Krylov methods for nonlinear eigenvalue problems and matrix equations

GIAMPAOLO MELE

Doctoral Thesis
Stockholm, Sweden, 2020

KTH School of Engineering Sciences
TRITA-SCI-FOU 2019:59
SE-100 44 Stockholm
ISBN 978-91-7873-392-7
SWEDEN

Akademisk avhandling som med tillstånd av Kungliga Tekniska Högskolan framlägges till offentlig granskning för avläggande av teknologie doktorsexamen tisdag 11 februari 2020 kl 10:00 i sal F3, Kungliga Tekniska Högskolan, Lindstedtsvägen 26, Stockholm.

Tryck: Universitetsservice US AB

Abstract

Nonlinear eigenvalue problems (NEPs) arise in many fields of science and engineering. Such problems are often defined by large matrices, which have specific structures, such as being sparse, low-rank, etc. Like the linear eigenvalue problem, the eigenvector appears in a linear form, whereas the eigenvalue appears in a nonlinear form. This feature allows for an extension of several methods, which are originally derived for the linear eigenvalue problem, to the nonlinear case. Among these methods, Krylov algorithms have been successfully extended in various ways. These methods are designed to take advantage of the matrix structures mentioned above. In this thesis, we present two Krylov-based methods for solving NEPs: the tensor infinite Arnoldi (TIAR), with its restarting variant, and infinite Lanczos (ILAN). We illustrate the flexibility of TIAR by adapting it for solving a NEP which comes from the study of waves propagating in periodic mediums.

Despite the fact that Krylov methods are, in a sense, globally convergent, the convergence to the targeted eigenvalues, in certain cases, may be slow. When an accurate solution is required, the obtained approximations are refined with methods which have higher convergence order, e.g., Newton-like methods, which are also analyzed in this thesis. In the context of eigenvalue computation, the framework used to analyse Newton methods can be combined with the Keldysh theorem in order to better characterize the convergence factor. We also show that several well-established methods, such as *residual inverse iteration* and *Ruhe's method of successive linear problems*, belong to the class of Newton-like methods. In this spirit, we derive a new quasi-Newton method, which is, in terms of convergence properties, equivalent to *residual inverse iteration*, but does not require the solution of a nonlinear system per iteration.

The mentioned methods are implemented in NEP-PACK, which is a registered Julia package for NEPs that we develop. This package consists of: many state-of-the-art, but also well-established, methods for solving NEPs, a vast problem collection, and types and structures to efficiently represent and do computations with NEPs.

Many problems in control theory, and many discretized partial differential equations, can be efficiently solved if formulated as matrix equations. Moreover, matrix equations arise in a very large variety of areas as intrinsic problems. In our framework, for certain applications, solving matrix equations is a part of the process of solving a NEP. In this thesis we derive a preconditioning technique which is applicable to linear systems which can be formulate as generalized Sylvester equation. More precisely, we assume that the matrix equation can be formulated as the sum of a Sylvester operator and another term which can be low-rank approximated. Such linear systems arise, e.g., when solving certain NEPs which come from wave propagation problems. We also derive an algorithm, which consists of applying a Krylov method directly to the the matrix equation rather then to the vectorized linear system, that exploits certain structures in the matrix coefficients.

Sammanfattning

Icke-linjära egenvärdesproblem, förkortat NEP från engelskans nonlinear eigenvalue problem, uppstår inom många områden inom vetenskap och teknik. Sådana problem definieras ofta av stora matriser med specifika strukturer, såsom gleshet, låg rang osv. Liksom det i linjära egenvärdesproblemet är beroendet på egenvektorn linjärt, medan beroendet på egenvärdet är icke-linjärt. Denna egenskap möjliggör en utvidgning av flera metoder, som ursprungligen härleds för det linjära egenvärdesproblemet, till det icke-linjära fallet. Bland dessa metoder har Krylov-metoder framgångsrikt vidareutvecklats på olika sätt. Dessa metoder är utformade för att dra fördel av de ovan nämnda matrisstrukturerna. I den här avhandlingen presenterar vi två Krylov-baserade metoder för att lösa NEP: tensor Infinite Arnoldi (TIAR), med en variant som möjliggör omstart, och Infinite Lanczos (ILAN). Vi illustrerar flexibiliteten i TIAR genom att anpassa den till att lösa en NEP som kommer från studien av vågor som sprider sig i periodiska medier.

Även om Krylov-metoder på sätt och vis är globalt konvergenta, kan konvergensen till de önskade egenvärdena i vissa fall vara långsam. När en noggrann lösning erfordras kan de erhållna approximationerna förfinas med metoder som har högre konvergensordning, t.ex. Newton-liknande metoder, som också analyseras i denna avhandling. I detta sammanhang kan ramverket som används för att analysera Newton-metoder kombineras med Keldysh sats för att bättre karakterisera konvergensfaktorn. Vi visar också att flera väletablerade metoder, såsom residual inversiteration och Ruhes metod för successiva linjära problem, tillhör klassen Newton-liknande metoder. I denna anda erhåller vi en ny quasi-Newton metod som motsvarar residual inversiteration, när det gäller konvergensegenskaper, men inte kräver lösning av en olinjär ekvation per iteration.

De nämnda metoderna är implementerade i NEP-PACK, som är ett registrerat Julia-paket för NEP som vi har utvecklat. Detta paket består av många nyutvecklade samt väletablerade metoder för att lösa NEP, en stor problemsamling samt typer och strukturer för att effektivt representera och göra beräkningar med NEP.

Många problem inom styrteori, samt många diskretiserade partiella differentialekvationer, kan lösas effektivt om de formuleras som matrisekvationer. Dessutom uppstår matrisekvationer som delproblem i ett mycket stort antal områden. I vårt ramverk, är lösning av matrisekvationer en del av processen för att lösa en NEP i vissa tillämpningar. I denna avhandling härleder vi en förkonditioneringsteknik som är tillämplig på vissa linjära system vilka kan formuleras som en generaliserad Sylvesterekvation. Mer exakt antar vi att matrisekvationen kan skrivas som summan av en Sylvesteroperator och en annan term som kan approximeras med en operator med låg rang. Sådana linjära system uppstår, t.ex., vid lösning av vissa NEP som kommer från vågutbredningsproblem. Vi presenterar också en algoritm, som består av att tillämpa en Krylov-metod direkt på matrisekvationen snarare än på det vektoriserade linjära systemet, vilken utnyttjar vissa strukturer i matriskoefficienterna.

Preface

This is a *thesis by publication*, also known as a *compilation thesis*, namely it consists of two parts. Part I is an introduction and a summary of the manuscripts produced during the PhD. This part is written for a wider audience, which does not necessarily consists only of experts in the field. In contrast to this, Part II constitutes the scientific contributions of the PhD work. It contains all the produced manuscripts in the form of scientific publications and preprints. The author contributions to the various manuscripts and projects are explicitly described at the end of Part I. Part II includes the following manuscripts:

Paper I
The waveguide eigenvalue problem and the tensor infinite Arnoldi method
E. Jarlebring, G. Mele, O. Runborg
SIAM J. Sci. Comput. 2017

Paper II
Sylvester-based preconditioning for the waveguide eigenvalue problem
E. Ringh, G. Mele, E. Jarlebring and J. Karlsson
Linear Algebra Appl., 2018

Paper III
On restarting the tensor infinite Arnoldi method
G. Mele and E. Jarlebring
BIT, 2018

Paper IV
Disguised and new quasi-Newton methods for nonlinear eigenvalue problems
E. Jarlebring, A. Koskela, G. Mele
Numer. Algorithms, 2018

Paper V
Krylov methods for low-rank commuting generalized Sylvester equations
E. Jarlebring, G. Mele, D. Palitta, E. Ringh
Numer. Linear Algebra Appl., 2018

Paper VI
The infinite Lanczos method for symmetric nonlinear eigenvalue problems
G. Mele
Submitted, 2019
Preprint arXiv:1812.07557

Contents

Acknowledgements

I wish to thank my supervisor Elias Jarlebring for his continuous help and support. He did not only teach me a lot of mathematics, but also how to be professional and an independent researcher. I also thank my office mate, colleague, but more importantly friend, Emil Ringh. I really enjoyed our discussions (scientific and not) and working together. I am very grateful I had the opportunity to work with many people with different backgrounds. I want to express my gratitude to all my co-authors (in alphabetic order): Antti Koskela, Davide Palitta, Elias Jarlebring, Emil Ringh, Johan Karlsson, Max Bennedich, Olof Runborg and Parikshit Upadhyaya. Everyone taught me something new and made me grow academically and scientifically. The quality of this thesis increased with the help of Anna Brooms, Emil Ringh and Elias Jarlebring, who proofread and provided feedbacks and suggestions. I enjoyed working in the very friendly environment of KTH and being surrounded by very nice colleagues and friends. I wish to thank all of you for the endless lunches, the fikas, the dart games, for letting me practice the Swedish language and many more things. I wish to thank my parents and my sister for always supporting and motivating me. I wish to thank my wife Anastasia Kruchinina. Her love and support helped me in many situations. Finally, I thank my son Björn who motivates me to be a better person everyday.

Part I:
Introductory chapters

Abbreviations

NEP	nonlinear eigenvalue problem
PEP	polynomial eigenvalue problem
QEP	quadratic eigenvalue problem
REP	rational eigenvalue problem
WEP	waveguide eigenvalue problem
SPMF	sum of products of matrices and functions
IAR	infinite Arnoldi
TIAR	tensor infinite Arnoldi
ILAN	infinite Lanczos
PDE	partial differential equation
DDE	delay differential equation
DtN-map	Dirichlet-to-Neumann map
FD	finite difference
FEM	finite element method
SMW	Sherman-Morrison-Woodbury

Mathematical notation

Standard mathematical notation is used: $\mathbb{C}$ denotes the set of complex numbers. Vectors, matrices and tensors with complex coefficients are denoted by $\mathbb{C}^n$, $\mathbb{C}^{n\times m}$ and $\mathbb{C}^{n\times m\times p}$. The set of real numbers is denoted by $\mathbb{R}$ and analogous notation is used for vectors, matrices and tensors with real coefficients. Capital letters are used to denote matrices, whereas lower case letters denote vectors, numbers or tensors. Given a matrix $V \in \mathbb{C}^{n\times m}$, we denote by v_j its j-th column and by range(V) the linear space generated by these vectors. We also consider infinite dimensional matrices and vectors. More precisely, these are matrices which are infinite dimensional in two directions, i.e., $A = [a_{i,j}]_{i,j=1}^{\infty}$ and matrices which are infinite dimensional in one direction, e.g., with a finite number of columns but infinite number or rows. Infinite dimensional vectors are clearly a special case with only one column. We consider block partitions of infinite dimensional matrices. We partition the matrix A in blocks of size $n \times n$ and denote by $A_{i,j} \in \mathbb{C}^{n\times n}$ the (i, j)-block. Similarly, given an infinite dimensional matrix V with a finite number of columns, we partition the matrix in blocks of size $n \times 1$ and denote by $v_{i,j} \in \mathbb{C}^n$ the vector obtained by extracting the block i from the columns j. See Figure A for an illustration of such block structures.

$$A = \begin{array}{|c|c|c|c|} \hline A_{1,1} & A_{1,2} & A_{1,3} & \cdots \\ \hline A_{2,1} & A_{2,2} & A_{2,3} & \cdots \\ \hline A_{3,1} & A_{3,2} & A_{3,3} & \cdots \\ \hline \vdots & \vdots & \vdots & \ddots \\ \hline \end{array} \qquad V = \begin{array}{|c|c|c|c|c|} \hline v_{1,1} & v_{1,2} & v_{1,3} & \cdots & v_{1,m} \\ \hline v_{2,1} & v_{2,2} & v_{2,3} & \cdots & v_{2,m} \\ \hline v_{3,1} & v_{3,2} & v_{3,3} & \cdots & v_{3,m} \\ \hline \vdots & \vdots & \vdots & \vdots & \vdots \\ \hline \end{array}$$

Figure A: Illustration of the block structure.

Chapter 1

Introduction

Large scale nonlinear eigenvalue problems (NEPs) and matrix equations can be efficiently solved with a specific class of projection methods, named Krylov methods. In this chapter we review the basics of such methods.

Large scale problems arise in many applications. In the linear algebra context, such problems are often defined by large matrices that have specific structures, such as being sparse, low-rank, semiseparable, Toeplitz, etc. Several of the classical methods, which are general and do not exploit the structure of such matrices, cannot be applied to these problems due to memory and computation time limitations. However, the solutions to these problems can be approximated with iterative methods that are designed to take advantage of the matrix structures, e.g., they exploit the fact that matrix-vector products, with such structured matrices, are fast to compute. A class of iterative methods applicable to large scale problems are the *projection methods*. They are based on the following idea. The problem is projected into a smaller subspace, a so called *projection space*, which contains approximations to the wanted solutions. The projected problem, which has small size, is then solved. The solutions to the projected problem are lifted and provide approximations to the solutions of the original problem. This process is illustrated in Figure 1.1.

We now recall the *Arnoldi method*, which is a well-established projection method for approximating certain eigenvalues of large and structured matrices. Given the matrix $A \in \mathbb{C}^{n\times n}$ and the vector $v \in \mathbb{C}^n$ (randomly selected), we consider the projection space, which is referred to as *Krylov subspace*, defined as

$$\mathcal{K}_m(A, v) := \text{span}\left(v, Av, A^2v, \ldots, A^{m-1}v\right). \tag{1.1}$$

The Arnoldi method is summarized in Algorithm 1 and graphically illustrated in

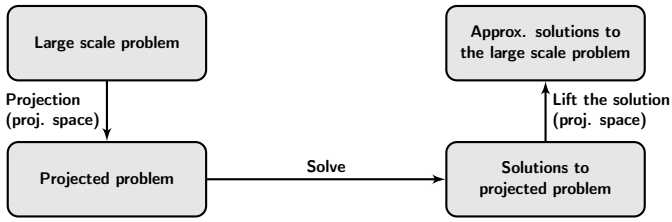

Figure 1.1: Flow chart of the projection methods framework.

Figure 1.2. The output matrices fulfill the following relation, which is referred to as an *Arnoldi factorization*:

$$AV_m = V_{m+1}\underline{H}_m. \tag{1.2}$$

The columns of the matrix V_m form an orthogonal basis to the Krylov space (1.1). The matrix H_m, obtained by removing the last row of $\underline{H}_m$, defines the projected problem. It corresponds to the projection of A into the Krylov space, i.e., $H_m = V_m^T A V_m$. The projected problem is then $H_m z = \lambda z$ and the eigenpair approximations of the matrix A are given by lifting the solutions to the projected problem as $(\lambda, V_m z)$. The eigenpair approximations obtained with this process are named *Ritz pairs*. In particular, the eigenvalue approximations are referred to as *Ritz values* and the eigenvector approximations as *Ritz vectors*. In general, the Krylov space (1.1) contains good approximations to the eigenvectors associated to the outermost eigenvalues of the matrix A. However, in some cases, a bad choice of the starting vector v may result in a Krylov space which will not have the desired properties.

We illustrate these ideas with the following application. We consider the matrix $A \in \mathbb{C}^{200\times 200}$ obtained by linearizing a damped mass spring system. See [29, Section 4.2] and Example 2.2.2. This is a benchmark problem of mechanical vibra-

Algorithm 1: Arnoldi method

input : Matrix $A \in \mathbb{C}^{n\times n}$, starting vector $v \in \mathbb{C}^n$ and nr. of iter. m
output: Matrices: V_{m+1} and $\underline{H}_m$
Set $v_1 = v/\|v\|$ and H_0 empty matrix

for $k = 1, 2, \ldots, m$ **do**
  $w = Av_k$
1 $h = V_k^H w$ and $w_\perp = w - V_k h$
2 $\beta = \|w_\perp\|$ and $v_{k+1} = w_\perp/\beta$
3 $\underline{H}_k = \begin{pmatrix} H_{k-1} & h \\ 0 & \beta \end{pmatrix}$
end

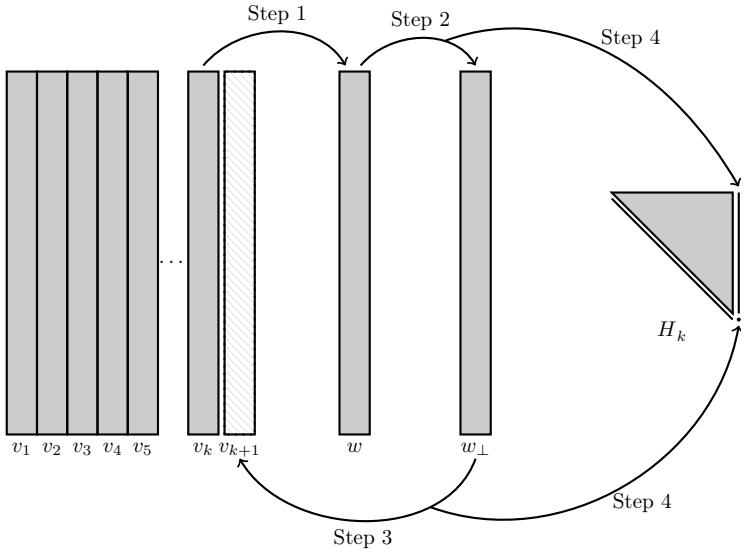

Figure 1.2: Graphical illustration of Algorithm 1. The rectangles on the left represent the columns of the matrix V_k and on the right the matrix $\underline{H}_k$ is represented. The rectangle filled with oblique line pattern is the next vector of the orthogonal basis.

tions. In Figure 1.3 is illustrated the spectrum of the matrix A and the eigenvalue approximations obtained after 20 iterations of Algorithm 1. We can clearly see that the Ritz values approximate the outermost eigenvalues of the matrix A.

In terms of computation time, the dominating part of the Arnoldi method is often the computation of the orthogonal basis V_m. Therefore, this approach becomes computationally demanding if m is too large and the targeted eigenvalues are not well approximated after m iterations. An appropriate restart of the algorithm can resolve these issues in many situations. In this thesis we consider two techniques: implicit and explicit restart. The implicit restart consists of shortening the Arnoldi factorization by transforming (1.2) into another Arnoldi factorization $A\tilde{V}_p = \tilde{V}_{p+1}\underline{\tilde{H}}_p$ with $p < m$. The new Arnoldi factorization involves different matrices and preserves the Ritz values of interest. The explicit restart consists of computing a proper starting vector v, as a linear combination of the columns of V_m, such that the Ritz values converge faster to the targeted eigenvalues.

The Arnoldi method is extendable to the generalized eigenvalue problem $Ax = \lambda Bx$, where $A, B \in \mathbb{C}^{n\times n}$ and B is nonsingular. More precisely, this extension consists of applying Algorithm 1 to the matrix $B^{-1}A$, which corresponds to using $\mathcal{K}_m(B^{-1}A, v)$ as projection space. If the eigenvalues of interest are located close to a target point $\sigma \in \mathbb{C}$, the shifted-and-inverted Krylov space $\mathcal{K}_m((A-\sigma B)^{-1}B, v)$ is employed. The eigenvalue approximations are computed as $\sigma + 1/\lambda_j$ where $\{\lambda_j\}_{j=1}^m$

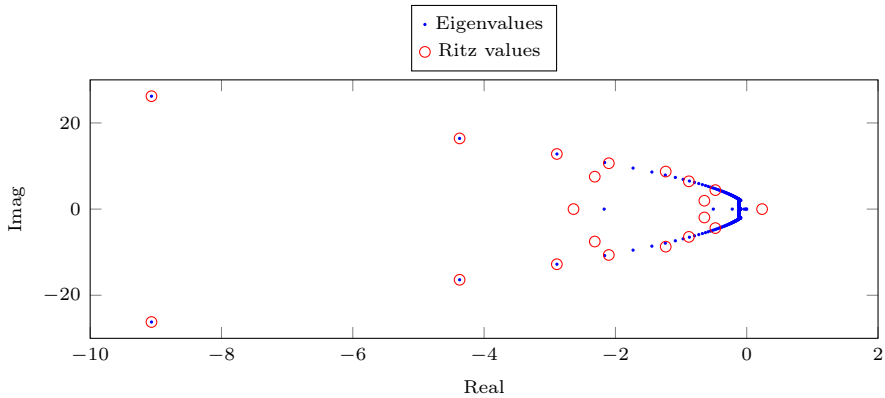

Figure 1.3: Eigenvalues and Ritz values, obtained after 20 *iterations of Algorithm 1, of the linearization of the damped mass spring system in Example 2.2.2, cf. [29, Section 4.2, eq (4.4)], with* $n = 200$ *masses. The parameters are* $m_j = j/200$, $k_j = 1/j$ *and* $b_j = 1/10$ *for* $j = 1, \ldots, 200$. *The outermost eigenvalues are well approximated.*

are the Ritz values. In the thesis we use, as projection space, also the following variations of Krylov spaces:

$E\mathcal{K}_m(A, v)$	$:= \mathcal{K}_m(A, v) + K_m(A^{-1}, v)$	extended Krylov space
$\mathcal{K}_m^{\square}(A, V)$	$:= \mathcal{K}_m(A, v_1) + \cdots + \mathcal{K}_m(A, v_k)$	block Krylov space
$E\mathcal{K}_m^{\square}(A, V)$	$:= E\mathcal{K}_m(A, v_1) + \cdots + E\mathcal{K}_m(A, v_k)$	extended block Krylov space

where $V \in \mathbb{C}^{n\times k}$. We refer to the projection methods that employ Krylov subspaces as *Krylov methods*. The Krylov methods framework is extendable to the NEP as presented in the next chapter.

Krylov methods are applicable to a large class of problems. We now show how these methods can be used to approximate the solutions to certain matrix equations. We consider the Sylvester equation with a specific structure in the matrix coefficients. The problem is formulated as follows: given $A, B \in \mathbb{C}^{n\times n}$ negative definite and structured, e.g., sparse, and $C_1, C_2 \in \mathbb{C}^{n\times r}$, with $r \ll n$, compute $X \in \mathbb{C}^{n\times n}$ such that

$$AX + XB^T = C_1 C_2^T. \tag{1.3}$$

The main idea is to assume that the solution can be approximated by a low-rank matrix $\tilde{X} = LR^T$ with $\text{Range}(L) \subseteq E\mathcal{K}_m^{\square}(A, C_1)$ and $\text{Range}(R) \subseteq E\mathcal{K}_m^{\square}(B, C_2)$. Let V_m and W_m be the orthogonal basis of the spaces $E\mathcal{K}_m^{\square}(A, C_1)$ and $E\mathcal{K}_m^{\square}(B, C_2)$, which we use as projection spaces. A way to compute a low rank approximation of the solution to (1.3) consists of imposing Galerkin orthogonality conditions. These conditions give the following projected problem: $\tilde{A}Z + Z\tilde{B}^T = \tilde{C}_1\tilde{C}_2^T$ where

$\tilde{A} := V_m^T A W_m, \tilde{B} := V_m^T B W_m, \tilde{C}_1 := V_m^T C_1$ and $\tilde{C}_2 := W_m^T C_2$ are the projected matrices. The approximation of the solution to (1.3) is then obtained by lifting the solution to the projected problem as $\tilde{X} := V_m Z_1 (W_m Z_2)^T$ where Z_1, Z_2 are the factors of the LU-factorization of Z. This method is referred to as *extended Krylov method*. For an overview of Krylov methods and matrix equations see [59, 84].

Even if Krylov methods are, in a sense, globally convergent, slow convergence may occur in certain applications. Moreover, solution approximations with high accuracy may be required. In such cases, methods with high convergence order are used. However, such methods are, in general, only locally convergent, i.e., an approximation to the solution has to be available. Therefore these approaches are in practice combined with globally convergent methods. In this thesis we consider Newton-type methods. These methods are used to compute zeros of functions, or equivalently to solve nonlinear systems of equations. Let $F : \mathbb{C}^n \to \mathbb{C}^n$ be the given function, and let $v_0 \in \mathbb{C}^n$ be an approximation to the solution of interest. Then, the Newton method generates a sequence of approximations of the wanted solution as

$$v_{k+1} = v_k - J_F(v_k)^{-1} F(v_k), \qquad k = 1, 2, \ldots,$$

where J_F denotes the Jacobian matrix of F. A class of methods derived in this framework consists of approximating the term $J_F(v_k)^{-1} F(v_k)$. These methods are called *quasi-Newton* methods or *inexact Newton-like* methods. A well known method that fits in this class is the Broyden's method, which is based on approximating the Jacobian matrix by means of a finite differences (FD) approach. This method is often used when the Jacobian matrix is not available. Another approach consists of keeping the Jacobian matrix constant during the iterations such that, $J_F(x_k)^{-1} F(x_k) \approx J_F(\sigma)^{-1} F(x_k)$ where σ is an approximation of the wanted solution. This approach is computationally attractive since a factorization of $J_F(\sigma)$ can be precomputed. For an extended overview of Newton methods see [30]. In this thesis, we show how to formulate NEPs as systems of nonlinear equations and how to apply and analyze Newton-like methods in this context.

The rest of Part I is structured as follows: Chapter 2 is dedicated to the NEP: the problem is presented in Section 2.1 and in Section 2.2 we describe various applications. In Section 2.3 we present two Krylov methods for NEP. In Section 2.4 we show how Newton-like methods can be applied and analyzed in the context of solving NEPs. A Julia package for NEP is presented in Section 2.5. In Chapter 3 we introduce the topic of matrix equations by pointing out the connections with the NEP. The problem is defined in Section 3.1 and two applications are described in Section 3.2. Finally, in Section 3.3 and Section 3.4 we present two methods for solving a specific class of matrix equations.

Chapter 2

Nonlinear eigenvalue problems

2.1 Problem description

We consider the *nonlinear eigenvalue problem* (NEP) defined as finding $(\lambda, v) \in \mathbb{C} \times \mathbb{C}^n \setminus \{0\}$ such that

$$M(\lambda)v = 0, \tag{2.1}$$

where $\lambda \in \Omega \subseteq \mathbb{C}$, Ω is a simply connected open set and $M : \Omega \to \mathbb{C}^{n\times n}$ is analytic. The vector v is called eigenvector and the scalar λ is called eigenvalue. The targeted eigenvalues are located in Ω. We also assume that the origin is an interior point of Ω. The NEP has received a considerable attention in literature. See the review papers [66, 41] and PhD theses [31, 95, 81]. These problems arise in many different areas such as vibration analysis, fluid dynamics, wave propagation, stability analysis, etc. See the problem collection [13, 44]. Specific types of NEPs have been extensively analyzed. If the λ-dependence in (2.1) is polynomial, i.e., if there are $A_j \in \mathbb{C}^{n\times n}$, for $j = 1, \ldots, d$, such that

$$M(\lambda) = \sum_{j=0}^{d} \lambda^j A_j, \tag{2.2}$$

the NEP is referred to as a *polynomial eigenvalue problem* (PEP) of degree d. It is always possible to decompose the NEP in sum of products of matrices and functions (SPMF), such that

$$M(\lambda) = \sum_{j=0}^{d} f_j(\lambda) A_j. \tag{2.3}$$

for some functions $\{f_j\}_{j=1}^d$. The representation (2.3) is computationally exploitable when d is small. If the functions $\{f_j\}_{j=1}^d$ are rational, the problem is referred to as a *rational eigenvalue problem* (REP). In case these functions are neither polynomial nor rational, the problems is often referred to as a *transcendental nonlinear eigenvalue problem*. Such kind of NEPs arise in, e.g., the simulation of optical fibers [52], cavity in accelerator design [58], double-periodic photonic crystals [34, 35], microelectromechanical systems [19], nano optics [3] and two-parameter eigenvalue problems [78]. Some of these NEPs cannot be efficiently represented as (2.3), as only the matrix vector products $M(\lambda)v$, and eventually the derivatives with respect to λ, are computable.

Given an eigenpair approximation (λ, v), we define the residual as $\|M(\lambda)v\|_2$. If the NEP can be efficiently formulated as (2.3), a more useful quantification of the error is obtained by scaling the residual as follows

$$\frac{\|M(\lambda)v\|_2}{\sum_{j=0}^d |f_j(\lambda)|\|A_j\|_\infty \|v\|_2}. \tag{2.4}$$

Throughout this thesis (2.4) will be referred to as *the relative residual*, consistent, e.g., with the terminology in [97, 58].

There are several classes of methods for solving NEPs, among them: linearization-based methods (PEPs and REPs) [79, 28, 1, 57, 89], projection-based methods [98, 14, 32, 100, 58], contour integral approaches [15, 4], Krylov methods and Newton approaches. In Section 2.3 and Section 2.4 we present the framework of Krylov and Newton methods for NEP and we describe some methods that belong to these two classes. In the following section we present several applications which are relevant in the context of this thesis.

2.2 Applications

2.2.1 Waveguide eigenvalue problem

The propagation of waves is governed by the Helmholtz equation

$$\Delta v(x,z) + \kappa(x,z)^2 v(x,z) = 0, \qquad (x,z) \in \mathbb{R}^2, \tag{2.5}$$

where $\kappa(x,z)$ is the *wavenumber*. We assume the wavenumber to be periodic in the z-direction, piecewise constant and constant for sufficiently large $|x|$. Under these assumptions, equation (2.5) describes a wave traveling though a waveguide with a specific structure. Namely, the wave travels though a medium which is periodic in one direction and consists of layers of homogeneous materials of different kinds. Figure 2.1 shows an example of this setup, which has also been studied, e.g., in [92]. We are interested in computing Bloch solutions to (2.5), namely those solutions that can be factorized as a product of a z-periodic function and $e^{\lambda z}$, i.e.,

$$v(x,z) = \hat{v}(x,z)e^{\lambda z}, \qquad \text{where} \qquad \hat{v}(x,z+1) = \hat{v}(x,z). \tag{2.6}$$

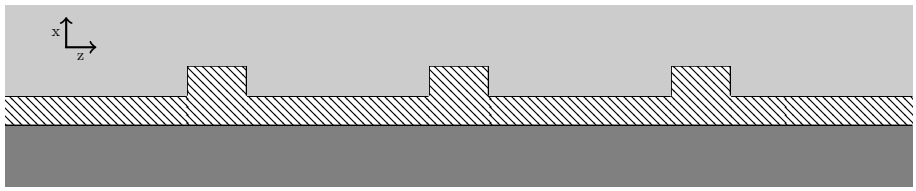

Figure 2.1: Illustration of a waveguide where $\kappa(x, z)$ is constant in three regions. The medium consists of three layers of different homogeneous materials.

In this context, Bloch solutions are also called *guided modes* of (2.5). If λ is purely imaginary, the mode is called propagating; if $|\operatorname{Re}\lambda|$ is small, it is called leaky. Both mode types are of great interest in various settings [37, 76, 91, 88, 9]. In Paper I we present a procedure for computing leaky modes, with $\operatorname{Re}\lambda < 0$ and $\operatorname{Im}\lambda \in (-2\pi, 0)$.

The pairs $(\lambda, \hat{v})$, representing the Bloch solution (2.6), are eigenpairs of the PDE-eigenvalue problem obtained by replacing (2.6) in (2.5). However, this PDE-eigenvalue problem is defined on an unbounded domain. We reformulate this problem by using the Dirichlet-to-Neumann-map (DtN-map). This results in an eigenvalue problem defined on a bounded domain with eigenvalue-dependent boundary conditions. The discretization of this problem obtained with the finite element method (FEM) results in a NEP with the following structure:

$$\begin{bmatrix} Q(\lambda) & C_1(\lambda) \\ C_2^T & P(\lambda) \end{bmatrix} v = 0. \tag{2.7}$$

The matrices $Q(\lambda)$ and $C_1(\lambda)$ are quadratic polynomials in λ, $Q(\lambda) = A_0 + \lambda A_1 + \lambda^2 A_2$, $C_1(\lambda) = C_{1,0} + C_{1,1}\lambda + C_{1,2}\lambda^2$, where $A_i, C_{1,i}$ and C_2^T are large and sparse. The matrix $P(\lambda)$ has the structure

$$P(\lambda) = \begin{bmatrix} R\Lambda_-(\lambda)R^{-1} & 0 \\ 0 & R\Lambda_+(\lambda)R^{-1} \end{bmatrix}$$

where $\Lambda_\pm(\lambda)$ are diagonal matrices containing nonlinear functions of λ involving square roots of polynomials. The matrix-vector product corresponding to R and R^{-1} can be computed with the fast Fourier transform. In this thesis, we refer to the NEP in (2.7) as waveguide eigenvalue problem (WEP). The solution to the WEP is used to reconstruct an approximation to the Bloch solution (2.6). One eigenfunction (Bloch solution) with respect to the waveguide in Figure 2.1 is illustrated in Figure 2.2. This problem is fully described in Paper I.

2.2.2 Delay eigenvalue problem

We now discuss the linear time delay systems described by the delay differential equation (DDE)

$$\dot{x}(t) = Ax(t) + \sum_{j=1}^{m} A_j x(t - \tau_j), \tag{2.8}$$

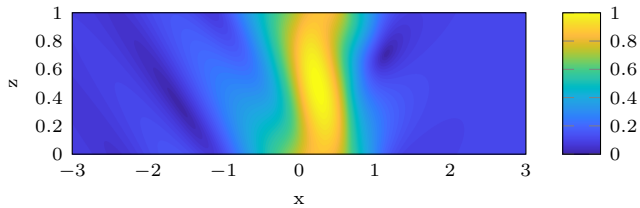

Figure 2.2: One eigenfunction of the WEP for the waveguide illustrated in Figure 2.1.

where $\tau_j > 0$ are the time-delay coefficients and $A, A_j \in \mathbb{C}^{n\times n}$ for $j = 1, \ldots, m$. DDEs arise in many applications, they are tools to model delay responses of a system. It is well known that the stability of non-delayed linear systems, namely $\{A_j\}_{j=1}^m = 0$, can be characterized by the spectrum of the coefficient matrix A. For the more general case (2.8), the stability can still be characterized by the matrix coefficients in the following way. Let us define the NEP

$$M(\lambda) = -\lambda I + A_0 + \sum_{j=1}^{m} A_j e^{-\tau_j \lambda}. \tag{2.9}$$

If the real parts of the eigenvalues of (2.9) are negative, the system (2.8) is stable. If there is at least one eigenvalue that has positive real part, the system (2.8) is unstable [69, Proposition 1.6]. The NEP in (2.9) is referred to as *delay eigenvalue problem* (DEP).

Example 2.2.1. *The following delay differential equation models the heat transfer in a beam coupled with an insulator*

$$\frac{\partial}{\partial t}u(x,t) = \frac{\partial^2}{\partial x^2}u(x,t) + ku(x,t-1), \qquad x \in [0,1]. \tag{2.10}$$

The feedback control is applied to the insulator and the propagation of the heat through the insulator is modeled with a delay term. A finite difference (FD) discretization in space of (2.10) *results in a DDE. We use a uniform grid and the second-order central scheme for approximating the second derivative. For $k = 0$ (no delay term), and in general small k, the system is stable, see Figure 2.3. For large values of k the system becomes unstable, see Figure 2.4.*

2.2.3 Other applications

A problem arising in modelling an electromagnetic cavity: The following NEP models a waveguide-loaded accelerator cavity [58].

$$M(\lambda) = A_1 - \lambda A_2 + i\sqrt{\lambda}A_3 + i\sqrt{\lambda - \sigma_2^2}A_4, \tag{2.11}$$

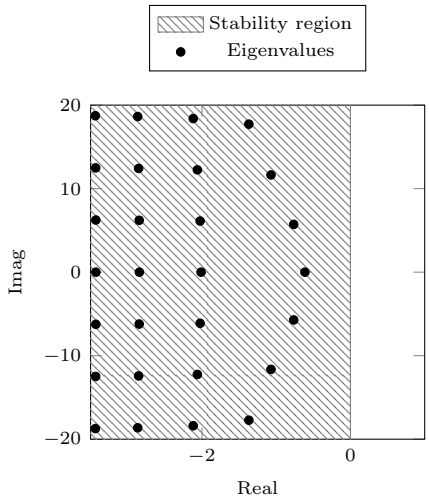

Figure 2.3: Spectrum of the DDE arising from the FD discretization of (2.10) *for* $k = 5$.

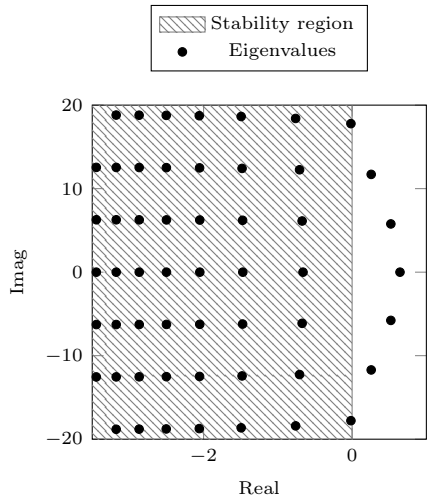

Figure 2.4: Spectrum of the DDE arising from the FD discretization of (2.10) *for* $k = 20$.

where $\sigma_2 = 108.8774$. The matrices $A_j \in \mathbb{R}^{9956\times 9956}$, for $j = 1, \ldots, 4$, are real and symmetric and are derived by a FEM discretization of Maxwell's equation. This problem is available in the problem collection [13] and in NEP-PACK. The eigenvalues of interest are located inside the closed disk centered in 250^2 with radius $5 \cdot 10^4$. This problem has been solved with various methods [51, 42, 97, 60, 38] and, by numerical evidences, the region of interest contains 21 eigenvalues, which are displayed in Figure 2.11. Similar to the WEP (2.7), the nonlinearity consists of square roots of polynomials.

Quadratic eigenvalue problem: The eigenvalues of the PEP,

$$M(\lambda) = \lambda^2 A_2 + \lambda A_1 + A_0,$$

referred to as a quadratic eigenvalue problem (QEP), characterize the stability of the linear second-order differential equation

$$A_2 \ddot{q}(t) + A_1 \dot{q}(t) + A_0 q(t) = 0. \tag{2.12}$$

More precisely, the system is stable if all the eigenvalues have negative real part and unstable if at least one eigenvalue has positive real part. Such problems arise in vibration analysis of structural systems. See [93] and reference therein.

Example 2.2.2. *Consider the damped mass-spring system illustrated in Figure 2.5. The i-th mass, of weight m_i, is connected to the $(i+1)$-st mass by a spring with constant k_{i+1} and it is also connected to the ground by a damper with constant b_i.*

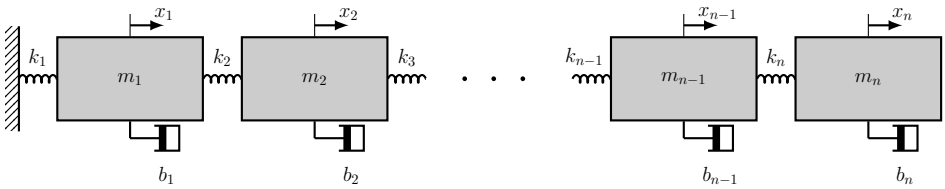

Figure 2.5: The damped mass-spring system described in Example 2.2.2.

The first mass is connected to the wall with a spring with constant k_1. The vibration of this system is governed by a second-order differential equation [29, Section 4.2] of the form (2.12)*, where*

$$A_0 = \begin{pmatrix} k_1+k_2 & -k_2 & & & \\ -k_2 & k_2+k_3 & -k_2 & & \\ & \ddots & \ddots & \ddots & \\ & & -k_{n-1} & k_{n-1}+k_n & -k_n \\ & & & -k_n & k_n \end{pmatrix}, \quad A_1 = \begin{pmatrix} b_1 & & \\ & \ddots & \\ & & b_n \end{pmatrix},$$

$$A_2 = \begin{pmatrix} m_1 & & \\ & \ddots & \\ & & m_n \end{pmatrix}.$$

Boundary element method (BEM) for a PDE eigenvalue problem: The $3D$ Laplace linear eigenvalue problem with Dirichlet boundary conditions

$$\begin{aligned} -\Delta u &= \lambda^2 u && \text{in } D \subseteq \mathbb{R}^3 \\ u &= 0 && \text{on } \partial D \end{aligned}$$

can be reformulated as a boundary integral equation [87, 33]. In particular, if we use a boundary element space of piecewise constant functions on a surface mesh consisting of triangles $\Delta_1, \ldots, \Delta_n$, we obtain a NEP whose entries are defined by a surface integral

$$[M(\lambda)]_{p,q} = \int_{\Delta_p} \int_{\Delta_q} \frac{e^{i\lambda\|\xi-\eta\|}}{\|\xi-\eta\|} dS(\xi) dS(\eta).$$

Notice that $M(\lambda)$ is a full matrix whose entries are computationally expensive to compute.

Order reduction in linear eigenvalue problem Given $\{A_{i,j}\}_{i,j=1}^{2} \subset \mathbb{C}^{n\times n}$ with $A_{1,2}$ nonsingular. The linear eigenvalue problem of size $2n$ given by

$$\begin{pmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \lambda \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \tag{2.13}$$

can be reduced to a QEP of size n. From the first block equation of (2.13) we obtain $x_2 = -A_{1,2}^{-1}(A_{1,1}x_1 - \lambda x_1)$. By replacing this in the second block equation we get the QEP

$$\left[\lambda^2 A_{1,2}^{-1} - \lambda(A_{2,2}A_{1,2}^{-1} + A_{1,2}^{-1}A_{1,1}) + (A_{2,2}A_{2,1}^{-1}A_{1,1} - A_{2,1})\right] x_1 = 0.$$

This technique, presented in [57, Ch.5, Sec.1], can be extended to reduce a linear eigenvalue problem of size dn to a PEP of size n and degree d. Eigenvalue problems defined by block matrices arise in many applications. See [12] and reference therein.

2.3 Krylov-based methods

Krylov methods have been extensively used for solving structured large scale linear eigenvalue problem. One way to extend these methods to the NEP consists of the following process. We construct a linear eigenvalue problem, referred to as a *linearization*, whose eigenvalues approximate, or correspond to, the eigenvalues of the original NEP. For example, a linearization to the PEP in (2.2) is given by

$$\begin{pmatrix} -A_0 & & & & \\ & I & & & \\ & & I & & \\ & & & \ddots & \\ & & & & I \end{pmatrix} \begin{pmatrix} v \\ \lambda v \\ \lambda^2 v \\ \vdots \\ \lambda^d v \end{pmatrix} = \lambda \begin{pmatrix} A_1 & A_2 & A_3 & \dots & A_d \\ I & & & & \\ & I & & & \\ & & I & & \\ & & & \ddots & \end{pmatrix} \begin{pmatrix} v \\ \lambda v \\ \lambda^2 v \\ \vdots \\ \lambda^d v \end{pmatrix}. \tag{2.14}$$

In particular, the first block equation gives $M(\lambda)v = 0$. This linearization technique is referred to as *companion linearization*. There are many different techniques to linearize PEPs and REPs. See [79, 28, 1, 57, 89] and reference therein. Krylov methods are then applied to the linearization. The performance of such methods improves by exploiting the structure of the NEP and of the linearization. For example, there are linearizations that preserve the structure of the NEP such as being symmetric, Hermitian, Hamiltonian, palindromic, etc. See [64, 89, 68, 63, 23, 24, 36]. Structure preserving methods are, for certain applications, more robust and efficient than methods that do not preserve the structure [67, 65].

The structure of the matrix (2.14), and of other linearizations, leads to a linear dependence of the block vectors representing the basis matrix V_m in (1.2). This is exploited in memory efficient Krylov methods. These methods do not explicitly store the full basis matrix [7, 55]. A class of Krylov-like methods for solving the NEP when the nonlinearity is neither polynomial nor rational consists of first approximate $M(\lambda)$ with a PEP or a REP and then apply a Krylov method to the obtained linearization [33, 61]. There are also Kryov-based methods that iteratively expand the linearization or that are applied directly to an equivalent infinite dimensional problem: NLEIGS [42], the Hermite-based approach in [96], IAR [51] and bi-Lanczos approach [38]. Recent research has been focused on variations of these methods which are also memory efficient: CORK [97], its two sided version [60] and TIAR in Paper I. In the following section we present two methods, developed

during the PhD, that correspond to applying the Arnoldi method to an equivalent linear infinite dimensional eigenvalue problem.

2.3.1 Tensor Infinite Arnoldi method

We can express $M(\lambda)$, by using its analyticity, as

$$M(\lambda) = \sum_{j=0}^{\infty} \frac{\lambda^j}{j!} M_j, \tag{2.15}$$

where $M_j := M^{(j)}(0)$. Similar to the companion linearization (2.14), we can derive a linearization to (2.15) that involves infinite dimensional matrices. More precisely, if (λ, v) is an eigenpair of $M(\lambda)$, the following relation between vectors and matrices of infinite length is fulfilled

$$\begin{pmatrix} -M_0 & & & & \\ & I & & & \\ & & I & & \\ & & & I & \\ & & & & \ddots \end{pmatrix} \begin{pmatrix} \frac{\lambda^0}{0!}v \\ \frac{\lambda^1}{1!}v \\ \frac{\lambda^2}{2!}v \\ \frac{\lambda^3}{3!}v \\ \vdots \end{pmatrix} = \lambda \begin{pmatrix} M_1 & \frac{1}{2}M_2 & \frac{1}{3}M_3 & \frac{1}{4}M_4 & \dots \\ \frac{1}{1}I & & & & \\ & \frac{1}{2}I & & & \\ & & \frac{1}{3}I & & \\ & & & \ddots & \end{pmatrix} \begin{pmatrix} \frac{\lambda^0}{0!}v \\ \frac{\lambda^1}{1!}v \\ \frac{\lambda^2}{2!}v \\ \frac{\lambda^3}{3!}v \\ \vdots \end{pmatrix}. \tag{2.16}$$

The infinite Arnoldi method (IAR), presented in [51], corresponds to applying the shift-and-invert Arnoldi method, with shift in zero, to the linear eigenvalue problem (2.16). By choosing a starting vector that has only the first block which is nonzero, the method can be carried out in finite arithmetic and without introducing any approximation, or truncation, to the infinite dimensional pencil (2.16). In (1.2), the orthogonal basis matrix V_m of the Krylov space, which has an infinite number of rows, has only a finite number of nonzero elements. In particular, the vector v_k has only the first k blocks which are nonzero. The structure of the basis matrix when the Arnoldi method is applied to (2.16) is illustrated in Figure 2.6 . In Paper I we derive a memory efficient version of IAR, referred to as *tensor infinite Arnoldi* (TIAR). This method is based on the following result.

Theorem 2.3.1. *Assume that the infinite length vector v_1 has only the first block which is nonzero. Let $(V_{m+1}, \underline{H}_m)$ be the output of IAR, which corresponds to Algorithm 1 (shift-and-invert version) applied to* (2.16)*. Let $Z \in \mathbb{C}^{n\times(m+1)}$ be an orthogonal basis of the first block row of V_{m+1}. Then, there exist $a \in \mathbb{C}^{(m+1)\times(m+1)\times(m+1)}$ such that*

$$v_{i,j} = \sum_{\ell=1}^{m+1} a_{i,j,\ell} z_\ell \qquad \text{for} \quad i,j = 1,\dots,m+1. \tag{2.17}$$

Moreover, Algorithm 1 can be applied to (2.16) *without ever explicitly constructing the basis matrix V_{m+1} but only using (and expanding) its implicit representation*

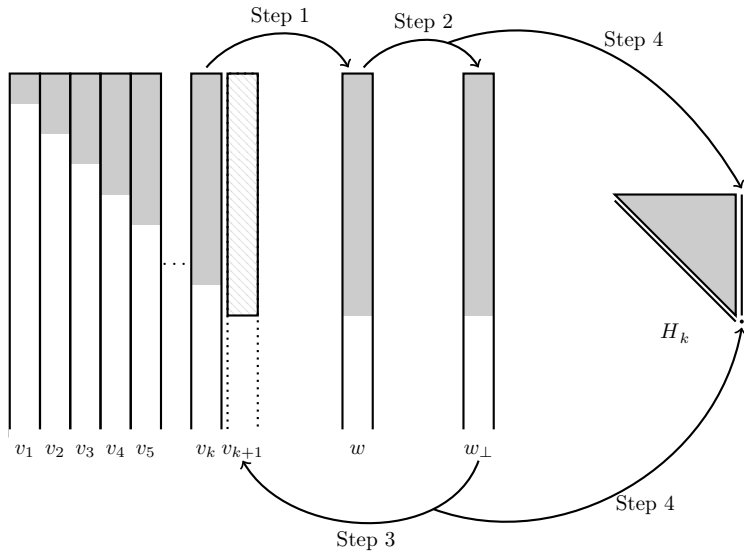

Figure 2.6: Algorithm 1 (shift-and-invert Arnoldi) applied to (2.16)*. The opaque and oblique line pattern represent the nonzero part of the infinite length vectors.*

given by the tensor a and the matrix Z in (2.17)*.*

The above theorem states that all the block vectors of V_m can be generated as a linear combination of the first block row. In Figure 2.7 we illustrate the dependence between the blocks of the basis matrix V_m. TIAR consists of applying IAR without ever explicitly constructing the basis matrix V_m but only using, and expanding, its implicit representation (2.17). Therefore TIAR is mathematically equivalent to IAR. An advantage of TIAR consists of a lower memory usage since only the orthogonal matrix $Z \in \mathbb{C}^{n\times(m+1)}$ and the tensor $a \in \mathbb{C}^{(m+1)\times(m+1)\times(m+1)}$ are stored. The columns of the matrix Z are an orthogonal basis of the space spanned by the vectors in first block row of the matrix V_m. The tensor a contains the coefficients to represent the whole matrix V_m. The matrix Z is chosen to be orthogonal because of numerical stability. The convergence history of TIAR applied to the WEP, described in Section 2.2.1, is illustrated in Figure 2.8.

Due to the memory efficient representation of the basis, TIAR can applied to NEPs of larger size with respect to IAR. However, when too many iterations are required to compute the eigenvalues of interest, the computation time, and the memory usage, become restrictive. Therefore, it is important to have an efficient and robust restarting strategy. In Paper III we derive two restarting techniques: a semi-explicit restart and an implicit restart. These restarting techniques are based on an equivalent infinite-dimensional formulation of (2.1) which we now recall.

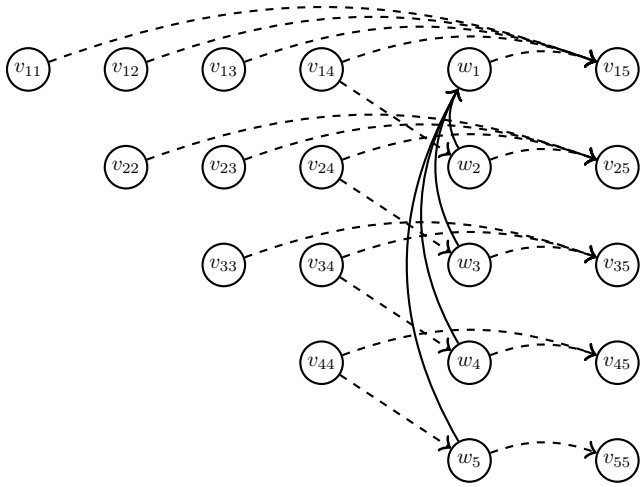

Figure 2.7: Structure of the basis matrix V_5 generated by Algorithm 1 (shift-and-invert Arnoldi) applied to (2.16). The vector $v_{i,j}$ is the i-th block of v_j. The computation of v_5 is highlighted. Dashed lines correspond to computing linear combinations. The only block of v_5 that is not linearly dependent on the others is the first.

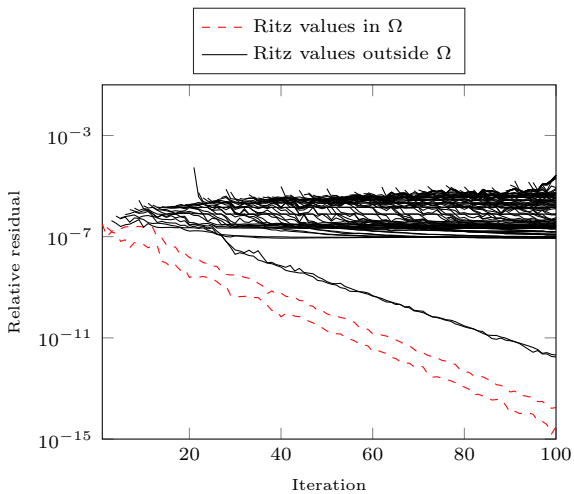

Figure 2.8: Convergence history of TIAR applied to the WEP described in Section 2.2.1.

Under the assumption that zero is not an eigenvalue, the problem (2.1) can be reformulated as $\lambda B(\lambda)v = v$, where $B(\lambda) = M(0)^{-1}(M(0) - M(\lambda))/\lambda$. We now define the linear operator $\mathcal{B}$ which maps functions to functions

$$\mathcal{B}\psi(\theta) := \int_0^\theta \psi(\hat{\theta})d\hat{\theta} + \sum_{i=0}^{\infty} \frac{B^{(i)}(0)}{i!}\psi^{(i)}(0). \tag{2.18}$$

The linear infinite dimensional eigenvalue problem $\lambda\mathcal{B}\psi = \psi$ is equivalent to the NEP [51, Theorem 3]. More precisely, if (λ, v) is an eigenpair of the NEP then (λ, ve^{λ}) is an eigenpair of $\mathcal{B}$. IAR (and TIAR) is equivalent to applying the Arnoldi method to the operator $\mathcal{B}$ with a specific scalar product and the starting function $\psi(\theta) = v_{1,1} \in \mathbb{C}^n$, i.e., a constant function. The implicit restarting technique that we present in Paper III consists of using the Krylov-Schur restart method to the operator (2.18). This restarting technique transforms the Arnoldi factorization (1.2) into another Arnoldi factorization $\mathcal{B}\tilde{V}_p = \tilde{V}_{p+1}\underline{\tilde{H}}_p$, with $p < m$, such that the eigenvalues of $\tilde{H}_p$ are also eigenvalues of H_m and are the Ritz values of interest. In the context of TIAR, this corresponds to transforming the implicit representation of the basis V_m. An efficient implementation of this approach is based further approximations which are discussed in Paper III. The semi-explicit restart instead consists of using the sum of a polynomial and an exponential function as starting function $\psi(\theta)$. This approach is based on the fact that the eigenpairs of $\mathcal{B}$ are exponential functions. Therefore, at restart, the converged eigenpairs are locked in the exponential part of the starting function $\psi(\theta)$ in the same fashion as in [49], but in the TIAR framework. The two restarting strategies have different features and their performances depend on the problem structure. Figure 2.9 and Figure 2.10, which are extracted form Paper III, show a comparison between these two approaches applied to a DEP. For this specific example, the semi-explicit restart has a slower converge than respect to the implicit restart but requires less memory.

2.3.2 Infinite Lanczos

If the NEP is symmetric, namely $M(\lambda)^T = M(\lambda)$ for every $\lambda \in \Omega$, it is possible to construct a linearization which is also symmetric. Let us define the infinite dimensional matrix

$$\begin{pmatrix} I & & & & & \\ & c_{1,1}M_2 & c_{1,2}M_3 & c_{1,3}M_4 & c_{1,4}M_5 & \dots \\ & c_{2,1}M_3 & c_{2,2}M_4 & c_{2,3}M_5 & & \\ & c_{3,1}M_4 & c_{3,2}M_5 & & & \\ & c_{4,1}M_5 & & & & \\ & \vdots & & & & \end{pmatrix}, \quad \begin{aligned} c_{i,1} &:= \frac{1}{i+1} && i \geq 1, \\ c_{i-1,j} &:= \frac{jc_{i,j-1}}{i} && i,j > 1. \end{aligned} \tag{2.19}$$

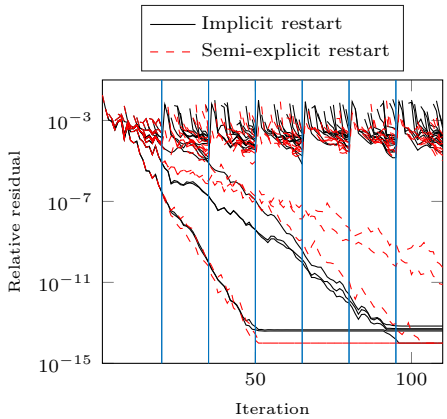

Figure 2.9: Error history of TIAR with implicit and explicit restart applied to the DEP in Paper III. The blue vertical lines represent the iteration in which the restart is performed.

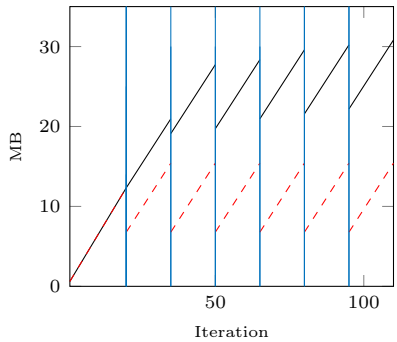

Figure 2.10: Memory usage of TIAR with implicit and explicit restart applied to the DEP in Paper III. The blue vertical lines represent the iteration in which the restart is performed.

By multiplying (2.16) on the left by (2.19) we obtain the eigenvalue problem

$$\begin{pmatrix} -M_0 & & & & \\ & c_{1,1}M_2 & c_{1,2}M_3 & c_{1,3}M_4 & \dots \\ & c_{2,1}M_3 & c_{2,2}M_4 & & \\ & c_{3,1}M_4 & & & \\ & \vdots & & & \end{pmatrix} \begin{pmatrix} \frac{\lambda^0}{0!}v \\ \frac{\lambda^1}{1!}v \\ \frac{\lambda^2}{2!}v \\ \frac{\lambda^3}{3!}v \\ \vdots \end{pmatrix} = \lambda \begin{pmatrix} \frac{1}{1}M_1 & \frac{1}{2}M_2 & \frac{1}{3}M_3 & \frac{1}{4}M_4 & \dots \\ \frac{c_{1,1}}{1}M_2 & \frac{c_{1,2}}{2}M_3 & \frac{c_{1,3}}{3}M_4 & & \\ \frac{c_{2,1}}{1}M_3 & \frac{c_{2,2}}{2}M_4 & & & \\ \frac{c_{3,1}}{1}M_4 & & & & \\ \vdots & & & & \end{pmatrix} \begin{pmatrix} \frac{\lambda^0}{0!}v \\ \frac{\lambda^1}{1!}v \\ \frac{\lambda^2}{2!}v \\ \frac{\lambda^3}{3!}v \\ \vdots \end{pmatrix}. \quad (2.20)$$

In Paper V, we show that the matrices in (2.20) are symmetric. A method that can be applied to the symmetric eigenvalue problem, which exploits the symmetry of the matrices, is the indefinite Lanczos method [75] [5, Ch.15, Sec.11]. In particular, the orthogonal basis of the Krylov space can be computed with a three term recurrence. In Paper V, we present a method, referred to as *infinite Lanczos* (ILAN), which corresponds to applying the indefinite Lanczos method to (2.20). The drawback of using Lanczos-like methods is the loss of orthogonality. In particular, the projection matrix $\underline{H}_m$ in (1.2), which contains the orthogonalization coefficients, is effected by high roundoff errors. Therefore, the Ritz values may not provide a good approximation to the eigenvalues. We resolve this problem by extracting the eigenvalue approximations in a more robust way. More precisely, we use the nonlinear Rayleigh-Ritz procedure, which consists of solving the projected problem $\tilde{V}^T M(\lambda)\tilde{V}z = 0$, where $\tilde{V}$ is a chosen orthogonal matrix, and computing the eigenpair approximations as $(\lambda, \tilde{V}z)$. This method is effective if the eigenvectors of the NEP can be approximated as a linear combination of the columns of $\tilde{V}$. Due to the structure of the matrices in (2.20), a natural candidate for the matrix $\tilde{V}$ is

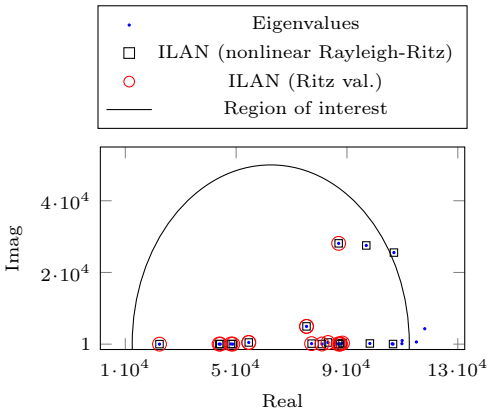

Figure 2.11: Converged eigenvalues after 100 *iterations of ILAN applied to the NEP* (2.11). *The projected problem is solved with IAR.*

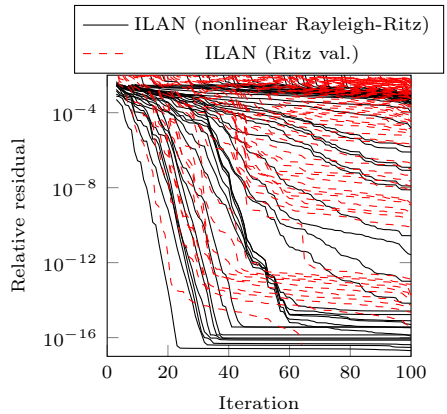

Figure 2.12: Error history for ILAN applied to the NEP (2.11). *In each iteration the projected problem is solved with IAR.*

an orthogonal basis of the first block row of V_m. In Figure 2.11 and Figure 2.12 we illustrate the performance of ILAN applied to the NEP (2.11). Clearly the eigenpair extraction based on the nonlinear Rayleigh-Ritz procedure is more effective than the standard Ritz procedure.

2.4 Newton-based methods

Newton methods are an attractive class of methods for solving NEPs due to their fast local convergence. These methods are, in general, effective only if a good initial approximation of the eigenpairs is available. The main idea consists of formulating the NEP as a system of nonlinear equations and then iteratively solve this problem with a Newton-based approach. Newton's methods for NEPs have been developed and analyzed for decades, and the field is still under active development, as can be observed in the summaries in [66, 99] and the PhD thesis [81]. There are many variations of such methods, among them : QR-based methods [56, 39], Jacobi–Davidson-type methods [14, 20, 85], implicit determinant approach [86], Broyden's method [47], etc. Recent variations include block versions [54, 16, 17, 33], which allow the computation of many eigenvalues simultaneously. The convergence of several Newton-like methods has also been studied in [90, 46, 94, 50] and Paper IV.

A way to phrase the NEP as a system of nonlinear equations is

$$F\begin{pmatrix} v \\ \lambda \end{pmatrix} := \begin{pmatrix} M(\lambda)v \\ c^H v - 1 \end{pmatrix} = 0. \tag{2.21}$$

The first block equation corresponds to (2.1) whereas the second equation gives the normalization condition $c^H v = 1$. Newton's method can be directly applied to (2.21) resulting in the sequence

$$J_k \begin{pmatrix} z_{k+1} - z_k \\ \mu_{k+1} - \mu_k \end{pmatrix} = -\begin{pmatrix} M(\mu_k)z_k \\ c^H z_k - 1 \end{pmatrix} \quad \text{where} \quad J_k := \begin{pmatrix} M(\mu_k) & M'(\mu_k)z_k \\ c^H & 0 \end{pmatrix}. \tag{2.22}$$

In Paper IV, we consider quasi-Newton methods which consist of approximating the Jacobian matrix J_k in different ways. We focus on the following four approximations:

$$\tilde{J}_k^{(1)} := \begin{pmatrix} M(\sigma) & M'(\sigma)z_0 \\ c^H & 0 \end{pmatrix} \tag{2.23}$$

$$\tilde{J}_k^{(2)} := \begin{pmatrix} M(\sigma) & M'(\mu_k)z_k \\ c^H & 0 \end{pmatrix} \tag{2.24}$$

$$\tilde{J}_k^{(3)} := \begin{pmatrix} M(\sigma) & M[\mu_{k+1}, \mu_k]z_k \\ c^H & 0 \end{pmatrix} \tag{2.25}$$

$$\tilde{J}_k^{(4)} := \begin{pmatrix} M(\mu_k) & M'(\mu_k)z_{k+1} \\ c^H & 0 \end{pmatrix} \tag{2.26}$$

where $\sigma \in \mathbb{C}$ is an approximation to the wanted eigenvalue and where we use the standard notation for divided differences

$$M[\lambda, \mu] = \begin{cases} \frac{M(\lambda) - M(\mu)}{\lambda - \mu} & \lambda \neq \mu, \\ M'(\lambda) & \lambda = \mu. \end{cases}$$

Note that the modification of the Jacobian matrix in (2.25) and (2.26) makes the iteration (2.22) implicit, in the sense that, the formula (2.22) for the new approximation (μ_{k+1}, z_{k+1}) involves μ_{k+1} and z_{k+1} in a nonlinear way. Many implicit variations of Newton's method have been considered in the literature. See, e.g., [45, 53] and references therein. The quasi-Newton methods based on the Jacobian approximations presented above can be formulated with iterations that only involve the original matrix $M(\lambda)$ instead of the augmented matrices. In particular, in Paper IV we prove that quasi-Newton methods based on the approximation (2.25) and (2.26), are respectively equivalent to the well-established methods known as *residual inverse iteration* [70] and *Ruhe's method of successive linear problems* [80].

We denote by QNi the quasi-Newton approach based on the Jacobian approximation $\tilde{J}_k^{(i)}$. We combine the standard approach for the Newton convergence analysis with the Keldysh theorem, which is stated in the context of NEP, e.g., in [15, Section 2]. The result of this analysis, which is fully described in Paper IV, shows that the convergence of QN4 is quadratic. This is consistent with the equivalence

with *Ruhe's method of successive linear problems*. The converge of QN1, QN2 and QN3 is linear. More precisely, assuming that (λ_1, v_1) is the eigenpair of interest, it holds that

$$\begin{pmatrix} z_{k+1} - v_1 \\ \mu_{k+1} - \lambda_1 \end{pmatrix} = A \begin{pmatrix} z_k - v_1 \\ \mu_k - \lambda_1 \end{pmatrix} + \mathcal{O}\left\| \begin{pmatrix} z_k - v_1 \\ \mu_k - \lambda_1 \end{pmatrix} \right\|^2.$$

The spectral radius $\rho(A)$ determines the convergence factor of these methods. The convergence factor of QN1 is larger than the convergence factor of QN2 and QN3. The following theorem relates the convergence factor of QN2 and QN3, which are the same, to the clustering of the eigenvalues.

Theorem 2.4.1. *Consider $\Gamma \subset \Omega$ a simple, closed, piecewise-smooth curve containing the eigenvalues $\{\lambda_i\}_{i=1}^k$ and let $\{v_i, w_i\}_{i=1}^k \subset \mathbb{C}^n$ be the corresponding right and left eigenvectors, i.e., $w_i^H M(\lambda_i) = 0$ and $M(\lambda_i)v_i = 0$ for $i = 1, \ldots, k$. The convergence factor for QN2 and QN3, with respect to the eigenvalue λ_1, is bounded by*

$$\rho(A) \leq \|P_1\| \left\| M(\lambda_1) - M(\sigma) + \frac{M'(\lambda_1)v_1 w_\sigma^H M(\lambda_1)}{w_\sigma^H M'(\lambda_1) v_1} \right\| \left(\sum_{i=2}^{k} \frac{\kappa_i}{|\sigma - \lambda_i|} + \|R_\Gamma(\sigma)\| \right),$$

where $\kappa_i := (\|u_i\|\|v_i\|)/(|u_i^H M'(\lambda_i) v_i|)$, $P_1 := I - v_1 c^H$, $w_\sigma^H := c^H M(\sigma)^{-1}$ and

$$R_\Gamma(z) = \frac{1}{2\pi \mathrm{i}} \oint_\Gamma \frac{M(\lambda)^{-1}}{\lambda - z} \, \mathrm{d}\lambda. \tag{2.27}$$

The above theorem states that: the closer σ is to the eigenvalue λ_1, the smaller is the convergence factor. On the other hand, the closer the other eigenvalues are to λ_1, the larger is the convergence factor. This implies that computing eigenvalues which are clustered requires more iterations. The remainder term (2.27) can be fully characterized in certain cases. See Paper IV. The convergence properties of these quasi-Newton methods are illustrated in Figure 2.13 and Figure 2.14. These methods are applied to the REP 'loaded string' from the collection [13].

2.5 NEP-PACK: A Julia package for nonlinear eigenvalue problems

Specialized software for NEPs has been recently developed. For example, the modules `NEP` [25] and `PEP`, which belong to the SLEPc library [43], are specialized to efficiently represent and solve NEPs in the SLEPc framework. Many authors have released open-source software reproducing numerical results presented in their papers. We are currently developing NEP-PACK[1] [48] which is a Julia [18] package for NEPs. This package is designed to be easy to use for application researchers as well as for algorithm developers. In order to make, in this specific framework, the

[1] `https://github.com/nep-pack/NonlinearEigenproblems.jl`

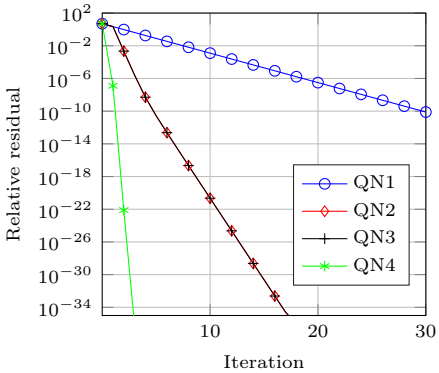

Figure 2.13: Converge diagram of the quasi-Newton methods presented in Section 2.4 applied to the REP 'loaded string' from the collection [13]. The right-most eigenpair (λ_1, v_1) is targeted. The initial guess is $(\mu_0, x_0) = (\lambda_1 + 5, v_1 + 0.15(1, \ldots, 1))$.

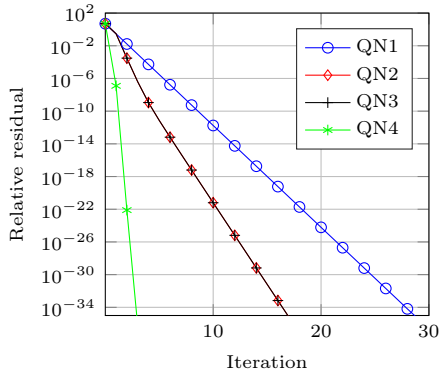

Figure 2.14: Converge diagram of the quasi-Newton methods presented in Section 2.4 applied to the REP 'loaded string' from the collection [13]. The right-most eigenpair (λ_1, v_1) is targeted. The initial guess is $(\mu_0, x_0) = (\lambda_1 + 5, v_1 + 0.05(1, \ldots, 1))$.

performance comparisons between algorithms fair, particular attention is paid to the algorithmic neutrality. NEP-PACK is a registered package in the Julia central package repository, which makes it possible to install the package with very little effort:

```
(v1.2) pkg> add NonlinearEigenproblems
julia> using NonlinearEigenproblems
```

NEP-PACK contains:

- Common types.
  Data structures to efficiently represent NEPs of the most common types, e.g., PEPs, DEPs, REPs, SPMF, etc.
- Structures with different ways of accessing the data.
  Different methods require data from the NEP in different ways, e.g., in TIAR and IAR the computation of the quantity

$$\sum_{i=1}^{k} M^{(i-1)}(\lambda) v_i$$

  is required. Other methods require the computation of other quantities, or in general, to access the NEP in different ways. Our package is designed in such a way that these quantities are natively accessible for many common types of NEPs.

- A large gallery of problems.
  Many benchmark and application derived NEPs are easily accessible, e.g., the WEP described in Section 2.2.1, the problem collection [13], the BEM in Section 2.2.3, etc.
- Methods.
  Many state-of-the-art and well-established methods are implemented, including all the methods presented in this thesis. The method of choice depends on many factors, e.g., the NEP structure, the region where the eigenvalues of interest are located, etc. Many methods of the following classes are accessible: Krylov-based methods (IAR, TIAR, NLEIGS, etc), Newton-type methods (residual inverse iteration, quasi Newton, successive linear problems, etc), projection methods and contour integral methods.
- Transformations.
  Several transformations are available, e.g., the Effenberger's deflation, the Cayley transformation, shift-and-scaling, and many more. In certain methods, such transformations are included in the algorithm level, which make the computations more robust.
- Interfaces.
  Functions from other programming languages, e.g., MATLAB, Python, Fortran, etc., are callable.

As example of usage, we apply TIAR to the following NEP of size 200×200.

$$M(\lambda) = \begin{pmatrix} -2 & 1 & & \\ 1 & \ddots & \ddots & \\ & \ddots & & 1 \\ & & 1 & -2 \end{pmatrix} + 100\lambda I + \sin\left(\lambda + e^{-\lambda}\right) \begin{pmatrix} 0 & 1 & & \\ 1 & \ddots & \ddots & \\ & \ddots & & 1 \\ & & 1 & 0 \end{pmatrix}.$$

```
using NonlinearEigenproblems, SparseArrays
n=200; ee = ones(n)
A1 = spdiagm(-1 => ee[1:n-1], 0 => -2*ee, 1 => ee[1:n-1])
A2 = 100*one(A1);
A3 = spdiagm(-1 => ee[1:n-1], 1 => ee[1:n-1])
f1 = λ -> one(λ); f2 = λ -> λ; f3 = λ -> sin(λ+(exp(-λ)))
nep=SPMF_NEP([A1,A2,A3],[f1,f2,f3])
λ,v=tiar(nep)
```

Then we can check that eigenpairs have been properly computed

```
julia> for i=1:length(λ) display(compute_resnorm(nep,λ[i],v[:,i])) end
1.7189719054904258e-15
1.8685309195718827e-15
2.998268409348198e-15
3.783696734688339e-15
1.3748320293976317e-12
8.884068070224529e-11
```

More examples are available in the NEP-PACK online user's guide [2].

[2] https://nep-pack.github.io/NonlinearEigenproblems.jl/

Chapter 3

Matrix equations

3.1 Problem description

We consider the *linear matrix equation*, in the unknown $X \in \mathbb{C}^{n\times n}$, defined as

$$\sum_{j=1}^{m} A_j X B_j^T = C, \tag{3.1}$$

where $A_1, \dots, A_m, B_1, \dots, B_m, C \in \mathbb{C}^{n\times n}$ are given matrix coefficients. Linear matrix equations arise in many applications and can be solved with various approaches. Efficient methods for solving these problems often exploit the structure of the matrix coefficients and the problem features. See the review paper [84] for an overview of the field. We focus on a specific class of matrix equations. We assume that (3.1) can be formulated as

$$\mathcal{L}(X) + \Pi(X) = C_1 C_2^T, \tag{3.2}$$

where $C_1, C_2 \in \mathbb{C}^{n\times r}$, $\mathcal{L}(X) = AX + XB^T$ with $A, B \in \mathbb{C}^{n\times n}$ and $\Pi(X)$ is a linear operator with specific properties defined later. The operator $\mathcal{L}$ is referred to as the *Sylvester operator* and the equation (3.2) as a *generalized Sylvester equation*[1]. These problems arise in many applications such as discretized PDEs [74, 77, 73], model-order reduction for bilinear systems [11, 21, 6], stability analysis of linear stochastic differential equations [27], fractional PDEs [22], etc. Generalized Sylvester equations have been efficiently solved under various assumptions on the matrix coefficients and on the operator $\Pi(X)$. See [26, 10, 83] and the PhD the-

[1] The nomenclature is ambiguous. The name *generalized Sylvester equation* is also used to refer to the matrix equation (3.1) with $m = 2$.

ses [82, 72].

Krylov-based methods have been successfully used under the assumption that $\rho(\mathcal{L}^{-1}\Pi) < 1$, A and B negative definite, and $r \ll n$, i.e., low rank right-hand side in (3.2). The effectiveness of Krylov-based method for solving matrix equations is related to the low-rank approximability properties of the solution. This has been studied, e.g., in [40, 10, 8] and Paper IV.

Matrix equations appear as intrinsic problem in many applications. In the context of solving NEPs, many iterative methods require the solution of a linear system at each iteration. In certain cases, these linear systems can be formulated as matrix equations. In this form, they can be efficiently solved with matrix equation approaches. See Paper II and Section 3.3.

3.2 Applications

3.2.1 Discretized PDEs

A large class of linear systems arising from the discretization of PDEs can be formulated as a generalized Sylvester equation. We present an example extracted from Paper IV.

Example 3.2.1. *The FD discretization, with a uniform grid, of the inhomogeneous Helmholtz equation*

$$\begin{cases} -\Delta u(x,y) + \kappa(x,y)u(x,y) = f(x,y), & (x,y) \in [0,1] \times \mathbb{R}, \\ u(0,y) = y(1,y) = 0, \\ u(x,y) = u(x,y+1), \end{cases}$$

results in the following generalized Sylvester equation

$$AU + UB + K \circ U = F,$$

where $\circ$ denoted the Hadamard product. Let us denote by n the number of discretization nodes in each direction and by $h := 1/(n-1)$ the mesh-size, the matrix coefficients $A, B, K, F \in \mathbb{R}^{n \times n}$ are defined as

$$B := \frac{1}{h^2} \begin{pmatrix} -2 & 1 & & \\ 1 & \ddots & \ddots & \\ & \ddots & & 1 \\ & & 1 & -2 \end{pmatrix}, \qquad \begin{aligned} A &:= B - (e_1, e_n)(e_n, e_1)^T/h^2, \\ K &:= [\kappa(x_i, y_j)]_{i,j=1}^N, \\ F &:= [f(x_i, y_j)]_{i,j=1}^N. \end{aligned}$$

For certain choices of the forcing term f and the wavenumber κ, the matrices F and K have low-rank. In particular it holds the decomposition $K \circ X = \sum_{j=1}^m N_j X M_j$, where N_j, M_j are diagonal matrices and m is the rank of K.

3.2.2 Model-order reduction

The controllability and observability of the bilinear system

$$\begin{cases} \dot{x}(t) = Ax(t) + \sum_{j=1}^{m} A_j u_j(t) x(t) + Bu(t), \\ y(t) = Cx(t), \end{cases} \tag{3.3}$$

where $A, A_1, \ldots, A_m \in \mathbb{R}^{n\times n}$ are symmetric matrices, A is negative definite, $B \in \mathbb{R}^{n\times k}$ and $C \in \mathbb{R}^{n\times s}$, is locally characterized by the Gramian matrices $P, Q \in \mathbb{R}^{n\times n}$. These matrices are the solutions to the matrix equations

$$AP + PA^T + \sum_{j=1}^{m} A_j P A_j^T = -BB^T, \tag{3.4}$$

$$A^T Q + QA + \sum_{j=1}^{m} A_j^T Q A_j = -C^T C. \tag{3.5}$$

The above matrix equations are referred to as *generalized Lyapunov equations* and are a specific case of (3.2). The matrix P is referred to as the *generalized controllability Gramian* and Q is the *generalized observability Gramian*. We now review several properties of these Gramians as presented in [11]. The existence of the Gramians is guaranteed by the condition $\|\mathcal{L}^{-1}\Pi\| < 1$, which also implies that P and Q are positive definite. The controllability Gramian is used to estimate the energy needed to reach a specific state $\bar{x}$, which is measured as

$$E_c(\bar{x}) := \min \left\{ \|u\|^2_{L^2[0,\infty)} \text{ such that } x(0) = 0 \text{ and } \lim_{t\to\infty} x(t) = \bar{x} \right\}.$$

It holds that $E_c(\bar{x}) > \bar{x}^T P^{-1} \bar{x}$, that implies that the smaller is $\bar{x}^T P \bar{x}$ the larger is $E_c(\bar{x})$, and this means that the state $\bar{x}$ is "hard" to reach. Similarly, if $\bar{x}^T Q \bar{x}$ is small, the state $\bar{x}$ is "hard" to observe. Several model-order reduction approaches reduce the size of the system by removing all the states that are hard to reach and observe. There are many methods for performing such a reduction of a system. See [71] [2, Chap 7] and references therein. We briefly recall the square root balanced truncation method.

Let $P = SS^T$ and $Q = RR^T$ be the Cholesky decompositions of the solutions to (3.4) and (3.5) and let $U\Sigma V^T \approx SR^T$ be the truncated SVD decomposition such that $U, V \in \mathbb{R}^{r\times n}$ and $\Sigma \in \mathbb{R}^{r\times r}$. We define $W = R^T V \Sigma^{-1/2}, V = S^T U \Sigma^{-1/2}$. Then, the reduced problem to (3.3) is

$$\begin{cases} \dot{\tilde{x}}(t) = \tilde{A}\tilde{x}(t) + \sum_{j=1}^{m} \tilde{A}_j u_j(t) \tilde{x} + \tilde{B}u(t), \\ \tilde{y}(t) = \tilde{C}\tilde{x}(t), \end{cases}$$

where $\tilde{A} = W^T A V, \tilde{B} = W^T B, \tilde{A}_j = W^T A_j V, \tilde{C} = CV$. We illustrate these ideas

with an example extracted from Paper IV.

Example 3.2.2. *We consider the time invariant multi-input and multi-output bilinear system described in [62, Example 2] which has the form* (3.3) *with* $m = 2$ *and is defined by the matrices*

$$A = \begin{pmatrix} -5 & 2 & & \\ 2 & \ddots & \ddots & \\ & \ddots & & 2 \\ & & 2 & -5 \end{pmatrix}, A_1 = \gamma^2 \begin{pmatrix} 0 & -3 & & \\ 3 & \ddots & \ddots & \\ & \ddots & & -3 \\ & & 3 & 0 \end{pmatrix}, A_2 = \gamma^2 I - A_1,$$

where $A, A_1, A_2 \in \mathbb{R}^{100\times 100}$ *and* $\gamma > 0$. *We choose* $C \in \mathbb{R}^{1\times 100}$ *and* $B \in \mathbb{R}^{100\times 2}$ *as normalized random matrices and* $\gamma = 1/4$. *We show the effectiveness of the order reduction. The initial state is set as zero and the input is selected as* $u_1(t) = e^{-t}$ *and* $u_2(t) = e^{-2t}$. *We measure the quality of the order reduction by plotting* $|y(t) - \tilde{y}(t)|$. *For this example* $\|\mathcal{L}^{-1}(\Pi)\| < 1$ *and therefore the Gramians exist. This condition may suggest that we can approximate the Gramians by simply neglecting the terms* A_j *in* (3.4) *and* (3.5). *This approach results in solving Lyapunov equations. As illustrated in Figure 3.1 and Figure 3.2, the approximation of the Gramians obtained with this approach leads to a less effective reduction than using the Gramians obtained by solving the generalized Lyapunov equations* (3.4) *and* (3.5).

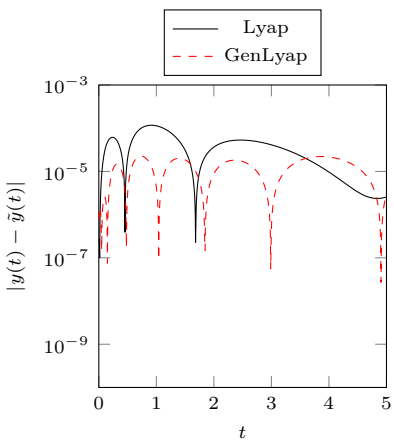

Figure 3.1: Error in the output of the reduced system, which has order $r = 5$

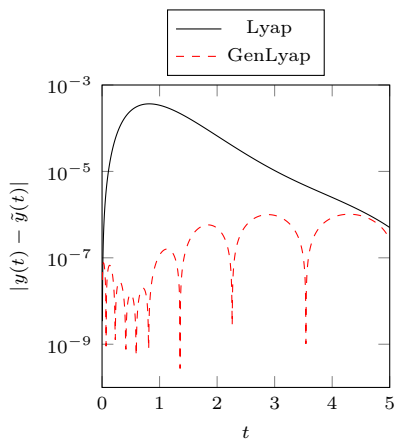

Figure 3.2: Error in the output of the reduced system, which has order $r = 10$

3.3 Sherman-Morrison-Woodbury preconditioner

In this section we assume that the operator $\Pi(X)$ can be formulated as

$$\Pi(X) = \sum_{k=1}^{m} \mathscr{W}_k(X) E_k, \tag{3.6}$$

where, for $k = 1, \ldots, m$, $\mathscr{W}_k(X) : \mathbb{C}^{n\times n} \to \mathbb{C}$ are linear functionals, $E_k \in \mathbb{C}^{n\times n}$ and $m \ll n$. In the context of matrix equations, the approach we present corresponds an enhancement of the Sherman-Morrison-Woodbury (SMW) formula. Given a matrix M and a low-rank matrix UV^T, the SMW formula express the inverse of $M + UV^T$ as low-rank correction of the inverse of M. In our case, we invert the operator $\mathcal{L} + \Pi$ where Π is the low-rank correction. This approach is efficient if the action of $\mathcal{L}^{-1}$ can be efficiently computed. The equation (3.2) can be efficiently solved by only solving $m+2$ Sylvester equations and an $m \times m$ linear system. More precisely, we can express

$$X = \mathcal{L}^{-1}\left(C - \sum_{k=1}^{m} \alpha_k E_k\right), \tag{3.7}$$

where the coefficients $\{\alpha_k\}_{k=1}^{m}$ are the solution to the following linear system:

$$\begin{pmatrix} 1+\mathscr{W}_1(F_1) & \mathscr{W}_1(F_2) & \ldots & \mathscr{W}_1(F_m) \\ \mathscr{W}_2(F_1) & 1+\mathscr{W}_2(F_2) & \ldots & \mathscr{W}_2(F_m) \\ \vdots & \vdots & \ddots & \vdots \\ \mathscr{W}_N(F_1) & \mathscr{W}_m(F_2) & \ldots & 1+\mathscr{W}_N(F_m) \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_m \end{pmatrix} = \begin{pmatrix} \mathscr{W}_1(G) \\ \mathscr{W}_2(G) \\ \vdots \\ \mathscr{W}_m(G) \end{pmatrix}, \tag{3.8}$$

where $G := \mathcal{L}^{-1}(C)$, and $F_k := \mathcal{L}^{-1}(E_k) \qquad k = 1, 2, \ldots, m.$

The preconditioner we developed in Paper II is based on the following argument. The linear system associated to the WEP, i.e., equation (2.7) with fixed λ, can be formulated as $\mathcal{L}(X) + \Phi(X) = C$ where $\Phi(X)$ cannot be represented as (3.6) with small m. The idea is to approximate the operator $\Phi(X)$. Let m be a divisor of n, we consider the partition of the matrix $X \in \mathbb{C}^{n\times n}$ in blocks of size $(n/m) \times (n/m)$ and denote by $X_{i,j}$ the (i,j)-block. We define the operator

$$\mu(X) := \sum_{i,j=1}^{m} \mathscr{W}_{i,j}(X) E_{i,j},$$

where $\mathscr{W}_{i,j}(X)$ is the average of the block $X_{i,j}$, and $E_{i,j}$ is the matrix with ones in the (i,j)-block and zeros otherwise. In Figure 3.3 and Figure 3.4 we display the colormap of a matrix whose entries correspond to the function evaluation, of a specific test function, in a equispaced grid, and the action of $\mu(\cdot)$ on that specific matrix. Then we define $\Pi(X)$, which we aim to be an approximation to $\Phi(X)$, as

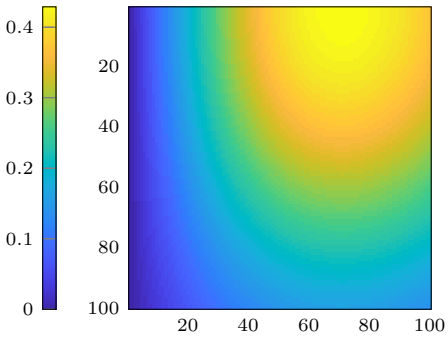

Figure 3.3: Colormap of the matrix X resulting from the evaluation of the function $f(x, y) = xe^{-x^2-y^2}$ in the uniformly discretized domain $[0, 1] \times [0, 1]$.

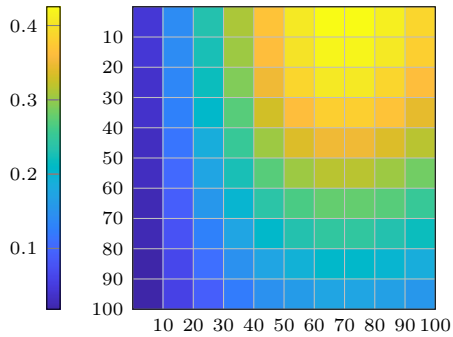

Figure 3.4: Colormap of $\mu(X)$ where X is the matrix in Figure 3.3.

$$\Pi(X) := (\Phi \circ \mu)(X) = \sum_{i,j=1}^{m} \mathscr{W}_{i,j}(X)\Phi(E_{i,j}). \tag{3.9}$$

In the context of the WEP, or in general when the matrix equation comes from the discretization of a PDE, the matrix X corresponds to the evaluation of a function in a discretized domain. Therefore, the elements of X are expected to have a slow/regular variation. In this scenario, $\mu(X)$ is a reasonable approximation to X and then Π is an approximation to Φ. Therefore, the matrix equation where Φ is approximated by Π, defined in (3.9), can be efficiently solved with the SMW formula (3.7) - (3.8) and, for m large enough, it is expected to be a good preconditioner for the original problem. In Figure 3.5 the convergence history of the described preconditioner, combined with gmres, applied to the WEP is shown. In Figure 3.6 the spectrum of the preconditioned operator is shown. These two figures are extracted from Paper II.

3.4 Low rank commuting generalized Sylvester equations

We now assume that $\rho(\mathcal{L}^{-1}\Pi) < 1$ and that $r \ll n$. Under these assumptions, as showed in Paper V, the solution to (3.2) has good low rank approximability properties and can be expressed as a converging *Neumann series* as

$$X = \sum_{j=0}^{\infty} Y_j, \qquad \text{where} \qquad \begin{cases} Y_0 & := \mathcal{L}^{-1}(C), \\ Y_{j+1} & := -\mathcal{L}^{-1}(\Pi(Y_j)), \end{cases} \qquad j \geq 0. \tag{3.10}$$

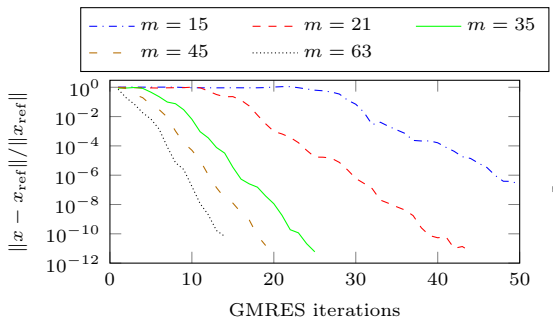

Figure 3.5: Convergence history of gmres when solving the WEP problem with the preconditioner described in Section 3.3. The domain is discretized with 945 *points in each direction.*

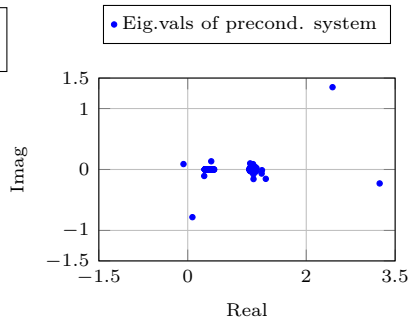

Figure 3.6: Spectrum of the preconditioned operator $(\mathcal{L}+\Pi)^{-1}(\mathcal{L}+\Phi)$ *described in Section 3.3 for* $m = 35$*. The domain is discretized with* 945 *points in each direction.*

We combine this result with Krylov-type methods for a specific structure of Π. More precisely, we assume that the operator Π can be decomposed as

$$\Pi(X) = \sum_{k=1}^{m} N_k X M_k^T, \qquad m \ll n. \tag{3.11}$$

This problem can be efficiently solved if the matrices $\{M_k, N_k\}_{k=1}^m$ and C have low rank [10]. We consider a more general assumption, namely a low-rank commutation property of the matrix coefficients. More precisely, let $[A, B] := AB - BA$ denote the *commutator* of two matrices, we assume that there exist $U_i, \tilde{U}_i \in \mathbb{R}^{n\times s_i}$ and $Q_i, \tilde{Q}_i \in \mathbb{R}^{n\times t_i}$ such that $s_i, t_i \ll n$ and the commutators fulfill

$$\begin{array}{rcccl} [A, N_k] & = & AN_k - N_k A & = & U_k \tilde{U}_k^T, \\ [B, M_k] & = & BM_k - M_k B & = & Q_k \tilde{Q}_k^T, \end{array} \qquad \text{for } k = 1, \ldots, m. \tag{3.12}$$

Under these assumptions, in Paper V we present an algorithm which consists of approximating the solution to (3.2) with a projection method. More precisely we use the block extended Krylov method, which corresponds to using, as projection spaces, $E\mathcal{K}_q^{\square}(A, \bar{C}_1)$ and $E\mathcal{K}_q^{\square}(B, \bar{C}_2)$ where the starting blocks $\bar{C}_1, \bar{C}_2 \in \mathbb{C}^{n\times\tilde{r}}$ have good approximation properties and $\tilde{r} \ll n$. The extended Krylov method can be applied to (3.2) in the following way. Let V_q and W_q the orthogonal basis of $E\mathcal{K}_q^{\square}(A, \bar{C}_1)$ and $E\mathcal{K}_q^{\square}(B, \bar{C}_2)$, the Galerkin orthogonality condition give the projected problem

$$\tilde{A}Z + Z\tilde{B}^T + \sum_{k=1}^{m} \tilde{N}_k Z \tilde{M}_k^T = \tilde{C}_1 \tilde{C}_2^T \tag{3.13}$$

where $\tilde{A} = V_q^T A W_q$, $\tilde{B} = V_q^T B W_q$, $\tilde{N}_k = V_q^T N_k W_q$ $\tilde{M}_k = V_q^T M_k W_q$ for $k = 1, \ldots, m$. The approximation of the solution to (3.2) is then given by $\tilde{X} = (V_q Z_1)(W_q Z_2)^T$ where Z_1, Z_2 are the factors of the LU-factorization of Z. The projected problem (3.13) can be solved with a direct method, e.g., by vectorization, which corresponds to solving a linear system. This approach is efficient only if the starting blocks have good approximation propertied. We consider the construction of the two starting blocks $\bar{C}_1, \bar{C}_2$ based by this argument.

- The matrix $X^{(\ell)}$, obtained by truncating the series (3.10) with ℓ terms, gives an approximation to the solution to (3.2).
- The coefficients Y_j of the Neumann series (3.10) satisfy a sequence of Sylvester equations where, for each Sylvester equation, the right-hand side depends on the solution to the previous Sylvester equation. We consider approximate solutions to this sequence. More precisely, let $\tilde{Y}_j$ be the result of the extended Krylov method applied to each Sylvester equation (3.10).
- The matrix $\tilde{X}^{(\ell)} = \sum_{j=0}^{\ell} \tilde{Y}_j$, which can be viewed as an approximation to the solution of (3.2), can be factorized as $\tilde{X}^{(\ell)} = LR^T$ such that $\mathrm{Range}(L) \subseteq E\mathcal{K}^{\square}_{(\ell+1)k}(A, \tilde{C}_1^{(\ell)})$ and $\mathrm{Range}(R) \subseteq E\mathcal{K}^{\square}_{(\ell+1)k}(B, \tilde{C}_2^{(\ell)})$. We give a characterization and a procedure for computing $\tilde{C}_1^{(\ell)}, \tilde{C}_2^{(\ell)} \in \mathbb{C}^{n \times \tilde{r}}$. A condition for these matrices to have few columns, i.e., $\tilde{r} \ll n$, concerns the commutators (3.12) being low-rank. These two matrices are used as starting blocks in the block extended Krylov method. Namely, fixed ℓ, we select as starting blocks $\bar{C}_1 = \tilde{C}_1^{(\ell)}$ and $\bar{C}_2 = \tilde{C}_2^{(\ell)}$.

Although the above reasoning is based on solving a sequence of Sylvester equations, our approach consists of applying the block extended Krylov method only one time, directly to the generalized Sylvester equation (3.2). In particular, in Theorem 3.6 of Paper V, for a fixed ℓ, we give a procedure for computing the starting blocks $\tilde{C}_1^{(\ell)}$ and $\tilde{C}_2^{(\ell)}$ such that the extended Krylov method approximates $X^{(\ell)}$. In general, the larger ℓ is the larger is the number of columns of the starting blocks. The following theorem is a simplified version of Theorem 3.6 of Paper V (for $m = 1$ in (3.11)).

Theorem 3.4.1. *Consider the generalized Sylvester equation*

$$AX + XB^T + NXM^T = C_1 C_2^T$$

such that $[A, N] = U\tilde{U}^T$ *and* $[B, M] = Q\tilde{Q}^T$. *Let* $\tilde{Y}_i$ *be the low–rank numerical solution of*

$$\begin{aligned} AY_0 + Y_0 B^T &= C_1 C_2^T, \\ AY_{j+1} + Y_{j+1} B^T &= N\tilde{Y}_j M^T, \qquad j = 0, \ldots, \ell - 1, \end{aligned}$$

obtained with the extended Krylov method with k iterations. Let $\tilde{X}^{(\ell)} := \sum_{j=0}^{\ell} (-1)^j \tilde{Y}_j$. *Then, there exist* $S_1 \in E\mathcal{K}^{\square}_{(\ell+1)k}(A, \tilde{C}_1^{(\ell)})$ *and* $S_2 \in E\mathcal{K}^{\square}_{(\ell+1)k}(B, \tilde{C}_2^{(\ell)})$ *such that*

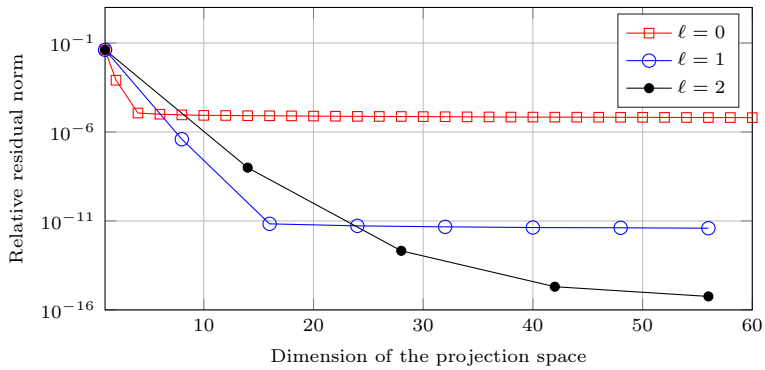

Figure 3.7: Convergence history of the extended Krylov method applied to a (randomly generated) generalized Lyapunov equation $AX + XA + NXN = cc^T$ with A circulant and N sum of a circulant matrix plus a rank one uu^T correction. The algorithm is tested for the starting blocks $\tilde{C}_1 = \tilde{C}_2 = \hat{C}^{(\ell)}$ with $\ell = 0, 1, 2$ selected according to Theorem 3.4.1. This results in the blocks $\hat{C}^{(0)} = c$, $\hat{C}^{(1)} = (c, Nc, u)$ and $\hat{C}^{(2)} = (c, Nc, N^2c, u, Nu)$.

$\tilde{X}^{(\ell)} = S_1 S_2^T$, *where* $C_1^{(0)} = C_1$, $C_2^{(0)} = C_2$ *and for* $\ell > 0$

$$\tilde{C}_1^{(\ell)} = [C_1, NC_1, \ldots, N^\ell C_1, U, NU, \ldots, N^{\ell-1}U],$$
$$\tilde{C}_2^{(\ell)} = [C_2, MC_2, \ldots, M^\ell C_2, Q, MQ, \ldots, M^{\ell-1}Q].$$

In Figure 3.7, extracted from Paper V, we show the effectiveness of this approach. In this specific problem, smaller ℓ gives a faster convergence but bigger error. Therefore the choice of this parameter is application dependent.

Thesis contribution

The scientific contribution of the research presented in this thesis can be summarized as follows. The new method TIAR for solving NEPs, and its restarting variant, is derived and analyzed in Paper I and Paper III. TIAR belongs to the class of memory-efficient Krylov approaches. These methods have received much attention in recent literature due to their robustness and good convergence properties. In Paper I, TIAR was derived, analyzed and used to solve the WEP. This problem arises from a FEM discretization of a partial differential equation, which is used in the study of waves propagating in periodic mediums. The DtN-map is used to reduce the domain of this PDE, which is unbounded, to a bounded domain. In our settings, we are able to establish an explicit characterization of the DtN-map, which allows us to incorporate the structure at an algorithmic level in TIAR. The exploitation of the structure improves the performance of the resulting method drastically. Another Krylov method for NEP, named ILAN, is presented in Paper VI. This method is applicable to a specific class of NEPs, namely those that are symmetric. ILAN is equivalent to applying the indefinite Lanczos method to a symmetrized companion matrix, which is infinite dimensional. Due to the structure exploitation, this method results faster then several methods that do not take the structure into account. The loss of orthogonality, which usually occurs in many Lanczos-type methods, is cured by using a more robust eigenpair extraction, which is based on the companion-structure of the infinite matrix and on the nonlinear Rayleigh–Ritz procedure.

In Paper III, we show that certain well-established methods, such as *method of successive linear problems* and *residual inverse iteration*, can be seen as quasi-Newton methods. Therefore, the convergence properties of these methods can be studied by using the convergence analysis framework of quasi-Newton methods. We characterize the convergence rate of such methods by combining the standard approach used for the analysis of quasi-Newton methods with the Keldysh's theorem. In this spirit, we present one more quasi-Newton method, which has the same converge properties of residual inverse iteration, but does not require the solution

of a nonlinear equation per iteration. On the software side, we have developed NEP-PACK, which is a registered Julia package for NEPs. This package includes: a framework for efficiently representing many NEPs, several state-of-the-art methods and a large problem collection. There are several benchmark problems but also more challenging application derived problems.

We approached the topic of matrix equations while working on Paper I. More precisely, at each iteration of TIAR, a linear system has to be solved. The matrix defining this linear system is unchanged during the iterations. For efficiency reasons, a factorization of that matrix needs to be precomputed. In the context of WEP, the factorization of this matrix is the bottleneck of the method in terms of memory requirements. However, in Paper II, we show that these linear systems can be formulated as generalized Sylvester equations. We develop a preconditioner which is based on approximating that generalized Sylvester equation with another generalized Sylvester equation defined by the Sylvester operator with a low-rank correction. We show how this last problem can be efficiently solved and used as preconditioner in iterative methods such as gmres, BiCGstab, etc. We do not only investigate the quality of this preconditioner in terms of solvability of linear systems, but we also integrate this approach in a NEP solver.

We have explored other ways of solving generalized Sylvester equations. More precisely, instead of applying a Krylov method to the associated linear system, we apply a Krylov method directly to the matrix equation. Krylov methods for matrix equations have also been extensively studied. We focus on a specific class of generalized Sylvester equations in which the matrix coefficients low-rank commute, in the sense that their commutator has low rank. This feature is present in many applications such as discretized PDEs, but also in control theory. We derive a Krylov method that exploits the low-rank commutation property and we prove a result concerning the low-rank approximability of the solution to this matrix equation. The low-rank approximability is a necessary condition for the success of Krylov methods in the context of matrix equations.

Author contribution

Several of the manuscripts presented in this thesis are the result of scientific collaborations. Most of the theoretical results and ideas where produced during meeting and discussions. Here it is stated, for each manuscript and each project, in which part and form the author of this thesis has contributed.

Paper I The ideas were developed in a close collaboration with the co-authors. The author of this thesis was responsible for the implementation, numerical experiments and writing Section 5 (Numerical experiments), Appendix B, Appendix C and contributed to the writing of the rest of the manuscript.

Paper II The ideas were developed in a close collaboration with the co-authors.

Paper III The author of this thesis derived, implemented and analyzed the methods presented in this paper and wrote the final version of the manuscript. The main ideas were discussed and developed with the co-author.

Paper VI The ideas were developed in a close collaboration with the co-authors. The author of this thesis contributed writing: Section 2 (Explicit reformulations of the quasi-Newton methods), Section 3 (Local convergence analysis), Section 5 (Numerical simulations) and designed part of the numerical experiments.

Paper V All the ideas were developed during meetings and discussions. The author of this thesis wrote the final version of the manuscript.

Paper VI The author of this thesis is the only author of Paper VI.

NEP-PACK The following methods were implemented, in large part, by the author of this thesis: `iar, iar_chebyshev, tiar, ilan`. Moreover there

was a contribution in many other parts of the code and on the design of the whole package.

Bibliography

[1] A. Amiraslani, R. M. Corless, and P. Lancaster. Linearization of matrix polynomials expressed in polynomial bases. *IMA J. Numer. Anal.*, 29(1):141–157, 2008.

[2] A. C. Antoulas. *Approximation of large-scale dynamical systems*, volume 6. SIAM, 2005.

[3] J. C. Araujo-Cabarcas. *Reliable hp finite element computations of scattering resonances in nano optics*. PhD thesis, Umeå Universitet, 2019.

[4] J. Asakura, T. Sakurai, H. Tadano, T. Ikegami, and K. Kimura. A numerical method for polynomial eigenvalue problems using contour integral. *Japan J. Ind. Appl. Math.*, 27(1):73–90, 2010.

[5] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst. *Templates for the solution of algebraic eigenvalue problems: a practical guide*. SIAM, 2000.

[6] Z. Bai and D. Skoogh. A projection method for model reduction of bilinear dynamical systems. *Linear Algebra Appl.*, 415(2-3):406–425, 2006.

[7] Z. Bai and Y. Su. SOAR: A second-order Arnoldi method for the solution of the quadratic eigenvalue problem. *SIAM J. Matrix Anal. Appl.*, 26(3):640–659, 2005.

[8] J. Baker, M. Embree, and J. Sabino. Fast singular value decay for Lyapunov solutions with nonnormal coefficients. *SIAM J. Matrix Anal. Appl.*, 36(2):656–668, 2015.

[9] G. Bao. Finite element approximation of time harmonic waves in periodic structures. *SIAM J. Numer. Anal.*, 32(4):1155–1169, 1995.

[10] P. Benner and T. Breiten. Low rank methods for a class of generalized Lyapunov equations and related issues. *Numer. Math.*, 124(3):441–470, 2013.

[11] P. Benner and T. Damm. Lyapunov equations, energy functionals, and model order reduction of bilinear and stochastic systems. *SIAM J. Control Optim.*, 49(2):686–711, 2011.

[12] M. Benzi, G. H. Golub, and J. Liesen. Numerical solution of saddle point problems. *Acta Numer.*, 14:1–137, 2005.

[13] T. Betcke, N. J. Higham, V. Mehrmann, C. Schröder, and F. Tisseur. NLEVP: A collection of nonlinear eigenvalue problems. *ACM Trans. Math. Softw.*, 39(2):7, 2013.

[14] T. Betcke and H. Voss. A Jacobi–Davidson-type projection method for nonlinear eigenvalue problems. *Future Generation Computer Systems*, 20(3):363–372, 2004.

[15] W.-J. Beyn. An integral method for solving nonlinear eigenvalue problems. *Linear Algebra Appl.*, 436(10):3839–3863, 2012.

[16] W.-J. Beyn, C. Effenberger, and D. Kressner. Continuation of eigenvalues and invariant pairs for parameterized nonlinear eigenvalue problems. *Numer. Math.*, 119(3):489, 2011.

[17] W. J. Beyn and V. Thümmler. Continuation of invariant subspaces for parameterized quadratic eigenvalue problems. *SIAM J. Matrix Anal. Appl.*, 31(3):1361–1381, 2009.

[18] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah. Julia: A fresh approach to numerical computing. *SIAM rev.*, 59(1):65–98, 2017.

[19] D. S. Bindel and S. Govindjee. Elastic PMLs for resonator anchor loss simulation. *Int. J. Numer. Methods Eng.*, 64(6):789–818, 2005.

[20] M. Botchev, G. Sleijpen, and A. Sopaheluwakan. An SVD-approach to Jacobi–Davidson solution of nonlinear Helmholtz eigenvalue problems. *Linear Algebra Appl.*, 431(3-4):427–440, 2009.

[21] T. Breiten and T. Damm. Krylov subspace methods for model order reduction of bilinear control systems. *Systems Control Lett.*, 59(8):443–450, 2010.

[22] T. Breiten, V. Simoncini, and M. Stoll. Low-rank solvers for fractional differential equations. *Electron. Trans. Numer. Anal.*, 45:107–132, 2016.

[23] M. Bueno, K. Curlett, and S. Furtado. Structured strong linearizations from Fiedler pencils with repetition i. *Linear Algebra Appl.*, 460:51–80, 2014.

[24] M. Bueno and S. Furtado. Structured strong linearizations from Fiedler pencils with repetition ii. *Linear Algebra Appl.*, 463:282–321, 2014.

[25] C. Campos and J. E. Roman. NEP: a module for the parallel solution of nonlinear eigenvalue problems in slepc. *arXiv:1910.11712*, 2019.

[26] T. Damm. Direct methods and ADI-preconditioned Krylov subspace methods for generalized Lyapunov equations. *Numer. Linear Algebra Appl.*, 15(9):853–871, 2008.

[27] T. Damm and D. Hinrichsen. Newton's method for a rational matrix equation occurring in stochastic control. *Linear Algebra Appl.*, 332:81–109, 2001.

[28] F. De Terán, F. M. Dopico, and D. S. Mackey. Fiedler companion linearizations and the recovery of minimal indices. *SIAM J. Matrix Anal. Appl.*, 31(4):2181–2204, 2010.

[29] J. W. Demmel. *Applied numerical linear algebra*, volume 56. SIAM, 1997.

[30] P. Deuflhard. *Newton methods for nonlinear problems: affine invariance and adaptive algorithms*, volume 35. Springer Science & Business Media, 2011.

[31] C. Effenberger. *Robust solution methods for nonlinear eigenvalue problems.* PhD thesis, EPF Lausanne, 2013.

[32] C. Effenberger. Robust successive computation of eigenpairs for nonlinear eigenvalue problems. *SIAM J. Matrix Anal. Appl.*, 34(3):1231–1256, 2013.

[33] C. Effenberger and D. Kressner. Chebyshev interpolation for nonlinear eigenvalue problems. *BIT*, 52(4):933–951, 2012.

[34] C. Effenberger, D. Kressner, and C. Engström. Linearization techniques for band structure calculations in absorbing photonic crystals. *Int. J. Numer. Methods Eng.*, 89(2):180–191, 2012.

[35] C. Engström. Spectral approximation of quadratic operator polynomials arising in photonic band structure calculations. *Numer. Math.*, 126(3):413–440, 2014.

[36] H. Faßbender and P. Saltenberger. Block Kronecker ansatz spaces for matrix polynomials. *Linear Algebra Appl.*, 542:118–148, 2018.

[37] S. Fliss. A Dirichlet-to-Neumann approach for the exact computation of guided modes in photonic crystal waveguides. *SIAM J. Sci. Comput.*, 35(2):B438–B461, 2013.

[38] S. W. Gaaf and E. Jarlebring. The infinite bi-Lanczos method for nonlinear eigenvalue problems. *SIAM J. Sci. Comput.*, 39(5):S898–S919, 2017.

[39] C. K. Garrett, Z. Bai, and R.-C. Li. A nonlinear QR algorithm for banded nonlinear eigenvalue problems. *ACM Trans. Math. Softw.*, 43(1):4, 2016.

[40] L. Grasedyck. Existence and computation of low Kronecker-rank approximations for large linear systems of tensor product structure. *Computing*, 72(3-4):247–265, 2004.

[41] S. Güttel and F. Tisseur. The nonlinear eigenvalue problem. *Acta Numer.*, 26:1–94, 2017.

[42] S. Güttel, R. Van Beeumen, K. Meerbergen, and W. Michiels. NLEIGS: A class of fully rational Krylov methods for nonlinear eigenvalue problems. *SIAM J. Sci. Comput.*, 36(6):A2842–A2864, 2014.

[43] V. Hernandez, J. E. Roman, and V. Vidal. SLEPc: A scalable and flexible toolkit for the solution of eigenvalue problems. *ACM Trans. Math. Softw.*, 31(3):351–362, 2005.

[44] N. J. Higham, G. M. Negri Porzio, and F. Tisseur. An updated set of nonlinear eigenvalue problems. 2019.

[45] H. Homeier. A modified Newton method for rootfinding with cubic convergence. *J. Comput. Appl. Math.*, 157(1):227–230, 2003.

[46] E. Jarlebring. Convergence factors of Newton methods for nonlinear eigenvalue problems. *Linear Algebra Appl.*, 436(10):3943–3953, 2012.

[47] E. Jarlebring. Broyden's method for nonlinear eigenproblems. *SIAM J. Sci. Comput.*, 41(2):A989–A1012, 2019.

[48] E. Jarlebring, M. Bennedich, G. Mele, E. Ringh, and P. Upadhyaya. NEP-PACK: A Julia package for nonlinear eigenproblems-v0. 2. *arXiv preprint arXiv:1811.09592*, 2018.

[49] E. Jarlebring, K. Meerbergen, and W. Michiels. Computing a partial Schur factorization of nonlinear eigenvalue problems using the infinite Arnoldi method. *SIAM J. Matrix Anal. Appl.*, 35(2):411–436, 2014.

[50] E. Jarlebring and W. Michiels. Analyzing the convergence factor of residual inverse iteration. *BIT*, 51(4):937–957, 2011.

[51] E. Jarlebring, W. Michiels, and K. Meerbergen. A linear eigenvalue algorithm for the nonlinear eigenvalue problem. *Numer. Math.*, 122(1):169–195, 2012.

[52] L. Kaufman. Eigenvalue problems in fiber optic design. *SIAM J. Matrix Anal. Appl.*, 28(1):105–117, 2006.

[53] J. Kou. The improvements of modified Newton's method. *Appl. Math. Comput.*, 189(1):602–609, 2007.

[54] D. Kressner. A block Newton method for nonlinear eigenvalue problems. *Numer. Math.*, 114(2):355–372, 2009.

[55] D. Kressner and J. E. Roman. Memory-efficient Arnoldi algorithms for linearizations of matrix polynomials in Chebyshev basis. *Numer. Linear Algebra Appl.*, 21(4):569–588, 2014.

[56] V. Kublanovskaya. On an approach to the solution of the generalized latent value problem for λ-matrices. *SIAM J. Numer. Anal.*, 7(4):532–537, 1970.

[57] P. Lancaster. *Lambda-matrices and vibrating systems*. Courier Corporation, 2002.

[58] B.-S. Liao, Z. Bai, L.-Q. Lee, K. Ko, et al. Nonlinear Rayleigh-Ritz iterative method for solving large scale nonlinear eigenvalue problems. *Taiwanese J. Math.*, 14(3A):869–883, 2010.

[59] J. Liesen and Z. Strakos. *Krylov subspace methods: principles and analysis*. OUP Oxford, 2012.

[60] P. Lietaert, K. Meerbergen, and F. Tisseur. Compact two-sided Krylov methods for nonlinear eigenvalue problems. *SIAM J. Sci. Comput.*, 40(5):A2801–A2829, 2018.

[61] P. Lietaert, J. Pérez, B. Vandereycken, and K. Meerbergen. Automatic rational approximation and linearization of nonlinear eigenvalue problems. *arXiv preprint arXiv:1801.08622*, 2018.

[62] Y. Lin, L. Bao, and Y. Wei. Order reduction of bilinear MIMO dynamical systems using new block Krylov subspaces. *Comput. Math. Appl.*, 58(6):1093–1102, 2009.

[63] D. S. Mackey, N. Mackey, C. Mehl, and V. Mehrmann. Numerical methods for palindromic eigenvalue problems: Computing the anti-triangular Schur form. *Numer. Linear Algebra Appl.*, 16(1):63–86, 2009.

[64] D. S. Mackey, N. Mackey, C. Mehl, and V. Mehrmanns. Palindromic polynomial eigenvalue problems: Good vibrations from good linearizations. Technical report, DFG Research Center Matheon, "Mathematics for key technologies" in Berlin, TU Berlin, Berlin, Germany, 2005. available online at http://www.matheon.de/.

[65] D. S. Mackey, N. Mackey, C. Mehl, and V. Mehrmanns. Structured polynomial eigenvalue problems: Good vibrations from good linearizations. *SIAM J. Matrix Anal. Appl.*, 28(4):1029–1051, 2006.

[66] V. Mehrmann and H. Voss. Nonlinear eigenvalue problems: A challenge for modern eigenvalue methods. *GAMM-Mitteilungen*, 27(2):121–152, 2004.

[67] V. Mehrmann and D. Watkins. Structure-preserving methods for computing eigenpairs of large sparse skew-Hamiltonian/Hamiltonian pencils. *SIAM J. Sci. Comput.*, 22(6):1905–1925, 2001.

[68] V. Mehrmann and D. Watkins. Polynomial eigenvalue problems with Hamiltonian structure. *Electron. trans. numer. anal.*, 13:106–118, 2002.

[69] W. Michiels and S.-I. Niculescu. *Stability, control, and computation for time-delay systems: an eigenvalue-based approach*, volume 27. SIAM, 2014.

[70] A. Neumaier. Residual inverse iteration for the nonlinear eigenvalue problem. *SIAM J. Numer. Anal.*, 22(5):914–923, 1985.

[71] G. Obinata, B. Anderson, and F. Lutze. Model reduction for control system design, 2001.

[72] D. Palitta. *Numerical solution of large-scale linear matrix equations.* PhD thesis, Università di Bologna, 2018.

[73] D. Palitta. Matrix equation techniques for certain evolutionary partial differential equations. *arXiv preprint arXiv:1908.11851*, 2019.

[74] D. Palitta and V. Simoncini. Matrix-equation-based strategies for convection–diffusion equations. *BIT*, 56(2):751–776, 2016.

[75] B. N. Parlett and H.-C. Chen. Use of indefinite pencils for computing damped natural modes. *Linear Algebra Appl.*, 140:53–88, 1990.

[76] S. Peng, T. Tamir, and H. L. Bertoni. Theory of periodic dielect waveguides. *IEEE Trans. Microw. Theory Tech.*, 23(1):123–133, 1975.

[77] C. E. Powell, D. Silvester, and V. Simoncini. An efficient reduced basis solver for stochastic Galerkin matrix equations. *SIAM J. Sci. Comput.*, 39(1):A141–A163, 2017.

[78] E. Ringh and E. Jarlebring. Nonlinearization of two-parameter eigenvalue problems. *arXiv preprint arXiv:1907.00913*, 2019.

[79] L. Robol, R. Vandebril, and P. V. Dooren. A framework for structured linearizations of matrix polynomials in various bases. *SIAM J. Matrix Anal. Appl.*, 38(1):188–216, 2017.

[80] A. Ruhe. Algorithms for the nonlinear eigenvalue problem. *SIAM J. Numer. Anal.*, 10(4):674–689, 1973.

[81] K. Schreiber. *Nonlinear eigenvalue problems: Newton-type methods and nonlinear Rayleigh functionals.* PhD thesis, TU Berlin, 2008.

[82] S. D. Shank. *Low-rank solution methods for large-scale linear matrix equations*. PhD thesis, Temple University Libraries, 2014.

[83] S. D. Shank, V. Simoncini, and D. B. Szyld. Efficient low-rank solution of generalized Lyapunov equations. *Numer. Math.*, 134(2):327–342, 2016.

[84] V. Simoncini. Computational methods for linear matrix equations. *SIAM Rev.*, 58(3):377–441, 2016.

[85] G. L. Sleijpen, A. G. Booten, D. R. Fokkema, and H. A. Van der Vorst. Jacobi-Davidson type methods for generalized eigenproblems and polynomial eigenproblems. *BIT*, 36(3):595–633, 1996.

[86] A. Spence and C. Poulton. Photonic band structure calculations using nonlinear eigenvalue techniques. *J. Comput. Phys.*, 204(1):65–81, 2005.

[87] O. Steinbach and G. Unger. A boundary element method for the Dirichlet eigenvalue problem of the Laplace operator. *Numer. Math.*, 113(2):281–298, 2009.

[88] D. Stowell and J. Tausch. Variational formulation for guided and leaky modes in multilayer dielectric waveguides. *Commun. Comput. Phys.*, 7(3):564, 2010.

[89] Y. Su and Z. Bai. Solving rational eigenvalue problems via linearization. *SIAM J. Matrix Anal. Appl.*, 32(1):201–216, 2011.

[90] D. B. Szyld and F. Xue. Local convergence analysis of several inexact Newton-type algorithms for general nonlinear eigenvalue problems. *Numer. Math.*, 123(2):333–362, 2013.

[91] J. Tausch. Computing Floquet–Bloch modes in biperiodic slabs with boundary elements. *J. Comput. Appl. Math.*, 254:192–203, 2013.

[92] J. Tausch and J. Butler. Floquet multipliers of periodic waveguides via Dirichlet-to-Neumann maps. *J. Comput. Phys.*, 159(1):90–102, 2000.

[93] F. Tisseur and K. Meerbergen. The quadratic eigenvalue problem. *SIAM rev.*, 43(2):235–286, 2001.

[94] G. Unger. Convergence orders of iterative methods for nonlinear eigenvalue problems. In *Advanced Finite Element Methods and Applications*, pages 217–237. Springer, 2013.

[95] R. Van Beeumen. *Rational Krylov methods for nonlinear eigenvalue problems*. PhD thesis, KU Leuven, 2015.

[96] R. Van Beeumen, K. Meerbergen, and W. Michiels. A rational Krylov method based on Hermite interpolation for nonlinear eigenvalue problems. *SIAM J. Sci. Comput.*, 35(1):A327–A350, 2013.

[97] R. Van Beeumen, K. Meerbergen, and W. Michiels. Compact rational Krylov methods for nonlinear eigenvalue problems. *SIAM J. Matrix Anal. Appl.*, 36(2):820–838, 2015.

[98] H. Voss. An Arnoldi method for nonlinear eigenvalue problems. *BIT*, 44(2):387–401, 2004.

[99] H. Voss. Nonlinear eigenvalue problems. *Handbook of Linear Algebra*, 164, 2013.

[100] F. Xue. A block preconditioned harmonic projection method for large-scale nonlinear eigenvalue problems. *SIAM J. Sci. Comput.*, 40(3):A1809–A1835, 2018.

Part II:
Research Papers and Preprints

Paper I

The waveguide eigenvalue problem and the tensor infinite Arnoldi method

Paper II

Sylvester-based preconditioning for the waveguide eigenvalue problem

Paper III

On restarting the tensor infinite Arnoldi method

Paper IV

Disguised and new quasi-Newton methods for nonlinear eigenvalue problems

Paper V

Krylov methods for low-rank commuting generalized Sylvester equations

Paper VI

The infinite Lanczos method for symmetric nonlinear eigenvalue problems