$ block of $Q_k z_1^{(k)}$. Then $s_r$ is an approximation to $x$, the first length-$n$ block of $\mathbf{v}$, and thus, by Theorem 4.2.1 $((\theta_1^{(k)})^{-1}, s_r)$ is an approximate solution to (4.1a).

Similarly, the left eigenpair $(\theta_1^{(k)}, \widetilde{Q}_k \widetilde{z}_1^{(k)})$ approximates a left eigenpair of $\mathbf{A}$ of the form $(\lambda, \mathbf{w})$, with $\mathbf{w} = \sum_{j=1}^{\infty} (\mathbf{S}^T \otimes I)^{j-1} \mathbf{N}^* \lambda^j z$ (see (4.10)). Again, we can deduce an approximate solution to (4.1b) from this approximate pair of $\mathbf{A}$. The columns of $\widetilde{Q}_k$ represent vectors of type (4.17b) such that a linear combination of the columns is itself a representation of a vector of this type. Suppose the first length-$n$ block of $\widetilde{Q}_k \widetilde{z}_1^{(k)}$ is called $s_\ell$. By Theorem 4.2.2 we know that $s_\ell$ is an approximation to $\lambda z = \lambda M(0)^* y$. Hence $((\theta_1^{(k)})^{-1}, \theta_1^{(k)} M(0)^{-*} s_\ell)$ is an approximate solution to (4.1b).

To recover the approximate eigenvectors of (4.1) we do not have to store the entire matrices $Q_k$ and $\widetilde{Q}_k$. As we have just shown, the storage of (only) $k$ vectors of length $n$ for each subspace is sufficient.

**Algorithm 4.2:** Infinite bi-Lanczos

**Input:** Vectors $q_1, \widetilde{q}_1 \in \mathbb{C}^n$, with $\widetilde{q}_1^* M'(0) q_1 = 1$, $P_0 = \widetilde{P}_0 = [\ ]$, $P_1 = [q_1]$, $\widetilde{P}_1 = [\widetilde{q}_1]$, $\gamma_1 = \beta_1 = 0$.

**Output:** Approximate eigentriplets $((\theta_i^{(k)})^{-1}, x_i^{(k)}, y_i^{(k)})$ to nonlinear eigenvalue problem (4.1).

**for** $k = 1, 2, \ldots,$ until convergence

(1) Compute $R_{k+1} := [b_1, \ldots, b_{k+1}] \in \mathbb{C}^{n \times (k+1)}$ with (4.24) where, $k_a = k$, $a_\ell = P_{k,\ell}$ for $\ell = 1, \ldots, k$.

(2) Compute $\widetilde{R}_{k+1} := [\widetilde{b}_1, \ldots, \widetilde{b}_{k+1}] \in \mathbb{C}^{n \times (k+1)}$ with (4.25) where, $k_{\widetilde{a}} = k$, $\widetilde{a}_\ell = \widetilde{P}_{k,\ell}$ for $\ell = 1, \ldots, k$.

(3) $R_{k+1} = R_{k+1} - \gamma_k [P_{k-1}, 0, 0]$

(4) $\widetilde{R}_{k+1} = \widetilde{R}_{k+1} - \bar{\beta}_k [\widetilde{P}_{k-1}, 0, 0]$

(5) Compute $\alpha_k = \widetilde{\mathbf{a}}^* \mathbf{b}$ with (4.26) where $\widetilde{a}_\ell = \widetilde{P}_{k,\ell}$, $\ell = 1, \ldots, k$, and $b_\ell = R_{k+1,\ell}$ for $\ell = 1, \ldots, k+1$ and $k_{\widetilde{a}} = k$ and $k_b = k+1$.

(6) $R_{k+1} = R_{k+1} - \alpha_k [P_k, 0]$

(7) $\widetilde{R}_{k+1} = \widetilde{R}_{k+1} - \bar{\alpha}_k [\widetilde{P}_k, 0]$

(8) Compute $\omega_k = \widetilde{\mathbf{a}}^* \mathbf{b}$ with (4.26) where $\widetilde{a}_\ell = \widetilde{R}_{k+1,\ell}$, $b_\ell = R_{k+1,\ell}$ for $\ell = 1, \ldots, k+1$, where $k_{\widetilde{a}} = k_b = k+1$.

(9) $\beta_{k+1} = |\omega_k|^{1/2}$

(10) $\gamma_{k+1} = \bar{\omega}_k / \beta_{k+1}$

(11) $P_{k+1} = R_{k+1} / \beta_{k+1}$

(12) $\widetilde{P}_{k+1} = \widetilde{R}_{k+1} / \bar{\gamma}_{k+1}$

(13) Compute eigentriplets $(\theta_i^{(k)}, z_i^{(k)}, \widetilde{z}_i^{(k)})$ of $T_k$.

(14) Test for convergence.

**end**

(15) Compute approximate eigenvectors $x_i^{(k)}$ and $y_i^{(k)}$.

### 4.3.3 *Computational representation of the infinite vectors*

The algorithm described in the previous subsection is complete in the sense that it shows how one can carry out the two-sided Lanczos for the infinite matrix $\mathbf{A}$ in finite-dimensional arithmetic. However, it needs several modifications to become a practical algorithm. Most importantly, by inspection of (4.25) and (4.26) we directly conclude that it requires a large number of linear solves corresponding to $M(0)^{-1}$ and $M(0)^{-*}$. With a change of variables we now show how the number of

linear solves per step can be reduced to two linear solves per iteration by a particular representation of $\widetilde{\mathbf{a}}$ and $\widetilde{\mathbf{b}}$.

The choice of representation is motivated by the fact that the computational formulas involving the vectors $\widetilde{a}_j$ appear in combination with a linear solve with $M(0)^{-*}$, in particular in formulas (4.25b) and (4.26b). This property is also naturally expected from the fact that any infinite vector of type (4.17b) can be factorized as

$$\begin{aligned}\widetilde{\mathbf{a}} &= \sum_{j=1}^{k_{\widetilde{a}}} (\mathbf{S}^T \otimes I)^{j-1} \mathbf{N}^* \widetilde{a}_j \\ &= -\sum_{j=1}^{k_{\widetilde{a}}} (\mathbf{S}^T \otimes I)^{j-1} \begin{bmatrix} M'(0) & \frac{1}{2}M^{(2)}(0) & \dots \end{bmatrix}^* M(0)^{-*} \widetilde{a}_j.\end{aligned}$$

Instead of storing the $k_{\widetilde{a}}$ vectors $\widetilde{a}_j$ that represent the infinite vector $\widetilde{\mathbf{a}}$, and storing the $k_{\widetilde{b}}$ vectors $\widetilde{b}_j$ that represent the infinite vector $\widetilde{\mathbf{b}}$, we store the vectors

$$\widetilde{a}_j^{\text{comp}} := M(0)^{-*} \widetilde{a}_j, \text{ for } j = 1, \dots, k_{\widetilde{a}}, \tag{4.27a}$$

$$\widetilde{b}_j^{\text{comp}} := M(0)^{-*} \widetilde{b}_j, \text{ for } j = 1, \dots, k_{\widetilde{b}}. \tag{4.27b}$$

The superscript 'comp' is used to indicate that this vector is the representation which is used in the computation.

Some additional efficiency can be achieved by also modifying the representation of $\mathbf{a}$ and $\mathbf{b}$. Instead of representing these vectors with $a_j$, $j = 1, \dots, k_a$ and $b_j$, $j = 1, \dots, k_b$, we set

$$a_j^{\text{comp}} := (j-1)! a_j, \text{ for } j = 1, \dots, k_{\widetilde{a}}, \tag{4.28a}$$

$$b_j^{\text{comp}} := (j-1)! b_j, \text{ for } j = 1, \dots, k_{\widetilde{b}}. \tag{4.28b}$$

This reduces the number of scalar operations and simplifies the implementation.

The substitutions (4.27) and (4.28) translate the steps of the algorithm as follows. Since the substitution is linear and changes the representation of both $\widetilde{\mathbf{a}}$ and $\mathbf{a}$, the operations associated with steps (1), (2), (5) and (8) need to be modified.

The substitution (4.28) changes the operations associated with the action of $\mathbf{A}$ in step (1). Instead of (4.24) we use

$$b_j^{\text{comp}} = a_{j-1}^{\text{comp}} \text{ for } j = 2, \dots, k_a + 1, \tag{4.29a}$$

$$b_1^{\text{comp}} = -M(0)^{-1} \sum_{j=1}^{k_a} M^{(j)}(0) \tfrac{1}{j!} a_j^{\text{comp}}. \tag{4.29b}$$

The reason for this substitution is that (4.29a) can now be computed without any operations on the vectors, and that (4.29b) is completely analogous to (4.30b) with a complex conjugate transpose.

We need to compute the action of $\mathbf{A}^*$ by using (4.25) in step (2). The substitution corresponding to the representation (4.27) into (4.25) leads to the formulas

$$\widetilde{b}_j^{\text{comp}} = \widetilde{a}_{j-1}^{\text{comp}} \text{ for } j = 2, \dots k_{\widetilde{a}} + 1, \tag{4.30a}$$

$$\widetilde{b}_1^{\text{comp}} = M(0)^{-*}\widetilde{b}_1 = -M(0)^{-*} \sum_{j=1}^{k_{\widetilde{a}}} M^{(j)}(0)^* \tfrac{1}{j!} \widetilde{a}_j^{\text{comp}}. \tag{4.30b}$$

Note that in contrast to (4.25), (4.30) only involves one linear solve.

We need to compute the scalar product of infinite vectors in step (5) and (8). Instead of using (4.26), we can now reformulate formula (4.26) with the new representation as

$$\begin{aligned} \widetilde{\mathbf{a}}^*\mathbf{b} &= -\sum_{j=1}^{k_{\widetilde{a}}} (M(0)^{-*}\widetilde{a}_j)^* \left( \sum_{\ell=1}^{k_b} M^{(j+\ell-1)}(0) \tfrac{(\ell-1)!}{(j+\ell-1)!} b_\ell \right) \\ &= -\sum_{j=1}^{k_{\widetilde{a}}} (\widetilde{a}_j^{\text{comp}})^* \sum_{\ell=1}^{k_b} M^{(j+\ell-1)}(0) \tfrac{1}{(j+\ell-1)!} b_\ell^{\text{comp}}. \end{aligned} \tag{4.31}$$

This formula does not require any linear solve, which should be seen in contrast to (4.26) which requires $k_{\widetilde{a}}$ linear solves. Despite this improvement, we will see in the following section and numerical examples that the computation of the scalar product in (4.31) is often the dominating part of the algorithm.

### 4.3.4 *Complexity considerations and implementation*

The computational resources and problem specific aspects for the algorithm can be summarized as follows. The description below is based on the representation in section 4.3.3. Again our discussion is conducted with references to the steps of the algorithm. We neglect the computation associated with the scalars in steps (9) and (10).

(1)+(2) In the representation of section 4.3.3 we need to evaluate (4.30) and (4.29). The main computational effort of evaluating these for-

mulas consists of first computing a linear combination of derivatives, in the sense that we need to call the functions

$$\text{lincomb}(z_1, \ldots, z_m) = \sum_{i=1}^{m} M^{(i)}(0) z_i, \tag{4.32a}$$

$$\text{lincombstar}(\widetilde{z}_1, \ldots, \widetilde{z}_m) = \sum_{i=1}^{m} M^{(i)}(0)^* \widetilde{z}_i. \tag{4.32b}$$

The output of the functions $\text{lincomb}(\cdot)$ and $\text{lincombstar}(\cdot)$ are used for a linear solve associated with $M(0)$ and $M(0)^*$. Note that $M(0)$ and $M(0)^*$ are not changed throughout the iteration such that for many large and sparse eigenvalue problems efficiency improvements can be achieved by computing an LU-factorization before the iteration starts.

(3)+(4) These steps consist of simple operations on a full matrix of size $n \times k$ and are in general not computationally demanding. The same holds for steps (6)+(7) and (11)+(12) of the algorithm.

(8)+(9) The scalar products are computed with (4.31). Note that one part of that formula is a linear combination of derivatives, such that it can be computed by calling the function defined in (4.32a), i.e., $\text{lincomb}(\cdot)$, $k_{\widetilde{a}}$ times. Since both $k_a$ and $k_{\widetilde{a}}$ increase in every iteration, the double sum in (4.31) accumulates after $k$ steps to a total complexity

$$t_{\text{scalarprod}}(k, n) = \mathcal{O}(k^3 n). \tag{4.33}$$

(13) This step consists of computing an eigentriplet of a $k \times k$ tridiagonal matrix, which in general is not a computationally dominating part of the algorithm.

We conclude that in order to apply our algorithm to a specific problem the user needs to provide a function to solve linear systems corresponding to $M(0)$ and $M(0)^*$ and a procedure to compute linear combinations of derivatives as defined in (4.32). This can be seen in relation to IAR [53] and TIAR [52] where the user needs to provide a function to carry out linear solves corresponding to $M(0)$ and compute linear combinations as in (4.32a).

**Remark 4.3.1 (Scalar product complexity and improvement)** We mention that both the Infinite Arnoldi method (IAR) [53] and the Tensor Infinite Arnoldi method (TIAR) [52] have a complexity (in terms of number of floating point operations) of $\mathcal{O}(k^3 n)$, although generally in practice TIAR is considerably faster than IAR. Due to the scalar product complexity (4.33) our algorithm also has a computational complexity $\mathcal{O}(k^3 n)$. However, it turns out that the scalar product computation can be improved in problem specific cases.

To ease the notation let us collect the vectors $\tilde{a}_j^{\text{comp}}$ for $j = 1, \ldots, k_{\tilde{a}}$ in $\tilde{A} \in \mathbb{R}^{n \times k_{\tilde{a}}}$ and $b_\ell^{\text{comp}}$ for $\ell = 1, \ldots, k_b$ in $B \in \mathbb{R}^{n \times k_b}$ (in correspondence to the notation in Algorithm 2). Moreover, without loss of generality we decompose the NEP as a sum of products of matrices and scalar functions

$$M(\lambda) = M_1 f_1(\lambda) + \cdots + M_p f_p(\lambda),$$

where $f_1, \ldots, f_p$ are analytic functions. Although the assumption is not a restriction of generality, the following approach is only efficient if $p$ is small. This is the case for many NEPs, e.g., those in section 4.4. The scalar product (4.31) is now

$$\begin{aligned}
\widetilde{\mathbf{a}}^* \mathbf{b} &= -\sum_{j=1}^{k_{\tilde{a}}} \sum_{\ell=1}^{k_b} \sum_{k=1}^{p} (\widetilde{a}_j^{\text{comp}})^* M_k f_k^{(j+\ell-1)}(0) \tfrac{1}{(j+\ell-1)!} b_\ell^{\text{comp}} \\
&= -\sum_{j=1}^{k_{\tilde{a}}} \sum_{\ell=1}^{k_b} \sum_{k=1}^{p} (\widetilde{a}_j^{\text{comp}})^* M_k b_\ell^{\text{comp}} \tfrac{1}{(j+\ell-1)!} f_k^{(j+\ell-1)}(0) \\
&= -\sum_{j=1}^{k_{\tilde{a}}} \sum_{\ell=1}^{k_b} \sum_{k=1}^{p} \hat{M}_{k,j,\ell} \tfrac{1}{(j+\ell-1)!} f_k^{(j+\ell-1)}(0), \qquad (4.34)
\end{aligned}$$

where

$$\hat{M}_k := A M_k B, \text{ for } k = 1, \ldots, p. \qquad (4.35)$$

The matrices $\hat{M}_1, \ldots, \hat{M}_p$ can be computed before computing the sum. In this fashion the last line of (4.34) is independent of the size of the problem $n$. Moreover, the sum in (4.34) can be carried out by appropriate matrix vector products. This reformulation of the step changes the accumulated computation time complexity of the scalar product to

$$\tilde{t}_{\text{scalarprod}} = \mathcal{O}(pk^3) + \mathcal{O}(npk^2),$$

under the assumption that $M_k B$ is carried out in $\mathcal{O}(nk_b)$ operations. In comparison to (4.33), this approach is advantageous if $p$ is small and $n$ is large. Moreover, in modern computer architectures matrix-matrix products are more efficient than (non-optimized) double sums, due to more efficient usage of CPU-cache.

## 4.4 NUMERICAL EXPERIMENTS

Our approach is intended for large and sparse problems, and we illustrate the properties of the proposed algorithm by solving two nonlinear problems. To increase reproducibility of our results we have made the MATLAB-codes freely available online[1]; it can redo the simulations of this section.

### 4.4.1 *A second order delay-differential equation*

We start with the illustration of the properties and competitiveness of the algorithm by computing solutions to an artificial large-scale NEP stemming from a second order delay-differential equation,

$$M(\lambda) = -\lambda^2 I + A_0 + e^{-\tau\lambda} A_1, \tag{4.36}$$

where $A_0$ and $A_1$ are randomly generated sparse matrices with normally distributed random entries and $\tau = 1$. Solutions to (4.36) can for instance be used to study the stability of time-delay systems. See [66] for further literature on time-delay systems. For the experiments we choose the matrices to be of dimension $n = 1000$. The total number of iterations is equal to $k = 50$.

Figure 4.1 shows the approximated eigenvalues and distinguishes those converged after $k = 50$ iterations by a circle around them, which are obviously the ones closest to zero. The two-sided approach has the advantage that during the process a condition number estimate is available, enabling the user to define a satisfying convergence criterion. The condition numbers shown in table 2.1 correspond to the converged eigenvalues and can be computed as (cf. [82])

$$\kappa(\lambda, M) := \frac{\left(|\lambda|^2\|I\|_2 + \|A_0\|_2 + |e^{-\lambda}|\|A_1\|_2\right)\|x\|_2\|y\|_2}{|\lambda|\,|y^*(-2\lambda I - e^{-\lambda}A_1)x|},$$

1 The MATLAB codes are online available: `http://www.math.kth.se/~eliasj/src/infbilanczos/`; see also [27].

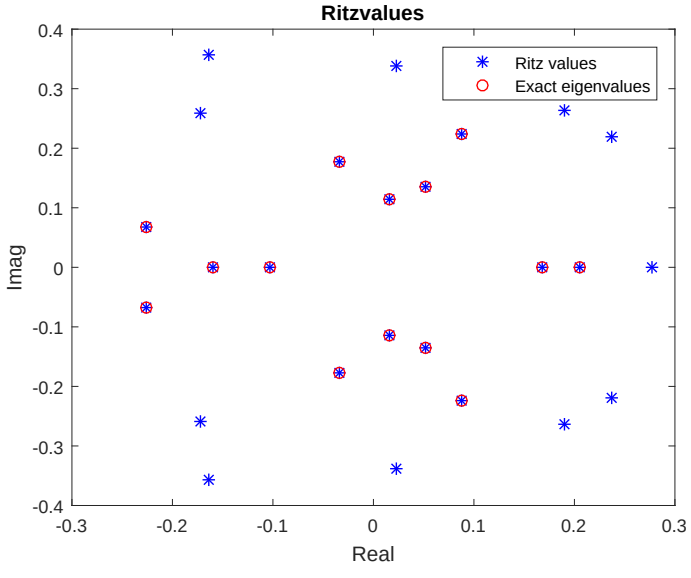



Figure 4.1: Eigenvalue approximations of the infinite bi-Lanczos method applied to problem (4.36). Circles correspond to approximations that have converged after $k = 50$.

| $i$ | $\lvert\theta_i^{(k)}\rvert^{-1}$ | $\kappa((\theta_i^{(k)})^{-1})$ |
|---|---|---|
| 1 | $1.029 \cdot 10^{-1}$ | $1.267 \cdot 10^{3}$ |
| 2 | $1.157 \cdot 10^{-1}$ | $2.510 \cdot 10^{3}$ |
| 3 | $1.157 \cdot 10^{-1}$ | $2.510 \cdot 10^{3}$ |
| 4 | $1.440 \cdot 10^{-1}$ | $1.697 \cdot 10^{3}$ |
| 5 | $1.440 \cdot 10^{-1}$ | $1.697 \cdot 10^{3}$ |
| 6 | $1.593 \cdot 10^{-1}$ | $1.846 \cdot 10^{3}$ |
| 7 | $1.593 \cdot 10^{-1}$ | $1.925 \cdot 10^{3}$ |
| 8 | $1.803 \cdot 10^{-1}$ | $7.315 \cdot 10^{2}$ |
| 9 | $1.803 \cdot 10^{-1}$ | $7.315 \cdot 10^{2}$ |

Table 4.1: The condition numbers for the nine converged eigenvalues closest to zero. The values are computed using the approximate eigentriplets after $k = 50$ iterations.

where one can notice that $-2\lambda I - e^{-\lambda} A_1 = M'(\lambda)$. We also compare the infinite bi-Lanczos method to the infinite Arnoldi method (IAR) as

presented in [53]. Figure 4.2 shows for both methods the error in the eigenvalues versus the number of iterations. Figure 4.3 depicts the error of both methods versus the computing time in seconds. The results strongly depend on the computing environment. We have run our simulations on several environments, including changing computer and MATLAB-version, and observed a similar behavior in most simulations. For the infinite bi-Lanczos method the Ritz values converge in less iterations, and in general the first couple of eigenvalues converged in less CPU-time, but with such a small margin that the simulation is not conclusive regarding CPU-time.

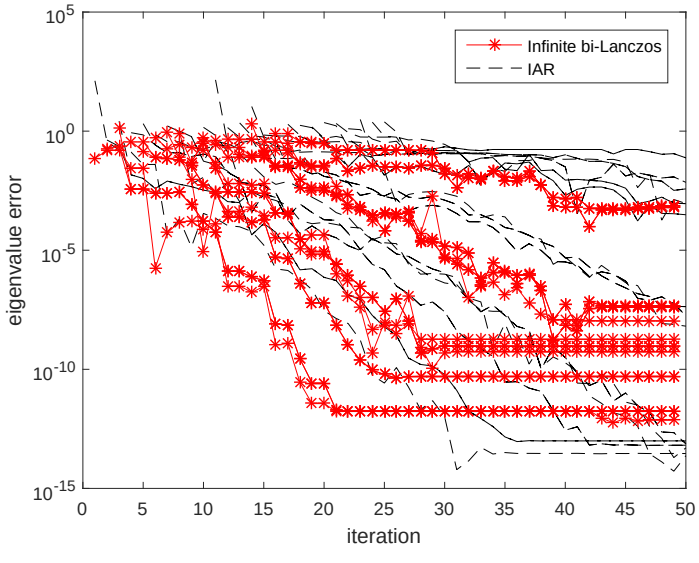



Figure 4.2: Convergence diagram, eigenvalue error against the iterations.

The faster convergence of the infinite bi-Lanczos method can be explained by the two subspaces that are built in the infinite bi-Lanczos method. In fact, with respect to one multiplication with $\mathbf{A}$ per iteration of IAR, infinite bi-Lanczos contains per iteration a multiplication with both $\mathbf{A}$ and $\mathbf{A}^*$. Because of the short recurrences the computing time of infinite bi-Lanczos can be kept decently low (and may even outperform IAR), as shown in figure 4.3. In contrast to IAR, the bi-Lanczos procedure exhibits a stagnation in convergence. This is due to finite precision arithmetic. It is known that short-recurrence methods (such as the Lanczos method) are in general considered to be more sensi-

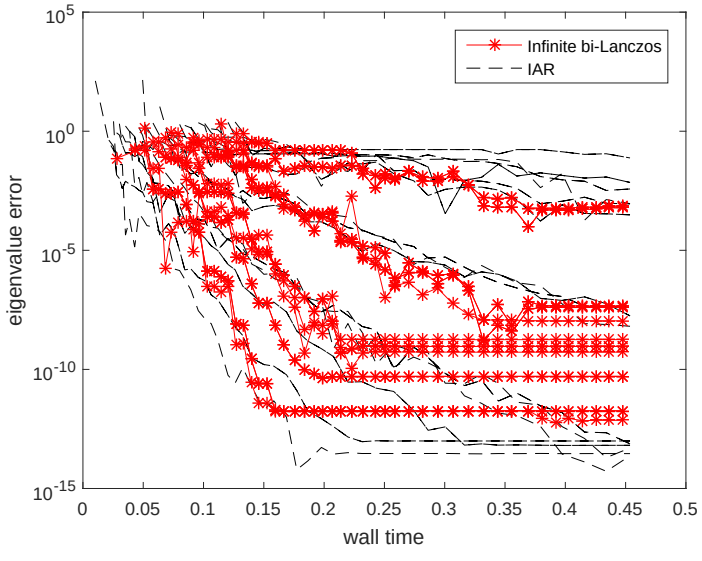



Figure 4.3: Convergence diagram, eigenvalue error against the computation time (s).

tive to round-off errors than methods that use orthogonalization with respect to all vectors in the basis matrix (such as the Arnoldi method).

Our implementation is based on using the computation of the scalar product as described in remark 4.3.1. To illustrate the advantage of this optimization technique, we present a comparison of the computing times in figure 4.4. The exploitation of the technique described in remark 4.3.1 has a clear gain in terms of computing time.

To characterize the impact of round-off errors in the proposed algorithm, we have also carried out simulations with an implementation in which some of the operations are done in high-precision arithmetic. We consider problem (4.36) with randomly generated $3 \times 3$-matrices and $\tau = 0.5$. We have run our double precision and high precision implementations with exactly the same initial values. The convergence behavior is given in figure 4.5. The figure shows that the stagnation is due to round-off errors, since the high precision version of our algorithm continues to converge also after 30 iterations. The stagnation that is due to round-off errors is consistent with the fact that Lanczos-type methods can suffer from a loss of orthogonality, which often happens after the first eigenvalue has converged.

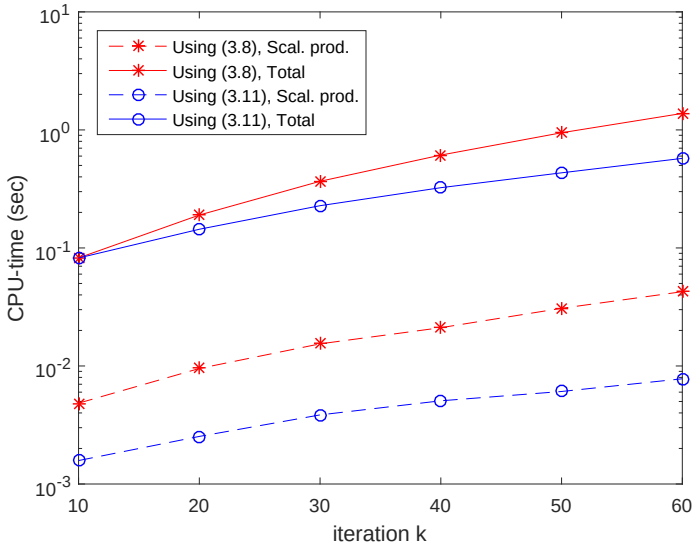



Figure 4.4: Computing time for the two different procedures to compute the scalar product. The figure shows the accumulated CPU-time spent for $k$ iterations in the algorithm (total) and in the computation of the scalar product (scal. prod.).

### 4.4.2 *A benchmark problem representing an electromagnetic cavity*

We now consider the NEP presented in [61] which is also available in the collection [8, problem "gun"]. The problem stems from the modelling of an electromagnetic cavity in an accelerator device. The discretization of Maxwell's equation with certain boundary conditions leads to the NEP

$$M(\lambda) = A_0 - \lambda A_1 + i\sqrt{\lambda}A_2 + i\sqrt{\lambda - \sigma_2^2}A_3, \tag{4.37}$$

where $\sigma_2 = 108.8774$. Before applying a numerical method, the problem is usually shifted and scaled. We set $\lambda = \lambda_0 + \alpha\hat{\lambda}$ where $\lambda_0 = 300^2$ and $\alpha = (300^2 - 200^2) \cdot 10$. This problem has been solved with a number of methods [3, 5, 37, 53]. We use it as a benchmark problem to illustrate the generality and validity of our approach.

The convergence of the infinite bi-Lanczos method and IAR is visualized in figures 4.6 and 4.7. Unlike the previous example, the convergence of infinite bi-Lanczos does not stagnate. For this particular choice of shift, infinite bi-Lanczos is more efficient than IAR regarding

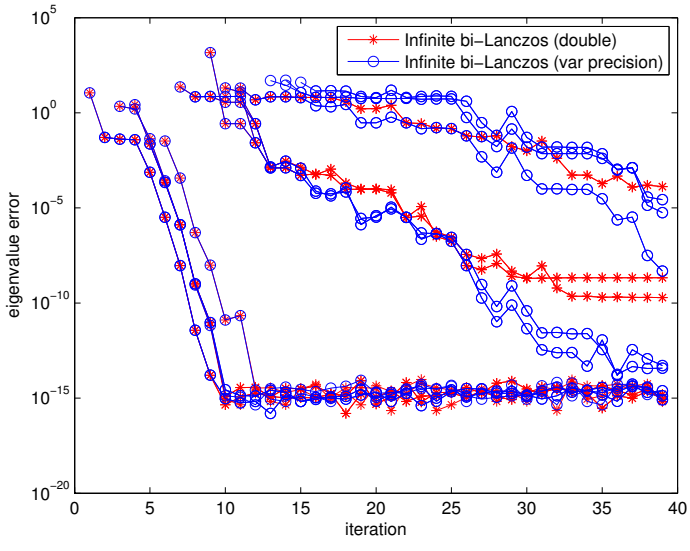



Figure 4.5: Comparison of infinite bi-Lanczos with different arithmetic: double and high-precision.

the number of iterations, but slightly less efficient regarding the computing time. We carried out experiments for several parameter choices, and found nothing conclusive regarding which method is more efficient in general. Hence, the infinite bi-Lanczos method is favorable if also left eigenvectors are of interest. On the other hand, one could replace IAR by the mathematically equivalent method TIAR, which is often faster. In fact, concerning the CPU-time, in the same computing environment, 20 steps of TIAR require 0.48 seconds. Note however that TIAR uses a compact tensor representation of the basis, and therefore belongs to a slightly different class of methods, as mentioned in the previous section. See [5, 58, 87] for related methods based on compact representations in various settings.

## 4.5 FINAL CONSIDERATIONS

We have proposed a new two-sided Lanczos method for the nonlinear eigenvalue problem. The method works implicitly with matrices and vectors with infinite size. The new way of representing left type of infinite vectors is crucial to frame the two-sided method. We intend to

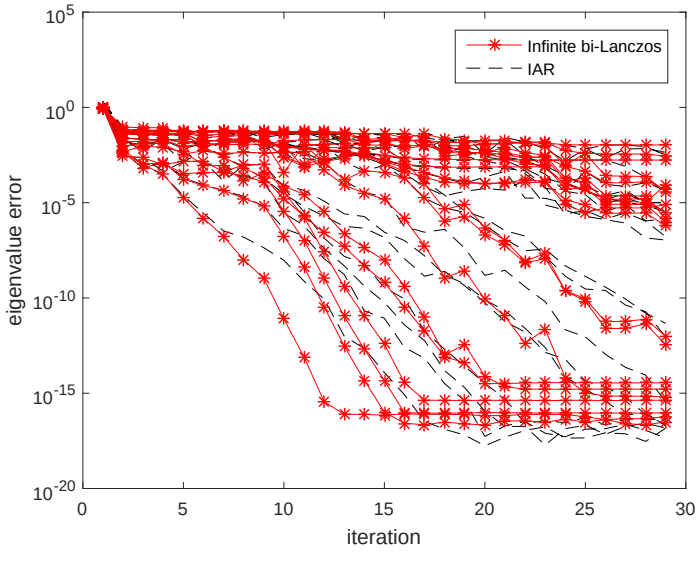



Figure 4.6: Convergence diagram, eigenvalue error against the iterations.

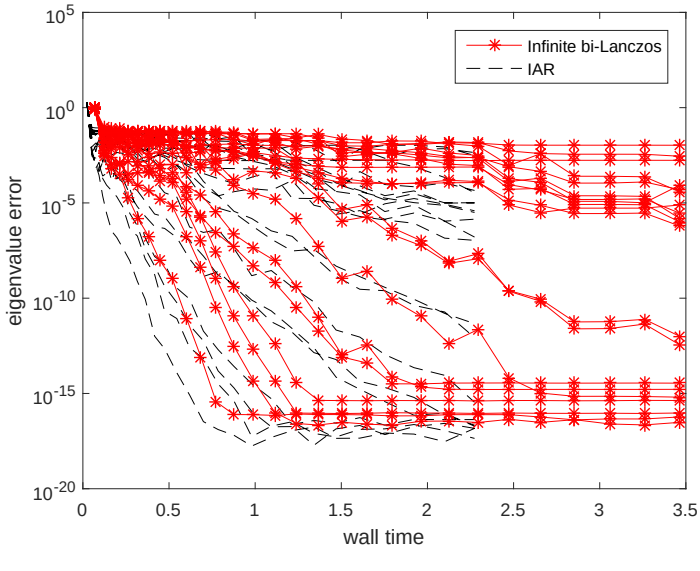



Figure 4.7: Convergence diagram, eigenvalue error against the computation time (s).

make the code adaptive, as the condition numbers which become available as the iterations proceed may be used to define a satisfying con-

vergence criterion. We have seen that infinite bi-Lanczos can converge faster per iteration than the infinite Arnoldi method (IAR), which could be expected because in general two-sided methods have faster convergence (per iteration), and moreover, since infinite bi-Lanczos uses a low-term recurrence it has a lower orthogonalization cost per iteration than IAR.

Several enhancements of IAR have been presented in the literature. Some of the developments appear to be extendable to this Lanczos-setting, e.g., the tensor representation [52] and the restart techniques [51, 64]. Moreover, for certain problems it is known that a different version of IAR is more efficient, which can only be characterized with a continuous operator (as in [53]). These types of adaptations are however somewhat involved due to the fact that in the setting of infinite bi-Lanczos the left eigenvectors and the vectors representing the left Krylov-subspace are more complicated than the right eigenvectors and subspace.

Although our approach is rigorously derived from an equivalence with the standard two-sided Lanczos method, we have not yet developed any convergence theory. Convergence theory for standard two-sided Lanczos for linear eigenvalue problems is already quite involved and its specialization to the infinite dimensional method is certainly beyond the scope of the presented work. This also holds for other theory and procedures specifically designed for the standard method, such as the various possibilities to detect or overcome breakdowns (as mentioned in section 4.3.1), and approaches to control the loss of biorthogonality (see [15]). We have carried out some simulations for problems with a high eigenvalue condition number and the results indicate that the method is less successful for these problems.

# 5

# CONCLUSION

Krylov methods are widely used to approximate solutions to a large variety of problems such as eigenvalue problems or linear systems of equations. Large-scale problems often lead to special requirements and standard Krylov subspaces may be insufficient for these demands. For example, standard Krylov subspaces are usually unsuitable for finding a good approximation to the smallest singular value of a large-scale matrix $A$. Or, if one needs to build the Krylov subspace $\mathcal{K}(A, b)$, but neither the matrix $A$ is available nor an exact matrix-vector multiplication with the matrix $A$ can be computed. Krylov methods are usually applied to linear eigenvalue problems, but can they be applied to nonlinear eigenvalue problems as well? In this work we have focused on the three above-mentioned problems. We have developed the following three Krylov methods to overcome various issues for large-scale matrices:

- The **extended Lanczos bidiagonalization** method, which yields (probabilistic) bounds for the largest and smallest singular values of a matrix, and therefore provides bounds for the condition number of the matrix.
- The **inexact Lanczos bidiagonalization** method, which approximates the largest singular values and corresponding singular vectors of a matrix function, and hence enables that the 2-norm of a matrix function can be estimated.
- The **infinite bi-Lanczos** method, which is developed to approximate the eigenvalues and both the left and right eigenvectors of a nonlinear eigenvalue problem.

The first two methods are designed for the approximation of singular values, whereas the third method approximates eigenvalues. All three methods are *Lanczos* methods, and indeed, they all make use of certain symmetry structures leading to advantageous short recurrences.

## 5.1 OVERVIEW OF THE CHAPTERS

Below we give a short overview of the conclusions from the three research chapters of this dissertation. We would like to mention again that the MATLAB-codes for the numerical experiments of chapters 2 and 4 can be found at `www.win.tue.nl/~hochsten/eigenvaluetools/` and `www.math.kth.se/~eliasj/src/infbilanczos/`, respectively. The codes used for the experiments of chapter 3 will be provided upon request.

### 5.1.1 *Overview of chapter 2*

In chapter 2 we proposed a new extended Lanczos bidiagonalization method that simultaneously approximates both the smallest and the largest singular value of $A$. A lower bound of good quality for the condition number $\kappa(A)$ is obtained. Furthermore, the method yields probabilistic upper bounds for $\kappa(A)$ that are true upper bounds with a user-chosen probability $1 - 2\varepsilon$. Given a user-selected $\varepsilon$ and desired ratio $\kappa_{\text{up}}(A)/\kappa_{\text{low}}(A) < \zeta$, the results show that generally the number of iterations $k$ is fairly small, even for $\zeta = 1.1$. We reveal the special structure of the tridiagonal matrices that are created by the extended Lanczos bidiagonalization method, and we show that only 3 vectors of storage are required. The method is based on the assumption that the computation of an LU-factorization is affordable in a reasonable amount of time.

### 5.1.2 *Overview of chapter 3*

Chapter 3 deals with the approximation of the leading singular values and corresponding left and right singular vectors of a matrix function $f(A)$, for some large square matrix $A$ and a sufficiently regular function $f$ so that $f(A)$ is well defined. In particular we were interested in the approximation of $\|f(A)\|$, where $\|\cdot\|$ is the matrix norm induced by the Euclidean vector norm. We assumed neither $f(A)$ nor products of the type $f(A)v$ are available exactly and therefore we introduced an inexact Lanczos bidiagonalization procedure. The inexactness is associated with the inaccuracy of the operations $f(A)v$ and $f(A)^*v$. The inexact method yields a non-Hermitian perturbation of the original Hermitian

matrix. Furthermore, the lack of a true residual demands for particular outer and inner stopping criteria. The numerical results demonstrate that the two inner iterations for the approximation of $f(A)v$ and $f(A)^*u$ may be very time and memory consuming. We have shown that the relaxed strategy alleviates this problem whenever accurate approximations are required.

### 5.1.3 *Overview of chapter 4*

In chapter 4 we proposed a two-sided Lanczos method for the nonlinear eigenvalue problem (NEP). The infinite bi-Lanczos method provides approximations to both the right and left eigenvectors of the eigenvalues of interest. The method implicitly works with matrices and vectors with infinite size. Particular (starting) vectors are used that allow all computations to be carried out efficiently with finite matrices and vectors. We specifically introduced a new way to represent infinite vectors that span the subspace corresponding to the conjugate transpose operation for approximating the left eigenvectors, which is crucial to frame the two-sided method. Furthermore, we showed that also in this infinite-dimensional interpretation the short recurrences inherent to the Lanczos procedure offer an efficient algorithm regarding both the computational cost and the storage. The numerical results show that infinite bi-Lanczos can have faster convergence and a lower orthogonalization cost per iteration than the infinite Arnoldi method (IAR).

## 5.2 DISCUSSION AND OUTLOOK

The results that have been presented in this dissertation are based on certain assumptions and choices to clearly demarcate the research questions. Furthermore, the discussed methods have shown their shortcomings but more importantly we have discovered their strength, usefulness and elegance. Below we give possible directions of further research that arise from the previous chapters.

- The theory developed in chapter 2 is based on the assumption that $A \in \mathbb{R}^{n\times n}$. It is interesting to investigate which parts of the results can be adjusted to (full rank) rectangular matrices $A \in \mathbb{R}^{m\times n}$, $m > n$, or complex matrices $A \in \mathbb{C}^{m\times n}$, $m \geq n$. Note that for rectangular matrices operations with the pseudo-inverse and

its consequences require attention. Regarding complex matrices, the current theory for probabilistic upper bounds uses a random starting vector on the unit sphere in $\mathbb{R}^n$. Extending the results for a starting vector on the unit sphere in $\mathbb{C}^n$ is non-trivial.

- The inverse operations in the extended Lanczos bidiagonalization treated in chapter 2 are in general costly. One could generalize the method, e.g., as in [50], and build a Krylov subspace $\mathcal{K}^{k,\ell}$, for $k \neq \ell$, in contrast to the extended subspace $\mathcal{K}^{k,k}$. Based on the problem properties a user then has the choice to diminish the number of inverse operations.

- The extended Lanczos bidiagonalization presented in chapter 2 assumes it is possible to compute the LU-decomposition of the matrix $A$, but for large matrices this may not be the case. An investigation whether inexact matrix-vector products could be used instead entails combining results from chapter 2 and chapter 3, where some forethought leads to the following two issues. First of all, the basis vectors that are created during an inexact Lanczos procedure will no longer span a Krylov subspace. As a consequence, the polynomials that arise naturally in the exact Lanczos procedure, are not defined in the inexact procedure, making the current theory to obtain probabilistic bounds inadequate for the inexact execution. Secondly, the Ritz values that result from an inexact process will approximate singular values of a perturbed matrix. Whether these values are upper or lower bounds for the true singular values of $A$ will not be clear, in contrast to the method presented in chapter 2.

- In the introduction of chapter 2 we mention that the extended Lanczos bidiagonalization will not yield a ratio between the probabilistic upper bound and lower bound given a fixed $k$ and $\varepsilon$, as is provided in [16] or [59]. In other words, there is no prior knowledge about the number of iterations that have to be performed to obtain bounds of a specific quality. We expect this analysis to be involved, and therefore, before investigating this theory for the extended Lanczos bidiagonalization, one has to take hold of the analysis for other methods, such as extended Lanczos or Lanczos bidiagonalization.

- The inexact Lanczos bidiagonalization presented in chapter 3 to approximate the 2-norm of a matrix function may also be used to estimate the norm of other matrix objects. One could think of the geometric mean [9], or the *derivatives* of matrix functions, such as the Fréchet derivative of the matrix exponential or of other functions [40]. Furthermore, in the solution of time-dependent differential equations, the evaluation of $\|f(tA)\|$, for $t > 0$ and a sufficiently regular function $f$ such that $f(A)$ is well defined, is of great interest to monitor the presence of transient behaviors for $A$ nonnormal.

- Although the approach of chapter 4 is rigorously derived from an equivalence with the standard two-sided Lanczos method, no convergence theory is developed yet for the infinite case. Convergence theory for standard two-sided Lanczos (for linear eigenvalue problems) is involved, but nevertheless its specialization could be an interesting extension of the work presented here. This also holds for other theory and procedures specifically designed for the standard method, such as the various possibilities to detect or overcome breakdowns (as mentioned in section 4.3.1), and approaches to control the loss of biorthogonality (see [15]).

- Regarding chapter 4, several enhancements of the infinite Arnoldi method have been presented in the literature. Some of the developments appear to be extendable to the setting of infinite bi-Lanczos, e.g., the tensor representation [52] and the restart techniques [51, 64]. These types of adaptations are however more complex due to the fact that theory for the left eigenvectors and the vectors representing the left Krylov-subspace in the infinite bi-Lanczos method is more complicated than for the right eigenvectors and subspace.

- The theory presented in chapter 4 for representing vectors that characterize the left Krylov subspace is now used for the infinite bi-Lanczos method. It may be worthwhile to explore these techniques also for other types of Krylov methods that make use of the transpose operations of a matrix, for instance Lanczos bidiagonalization.

- In chapter 4 we used the Taylor series of (4.1a), but another choice of expansion can be made to fit the (convergence) properties of the problem in a better way.

# BIBLIOGRAPHY


[1] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. A. van der Vorst. *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*. SIAM, 2000.

[2] B. J. C. Baxter. "Norm estimates for inverses of Toeplitz distance matrices." *Journal of Approximation Theory* 79.2 (1994), pp. 222–242.

[3] R. van Beeumen. "Rational Krylov Methods for Nonlinear Eigenvalue Problems." PhD thesis. KU Leuven, 2015.

[4] R. van Beeumen, K. Meerbergen, and W. Michiels. "A rational Krylov method based on Hermite interpolation for nonlinear eigenvalue problems." *SIAM Journal on Scientific Computing* 35.1 (2013), A327–A350.

[5] R. van Beeumen, K. Meerbergen, and W. Michiels. "Compact rational Krylov methods for nonlinear eigenvalue problems." *SIAM Journal on Scientific Computing* 36.2 (2015), pp. 820–838.

[6] M. Benzi, P. Boito, and N. Razouk. "Decay properties of spectral projectors with applications to electronic structure." *SIAM Review* 55.1 (2013), pp. 3–64.

[7] T. Betcke and H. Voss. "A Jacobi–Davidson type projection method for nonlinear eigenvalue problems." *Future Generation Computer Systems* 20.3 (2004), pp. 363–372.

[8] T. Betcke, N. J. Higham, V. Mehrmann, C. Schröder, and F. Tisseur. "NLEVP: A collection of nonlinear eigenvalue problems." *ACM Transactions on Mathematical Software (TOMS)* 39.2 (2013), pp. 1–28.

[9] R. Bhatia and P. Grover. "Norm inequalities related to the matrix geometric mean." *Linear Algebra and its Applications* 437.2 (2012), pp. 726–733.

[10] A. Bouras and V. Frayssé. "Inexact matrix-vector products in Krylov methods for solving linear systems: A relaxation strategy." *SIAM Journal on Matrix Analysis and Applications* 26.3 (2005), pp. 660–678.

[11] D. Choi. "Estimating Norms of Matrix Functions Using Numerical Ranges." PhD thesis. University of Washington, 2013.

[12] M. Crouzeix. "Numerical range and functional calculus in Hilbert space." *Journal of Functional Analysis* 244.2 (2007), pp. 668–690.

[13] J. K. Cullum and R. A. Willoughby. *Lanczos Algorithms for Large Symmetric Eigenvalue Computations: Vol. 1: Theory*. SIAM, 2002.

[14] T. A. Davis and Y. Hu. "The University of Florida sparse matrix collection." *ACM Transactions on Mathematical Software (TOMS)* 38.1 (2011), pp. 1–28.

[15] D. Day. "An efficient implementation of the nonsymmetric Lanczos algorithm." *SIAM Journal on Matrix Analysis and Applications* 18.3 (1997), pp. 566–589.

[16] J. D. Dixon. "Estimating extremal eigenvalues and condition numbers of matrices." *SIAM Journal on Numerical Analysis* 20.4 (1983), pp. 812–814.

[17] J. L. M. van Dorsselaer, M. E. Hochstenbach, and H. A. van der Vorst. "Computing probabilistic bounds for extreme eigenvalues of symmetric matrices with the Lanczos method." *SIAM Journal on Matrix Analysis and Applications* 22.3 (2000), pp. 837–852.

[18] V. Druskin and L. Knizhnerman. "Extended Krylov subspaces: approximation of the matrix square root and related functions." *SIAM Journal on Matrix Analysis and Applications* 19.3 (1998), pp. 755–771.

[19] V. Druskin, L. Knizhnerman, and M. Zaslavsky. "Solution of large scale evolutionary problems using rational Krylov subspaces with optimized shifts." *SIAM Journal on Scientific Computing* 31.5 (2009), pp. 3760–3780.

[20] C. Effenberger. "Robust Solution Methods for Nonlinear Eigenvalue Problems." PhD thesis. EPF Lausanne, 2013.

[21] M. Eiermann and O. G. Ernst. "A restarted Krylov subspace method for the evaluation of matrix functions." *SIAM Journal on Numerical Analysis* 44.6 (2006), pp. 2481–2504.

[22] J. van den Eshof and M. Hochbruck. "Preconditioning Lanczos approximations to the matrix exponential." *SIAM Journal on Scientific Computing* 27.4 (2006), pp. 1438–1457.

[23] R. W. Freund, M. H. Gutknecht, and N. M. Nachtigal. "An implementation of the look-ahead Lanczos algorithm for non-Hermitian matrices." *SIAM Journal on Scientific Computing* 14.1 (1993), pp. 137–158.

[24] A. Frommer, S. Güttel, and M. Schweitzer. "Convergence of restarted Krylov subspace methods for Stieltjes functions of matrices." *SIAM Journal on Matrix Analysis and Applications* 35.4 (2014), pp. 1602–1624.

[25] A. Frommer and V. Simoncini. "Matrix functions." In: *Model Order Reduction: Theory, Research Aspects and Applications*. Springer, 2008, pp. 275–303.

[26] S. W. Gaaf and M. E. Hochstenbach. "Probabilistic bounds for the matrix condition number with extended Lanczos bidiagonalization." *SIAM Journal on Scientific Computing* 37.5 (2015), S581–S601.

[27] S. W. Gaaf and E. Jarlebring. "The infinite bi-Lanczos method for nonlinear eigenvalue problems." *To appear in SIAM Journal on Scientific Computing* (2017).

[28] S. W. Gaaf and V. Simoncini. "Approximating the leading singular triplets of a large matrix function." *Applied Numerical Mathematics* 113 (2017), pp. 26–43.

[29] C. W. Gear. "Numerical solution of ordinary differential equations: is there anything left to do?" *SIAM Review* 23.1 (1981), pp. 10–24.

[30] M. I. Gil. "Perturbations of functions of diagonalizable matrices." *Electronic Journal of Linear Algebra* 20.1 (2010), pp. 303–313.

[31] G. H. Golub and W. Kahan. "Calculating the singular values and pseudo-inverse of a matrix." *Journal of the Society for Industrial and Applied Mathematics, Series B: Numerical Analysis* 2.2 (1965), pp. 205–224.

[32] G. H. Golub and C. F. van Loan. *Matrix Computations*. Vol. 4. The Johns Hopkins University Press, 2012.

[33] S. Güttel and F. Tisseur. "The nonlinear eigenvalue problem." *Acta Numerica* 26 (2017), 1–94.

[34] T. Gudmundsson, C. S. Kenney, and A. J. Laub. "Small-sample statistical estimates for matrix norms." *SIAM Journal on Matrix Analysis and Applications* 16.3 (1995), pp. 776–792.

[35] K. E. Gustafson and D. K. M. Rao. *Numerical Range: The Field of Values of Linear Operators and Matrices*. Springer, 1997.

[36] S. Güttel. "Rational Krylov approximation of matrix functions: Numerical methods and optimal pole selection." *GAMM-Mitteilungen* 36.1 (2013), pp. 8–31.

[37] S. Güttel, R. van Beeumen, K. Meerbergen, and W. Michiels. "NLEIGS: A class of fully rational Krylov methods for nonlinear eigenvalue problems." *SIAM Journal on Scientific Computing* 36.6 (2014), A2842–A2864.

[38] N. Halko, P. G. Martinsson, and J. A. Tropp. "Finding structures with randomness: Probabilistic algorithms for constructing approximate matrix decompositions." *SIAM Review* 53.2 (2011), pp. 217–288.

[39] N. J. Higham. *Functions of Matrices: Theory and Computation*. SIAM, 2008.

[40] N. J. Higham and S. D. Relton. "Estimating the condition number of the Fréchet derivative of a matrix function." *SIAM Journal on Scientific Computing* 36.6 (2014), pp. C617–C634.

[41] N. J. Higham and F. Tisseur. "A block algorithm for matrix 1-norm estimation, with an application to 1-norm pseudospectra." *SIAM Journal on Matrix Analysis and Applications* 21.4 (2000), pp. 1185–1201.

[42] M. Hochbruck and C. Lubich. "Exponential integrators for quantum-classical molecular dynamics." *BIT Numerical Mathematics* 39.4 (1999), pp. 620–645.

[43] M. Hochbruck and A. Ostermann. "Exponential integrators." *Acta Numerica* 19 (2010), pp. 209–286.

[44] M. E. Hochstenbach. "A Jacobi–Davidson type SVD method." *SIAM Journal on Scientific Computing* 23.2 (2001), pp. 606–628.

[45] M. E. Hochstenbach. "Harmonic and refined extraction methods for the singular value problem, with applications in least squares problems." *BIT Numerical Mathematics* 44.4 (2004), pp. 721–754.

[46] M. E. Hochstenbach. "Probabilistic upper bounds for the matrix two-norm." *Journal of Scientific Computing* 57.3 (2013), pp. 464–476.

[47] R. A. Horn and C. R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, 1991.

[48] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 2012.

[49] C. Jagels and L. Reichel. "The extended Krylov subspace method and orthogonal Laurent polynomials." *Linear Algebra and its Applications* 431 (2009), pp. 441–458.

[50] C. Jagels and L. Reichel. "Recursion relations for the extended Krylov subspace method." *Linear Algebra and its Applications* 434 (2011), pp. 1716–1732.

[51] E. Jarlebring, K. Meerbergen, and W. Michiels. "Computing a partial Schur factorization of nonlinear eigenvalue problems using the infinite Arnoldi method." *SIAM Journal on Matrix Analysis and Applications* 35.2 (2014), pp. 411–436.

[52] E. Jarlebring, G. Mele, and O. Runborg. "The waveguide eigenvalue problem and the tensor infinite Arnoldi method." *SIAM Journal on Scientific Computing* 39.3 (2017), A1062–A1088.

[53] E. Jarlebring, W. Michiels, and K. Meerbergen. "A linear eigenvalue algorithm for the nonlinear eigenvalue problem." *Numerische Mathematik* 122.1 (2012), pp. 169–195.

[54] E. Jarlebring and E. H. Rubensson. *On the condition number and perturbation of matrix functions for Hermitian matrices*. `arXiv:1206.1762v1`. Preprint. June 2012.

[55] E. R. Jessup and I. C. F. Ipsen. "Improving the accuracy of inverse iteration." *SIAM Journal on Scientific and Statistical Computing* 13.2 (1992), pp. 550–572.

[56] L. Knizhnerman and V. Simoncini. "A new investigation of the extended Krylov subspace method for matrix function evaluations." *Numerical Linear Algebra with Applications* 17.4 (2010), pp. 615–638.

[57] D. Kressner. "A block Newton method for nonlinear eigenvalue problems." *Numerische Mathematik* 114.2 (2009), pp. 355–372.

[58] D. Kressner and J. Roman. "Memory-efficient Arnoldi algorithms for linearizations of matrix polynomials in Chebyshev basis." *Numerical Linear Algebra with Applications* 21.4 (2014), pp. 569–588.

[59] J. Kuczyński and H. Woźniakowski. "Estimating the largest eigenvalue by the power and Lanczos algorithms with a random start." *SIAM Journal on Matrix Analysis and Applications* 13.4 (1992), pp. 1094–1122.

[60] R. M. Larsen. *Lanczos bidiagonalization with partial reorthogonalization*. DAIMI Report Series–537 27. Department of Computer Science, Aarhus University, 1998.

[61] B.-S. Liao, Z. Bai, L.-Q. Lee, and K. Ko. "Nonlinear Rayleigh-Ritz iterative method for solving large scale nonlinear eigenvalue problems." *Taiwanese Journal of Mathematics* 14.3 (2010), pp. 869–883.

[62] J. Liesen and Z. Strakos. *Krylov Subspace Methods. Principles and Analysis*. Oxford University Press, 2012.

[63] V. Mehrmann and H. Voss. "Nonlinear eigenvalue problems: A challenge for modern eigenvalue methods." *GAMM-Mitteilungen* 27.2 (2004), pp. 121–152.

[64] G. Mele and E. Jarlebring. "On restarting the tensor infinite Arnoldi method." *BIT Numerical Mathematics* (2017), pp. 1–30.

[65] E. Mengi and M. L. Overton. "Algorithms for the computation of the pseudospectral radius and the numerical radius of a matrix." *IMA Journal of Numerical Analysis* 25.4 (2005), pp. 648–669.

[66] W. Michiels and S.-I. Niculescu. *Stability and Stabilization of Time-Delay Systems: An Eigenvalue-Based Approach*. Advances in Design and Control 12. SIAM, 2007.

[67] B. N. Parlett. *The Symmetric Eigenvalue Problem*. SIAM, 1998.

[68] E. H. Rubensson. "Controlling errors in recursive Fermi-Dirac operator expansions with applications in electronic structure theory." *SIAM Journal on Scientific Computing* 34.1 (2012), B1–B23.

[69] A. Ruhe. "Algorithms for the nonlinear eigenvalue problem." *SIAM Journal on Numerical Analysis* 10 (1973), pp. 674–689.

[70] Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, 2003.

[71] B. Schmitt. "Norm bounds for rational matrix functions." *Numerische Mathematik* 42.3 (1983), pp. 379–389.

[72] K. Schreiber. "Nonlinear Eigenvalue Problems: Newton-type Methods and Nonlinear Rayleigh Functionals." PhD thesis. TU Berlin, 2008.

[73] V. Simoncini. "Variable accuracy of matrix-vector products in projection methods for eigencomputation." *SIAM Journal on Numerical Analysis* 43.3 (2005), pp. 1155–1174.

[74] V. Simoncini. "A new iterative method for solving large-scale Lyapunov matrix equations." *SIAM Journal on Scientific Computing* 29.3 (2007), pp. 1268–1288.

[75] V. Simoncini and D. B. Szyld. "Theory of inexact Krylov subspace methods and applications to scientific computing." *SIAM Journal on Scientific Computing* 25.2 (2003), pp. 454–477.

[76] G. W. Stewart and J.-G. Sun. *Matrix Perturbation Theory (Computer Science and Scientific Computing)*. Academic Press Boston, 1990.

[77] Z. Strakoš and P. Tichý. "On error estimation in the conjugate gradient method and why it works in finite precision computations." *Electronic Transactions on Numerical Analysis* 13.8 (2002), pp. 56–80.

[78] D. B. Szyld and F. Xue. "Local convergence analysis of several inexact Newton-type algorithms for general nonlinear eigenvalue problems." *Numerische Mathematik* 123 (2012), pp. 333–362.

[79] D. B. Szyld and F. Xue. "Preconditioned eigensolvers for large-scale nonlinear Hermitian eigenproblems with variational characterizations. I. Extreme eigenvalues." *Mathematics of Computation* 85.302 (2016), pp. 2887–2918.

[80] *The Matrix Market*. A repository for test matrices, `http://math.nist.gov/MatrixMarket`.

[81] *The University of Florida Sparse Matrix Collection*. A repository for test matrices, `www.cise.ufl.edu/research/sparse/matrices/`.

[82] F. Tisseur. "Backward error and condition of polynomial eigenvalue problems." *Linear Algebra and its Applications* 309 (2000), pp. 339–361.

[83] L. N. Trefethen and M. Embree. *Spectra and Pseudospectra: The Behavior of Nonnormal Matrices and Operators*. Princeton University Press, 2005.

[84] H. Voss. "An Arnoldi method for nonlinear eigenvalue problems." *BIT Numerical Mathematics* 44.2 (2004), pp. 387–401.

[85] H. Voss. "Nonlinear eigenvalue problems." In: *Handbook of Linear Algebra, Second Edition*. Discrete Mathematics and Its Applications 164. Chapman and Hall/CRC, 2013.

[86] G. A. Watson. "Computing the numerical radius." *Linear Algebra and its Applications* 234 (1996), pp. 163–172.

[87] Y. Zhang and Y. Su. "A memory-efficient model order reduction for time-delay systems." *BIT Numerical Mathematics* 53 (2013), pp. 1047–1073.

# SUMMARY


## ADVANCED LANCZOS METHODS FOR LARGE-SCALE MATRIX PROBLEMS

In many scientific and engineering fields, problems of a linear algebraic nature arise, such as eigenvalue and singular value problems. Frequently, these problems involve sparse matrices of large dimension. Since direct methods may not be feasible for large-scale problems, iterative methods are developed. Subspace methods, in particular Krylov methods, form a widely used subclass of computationally efficient iterative methods. Krylov methods project the problem onto a subspace of low dimension. In this way, the computing time and memory usage are reduced, since an approximate solution is provided by solving a small-scale problem. This dissertation presents three novel Lanczos methods, specific types of Krylov subspace methods, to compute and approximate important quantities in large-scale problems. Essential properties of the new techniques are revealed.

The condition number of large matrices is an important quantity for which adequate approximations are needed, for example when solving large-scale linear systems. A new extended Lanczos bidiagonalization method is developed, which turns out to be ideal for the simultaneous approximation of both the smallest and largest singular value of a matrix, providing a lower bound for the condition number. Recently, probabilistic techniques have become popular, and the method provides, besides the lower bound, also a probabilistic upper bound for the condition number, exploiting the fact that the initial vector is randomly chosen.

Another topic that has been examined involves the approximation of the leading singular triplets, and in particular the 2-norm, of a large-scale matrix function. This norm arises for instance in the solution to stiff initial-value problems and is also used in the analysis of iterative processes. The research treats inexact iterative solution methods and the approximation of matrix functions, two subjects in the center of

numerical linear algebra these days. A new inexact Lanczos bidiagonalization procedure is introduced, where the inexactness is related to the inaccuracy of the operations involving matrix functions. The lack of a true residual requires particular outer and inner stopping criteria. These are devised in this dissertation.

Whereas in the past Krylov methods were used only for linear problems, nowadays they are being invoked more and more for other problems, for instance for nonlinear eigenvalue problems such as the delay eigenvalue problem. A third research topic in this dissertation is the development of a new two-sided Lanczos method for nonlinear eigenvalue problems. To work with Krylov methods the problem has to be linearized, leading to infinite matrices and vectors. Compared to the standard two-sided method, the new method implicitly works with matrices and vectors with infinite size. Particular (starting) vectors are used, enabling the method to carry out all computations efficiently with finite matrices and vectors.

# CURRICULUM VITAE

Sarah Gaaf was born on 29 March 1987 in Leeuwarden, the Netherlands. After finishing the gymnasium in 2005 at the Stedelijk Dalton College in Zutphen, the Netherlands, she studied for one year Italian Language and Culture at the Università per Stranieri in Perugia, Italy. Subsequently she obtained a Bachelor (2010) and Master (2012) degree in Mathematics at the Universiteit van Amsterdam, the Netherlands.

In January 2013 she started a PhD project at the Technische Universiteit Eindhoven in the Netherlands under the supervision of dr. M.E. Hochstenbach and prof.dr.ir. B. Koren. Her research was part of the project *Innovative methods for large matrix problems* funded by the Netherlands Organisation for Scientific Research, and the results are presented in this dissertation. She won a poster prize (2014) at the Woudschoten Conference of the Dutch-Flemish Research Community Scientific Computing, and she won twice (2014 and 2016) the student paper prize of the Copper Mountain Conference on Iterative Methods in the United States of America.

From October 2017 she will be employed at the Universiteit Utrecht, the Netherlands, as a junior assistant professor.

# LIST OF PUBLICATIONS

## REFEREED JOURNAL PAPERS

1) S.W. Gaaf and E. Jarlebring, *The infinite bi-Lanczos method for nonlinear eigenvalue problems*, to appear in SIAM Journal on Scientific Computing.

2) S.W. Gaaf and V. Simoncini, *Approximating the leading singular triplets of a large matrix function*, Applied Numerical Mathematics 113 (2017): 26-43.

3) S.W. Gaaf and M.E. Hochstenbach, *Probabilistic bounds for the matrix condition number with extended Lanczos bidiagonalization*, SIAM Journal on Scientific Computing 37.5 (2015): S581-S601.

## NON-REFEREED PROCEEDINGS PAPERS

4) A. Bose, S. Fanzon, S.W. Gaaf, T.G. de Jong, P. Madhikar, K. Marova, K. Mitra, B. Mortier, V. Ogesa, J. Richter, G. Uraltsev, R. De Maio, *Analysis of shrinkage of micro-cellular foam rubber for foot-ware*, Proceedings of the 124th European Study Group Mathematics with Industry, 2016.

5) M. Bosmans, S. Gaaf, C. Groothede, R. Gupta, M. Regis, M. Tsardakas, A. Vromans, K. Vuik, *Nonlinear Cochlear Dynamics*, Proceedings of the 98th European Study Group Mathematics with Industry, 2014.

6) C. Bud, J. Evers, J. Frank, S. Gaaf, R. Hoogwater, D. Lahaye, C. Meerman, E. Siero, T. van Zalen, *Effective water storage as flood protection; the Rijnstrangen study case*, Proceedings of the 90th European Study Group Mathematics with Industry, 2013.

# ACKNOWLEDGMENTS

It would be easier to thank all of you with this one sentence. But I owe you the effort of expressing my gratitude to each of you separately.

As this dissertation is the final product of the research I have done at the TU/e, I first of all need to thank Jan Brandts. I would never have started the trajectory to obtain a PhD at the TU/e without his incentive to apply for the position. It turned out to be an optimal choice and I am thankful for this. Furthermore, without the funding of NWO this position would never have been created.

I would like to thank both of my supervisors, Michiel and Barry, to guide me, stimulate me, and let me discover and experience many aspects of working in academia with the most possible freedom. Michiel, thank you for giving me exclusively constructive advice, along with examples from your own experiences. I will not forget how often our meetings ended in the corridor while you were telling some anecdote. Barry, I can describe you best as a tower of strength. Your door was always open for advice, and I left your room with new positive energy.

In the past years I got the opportunity to travel to many countries, for conferences as well as for study groups and research visits. I would like to thank Valeria Simoncini, Elias Jarlebring and Daniel Kressner to kindly welcome me to their institutes.

Valeria, you have inspired and stimulated me to become a better researcher, always passionately searching for improvements and good ideas. I am grateful you have taken me under your wings for three months, while I could learn from your ways to do research and your ways to structure your work.

Elias, thank you for all the time you dedicated to me during the ten day visit in which we worked together intensively. I remember the large number of notes that we produced every day while discovering new theory and implementing the results right away, always with a lot of care.

Daniel, it has been an inspiring week in Lausanne. I have had interesting discussions both with you and with all the members of your group, who have given me a warm welcome.

The trips to conferences and study groups were almost as holidays, thanks to my colleagues that have become friends. Tania, Erna, Davide, Scott, and all the others, I cherish the memories of the days and evenings spent together. I hope to continue seeing you in the future, whatever career paths we pursue.

I would like to thank Valeria Simoncini, Elias Jarlebring, Martin van Gijzen, Harald van Brummelen, and Wil Schilders for the willingness to take part in my doctoral committee and for carefully reading my dissertation.

I would like to thank all present and former CASA members. I have only warm feelings and many nice memories when looking back at all lunches, colloquia, morning sessions, discussions, coffee breaks, outings, dinners and other festivities together.

In particular I would like to thank Ian, Sangye, Xiulei and René for all the conversations and discussions we have had in our office. Apart from mathematical problems we debated a lot about global (political, ethical, philosophical) issues which I enjoyed enormously.

A special thanks to Enna, for all the help you have given me around my daily work. Organizing the Sinterklaas afternoons together has been a lot of fun.

I would like to thank all the former and present members of the PhD-council to make the council a success. I have enjoyed this experience very much and I am proud of everything we have accomplished in a short period of time. I admire every council member for the effort he/she enthusiastically put into this new adventure.

I also would like to extend my appreciation to the board of the Department and everyone working closely together with them. I have learned a lot from my experiences as chair of the PhD council, which has been facilitated by all your help and willingness. It was wonderful to see there was so much support for our ideas and undertakings.

In the last months of writing my dissertation I have spent many hours in the library of the CWI. Alessandro, you have alleviated this period by being there for distracting conversations during lunch. Thank you for the 24/7 LaTeX-linea and for becoming a friend.

Jos and Apo, my first friends from mathematics, we have discovered the first theorems together, and slowly grew up to adults with a critical

mind. Thank you for our strong friendship and for the feeling that you will always be there for me.

To all my friends, to those I have cooked for many evenings, to those I drank beer with, to those I watched football with, and went on vacation with, thank you for all these wonderful memories. Without this I could never have focused on the mathematics during the day.

To my family, Carel, Frédérique, Mecheline and Cas, sometimes it is a pity I cannot share the theory with you, but mostly it is the best way to divide work and private life. Thank you for being proud of me without exactly knowing what I was doing all day. I always have felt helped and stimulated in everything I do.

To Alessandro, you make me relativize with endless humor every day. To Minerva, you showed me that life has so many hidden aspects. While writing these last two sentences, tears of love for both of you are rolling over my cheeks.