
KTH ROYAL INSTITUTE
OF TECHNOLOGY

Doctoral Thesis in Applied and Computational Mathematics

# Numerical methods for Sylvester-type matrix equations and nonlinear eigenvalue problems

EMIL RINGH

Stockholm, Sweden 2021

# Numerical methods for Sylvester-type matrix equations and nonlinear eigenvalue problems

EMIL RINGH

Academic Dissertation which, with due permission of the KTH Royal Institute of Technology, is submitted for public defence for the Degree of Doctor of Philosophy on Wednesday May 12th, 2021, at 10:00 a.m. in F3, KTH Royal Institute of Technology, Lindstedtsvägen 26, Stockholm. For digital participation, see instructions at www.kth.se.

Doctoral Thesis in Applied and Computational Mathematics
KTH Royal Institute of Technology
Stockholm, Sweden 2021

ISBN 978-91-7873-812-0
TRITA-SCI-FOU 2021:08

Printed by: Universitetsservice US-AB, Sweden 2021

*To Xin Zhou (周馨)*
*my better half*

*To Axel Ringh*
*my other better half*

**Abstract**

Linear *matrix equations* and *nonlinear eigenvalue problems* (NEP) appear in a wide variety of applications in science and engineering. Important special cases of the former are the Lyapunov equation, the Sylvester equation, and their respective generalizations. These appear, e.g., as *Gramians* to linear and bilinear systems, in computations involving block-triangularization of matrices, and in connection with discretizations of some partial differential equations. The NEP appear, e.g., in stability analysis of time-delay systems, and as results of transformations of linear eigenvalue problems.

This thesis mainly consists of 4 papers that treats the above mentioned computational problems, and presents both theory and methods. In paper A we consider a NEP stemming from the discretization of a partial differential equation describing wave propagation in a waveguide. Some NEP-methods require in each iteration to solve a linear system with a fixed matrix, but different right-hand sides, and with a fine discretization, this linear solve becomes the bottleneck. To overcome this we present a Sylvester-based preconditioner, exploiting the Sherman–Morrison–Woodbury formula.

Paper B treats the generalized Sylvester equation and present two main results: First, a characterization that under certain assumptions motivates the existence of low-rank solutions. Second, a Krylov method applicable when the matrix coefficients are low-rank commuting, i.e., when the commutator is of low rank.

In Paper C we study the generalized Lyapunov equation. Specifically, we extend the motivation for applying the alternating linear scheme (ALS) method, from the stable Lyapunov equation to the stable generalized Lyapunov equation. Moreover, we show connections to $\mathcal{H}_2$-optimal model reduction of associated bilinear systems, and show that ALS can be understood to construct a rank-1 model reduction subspace to such a bilinear system related to the residual. We also propose a residual-based generalized rational-Krylov-type subspace as a solver for the generalized Lyapunov equation.

The fourth paper, Paper D, connects the NEP to the *two-parameter eigenvalue problem*. The latter is a generalization of the linear eigenvalue problem in the sense that there are two eigenvalue-eigenvector equations, both depending on two scalar variables. If we fix one of the variables, then we can use one of the equations, which is then a *generalized eigenvalue problem*, to solve for the other variable. In that sense, the solved-for variable can be understood as a family of functions of the first variable. Hence, it is a variable elimination technique where the second equation can be understood as a family of NEPs. Methods for NEPs can thus be adapted and exploited to solve the original problem. The idea can also be reversed, providing linearizations for certain NEPs.

**Keywords:** Matrix equations, Lyapunov equation, Sylvester equation, nonlinear eigenvalue problems, two-parameter eigenvalue problems, Krylov methods, iterative methods, preconditioning, projection methods

## Sammanfattning

Linjära *matrisekvationer* är en vanligt förekommande variant av linjära ekvationssystem. Viktiga specialfall är *Lyapunovekvationen* och *Sylvesterekvationen*, samt deras respektive generaliseringar. Dessa ekvationer uppstår till exempel som karakteriseringar av *Gramianer* till linjära och bilinjära dynamiska system, i beräkningar som innebär blocktriangularisering av matriser, och vid diskretiseringar av vissa partiella differentialekvationer. Det *icke-linjära egenvärdesproblemet*, från engelskan förkortat NEP, är en generalisering av det linjära egenvärdesproblemet för en matris. I det icke-linjära fallet tillåts matrisens beroende på den skalära parametern att vara just icke-linjärt. Formellt betraktas problemet som en funktion vars definitionsmängd är en delmängd av de komplexa talen, och vars värdemängd är (en delmängd av de) komplexvärda matriserna. Problemet kan beskrivas som att hitta värden, så kallade *egenvärden*, som gör att den tillhörande matrisen i värdemängden är singulär; en vektor i nollrummet kallas för en *egenvektor*. Notera att beroendet på egenvektorn är linjärt. Tillämpningar inkluderar bland annat studier av dynamiska system med tidsfördröjning, samt vid transformationer av linjära egenvärdesproblem. En annan generalisering av egenvärdesproblemet är *två-parameters egenvärdesproblemet* vilket består av två matrisvärda funktioner som båda beror på två parametrar. Målet är att hitta par av parametrar så att båda matriserna är singulära.

Denna avhandling är en sammanläggningsavhandling och består i huvudsak av 4 artiklar. Dessa artiklar berör både praktiska och teoretiska aspekter av de ovan nämnda beräkningsproblemen. I artikel A betraktas ett NEP som härstammar från en partiell differentialekvation, vilken beskriver vågutbredning i en vågledare. På det diskretiserade problemet tillämpas *residual inversiteration* (residual inverse iteration). Metoden kräver att man löser likartade linjära ekvationssystem många gånger, med olika högerled. När diskretiseringen blir noggrannare blir beräkningen av lösningen till det linjära ekvationssystemen en flaskhals. För att komma runt detta presenteras en Sylvester-baserad *förkonditionerare* som utnyttjar Sherman–Morrison–Woodburys formel för invertering av matriser med lågrangstermer.

Artikel B behandlar den generaliserade Sylvesterekvationen och har två huvudresultat: Ett resultat är en karakterisering som under vissa antaganden motiverar existensen av lösningar som kan approximeras med matriser med låg rang. Resultatet är viktigt då många metoder för storskaliga problem har som mål att hitta en approximation av låg rang. Ett annat resultat är en Krylovmetod som kan användas när matriskoefficienterna lågrangskommuterar, d.v.s. när kommutatorn är en matris av låg rang.

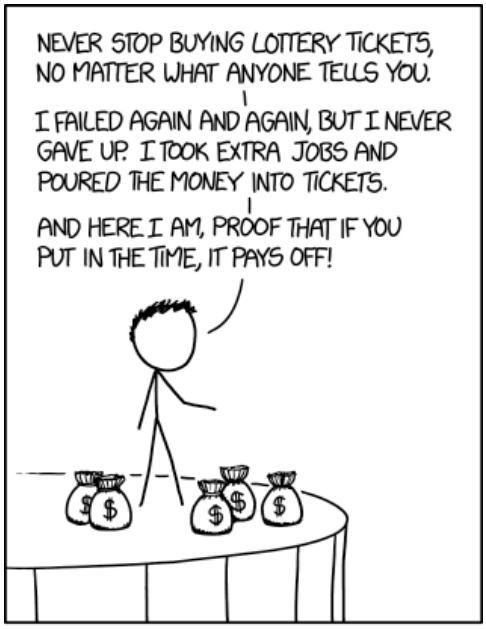
I artikel C undersöker vi den generaliserade Lyapunovekvationen. ALS-metoden, från engelskan *alternating linear scheme*, är en girig algoritm som presenterats i literaturen, och som iterativt utökar approximationen med en matris av rang 1. Denna utökning definieras utifrån att den är ett lokalt minimum av felet, när det senare mäts i en relaterad energinorm. Vi presenterar en utvidgning av den teoretiska motiveringen till användandet av ALS-metoden, från den stabila Lyapunovekvationen till den stabila generaliserade Lyapunovekvationen. Vi visar också på kopplingar till $\mathcal{H}_2$-optimal modellreduktion för bilinjära dynamiska system, och hur rang-1-uppdateringarna i ALS-metoden kan ses som lokalt $\mathcal{H}_2$-optimala till relaterade modellreduktionsproblem. Vi presenterar även varianter av den rationella Krylovmetoden som är anpassade till den generaliserade Lyapunovekvationen.

Den fjärde artikeln, artikel D, presenterar en koppling mellan två-parameters egenvärdesproblemet och NEP. Genom att använda den ena ekvationen för att genomföra en variabeleliminering kan den andra skrivas som en familj av NEP:ar. Elimineringen sker på bekostnad av att ett *generaliserat egenvärdesproblem* behöver lösas för varje funktionsevaluering av NEP:en. Metoder för NEP kan på så sätt anpassas för att lösa två-parameters egenvärdesproblemet. Icke-linjärisering kan även tillämpas i omvänd riktning och kan på så sätt leda till linjäriseringar av vissa NEP:ar.

**Nyckelord:** Matrisekvationer, Lyapunovekvationen, Sylvesterekvationen, icke-linjära egenvärdesproblem, två-parameters egenvärdesproblem, Krylovmetoder, iterativa metoder, förkonditionering, projektionsmetoder

*I have been nothing but lucky.*

*– David Letterman* [i]




*– xkcd* [ii]

# Acknowledgments

I come back to the sentence printed a few pages ago, *"I have been nothing but lucky"*. And have I been lucky! I consider myself fortunate to have been given this opportunity, and it would not have been possible if it were not for a lot of fantastic people. I would like to extend my gratitude to people I have met and who has, in one way or another, helped me on this journey.

First, and foremost, I would like to thank my main supervisor Elias Jarlebring, and my co-supervisors Johan Karlsson and Per Enqvist — Thank you for everything: For your guidance and support. For always keeping your door open and always taking your time with me. For letting me take my time, but without getting too lost. For all the things I have learned from you, both technical and non-technical, professional and personal. I have been lucky. Thank you for everything!

Second, I would like to express my gratitude to former my office mate, academic older brother, and, most importantly, friend, Giampaolo Mele — Not only did your friendship make it a joy to go to the office, you were also an invaluable scientific partner. It was a blessing to have you as a discussion partner. I am still amazed by the structure in our reasoning and the way you could take an argument and formulate it in such a concise and exact, yet graspable, manner. You always challenged me, and you really helped me learn and improve.

I would also like to express my thankfulness to all my collaborators and co-authors. It would not have been the same without you. Davide Palitta — It was great fun having you here, and collaborating with you was a great experience. I enjoyed it a lot. It was also great fun to meet you in different places all over the world. I have learned a lot from you. Tobias Breiten — Thank you for a great collaboration, and for your hospitality during my visit. Your patience, interest, and guidance helped me a lot. I have said it to you before and I will say it again: I believe you will make an excellent supervisor! Parikshit Upadhyaya and Massimiliano Fasi — I enjoyed working with you both, and I believe I have learned something new from each of you. We have seen some nice conferences together, and had both technical and non-technical discussions. Max Bennedich — Working with you was fun, and you showed us a few things about best practice when it comes to software.

I am grateful to the PhD students (past and present) at the department of Mathematics — For being such nice colleagues. We have had technical discussions and also talked about life in general. Names that come to mind as I write this are Michelle Böck, Siobhan Correnty, Isabel Haasler, Federico Izzo, Adam Lindhe, Lovisa Engberg, Silun Zhang, David Ek, Han Zhang, Sara Frimodig, Yibei Li, Göran Svensson, Eric Staffas, Thomas Frachon, Martina Favero, Fredrik Fryklund, Sara Pålsson, Davoud Saffar Shamshirgar, Lena Leitenmaier, Gabriela Malenová, Celia Garcia Pareja,

Nasrin Altafi, Anna Broms, Gerard Farré, Gustav Zickert, Aleksa Stankovic, Hanna Hultin, Petter Restadh, Joar Bagge, Julian Mauersberger, Samuel Fromm, Johan Wärnegård, and Alexander Aurell. (Apologies to anyone I missed here!) You were part my *"Alderland"* here. Thank you to the faculty at the department — The atmosphere has always felt open and the hierarchy flat. To the administration at the department — For keeping things up and running, and for helping me with all my questions. And to the library — For providing access to such a vast set of publications, and helping me dig out paper copies of hard-to-get articles.

I would also like to thank the communities I have been working in, both the control community and the (applied/numerical) linear algebra communities. Thank you to friends and colleagues from all over the world. I moreover owe a lot of thanks to the open source communities — This thesis, as well as my papers, are typeset in LaTeX and the figures generated with PGF and Ti*k*Z; working in Julia has been a pleasant experience; and I probably run a lot of code daily, in both private and professional, that is due to the different communities.

On the personal side I want to thank my family. All my parents and my brothers — For believing in me, and supporting me in doing what I wanted. For everything I have learned from you. The way you share your experience with me. For all the help I have received. For all the love!

I am also immensely grateful to all my friends: Jennifer Lemne, Nabila Ahlgren, Evelina Stadin, Sara Aschan, Tomas Aschan, Frida Halfvarson, Emil Lundberg, Ludvig Hult, Sara Lidbaum, Carl Aronsson, Gustav Johnsson, Chloé Johnsson, Ally (Xiyue) Huang, Jonas Adler, Felix Malmenbeck, Yujiao Wu, and Oksana Goroshchuk — For keeping the work–life balance (somewhat) properly balanced. I treasure all the time we spent together, and I hope to be able to meet you all again soon.

To my musketeers, the Dalton brothers: Axel Ringh, Martin Larsson, Björn Ahlgren — I cannot imagine what I would have done without you guys. It is a lot of fun and games, but also steadfast support in the downs. Thank you for being who you are!

To my wife, Xin Zhou — You are so amazing, one of the most intelligent people I have ever met. Thank you for standing by my side. I love you! ♡

To my daughter, Felicia (静初) Ringh — You are the best. Papa loves you limitlessly and forever!

Solna, March 2021

*Emil Ringh*

# Table of Contents

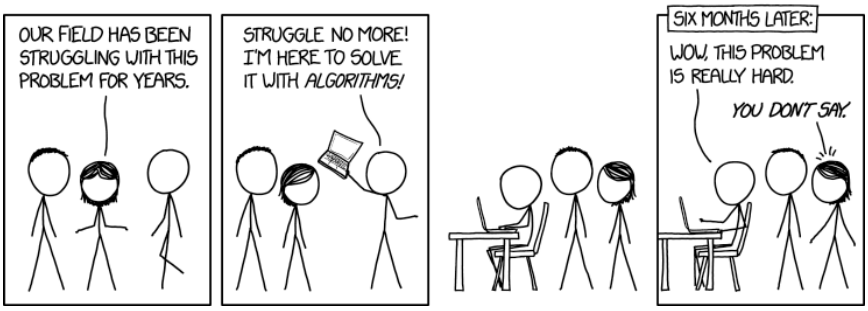
## Abbreviations

| | |
|---|---|
| ADI | Alternating direction implicit (method) |
| ALS | Alternating linear scheme (method) |
| BiCGStab | Biconjugate gradient stabilized method |
| BilADI | Bilinear ADI (method) |
| BIRKA | Bilinear iterative rational Krylov (method) |
| BVP | Boundary value problem |
| CPU | Central processing unit |
| DtN | Dirichlet-to-Neumann (map) |
| FD | Finite difference |
| FEM | Finite element |
| FFT | Fast Fourier transform |
| GEP | Generalized eigenvalue problem |
| GLEK | Generalized Lyapunov Extended Krylov method |
| GMRES | Generalized minimal residual method |
| ILU | Incomplete LU-factorization |
| IRKA | Iterative rational Krylov (method) |
| K-PIK | Krylov-Plus-Inverted-Krylov (method) |
| LMI | Linear matrix inequality |
| MIMO | Multiple input multiple output (system) |
| NEP | Nonlinear eigenvalue problem |
| ODE | Ordinary differential equation |
| PDE | Partial differential equation |
| RAM | Random-access memory |
| RC | Resistor capacitor (circuit) |
| REPL | Read-eval-print loop (Julia) |
| SISO | Single input single output |
| SMW | Sherman–Morrison–Woodbury (formula) |
| SOR | successive over-relaxation (method) |
| SPMF | Sum of products of matrices and functions |
| SVD | Singular value decomposition |
| TRITA | Transactions of the Royal Institute of Technology A |
| WEP | Waveguide eigenvalue problem |

## Copyright notice

*With four parameters I can fit an elephant,*
*and with five I can make him wiggle his trunk.*

*– John von Neumann* [iii]




*– xkcd* [iv]

# Part I:
# Introduction and Preliminaries

# Chapter: 1

# Introduction

This thesis is a so called *compilation thesis*, sometimes referred to as a *thesis by publication*. As such it consists of two main parts: Part I is an introduction and summary, and Part II contains the appended papers. Hence, Part I introduces and summarizes the topics covered in the thesis and puts the included papers in a context. Moreover, it highlights the author's contribution to the papers, as these are written together with co-authors. In contrast, Part II constitutes the main scientific contributions of this work. For Part I there is a common bibliography in the end, and for Part II each paper has its own bibliography, since each paper is self-contained. The appended papers in Part II are, except for some minor editing, the same as the corresponding published papers. An electronic version of the thesis might only contain Part I. However, forward references from Part I should still be possible to follow knowing that the thesis follows the numbering convention: *Paper.Section.Item* for theorems, propositions, lemmas, and remarks; *(Paper.Equation)* for equations; and *Paper.Item* for figures, tables, and algorithms.

Within the scope of this PhD there has also been involvement in the development of a software for nonlinear eigenvalue problems: NEP-PACK, and the writing of a textbook: *Preconditioning for linear systems*. These works are briefly treated in Chapter 3.

## 1.1 Motivation

A common theme throughout this thesis is the development of numerical methods and algorithms, where the goal is often a combination of being able to handle larger problems, faster, with more accuracy, and/or higher robustness. The importance of this task is reflected in that many scientific problems are nowadays approached by computations and simulations, or as a public report put it:

> *"Over the past several decades, simulation has become the third pillar of science, complementing theory and experiment [...]."* [123, pp. 66–67]

Thus, simulation capacity is vital to many parts of science today. Although hardware has developed drastically over the last several decades, the importance of algorithm development should not be underestimated, as illustrated in another report:

> *"Improved algorithms and libraries have contributed as much to increases in capability as have improvements in hardware."* [107, p. 54]

More specifically, this thesis focuses on development in the field of numerical linear algebra. The topic is at the heart of many computations, and improvements are contributing to many other fields, e.g.,

> *"Improvements in linear algebra algorithms are not central to the theory of linear programming, but are nevertheless central to computational progress in this subject."* [25, p. 310]

The field of numerical linear algebra involves many classes of problems, such as, e.g., eigenvalue computations; matrix function evaluations; not to mention solutions to linear systems of equations. In fact:

> *"Many scientific problems lead to the requirement to solve linear systems of equations as part of the computations. From a pure mathematical point of view, this problem can be considered as being solved [...]. The actual computation of the solution(s) may however lead to severe complications, [...]."* [121, p. 1]

Hence, even if some problems are mathematically solved, and maybe since long, the actual application still constitutes a both interesting and important field of research. However, many problems are nonlinear and today nonlinear models are becoming more and more adopted in many fields. Nevertheless, numerical linear algebra will continue to be at the core of many computational problems as many methods for nonlinear problems rely on solving linear subproblems. Thus, the continued development and improvement of methods and algorithms, both inside and outside the field of numerical linear algebra, will remain an important task.

# Chapter: 2

# Preliminaries

Chapter two contains some introductory and well-established results from the literature. The chapter serves as an introduction to the topics discussed in this thesis and the aim is to provide an overview and a foundation for readers who are not familiar with the topic. The chapter also introduces some standard notation. Hence, readers who are experts in the field may, at their convenience, skip this chapter and go directly to Chapter 3 and the appended papers in Part II.

The chapter starts on a basic level, but quickly reaches more advanced topics. It mostly covers the theoretical background. In the outline that follows letters in parenthesis indicates relevance to the corresponding appended paper. Section 2.1 discusses some basic linear algebra and matrix-theoretical results relevant for most of the exposition. Specifically, it contains a part on the *generalized eigenvalue problem* (D) and on the *Kronecker product* (A, B, C). *Matrix functions* are introduced in Section 2.2 since they are relevant for some later analysis. In Section 2.3 *linear matrix equations* (A, B, C) are presented, and the *nonlinear eigenvalue problem* (A, D) is introduced in Section 2.4.

## 2.1 Basic linear algebra and matrix theory

We summarize some standard concepts and results usually found in textbooks, such as, e.g., Horn and Johnson [70], Golub and Van Loan [51], Bellman [14], and Gantmacher [48].

### Notation and initial definitions

The notation used regarding matrices is standard in the literature. We use upper case letters for matrices and lower case letters with subscripts for elements of the matrix, i.e., the matrix $A \in \mathbb{C}^{n\times n}$ has elements $a_{i,j}$ on row $i$ and column $j$. Occasionally we may also

write $[AB]_{i,j}$ which is a way to express $c_{i,j}$ when the matrix $C = AB$ is not named explicitly. Note that lower case letters are also used to denote both vectors, e.g., $x \in \mathbb{C}^n$, and scalars $z \in \mathbb{C}$. However, the dimension will be clear from the context. The *transpose* of the matrix $A$ is denoted $A^T$, and the *Hermitian transpose* $A^H$. The *determinant* is denoted $\det(A)$. We define the *commutator* for two matrices $A, B \in \mathbb{C}^{n\times n}$ as $\mathrm{com}(A, B) := AB - BA$, and say that the two matrices are *commuting* if $\mathrm{com}(A, B) = 0$. Furthermore, a matrix is called *normal* if it commutes with its Hermitian transpose, i.e., $AA^H = A^HA$. We say that a matrix, possibly rectangular, has orthogonal columns if $A^TA = I$, and we call a square matrix *orthogonal* if $A^TA = AA^T = I$ and *unitary* if $A^HA = AA^H = I$.

Two subspaces associated with a matrix $A \in \mathbb{C}^{n\times m}$ are the range and the kernel. The *range* is the subspace of vectors that is the result of (at least) one vector mapped by $A$, i.e., $\mathrm{range}(A) := \{b \in \mathbb{C}^n : b = Ax \text{ for some } x \in \mathbb{C}^m\}$. The *kernel*, sometimes *nullspace*, is the subspace of vectors that $A$ maps to zero, i.e., $\ker(A) := \{x \in \mathbb{C}^m : Ax = 0\}$. Related to the range is the *span*, which is the subspace of vectors, generated by vectors in a set $\mathcal{A}$, i.e, $\mathrm{span}(\mathcal{A}) := \{b \in \mathbb{C}^n : b = \sum_{i\in\mathcal{I}} \alpha_i x_i \text{ for some } \mathcal{I}, \text{ and where } x_i \in \mathcal{A} \text{ and } \alpha_i \in \mathbb{C} \text{ for } i \in \mathcal{I}\}$.

## Spectral theory

**Definition 2.1.1** (Eigenpair)**.** Let $A \in \mathbb{C}^{n\times n}$. A scalar $\lambda_0 \in \mathbb{C}$ is called an *eigenvalue* if $A - \lambda_0 I$ is singular. An *eigenvector* is a vector $x_0 \in \mathbb{C}^n$ such that $x_0 \neq 0$ and $x_0 \in \ker(A - \lambda_0 I)$, for an eigenvalue $\lambda_0$. The pair $(\lambda_0, x_0)$ is called an *eigenpair*.

Finding an eigenpair $(\lambda, x)$ such that $Ax = \lambda x$, or equivalently

$$(A - \lambda I)x = 0,$$

is referred to as the *eigenvalue problem*, and is an important problem with many applications. The eigenvalue problem has been extensively studied in the field of numerical linear algebra and considerable material is available, e.g., in [51], [6], [120], and [133]. The eigenvalue can be equivalently described as a value $\lambda_0$ such that $\det(A - \lambda_0 I) = 0$, and from the properties of the determinant we see that this is a root-finding problem for a polynomial. The polynomial $p(z) := \det(A - zI)$ is called the *characteristic polynomial*.

**Definition 2.1.2** (Algebraic and geometric multiplicity)**.** Let $A \in \mathbb{C}^{n\times n}$, and let $\lambda_0 \in \mathbb{C}$ be an eigenvalue. The *algebraic multiplicity* of the eigenvalue is defined as the multiplicity of the root $\lambda_0$ to the characteristic polynomial. Moreover, the *geometric multiplicity* of the eigenvalue is defined as the dimension of the kernel of $A - \lambda_0 I$, i.e., $\dim(\ker(A - \lambda_0 I))$.

**Definition 2.1.3** (Spectrum and spectral radius)**.** Let $A \in \mathbb{C}^{n\times n}$. The *spectrum* of $A$ is the set of all eigenvalues, i.e., $\sigma(A) := \{\lambda \in \mathbb{C} : \lambda \text{ is an eigenvalue of } A\}$, and the *spectral radius* of $A$ is the largest modulus of its eigenvalues, i.e., $\rho(A) := \max_{\lambda\in\sigma(A)} |\lambda|$.

**Definition 2.1.4** (Field of values)**.** For a matrix $A \in \mathbb{C}^{n\times n}$ the *field of values*, sometimes called *numerical range*, is defined as

$$W(A) := \{z \in \mathbb{C} : z = x^HAx,\ x \in \mathbb{C}^n,\ x^Hx = 1\}.$$

**Proposition 2.1.5** ([69, Property 1.2.6]). *For a matrix $A \in \mathbb{C}^{n\times n}$, the spectrum is contained in the field of values, i.e., $\sigma(A) \subset W(A)$.*

**Proposition 2.1.6** ([70, Theorem 4.1.4]). *Let $A \in \mathbb{C}^{n\times n}$. The matrix $A$ is Hermitian if and only if $W(A) \subseteq \mathbb{R}$. Hence, if $A$ is Hermitian, then the eigenvalues are real.*

**Definition 2.1.7** (Invariant subspace). Let $A \in \mathbb{C}^{n\times n}$. A subspace $\mathcal{H} \subset \mathbb{C}^n$ is called *invariant* if $Ax \in \mathcal{H}$ for all $x \in \mathcal{H}$.

The term $A$-invariant is sometimes used to highlight that the invariance depends on the matrix in consideration. Examples are different spaces spanned by eigenvectors, i.e., $\mathcal{H} = \{v \in \mathbb{C}^n : v = \sum_{i=1}^{k} \alpha_i x_i, \alpha_i \in \mathbb{C} \text{ and } x_i \text{ eigenvectors of } A \text{ for } i = 1, 2, \ldots, k\}$. Specifically, spaces spanned by a single eigenvector, i.e., $\mathcal{H} = \operatorname{span}\{x_0\}$ where $x_0$ is an eigenvector of $A$, are $A$-invariant.

**Definition 2.1.8** (Definite matrix). A matrix $A \in \mathbb{R}^{n\times n}$ is called *positive definite* if for any $x \in \mathbb{R}^n$ such that $x \neq 0$, then $x^T Ax > 0$. Similarly, we call it *negative definite* if $x^T Ax < 0$. The term positive/negative *semidefinite* is used if the respective inequality is non-strict. A matrix that is not semidefinite is called *indefinite*.

In the complex case the quantity $x^H Ax$ is considered, and required to be real. Hence, according to this definition, in order to talk about definiteness in the complex case it is required that the matrix is Hermitian.[1] A related concept is the following characterization.

**Definition 2.1.9** (Stable matrix). A matrix is called *stable*, sometimes *Hurwitz*, if all the eigenvalues have strictly negative real parts, i.e., $A \in \mathbb{C}^{n\times n}$ is stable if $\operatorname{Re}(\lambda) < 0$ for all $\lambda \in \sigma(A)$.

The name stable originates from that the dynamical system $\dot{x}(t) = Ax(t)$ is stable, i.e., $\lim_{t\to\infty} x(t) = 0$, if and only if $A$ is a stable matrix [5, Theorem 5.14]. There is also a notion of a matrix $A$ being *anti-stable*, which means that $-A$ is stable, or in other words, that all the eigenvalues of $A$ have strictly positive real parts. It follows from Proposition 2.1.5 that negative definiteness is a sufficient condition for stability. Moreover, for symmetric matrices it is also necessary.

**Proposition 2.1.10** ([70, Theorem 4.1.10]). *A symmetric matrix $A \in \mathbb{R}^{n\times n}$ is stable if and only if it is negative definite.*

However, negative definiteness is not a necessary condition for stability if the matrix is non-symmetric. Consider the following counterexample: Let $A = \begin{bmatrix} -1 & \alpha \\ 0 & -2 \end{bmatrix}$, for some $\alpha \in \mathbb{R}$. The matrix $A$ has the eigenvalues $-1$ and $-2$, but for $x = \begin{bmatrix} 1 & 1 \end{bmatrix}^T$, we have that $x^T Ax = \alpha - 3$, which is greater than zero for $\alpha > 3$. Thus, $A$ is stable, but indefinite.

[1] Definition 2.1.8 is adopted in, e.g., [70]. However, there is a weaker definition in which one considers the definiteness of the Hermitian part of the matrix $A$, i.e., the definiteness of $(A + A^H)/2$. The latter is equivalent to considering $\operatorname{Re}(x^H Ax)$.

**Definition 2.1.11** (Similarity)**.** Two matrices $A \in \mathbb{C}^{n\times n}$ and $B \in \mathbb{C}^{n\times n}$ are called *similar* if there exists a nonsingular matrix $S \in \mathbb{C}^{n\times n}$ such that $B = S^{-1}AS$. The matrix $S$ is called the *similarity transform*. Moreover, if $S$ is unitary or orthogonal, then $A$ and $B$ are called *unitarily similar* or *orthogonally similar* respectively.

**Proposition 2.1.12** ([70, Corollary 1.3.4])**.** *Let $A, B \in \mathbb{C}^{n\times n}$. If $A$ and $B$ are similar, then they have the same eigenvalues, i.e., $\sigma(A) = \sigma(B)$.*

**Definition 2.1.13** (Diagonalization)**.** A matrix is called *diagonalizable* if it is similar to a diagonal matrix, and *unitarily diagonalizable* if the similarity transform is unitary.

A necessary condition for a matrix to be diagonalizable is that the eigenvalues are distinct, and a matrix is unitarily diagonalizable if and only if it is normal. However, not all matrices are diagonalizable. Hence, we present a more general type of similarity transformation, that exists for all matrices. To do so we need the following definition.

**Definition 2.1.14** (Jordan block)**.** The following type of matrix is called a *Jordan block*,

$$J_n(\lambda) := \begin{bmatrix} \lambda & 1 & 0 & \dots & 0 & 0 \\ 0 & \lambda & 1 & \dots & 0 & 0 \\ 0 & 0 & \lambda & \ddots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \lambda & 1 \\ 0 & 0 & 0 & \dots & 0 & \lambda \end{bmatrix} \in \mathbb{C}^{n\times n}.$$

The Jordan block $J_n(\lambda)$ has the eigenvalue $\lambda$ with algebraic multiplicity $n$ and geometric multiplicity 1. We may omit the subscript $n$ if the size is clear form the context. A *Jordan matrix* is a block-diagonal matrix whose diagonal blocks all are Jordan blocks. The following result establishes the existence of an important decomposition, and as such also serves as a definition of it.

**Proposition 2.1.15** ([70, Theorem 3.1.11])**.** *For any matrix $A \in \mathbb{C}^{n\times n}$ there exist; a nonsingular $S \in \mathbb{C}^{n\times n}$; positive integers $m$ and $n_i$ for $i = 1, 2, \dots, m$ such that $n = \sum_{i=1}^{m} n_i$; and scalars $\lambda_i$, not necessarily distinct, for $i = 1, 2, \dots, m$; such that*

$$A = SJS^{-1}, \qquad \textit{where} \qquad J := \begin{bmatrix} J_{n_1}(\lambda_1) & 0 & \dots & 0 \\ 0 & J_{n_2}(\lambda_2) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & J_{n_m}(\lambda_m) \end{bmatrix}.$$

*The decomposition is called the* Jordan form*.*

The Jordan form is also known as the *Jordan canonical form*, *Jordan normal form*, and *Jordan decomposition*, and it is unique for any given matrix, up to a permutation of the Jordan blocks. Moreover, if $A \in \mathbb{R}^{n\times n}$ and $\sigma(A) \subset \mathbb{R}$, then the Jordan blocks are real

and we may choose $S \in \mathbb{R}^{n\times n}$. Since $A$ is similar to the Jordan matrix, the decomposition reveals the eigenvalues of $A$ to be the $\lambda$s on the diagonal of $J$. The geometric multiplicity of the eigenvalue $\lambda \in \sigma(A)$ is the number of Jordan blocks such that $\lambda = \lambda_i$, and the algebraic multiplicity is the sum of the numbers $n_i$ such that $\lambda = \lambda_i$. The size of the largest Jordan block such that $\lambda = \lambda_i$ is the eigenvalue multiplicity in the minimal polynomial.

In practice the following decomposition may be more useful. Likewise, the proposition asserts the existence, and as such also serves as the definition.

**Proposition 2.1.16** ([70, Theorem 2.3.1])**.** *Any matrix $A \in \mathbb{C}^{n\times n}$ is unitarily similar to an upper triangular matrix, i.e., there exists a unitary matrix $Q \in \mathbb{C}^{n\times n}$ and an upper triangular matrix $T_A \in \mathbb{C}^{n\times n}$, such $A = QT_AQ^H$. The relation is called a* Schur decomposition*.*

A *Schur decomposition* is also known as a *Schur factorization*, or some times a *Schur triangularization*. We note that since it is a similarity transformation and the matrix is triangular, a Schur decomposition reveals the eigenvalues of the matrix $A$.

### The generalized eigenvalue problem

We introduce an extension of the eigenvalue problem defined above. In the *generalized eigenvalue problem* we have two matrices $A$ and $B$, and want to find an eigenpair $(\lambda, x)$ such that

$$Ax = \lambda Bx. \tag{2.1}$$

**Definition 2.1.17** (Pencil)**.** The pair $(A, B)$ associated with (2.1) is called a (matrix) *pencil*. The notation $A - zB$ is also commonly used in the literature to denote the pencil.

First, we observe that $B = I$ reduces the problem to the (classical) eigenvalue problem. Moreover, if $B$ is invertible, then we have the equivalent problem $B^{-1}Ax = \lambda x$. Second, we note that the introduction of the matrix $B$ can make the problem more complicated, e.g.: If $B$ is rank deficient then the *characteristic polynomial* $p(z) := \det(A - zB)$ does not have degree $n$, and hence not $n$ roots; or, if $x$ is a vector in the kernel of both $A$ and $B$, i.e., $x \in \ker(A)$ and $x \in \ker(B)$, then (2.1) is fulfilled for all values $\lambda$; and in general, the matrices in problem (2.1) does not even need to be square. Some of these complications give rise to the following definition.

**Definition 2.1.18** (Regular and singular)**.** The pencil $(A, B)$ is called *regular* if $A$ and $B$ are square, and there exists $z \in \mathbb{C}$ such that the characteristic polynomial is not identically zero, i.e., $\det(A - zB) \neq 0$ . A pencil that is not regular is called *singular*.

A regular pencil is more well-behaved. For instance, it excludes the cases when $A - zB$ is rank deficient for all $z$ which happens, e.g., when $A$ and $B$ have a common kernel. Thus, we can define an eigenpair similarly as before.

**Definition 2.1.19** (Eigenvalue)**.** Let $(A, B)$ be a regular pencil. A scalar $\lambda \in \mathbb{C}$ is called a (generalized) *eigenvalue* if $A - \lambda B$ is singular. A (generalized) *eigenvector* is a vector

$x \in \mathbb{C}^n$ such that $x \neq 0$ and $x \in \ker(A - \lambda B)$, for an eigenvalue $\lambda$. The pair $(\lambda, x)$ is called a (generalized) *eigenpair*.

Moreover, we say that $\infty$ is an eigenvalue of the pencil $(A, B)$ if 0 is an eigenvalue of the pencil $(B, A)$, i.e., if there is a solution $1/\lambda = \mu = 0$ to $Bx = \mu Ax$.

The infinite eigenvalues originates from rank-deficiency of $B$, and the definition given above of $\infty$ as an eigenvalue is equivalent to another common definition, found in, e.g., [6, Section 2.6]. This equivalent second definition states that: If the degree of the characteristic polynomial is $d$, then the pencil has $n - d$ infinite eigenvalues.

For a singular pencil, the definition is somewhat different; [6, Sections 2.6.9 and 8.7]. The definition of an eigenvalue in Definition 2.1.19 is a special case of the following definition. However, note that the following definition does not define eigenvectors.

**Definition 2.1.20** (Eigenvalue)**.** Let $(A, B)$ be a (singular) pencil. A scalar $\lambda \in \mathbb{C}$ is called a (generalized) *eigenvalue* if $A - \lambda B$ has lower rank than $A - zB$ for almost all values $z \in \mathbb{C}$.

**Definition 2.1.21** (Spectrum)**.** The *spectrum* of the pencil $(A, B)$ is the set of all eigenvalues, i.e., $\sigma(A, B) := \{\lambda \in \mathbb{C} : \lambda \text{ is an eigenvalue of } (A, B)\}$.

In this notation we have that $\sigma(A) = \sigma(A, I)$. Moreover, we noted above that if $A$ and $B$ are square, and $B$ is nonsingular, then there is an equivalent (linear) eigenvalue problem. The observation entails that $\sigma(A, B) = \sigma(B^{-1}A)$, when $B$ is nonsingular.

The generalized eigenvalue problem can be equivalently described as finding a triplet $(\alpha, \beta, x)$ such that

$$\beta Ax = \alpha Bx.$$

The eigenvalues are then formally denoted by the tuple $(\alpha, \beta)$, sometimes as $\alpha/\beta$. Moreover, if $\beta \neq 0$, then the eigenvalue $\lambda$ from (2.1) is $\lambda = \alpha/\beta$. Hence, we can transform between the two different ways of describing the eigenvalues. Moreover, $\beta = 0$ (with $\alpha \neq 0$) corresponds to an eigenvalue being $\infty$ according to our definition. To avoid confusion and technical difficulties with the $(\alpha, \beta)$ notation, uniqueness has to be assured. It can be achieved either by imposing a normalization $|\alpha|^2 + |\beta|^2 = 1$, or by considering equivalence classes of quotients as in [30]. The description in terms of the tuple $(\alpha, \beta)$ has the advantage that it can naturally describe cases when $\beta = 0$ which is advantageous, e.g., to avoid over/underflow in computations [6, Section 8.4] and in terms of conditioning see [6, Section 2.6.5] and [51, Section 7.7.3]. An eigenvalue given by $\alpha = \beta = 0$ is called an *indeterminate eigenvalue*, and corresponds to a singular pencil; see [6, Section 8.7.1].

**Proposition 2.1.22.** *Let $A, B \in \mathbb{C}^{n\times n}$ be given and assume that the pencil $(A, B)$ is regular. Let $(\lambda_0, x_0) \in \mathbb{C} \times \mathbb{C}^m$ be an eigenpair of the generalized eigenvalue problem*

$$Ax = \lambda Bx.$$

*Then $Bx_0 \neq 0$.*

*Proof.* The proof is by contradiction. Assume that $Bx_0 = 0$. The cases $Ax_0 = 0$ and $Ax_0 \neq 0$ are investigated separately.

Assume that $Ax_0 = 0$, then $x_0$ is in the kernel of $A$ and $B$. Hence, $\det(A - \lambda B) = 0$ for all values of $\lambda$, which contradicts that the matrix pencil is regular.

Assume that $Ax_0 \neq 0$, then $\beta_0 Ax = 0$ implies that $\beta_0 = 0$. The case $\alpha_0 = 0$ implies that the pencil is singular, which contradicts that it is regular. The case $\alpha_0 \neq 0$ fulfills the definition of an eigenvalue at infinity which contradicts $\lambda_0 \in \mathbb{C}$. □

Similar to that eigenvalues can be computed with a Schur decomposition, eigenvalues of the generalized eigenvalue problem can be computed with a generalized Schur decomposition.

**Proposition 2.1.23** ([129, Theorem 3.1], [70, Theorem 2.6.1], [51, Theorem 7.7.1])**.** *For any two matrices $A, B \in \mathbb{C}^{n\times n}$, there exist unitary matrices $Q, Z \in \mathbb{C}^{n\times n}$ and upper triangular matrices $T_A, T_B \in \mathbb{C}^{n\times n}$, such that $A = QT_AZ^H$, and $B = QT_BZ^H$. This is known as a* QZ decomposition*, or a* generalized Schur decomposition*.*

**Proposition 2.1.24** ([70, Theorem 2.6.1], [51, Theorem 7.7.1])**.** *Let $A, B \in \mathbb{C}^{n\times n}$ and $(A, B)$ be a regular pencil. Moreover, let $A = QT_AZ^H$ and $B = QT_BZ^H$ be a generalized Schur decomposition, where $Q, Z \in \mathbb{C}^{n\times n}$ are unitary matrices and $T_A, T_B \in \mathbb{C}^{n\times n}$ upper triangular matrices. The eigenvalues of the pencil is given by the quotients of the diagonal elements, i.e., $\sigma(A, B) = \{(\alpha, \beta) : \alpha = [T_A]_{i,i}, \beta = [T_B]_{i,i}, i = 1, 2, \ldots, n\}$.*

Recently, further generalizations with applications to periodic eigenvalue problems and systems of Sylvester equations are investigated in, e.g., [36].

### The Kronecker product

We close this section by introducing the Kronecker product and the vectorization operation. These operators and the properties presented will be especially useful when we consider linear matrix equations in Section 2.3.

**Definition 2.1.25** (Kronecker product)**.** Let $A \in \mathbb{R}^{n_A\times m_A}$ and $B \in \mathbb{R}^{n_B\times m_B}$, then the *Kronecker product* $A \otimes B \in \mathbb{R}^{n_An_B\times m_Am_B}$ is defined by the following block matrix:

$$A \otimes B := \begin{bmatrix} a_{1,1}B & \ldots & a_{1,m_A}B \\ \vdots & \ddots & \vdots \\ a_{n_A,1}B & \ldots & a_{n_A,m_A}B \end{bmatrix}.$$

**Definition 2.1.26** (Vectorization)**.** Let $A \in \mathbb{R}^{n\times m}$ and let $a_i$ be the $i$th column of $A$. The *vectorization* $\mathrm{vec}(A)$ is the vector obtained by stacking the columns of $A$ on top of each other. More specifically, $\mathrm{vec}(A) := \begin{bmatrix} a_1^T & a_2^T & \ldots & a_m^T \end{bmatrix}^T \in \mathbb{R}^{nm}$.

**Proposition 2.1.27** ([69, Chapter 4], [14, Chapter 12])**.** *Let $A, C \in \mathbb{R}^{n\times n}$ and $B, D \in \mathbb{R}^{m\times m}$. Furthermore, let $X \in \mathbb{R}^{n\times m}$ and $\alpha \in \mathbb{R}$. It holds that:*

1. $(\alpha A) \otimes B = A \otimes (\alpha B)$
2. $(A \otimes B)^T = A^T \otimes B^T$
3. $(A + C) \otimes B = (A \otimes B) + (C \otimes B)$ *and* $A \otimes (B + D) = (A \otimes B) + (A \otimes D)$
4. $(A \otimes B)(C \otimes D) = (AC) \otimes (BD)$
5. $\lambda \in \sigma(A \otimes B)$ *if and only if* $\lambda = \lambda_A \lambda_B$ *for some* $\lambda_A \in \sigma(A)$ *and* $\lambda_B \in \sigma(B)$
6. $\lambda \in \sigma(I \otimes A + B \otimes I)$ *if and only if* $\lambda = \lambda_A + \lambda_B$ *for some* $\lambda_A \in \sigma(A)$ *and* $\lambda_B \in \sigma(B)$
7. $A \otimes B$ *is nonsingular if and only if* $A$ *and* $B$ *are nonsingular. Moreover,* $(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$
8. $\operatorname{vec}(AXB) = (B^T \otimes A) \operatorname{vec}(X)$

**Corollary 2.1.28.** *If* $a, b \in \mathbb{R}^n$*, then*

$$\operatorname{vec}(ab^T) = b \otimes a.$$

*Proof.* From point 8 in Proposition 2.1.27, or by direct calculation since we have that

$$\operatorname{vec}(ab^T) = \begin{bmatrix} a_1 b_1 & a_2 b_1 & \dots & a_n b_1 & a_1 b_2 & a_2 b_2 & \dots & a_n b_n \end{bmatrix}^T = b \otimes a.$$

□

**Remark 2.1.29** (The name of $\otimes$)**.** *The operation* $\otimes$ *is also known as* direct product *or* tensor product*, see, e.g., [83]. However, some properties was allegedly first studied in 1858 by Zehfuss [142], and later worked out further by Hurwitz [72] in 1894.*[2] *Thus, it is argued that the operation* $\otimes$ *should be called the* Zehfuss product*; see [60] for an original reference, and also [51, Notes and references for Section 12.3] as well as [69, Notes and further readings for Section 4.2] for more recent mentioning (albeit with reference to [60]).*

## 2.2 Matrix functions

The term *matrix function* is used to denote a generalization of a (scalar) function $f : \mathbb{C} \to \mathbb{C}$, to a (matrix) function $f : \mathbb{C}^{n \times n} \to \mathbb{C}^{n \times n}$. Naturally, the two functions named $f$ are different objects, yet it is customary to give them the same name. Typically one uses $f(z)$ to denote the scalar version, sometimes denoted the *stem function* [69], and $f(A)$ to denote the matrix version. The notation has its roots in the common usage of lower case $z$ as a complex variable, and upper case $A$ as a matrix. The following summary on the topic of matrix functions is based on [63, 69, 73], among others. The topic is also treated in, e.g., [48, 14]. One way to generalize the concept would be to apply the

[2] The original articles are written in a language I do not master. Hence, I am unable to fully guarantee the validity. Nevertheless, the formulas and the parts I do manage to translate seems to verify the alleged result.

function element-wise. However, that is not what is considered here. Instead we start with (positive) integer powers, $f(z) = z^n$ for some $n \in \mathbb{N}$. The natural generalization is then simply $f(A) = A^n$. From there we can go to polynomials $p(z) = \sum_{i=0}^{m} a_i z^i$ which are generalized as $p(A) = \sum_{i=0}^{m} a_i A^i$, where we interpret $A^0 = I$ in analogy with $z^0 = 1$, i.e., power zero gives the multiplicative identity. With this in mind we arrive at the following definition.

**Definition 2.2.1** (Matrix function - power series)**.** Let $f : \mathbb{C} \to \mathbb{C}$ be an analytic function with locally convergent power series $f(z) = \sum_{i=0}^{\infty} a_i z^i$. Moreover, let $A \in \mathbb{C}^{n\times n}$ be a matrix with spectral radius $\rho(A) < r$, where $r$ is the radius of convergence of the power series. Then we define

$$f(A) := \sum_{i=0}^{\infty} a_i A^i.$$

For Definition 2.2.1 to make sense it is required that the power series in the matrix converges. The convergence follows from that the spectral radius is smaller than the radius of convergence of the power series, for a complete proof see, e.g., [69, Theorem 6.2.8]. A specific case of the definition is a Taylor series expansion, around $z = \mu$, which can be written as

$$f(A) := \sum_{i=0}^{\infty} \frac{f^{(i)}(\mu)}{i!}(A - \mu I)^i.$$

The definition is usable and intuitive for entire functions, and a good starting point for further investigation.

A second definition comes from Cauchy's integral formula, i.e., for a function $f$ that is analytic on and inside a simple, piecewise-smooth, and closed contour $\Gamma$, it holds that

$$f(x) = \frac{1}{2\pi i} \oint_{\Gamma} \frac{f(z)}{(z - x)} dz, \tag{2.2}$$

see, e.g., [1, Section 4.2.2]. Moreover, in the same spirit as we generalized (positive) integer powers we find that the generalization of $(z - x)^{-1}$ to the matrix argument $(zI - A)^{-1}$ is rather natural. In words it can be viewed as generalizing the operation of a scalar times the multiplicative identity, and the operation of multiplicative inverse. In operator theory the corresponding integral is called the Dunford–Taylor integral [76].

**Definition 2.2.2** (Matrix function - Dunford–Taylor integral)**.** Let $f : \mathbb{C} \to \mathbb{C}$ be analytic on and inside a simple, piecewise-smooth, and closed contour $\Gamma$. Moreover, let $A \in \mathbb{C}^{n\times n}$ be a matrix with eigenvalues strictly enclosed by $\Gamma$. Then we define

$$f(A) := \frac{1}{2\pi i} \oint_{\Gamma} f(z)(zI - A)^{-1} dz.$$

With two definitions available we need to guarantee that these are not contradictory. However, we postpone the presentation of such result to later and instead introduce a

third definition of a matrix function. The definition is based on the Jordan form (Proposition 2.1.15), and is a more general definition than the two above. To get an intuitive understanding of the definition we highlight two properties that hold for the two definitions above, and hence must hold for the new definition to be consistent (we formalize the results below). First, the function of a matrix and the function of a similarity transform of that matrix are related through the same similarity transform, i.e., $f(SAS^{-1}) = Sf(A)S^{-1}$. Second, for block-diagonal matrices the matrix function becomes the application of the (matrix) function to each diagonal block separately. Hence, the definition will be based on similarity transform to Jordan form and block-diagonal application of the function to each Jordan block. We need to define what $f(J(\lambda))$ means, where $J(\lambda)$ is a Jordan block. To get intuition we consider the $2 \times 2$ example

$$A_\varepsilon := \begin{bmatrix} \lambda+\varepsilon & 1 \\ 0 & \lambda \end{bmatrix} = \begin{bmatrix} 1 & \frac{-1}{\varepsilon} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \lambda+\varepsilon & 0 \\ 0 & \lambda \end{bmatrix} \begin{bmatrix} 1 & \frac{1}{\varepsilon} \\ 0 & 1 \end{bmatrix} =: S_\varepsilon \begin{bmatrix} \lambda+\varepsilon & 0 \\ 0 & \lambda \end{bmatrix} S_\varepsilon^{-1}$$

where $\lambda \in \mathbb{C}$. We have $\sigma(A_\varepsilon) = \{\lambda, \lambda+\varepsilon\}$. Hence, for $\varepsilon = 0$ we have that $A_0$ is a Jordan block, but if $\varepsilon \neq 0$ then $A_\varepsilon$ is diagonalizable, as seen in the right-hand side of the equation. We let $f$ be a suitable function (enough differentiable and continuous around $\lambda$), and use the properties listed above to get

$$f(A_\varepsilon) = S_\varepsilon \begin{bmatrix} f(\lambda+\varepsilon) & 0 \\ 0 & f(\lambda) \end{bmatrix} S_\varepsilon^{-1} = \begin{bmatrix} f(\lambda+\varepsilon) & \frac{f(\lambda+\varepsilon)-f(\lambda)}{\varepsilon} \\ 0 & f(\lambda) \end{bmatrix} \to \begin{bmatrix} f(\lambda) & f'(\lambda) \\ 0 & f(\lambda) \end{bmatrix},$$

as $\varepsilon \to 0$. By continuity of $f$ we require that $f(A_0) = \lim_{\varepsilon \to 0} f(A_\varepsilon)$. We see that for $f$ of the Jordan block $A_0$ the main diagonal is constant $f(\lambda)$ and that derivatives of $f$ in $\lambda$ are showing up in the upper diagonals. The general definition is as follows.

**Definition 2.2.3** (Matrix function - Jordan block)**.** Let $J_k(\lambda) \in \mathbb{C}^{k \times k}$ be a Jordan block, as in Definition 2.1.14. Moreover, let $f : \mathbb{C} \to \mathbb{C}$ be a function such that $\lambda$ is in the interior of the domain of $f$, and $f$ is $k-1$ times differentiable at $\lambda$. Then

$$f(J_k(\lambda)) := \begin{bmatrix} f(\lambda) & f'(\lambda) & \frac{1}{2}f''(\lambda) & \dots & \frac{1}{(k-1)!}f^{(k-1)}(\lambda) \\ 0 & f(\lambda) & f'(\lambda) & \dots & \frac{1}{(k-2)!}f^{(k-2)}(\lambda) \\ 0 & 0 & f(\lambda) & \dots & \frac{1}{(k-3)!}f^{(k-3)}(\lambda) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & f(\lambda) \end{bmatrix}.$$

We note that in the case of $k = 1$ then the Jordan block is trivial, i.e., $J_1(\lambda) = \big[\lambda\big]$, and hence $f(J_1(\lambda)) = \big[f(\lambda)\big]$. Thus, the requirement that $\lambda$ needs to be in the interior of the domain can be relaxed for semisimple eigenvalues, see [69, Definition 6.2.4]. Based on this definition, the general definition of a matrix function is as follows.

**Definition 2.2.4** (Matrix function - Jordan form)**.** Let $A \in \mathbb{C}^{n \times n}$ be a matrix with Jordan form $A = SJS^{-1}$ as in Proposition 2.1.15. Moreover, let $f : \mathbb{C} \to \mathbb{C}$ be a function such

that it is applicable to each Jordan block of $A$, in accordance with Definition 2.2.3. Then

$$f(A) := S \begin{bmatrix} f(J_{n_1}(\lambda_1)) & 0 & \dots & 0 \\ 0 & f(J_{n_2}(\lambda_2)) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & f(J_{n_m}(\lambda_m)) \end{bmatrix} S^{-1}.$$

The above definition is independent of the actual Jordan form used [69, Theorem 6.2.9], and is the most general of the definitions. Definition 2.2.4 only requires the existence of a finite number of derivatives, and only locally at the eigenvalues of $A$, whereas both Definitions 2.2.1 and 2.2.2 requires the function to be analytic in some regions. Hence, it is possible to only consider Definition 2.2.4, and derive the other results as theorems, as in [69, Section 6.2]; see also [63, Section 1.2]. However, from the point of view of intuition, we find it easier to start with Definition 2.2.1 since polynomials are quite familiar. We can now assert the previously promised equivalences, when applicable.

**Proposition 2.2.5** ([69, Theorems 6.2.8, 6.2.9, and 6.2.28], [63, Theorem 1.12], [76])**.** *Let $f : \mathbb{C} \to \mathbb{C}$ and $A \in \mathbb{C}^{n\times n}$ be given, and let $f_A := f(A)$ as given by Definition 2.2.4.*

- *If $f$ and $A$ are such that Definition 2.2.1 is applicable, then $f(A)$ as given by Definition 2.2.1 is equal to $f_A$.*
- *If $f$ and $A$ are such that Definition 2.2.2 is applicable, then $f(A)$ as given by Definition 2.2.2 is equal to $f_A$.*

**Remark 2.2.6.** *We have three definitions of a matrix function. However, given the equivalence from Proposition 2.2.5, there is no ambiguity in simply considering a matrix function $f(A)$. When in doubt, the more general Definition 2.2.4 can be considered.*

**Remark 2.2.7.** *We note that there are further ways of defining matrix functions than presented here, e.g., in [63, Section 1.2.2] a definition based on Hermite interpolation is presented. The Hermite interpolation is shown to be equivalent to the Jordan form definition. We omit the details but emphasize that a (primary) matrix function can, in the general case, also be understood as a polynomial in the matrix. See also [63, Problem 1.3]*

With the more general definition in place, we can also formalize what we said above regarding similarity transforms of matrix functions.

**Proposition 2.2.8** ([48, Section V.1.2], [63, Theorem 1.13], [65, Theorem 2.3])**.** *Let $A \in \mathbb{C}^{n\times n}$ be given, and let $f : \mathbb{C} \to \mathbb{C}$ be such that $f(A)$ is well-defined. Moreover, let $V \in \mathbb{C}^{n\times n}$ be invertible. Then $f(VAV^{-1}) = Vf(A)V^{-1}$.*

*Proof.* It follows directly from Definition 2.2.4 by noticing that if $A = SJS^{-1}$ is a Jordan form of $A$, then $(VS)J(VS)^{-1}$ is a Jordan form of $VAV^{-1}$. □

The same observation can also be used to prove the following result regarding the eigenvalues of a matrix function.

**Proposition 2.2.9** ([63, Theorem 1.13], [65, Theorem 2.3], [76])**.** *Let $A \in \mathbb{C}^{n\times n}$ be given and let $f : \mathbb{C} \to \mathbb{C}$ be such that $f(A)$ is well-defined. Moreover, let $\lambda_i$ $i = 1, 2, \ldots, n$ be the eigenvalues of A, i.e., $\lambda_i \in \sigma(A)$. Then $f(\lambda_i)$ are the eigenvalues of $f(A)$, i.e., $f(\lambda_i) \in \sigma(f(A))$, for $i = 1, 2, \ldots, n$.*

We can also see that if the function does not map any two eigenvalues to the same value, i.e., $f(\lambda_i) \neq f(\lambda_j)$ for all $\lambda_i \neq \lambda_j$ and $\lambda_i, \lambda_j \in \sigma(A)$, then the algebraic multiplicities of the eigenvalues are also preserved. However, the geometric multiplicities may change.

Having the more general definition in place, we can also follow up on the above claim regarding the application of matrix functions to block-diagonal matrices.

**Proposition 2.2.10** ([63, Theorem 1.13], [65, Theorem 2.3])**.** *Let $A_i \in \mathbb{C}^{n\times n}$ for $i = 1, 2, \ldots, m$, and let $f : \mathbb{C} \to \mathbb{C}$ be such that $f(A_i)$ is well-defined for $i = 1, 2, \ldots, m$. Then*

$$f\left(\begin{bmatrix} A_1 & 0 & \ldots & 0 \\ 0 & A_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & A_m \end{bmatrix}\right) = \begin{bmatrix} f(A_1) & 0 & \ldots & 0 \\ 0 & f(A_2) & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & f(A_m) \end{bmatrix}.$$

*Proof.* The proof can be done by induction. We show the base case, when $m = 2$. Let $A_1 = S_1 J_1 S_1^{-1}$ and $A_2 = S_2 J_2 S_2^{-1}$ be Jordan forms, according to Proposition 2.1.15. Then the block-diagonal matrix has a Jordan form

$$\begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix} = \begin{bmatrix} S_1 & 0 \\ 0 & S_2 \end{bmatrix} \begin{bmatrix} J_1 & 0 \\ 0 & J_2 \end{bmatrix} \begin{bmatrix} S_1^{-1} & 0 \\ 0 & S_2^{-1} \end{bmatrix}.$$

The above expression is a Jordan form of the block matrix. Hence,

$$\begin{aligned} f\left(\begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix}\right) &= \begin{bmatrix} S_1 & 0 \\ 0 & S_2 \end{bmatrix} f\left(\begin{bmatrix} J_1 & 0 \\ 0 & J_2 \end{bmatrix}\right) \begin{bmatrix} S_1^{-1} & 0 \\ 0 & S_2^{-1} \end{bmatrix} \\ &= \begin{bmatrix} S_1 & 0 \\ 0 & S_2 \end{bmatrix} \begin{bmatrix} f(J_1) & 0 \\ 0 & f(J_2) \end{bmatrix} \begin{bmatrix} S_1^{-1} & 0 \\ 0 & S_2^{-1} \end{bmatrix} = \begin{bmatrix} f(A_1) & 0 \\ 0 & f(A_2) \end{bmatrix}. \end{aligned}$$

The induction step follows by assuming that $A_1$ is block-diagonal with $m - 1$ blocks. □

There is also a related result, that deals with block-triangular matrices rather than block-diagonal matrices.

**Proposition 2.2.11** ([97, Corollary, p. 7], [63, Theorem 1.13], [65, Theorem 2.3])**.** *Let $A$ be block-triangular with the blocks, $A_{i,j} \in \mathbb{C}^{n\times n}$ for $i, j = 1, 2, \ldots, m$. Moreover, let $f : \mathbb{C} \to \mathbb{C}$ be such that $f(A)$ is well-defined. Then $f(A)$ is block-triangular with the same structure as A, and $f(A_{i,i})$ for $i = 1, 2, \ldots, m$ as diagonal blocks.*

With regard to Proposition 2.2.11 we note that it might be tempting to believe that if $f(A_{i,i})$ are well-defined for $i = 1, 2, \ldots, m$, then it would imply that $f(A)$ would

be well-defined. An improper argument would be that if $A_{i,i} = Q_{i,i}T_{i,i}Q_{i,i}^H$ are Schur decompositions for $i = 1, 2, \ldots, m$, then $A = QTQ^H$ is a Schur decomposition of $A$, where $Q$ is block-diagonal with $Q_i$ as diagonal block $i$, for $i = 1, 2, \ldots, m$. However, note that the eigenvalue multiplicity of $A$, and specifically the Jordan structure, depends on the structure and relation between the block-matrices in the definition of $A$. A counterexample is the Jordan block $A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$, which is block-triangular matrix with the $1 \times 1$ blocks $A_{1,1} = A_{1,2} = A_{2,2} = 1$. Even if $f(1)$ is well-defined, it is not necessarily true that $f'(1)$ exists, e.g., for $f(z) = |1 - z|$. Nevertheless, with some further assumptions the situation can be remedied. One way is to assume that $A_{i,i}$ and $A_{j,j}$ does not share any eigenvalues if $i \neq j$. For a similar discussion regarding practical aspects, see [33]. A second way is to assume $f$ is analytic in a region enclosing the eigenvalues, and thus possesses an infinite number of derivatives in that region. In both cases the extra assumption assures that the relevant part of the eigenstructure of $A$ is determined by the blocks $A_{i,i}$ independently.

As a special case, the block-diagonal result (Proposition 2.2.10) is applicable when all the elements on the block-diagonal are the same, resulting in a formula for the Kronecker product, $f(I \otimes A) = I \otimes f(A)$. Moreover, it turns out that an analogous formula holds for $f(A \otimes I)$.

**Proposition 2.2.12** ([63, Theorem 1.13])**.** *Let $A \in \mathbb{C}^{n\times n}$ be given, and let $I \in \mathbb{R}^{m\times m}$ be the identity matrix. If $f : \mathbb{C} \to \mathbb{C}$ is such that $f(A)$ is well-defined, then $f(I \otimes A) = I \otimes f(A)$, and $f(A \otimes I) = f(A) \otimes I$.*

*Proof.* As mentioned, $f(I \otimes A) = I \otimes f(A)$ is a special case of Proposition 2.2.10. The relation $f(A \otimes I) = f(A) \otimes I$ follows from that $A \otimes I = P(I \otimes A)P^T$ for some permutation matrix $P$ (see, e.g., [86]). Hence, with an application of the above result and Proposition 2.2.8 we have $f(A\otimes I) = Pf(I\otimes A)P^T = P(I\otimes f(A))P^T = f(A)\otimes I$. □

From the power series definition of the matrix function it seems plausible that $f(A)$ and $A$ commutes, i.e., $\mathrm{com}(f(A), A) = f(A)A - Af(A) = 0$. It is indeed true in general, even if the power series definition is not valid. See Remark 2.2.7 for further intuition.

**Proposition 2.2.13** ([63, Theorem 1.13], [65, Theorem 2.3])**.** *Let $A \in \mathbb{C}^{n\times n}$ be given, and let $f : \mathbb{C} \to \mathbb{C}$ be such that $f(A)$ is well-defined. Then $A$ and $f(A)$ commutes. Moreover, if $X \in \mathbb{C}^{n\times n}$ commutes with $A$, then it commutes with $f(A)$.*

## Some matrix functions

### Constant function

The first matrix function we mention is the constant function, i.e., $f(z) := c$ for some fixed $c \in \mathbb{C}$. This function can be interpreted as $c1$, i.e., $c$ times the multiplicative identity. We observe that the generalization is $f(A) = cI$, i.e., the constant $c$ times the identity matrix, which is the multiplicative identity for matrices.

### Inverse

The second matrix function we mention is the inverse, i.e., $f(z) := z^{-1}$. We noted already above, just above Definition 2.2.2 (Dunford–Taylor integral), that $f(A) = A^{-1}$ is a natural generalization. Although the inverse is usually not viewed as a matrix function, there is already a well-established theory for the inverse of a linear operator, we find it interesting that there is a matrix-function viewpoint. An example is Proposition 2.2.9, which is a well-known result for eigenvalues, typically taught in introductory courses in linear algebra.

Related to the inverse we have the function $f(z) := (1+z)^{-1}$. In the domain given by $|z| < 1$ we have that $f(z) = g(z)$, where $g(z) := \sum_{i=0}^{\infty}(-z)^i$. However, outside of the domain $f(z)$ may still be well-defined, although $g(z)$ is not and $f(z) \neq g(z)$. Hence, thinking in terms of matrix functions of $f$ and $g$ is a nice viewpoint of the *Neumann series*, as treated in, e.g., Paper B.

### Exponential function

The matrix exponential, i.e., the generalization of $f(z) := e^z$, is an important function with applications in various fields. In the words of Moler and Van Loan:

> *"...availability of* `expm(A)` *in early versions of* MATLAB *quite possibly contributed to the system's technical and commercial success."* [93, p. 42]

The exponential function $f$ is an entire function, and thus all definitions are applicable. Many well-known properties of the exponential generalize nicely (we list some results below). However, it has been called *"the great matrix exponential tragedy"* [93, p. 40] that $e^{A+B}$ is in general not equal to $e^A e^B$. Nevertheless, there are some remedies.

**Proposition 2.2.14** ([63, p. 235])**.** *Let $A, B \in \mathbb{C}^{n\times n}$. If $A$ and $B$ commutes, i.e., $AB = BA$. Then $e^{A+B} = e^A e^B$.*

**Corollary 2.2.15.** *Let $A \in \mathbb{C}^{n\times n}$. It holds that $e^{A/2}e^{A/2} = e^A$.*

The property is a basis for the *scaling and squaring* method used to compute the matrix exponential; see, e.g., the works of Higham [64, 63, 65]. Commutation is a sufficient condition. However, there are special cases where commutation is also a necessary condition. One example is the following result.

**Proposition 2.2.16** ([63, Theorem 10.2])**.** *Let $A, B \in \mathbb{C}^{n\times n}$ and $t \in \mathbb{C}$. Then $e^{(A+B)t} = e^{tA}e^{tB}$ for all $t \in \mathbb{C}$ if and only if $A$ and $B$ commutes, i.e., $AB = BA$.*

**Proposition 2.2.17.** *Let $A \in \mathbb{C}^{n\times n}$. The matrix $e^A$ is nonsingular and the inverse is $e^{-A}$.*

*Proof.* Nonsingularity follows from the eigenvalues of $e^A$ (Proposition 2.2.9), since $e^z \neq 0$ for all $z \in \mathbb{C}$. The form of the inverse follows from Proposition 2.2.14, since $e^A e^{-A} = e^{A-A} = e^0 = I$. □

**Proposition 2.2.18** ([76, Example 4.8])**.** *Let $A \in \mathbb{C}^{n\times n}$. Moreover, let $t \in \mathbb{C}$ be a parameter. Then*

$$\frac{d}{dt}e^{tA} = Ae^{tA} = e^{tA}A.$$

**Proposition 2.2.19.** *Let $A \in \mathbb{C}^{n\times n}$ be stable. Moreover, let $t \in \mathbb{C}$ be a parameter. Then*

$$\lim_{t\to\infty} e^{tA} = 0.$$

*Proof.* From Proposition 2.2.9 we know that the eigenvalues of $e^{tA}$ are $e^{\lambda t}$ where $\lambda \in \sigma(A)$. Since $A$ is stable $\mathrm{Re}(\lambda) < 0$ and hence $\lim_{t\to\infty} e^{t\lambda} = 0$. □

The two propositions together illustrate what we said above about stable matrices and the connection to stability of the dynamical system $\dot{x}(t) = Ax(t)$ (page 7). However, even if the asymptotic convergence is towards the zero-matrix, it is well-known that $\|e^{tA}\|$ may increase at first, before it decays; see, e.g., [93, 132].

**Proposition 2.2.20.** *Let $A \in \mathbb{C}^{n\times n}$. Moreover, let $t \in \mathbb{C}$ be a parameter. Then*

$$\int_0^t e^{\tau A}d\tau = A^{-1}(e^{tA} - I) = (e^{tA} - I)A^{-1}.$$

*If in addition $A$ is stable, then the limit as $t \to \infty$ equals $-A^{-1}$.*

### Sign function

The matrix sign function is a generalization of the complex sign function

$$f(z) := \mathrm{sign}(z) = \begin{cases} 1 & \text{if } \mathrm{Re}(z) > 0 \\ -1 & \text{if } \mathrm{Re}(z) < 0. \end{cases}$$

The generalization was introduced by Roberts in [109] as a tool to solve matrix equations, as we will see below (page 27). We list some well-known properties for reference.[3]

**Proposition 2.2.21.** *Let $A \in \mathbb{C}^{n\times n}$. Moreover, let $A = SJS^{-1}$ be a Jordan form such that*

$$J = \begin{bmatrix} J_+ & 0 \\ 0 & J_- \end{bmatrix},$$

*where $J_+ \in \mathbb{C}^{p\times p}$ is anti-stable and $J_- \in \mathbb{C}^{q\times q}$ stable, and $p + q = n$. Then*

$$\mathrm{sign}(A) = S \begin{bmatrix} I_p & 0 \\ 0 & -I_q \end{bmatrix} S^{-1},$$

*where $I_p$ and $I_q$ are identity matrices of dimensions $p \times p$ and $q \times q$, respectively.*

**Corollary 2.2.22.** *Let $A \in \mathbb{C}^{n\times n}$ be stable, then* $\mathrm{sign}(A) = -I$.

**Corollary 2.2.23** ([109, p. 678])**.** *Let $A \in \mathbb{C}^{n\times n}$. It holds that* $\mathrm{sign}(A)^2 = I$.

[3] The complex sign function is sometimes defined, differently, as the point on the unit circle in $\mathbb{C}$ that is closes to the given point, i.e., $g(z) = e^{i\arg(z)}$. We do not use this definition for matrix functions.

## 2.3 Linear Matrix Equations

A general *linear matrix equation* is described by the equation

$$\sum_{i=1}^{m} A_i X B_i^T = C, \tag{2.3}$$

were $A_i \in \mathbb{R}^{n_A \times n_A}$, $B_i \in \mathbb{R}^{n_B \times n_B}$, $C \in \mathbb{R}^{n_A \times n_B}$ are given for $i = 1, 2 \ldots, m$, and $X \in \mathbb{R}^{n_A \times n_B}$ is the unknown.[4] Properties and solution methods to this type of problems have been studied for over a century, see, e.g., the work by Wedderburn [89]. Note that equation (2.3) is linear in the unknown $X$, and can be equivalently formulated as a linear system, by using the Kronecker product and the properties in Proposition 2.1.27. The equivalent linear system is called the *Kronecker form* and is given as

$$\left( \sum_{i=1}^{m} B_i \otimes A_i \right) \mathrm{vec}(X) = \mathrm{vec}(C). \tag{2.4}$$

The opposite implication also holds true. Any linear system (with a dimension that is not a prime number) can be written as a linear matrix equation. Therefore, the problem of solving a linear matrix equation is equivalent to solving a linear system. The claim is made precise in the following result.

**Proposition 2.3.1** (Parameterizing linear operators)**.** *Let $M \in \mathbb{R}^{N \times N}$ and let $c \in \mathbb{R}^N$. Assume that $N = n_A n_B$. The vector $x \in \mathbb{R}^{n_A n_B}$ is a solution to $Mx = c$ if and only if there exists an integer $m$, and $A_i \in \mathbb{R}^{n_A \times n_A}$, $B_i \in \mathbb{R}^{n_B \times n_B}$ for $i = 1, 2 \ldots, m$ such that $c = \mathrm{vec}(C)$ and $x = \mathrm{vec}(X)$, where $X$ is a solution to* (2.3)*.*

*Proof.* We know that (2.3) is equivalent to (2.4). Hence, it is enough to show that any matrix $M \in \mathbb{R}^{n_A n_B \times n_A n_B}$ can be written on the form (2.4) for some $A_i$ and $B_i$. Let $m_{k,\ell}$ be the element in row $k$ and column $\ell$ of $M$. By a direct calculation, albeit somewhat tedious, it can be seen that

$$M = \sum_{i_1=1}^{n_B} \sum_{i_2=1}^{n_B} \sum_{i_3=1}^{n_A} \sum_{i_4=1}^{n_A} m_{i_3+(i_1-1)n_A,\, i_4+(i_2-1)n_A} \left( e_{i_1} e_{i_2}^T \right) \otimes \left( e_{i_3} e_{i_4}^T \right).$$

Then by using a one-to-one mapping between the multi-index $(i_1, i_2, i_3, i_4)$ and the index $i = 1, 2, \ldots, n_A^2 n_B^2$, given by $1 \leftrightarrow (1,1,1,1)$, $2 \leftrightarrow (1,1,1,2)$, $\ldots$, $n_A \leftrightarrow (1,1,1,n_A)$, $n_A + 1 \leftrightarrow (1,1,2,1)$, $\ldots$, $n_A^2 n_B^2 \leftrightarrow (n_B, n_B, n_A, n_A)$, we define the matrices $B_i := m_{i_3+(i_1-1)n_A,\, i_4+(i_2-1)n_A} \left( e_{i_1} e_{i_2}^T \right)$ and $A_i := \left( e_{i_3} e_{i_4}^T \right)$. Thus, the equation $Mx = c$ can be written on the form (2.4). Since $M$ was arbitrary the conclusion follows. □

The linear-systems formulation (2.4) uniquely characterize the solvability and gives a means to compute the solution to (2.3). For a longer discussion see the review [83].

[4] We will mostly consider real problems, although a lot of the theory is valid also for complex matrices.

However, computing $X$ via the Kronecker form (with a direct method) has complexity about $\mathcal{O}(n^6)$, for $n_A = n_B = n$ . Because of Proposition 2.3.1 this is as good as can be expected for the general case. Nevertheless, in special cases it is possible to use the special structures of (2.3) to create vastly more efficient algorithms. We thus treat common and important special cases below.

### The Lyapunov and the Sylvester equation

The two most common special cases of (2.3) are the *Lyapunov equation*

$$AX + XA^T = CC^T, \tag{2.5}$$

and the *Sylvester equation*

$$AX + XB^T = C_1C_2^T, \tag{2.6}$$

where we typically (for simplicity) consider $A, B \in \mathbb{R}^{n\times n}$, and $C, C_1, C_2 \in \mathbb{R}^{n\times r}$. It can be observed that the Sylvester equation is a generalization of the Lyapunov equation in the sense that (2.5) is obtained from (2.6) with $B = A$ and $C_1 = C_2 = C$. Both equations appear frequently and in a large variety of applications; e.g., in

- the study of dynamical systems. More specifically, in investigation of stability of time-invariant linear systems [78, 96, 5, 53]; and more importantly when considering controllability/observability via *Gramians*, as well as associated model reduction techniques [17, 5, 53].
- Newton–Kleinman-type methods for computations of solutions to the algebraic Riccati equation [77, 19].
- Newton's method for computing the matrix square root [63, Section 6.3].
- blocked Schur–Parlett-type methods for computing matrix functions [33, 63]. More generally, for matrix function evaluation where the eigenvalue-separation idea is achieved with the block-triangularization; see, e.g., [97] for a historical reference, and [45] for a recent application.
- general block-triangularization. See Proposition 2.3.15 below (page 26).
- computation of solutions to discretized partial differential equations defined on rectangular domains. See, e.g., [24] and [87] for early accounts of discretizations similar to Paper A, and [51, Section 4.8.4] for a more recent discussion. It is also treated in, e.g., [128].
- some image processing problems [29, 26, 139].

We will occasionally come across what we call the *two-sided Sylvester equation*, a generalization of the Sylvester equation (2.6) defined as

$$A_1XB_1^T + A_2XB_2^T = C_1C_2^T, \tag{2.7}$$

where we typically (for simplicity) consider $A_1, A_2, B_1, B_2 \in \mathbb{R}^{n\times n}$, and $C_1, C_2 \in \mathbb{R}^{n\times r}$. The two-sided equation is a generalization of the Sylvester equation since the latter comes as a special case of the former with $B_1 = A_2 = I$. We use the name two-sided Sylvester equation for (2.7), and although the name is not standard in the literature it helps distinguishing equation (2.7) from the "generalized" equations introduced below; see Remark 2.3.34 (page 42).

The literature related to the Lyapunov and Sylvester equations is large. For completeness we summarize some well-known results below. We start with a specialization of (2.4).

**Proposition 2.3.2.** *Let $A, B \in \mathbb{R}^{n\times n}$, and $C_1, C_2 \in \mathbb{R}^{n\times r}$. A matrix $X \in \mathbb{R}^{n\times n}$ solves the Sylvester equation* (2.6) *if and only if the vector $x = \mathrm{vec}(X)$ solves*

$$((I \otimes A) + (B \otimes I))x = c, \tag{2.8}$$

*where $c = \mathrm{vec}(C_1 C_2^T)$. Equation* (2.8) *is known as the* Kronecker form.

The solvability of (2.8) is characterized by the eigenvalues of the system matrix in (2.8), i.e., $(I \otimes A) + (B \otimes I)$. We note that those eigenvalues are given by $\lambda_A + \lambda_B$, where $\lambda_A, \lambda_B$ are the respective eigenvalues of $A$ and $B$ (point 6 in Proposition 2.1.27). Hence, the following characterization of the solvability of the Sylvester equation follows.

**Proposition 2.3.3** ([69, Theorem 4.4.6])**.** *Let $A, B \in \mathbb{R}^{n\times n}$, and $C_1, C_2 \in \mathbb{R}^{n\times r}$. The Sylvester equation* (2.6) *has a unique solution if and only if the spectra of $A$ and $-B$ are disjoint, i.e., $\sigma(A) \cap \sigma(-B) = \emptyset$.*

There exists generalization of the above theorem for the case of the two-sided Sylvester equation. The condition on the spectrum of the coefficient matrices is generalized to the spectrum of corresponding matrix pencils.

**Proposition 2.3.4** ( [30, Theorem 1])**.** *Let $A_1, A_2, B_1, B_2 \in \mathbb{R}^{n\times n}$, $C_1, C_2 \in \mathbb{R}^{n\times r}$. The two-sided Sylvester equation* (2.7) *has a unique solution if and only if the pencils $(A_1, A_2)$ and $(-B_2, B_1)$ are regular, and their spectra disjoint, i.e., $\sigma(A_1, A_2) \cap \sigma(-B_2, B_1) = \emptyset$.*

Since the Lyapunov equation can be considered a special case of the Sylvester equation, the existence and uniqueness for (2.5) follows from Proposition 2.3.3.

**Corollary 2.3.5.** *Let $A \in \mathbb{R}^{n\times n}$, and $C \in \mathbb{R}^{n\times r}$. The Lyapunov equation* (2.5) *has a unique solution if and only if there are no eigenvalues $\lambda \in \sigma(A)$ such that $-\lambda \in \sigma(A)$.*

The Lyapunov equation exhibits more structure than the Sylvester equation. Something that is reflected in the properties of the solution.

**Proposition 2.3.6.** *Let $A \in \mathbb{R}^{n\times n}$, and $C \in \mathbb{R}^{n\times r}$. If the Lyapunov equation* (2.5) *has at least one solution, then it has a symmetric solution. Specifically, if the Lyapunov equation has a unique solution, then the solution is symmetric.*

*Proof.* Assume that $X$ solves the Lyapunov equation. Transposing the equation yields

$$AX^T + X^T A^T = CC^T.$$

Hence, $X^T$ solves the Lyapunov equation. If the solution is unique, then $X = X^T$, otherwise $X + X^T$ is, by linearity, a symmetric solution. □

In the theory of dynamical systems the class of stable matrices is important. By definition these matrices automatically fulfill the eigenvalue requirement in Proposition 2.3.3, which we summarize in the following result.

**Corollary 2.3.7.** *Let $A, B \in \mathbb{R}^{n\times n}$, $C_1, C_2 \in \mathbb{R}^{n\times r}$. If $A$ and $B$ are stable matrices, then the Sylvester equation* (2.6) *has a unique solution.*

However, when considering dynamical systems, the Lyapunov equation is more commonly occurring. For the Lyapunov equation with stable coefficients there is an even stronger result; see, e.g., [48, Section XV] or citeAntoulas:2005:Approximation.

**Proposition 2.3.8.** *Let $A \in \mathbb{R}^{n\times n}$ be stable, and let $C \in \mathbb{R}^{n\times r}$. The Lyapunov equation* (2.5) *has a unique solution which is symmetric and negative semidefinite. Moreover, if $CC^T$ is positive definite, then the unique solution is negative definite.*

To see that the positive definiteness of $CC^T$ is required to guarantee the negative definiteness of the solution we look at a classical counterexample. Let $A$ be symmetric and stable. Moreover, let $C$ be an eigenvector to $A$. Then $X$, the solution to the Lyapunov equation, is a scaled version of $CC^T$, and hence only rank 1. In the symmetric case this counterexample works when $CC^T$ does not have full rank, since due to linearity $X$ can be constructed from a sum of rank-1 examples. The assumption of positive definiteness of $CC^T$ can be loosened to the pair $(A, C)$ being a *controllable pair*, i.e., the matrix $\begin{bmatrix} C & AC & \dots & A^{n-1}C \end{bmatrix} \in \mathbb{R}^{x\times nr}$ has full row rank. To see why the controllability criterion effectively prohibits the previous counter example we note that the full row rank is equivalent to the controllability matrix having $n$ columns that are linearly independent. The latter is in turn equivalent to that the smallest invariant subspace of $A$ containing the subspace (of $\mathbb{R}^n$) spanned by the columns of $C$, is of dimension $n$. Note that if $CC^T$ is positive definite, then $C$ is full rank, and hence $(A, C)$ is a controllable pair.

The following few results characterize the solution in different ways, and as such they may form bases for different types of solution methods. The first result is according to some sources due to Heinz, see [125, 23], although other sources attributes it to Krein, see [83].

**Proposition 2.3.9** ([59, Satz 5][5])**.** *Let $A, B \in \mathbb{R}^{n\times n}$, $C_1, C_2 \in \mathbb{R}^{n\times r}$. If $A$ and $B$ are stable matrices, then the unique solution to the Sylvester equation* (2.6) *is given by*

$$X = -\int_0^\infty e^{At} C_1 C_2^T e^{B^T t} dt.$$

[5] See the footnote on Page 12.

*Proof.* First, the existence and uniqueness of the solution to (2.6) is asserted by Corollary 2.3.7. Second, the integral defining $X$ converges since $A$ and $B$ have eigenvalues with negative real parts. Last, to see that the defined $X$ is the solution to (2.6) we substitute the expression into the left-hand side of (2.6) and thus get

$$\begin{aligned} AX + XB^T &= -A\left(\int_0^\infty e^{At}C_1C_2^Te^{B^Tt}dt\right) - \left(\int_0^\infty e^{At}C_1C_2^Te^{B^Tt}dt\right)B^T \\ &= -\int_0^\infty Ae^{At}C_1C_2^Te^{B^Tt} + e^{At}C_1C_2^Te^{B^Tt}B^T dt \\ &= -\int_0^\infty \frac{d}{dt}\left(e^{At}C_1C_2^Te^{B^Tt}\right)dt = -\left[e^{At}C_1C_2^Te^{B^Tt}\right]_0^\infty = C_1C_2^T. \end{aligned}$$

□

The provided proof is a verification, as opposed to a construction. However, intuition can be gained from considering that

$$\frac{1}{a} = -\int_0^\infty e^{at}dt,$$

for a scalar $a \in \mathbb{C}$ such that $\mathrm{Re}(a) < 0$. Hence, the integral in Proposition 2.3.9 can be understood as an extension to the Sylvester operator; see also Proposition 2.2.20 (page 19). A more general, constructive, proof for tensors is found in [52, Section 2]. A related integral, that can also be used to prove the proposition above, was presented by Rosenblum in [111]. See also [23]. A recent treatment is due to Wimmer [140], who generalized the result to the two-sided Sylvester equation. See also [5, Remark 6.1.1].

**Proposition 2.3.10** ([140, Theorem 2.1])**.** *Let $A_1, A_2, B_1, B_2 \in \mathbb{R}^{n\times n}$, $C_1, C_2 \in \mathbb{R}^{n\times r}$. Moreover, let the pencils $(A_1, A_2)$ and $(-B_2, B_1)$ be regular, and their spectra disjoint, i.e., $\sigma(A_1, A_2) \cap \sigma(-B_2, B_1) = \emptyset$. Then there exists a simple and closed contour $\Gamma$ such that $\sigma(A_1, A_2)$ is in the interior and $\sigma(-B_2, B_1)$ in the exterior. Moreover, the unique solution to the two-sided Sylvester equation* (2.7) *is given by*

$$X = \frac{1}{2\pi i}\int_\Gamma (zA_1 - A_2)^{-1}C_1C_2^T(zB_1^T + B_2^T)^{-1}dz.$$

For the Sylvester equation we get the following corollary, see, e.g., [83, Theorem 6].

**Corollary 2.3.11.** *Let $A, B \in \mathbb{R}^{n\times n}$, $C_1, C_2 \in \mathbb{R}^{n\times r}$, and let $\sigma(A) \cap \sigma(-B) = \emptyset$. Then there exists a simple and closed contour $\Gamma$ such that $\sigma(A)$ is in the interior and $\sigma(-B)$ in the exterior. Moreover, the unique solution to the Sylvester equation* (2.6) *is given by*

$$X = \frac{1}{2\pi i}\int_\Gamma (zI - A)^{-1}C_1C_2^T(zI + B^T)^{-1}dz.$$

For the stable Lyapunov equation, the contour can be taken around the right-half plane, with only integration over the imaginary axis giving nonzero contribution. The result is described as the Fourier transform of the integral in Proposition 2.3.9 and is as follows.

**Corollary 2.3.12** ([5, Equations (4.51) and (6.10)])**.** *Let $A \in \mathbb{R}^{n\times n}$, $C \in \mathbb{R}^{n\times r}$. If $A$ is a stable matrix, then the unique solution to the Lyapunov equation* (2.5) *is given by*

$$X = -\frac{1}{2\pi}\int_{-\infty}^{\infty}(izI - A)^{-1}CC^T(-izI - A^T)^{-1}dz.$$

The solution to the Sylvester equation can also be described in terms of an ordinary differential equation (ODE).

**Proposition 2.3.13** ([14, Chapter 10], [34])**.** *Let $A, B \in \mathbb{R}^{n\times n}$, $C_1, C_2 \in \mathbb{R}^{n\times r}$, and let $A$ and $B$ be stable matrices. Consider the matrix differential equation*

$$\dot{Y}(t) = AY(t) + Y(t)B^T - C_1C_2^T,$$

*with initial value $Y(0) = Y_0$. The unique solution is $Y(t) = e^{At}(Y_0 - X)e^{B^Tt} + X$, where $X$ is the unique solution to the Sylvester equation* (2.6)*. Moreover, $X = \lim_{t\to\infty} Y(t)$.*

*Proof.* The solution to the differential equation can be obtained by vectorizing the system to a standard ordinary differential equation (similar to Proposition 2.3.2). The matrix-form follows from $e^{((I\otimes A)+(B\otimes I))t} = e^{I\otimes At}e^{Bt\otimes I} = (I \otimes e^{At})(e^{Bt} \otimes I) = e^{Bt} \otimes e^{At}$. Commutation of $(I \otimes A)$ and $(B \otimes I)$ follows from point 4 in Proposition 2.1.27. □

**Proposition 2.3.14** ([127, Equation (3)])**.** *Let $A, B \in \mathbb{R}^{n\times n}$, and $C := C_1C_2^T$ where $C_1, C_2 \in \mathbb{R}^{n\times r}$. If $A$ and $B$ are stable matrices, then the unique solution to the Sylvester equation* (2.6) *is given by*

$$X = \sum_{i=0}^{\infty}\tilde{A}^i\tilde{C}\tilde{B}^i,$$

*where $\tilde{A} = (qI - A)^{-1}(qI + A)$, $\tilde{B} = (qI - B^T)^{-1}(qI + B^T)$, and $\tilde{C} = -2q(qI - A)^{-1}C(qI - B^T)^{-1}$, with $q \in \mathbb{R}$ a scalar such that $q > 0$, $q \notin \sigma(-A)$, and $q \notin \sigma(-B)$.*[6]

*Proof.* The idea is to rewrite the Sylvester equation to a type of Stein equation, by noticing that $(qI - A)X(qI - B^T) - (qI + A)X(qI + B^T) = -2q(AX + XB^T) = -2qC$, which is equivalent to $X - \tilde{A}X\tilde{B} = \tilde{C}$. The latter has a convergent series solution since $\rho(\tilde{A}) < 1$ and $\rho(\tilde{B}) < 1$. □

Smith uses that $f(z) := (qI - z)^{-1}(qI + z)$ maps the left-half plane into the disk with radius less than one. The proof reveals a correspondence between a solution to the Lyapunov and the Stein equation. The latter was pointed out by Smith already in [126, Theorem 3], and presented by Power who points out that $f(z) = I + 2(z - I)^{-1}$ [105], although it was also known before [10]. Hence, a (continuous time) stable matrix, can be mapped to a discrete (time) stable matrix, i.e., a matrix with spectral radius less than one.

The following result is a classical characterization of existence and is called *Roth's solvability criterion*. It originates from [112]. For later accounts see, e.g., [69, Theorem 4.4.22], [5, Proposition 6.1], or [125, Equation (19)].

[6] For notational convenience the matrix $\tilde{B}$ is defined using $B^T$.

**Proposition 2.3.15** ([112]). *Let $A, B \in \mathbb{R}^{n\times n}$, and $C := C_1C_2^T$ where $C_1, C_2 \in \mathbb{R}^{n\times r}$. The Sylvester equation* (2.6) *has a solution if and only if*

$$\begin{bmatrix} A & -C \\ 0 & -B^T \end{bmatrix} \quad \textit{is similar to} \quad \begin{bmatrix} A & 0 \\ 0 & -B^T \end{bmatrix}.$$

*Furthermore, the similarity transformation is given by*

$$\begin{bmatrix} I & X \\ 0 & I \end{bmatrix} \quad \textit{which has the inverse} \quad \begin{bmatrix} I & -X \\ 0 & I \end{bmatrix},$$

*where $X$ is a solution to* (2.6)*.*

Roth's solvability criterion can be used to prove certain characterizations of the solution to the Sylvester equation, as in [7, 109]. There is also a related analysis, of a related block-matrix, connected with the algebraic Riccati equation; see [103, 4, 109].

**Proposition 2.3.16** ([7]). *Let $A, B \in \mathbb{R}^{n\times n}$, and $C := C_1C_2^T$ where $C_1, C_2 \in \mathbb{R}^{n\times r}$. Assume that the spectra of $A$ and $-B$ is disjoint, i.e., $\sigma(A) \cap \sigma(-B) = \emptyset$. Define*

$$G := \begin{bmatrix} A & -C \\ 0 & -B^T \end{bmatrix} \in \mathbb{R}^{2n\times 2n}.$$

*Moreover, let $E := \begin{bmatrix} N^T & M^T \end{bmatrix}^T \in \mathbb{C}^{2n\times n}$, be part of a (partial) Jordan decomposition of $G$ such that $GE = EJ_B$, where $J_B$ is a Jordan matrix corresponding to $-B^T$. It holds that $M$ is nonsingular, and unique solution to the Sylvester equation* (2.6) *is given by $X = NM^{-1}$.*

*Proof.* Existence and uniqueness follows from the spectra of $A$ and $B$ (Proposition 2.3.3), and the rest follows from Roth's solvability criterion (Proposition 2.3.15). We have that

$$EJ_B = GE = \begin{bmatrix} I & X \\ 0 & I \end{bmatrix} \begin{bmatrix} A & 0 \\ 0 & -B^T \end{bmatrix} \begin{bmatrix} I & -X \\ 0 & I \end{bmatrix} \begin{bmatrix} N \\ M \end{bmatrix}.$$

First, from the bottom row we find that $MJ_B = -B^TM$, and hence $M$ is the set generalized eigenvectors of $B$ constituting the similarity transform of the Jordan decomposition corresponding to $J_B$. Thus, $M$ is nonsingular. Second, from the top row we have

$$NJ_B = AN - AXM - XB^TM = AN - CM.$$

We multiply with $M^{-1}$ from the right and utilize $J_BM^{-1} = -M^{-1}B^T$ to get

$$C = A(NM^{-1}) + (NM^{-1})B^T.$$

Thus, $NM^{-1}$ is a solution to the Sylvester equation, and from uniqueness $X = NM^{-1}$. □

**Proposition 2.3.17** ([109]). *Let $A, B \in \mathbb{R}^{n\times n}$, and $C := C_1C_2^T$ where $C_1, C_2 \in \mathbb{R}^{n\times r}$. If $A$ and $B$ are stable matrices, then the unique solution to the Sylvester equation* (2.6) *is given by $X$ such that*

$$\begin{bmatrix} -I & 2X \\ 0 & I \end{bmatrix} = \operatorname{sign}\left(\begin{bmatrix} A & -C \\ 0 & -B^T \end{bmatrix}\right)$$

*Proof.* Existence and uniqueness follows from stability of $A$ and $B$ (Corollary 2.3.7). The rest of the proof follows from Roth's solvability criterion (Proposition 2.3.15) and the properties of the matrix sign function. Direct calculations give

$$\begin{aligned}\operatorname{sign}\left(\begin{bmatrix} A & -C \\ 0 & -B^T \end{bmatrix}\right) &= \begin{bmatrix} I & X \\ 0 & I \end{bmatrix} \operatorname{sign}\left(\begin{bmatrix} A & 0 \\ 0 & -B^T \end{bmatrix}\right) \begin{bmatrix} I & -X \\ 0 & I \end{bmatrix} \\ &= \begin{bmatrix} I & X \\ 0 & I \end{bmatrix} \begin{bmatrix} -I & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} I & -X \\ 0 & I \end{bmatrix} = \begin{bmatrix} -I & 2X \\ 0 & I \end{bmatrix}.\end{aligned}$$

□

Using Proposition 2.3.17, the sign function can be expressed using the Dunford–Taylor integral formula (Definition 2.2.2), thus giving an integral expression for the solution of the Sylvester equation in the following way.

**Proposition 2.3.18** ([109], [63, Equation (5.3)]). *Let $A, B \in \mathbb{R}^{n\times n}$, and $C := C_1C_2^T$ where $C_1, C_2 \in \mathbb{R}^{n\times r}$. If $A$ and $B$ are stable matrices, then the unique solution to the Sylvester equation* (2.6) *is given by $X$ such that*

$$\begin{bmatrix} -I & 2X \\ 0 & I \end{bmatrix} = \frac{2}{\pi} \begin{bmatrix} A & -C \\ 0 & -B^T \end{bmatrix} \int_0^\infty \left( t^2 I + \begin{bmatrix} A & -C \\ 0 & -B^T \end{bmatrix}^2 \right)^{-1} dt.$$

The result can be re-written by multiplication with the block vector $\begin{bmatrix} 0 & I/2 \end{bmatrix}^T$ from the right, and $\begin{bmatrix} I & 0 \end{bmatrix}$ from the left, and exploiting a Schur-complement technique on the inverse in the integrand. We arrive at the following related characterization of the solution.

**Corollary 2.3.19.** *Let $A, B \in \mathbb{R}^{n\times n}$, and $C := C_1C_2^T$ where $C_1, C_2 \in \mathbb{R}^{n\times r}$. If $A$ and $B$ are stable matrices, then the unique solution to the Sylvester equation* (2.6) *is given by*

$$\begin{aligned} X &= \frac{-1}{\pi} \int_0^\infty \left(A^2 + t^2 I\right)^{-1} \left(ACB^T + t^2 C\right) \left((B^T)^2 + t^2 I\right)^{-1} dt \\ &= \frac{-1}{\pi} \int_0^\infty \left(A + t^2 A^{-1}\right)^{-1} \left(C + t^2 A^{-1} C B^{-T}\right) \left(B^T + t^2 B^{-T}\right)^{-1} dt \\ &= \frac{-1}{\pi} \int_0^\infty \left(A(t) + A(t)^{-1}\right)^{-1} \left(\frac{1}{t^2} C + A^{-1} C B^{-T}\right) \left(B(t)^T + B(t)^{-T}\right)^{-1} dt, \end{aligned}$$

*where, on the last line, we have defined $A(t) := A/t$ and $B(t) := B/t$.*

The solution to the Sylvester and the Lyapunov equation can also be characterized as optimal solutions to certain optimization problems. The first one presented below is a standard residual minimization, stated here for completeness. The norm in the objective function can be essentially any matrix norm.

**Proposition 2.3.20.** *Let $A, B \in \mathbb{R}^{n\times n}$, $C_1, C_2 \in \mathbb{R}^{n\times r}$, and let $\sigma(A) \cap \sigma(-B) = \emptyset$. The unique solution to the Sylvester equation* (2.6) *is the optimal solution to*

$$\begin{aligned} \min_{Y} \quad & \|\mathcal{R}\|^2 \\ \text{s.t.} \quad & \mathcal{R} = AY + YB^T - C_1C_2^T. \end{aligned}$$

*Proof.* First, the existence and uniqueness of the solution, call it $X$, to (2.6) is asserted by Corollary 2.3.7. Second, it follows that $\|\mathcal{R}\|^2 > 0$ if $Y \neq X$ and $\|\mathcal{R}\|^2 = 0$ if $Y = X$. □

The following proposition is specialized for the Lyapunov equation.

**Proposition 2.3.21.** *Let $A \in \mathbb{R}^{n\times n}$ be stable, and let $C \in \mathbb{R}^{n\times r}$. The unique solution to the Lyapunov equation* (2.5) *is the optimal solution to*

$$\begin{aligned} \max_{Y} \quad & -\operatorname{Tr}(Y) \\ \text{s.t.} \quad & AY + YA^T - CC^T \preceq 0 \\ & Y = Y^T. \end{aligned}$$

*Proof.* The Lyapunov equation has a unique solution, call it $X$, that is symmetric and negative semidefinite (Proposition 2.3.8). The proof is based on the *residual equation*. Take any feasible $Y$ such that $\mathcal{R} := AY + YA^T - CC^T \prec 0$, and note that $\mathcal{R} = \mathcal{R}^T$. Consider $E := X - Y$ which, from linearity we know fulfills $AE + EA^T = -\mathcal{R}$. Furthermore, we know that $E$ is nonzero, symmetric, and negative semidefinite (Proposition 2.3.8). Thus, $0 < -\operatorname{Tr}(E) = -\operatorname{Tr}(X) - (-\operatorname{Tr}(Y))$, implying that $-\operatorname{Tr}(Y) < -\operatorname{Tr}(X)$, from which the result follows. □

We close the section with some remarks on the naming convention in the literature, on the history of the names Lyapunov and Sylvester, and on (controllability) Gramians.

**Remark 2.3.22** (Conventions in the literature)**.** *We call* (2.5) *the Lyapunov equation, as is done in some parts of the literature [5, Equation (6.2)]. The name can also, see [69], refer to the equation*

$$XA + A^TX = CC^T.$$

*However, the latter formulation is equivalent in the sense that it is just a matter of how the matrix $A$ is defined. Similarly, and due to its connection to systems theory, in parts of the literature the Lyapunov equation is written as*

$$AX + XA^T + CC^T = 0,$$

*see [5, Equation (4.45)], and also Paper C. The latter is simply a way to prescribe that the right-hand side is negative semidefinite.*

*Analogously we call* (2.6) *the Sylvester equation, as is also done in parts of the literature, e.g., [80]. In other parts of the literature the name refers to equations of the form*

$$AX + XB = C_1C_2^T, \qquad or \qquad AX - XB = C_1C_2^T,$$

*see [51]. Similar to the Lyapunov case, it is just a question of how the matrix $B$ is defined.*

*The formulations* (2.5) *and* (2.6) *are natural from our perspective since the equations are written as linear-operator-acting-on-unknown equals right-hand-side, and convenient when working with projection methods; typically Krylov subspaces. Moreover, the generalization from Lyapunov to Sylvester is immediate.*

*It shall also be noted that in some parts of the literature, e.g., [49, 140], the equation*

$$AXA^T - X + CC^T = 0$$

*is called the (discrete time) Lyapunov equation, because it has an analogous connection to discrete time dynamical systems [5]. The latter equation is also known as the* Stein equation*, see [125, Section 6]. For a relation with the Lyapunov equation see [83, Section 5], [5, Section 4.3.3], or Proposition 2.3.14.*

**Remark 2.3.23** (Lyapunov)**.** *The name Lyapunov is attached to equation* (2.5) *in honor of his contributions to stability theory of dynamical systems [125]. The Lyapunov equation plays a key role in the stability analysis of linear dynamical systems; see, e.g., [48, Sections XIV and XV]. For example, the solution can be used to construct Lyapunov functions; see, e.g., [126, 35], [96, Section 4.3], and [78, remark following Theorem 21.1]. To the best of our knowledge, it seems as if the name Lyapunov was attached to equation* (2.5) *some time during the mid to late 1960s. In some earlier treatements, e.g., the 1932 paper by Rutherford [117] and the 1952 paper by Roth [112], the name Lyapunov seems not to be associated with* (2.5)*. Neither the book [48] from 1959 (Russian 1953), nor the book [78] from 1963, nor the paper [88] by Ma from 1966, nor the paper [127] by Smith from 1968, seems to directly connect the name Lyapunov directly with the matrix equation. However, in a series of notes from 1966–1967 Barnett and Storey begin to call equation* (2.5) *for the Lyapunov equation [9, 8, 10, 12]. Also Power references* (2.5) *as the Lyapunov equation in 1967 [105, 106]. The transcription "Liapunov" is used for* (2.5) *in 1967 by Barnett and Storey [11], and in 1968 by Davison and Man [35]. These may or may not be the first instances where* (2.5) *is called the Lyapunov equation, but nevertheless, in the light of the above cited literature, they give indications of the spread of the name. In the 1970 review paper [83] by Lancaster the name Lyapunov is attached to the matrix equation, and seems to have established the convention of calling* (2.5) *the* Lyapunov equation*.*

**Remark 2.3.24** (Sylvester)**.** *Equation* (2.6) *is called the Sylvester equation since the 1884 work by Sylvester [131][7] is regarded as the first work on the problem; see [125]. The equation is sometimes also named the Sylvester–Rosenblum equation due to the early results by Rosenblum on the operator case, presented in [111]; see also [23].*

[7] See the footnote on Page 12.

**Remark 2.3.25** (Gramian)**.** *Let $A \in \mathbb{R}^{n\times n}$ be stable, and let $C \in \mathbb{R}^{n\times r}$. Consider the linear time-invariant control system $\dot{x}(t) = Ax(t) + Cu(t)$, $x(0) = 0$, where $x(t) \in \mathbb{R}^n$ is the state and $u(t) \in \mathbb{R}^r$ is the control input. The* impulse response *is resulting from applying the input $u_i(t) = \delta(t)$ for $i = 1, 2, \ldots, r$, where $\delta(t)$ is the* Dirac delta function*. If we denote the impulse response with $h(t)$, then $h(t) = e^{At}C$. The* Gramian *of the functions given by the components, $h_1(t), h_2(t), \ldots, h_n(t)$, is defined as*

$$G := \begin{bmatrix} \langle h_1(t), h_1(t)\rangle & \langle h_1(t), h_2(t)\rangle & \ldots & \langle h_1(t), h_n(t)\rangle \\ \langle h_2(t), h_1(t)\rangle & \langle h_2(t), h_2(t)\rangle & \ldots & \langle h_2(t), h_n(t)\rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle h_n(t), h_1(t)\rangle & \langle h_n(t), h_2(t)\rangle & \ldots & \langle h_n(t), h_n(t)\rangle \end{bmatrix} = \int_0^\infty e^{At}C\left(e^{At}C\right)^T dt.$$

*Hence, from Proposition 2.3.9 we have that $G$, the controllability Gramian, is the unique solution to the Lyapunov equation*

$$AG + GA^T + CC^T = 0.$$

*Moreover, the Gramian is nonsingular if and only if the vectors are linearly independent [48, Chapter IX][70, Theorem 7.2.10][14, Chapter 4]. Hence, the controllability Gramian $G$ is nonsingular (and thus full rank) if and only if the components of the impulse response are linearly independent functions. Less formally that means if and only if the system can be steered to an arbitrary point in $\mathbb{R}^n$ by the use of a control consisting of a sum of Dirac delta pulses. Compare with the discussion above on the definiteness of the solution to the Lyapunov equation and relation to controllable pair (page 23).*

*A similar discussion is found in [5, Chapter 4]. The associated Lyapunov equation is formulated in a slightly different way compared to* (2.5)*, see Remark 2.3.22 above.*

## Computational methods for the Lyapunov and the Sylvester equation

There exists a plethora of algorithms for the Lyapunov and the Sylvester equation, all with different aspects, strengths, and weaknesses. Some are designed for computing the exact solution, while others are aimed at computing approximations thereof. We give an overview of a few here, with special focus on iterative methods, and more precisely projection methods, as these are further treated in Papers B and C.

### Kronecker-form methods

As mentioned in the introduction to the section, one class of methods for solving a linear matrix equation can be derived by directly utilize the Kronecker form. For the Sylvester equation the system needed to be solved is given by Proposition 2.3.2, i.e.,

$$((I \otimes A) + (B \otimes I))x = c.$$

The most naive way to solve the linear system is to use a direct solver. However, the system is typically large since $(I\otimes A)+(B\otimes I) \in \mathbb{R}^{n^2\times n^2}$. Thus, a direct method quickly become prohibitively expensive.

Another way to utilize the Kronecker form is to use a iterative solver to compute a solution (or approximation thereof). The large matrix $(I \otimes A) + (B \otimes I)$ does not need to be formed, since that action of the matrix on a vector $v \in \mathbb{R}^{n^2}$ can be implemented using the relation $((I \otimes A) + (B \otimes I))v = \mathrm{vec}(AV + VB^T)$, where $V \in \mathbb{R}^{n \times n}$ such that $v = \mathrm{vec}(V)$. It is also possible incorporate preconditioners. This is similar to the approach used in Paper A. See [66] for an early account on preconditioners. These subspace methods come close to the projection methods treated below; see [71] for tensorized Krylov methods for the Sylvester equation, and [82] for higher order tensors.

It is also possible to exploit the Kronecker form by applying a classical method for linear systems, which can be set in the matrix equation context by reversing the Kronecker form. Such a derivation is exploited in, e.g., [128] where a block version of the successive over-relaxation (SOR) is adapted to the Sylvester equation.

### Decomposition methods

As the name suggests this class of methods is based on different decompositions of the coefficient matrices. The most famous method is probably the *Bartels–Stewart algorithm* [13]. To explain the idea we start by considering the simplest case, which we call the *diagonalization method*. Consider the Sylvester equation (2.6) under the further assumption that $A$ and $B$ are diagonalizable, i.e., $A = V\Lambda_A V^{-1}$, and $B = W\Lambda_B W^{-1}$. We can rewrite the equation in the following equivalent ways:

$$\begin{aligned} AX + XB^T &= C_1 C_2^T \\ V\Lambda_A V^{-1} X + XW^{-T}\Lambda_B W^T &= C_1 C_2^T \\ \Lambda_A V^{-1} X W^{-T} + V^{-1} X W^{-T} \Lambda_B &= V^{-1} C_1 C_2^T W^{-T} \\ \Lambda_A Y + Y \Lambda_B &= \tilde{C}, \end{aligned}$$

where $X = VYW^T$ and $\tilde{C} = V^{-1}CW^{-T}$. The solution $Y$ to the new matrix equation is given by

$$Y_{k,\ell} = \frac{\tilde{C}_{k,\ell}}{[\Lambda_A]_{k,k} + [\Lambda_B]_{\ell,\ell}}.$$

See, e.g., [88, Equation (11)] or [125, Section 4]. Hence, the computation of $Y$ is simple, and from $Y$ we can get $X$ directly. Still, the method as such is of limited use since $A$ and $B$ needs to be diagonalizable, and diagonalization can be numerically unstable. However, in special cases, where the diagonalization is stable and efficient, the method can be useful. See, e.g., Paper A.

A generalized approach to solving (2.6) was presented by Rutherford in [117] and refined by Ma in [88]. In this approach the matrices $A$ and $B$ are not diagonalized, instead their Jordan form are computed (Proposition 2.1.15) and small scale Sylvester equations are solved in a structured way. Similarly, in [106] Power presents solution methods based on the Schwarz and Routh canonical forms. However, these methods are of limited practical application.

The idea of the *Bartels–Stewart algorithm* [13] is similar to the methods above, but the method utilizes the Schur decomposition (Proposition 2.1.16). Let $A = VT_AV^T$ and $B = WT_BW^T$ be Schur decompositions, where $V$ and $W$ are orthogonal matrices. A similar derivation as above leads to the transformed matrix equation

$$T_AY + YT_B^T = \tilde{C},$$

where $X = VYW^T$ and $\tilde{C} = V^TCW$. The transformed equation can then be solved using a backward substitution technique.[8] More precisely, Let $y_i$ and $\tilde{c}_i$ be the $i$th columns of $Y$ and $\tilde{C}$, respectively, then the columns of $Y$ can be computed by solving the triangular systems of equations,

$$\left(T_A - \left[T_B^T\right]_{i,i} I\right) y_i = \tilde{c}_i - \sum_{k=1}^{i-1} \left[T_B^T\right]_{k,i} y_k.$$

Note that there are specializations using the real Schur decomposition that can be used if the constituent matrices are real. In such case, the backward substitution is somewhat more technical. Although algorithms for computing Schur decompositions are inherently iterative in nature [133, Lectures 24-25], it is generally argued that the Bartels–Stewart algorithm is a direct method and the complexity, for $n_A = n_B = n$, is $\mathcal{O}(n^3)$. Some advantages of the Bartels–Stewart algorithm is that there exists Schur decompositions for all matrices, and that the transformations involve orthogonal matrices and hence have good numerical properties.

**Remark 2.3.26** (The two-sided equation)**.** *A similar technique can be applied to the two-sided Sylvester equation* (2.7)*, i.e., $A_1XB_1^T + A_2XB_2^T = C_1C_2^T$. The solution method utilizes the QZ decomposition (Proposition 2.1.23). The decomposition does not constitute a similarity transform, but can be used to simultaneously reduce two matrices to triangular form using a pair of orthogonal matrices. The matrix pairs $A_1$, $A_2$ and $B_1$, $B_2$ are jointly reduced, allowing for a transformation of the equation similar to above. Compare the existence criterion in Proposition 2.3.4 and the relation between eigenvalues of pencils and the QZ decomposition in Proposition 2.1.24. See [30, 49, 100] for a further discussion.*

### Smith method

What has come to be called the *Smith method* was presented by Smith in [127]. In the paper, Smith also presents Proposition 2.3.14, and the method is based on truncating the infinite sum and compute it iteratively in a cleaver way. More precisely, the iteration is

$$Y_{k+1} = Y_k + \tilde{A}^{2^k} Y_k \tilde{B}^{2^k}, \quad k = 0, 1, \ldots,$$

with $Y_0 = \tilde{C}$, where $\tilde{A}$, $\tilde{B}$, and $\tilde{C}$ are given in Proposition 2.3.14. It can be shown that $Y_k = \sum_{i=0}^{2^k-1} \tilde{A}^i \tilde{C} \tilde{B}^i$, and by noticing that $\tilde{A}^{2^k} = \tilde{A}^{2^{k-1}} \tilde{A}^{2^{k-1}}$, the method can be implemented using five matrix-matrix multiplications per step.

[8] Note that the equivalent linear system for the transformed matrix equation is (block) triangular.

A similar method was, at the same time, proposed for the Lyapunov equation [35]. Davison and Man base their derivation on Crank–Nicolson integration of the corresponding Lyapunov function. In their case $\tilde{A} := (I - Ah/2 + A^2h^2/12)^{-1}(I + Ah/2 + A^2h^2/12)$, and $\tilde{B}$ analogously but with $A^T$, and $\tilde{C} = hCC^T$. The step-size $h$ is suggested to be set to $h = 1/(200\rho(A))$.

## ADI iteration

The *ADI iteration* (Alternating Direction Implicit), sometimes known as the *Peaceman–Rachford iteration* was presented in 1955 by Peaceman and Rachford [98, 119]. For the Sylvester equation it can be seen as a generalization of the Smith method; see [44, 138] and [125, Section 4.4.2], and the method was presented in 1986 by Ellner and Wachspress [44]. Observe that the identity used in the proof of Proposition 2.3.14 can be generalized to

$$(qI - A)X(pI - B^T) - (pI + A)X(qI + B^T) = -(q + p)C_1C_2^T,$$

and hence a similar result as Proposition 2.3.14 holds true, with direct analogues to $\tilde{A}$ and $\tilde{B}$. Thus, the Smith iteration can be generalized. Specifically, Ellner and Wachspress notice that $q$ and $p$ should be optimized to minimize the spectral radius of $\tilde{A}$ and $\tilde{B}$, which leads to the ADI min-max problem. Wachspress [138] presents the method in the classical ADI-form

$$\begin{aligned}(A + p_iI)X_{i-1/2} &= C_1C_2^T - X_{i-1}(B^T - p_iI)\\ X_i(B^T + q_iI) &= C_1C_2^T - (A - q_iI)X_{i-1/2},\end{aligned}$$

where the shifts $p_i$ and $q_i$ can be different in each iteration. Finding appropriate shifts is of great importance for the method to be efficient; for shift-search strategies see, e.g., [18] for recent development. There are also versions of the method that computes a low-rank factorization directly, developed in works such as [101, 84, 20]. The latter works also which partially covers shift search. An overview can be found in [125].

### Sign-function iteration

The *sign-function iteration*, sometimes called *sign-function algorithm*, was originally presented by Roberts in [109]. The algorithm is based on Proposition 2.3.17, together with an iterative method for computing the matrix sign function. One way to achieve the latter is to consider the equation $f(X) = X^2 - I$, and noticing that $\operatorname{sign}(X)$ is a root of $f(X) = 0$. The Newton iteration turns out to be

$$Z_{k+1} = \frac{Z_k + Z_k^{-1}}{2}, \quad k = 0, 1, \ldots,$$

with $Z_0 = A$. It holds that $\lim_{k\to\infty} Z_k = \operatorname{sign}(A)$; the convergence is global and quadratic. See, e.g., [109], [63, Chapter 5], or [51, Section 9.4.1] for further details and enhancements. The application of this for solving the Sylvester equation is straight forward given the result in Proposition 2.3.17.

**Remark 2.3.27** (Riccati equation)**.** *An analogous technique can also be applied to solve the algebraic Riccati equation. The result is also presented in [109].*

### Optimization methods

By the nature of Proposition 2.3.20, constructing an optimization-based method for the Sylvester equation from the proposition would result in a method similar to a tensorized solver for the Kronecker form, see above. For the stable Lyapunov equation it is possible to consider optimization methods based on the formulation in Proposition 2.3.21, or versions thereof. The constraint is called a *linear matrix inequality* (LMI) and the optimization problem could be approached with *semidefinite programming* (SDP), see, e.g., [54, Section 16.8] However, there are more advanced optimization based approaches to compute approximations of the Lyapunov equation. In the paper [136] a Riemannian optimization method is proposed for computing low-rank solutions to large-scale Lyapunov equations with symmetric positive definite coefficients. The optimization minimizes the objective function $\mathrm{Tr}(XAX) - \mathrm{Tr}(XC)$, over the smooth manifold of symmetric positive semidefinite matrices of rank $k$. The objective function is based on the non-constant part of a related energy norm of the actual error, similar to what is considered in [80] as well as in Paper C. If the residual is not small enough when a minimum for the current optimization is found, then the process is continued but on the manifold of rank $k + 1$ matrices.

### General projection methods

The *projection methods* are part of a class of methods called *subspace methods*, and the idea is to search for an approximation in a restricted subspace. As previously noted, a linear matrix equation can be cast into a linear system, and hence projection methods for matrix equations is tightly connected to projection methods for linear systems; a further introduction to projection methods for linear systems can be found in, e.g., [119, Chapter 5].

There are two important factors for a projection method to be efficient: First, the problem needs to be well approximated in some restricted subspace; second, there must be an efficient way of computing such a subspace. For matrix equations, the first condition means that the solution matrix needs to be well approximated by a low-rank matrix, i.e., the solution needs to exhibit a strong decay in singular values. In the literature this is frequently expressed as the problem admitting *low-rank solutions*, although the exact solution may be of full rank. What is meant by efficient in the second criterion will naturally depend on the circumstances. However, comparing to direct methods for the Lyapunov and the Sylvester equation which have complexity $\mathcal{O}(n^3)$ gives an upper bound, and usually complexities of the order $\mathcal{O}(n^2)$, or even $\mathcal{O}(n)$ for sparse matrices, are desirable. In practice, the subspaces are often iteratively constructed, and nested, such that the approximation can be improved until desired accuracy is reached, although this is not always the case; see, e.g., applications of *Iterative Rational Krylov Algorithm* (IRKA) [56, 46] for solving the Lyapunov and the Sylvester equation.

We consider a projection method for matrix equations to consist of constructing nested subspaces $\mathcal{K}_{k-1} \subset \mathcal{K}_k \subset \mathbb{R}^n$, and $\mathcal{H}_{k-1} \subset \mathcal{H}_k \subset \mathbb{R}^n$, called the left and the right sub-

space, respectively. Furthermore, we let $\mathcal{V}_k$ and $\mathcal{W}_k$ be bases of $\mathcal{K}_k$ and $\mathcal{H}_k$, respectively, and for numerical reasons it is good to let these be orthonormal, i.e., $\mathcal{V}_k^T \mathcal{V}_k = I$ and $\mathcal{W}_k^T \mathcal{W}_k = I$. We search for an approximation $X_k$ such that columns of $X_k$ are in $\mathcal{K}_k$ and the rows are in $\mathcal{H}_k$, i.e., $X_k = \mathcal{V}_k Y_k \mathcal{W}_k^T$, where the matrix $Y_k$ needs to be determined (and is much smaller than the matrix $X_k$). However, from the formulation so far the matrix $Y_k$ is not unique. A common way to determine $Y_k$ is to impose that the residual is orthogonal to the subspaces, where orthogonality is defined through the *Frobenius inner product*, i.e, $\langle A, B\rangle = \mathrm{Tr}(B^T A)$ for $A, B \in \mathbb{R}^{n\times n}$. Let $\mathcal{K}_k$ and $\mathcal{H}_k$ be of dimension $\kappa$; a generic element in our space is given by $\mathcal{V}_k Z \mathcal{W}_k^T$. Hence, for all $Z \in \mathbb{R}^{\kappa\times\kappa}$, we require that

$$0 = \langle \mathcal{R}_k, (\mathcal{V}_k Z \mathcal{W}_k^T)\rangle = \mathrm{Tr}(\mathcal{W}_k Z^T \mathcal{V}_k^T \mathcal{R}_k) = \mathrm{Tr}(Z^T \mathcal{V}_k^T \mathcal{R}_k \mathcal{W}_k) = \langle \mathcal{V}_k^T \mathcal{R}_k \mathcal{W}_k, Z\rangle,$$

where $\mathcal{R}_k := AX_k + X_k B^T - C_1 C_2^T$ is the *residual*, and where $X_k = \mathcal{V}_k Y_k \mathcal{W}_k^T$ as before. Since $Z$ is arbitrary, we can conclude that the *Galerkin condition*[9] is

$$\mathcal{V}_k^T \mathcal{R}_k \mathcal{W}_k = 0.$$

The condition can be further simplified and the resulting equation from which $Y_k$ can be determined is commonly known as the *projected problem*. For the Sylvester equation (2.6), given the left and right subspaces from above, the projected problem is

$$A_k Y_k + Y_k B_k^T = C_{1,k} C_{2,k}^T,$$

where $A_k = \mathcal{V}_k^T A \mathcal{V}_k$, $B_k = \mathcal{W}_k^T B \mathcal{W}_k$, $C_{1,k} = \mathcal{V}_k^T C_1$, and $C_{2,k} = \mathcal{W}_k^T C_2$. The computation of $Y_k$ can typically be done using methods for small to medium scale Sylvester equations. Another way to make $Y_k$ unique is to enforce a *Petrov–Galerkin condition*, i.e., generate a separate trial and test space. More precisely, the process generates two extra spaces, $\hat{\mathcal{K}}_k$ and $\hat{\mathcal{H}}_k$ with corresponding orthogonal bases $\hat{\mathcal{V}}_k$ and $\hat{\mathcal{W}}_k$. The ansatz is still $X_k = \mathcal{V}_k Y_k \mathcal{W}_k^T$, but the Petrov–Galerkin condition is $\hat{\mathcal{V}}_k^T \mathcal{R}_k \hat{\mathcal{W}}_k = 0$.

We exemplify with a schematic algorithm utilizing a Galerkin projection, and target the Lyapunov equation (2.5) for simplicity, e.g., only one space has to be generated. The procedure is described in Algorithm 2.1. For the algorithm to be efficient, the two criteria described above needs to carefully balanced (good approximation properties of the subspace, and efficient ways to construct the subspace). As an illustration consider extending the subspace in Step 2 of Algorithm 2.1 with a random direction in $\mathbb{R}^n$. In most cases such construction is useless. It has a too high emphasis on the second criterion and does not include (enough) information about the problem. Thus, the space does not, in the generic case, have the desired approximation properties. On the contrary, an optimal projection space of dimension $k$ for approximating the solution to the Lyapunov equation would be the space spanned by the $k$ most dominant singular vectors of the actual solution of the equation (in the sense of minimizing the error measured in the Frobenius norm). However, this idea is in the other extreme. The projection space has desired approximation

[9] We once again highlight the connection to projection methods for linear systems by pointing out that an equivalent way of formulating the condition is $(\mathcal{W}_k^T \otimes \mathcal{V}_k^T)\,\mathrm{vec}(\mathcal{R}_k) = 0$; see, e.g., [125, Section 4.4.1], which can be compared to [119, Chapter 5].

properties, but the construction of the projection space in Step 2 is prohibitively expensive. Hence, this construction is also useless for most problems.

**Algorithm 2.1:** A generic projection algorithm for the Lyapunov equatio (2.5)

**input :** $A$, $C$, `tol`
**output:** $X$

$\mathcal{V}_0 = \emptyset$
**for** $k = 1, 2, \ldots$ *until convergence* **do**
  $v_k \leftarrow$ choose one or a few vectors in $\mathbb{R}^n$
  $\hat{v}_k \leftarrow$ orthogonalize $v_k$ w.r.t. $v_k$ and $\mathcal{V}_{k-1}$, and normalize
  $\mathcal{V}_k = [\mathcal{V}_{k-1}, \hat{v}_k]$
  Solve the projected problem $A_k Y_k + Y_k A_p^T = C_k C_k^T$,
    where $A_k = \mathcal{V}_k^T \, A \, \mathcal{V}_k$, and $C_k = \mathcal{V}_k^T \, C$
  $X_k = \mathcal{V} \, Y_k \, \mathcal{V}_k^T$
  $\mathcal{R}_k = AX_k + X_k A^T - CC^T$
  **if** $\|\mathcal{R}_k\| <$ `tol` **then**
    Break
**return** $X = X_k$

Projection methods is a class of tools used in many different approximation algorithms. One prime example is to evaluate matrix functions multiplied with vectors, such as, e.g., $e^{At}C_1$. It is hence close at hand to think of methods based on evaluating the integrand in Proposition 2.3.9, using a projection method. However, in [118] Saad showed that approximating the integral of Proposition 2.3.9 using subspace techniques to approximate the functions $e^{At}C_1$ and $e^{Bt}C_2$ is equivalent to a direct Galerkin projection of the Sylvester equation (2.6).

**Proposition 2.3.28** ([118, Theorem 4.2])**.** *Let $A, B \in \mathbb{R}^{n\times n}$ and $C_1, C_2 \in \mathbb{R}^{n\times r}$, and let $A$ and $B$ be stable matrices. Moreover, let $\mathcal{K}_k$ and $\mathcal{H}_k$ be two $\kappa$-dimensional subspaces of $\mathbb{R}^n$, and let $\mathcal{V}_k, \mathcal{W}_k \in \mathbb{R}^{n\times\kappa}$ be orthonormal bases of respective subspace.*

*Assume that the matrices $A_k := \mathcal{V}_k^T \, A \, \mathcal{V}_k$ and $B_k := \mathcal{W}_k^T \, B \, \mathcal{W}_k$ are stable. Construct an approximation to the solution of the Sylvester equation* (2.6) *by evaluating*

$$X_k := -\int_0^\infty \mathcal{V}_k \, e^{A_k t} \, \mathcal{V}_k^T \, C_1 C_2^T \, \mathcal{W}_k \, e^{B_k^T t} \, \mathcal{W}_k^T \, dt.$$

*Moreover, construct a second approximation, we call it $\hat{X}_k$, by direct Galerkin projection, i.e., $\hat{X}_k := \mathcal{V}_k \, \hat{Y}_k \, \mathcal{W}_k^T$, where $\hat{Y}_k \in \mathbb{R}^{\kappa\times\kappa}$ is the solution to the projected problem*

$$A_k \hat{Y}_k + \hat{Y}_k B_k^T = C_{1,k} C_{2,k}^T,$$

*where $A_k, B_k$ are given above and $C_{1,k} := \mathcal{V}_k^T \, C_1$, and $C_{2,k} := \mathcal{W}_k^T \, C_2$.*

*Then $X_k = \hat{X}_k$.*

*Proof.* We have

$$X_k = \mathcal{V}_k \left( - \int_0^\infty e^{A_k t} (\mathcal{V}_k^T C_1)(\mathcal{W}_k^T C_2)^T e^{B_k^T t} dt \right) \mathcal{W}_k^T = \mathcal{V}_k \hat{Y}_k \mathcal{W}_k^T = \hat{X}_k.$$

The second equality follows since the integral is the solution to the projected problem. □

The idea of the proof can be described as involving a trivariate matrix function $T$ of the matrix functions $f(A)$, $g(B^T)$, and $h(C_1 C_2^T)$, i.e., the solution can be written as $X = T(f(A), g(B^T), h(C_1 C_2^T))$. Additionally, there is a special structure required in the projection-approximation of $T$. More precisely, a projection is evaluated as $\mathcal{V}_k \hat{X}_k \mathcal{W}_k^T = \mathcal{V}_k T(f(A_k), g(B_k^T), h(C_{1,k} C_{2,k}^T)) \mathcal{W}_k^T$. Hence, the proof is easily adapted to the other integral characterizations mentioned above. Moreover, an analogous result can be established for the sign-function characterization in Proposition 2.3.17.

**Proposition 2.3.29.** *Let $A, B \in \mathbb{R}^{n \times n}$ and $C_1, C_2 \in \mathbb{R}^{n \times r}$, and let $A$ and $B$ be stable matrices. Furthermore, let $\mathcal{K}_k$ and $\mathcal{H}_k$ be two $\kappa$-dimensional subspaces of $\mathbb{R}^n$, and let $\mathcal{V}_k, \mathcal{W}_k \in \mathbb{R}^{n \times \kappa}$ be orthonormal bases of respective subspace.*

*Assume that the matrices $A_k := \mathcal{V}_k^T A \mathcal{V}_k$ and $B_k := \mathcal{W}_k^T B \mathcal{W}_k$ are stable. Construct an approximation to the solution of the Sylvester equation* (2.6) *as $X_k := \mathcal{V}_k Y_k \mathcal{W}_k^T$, where $Y_k$ is given by*

$$\begin{bmatrix} -I & 2Y_k \\ 0 & I \end{bmatrix} = \operatorname{sign} \left( \begin{bmatrix} A_k & -C_{1,k} C_{2,k}^T \\ 0 & -B_k^T \end{bmatrix} \right)$$

*Moreover, construct a second approximation, $\hat{X}_k$, by direct Galerkin projection, i.e., define $\hat{X}_k := \mathcal{V}_k \hat{Y}_k \mathcal{W}_k^T$, where $\hat{Y}_k \in \mathbb{R}^{\kappa \times \kappa}$ is the solution to the projected problem*

$$A_k \hat{Y}_k + \hat{Y}_k B_k^T = C_{1,k} C_{2,k}^T,$$

*where $A_k, B_k$ are given above and $C_{1,k} := \mathcal{V}_k^T C_1$, and $C_{2,k} := \mathcal{W}_k^T C_2$.*

*Then $X_k = \hat{X}_k$.*

*Proof.* The sign-function approximation can be written as

$$\begin{aligned} X_k &= \frac{1}{2} \begin{bmatrix} I_n & 0_n \end{bmatrix} \begin{bmatrix} \mathcal{V}_k & 0 \\ 0 & \mathcal{W}_k \end{bmatrix} \operatorname{sign} \left( \begin{bmatrix} A_k & -C_{1,k} C_{2,k}^T \\ 0 & -B_k^T \end{bmatrix} \right) \begin{bmatrix} \mathcal{V}_k & 0 \\ 0 & \mathcal{W}_k \end{bmatrix}^T \begin{bmatrix} 0_n \\ I_n \end{bmatrix} \\ &= \mathcal{V}_k \frac{1}{2} \begin{bmatrix} I_\kappa & 0_\kappa \end{bmatrix} \operatorname{sign} \left( \begin{bmatrix} A_k & -C_{1,k} C_{2,k}^T \\ 0 & -B_k^T \end{bmatrix} \right) \begin{bmatrix} 0_\kappa \\ I_\kappa \end{bmatrix} \mathcal{W}_k^T, \end{aligned}$$

which is exactly on the aforementioned form. □

A shorter proof would be to note that $X_k = \hat{X}_k$ since $Y_k = \hat{Y}_k$, and the latter follows since from construction $Y_k$ also solves the projected problem (which is unique). However, the former highlights the aforementioned structure and that the matrix sign-function is approximated using the specially structured basis

$$\begin{bmatrix} \mathcal{V}_k & 0 \\ 0 & \mathcal{W}_k \end{bmatrix} \in \mathbb{R}^{2n \times 2\kappa}.$$

**Remark 2.3.30.** *In Propositions 2.3.28 and 2.3.29 we assume that the projected matrices $A_k$ and $B_k$ were stable. If $A$ and $B$ are symmetric, then stability follows directly from negative definiteness (Proposition 2.1.10), since also the projected matrices are symmetric and negative definite. However, the example following Proposition 2.1.10 (page 7) illustrates why this is not the case for a non-symmetric matrix.*

### Krylov subspace methods

One particular class of spaces commonly used in connection to the Lyapunov and the Sylvester equation are Krylov-type subspaces. These subspaces are widely used in many fields and problems. With the notation used in the previous section for left and right subspaces, we have the following different types of Krylov subspaces: The *Krylov subspaces*

$$\begin{aligned}\mathcal{K}_k &:= \operatorname{span}\left\{C_1, AC_1, \ldots, A^k C_1\right\} \\ \mathcal{H}_k &:= \operatorname{span}\left\{C_2, BC_2, \ldots, B^k C_2\right\},\end{aligned}$$

the *extended Krylov subspaces*

$$\begin{aligned}\mathcal{K}_k &:= \operatorname{span}\left\{C_1, A^{-1}C_1, AC_1, A^{-2}C_1 \ldots, A^k C_1, A^{-k-1}C_1\right\} \\ \mathcal{H}_k &:= \operatorname{span}\left\{C_2, B^{-1}C_2, BC_2, B^{-2}C_2 \ldots, B^k C_2, B^{-k-1}C_2\right\},\end{aligned}$$

and the *rational Krylov subspaces*

$$\begin{aligned}\mathcal{K}_k &:= \operatorname{span}\left\{C_1, (s_1^a I - A)^{-1}C_1, \ldots, \prod_{i=1}^{k}(s_i^a I - A)^{-1}C_1\right\} \\ \mathcal{H}_k &:= \operatorname{span}\left\{C_2, (s_1^b I - B)^{-1}C_2, \ldots, \prod_{i=1}^{k}(s_i^b I - B)^{-1}C_2\right\},\end{aligned}$$

where the two sequences of complex shifts, i.e., $\{s_i^a\}_{i=1}^k$ and $\{s_i^b\}_{i=1}^k$, are such that the corresponding shifted matrices are nonsingular.

The application of Krylov subspaces to solve the Lyapunov and the Sylvester equation were presented in 1989 by Saad [118]. In terms of the Kronecker form (Proposition 2.3.2), a Krylov subspace method could seem like a natural approach to the problem. However, it is not a Krylov subspace based on the system matrix $(I \otimes A) + (B \otimes I)$ that is of interest here. Rather it is a tensorized version $\mathcal{H}_k \otimes \mathcal{K}_k$, which is what is presented and motivated in [118]. For more on tensorized Krylov subspaces, see [82].

The extended Krylov subspaces for the Lyapunov and the Sylvester equation were introduced in [124, 28], where they form the basis of the method called Krylov-plus-inverted Krylov (K-PIK). The method uses a modified Gram–Schmidt method for the orthogonalization and the projected matrices, i.e., $A_k = \mathcal{V}_k^T A \mathcal{V}_k$ and $B_k = \mathcal{W}_k^T B \mathcal{W}_k$, are computed from the orthogonalization coefficients, avoiding the need for a direct projection. Further efficiency is gained from exploiting the structure of the residual, and thus allowing computations of the residual norm without explicitly forming the residual.

The Rational Krylov subspace was introduced in 1984 by Ruhe, and used for eigenvalue computation [115]. It has been used in model reduction and for computations of matrix functions; see, e.g., [5, Chapter 11] and the additional references mentioned in [125, p. 400]. Although there are close connections with the previous applications, the first application of rational Krylov subspaces for solving matrix equations seems to be in paper [39] from 2011 (see [125, p. 409]). As mentioned for the ADI iteration, it is of great importance to find good shifts. This challenge has been treated recently by Druskin and co-authors in [37, 38, 39], resulting in an (almost) parameter-free method applicable to the Lyapunov and the Sylvester equation. The case when the right-hand side is not of low rank (i.e., $r$ is not small in the notation around (2.6)) is treated in [40] where tangential directions are used. Rational Krylov subspaces are many times written and even implemented in a more compact form, as presented in Proposition 2.3.32 below. In order to prove the proposition we need a lemma, known as the *resolvent equation*, or sometimes the *resolvent identity*.

**Lemma 2.3.31** ([83, Equation (15)])**.** *Let $A \in \mathbb{R}^{n\times n}$, and let $\mu, \lambda \in \mathbb{C}$ be such that $\mu I - A$ and $\lambda I - A$ are nonsingular. Then*

$$(\mu - \lambda)(\lambda I - A)^{-1}(\mu I - A)^{-1} = (\lambda I - A)^{-1} - (\mu I - A)^{-1}.$$

*Proof.* Consider the identity $(\mu - \lambda)I = (\mu I - A) - (\lambda I - A)$, and multiply it with $(\lambda I - A)^{-1}$ from the left and $(\mu I - A)^{-1}$ from the right. □

The resolvent equation is a general result and holds true for general (infinite dimensional) linear operators; see, e.g., [76, p. 36]. By inductively applying the resolvent equation to the definition of the rational Krylov subspace, we reach the following conclusion.

**Proposition 2.3.32.** *Let $A \in \mathbb{R}^{n\times n}$, $C \in \mathbb{C}^{n\times r}$, and let $\{s_i\}_{i=1}^m$ be a set of nonzero scalars, $s_i \in \mathbb{C}$, such that $s_i \neq s_j$ if $i \neq j$ and $s_i I - A$ is nonsingular for $i = 1, 2, \ldots, m$. Then*

$$\operatorname{span}\left\{C, \ldots, \prod_{i=1}^{m}(s_i I - A)^{-1}C\right\} = \operatorname{span}\left\{C, (s_1 I - A)^{-1}C, \ldots, (s_m I - A)^{-1}C\right\}.$$

We emphasize the requirement that the shifts are different (i.e., $s_i \neq s_j$) for the equality to hold (the resolvent equation reads $0 = 0$ if $\mu = \lambda$). Hence, for cases of cyclic reuse of shifts the standard definition has to be used.

For many problems the input quantities are real, and a complex-valued approximation would be nonsensical. Still, when applying the rational Krylov method with a nonsymmetric matrix it may be desirable to use complex shifts, since the spectrum of the matrix may contain complex (conjugate) eigenvalues. If the matrices and vectors involved are real, then it is possible to avoid using a complex valued basis for the rational Krylov subspace, as long as both the shift and its complex conjugate are used in constructing the basis.

**Proposition 2.3.33** ([116]). *Let $A \in \mathbb{R}^{n\times n}$, $C \in \mathbb{R}^{n\times r}$, and let $s \in \mathbb{C}$ be such that $sI - A$ and $\bar{s}I - A$ are nonsingular, where $\bar{s}$ is the conjugate of $s$. Then*

$$\operatorname{span}\left\{(sI-A)^{-1}C,\ (\bar{s}I-A)^{-1}C\right\} = \operatorname{span}\left\{\operatorname{Re}((sI-A)^{-1}C),\ \operatorname{Im}((sI-A)^{-1}C)\right\}.$$

*Proof.* The proposition follows directly by the observations

$$\operatorname{Re}((sI-A)^{-1}) = \frac{1}{2}\left((sI-A)^{-1} + (\bar{s}I-A)^{-1}\right)$$
$$\operatorname{Im}((sI-A)^{-1}) = \frac{1}{2i}\left((sI-A)^{-1} - (\bar{s}I-A)^{-1}\right).$$

□

**Quadrature and time-stepping methods**

There is a great number of integral formulations of the solution of the Lyapunov and the Sylvester equation, e.g., Proposition 2.3.9, Corollary 2.3.11, Corollary 2.3.12, Proposition 2.3.18, and Corollary 2.3.19 mentioned above. In theory quadrature can be applied to all of them.

In [118], Saad suggests approximation of the integral in Proposition 2.3.9 and show that a subspace approximation of the integrand is equivalent to a direct Galerkin approximation (Proposition 2.3.28). Hence, Saad motivates the use of Krylov subspace methods for the Lyapunov equation. However, the paper also contains some analysis of approximation of the matrix exponential and computation using quadrature. For example, it is noted that there is a relation with an ODE related to the one presented in Proposition 2.3.13. More precisely, we have that,

$$\dot{Y}(t) = AY(t) + Y(t)B^T,$$

with $Y(0) = C_1C_2^T$ has the solution $Y(t) = e^{At}C_1C_2^Te^{B^Tt}$, which is exactly the integrand described in Proposition 2.3.9.

A somewhat related approach is described in [55], where Gudmundsson and Laub note that for $A, B \in \mathbb{R}^{n\times n}$ stable, the system

$$\dot{x}(t) = B^Tx(t), \qquad x(0) = v, \qquad y(t) = C_2^Tx(t)$$

has the output $y(t) = C_2^Te^{tB^T}v$. Thus, the system

$$\dot{z}(t) = Az(t) + C_1y(-t), \qquad z(-\infty) = 0,$$

has the solution at $t = 0$ given by $z(0) = \int_{-\infty}^{0} e^{-tA}C_1C_2^Te^{-tB^T}v\,dt = Xv$. It is suggested to numerically integrate the ODEs with, e.g., a Runge-Kutta or linear multistep method. Based on the approximative matrix-vector product we can think of applying the method to a set of orthogonal and normalized initial input vectors $\{v_i\}$, and orthogonalize (and normalize) the set of output vectors $\{z_i(0)\}$. Thus, the procedure can be repeated with

the new input vectors being the previous (orthogonalized and normalized) output vectors. The presented procedure effectively becomes an approximative blocked power method for computing an invariant space corresponding to the dominant eigenvalues. An approximative power method is also described in [67], albeit based on a different approach. The difference is that the matrix-vector product is approximated by using a Krylov subspace and computing the solution to a projected Sylvester equation with one large coefficient and the other small, in each iteration.

A different type of exploitation of quadrature is found in the paper by Grasedyck [52]. The underlying integral is related to the integral in Proposition 2.3.9, but formulated for tensorized equations with a (potentially) high number of tensor modes. In the paper a sophisticated Stenger quadrature is used, and the result can also be used to motivate the existence of (relatively) low-rank solutions; specifically also applicable to the Sylvester equation, which is used in the low-rank characterization in Paper B (see Section B.2.2).

In [90] the integral in Proposition 2.3.9 is treated directly with the variable substitution $x = L\cot(\theta/2)^2$, where $L$ is a parameter and $\cot(\theta) = 1/\tan(\theta)$. The presented method exploits quasiseparability of the coefficients, a structure informally described as the off-diagonal blocks being of low rank.

### The generalized Lyapunov and Sylvester equation

Other special cases of the general linear matrix equation (2.3) that has attracted a lot of attention in recent years are the *generalized Lyapunov equation*

$$AX + XA^T + \sum_{i=1}^{m} N_i X N_i^T = CC^T, \tag{2.9}$$

and the *generalized Sylvester equation*

$$AX + XB^T + \sum_{i=1}^{m} N_i X M_i^T = C_1 C_2^T. \tag{2.10}$$

In relation to these equations we have found it natural to define the (linear) operators $\mathscr{L}(X) := AX + XB^T$ and $\Pi(X) := \sum_{i=1}^{m} N_i X M_i^T$. Similar to the standard equations, (2.9) and (2.10) show up in many different applications, e.g, as

- (generalized) *Gramians* for time-invariant bilinear dynamical systems. See [113, 31] for the background regarding control and observability, [3] for the formulation of the matrix equation, [143] for further treatment and connection with $\mathcal{H}_2$-model reduction, and [17] for a more modern treatment with focus on energy estimates and model reduction. Gramians are computed in the examples in Paper C.

- discretizations of partial differential equations (PDEs); see [24] for an early treatment. Further examples are, e.g., a convection–diffusion in the thesis [95], the problem treated in Paper A, and Example B.4.3 in Paper B. Recent research regards techniques for more complex domains [58].

A common assumption in the literature is that $\rho(\mathscr{L}^{-1}\,\Pi) < 1$, where $\rho$ denotes the spectral radius. The assumption is reasonable in the sense that (2.3) can, as pointed out above, parametrize any linear operator. Hence, in order to say something specific the operator $\Pi$ must have special structure and/or be bounded in relation to $\mathscr{L}$. If $\rho(\mathscr{L}^{-1}\,\Pi) < 1$, then writing the right-hand side operator in (2.10) as $\mathscr{L} + \Pi$ constitutes a *convergent splitting*. Methods relying on fixed-point iteration and Neumann series (equivalent in exact arithmetic [130]) have been developed using this splitting. The basic idea has strong similarities to Jacobi and Gauss–Seidel iterations, see, e.g., [51, Section 11.2] and [119, Section 4.1]. For methods exploiting fixed-point iteration and Neumann series see, e.g., [32, 16, 122, 90] as well as Papers B and C. However, we stress that there are many interesting applications where this condition on the spectral radius is not satisfied, e.g., indefinite solution matrices related to bilinear Gramians, as mentioned in [16]; the general linear matrix equations stemming from discretizations of PDEs in [104]; and the problem treated in Paper A.

Computational methods for the generalized Lyapunov and Sylvester equation is still an active research topic, and so far, and to the best of our knowledge, with no clear method of choice. Moreover, there is no sharp border between the generalized Lyapunov and Sylvester equation, and general linear matrix equations. Recent contributions include: exploiting the fixed-point iterations, as mentioned above ([32, 16, 122, 90]); preconditioned Krylov subspaces [32], and general methods for linear systems with tensor product structure [80, 81]; specific equations with low-rank corrections, .i.e., $\Pi$ having low-rank coefficients, [32, 16, 90]; a bilinear version of ADI (BilADI) [16]; greedy low-rank method based on the *alternating linear scheme* (ALS) [80]; rational Krylov-type methods [104]; and from the connection with bilinear control systems there is the *bilinear iterative rational Krylov* (BIRKA) method [15, 47].

**Remark 2.3.34** (The term generalized Sylvester equation)**.** *We call* (2.10) *the generalized Sylvester equation, as is also done in some parts of the literature [16, 17, 32, 122]. However, in other parts of the literature the term is used for other equations such as, e.g.,*

$$AX + YB = EXF,$$

*in [141, 144], where both $X$ and $Y$ are unknown; as well as for the pair of equations*

$$A_1X - YA_2 = B_1 \qquad \textit{and} \qquad A_3X - YA_4 = B_2,$$

*in [51, Notes and references for Section 7.7]. The term generalized is also used to denote equation* (2.7)*, i.e.,*

$$A_1XB_1^T + A_2XB_2^T = C_1C_2^T,$$

*that we call the two-sided Sylvester equation; see [125, Section 7] as well as [80], and analogously for the Lyapunov equation in [136]. However, in [80] equation* (2.9) *is termed generalized Lyapunov equation.*

## 2.4 Nonlinear eigenvalue problems

The *nonlinear eigenvalue problem* (NEP) can informally be understood as having $M(\lambda) \in \mathbb{C}^{n\times n}$, a matrix depending on a parameter $\lambda$, with the goal to find a value for $\lambda$ and corresponding vector $x$ such that $M(\lambda)$ is singular and $x$ is in the kernel. More formally we define it as: Let $\Omega$ be a subset of the complex numbers, i.e., $\Omega \subset \mathbb{C}$. Given a function $M$ mapping scalars to matrices, i.e., $M : \Omega \to \mathbb{C}^{n\times n}$, find a pair $(\lambda_0, x_0) \in \Omega \times \mathbb{C}^n$, $x_0 \neq 0$, such that

$$M(\lambda_0)x_0 = 0. \tag{2.11}$$

The more formal function-viewpoint is advantageous since is allows us to define what type of functions we are studying, e.g., in many cases it is assumed that $M$ is an analytic function on $\Omega$, albeit not entire. The NEP has been extensively studied over more than half a century; see, e.g., [114, 91, 137, 57] for overviews of the field; this exposition is in part based on those. The large number of algorithms and techniques developed for NEP have in recent years been incorporated in specialized software, e.g., the SLEPc library [110, 61, 62], and NEP-PACK as described in Section 3.5.

Special cases of the nonlinear eigenvalue problem (2.11) are, e.g.:

- The *linear eigenvalue problem*, $M(\lambda) := A - \lambda I$.
- The *generalized eigenvalue problem*, $M(\lambda) := A - \lambda B$.
- The *quadratic eigenvalue problem*, $M(\lambda) := A_0 + A_1\lambda + A_2\lambda^2$.
- The *polynomial eigenvalue problem*, $M(\lambda) := \sum_{i=0}^{m} A_i\lambda^i$.
- The *rational eigenvalue problem*, $M(\lambda) := \sum_{i=0}^{m_1} A_i\lambda^i + \sum_{i=0}^{m_2} B_i r_i(\lambda)$, where $r_i$ are given rational functions.
- The *delay eigenvalue problem*, $M(\lambda) := \lambda I - A_0 - \sum_{i=1}^{m} A_i e^{-\tau_i\lambda}$, where the scalars $\tau_1, \tau_2, \ldots, \tau_m$ are given (delays).

From the first and second special cases listed above it is clear how the nonlinear eigenvalue problem generalizes the linear eigenvalue problem. Hence, the following definition of an eigenpair generalizes Definition 2.1.1, and thus some of the definitions and propositions below generalizes some results and definitions from the linear eigenvalue problem. To avoid degenerate cases we often (although sometimes implicitly) work with regular NEPs, a concept analogous to regular pencils (Definition 2.1.18); see, e.g., [137].

**Definition 2.4.1** (Regular and singular)**.** A nonlinear eigenvalue problem (2.11) is called *regular* if there exists $\lambda \in \Omega$ such that $\det(M(\lambda)) \neq 0$. A NEP that is not regular is called *singular*.

**Definition 2.4.2** (Eigenpair)**.** For a regular NEP, a pair $(\lambda_0, x_0) \in \Omega \times \mathbb{C}^n$, $x_0 \neq 0$, such that (2.11) is satisfied is called an *eigenpair*. The scalar $\lambda_0$ is called an *eigenvalue*, and the vector $x_0$ is called (a corresponding) *eigenvector*. To be more precise, the vector $x_0$ is a *right eigenvector*; a *left eigenvector* should satisfy $v_0^H M(\lambda_0) = 0$, and $v_0 \neq 0$.

It follows immediately that an equivalent definition of an eigenvalue is a value $\lambda \in \Omega$ such that $g(\lambda) := \det(M(\lambda)) = 0$. However, note that for the NEP the determinant is not necessarily a polynomial in $\lambda$, in general it is not. Nevertheless, if $M : \Omega \to \mathbb{C}^{n\times n}$ is analytic in $\Omega$, then so is $g : \Omega \to \mathbb{C}$. We illustrate a couple of properties of NEPs with the following example. Let $a \in \mathbb{C}$ be a fixed parameter, and consider the NEP

$$M(\lambda) := \begin{bmatrix} 1 & 0 \\ 0 & a \end{bmatrix} + \begin{bmatrix} e^{i\lambda} & 0 \\ 0 & 0 \end{bmatrix}. \tag{2.12}$$

For $a \neq 0$ the NEP is regular. We can see that the eigenvalues are given by solutions to the equation $1 + e^{i\lambda} = 0$. Three observations follow. First, NEPs are a generalization of scalar root-finding problems [57], as can also be seen from $\det(M(\lambda)) = 0$. Second, a NEP can have any number of eigenvalues, from zero to infinity. For the given example the eigenvalues are $\lambda_0 = \pi + 2\pi k$ for $k = \ldots, -2 - 1, 0, 1, 2, \ldots$. Third, eigenvectors of distinct eigenvalues need not be linearly independent [57]. For the given example $\begin{bmatrix} 1 & 0 \end{bmatrix}^T$ is a corresponding eigenvector to all eigenvalues. However, for $a = 0$ the NEP is singular, since $\begin{bmatrix} 0 & 1 \end{bmatrix}^T$ is in the kernel of $M(\lambda)$ regardless of the value of $\lambda$. Another example of a singular NEP is the $1 \times 1$ root-finding problem $M(\lambda) := \begin{bmatrix} 1 - |e^{i\lambda}| \end{bmatrix} = 0$. We see that the eigenvalues of the latter example are all $\lambda_0 \in \mathbb{R}$. Hence, the eigenvalues form a continuum. However, for a regular and analytic NEP, the eigenvalues are isolated [57, Theorem 2.1]. A third example illustrating properties of the NEP is

$$M(\lambda) := \begin{bmatrix} \lambda^k & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & \frac{1}{\lambda - a} \end{bmatrix}. \tag{2.13}$$

We directly observe that $g(\lambda) := \det(M(\lambda)) = \lambda^k/(\lambda - a)$. The NEP, here a rational eigenvalue problem, has no (finite) eigenvalue if $a = 0$. It is easy to see if $k = 1$ since $g(\lambda) = 1 \neq 0$. However, even for $k > 1$ it holds that $\lambda = 0$ is not an eigenvalue, since it is a pole. Example (2.13) does not contradict what what said above about $g(\lambda) = 0$ characterizing the eigenvalues, since $0 \notin \Omega$ for $a = 0$. Although, if $a \neq 0$, then $\lambda_0 = 0$ is an eigenvalue.

A general class to which many NEPs belong, e.g., all the NEPs mentioned above, are the *sum of products of matrices and functions* (SPMF). A NEP written in the SPMF format can be expressed as

$$M(\lambda) := \sum_{i=0}^{m} A_i f_i(\lambda), \tag{2.14}$$

where $f_i$ are scalar valued functions, i.e., $f_i : \Omega \to \mathbb{C}$, for $i = 1, 2, \ldots, m$. In particular, all analytical NEPs can be written in the SPMF format, with $m \leq n^2$; see, e.g., [21, 57]. However, from a computational perspective it might not be desirable; see, e.g., the problems treated in [43] and [75]. With respect to example (2.12) we note that it can be written with $f_1(\lambda) = 1$ and $f_2(\lambda) = e^{i\lambda}$, and remark that singularity in the case of $a = 0$ stems from a common kernel of $A_1$ and $A_2$. For a NEP written in the SPMF format a common kernel of $A_1, A_2, \ldots, A_m$ is in general a sufficient, but not a necessary, condition for singularity; thus analogous to the case for the generalized eigenvalue problem.

**Definition 2.4.3** (Multiplicity of a root)**.** Let $f : \Omega \to \mathbb{C}$ be an analytic function for some $\Omega \subset \mathbb{C}$. The *root finding problem* associated with $f$ is to find $z \in \mathbb{C}$ such that $f(z) = 0$. A *root* $z_0 \in \mathbb{C}$ is a value such that $f(z_0) = 0$. The *multiplicity* of the root $z_0$ is the smallest integer $k$ such that $f^{(k)}(z_0) \neq 0$.

**Definition 2.4.4** (Algebraic and geometric multiplicity)**.** Let the NEP (2.11) be regular and analytic, and let $\lambda_0 \in \Omega$ be an eigenvalue. The *algebraic multiplicity* of the eigenvalue is defined as the multiplicity of the root $\lambda_0$ to the function $g(\lambda) := \det(M(\lambda))$, i.e., the smallest integer $k$ such that $g^{(k)} \neq 0$. Moreover, the *geometric multiplicity* of the eigenvalue is defined as $\dim(\ker(M(\lambda_0)))$, i.e., the dimension of the kernel of $M(\lambda_0)$.

The classical $1 \times 1$ example $M(\lambda) = \begin{bmatrix}\lambda^k\end{bmatrix}$ has eigenvalue $\lambda_0 = 0$ with algebraic multiplicity $k$, and it shows that, in contrast to the linear eigenvalue problem, for a NEP the algebraic multiplicity is not bounded by the size of the problem; see, e.g., [137, 57]. However, if the NEP is regular and analytic, then the algebraic multiplicity is finite, since the converse would imply that $\det(M(\lambda)) = 0$ identically for all $\lambda \in \Omega$.[10]

The concept of a generalized eigenvector is defined from a Jordan chain.

**Definition 2.4.5** (Jordan chain)**.** Let the NEP (2.11) be regular and analytic. Moreover, let $(\lambda_0, x_0) \in \Omega \times \mathbb{C}^n$ be an eigenpair. A tuple of vectors $(x_0, x_1, \ldots, x_{r-1})$ is called a *Jordan chain* if

$$\sum_{k=0}^{\ell} \frac{1}{k!} M^{(k)}(\lambda_0) x_{\ell-k} = 0 \qquad \text{for } \ell = 0, 1, \ldots, r-1.$$

The vectors $x_1, \ldots, x_{r-1}$ are known as the *generalized eigenvectors*, $r$ is known as the *length* of the Jordan chain, the maximal length of a Jordan chain starting with $x_0$ is known as the *rank* of $x_0$.

We exemplify by writing out the general defining equations of Jordan chains of length 1, 2, and 3. If $(x_0)$, $(y_0, y_1)$ and $(z_0, z_1, z_2)$ are Jordan chains, then it holds that

$$\begin{aligned}
&M(\lambda_0)x_0 = 0 \\
&M(\lambda_0)y_0 = 0 \qquad M(\lambda_0)y_1 = -M'(\lambda_0)y_0 \\
&M(\lambda_0)z_0 = 0 \qquad M(\lambda_0)z_1 = -M'(\lambda_0)z_0 \qquad M(\lambda_0)z_2 = -\frac{1}{2}M''(\lambda_0)z_0 - M'(\lambda_0)z_1.
\end{aligned} \tag{2.15}$$

We see directly that $x_0$, $y_0$, and $z_0$ are eigenvectors corresponding to the eigenvalue $\lambda_0$. In the above example it is possible to take $x_0 = y_0$, as well as $y_0 = z_0$ and $y_1 = z_1$, motivating the notion of the rank of $x_0$. However, it is also possible that $x_0 \neq y_0$, and $x_0 \neq z_0$, although it requires that the geometric multiplicity of the eigenvalue $\lambda_0$ is larger than one.

[10] Consider, e.g., a Taylor series expansion around the eigenvalue with infinite multiplicity.

**Remark 2.4.6** (Jordan chains for the linear eigenvalue problem)**.** *The definition generalizes the notion of Jordan chains for linear eigenvalue problems since if $M(\lambda) = A - \lambda I$, then $M'(\lambda) = -I$, and $M^{(k)}(\lambda) = 0$ for $k \geq 2$. The result is the classical set of equations*

$$(A - \lambda_0 I)x_0 = 0 \quad (A - \lambda_0 I)x_1 = x_0 \quad \dots \quad (A - \lambda_0 I)x_{r-1} = x_{r-2}.$$

*We observe that the equations can be written together in blocked form as,*

$$A \begin{bmatrix} x_0 & x_1 & x_2 & \dots & x_{r-1} \end{bmatrix} = \begin{bmatrix} x_0 & x_1 & x_2 & \dots & x_{r-1} \end{bmatrix} J_{r-1}(\lambda_0).$$

*The observation forms a basis for proving Proposition 2.1.15 about the existence of a Jordan form; namely, the matrix $S$ in the proposition consists of the generalized eigenvectors.*

The definition of a Jordan chain given above is along the lines of [50, Equation (1.28)], [41, Definition 3.1.10], and [137]. However, there is an equivalent way to present a Jordan chain found in, e.g.,[57, Definition 2.3] and [21]. We present the latter in form of a proposition, with an explicit proof, stating the alternative definition as an equivalent characterization.

**Proposition 2.4.7.** *Let the NEP* (2.11) *be regular and analytic, and let $(\lambda_0, x_0) \in \Omega \times \mathbb{C}^n$ be an eigenpair. Consider the tuple of vectors $(x_0, x_1, \dots, x_{r-1})$. Moreover, define the vector valued functions $\chi_\ell : \Omega \to \mathbb{C}^n$ as*

$$\chi_\ell(\lambda) := \sum_{k=0}^{\ell} x_k(\lambda - \lambda_0)^k, \tag{2.16}$$

*for $\ell = 0, 1, \dots, r-1$. The tuple $(x_0, x_1, \dots, x_{r-1})$ is a* Jordan chain *if and only if the function $M(\lambda)\chi_\ell(\lambda)$ has a root $\lambda_0$, i.e., $M(\lambda_0)\chi_\ell(\lambda_0) = 0$, and the root is of multiplicity at least $\ell$, for $\ell = 0, 1, \dots, r-1$.*

*Proof.* We begin by stating some relations that we will use to prove the statement. First note that $\chi_\ell(\lambda_0) = x_0$, and hence $M(\lambda_0)\chi_\ell(\lambda_0) = 0$. More generally we observe that the $m$th derivative is

$$\chi_\ell^{(m)}(\lambda) = \sum_{k=m}^{\ell} \frac{k!}{(k-m)!} x_k(\lambda - \lambda_0)^{k-m}.$$

Specifically, at the eigenvalue $\chi_\ell^{(m)}(\lambda_0) = m! x_m$ if $m \leq \ell$ and 0 otherwise. By using these properties we can, for a generic $\ell$ in $0 \leq \ell \leq r-1$, establish the following equalities:

$$M(\lambda_0)\chi_\ell(\lambda_0) = M(\lambda_0)x_0$$
$$\frac{d}{d\lambda}\Big|_{\lambda=\lambda_0} (M(\lambda)\chi_\ell(\lambda)) = M'(\lambda_0)\chi_\ell(\lambda_0) + M(\lambda_0)\chi_\ell'(\lambda_0) = M'(\lambda_0)x_0 + M(\lambda_0)x_1$$

$$\frac{d^2}{d\lambda^2}\Big|_{\lambda=\lambda_0}(M(\lambda)\chi_\ell(\lambda)) = M''(\lambda_0)\chi_\ell(\lambda_0) + \binom{2}{1}M'(\lambda_0)\chi_\ell'(\lambda_0) + M(\lambda_0)\chi_\ell''(\lambda_0)$$
$$= 2\left(\frac{1}{2}M''(\lambda_0)x_0 + M'(\lambda_0)x_1 + M(\lambda_0)x_2\right)$$
$$\vdots$$
$$\frac{d^\ell}{d\lambda^\ell}\Big|_{\lambda=\lambda_0}(M(\lambda)\chi_\ell(\lambda)) = \sum_{k=0}^{\ell}\binom{\ell}{k}M^{(k)}(\lambda_0)\chi_\ell^{(\ell-k)}(\lambda_0)$$
$$= \sum_{k=0}^{\ell}\frac{\ell!}{k!(\ell-k)!}M^{(k)}(\lambda_0)(\ell-k)!x_{\ell-k} = \ell!\left(\sum_{k=0}^{\ell}\frac{1}{k!}M^{(k)}(\lambda_0)x_{\ell-k}\right).$$

The conclusion now follows easily. First, assume that we have a Jordan chain. Then the right-most expressions of all the equalities above are all equal to zero, which proves that $M(\lambda)\chi_\ell(\lambda)$ has a root $\lambda_0$ of multiplicity at least $\ell$, for all $\ell = 0, 1, \ldots, r-1$.

Second, assume that $M(\lambda)\chi_\ell(\lambda)$ has a root $\lambda_0$ of multiplicity at least $\ell$, for $\ell = 0, 1, \ldots, r-1$. Then the left-most expressions of all the equalities above are all equal to zero, which proves that the tuple $(x_0, x_1, \ldots, x_{r-1})$ is a Jordan chain of length $r$. □

In general a vector valued function $\chi(\lambda)$ such that $M(\lambda_0)\chi(\lambda_0) = 0$ for is called a *root function* of $M$ at $\lambda_0$. The functions $\chi_\ell(\lambda)$ defined in (2.16) are root functions corresponding to the Jordan chains $(x_0)$, $(x_0, x_1)$,…,$(x_0, x_1, \ldots, x_{r-1})$. With these notions the rank of an eigenvector can get an equivalent characterization.

**Definition 2.4.8.** Let the NEP (2.11) be regular and analytic, and let $(\lambda_0, x_0) \in \Omega \times \mathbb{C}^n$ be an eigenpair. The *rank* of $x_0$ is the maximum of all multiplicities of root functions $\chi(\lambda)$, such that $\chi(\lambda_0) = x_0$.

There may be multiple Jordan chains for an eigenvalue $\lambda_0$, corresponding to each degree of the geometric multiplicity. However, with the proper definitions it is possible to establish a complete (canonical) system, which is known to always exist [57].

**Definition 2.4.9.** Let the NEP (2.11) be analytic, and let $\lambda_0 \in \Omega$ be an eigenvalue. Moreover, let $d$ be the geometric multiplicity, i.e., $d = \dim(\ker(M(\lambda_0)))$.

Consider $d$ eigenvectors $x_0^1, x_0^2, \ldots, x_0^d$ such that they form a basis of the kernel, i.e., $\operatorname{span}\{x_0^1, x_0^2, \ldots, x_0^d\} = \ker(M(\lambda_0))$. Specifically, let $x_0^1$ be such that the rank is maximal, and for $j = 2, 3, \ldots, d$ let $x_0^j$ be the vector with maximal rank in the complement of the previously defined vectors, i.e.,

$$x_0^j = \arg\max_x \left\{\operatorname{rank}(x) : x \in \ker(M(\lambda_0)\backslash \operatorname{span}\{v_0^k : 1 \le k \le j-1\}\right\}.$$

Denote the rank of $x_0^j$ with $m_j$ and define the corresponding $d$ Jordan chains, i.e., let $(x_0^j, x_1^j, \ldots, x_{m_j-1}^j)$ be Jordan chains for $j = 1, 2, \ldots, d$.

Then the tuples $(x_0^j, x_1^j, \ldots, x_{m_j-1}^j)$ for $j = 1, 2, \ldots, d$ are called a *complete system of Jordan chains* for $M$ at $\lambda_0$.

It is also known as a *canonical set of Jordan chains*, and we emphasize that they are known to always exist. The numbers $m_j$ referenced in Definition 2.4.9 are known as the *partial multiplicities* and are uniquely defined [137]. It is easily seen that these are ordered and at least equal to one, i.e., $m_1 \geq m_2 \geq \cdots \geq m_d \geq 1$. Moreover, the sum of the partial multiplicities is equal to the algebraic multiplicity [137, 57], i.e., $\sum_{j=1}^{d} m_j = \alpha$, where $\alpha$ is the algebraic multiplicity of $\lambda_0$. Hence, the geometric multiplicity is always smaller than or equal to the algebraic multiplicity; as we are used to from the linear eigenvalue problem. We get a corresponding characterization of eigenvalues.

**Definition 2.4.10** (Simple eigenvalue)**.** An eigenvalue to the NEP (2.11) is called *simple* if the algebraic multiplicity equals to 1. Furthermore, an eigenvalue is called *semisimple* if the algebraic multiplicity is strictly larger than 1, and the algebraic and geometric multiplicities are equal.

With reference to the complete system of Jordan chains we have that a simple eigenvalue means that $d = 1$ and $m_1 = 1$, and a semisimple eigenvalue means that $d > 1$ and $m_1 = m_2 = \cdots = m_d = 1$, which is analogous to the linear case. An important difference between the linear and nonlinear eigenvalue problem is that the generalized eigenvectors need not be linearly independent in the latter case. Even 0 is allowed as a generalized eigenvector, albeit not as an eigenvector. However, for simple and semisimple eigenvalues the following can be said about the eigenvectors.

**Proposition 2.4.11** ([57, Theorem 2.5], [137])**.** *Let the NEP* (2.11) *be regular and analytic. Moreover, let $\lambda_0 \in \Omega$ be an eigenvalue, and $v, x \in \mathbb{C}^n$ be corresponding left and right eigenvectors. Then $\lambda_0$ is algebraically simple if and only if $\lambda_0$ is geometrically simple and $v^H M'(\lambda_0)x \neq 0$.*

*Furthermore, assume that $\lambda_0$ is semisimple with algebraic multiplicity $d$. Then the left and right eigenvectors $v_i, x_i \in \mathbb{R}^n$ for $i = 1, 2, \ldots, d$, can be chosen $M'(\lambda_0)$-biorthogonal, i.e., $v_i^H M'(\lambda_0)x_j = 0$ if $i \neq j$ and $v_i^H M'(\lambda_0)x_i \neq 0$.*

If we think of $(\lambda, x)$ as an approximation of an eigenpair to an analytic NEP (2.11), we define the *residual* as $r := M(\lambda)x$, where $r \in \mathbb{C}^n$ and in general $r \neq 0$. As described in [41, 42], the NEP can be described using Cauchy's integral formula (2.2). Hence, under suitable assumptions, the residual can be expressed as

$$r(x, \lambda) = M(\lambda)x = \frac{1}{2\pi i} \oint_{\Gamma} M(z)x(z - \lambda)^{-1} dz,$$

where we have written out the explicit dependence of $r$ on $x$ and $\lambda$. The latter can be naturally extended to a block version; see also, [21].

**Definition 2.4.12** (Block residual)**.** Let the NEP (2.11) be regular and analytic. Moreover, let $X \in \mathbb{C}^{n \times m}$ and $\Lambda \in \mathbb{C}^{m \times m}$ be such such that the eigenvalues of $\Lambda$ are contained in $\Omega$. We define the *block residual* as

$$M(X, \Lambda) := \frac{1}{2\pi i} \oint_{\Gamma} M(z)X(zI - \Lambda)^{-1} dz, \tag{2.17}$$

where $\Gamma$ is a simple, piecewise-smooth, and closed contour strictly enclosing the eigenvalues of $\Lambda$.

**Proposition 2.4.13** ([21, Proposition 2.6])**.** *Let the NEP* (2.11) *be regular and analytic, and consider $M$ written in the SPMF format* (2.14)*. Moreover, let $X$, $\Lambda$ and $M(X,\Lambda)$ be as in Definition 2.4.12. Then*

$$M(X,\Lambda) = \sum_{i=0}^{m} A_i X f_i(\Lambda),$$

*where $f_i(\Lambda)$ is interpreted in the matrix function sense of Section 2.2.*

*Proof.* Direct computation and the usage of Definition 2.2.2 gives

$$\begin{aligned} M(X,\Lambda) &= \frac{1}{2\pi i}\oint_\Gamma M(z)X(zI-\Lambda)^{-1}dz = \frac{1}{2\pi i}\oint_\Gamma \left(\sum_{i=0}^{m} A_i f_i(z) X(zI-\Lambda)^{-1}\right) dz \\ &= \sum_{i=0}^{m} A_i X \left(\frac{1}{2\pi i}\oint_\Gamma f_i(z)(zI-\Lambda)^{-1}dz\right) = \sum_{i=0}^{m} A_i X f_i(\Lambda). \end{aligned}$$

□

**Definition 2.4.14** (Invariant pair)**.** Let the NEP (2.11) be regular and analytic. Moreover, let $X$, $\Lambda$ and $M(X,\Lambda)$ be as in Definition 2.4.12. The pair $(X,\Lambda)$ is called an *invariant pair* if $M(X,\Lambda) = 0$.

**Proposition 2.4.15.** *Let the NEP* (2.11) *be regular and analytic, and let $(X,\Lambda)$ be an invariant pair. Moreover, let $y$ be an eigenvector to $\Lambda$ with eigenvalue $\lambda_0$, i.e. $\Lambda y = \lambda_0 y$. If $Xy \neq 0$, then $(\lambda_0, Xy)$ is an eigenpair to $M$.*

*Proof.* It follows from that $(\Lambda, X)$ is an invariant pair, that $M$ can be written in the SPMF format, and that $M$ is analytic and thus the functions in the SPMF format has power-series expansions, i.e.,

$$\begin{aligned} 0 = M(X,\Lambda)y &= \sum_{i=0}^{m} A_i X f_i(\Lambda) y = \sum_{i=0}^{m} A_i X \left(\sum_{j=0}^{\infty} c_{ij}\Lambda^j\right) y \\ &= \sum_{i=0}^{m} A_i X y \left(\sum_{j=0}^{\infty} c_{ij}\lambda_0^j\right) = \sum_{i=0}^{m} A_i f_i(\lambda_0) X y = M(\lambda_0)(Xy). \end{aligned}$$

□

We see that the notion of invariant pair in some sense extends the notion of eigenpairs, and more generally of invariant subspaces. However, the definition is a bit too broad to be useful, e.g., if $(X,\Lambda)$ is an invariant pair then $\left(\begin{bmatrix} X & X \end{bmatrix}, \operatorname{diag}(\Lambda,\Lambda)\right)$ is an invariant pair;

and $X = 0$ gives a (trivial) invariant pair. The former example is redundant, and the latter not useful, since, e.g., Proposition 2.4.15 is not applicable. A remedy is to require that the (invariant) pair is minimal [79, 137, 57], as in the following definition.

**Definition 2.4.16** (Minimal pair)**.** A pair $(X, \Lambda) \in \mathbb{C}^{n\times m} \times \mathbb{C}^{m\times m}$ is called a *minimal* if the rank of $V_k(X, \Lambda) = m$, where

$$V_k(X, \Lambda) := \begin{bmatrix} X \\ X\Lambda \\ \vdots \\ X\Lambda^{k-1} \end{bmatrix}.$$

The smallest such $k$ is called the *minimality index* of the pair.

The result is related to controllable pairs (page 23) [41, Proposition 3.1.4]. We note that in the generic case the minimality index of a pair is 1 when $m \leq n$, since $m$ vectors of length $n$ are, in the generic case, linearly independent. However, in the context of an invariant pair for NEPs there are situations where $m > n$, since we have seen that there is no upper bound on the number of eigenvalues. In such situation, given that the pair is minimal, the minimality index is for sure greater than 1.

**Proposition 2.4.17** ([79, Lemma 4], [57, Lemma 2.14])**.** *Let the NEP* (2.11) *be regular and analytic, and let* $(X, \Lambda)$ *be an invariant pair. If* $(X, \Lambda)$ *is a minimal invariant pair, then the eigenvalues of* $\Lambda$ *are eigenvalues of the NEP.*

As we see, minimality removes the degenerate cases. However, a minimal invariant pair does not necessarily describe the complete eigenstructure in terms of multiplicities. To distinguish such a case, there is the definition of a simple invariant pair, introduced in [79, Section 2.2], see also [137] (called *complete invariant pair* in [57, Definition 2.16]).

**Definition 2.4.18** (Simple invariant pair)**.** An invariant pair, $(X, \Lambda) \in \mathbb{C}^{n\times m} \times \mathbb{C}^{m\times m}$, is called *simple* if it is minimal, and the algebraic multiplicities of the eigenvalues of $\Lambda$ are identical to the algebraic multiplicities of the corresponding eigenvalues of the NEP (2.11).

It is observed that the entries of a simple invariant pair vary analytically under analytic perturbations of the NEP $M$ [79, p. 362]. The observation is interesting from a perturbation-analysis perspective, and shows that the entity described in the definition (simple invariant pair) is well-posed and hence reasonable to compute. There is also a close connection between minimal invariant pairs and Jordan chains. The intuition is nicely described in [21] by considering the example with a Jordan chain of length 2 of a NEP written in the SPMF format (2.14). A chain of length 2 was explicit written out in (2.15) above, and when the NEP is in the SPMF format it becomes

$$\begin{aligned} \sum_{i=0}^{m} A_i f_i(\lambda_0) x_0 &= 0 \\ \sum_{i=0}^{m} A_i f_i(\lambda_0) x_1 + \sum_{i=0}^{m} A_i f_i'(\lambda_0) x_0 &= 0. \end{aligned}$$

From the definition of a matrix function applied to a Jordan block (Definition 2.2.3), we have that the above equations are equivalent to

$$\sum_{i=0}^{m} A_i \begin{bmatrix} x_0 & x_1 \end{bmatrix} f_i(J_2(\lambda_0)) = 0.$$

Hence, from Proposition 2.4.13 we have that $(\begin{bmatrix} x_0 & x_1 \end{bmatrix}, J_2(\lambda_0))$ is an invariant pair.

**Proposition 2.4.19** ([21, Proposition 2.4], [41, Proposition 3.1.12])**.** *Let the NEP* (2.11) *be regular and analytic. Let $\lambda_0 \in \Omega$ be an eigenvalue and the vectors $x_0^j$ be eigenvectors for $j = 1, 2, \ldots, d$. Moreover, consider the tuples of vectors $(x_0^j, x_1^j, \ldots, x_{m_j-1}^j)$ for $j = 1, 2, \ldots, d$. Every such tuple is a Jordan chain at $\lambda_0$ if and only if $(X, J)$ is an invariant pair, where*

$$X := \begin{bmatrix} x_0^1 & x_1^1 & \ldots & x_{m_1-1}^1 & x_0^2 & x_1^2 & \ldots & x_{m_2-1}^2 & \ldots & x_0^d & x_1^d & \ldots & x_{m_d-1}^d \end{bmatrix}$$

$$J := \begin{bmatrix} J_{m_1}(\lambda_0) & 0 & \ldots & 0 \\ 0 & J_{m_2}(\lambda_0) & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & J_{m_d}(\lambda_0) \end{bmatrix},$$

*and $X \in \mathbb{C}^{n \times (m_1+m_2+\ldots m_d)}$ and $J \in \mathbb{C}^{(m_1+m_2+\ldots m_d)\times(m_1+m_2+\ldots m_d)}$. Moreover, the pair $(X, J)$ is minimal if and only if the eigenvectors $\{x_0^j\}_{j=1}^d$ are linearly independent.*

The result can also be extended to cover distinct eigenvalues, see [41, Theorem 3.1.13].

A simple invariant pair is minimal and characterizes the algebraic multiplicities of these eigenvalues (the statement is unambiguous due to Proposition 2.4.17). Hence, the following can be concluded.

**Corollary 2.4.20** ([21])**.** *A simple invariant pair $(X, \Lambda)$ describes the complete system of Jordan chains for every eigenvalue in $\Lambda$.*

**Remark 2.4.21** ("The" nonlinear eigenvalue problem)**.** *The problem considered in* (2.11) *is a problem with an eigenvalue nonlinearity. However, it should be noted that there are other generalizations of eigenvalue problems, e.g., with eigenvector nonlinearities $M(x)x = \lambda x$, where $M : \Omega \to \mathbb{C}^{n\times n}$ with $\Omega \subset \mathbb{C}^n$. See, e.g., [74, 135]. Nevertheless, we focus on problems of the type* (2.11) *and other types of problems are generally not treated in this thesis.*

# Chapter: 3

# Contribution

We summarize the work and highlight the author's contribution. The first four sections cover the appended papers. The subsections titled "*Outlook and extensions*" outline some additional discussions connected with respective paper and require the reader to be more familiar with the concerned paper. Hence, it may be advisable to read the corresponding paper first. Some of the included material is briefly mentioned in the publications; some of it was, for reasons of space, excluded in the corresponding publication; and some have come from the author's additional thoughts during the work with this thesis.

The two last sections of the chapter cover other work which has been part of the doctoral studies: One software package and one textbook.

## 3.1 Regarding paper A: Sylvester-based preconditioning for the waveguide eigenvalue problem

The paper is co-authored with Giampaolo Mele, Johan Karlsson, and Elias Jarlebring. It is published in *Linear Algebra and its Applications*, volume 542, pages 441–463, April 2018.

**Contribution:** The work was initiated by E. Jarlebring, and builds on the previous work of E. Jarlebring and G. Mele [75]. The ideas were developed in close collaboration with the co-authors, and I participated in the development of the proofs and proof techniques that led up to the theorems. I did, or took part in, most of the implementation, and I was carrying out the main part of the simulations. The writing was mostly done by me and E. Jarlebring. The work is also included in the thesis of G. Mele [92].

Paper A concerns a special nonlinear eigenvalue problem characterizing a waveguide, which was previously studied in [75]. It had been identified that many methods for solving the NEP required the solution of multiple large linear systems, which constituted a bottle-

neck in terms of hardware. The method of choice in Paper A is residual inverse iteration [94], where a linear system with the same system matrix needs to be solved in each step of the algorithm. However, a pre-factorization was too expensive in terms of memory. Therefore, the method is instead to identify the linear system as equivalent to a matrix equation (Proposition A.3.1), and to use implicit vectorization and iterative methods to solve the linear system. The fact that it is the same system matrix in each iteration of the NEP method makes it possible to develop a powerful preconditioner. Although Paper A specifically concerns the method residual inverse iteration applied to the waveguide eigenvalue problem, the solution method can be adopted to other NEPs and solution algorithms.

For development and applications of the Sherman–Morrison–Woodbury (SMW) formula in the context of matrix equations, see [32, 108, 16, 90]. In the case of nonlinear matrix equations, one could consider adapting nonlinear generalizations of the SMW formula, presented in, e.g., [2].

## Outlook and extensions

### On the generalized Sylvester equation

We call the matrix equation studied in Paper A for a generalized Sylvester equation. Hence, the waveguide matrix equation (A.23) should be equivalent to an equation of the form (2.10). The equivalence is established by the following result.

**Proposition 3.1.1** ([68, p. 104])**.** *Assume that $K = uv^H$, with $u, v \in \mathbb{C}^n$ and $X \in \mathbb{C}^{n\times n}$. Then*

$$K \circ X = \operatorname{diag}(u) X \operatorname{diag}(v)^H,$$

*where* $\operatorname{diag}(u)$ *and* $\operatorname{diag}(v)$ *are diagonal matrices in $\mathbb{C}^{n\times n}$ with the elements of $u$ and $v$ on their respective diagonals.*

*Proof.* The proof is by direct calculation and follows by looking element-wise, i.e., $[K \circ X]_{i,j} = \left[uv^H \circ X\right]_{i,j} = u_i \bar{v}_j X_{i,j} = u_i \left[X \operatorname{diag}(\bar{v})\right]_{i,j} = \left[\operatorname{diag}(u) X \operatorname{diag}(\bar{v})\right]_{i,j}$. □

It follows that by, e.g., a singular value decomposition of the discretization of the wavenumber, $K$, the matrix equation (A.23) can be equivalently written in the form (2.10), albeit possibly with a long sum. However, if the wavenumber $\kappa^2(x, z)$ in (A.1a) is separable in the sense that

$$\kappa^2(x, z) = \sum_{i=1}^{m} f_i(x) g_i(z),$$

for some small number $m$, then $[u_i]_\ell = g_i(z_\ell)$ for $\ell = 1, 2, \ldots, n_z$ and $[\bar{v}_i]_k = f_i(x_k)$ for $k = 1, 2, \ldots, n_x$. In the special case that $\kappa^2(x, z) = f(x) + g(z)$, then $K \circ X = \operatorname{diag}(u)X + X \operatorname{diag}(v)^H$ and the term stemming from the wavenumber can be incorporated directly into the Sylvester operator. Hence, it could be interesting to look at preconditioners based on the approximation $\int_0^1 \kappa^2(x, z)dz + \frac{1}{x_+ - x_-} \int_{x_-}^{x_+} \kappa^2(x, z)dx$. However, tests with this preconditioning technique did not show satisfactory convergence for more than very simple geometries of the waveguide.

**A linear systems view of the preconditioner**

We present the ideas of the preconditioner in a way that is in some sense closer to how preconditioners are typically presented. With reference to Section A.2.2 and with a slight abuse of notation we can write the system matrix, (A.8), as

$$M(\gamma) := \begin{bmatrix} \bar{Q}(\gamma) + \bar{K} & C_1 \\ C_2^T & P(\gamma) \end{bmatrix}. \tag{3.1}$$

Loosely speaking $\bar{Q}(\gamma)$ represents the discretized derivatives on the interior and the constant part of the wavenumber, and $\bar{K}$ represents the (varying part of the) wavenumber, and $C_1$, $C_2^T$, and $P(\gamma)$ implements the boundary conditions.[1] Thus, the linear system of equations is

$$\begin{aligned} \bar{Q}(\gamma)q_{\text{int}} + \bar{K}q_{\text{int}} + C_1 q_{\text{ext}} &= r_{\text{int}} \\ C_2^T q_{\text{int}} + P(\gamma)q_{\text{ext}} &= r_{\text{ext}}. \end{aligned}$$

The interpretation is that $q_{\text{int}} = \text{vec}(X)$, where $X$ is the discretized solution on the interior of the domain and $q_{\text{ext}}$ similarly for the two boundary strips at $x = x_-$ and $x = x_+$.[2] The system is solved using a Schur-complement approach and the computational challenge is to solve

$$\bar{Q}(\gamma)q_{\text{int}} + \left(\bar{K} - C_1 P(\gamma)^{-1} C_2^T\right) q_{\text{int}} = r_{\text{int}} - C_1 P(\gamma)^{-1} r_{\text{ext}},$$

for $q_{\text{int}}$. The inversion of $\bar{Q}(\gamma)$ corresponds to solving a Sylvester equation, and the action of $P(\gamma)^{-1}$ can be computed efficiently with an FFT. The preconditioner introduced in Paper A can be understood as considering the following linear system

$$\bar{Q}(\gamma)q_{\text{int}} + \left(\bar{K} - C_1 P(\gamma)^{-1} C_2^T\right) \hat{P} q_{\text{int}} = r_{\text{int}} - C_1 P(\gamma)^{-1} r_{\text{ext}},$$

for some matrix $\hat{P}$. The matrix $\hat{P}$ can be found by considering an (approximation) operator $\Pi$ introduced in the paper in equation (A.26), i.e.,

$$\Pi(X) := \sum_{k=1}^{N} \mathscr{W}_k(X) E_k,$$

with $E_k$ being matrices and $\mathscr{W}_k$ being linear functionals. It follows that

$$\hat{P}q_{\text{int}} := \text{vec}(\Pi(X)) = \sum_{k=1}^{N} \text{vec}(E_k)\,\text{vec}(W_k)^T\,\text{vec}(X) = \sum_{k=1}^{N} \hat{P}_k q_{\text{int}},$$

where we have used that $\mathscr{W}_k$ are a linear functionals (as in the proof of Theorem A.4.1), and defined the rank-1 matrices $\hat{P}_k := \text{vec}(E_k)\,\text{vec}(W_k)^T$. We get $\hat{P} = \sum_{k=1}^{N} \hat{P}_k$. The (relatively) small rank of $\hat{P}$ and the ease of inverting $\bar{Q}(\gamma)$ makes a Sherman–Morrison–Woodbury inversion technique suitable.

[1] More precisely, $C_1$, $C_2^T$, and $P(\gamma)$ are defined in (A.10), (A.11), and (A.12), respectively. Moreover, we define $\bar{K} := \text{diag}(\text{vec}(K) - \bar{k})$, and $\bar{Q}(\gamma) := Q(\gamma) - \bar{K}$; see the definition of $K$ just above (A.10) and compare with the discussion just above Proposition A.3.1.

[2] Compare with the formulation and notation in Propositions A.2.2 and A.3.1

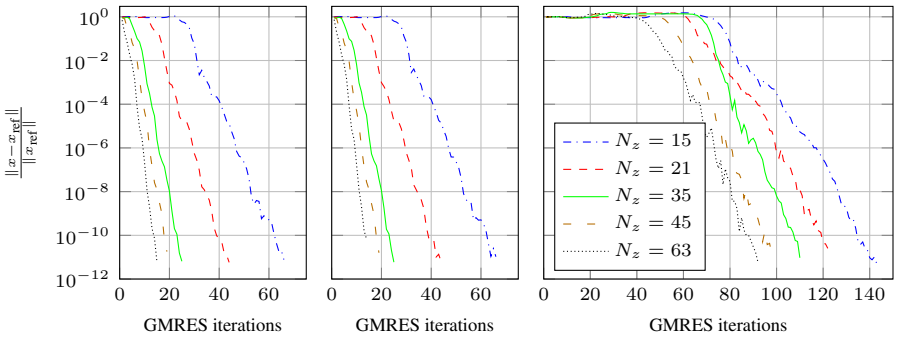



Figure 3.1: GMRES convergence for solving $S(\sigma)x = c$ for different coarse grids. Left plot: grid with 4 refined regions close to the boundary ($n_x = n_z + 8$ and $N_x = N_z + 8$). Middle plot: grid with 2 refined regions close to the boundary ($n_x = n_z + 4$ and $N_x = N_z + 4$). Right plot: uniform grid with no boundary refinement ($n_x = n_z$ and $N_x = N_z$).

#### Further refinement of the boundary

In Figure 3.1 the left plot shows the convergence for the experiments with the additionally refined grid around the boundary, as briefly reported in Section A.6. The middle and right plots of Figure 3.1 are identical to Figure A.3. From this comparison it can be seen that the further grid refinement close to the boundary does not seem to improve the convergence speed, which gives numerical evidence to the explanation regarding the grid refinement presented around equation (A.37).

#### Rank of an eigenvector

One idea relevant to the work of Paper A regards if eigenvectors can be represented as matrices of (numerically) low rank. In such case the Krylov subspace iterations could be adapted to make use of the low-rank structure. A numerical investigation of the rank is presented in Figure 3.2. More precisely, the figure shows the singular values of the matricization of the computed eigenvector approximation, i.e., $\text{mat}(v)$, visualized in Figure A.7b. As can be observed there is a sharp decay in the singular values. However, the rank was deemed not small enough and further attempts to optimize the Krylov subspace iterations were not made.

## 3.2 Regarding paper B: Krylov methods for low-rank commuting generalized Sylvester equations

Paper B is co-authored with Elias Jarlebring, Giampaolo Mele, and Davide Palitta. The paper is published in *Numerical Linear Algebra with Applications*, volume 25, issue 6, pages e2176, December 2018.

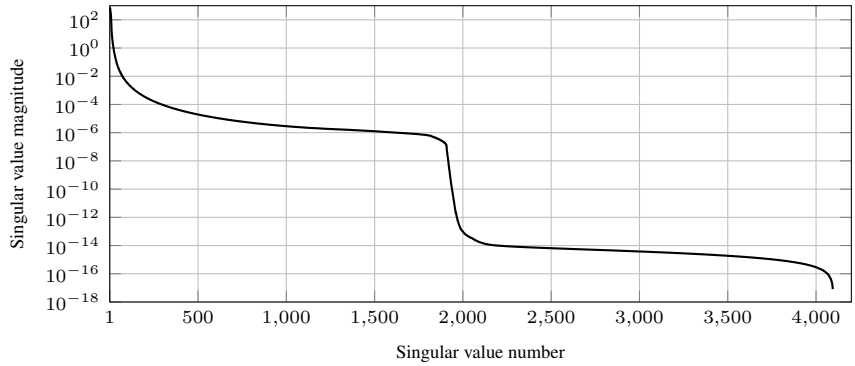



Figure 3.2: Singular values of the eigenvector visualized in Figure A.7(b), where the problem is of size $n_z = 4095$ times $n_x = 4099$.

**Contribution:** The ideas were developed in close collaboration with the co-authors. I participated in developing proofs that transformed intuition into theorems, and took part in some of the implementation. Most of the final editing was done by G. Mele. The results are also addressed in the theses of D. Palitta [95], and G. Mele [92].

Paper B is about method development and problem characterization. It concerns the generalized Sylvester equation (2.10), i.e.,

$$AX + XB^T + \sum_{i=1}^{m} N_i X M_i^T = C_1 C_2^T,$$

under the aforementioned assumption on the spectral radius, i.e., $\rho(\mathscr{L}^{-1}\,\Pi) < 1$. Moreover, for the method development an additional assumption is used, regarding low-rank commutation of the coefficients. We argue that the main contributions of the paper are twofold: The paper motivates when generalized Sylvester equations can be expected to have low-rank solutions, in a way that complements the result in [16]. Furthermore, in the case of low-rank commutation the paper provides a method for computing low-rank approximations of the solution. The term low-rank commutation is introduced in the paper and can be understood as "almost commuting", although the latter typically has a different meaning; see, e.g., [99]. It is well-know among practitioners that computing approximations to generalized Sylvester equations defined by commuting matrices is in general easier. This can be understood by considering the factorization methods discussed for Sylvester equations in Section 2.3 and noting that commuting matrices are simultaneously triangularizable, i.e., the matrices are unitary similar to upper triangular matrices with the same similarity transform (as in Propositon 2.1.16). Pure commutation comes naturally as a special case of the low-rank commutation introduced in Paper B, and for this special case the result reduces to what is already known, i.e., for commuting matrices the standard extended Krylov subspace provides a suitable approximation basis.

In the context of the literature on matrix equations, the fixed-point iteration has been treated in works as [32, 16, 122, 90], and Neumann series has been treated in, e.g., [15]. Moreover, [143, Equations (12)–(13)] treats both the Neumann series and the fixed-point iteration, and infinite series for general linear matrix equations considered as early as in the beginning of the 20th century [89].

The suggested search space has analogies to the multimoment-matching spaces from [27], although with extra structure exploitation in terms of the low-rank commutation.

## Outlook and extensions

### A different version of the low-rank approximability theorem

The result in Theorem B.2.4 can be presented differently, where the dependence on $\ell$ can be made more clear. This was avoided in the final presentation in the paper since the theorem was not intended for practical estimations but rather for theoretical justification and the additional steps in the proof would potentially cloud the main idea. Moreover, the different result is only valid in the Frobenius norm. We nevertheless state the different version of Theorem B.2.4 here.

**Theorem 3.2.1** (Low-rank approximability)**.** *Let $\mathscr{L}$ be the Sylvester operator* (B.1)*, let $\Pi$ be the linear operator from* (B.2)*, and let $C_1, C_2 \in \mathbb{R}^{n\times r}$. Moreover, let $k$ be a positive integer and let $X^{(\ell)}$ be the truncated Neumann series* (B.7)*. Then there exists a matrix $\hat{X}^{(\ell)}$ such that*

$$\operatorname{rank}(\hat{X}^{(\ell)}) \leq (2k+1)r + \sum_{j=1}^{\ell} (2k+1)^{j+1} m^j r,$$

*and*

$$\|X^{(\ell)} - \hat{X}^{(\ell)}\|_F \leq K e^{-\pi\sqrt{k}} \|C_1 C_2^T\|_F \cdot \left( \frac{1-\beta^{\ell+1}}{1-\beta} + P\left( \frac{1-\beta^{\ell}}{(1-\beta)^2} - \frac{\ell\beta^{\ell}}{1-\beta} \right) \right),$$

*where $\beta = \|\mathscr{L}^{-1}\Pi\|_F$, $P = \|\mathscr{L}^{-1}\|_F \|\Pi\|_F$, and $K$ is a constant that only depends on $\mathscr{L}$; specifically $K$ does not depend on $k$ or $\ell$.*

*Proof.* The proof is based on considering the sequence $Y_j$ of the Neumann series (B.6), and creating a sequence of approximations $\Psi_j$ for which we can bound the error and the rank. Consider the following three related sequences of matrices: Let $\Upsilon_0 = \mathscr{L}^{-1}(C_1 C_2^T)$ and $\bar{Y}_0 = \mathscr{L}_k^{-1}(C_1 C_2^T)$, and define the recursions

$$\begin{aligned}
\Psi_j &= \text{ The optimal (SVD-based) approximation of } \Upsilon_j \text{ with rank equal to that of } \bar{Y}_j \\
\Upsilon_j &= -\mathscr{L}^{-1}(\Pi(\Psi_{j-1})) \\
\bar{Y}_j &= -\mathscr{L}_k^{-1}(\Pi(\Psi_{j-1})),
\end{aligned}$$

where the first recursion is valid for $j = 0, 1, 2, \ldots$, the second and third for $j = 1, 2, \ldots$, and operator $\mathscr{L}_k^{-1}$ is the approximation given in equation (B.10). We claim that the matrix $\hat{X}^{(\ell)} := \sum_{j=0}^{\ell} \Psi_j$ fulfils the assertions.

First, note that from construction the rank bound for $\hat{X}^{(\ell)}$ is analogously to the rank bound in the proof of Theorem B.2.4.

Second, to prove the error bound we use the triangle inequality to get

$$\|X^{(\ell)} - \hat{X}^{(\ell)}\|_F \leq \sum_{j=0}^{\ell} \|Y_j - \Psi_j\|_F. \tag{3.2}$$

Hence, we consider $\|Y_j - \Psi_j\|_F$. We use the triangle inequality, and the fact that $\Psi_j$ is not a worse approximation of $\Upsilon_j$ than $\bar{Y}_j$ is, when measured in Frobenius norm. Thus,

$$\begin{aligned}\|Y_j - \Psi_j\|_F &\leq \|Y_j - \Upsilon_j\|_F + \|\Upsilon_j - \Psi_j\|_F \leq \|Y_j - \Upsilon_j\|_F + \|\Upsilon_j - \bar{Y}_j\|_F \\ &\leq \beta\|Y_{j-1} - \Psi_{j-1}\|_F + Ke^{-\pi\sqrt{k}}P\beta^{j-1}\|C_1C_2^T\|_F,\end{aligned}$$

where the last inequality follows from the low-rank approximability result for the Sylvester equation presented in Remark B.2.3, and $\|\Psi_{j-1}\|_F \leq \|\bar{Y}_{j-1}\|_F = \|-\mathscr{L}^{-1}(\Pi(\Psi_{j-2}))\|_F$. The recursion can be solved for equality, and then from non-negativity we conclude that

$$\|Y_j - \Psi_j\|_F \leq Ke^{-\pi\sqrt{k}}\|C_1C_2^T\|_F\beta^{j-1} \cdot (\beta + jP)\,.$$

The total error bound follows from summing up the bounds on each term in the sum (3.2). □

Based on the theorem it is possible to formulate an bound of error $\|X - \hat{X}^{(\ell)}\|_F$ by using the triangle inequality and the result in Remark B.2.2. Note that $\hat{X}^{(\ell)}$ is an explicit construction that gives a bound, and it is neither claimed that it is an optimal construction, nor that it is practically computable. We also note that $\|\mathscr{L}^{-1}\Pi\|_F$ refers to the operator norm induced by the Frobenius norm, which is the same as the induced operator 2-norm of the corresponding Kronecker matrices.

## 3.3 Regarding paper C: Residual-based iterations for the generalized Lyapunov equation

The paper is co-authored with Tobias Breiten and is published in *BIT Numerical Mathematics*, volume 59, pages 823-852, December 2019.

**Contribution:** The results emerged from a trial-and-error process based on a collaboration which to a large degree was framed in discussion with E. Jarlebring and T. Breiten. I worked closely together with, and under the supervision of, T. Breiten on the development of the ideas, the theorems, and the proofs. I did most of the implementation, conducted the simulations, and wrote most of the manuscript. However, the theory presented in Section C.3.2 is entirely the work of T. Breiten.

In Paper C the generalized Lyapunov equation (2.9) is considered, i.e.,

$$AX + XA^T + \sum_{i=1}^{m} N_i X N_i^T = CC^T.$$

Specifically, it is assumed that the $A$-matrix is symmetric and negative definite and that the spectral radius is less than one, i.e., $\rho(\mathscr{L}^{-1}\,\Pi) < 1$. We call such equations *stable generalized Lyapunov equations*. Under these assumptions $\mathscr{L} + \Pi$ constitutes a convergent splitting and the operator $\mathcal{M} := -(\mathscr{L} + \Pi)$ is positive definite. Therefore we can, based on $\mathcal{M}$ and a weighted Frobenius inner product, define an energy norm in the space $\mathbb{R}^{n\times n}$. An iterative strategy searching for rank-1 updates that are local minimums of the error, measured in the energy norm, is presented in [80]. A theoretical justification for the strategy is extended in Paper C, from the stable Lyapunov equation (as in [80]) to the stable generalized Lyapunov equation. Similar convergence results are also established for the fixed-point iteration, which is also shown to minimize an upper bound of the associated energy norm, although without the rank-1 constraint. Moreover, the generalized Lyapunov equation is connected to bilinear systems, the solution to the former being Gramians to the latter as mentioned above (page 41). Hence, there is a connection between model reduction based on $\mathcal{H}_2$-norm minimization of the bilinear systems, and solution methods to the generalized Lyapunov equation based on energy-norm minimization in $\mathbb{R}^{n\times n}$.

The title of Paper C reflects that the residual equation, a standard result in linear algebra, is a common viewpoint for many of the methods treated in the paper. Furthermore, based on this viewpoint, a residual-based generalized rational-Krylov-type subspace is proposed. Although there is no complete characterization of the suggested space, it is shown that it in a certain sense generalizes the rational Krylov subspace for the Lyapunov equation. The suggested search space is also accompanied with different possible changes, thus providing a family of related search spaces.

## Outlook and extensions

### Simulations of an RC circuit

Due to space constraints the following simulation was not included in the final version of the paper. In this example we consider an RC-circuit, reported in [85, Example 3]. Similarly to the examples in Section C.6 it is a bilinear control system, stemming from a Carleman bilinearization (similar to the Burgers' equation in Section C.6.3). We approximate the associated controllability Gramian of a problem of size $5112 \times 5112$, and as in Section C.6.3 the control law is scaled such that the control matrices are scaled with $\alpha = 0.5$. The scaling does not change the dynamical system, although it in some sense changes the difficulty of the computational problem. However, the scaling may change the region in which accurate estimates can be done; see, e.g., [17] for further details.

Convergence of the different methods are plotted in Figures 3.3 and 3.4. The plots to the left show convergence in relative residual norm, and to the right in relative error. The labels in the legends are as in the respective figures in Paper C, and method A are the same in both Figure 3.3 and 3.4. The results are in line with the conclusions of Paper C.

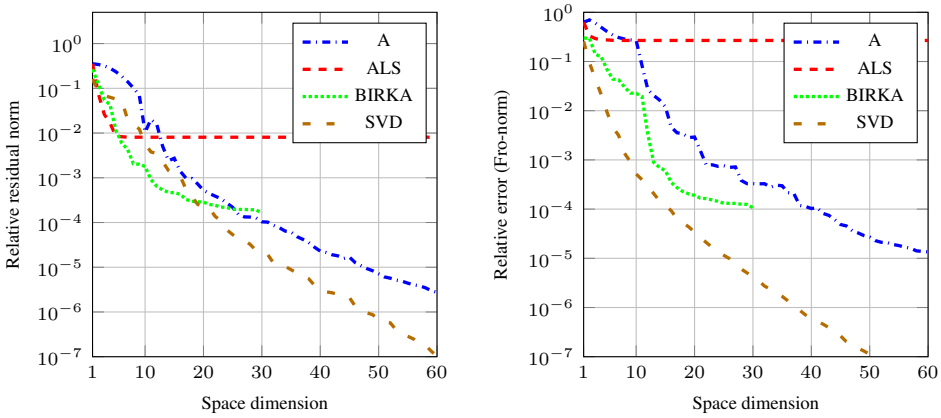



Figure 3.3: Cross-algorithm comparison for RC circuit. Relative residual norm (left) and relative error (right).

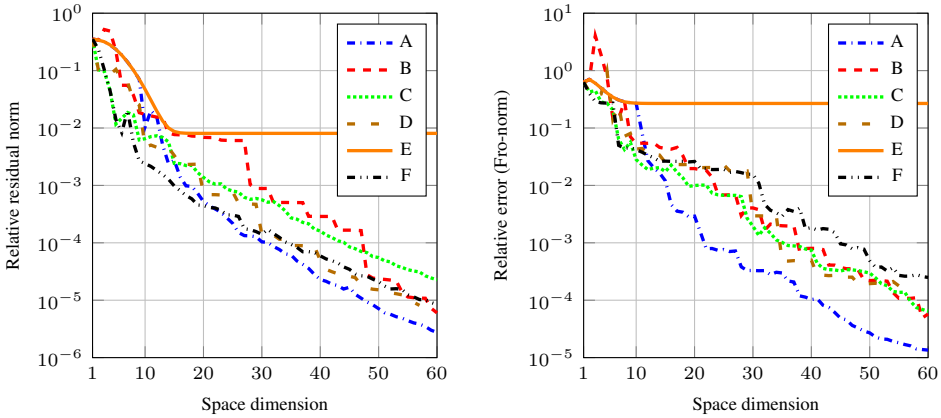



Figure 3.4: Rational-Krylov-type method comparison for RC circuit. Compare with Figure 3.3 as the lines for method A are the same in both figures. For a description of the labels, see the beggnining of Section C.6.

**Rewriting as a nonlinear eigenvalue problem with eigenvector nonlinearities**

Both ALS and BIRKA suffers from that if updates of higher ranks are desired, then the computational costs increase quickly, see Remark C.3.13 and [80, Remark 2.2]. The situation in Paper C is that ALS is applied iteratively to the residual equation, thus forming an inner–outer method for the generalized Lyapunov equation. However, it could be valuable to construct updates of higher ranks directly, as seen, e.g., in a comparison between convergence of BIRKA and ALS. Hence, we investigate possible ways to rewrite the problem that would allow for computing updates of higher ranks, without the need of solving a generalized Sylvester equation (with large left side and small right side). We present some ideas below, although in initial experiments they do unfortunately not seem to yield competitive results.

When utilizing the ALS iteration, or equivalently BIRKA with a rank-1 subspace, we expect, at convergence, to get a solution that satisfies

$$Ax + xv^T A^T v + \sum_{i=1}^{m} N x v^T N^T v + \mathcal{R}_k v = 0, \tag{3.3}$$

where $x = dv$ for some scalar $d$ and $\|v\| = 1$. Numerical investigation suggests that for higher ranks, BIRKA tends to converge to $X \approx VD$, where $V^T V = I$ and $D$ is a diagonal matrix, to about numerical precision. The idea is thus to try to compute a fixed point of this structure in a different way. To keep the initial investigation simpler we start by looking at the rank-1 case and numerically evaluate a few different schemes.

Equation (3.3) can be understood as a nonlinear eigenvalue problem, although not of the type treated above, but with eigenvector nonlinearities (Remark 2.4.21). Since $v$ is a vector, as opposed to a matrix, there is an ambiguity in how to formulate the eigenvector-dependent matrix function. We can formulate the problem both as

$$\left( A + \left(v^T A^T v\right) I + \sum_{i=1}^{m} N_i \left(v^T N_i^T v\right) \right) v = \lambda \mathcal{R}_k v, \tag{3.4}$$

and as

$$\left( A + v v^T A^T + \sum_{i=1}^{m} N_i v v^T N_i^T \right) v = \lambda \mathcal{R}_k v, \tag{3.5}$$

where $\lambda = -1/d$. In both cases we believe that eigenvalues with small magnitude would yield desired directions, since it correspond to vectors with large magnitude in (3.3). The formulation (3.4) is closer to how the ALS iteration looks like, although it utilizes that $\left(v^T A^T v\right)$ and $\left(v^T N_i^T v\right)$ are scalars, which is not true unless $v$ is a vector. On the contrary, that $v$ is a vector is not exploited in formulation (3.5). However, both formulations relies on $d$ being a scalar and are hence not directly applicable when generalized to higher ranks. Thus, we consider non-normalized versions, where we multiply (3.3) with $\sqrt{d}$ from the right (to allow for generalizations to higher ranks). The substitution $w = v\sqrt{d}$ and some

manipulation gives

$$\left( \left( w^T A^T w \right) I + \sum_{i=1}^{m} N_i \left( w^T N_i^T w \right) + \mathcal{R}_k \right) w = \lambda A w, \tag{3.6}$$

and

$$\left( w w^T A^T + \sum_{i=1}^{m} N_i w w^T N_i^T + \mathcal{R}_k \right) w = \lambda A w, \tag{3.7}$$

where $\lambda = -d$. In both cases we believe that looking for eigenvalues with large magnitude would yield desired directions. In addition, if the solution $X$ is real, then we believe that the eigenvalues should be negative and real, which is a complicating factor. However, note that formulation (3.7) would be possible to extend to updates of higher rank, with $V$ and a diagonal matrix $D$.

To test the performance we implement naive fixed-point solvers for the nonlinear eigenvalue problems (3.4)–(3.7) similar to the well-known self-consistent field iteration; see, e.g., [135]. In the implementation we do 10 steps, and additionally look for negative real eigenvalues. The eigenvalues of the generalized eigenvalue problems solved in each step are approximated with the MATLAB command `eigs`, i.e. approximating only a few eigenvalues with a Krylov method. We continue the simplified naming convention from Paper C and use the following short labels:

- G: The described naive fixed-point solver for (3.4)
- H: The described naive fixed-point solver for (3.5)
- I: The described naive fixed-point solver for (3.6)
- J: The described naive fixed-point solver for (3.7).

The approaches are also compared to the ALS approach and method A presented in Paper C. We test on the heat equation and the Burger's equation from Section C.6, and the results are found in Figures 3.5 and 3.6 respectively. We see that in the symmetric case (heat equation) method G has similar performance as ALS and method A. However, the method of interest, J, performs worse. Moreover, in the non-symmetric case (Burgers' equation) the performance is poor for all the methods G, H, I, and J. This discourages further studies of the more complicated case with higher ranks.

As a further note, if $\mathcal{R}_k = uu^T$, i.e., rank 1. Then the solution to (3.4) has the form $v = (A - \sigma I - \sum_{i=1}^{m} \mu_i N_i)^{-1} u$, where $\sigma$ and $\mu_i$ for $i = 1, \ldots, m$ are scalars, although dependent on $v$. The eigenvalue is $\lambda = 1/(u^T (A - \sigma I - \sum_{i=1}^{m} \mu_i N_i)^{-1} u)$. The update for $v$ can be compared with the spaces mentioned in the closing section of the paper, i.e., Section C.7. However, as noted in the section, we have not been able to utilize such type of space efficiently.

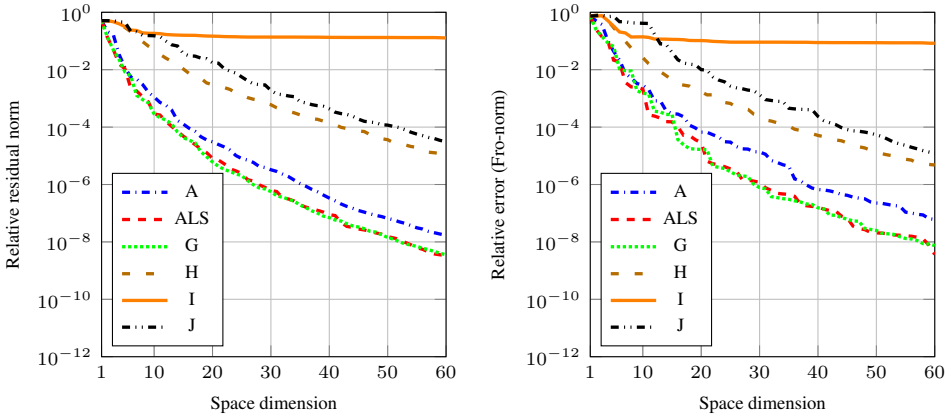



Figure 3.5: NEP-type method comparison for the heat equation. Compare with Figures C.1 and C.2. Relative residual norm (left) and relative error (right).

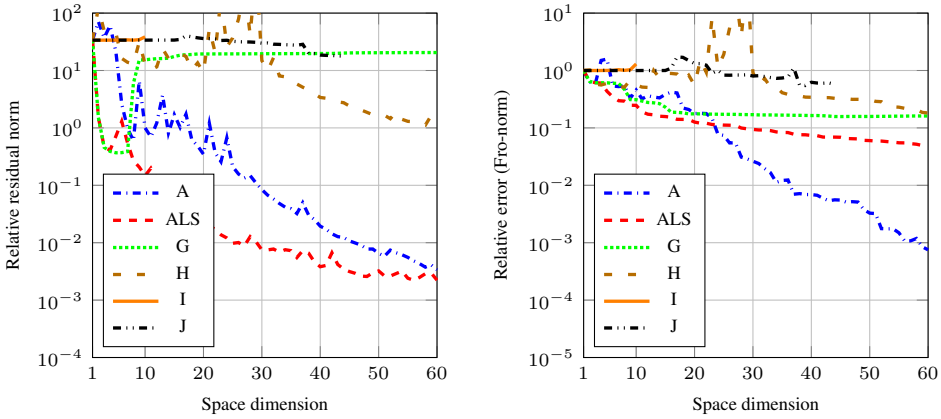



Figure 3.6: NEP-type method comparison for the Burgers' equation. Compare with Figures C.6 and C.7. Relative residual norm (left) and relative error (right).

## 3.4 Regarding paper D: Nonlinearizing two-parameter eigenvalue problems

The paper is co-authored with Elias Jarlebring, and is accepted for publication in *SIAM Journal on Matrix Analysis and Applications*, 2021.

**Contribution:** The initial research idea was due to E. Jarlebring, who also did most of the implementation and performed the numerical experiments in Section D.5. I worked together with, and under the supervision of, E. Jarlebring. The ideas were developed in close collaboration and the writing of the manuscript was a joint effort. I did most of the work on Section D.4.

In paper D we treat the *two-parameter eigenvalue problem*: Given two sets of matrices $A_1, A_2, A_3 \in \mathbb{C}^{n\times n}$ and $B_1, B_2, B_3 \in \mathbb{C}^{m\times m}$, determine eigenvalues $\lambda$ and $\mu$, and eigenvectors $x \in \mathbb{C}^{n\times n}$ and $y \in \mathbb{C}^{m\times m}$, with $x \neq 0$ and $y \neq 0$, such that

$$\begin{aligned} A_1x + \lambda A_2x + \mu A_3x &= 0 \\ B_1y + \lambda B_2y + \mu B_3y &= 0. \end{aligned}$$

If the second equation is of smaller dimension than the first, i.e., $m \ll n$, then we can imagine a variable elimination technique based on solving the second equation.[3] More specifically, for a fixed value $\lambda$ the second equation can be written as the generalized eigenvalue problem (GEP) to find $\mu$ such that

$$-\left(B_1y + \lambda B_2y\right) = \mu B_3y.$$

The GEP corresponds to the pencil $\left(-\left(B_1 + \lambda B_2\right), B_3\right)$; see Definition 2.1.17 (page 9). In principle, for each $\lambda$ we can solve for $\mu$ and thus we have $\mu_i = g_i(\lambda)$ being a family of functions of $\lambda$. When substituted into the first equation we get a NEP,

$$M(\lambda)x := A_1x + \lambda A_2x + g_i(\lambda)A_3x = 0.$$

Except for special cases, such as, e.g., the example in Section D.5.2, the NEP cannot be constructed explicitly. There are, e.g., in general multiple branches $i = 1, 2, \ldots, m$; see Remark D.3.3. However, $M(\lambda)$, as well as its derivative(s), can be evaluated. Thus, it is possible to apply NEP-methods to solve for $x$ and $\lambda$, as well as $\mu = g_i(\lambda)$, while exploiting the special structure of the problem. Results about existence of the nonlinearization, equivalence between solutions, and analyticity of the functions $g_i$ are presented.

We can also imagine a reversed idea. Given a NEP, e.g., of the form above, find a two-parameter eigenvalue problem such that the NEP is a nonlinearization of the two-parameter problem. Under certain conditions the two-parameter eigenvalue problem can be rewritten as a pair of GEPs called the *Delta equations*, (D.4)–(D.7). Hence, the reversed idea, treated in Sections D.2.2 and D.2.3, results in linearizations for NEPs.

[3] The theory does not require such a structure between the two equations. However, the computational efficiency depends on one of the equations being computationally cheaper to solve.

## Outlook and extensions

### An example of linearization

We exemplify the linearization proposed in Section D.2.3 and consider the NEP

$$M_+(\lambda) := A_1 + \lambda A_2 + \sqrt{p(\lambda)}A_3,$$

with $p(\lambda) = \sqrt{2(1+\lambda)(-1+\lambda)}$ and $A_1, A_2, A_3 \in \mathbb{R}^{287\times 287}$ random. With reference to Lemma D.2.5 we have $b = 2$, $c = -1$, $e = 2$, and $f = 1$, and the corresponding linearization is given by the GEP

$$\begin{bmatrix} A_1 & -2A_3 \\ A_3 & A_1 \end{bmatrix} z = \lambda \begin{bmatrix} -A_2 & 2A_3 \\ A_3 & -A_2 \end{bmatrix} z.$$

The GEP is also a linearization of $M_-(\lambda) := A_1+\lambda A_2-\sqrt{p(\lambda)}A_3$. Moreover, the square-root function has branch cuts along the imaginary axis, and the real part is non-smooth in the points $\lambda = \pm 1$ and $\lambda = \pm 1i$. The GEP is solved with the Julia command `eigen` and a corresponding eigenvector $x$ is extracted from a low-rank factorization of $Z \in \mathbb{C}^{287\times 2}$, where $z = \mathrm{vec}(Z)$ is a corresponding eigenvector of the GEP. We measure the error with the relative residual, i.e., $\|M_\pm(\lambda)x\|/\|x\|$, and with a tolerance of $10^{-10}$ all the computed eigenpairs to the GEP can be classified as belonging to exactly one of the two NEPs.[4] For this specific example we get that 284 eigenpairs are from the plus-problem, and 290 from the minus-problem. We also apply the infinite Arnoldi method directly to the NEP, with a set of different expansion points; the result is found in Figure 3.7.

### An example of singular pencils and a note about regularity

The following example is to further elaborate on the necessity of the assumption that the pencil is regular, as discussed in Remark D.2.4,. Let $A_1$, $A_2$, and $A_3$ be suitable matrices, possibly random. Specifically, let

$$B_1 = \begin{bmatrix} 1 & 0 \\ 0 & 3 \end{bmatrix} \quad B_2 = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} \quad B_3 = \begin{bmatrix} 0 & 0 \\ 0 & -1 \end{bmatrix}.$$

We note that if we fix $\lambda = -1$, then the pencil $(-(B_1 + \lambda B_2), B_3)$ is singular since the vector $\begin{bmatrix} 1 & 0 \end{bmatrix}^T$ is in a common kernel. In the literature about singular pencils, eigenvalues $\mu$ are defined as the numbers where the rank of $B_1+\lambda B_2+\mu B_3$ is lower than for almost all other values of $\mu$ (Definition 2.1.20). However, in the case of the two-parameter eigenvalue problem the interest is in the so called indeterminate eigenvalue(s), in the tuple-notation denoted as $(0, 0)$, or $0/0$. The reason is that for $\lambda = -1$ the second equation, i.e., (D.1b), is fulfilled independently of the value of $\mu$. Thus, $\mu$ can be (independently!) determined from the first equation, i.e., (D.1a), with $\lambda = -1$ fixed. Hence, we can get a set of eigenvalues

[4] With a lower tolerance some of the eigenpairs are not accurate enough. However, we do not find that a higher tolerance results in any ambiguity.

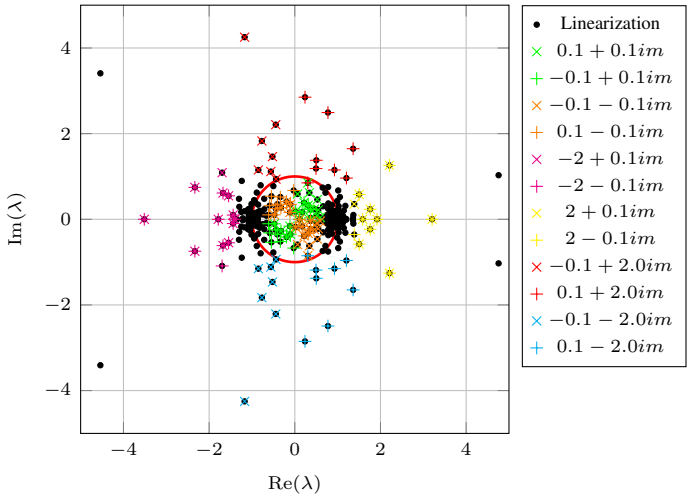



Figure 3.7: Linearization of a NEP. Eigenvalues of $M_+$ computed with the linearization are marked with as black dots. Expansion points of the different runs of the infinite Arnoldi method is described in the legend. The red circle is centered in 0 with radius 1.

$(\lambda, \mu_1), \dots (\lambda, \mu_p)$ for some $p \leq n$.[5] However, in the context of the methodology presented in Paper D, the GEP (D.9) cannot be used to determine $\mu$ as a function of $\lambda$ in these points and the proposed methods cannot be used to find these eigenvalues. Nevertheless, if a singular pencil is found during an iteration, the theoretical difficulty turn into a practical success since a lot of eigenvalues can be readily computed (depending on the $A$-matrices). Specifically for the example, the eigenvalue $\mu$ leading to a solution to the two-parameter eigenvalue problem is characterized by the GEP (D.9), in tuple-notation, as the eigenvalue $(-1-\lambda, 0)$. Thus, $\mu$ can be said to be infinite for all values of $\lambda$, except for $\lambda = -1$ where it is indeterminate (and therefore decoupled from equation (D.1b)).

We make the example concrete with the (random) matrices

$$A_1 = \begin{bmatrix} 0.576838 & 0.193553 & 0.879279 \\ -0.952504 & 0.176529 & -0.535042 \\ 0.109573 & -0.347555 & -0.661395 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} -0.143499 & -0.165619 & 0.553076 \\ -1.72746 & 0.872228 & 0.681078 \\ 1.2422 & -0.159712 & 0.479209 \end{bmatrix}$$

$$A_3 = \begin{bmatrix} 0.916539 & -1.34833 & -1.80063 \\ -0.911508 & 0.374915 & 0.375432 \\ 0.257238 & 0.416975 & 0.149073 \end{bmatrix}.$$

[5] The exact characterization depends on the $A$-matrices and the properties of the pencil corresponding to the GEP (D.1a) with $\lambda = -1$ fixed.

Solving the delta equations (specifically equation (D.7a)) using the Julia command `eigen` gives the following values for $\lambda$:

$$\begin{aligned}
&-1.0648851076328816 - 0.21119187635008602i && -0.9999999999999997 + 0.0i \\
&-1.064885107632882 + 0.21119187635008588i && -0.9999999999999999 + 0.0i \\
&-11.408278083951545 + 0.0i && -1.0000000000000002 + 0.0i,
\end{aligned}$$

where three eigenvalues are (up to about numerical precision) equal to $-1$, as suggested by the discussion above. Using the methodology from Paper D we apply the infinite Arnoldi method and compute the eigenvalue approximations

$$\begin{aligned}
&-1.0648851076328993 - 0.21119187635005965i \\
&-1.064885107632902 + 0.21119187635005895i \\
&-11.408278084043012 - 2.5762450681028322i \cdot 10^{-15}.
\end{aligned}$$

Convergence is achieved towards the eigenvalues that are different from $-1$, as expected.

**An example of non-simple eigenvalues**

Consider the example

$$\begin{aligned}
A_1 &= \begin{bmatrix} 3 & 1 \\ 0 & 7 \end{bmatrix} \; A_2 = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} \; A_3 = \begin{bmatrix} 2 & 0 \\ 0 & 5 \end{bmatrix} \\
B_1 &= \begin{bmatrix} 1 & 1 \\ 0 & 3 \end{bmatrix} \; B_2 = \begin{bmatrix} 0 & 0 \\ 0 & 2 \end{bmatrix} \; B_3 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.
\end{aligned}$$

Note that $\lambda = -1$ and $\mu = -1$ is a solution to the two-parameter eigenvalue problem. More precisely, it is a solution of algebraic multiplicity 4, and geometric multiplicity 1; see, e.g., [102]. The eigenvalues of the GEP (D.9) are $\mu = g(\lambda) = -1$ and $\mu = g(\lambda) = -3 - 2\lambda$. Hence, for $\lambda = -1$ we have that $\mu = -1$ is a double eigenvalue, and specifically a non-semisimple eigenvalue since $\begin{bmatrix} 1 & 0 \end{bmatrix}^T$ is the only eigenvector. Thus, the theory derived in Section D.2 does not guarantee that these eigenvalues of the two-parameter eigenvalue problem can be found with the nonlinearization method. See the discussion in Remark D.2.4. However, in practical tests with the infinite Arnoldi method we manage to compute the eigenvalue $\lambda = -1$ up to multiplicity two. Convergence is plotted in Figure 3.8. We observe that convergence seems to occur simultaneously, and it seem plausible that we find $\lambda = -1$ as a double eigenvalue for $g(\lambda) = -3 - 2\lambda$.

## 3.5 NEP-PACK: A Julia package for nonlinear eigenproblems

NEP-PACK is a software for solving nonlinear eigenvalue problems (page 43). It is a package for the programming language Julia[6] [22]. Within the Julia ecosystem it is known

[6] `https://julialang.org/`

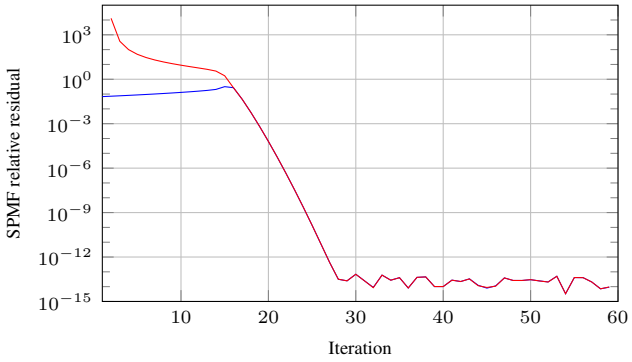



Figure 3.8: The SPMF relative residual is a weighted relative residual measured as $\|(A_1 + \lambda A_2 + g(\lambda)A_3)x\| / (\|x\| (\|A_1\|_F + \|A_2\|_F|\lambda| + \|A_3\|_F|g(\lambda)|))$.

as *NonlinearEigenproblems.jl*. The code is, at the time of writing, available on GitHub[7] and the main co-developers are Elias Jarlebring, Max Bennedich, Giampaolo Mele, and Parikshit Upadhyaya.

**Contribution:** I implemented (most of) the Jacobi–Davidson methods and safeguarded iteration. I moreover took part in the system design, e.g., designing and implementing basic interfaces and types, linear solver structure, and deflation and projection. I was also involved in practical development work such as, e.g., testing and writing documentation.

NEP-PACK implements a large number of state-of-the-art solvers in a coherent way. Thus, fair comparisons between algorithms are facilitated. However, the package allows for specializations based on exploiting problems-specific structures. In order to do so it utilizes the *multiple dispatch* feature built in to the Julia language and relies on a set of interfaces, that are implemented for the standard built-in types and are assumed to exists for user-specified types. Different types of NEP can have different properties, but to access the NEP the follwoing interfaces are defined:

- `compute_Mder(NEP,lambda,k)` returns the $k$th derivative of the NEP $M$ in $\lambda$, i.e.,
$$M^{(k)}(\lambda).$$
- `compute_Mlincomb(NEP,lambda,V,a)` returns a linear combination of the $k$ first derivatives of the NEP $M$ multiplied with the $k$ vectors in $V \in \mathbb{C}^{n\times k}$, and scaled with the coefficients in $a \in \mathbb{C}^k$, respectively, i.e.,
$$\sum_{i=0}^{k} a_i M^{(i)}(\lambda) v_i.$$

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

- `compute_MM(NEP,S,V)` returns the block residual (Definition 2.4.12) of the NEP $M$ evaluated in the pair $(X, S) \in \mathbb{C}^{n\times m} \times \mathbb{C}^{m\times m}$, i.e.,

$$\frac{1}{2\pi i} \oint_{\Gamma} M(z)V(zI - S)^{-1}dz.$$

Although the interfaces have standard implementations for the generic built-in types, the package is agnostic to the actual implementation, which is one point where structure exploitation enabled. As an example, for a NEP in the SPMF format (equation (2.14)), `compute_MM(NEP,S,V)` is computed as in Proposition 2.4.13, i.e., $\sum_{i=0}^{m} A_i V f_i(S)$. The NEP-interfaces are equivalent in the sense that there are ways to convert from one to the others, although it may be computationally costly. Further common interfaces covers steps and computations typically performed by an algorithm, and involves, e.g.,

- `lin_solve(solver, b)` for solving linear systems of the type $M(\lambda)v = b$. The first argument, the solver to be used, is constructed by the algorithm in a call to the interface `create_linsolver(creator,NEP,lambda)`. The latter can be influenced by the user through passing a proper `LinSolverCreator` as an argument to the NEP-method. The structure is useful since, e.g., a fixed-shift method will create a solver object only once, whereas variable-shift methods will need to create a new solver in each iteration. Hence, pre-factorization or construction of preconditioners can be used efficiently.

- projection and solving projected problems, i.e., $\mathcal{W}^H M(\lambda)\mathcal{V}z = 0$. If the original problem is a `ProjectableNEP`, then `create_proj_NEP(orgnep)` creates a wrapper of the original NEP handling the transformation. The interfaces covers both setting and extending of projection matrices. The projected problem can be solved using a variety of different methods and the user can decide by calling a NEP-method with an `InnerSolver`, which in turn defines another NEP-method.

- `estimate_error(errmeasure,lambda,v)` returns a measurement of the error for the approximate eigenpair $(\lambda, v)$ and is called by the NEP-methods when appropriate. If the errmeasure is a function, e.g., `err(lambda,v)`, then it is simply evaluated like that. More complicated behavior can be achieved by using the type `Errmeasure`.

- logging and printing can be adjusted by passing an appropriate `Logger` as an argument to the NEP-method. Default implementations are available for printouts and storing the error history. However, note that error history derives its meaning from the `errmeasure` used, see the point above.

All of these have default behavior, with some defaults relying on the above mentioned NEP-interfaces, but all of them can be adapted by an experienced user depending on the needs. As an example the results from Paper A are also implemented in NEP-PACK, with the linear solver based on the Schur-complement approach and applying the Sylvester-based preconditioning to iterative linear solvers such as GMRES. Moreover, NEP-PACK is used in Paper D to solve the NEPs stemming from the proposed nonlinearization.

The latest release of NEP-PACK can be easily downloaded from within Julia. In the Julia REPL (the interactive command-line prompt) hit the key `]` and write

```
(@v1.5) pkg> add NonlinearEigenproblems
```

Once downloaded, the package can be accessed by typing

```
julia> using NonlinearEigenproblems
```

We show an example on how to create, solve, and work with the NEP

$$M(\lambda) = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} + \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} \lambda^2 + \begin{bmatrix} 9 & 10 \\ 11 & 12 \end{bmatrix} \sin(\lambda).$$

The NEP is easily expressed on the SPMF-format, and can be constructed as follows.

```
julia> A1 = [1.0 2.0; 3.0 4.0];
julia> A2 = [5.0 6.0; 7.0 8.0];
julia> A3 = [9.0 10.0; 11.0 12.0];
julia> f1 = λ -> one(λ);
julia> f2 = λ -> λ^2;
julia> f3 = λ -> sin(λ);
julia> M = SPMF_NEP([A1,A2,A3], [f1,f2,f3]);
```

We apply the *method of successive linear problems* (MSLP), see [114], with the starting guess $\lambda = -1 - 1i$, and ask for the iterates to be printed (`logger=1`).

```
julia> λ,v = mslp(M, λ=-1.0-1.0im, logger=1)
iter 1 err:0.007553404639821915 λ=-0.6374605262901563 - 0.7586814312049742im
iter 2 err:0.0008492664592764308 λ=-0.4885316471868991 - 0.7695470071121764im
iter 3 err:9.395446772710654e-6 λ=-0.4883376602368649 - 0.7856553612819912im
iter 4 err:1.1471784761573207e-9 λ=-0.48837428951380446 - 0.7854814297206311im
iter 5 err:2.590028563744081e-17 λ=-0.4883742950536423 - 0.7854814083852718im
(-0.4883742950536423 - 0.7854814083852718im, Complex{Float64}[0.6732256366906284
+ 0.2162573543777702im, -0.6732256367855572 - 0.2162573484409407im])
```

The output tells us that the method converges, in 5 iterations, to an eigenvalue approximation $\lambda \approx -0.488 - 0.785i$. We can double-check the convergence by computing the relative residual[8], i.e., verify that $\|v\| = 1$ and compute $\|M(\lambda)v\|$ with `compute_Mlincomb`.

```
julia> norm(v)
1.0
julia> norm(compute_Mlincomb(M,λ,v))
9.742167503148516e-16
```

[8] Note that the default error measure for NEPs in the SPMF format is a relative residual weighted with $1/(\sum_{i=0}^{m} \|A_i\|_F |f_i(\lambda)|)$. Specifically, in the given example the weight is `0.026585752738361605`.

We can also verify that $\lambda$ is an eigenvalue by computing the singular values of $M(\lambda)$, utilizing a call to `compute_Mder`. We see that one singular value is numerically zero.

```
julia> svdvals(compute_Mder(M,λ))
2-element Array{Float64,1}:
 14.35097831034869
  7.021666928587148e-16
```

## 3.6 Preconditioning for linear systems

*Preconditioning for linear systems* is a textbook co-authored with Giampaolo Mele, David Ek, Federico Izzo, Parikshit Upadhyaya, and Elias Jarlebring. The book is self-published via Kindle Direct Publishing (Amazon), 2020. A PDF-version is freely available online.[9]

**Contribution:** All authors were more or less involved in all parts of the book. However, for each chapter there are two main authors. I was one of the main author of Chapters 2 (general preconditioners) and 5 (multigrid), as well as involved in proofreading, aligning notation, and compiling the index and references. The final typesetting, figure generation, and proof reading was mostly done by G. Mele and E. Jarlebring.

*Preconditioning* is a technique used to accelerate convergence of iterative methods for solving linear systems. To set the notation we recall the classical setting: Let $A \in \mathbb{C}^{n\times n}$ and $b \in \mathbb{C}^n$ be given, find $x \in \mathbb{C}^n$ such that

$$Ax = b.$$

When the matrix becomes large, direct methods are often too expensive in terms of both number of operations and memory requirements. Hence, iterative methods are applied since these are, typically, cheaper, but in return computes approximations (to some tolerance). Convergence of iterative methods is often characterized in terms of the spectrum of $A$, as well as the *pseudospectra*,[10] but the right-hand side also affects the convergence. Let $P_L, P_R \in \mathbb{C}^{n\times n}$ be nonsingular, then preconditioning can be viewed as working with one of the following equivalent problems

$$\begin{aligned} P_L A x &= P_L b \\ A P_R \left(P_R^{-1} x\right) &= b \\ P_L A P_R \left(P_R^{-1} x\right) &= P_L b, \end{aligned}$$

which are called *left preconditioning*, *right preconditioning*, and *left and right preconditioning*, respectively. The matrix $P_L$ is known as a *left preconditioner* and $P_R$ as a *right*

[9] `http://preconditioning.se`

[10] Pseudospectra characterizes of how sensitive the spectrum of the matrix is with respect to perturbations, which is especially useful for non-normal matrices. Formally it can be defined as $\sigma_\varepsilon(A) := \{\lambda \in \mathbb{C} : \lambda \in \sigma(A+\Delta), \|\Delta\| < \varepsilon\}$, although there are equivalent characterizations, see, e.g., [132, 134].

*preconditioner*. In the presence of a right preconditioner it is typical to consider the variable change $x = P_R y$, i.e., the linear system is solved for $y$, and $x$ is then retrieved by one more application of the right preconditioner. Although we can view preconditioning as working with the equivalent problems listed above, when preconditioners are incorporated into iterative solvers the algorithms are adapted in such a way that the corresponding matrices, e.g., $P_L A$, do not have to be explicitly computed; such computation is both costly and it may destroy features such as sparsity.

Designing a good preconditioner often relies on domain-specific knowledge, although there are general classes and techniques to construct preconditioners. A good preconditioner needs to balance two important factors: First, the preconditioned system must have better convergence properties than the original system, and second, there must be an efficient way of computing the preconditioner (or action thereof). We exemplify this balance with two extreme cases of left preconditioning. First, $P_L = A^{-1}$ has perfect approximation properties, but the computational challenge is as large as the original problem. Second, $P_L = I$ is trivial to compute, but it does not alter the convergence properties of the original linear system.[11]

We briefly describe some of the techniques from chapters 2 and 5 in the book. The former chapter covers classical iterative methods and how these can be utilized to derive preconditioners. The methods involved are the *Jacobi method*:

$$x_{k+1} = D^{-1}\left((L+U)\,x_k + b\right),$$

the *Gauss–Seidel method*:

$$x_{k+1} = (D-L)^{-1}\left(Ux_k + b\right),$$

and *successive over-relaxation* (SOR):

$$x_{k+1} = \omega\,(D-\omega L)^{-1}\left(\left(U + \frac{1-\omega}{\omega}D\right)x_k + b\right),$$

where we have $A = D - L - U$, with $D$ diagonal, $L$ and $U$ strictly lower and upper triangular, respectively, and $\omega$ a parameter in the range $0 < \omega < 2$. The methods are *fixed-point iterations*, and can all be written in a generic form as $x_{k+1} = P^{-1}\left(-Nx_k + b\right)$, where $A = P + N$ for some matrices $P$ and $N$. Specifically,

$$P_{\text{Jacobi}} = D, \qquad P_{\text{Gauss–Seidel}} = D - L, \qquad P_{\text{SOR}} = \frac{1}{\omega}\left(D - \omega L\right).$$

The fixed-point iteration, if convergent, is a way to solve $(I + P^{-1}N)x = P^{-1}b$. By using the identity $N = A - P$, an equivalent way to write the system is $P^{-1}Ax = P^{-1}b$. Hence, these schemes are equivalent to fixed-point iterations on a system preconditioned with $P^{-1}$ [119, Chapter 4]. The fixed-point iteration converges for all right-hand sides $b$ if and only if the spectral radius $\rho(P^{-1}N) < 1$. Moreover, the eigenvalues of the corresponding

[11] The discussion is a direct analogue to the one about projection spaces for the Sylvester equation (page 35).

preconditioned system are $\sigma(P^{-1}A) = \sigma(I + P^{-1}N) = \{z \in \mathbb{C} : z = 1 + \lambda, \text{ where } \lambda \in \sigma(P^{-1}N)\}$. Hence, in the case of a convergent fixed-point iteration, the eigenvalues of the preconditioned system are located in a disk with origin in 1 and radius less than 1.

Another classical method for linear systems is the *LU decomposition* combined with forward and backward substitution, i.e., the matrix is factorized as $A = LU$, where $L$ is lower triangular with ones on the main diagonal and $U$ upper triangular, and the two systems $Ly = b$ and $Ux = y$ are solved. The LU-factorization is a basis for direct methods. However, for large systems it is not desirable to compute the factorization. One problem is that even for a sparse matrix the factors $L$ and $U$ may be full. The *incomplete LU factorization* (ILU) is an approximation based on allowing nonzero elements only in certain positions. Usually a zero pattern is defined which specifies entries to be excluded from the computations, which saves both memory and computation. Let $\mathcal{L}$ and $\mathcal{U}$ be an ILU of $A$, then we write $A = \mathcal{L}\mathcal{U} + \mathcal{R}$, and for a certain class of matrices and class of zero patterns this splitting is a regular splitting. Hence, the fixed-point iteration converges, and from the analysis above about fixed-point iterations and preconditioners we have that $P_L := (\mathcal{L}\mathcal{U})^{-1}$ is a left preconditioner. The preconditioner $P_L$ is never formed explicitly, instead sparsity and structure of $\mathcal{L}$ and $\mathcal{U}$ is exploited so that the preconditioner can be applied efficiently. The derivation and motivation can be explained by connecting matrices to graphs, for details see the textbook. The material is also covered in, e.g., [119].

*Multigrid* is the topic of Chapter 5, and it is a technique originally developed for solving differential equations. It is a class of direct solvers, but less accurate versions are also used as preconditioners. A first guiding idea is the intuition that the solution is more easily obtained on a coarsely discretized grid. By clever interpolation onto a finer grid, that coarse solution can then be used as a starting guess for a solver on the fine grid. The argument can be made more precise by analyzing Fourier modes and their respective frequencies, and how iterative solvers suppresses errors in high frequencies more efficiently. A second guiding idea is based on a correction scheme. Consider the standard problem, i.e., $Ax = b$, and think of $y$ as an approximation of $x$. We define $e := x - y$ and $r := b - Ay$, which can be understood as the *error* and *residual*, respectively. From linearity we have the *residual equation*

$$Ae = r,$$

which relates the residual with the error. The residual $r$ is (fairly) easy to compute, and hence the residual equation allows us compute approximations of the error, that can be used to correct the approximation $y$. Combining the two ideas we can explain the *V-cycle*: The method starts the computations on the fine grid, computes an approximation, e.g., with a few iterations of the (damped) Jacobi method, and projects the residual onto the coarser grid. The procedure can be repeated a desired number of times until we reach the coarsest grid, and on that gird the system i solved with an appropriate solver and to desired accuracy. The computed correction (remember, it was the residual that was projected to a coarser grid) is interpolated to a finer grid, and used to correct the previously computed approximation on that grid. The corrected approximation is used as an initial guess for an iterative method, e.g., the (damped) Jacobi method. The process is repeated until it reaches the finest grid, where it started. The "*V*" in V-cycle is an illustration of the hierarchy of

grids used, where the number of points in the grids is on the $y$-axis, i.e., first from finer to coarser and then from coarser to finer.

*Algebraic multigrid* is an adaptation of multigrid to a general matrix, where there is no underlying physical grid. Hence, what could be previously motivated from geometry, i.e., spatial frequencies of the Fourier modes, selection of grid, and operators for projection and interpolation (although we did not treat the details above), have to be translated to algebraic analogies. Such an endeavor can be completed by, once again, connecting matrices to graphs; for details see, once again, the textbook.

# References

Figure R.1 is a bar chart over the year of publication of the works cited in Part I. The inspiration comes from the corresponding figure in the bibliography of [63], albeit this list of references is by no means as comprehensive.

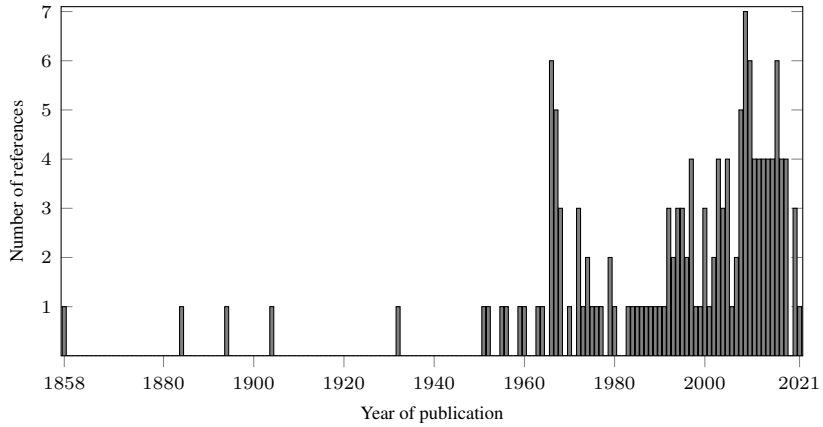



Figure R.1: Year of publication for the references cited in Part I.

[1] L. V. Ahlfors. *Complex analysis*. McGraw-Hill, New York, NY, 3rd edition, 1979.

[2] M. A. Akgün, J. H. Garcelon, and R. T. Haftka. Fast exact linear and non-linear structural reanalysis and the Sherman–Morrison–Woodbury formulas. *Internat. J. Numer. Methods Engrg.*, 50(7):1587–1606, 2001.

[3] S. A. Al-Baiyat and M. Bettayeb. A new model reduction scheme for k-power bilinear systems. In *Proceedings of 32nd IEEE Conference on Decision and Control*, pages 22–27, 1993.

[4] B. D. O. Anderson. Solution of quadratic matrix equations. *Electron. Lett.*, 2:371–372, 1966.

[5] A. Antoulas. *Approximation of Large-Scale Dynamical Systems*. SIAM publications, Philadelphia, PA, 2005.

[6] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst. *Templates for the Solution of Algebraic Eigenvalue Problems*. SIAM publications, Philadelphia, PA, 2000.

[7] Y. Bar-Ness and G. Langholz. The solution of the matrix equation $XC - BX = D$ as an eigenvalue problem. *Internat. J. Systems Sci.*, 8(4):385–392, 1977.

[8] S. Barnett and C. Storey. Solution of the Lyapunov matrix equation. *Electron. Lett.*, 2(12):466–467, 1966.

[9] S. Barnett and C. Storey. Stability analysis of constant linear systems by Lyapunov's second method. *Electron. Lett.*, 2(5):165–166, 1966.

[10] S. Barnett and C. Storey. Comments on the Lyapunov matrix equation [and reply]. *Electron. Lett.*, 3(3):122–123, 1967.

[11] S. Barnett and C. Storey. The Liapunov matrix equation and Schwarz's form. *IEEE Trans. Autom. Control*, 12(1):117–118, 1967.

[12] S. Barnett and C. Storey. Remarks on numerical solution of the Lyapunov matrix equation. *Electron. Lett.*, 3(9):417–418, 1967.

[13] R. H. Bartels and G. W. Stewart. Algorithm 432: Solution of the matrix equation $AX + XB = C$. *Comm. ACM*, 15:820–826, 1972.

[14] R. Bellman. *Introduction to Matrix Analysis*. SIAM publications, Philadelphia, PA, 2nd edition, 1997. Originally published by McGraw-Hill in 1970.

[15] P. Benner and T. Breiten. Interpolation-based $\mathcal{H}_2$-model reduction of bilinear control systems. *SIAM J. Matrix Anal. Appl.*, 33(3):859–885, 2012.

[16] 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.

[17] 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.

[18] P. Benner, P. Kürschner, and J. Saak. Self-generating and efficient shift parameters in ADI methods for large Lyapunov and Sylvester equations. *Electron. Trans. Numer. Anal.*, 43:142–162, 2014.

[19] P. Benner, J.-R. Li, and T. Penzl. Numerical solution of large-scale Lyapunov equations, Riccati equations, and linear-quadratic optimal control problems. *Numer. Linear Algebra Appl.*, 15(9):755–777, 2008.

[20] P. Benner, R.-C. Li, and N. Truhar. On the ADI method for Sylvester equations. *J. Comput. Appl. Math.*, 233(4):1035–1045, 2009.

[21] 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.

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

[23] R. Bhatia and P. Rosenthal. How and why to solve the operator equation $AX - XB = Y$. *Bull. Lond. Math. Soc.*, 29(1):1–21, 1997.

[24] W. G. Bickley and J. McNamee. Matrix and other direct methods for the solution of systems of linear difference equations. *Philos. Trans. A*, 252(1005):69–131, 1960.

[25] R. E. Bixby, M. Fenelon, Z. Gu, E. Rothberg, and R. Wunderling. *The Sharpest Cut: The Impact of Manfred Padberg and His Work*, chapter 18. Mixed-Integer Programming: A Progress Report, pages 309–325. SIAM publications, Philadelphia, PA, 2004.

[26] A. Bouhamidi and K. Jbilou. Sylvester Tikhonov-regularization methods in image restoration. *J. Comput. Appl. Math.*, 206(1):86–98, 2007.

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

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

[29] D. Calvetti and L. Reichel. Application of ADI iterative methods to the restoration of noisy images. *SIAM J. Matrix Anal. Appl.*, 17(1):165–186, 1996.

[30] K.-w. E. Chu. The solution of the matrix equations $AXB - CXD = E$ AND $(YA - DZ, YC - BZ) = (E, F)$. *Linear Algebra Appl.*, 93:93–105, 1987.

[31] P. D'Alessandro, A. Isidori, and A. Ruberti. Realization and structure theory of bilinear dynamical systems. *SIAM J. Control*, 12(3):517–535, 1974.

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

[33] P. I. Davies and N. J. Higham. A Schur–Parlett algorithm for computing matrix functions. *SIAM J. Matrix Anal. Appl.*, 25(2):464–485, 2003.

[34] E. Davison. The numerical solution of $X = A_1X + XA_2 + D$, $X(0) = C$. *IEEE Trans. Autom. Control*, 20(4):566–567, 1975.

[35] E. Davison and F. Man. The numerical solution of $A'Q + QA = -C$. *IEEE Trans. Autom. Control*, 13(4):448–449, 1968.

[36] A. Dmytryshyn. Schur decomposition of several matrices. Technical report, Örebro University, 2020. arXiv:2002.04550.

[37] V. Druskin, L. Knizhnerman, and M. Zaslavsky. Solution of large scale evolutionary problems using rational Krylov subspaces with optimized shifts. *SIAM J. Sci. Comput.*, 31(5):3760–3780, 2009.

[38] V. Druskin, C. Lieberman, and M. Zaslavsky. On adaptive choice of shifts in rational Krylov subspace reduction of evolutionary problems. *SIAM J. Sci. Comput.*, 32(5):2485–2496, 2010.

[39] V. Druskin and V. Simoncini. Adaptive rational Krylov subspaces for large-scale dynamical systems. *Syst. Control Lett.*, 60(8):546–560, 2011.

[40] V. Druskin, V. Simoncini, and M. Zaslavsky. Adaptive tangential interpolation in rational Krylov subspaces for MIMO dynamical systems. *SIAM J. Matrix Anal. Appl.*, 35(2):476–498, 2014.

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

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

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

[44] N. S. Ellner and E. L. Wachspress. New ADI model problem applications. In *Proceedings of 1986 ACM Fall Joint Computer Conference*, ACM '86, pages 528–534, Washington, DC, 1986. IEEE Press.

[45] M. Fasi, N. J. Higham, and B. Iannazzo. An algorithm for the matrix Lambert $W$ function. *SIAM J. Matrix Anal. Appl.*, 36(2):669–685, 2015.

[46] G. Flagg, C. Beattie, and S. Gugercin. Convergence of the iterative rational Krylov algorithm. *Syst. Control Lett.*, 61(6):688–691, 2012.

[47] G. Flagg and S. Gugercin. Multipoint Volterra series interpolation and $\mathcal{H}_2$ optimal model reduction of bilinear systems. *SIAM J. Matrix Anal. Appl.*, 36(2):549–579, 2015.

[48] F. R. Gantmacher. *The theory of matrices*. Chelsea Publishing Company, New York, NY, 1959.

[49] J. D. Gardiner, A. J. Laub, J. J. Amato, and C. B. Moler. Solution of the Sylvester matrix equation $AXB^T + CXD^T = E$. *ACM Trans. Math. Softw.*, 18(2):223–231, 1992.

[50] I. Gohberg, P. Lancaster, and L. Rodman. *Matrix Polynomials*. SIAM publications, Philadelphia, PA, 2009.

[51] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins Univ. Press, Baltimore, MD, 4th edition, 2013.

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

[53] M. Green and D. J. N. Limebeer. *Linear robust control*. Prentice-Hall, Englewood Cliffs, NJ, 1995.

[54] I. Griva, S. G. Nash, and A. Sofer. *Linear and nonlinear optimization*. SIAM publications, Philadelphia, PA, 2nd edition, 2009.

[55] T. Gudmundsson and A. J. Laub. Approximate solution of large sparse Lyapunov equations. *IEEE Trans. Autom. Control*, 39(5):1110–1114, 1994.

[56] S. Gugercin, A. C. Antoulas, and C. Beattie. $\mathcal{H}_2$ model reduction for large-scale linear dynamical systems. *SIAM J. Matrix Anal. Appl.*, 30(2):609–638, 2008.

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

[58] Y. Hao and V. Simoncini. Matrix equation solving of PDEs in polygonal domains using conformal mappings. Accepted for publication in *J. Numer. Math.*, 2020.

[59] E. Heinz. Beiträge zur Störungstheorie der Spektralzerleung. *Math. Ann.*, 123:415–438, 1951.

[60] H. V. Henderson, F. Pukelsheim, and S. R. Searle. On the history of the Kronecker product. *Linear and Multilinear Algebra*, 14(2):113–120, 1983.

[61] V. Hernandez, J. E. Roman, and V. Vidal. SLEPc: Scalable Library for Eigenvalue Problem Computations. *Lect. Notes Comput. Sci.*, 2565:377–391, 2003.

[62] 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.

[63] N. J. Higham. *Functions of Matrices*. SIAM publications, Philadelphia, PA, 2008.

[64] N. J. Higham. The scaling and squaring method for the matrix exponential revisited. *SIAM Rev.*, 51(4):747–764, 2009.

[65] N. J. Higham and A. H. Al-Mohy. Computing matrix functions. *Acta Numer.*, 19:159–208, 2010.

[66] M. Hochbruck and G. Starke. Preconditioned Krylov subspace methods for Lyapunov matrix equations. *SIAM J. Matrix Anal. Appl.*, 16(1):156–171, 1995.

[67] A. S. Hodel, B. Tenison, and K. R. Poolla. Numerical solution of the Lyapunov equation by approximate power iteration. *Linear Algebra Appl.*, 236:205–230, 1996.

[68] R. A. Horn. The Hadamard product. In *Proc. Symp. Appl. Math.*, volume 40, pages 87–169. AMS, 1990.

[69] R. A. Horn and C. .R Johnson. *Topics in Matrix Analysis*. Cambridge Univ. Press, Cambridge, UK, 1st edition, 1994.

[70] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge Univ. Press, Cambridge, UK, 2nd edition, 2012.

[71] D. Y. Hu and L. Reichel. Krylov-subspace methods for the Sylvester equation. *Linear Algebra Appl.*, 172:283–313, 1992.

[72] A. Hurwitz. Zur Invariantentheorie. *Math. Ann.*, 45(3):381–404, 1894.

[73] E. Jarlebring. Numerical methods for matrix functions, 2018. In *Lecture notes in numerical linear algebra*. Course material for the course matrix computations for large-scale systems, given at KTH Royal Institute of Technology.

[74] E. Jarlebring, S. Kvaal, and W. Michiels. An inverse iteration method for eigenvalue problems with eigenvector nonlinearities. *SIAM J. Sci. Comput.*, 36(4):A1978–A2001, 2014.

[75] E. Jarlebring, G. Mele, and O. Runborg. The waveguide eigenvalue problem and the tensor infinite Arnoldi method. *SIAM J. Sci. Comput.*, 39(3):A1062–A1088, 2017.

[76] T. Kato. *Perturbation Theory for Linear Operators*. Classics in Mathematics. Springer-Verlag, Berlin Heidelberg, 2nd edition, 1995.

[77] D. Kleinman. On an iterative technique for Riccati equation computations. *IEEE Trans. Autom. Control*, 13(1):114–115, 1968.

[78] N. N. Krasovskii. *Stability of Motion: Applications of Lyapunov's Second Method to Differential Systems and Equations with Delay*. Stanford Univ. Press, Stanford, CA, 1963. Translated by J. L. Brenner.

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

[80] D. Kressner and P. Sirković. Truncated low-rank methods for solving general linear matrix equations. *Numer. Linear Algebra Appl.*, 22(3):564–583, 2015.

[81] D. Kressner, M. Steinlechner, and B. Vandereycken. Preconditioned low–rank Riemannian optimization for linear systems with tensor product structure. *SIAM J. Sci. Comput.*, 38(4):A2018–A2044, 2016.

[82] D. Kressner and C. Tobler. Krylov subspace methods for linear systems with tensor product structure. *SIAM J. Matrix Anal. Appl.*, 31(4):1688–1714, 2010.

[83] P. Lancaster. Explicit solutions of linear matrix equations. *SIAM Rev.*, 12(4):544–566, 1970.

[84] J.-R. Li and J. White. Low rank solution of Lyapunov equations. *SIAM J. Matrix Anal. Appl.*, 24(1):260–280, 2002.

[85] 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.

[86] C. F. Van Loan. The ubiquitous Kronecker product. *J. Comput. Appl. Math.*, 123(1):85–100, 2000. Numerical Analysis 2000. Vol. III: Linear Algebra.

[87] R. E. Lynch, J. R. Rice, and D. H. Thomas. Tensor product analysis of partial difference equations. *Bull. Amer. Math. Soc.*, 70(3):378–384, 1964.

[88] E. Ma. A finite series solution of the matrix equation $AX - XB = C$. *SIAM J. Appl. Math.*, 14(3):490–495, 1966.

[89] J. H. Maclagan-Wedderburn. Note on the linear matrix equation. *Proc. Edinb. Math. Soc.*, 22:49–53, 1904.

[90] S. Massei, D. Palitta, and L. Robol. Solving rank-structured Sylvester and Lyapunov equations. *SIAM J. Matrix Anal. Appl.*, 39(4):1564–1590, 2018.

[91] V. Mehrmann and H. Voss. Nonlinear eigenvalue problems: A challenge for modern eigenvalue methods. *GAMM-Mitt.*, 27:121–152, 2004.

[92] G. Mele. *Krylov methods for nonlinear eigenvalue problems and matrix equations*. PhD thesis, KTH Royal Institute of Technology, 2020.

[93] C. Moler and C. F. Van Loan. Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM Rev.*, 45(1):3–49, 2003.

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

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

[96] P. C. Parks. A. M. Lyapunov's stability theory—100 years on. *IMA J. Math. Control Inform.*, 9(4):275–303, 1992.

[97] B. N. Parlett. Computation of functions of triangular matrices. Technical Report UCB/ERL M481, EECS Department, University of California, Berkeley, 1974.

[98] D. W. Peaceman and H. H. Rachford, Jr. The numerical solution of parabolic and elliptic differential equations. *J. Soc. Indust. Appl. Math.*, 3(1):28–41, 1955.

[99] C. Pearcy and A. Shields. Almost commuting matrices. *J. Funct. Anal.*, 33(3):332–338, 1979.

[100] T. Penzl. Numerical solution of generalized Lyapunov equations. *Adv. Comput. Math.*, 8:33–48, 1998.

[101] T. Penzl. A cyclic low-rank Smith method for large sparse Lyapunov equations. *SIAM J. Sci. Comput.*, 21(4):1401–1418, 1999.

[102] B. Plestenjak. Numerical methods for nonlinear two-parameter eigenvalue problems. *BIT*, 56(1):241–262, 2016.

[103] J. E. Potter. Matrix quadratic solutions. *SIAM J. Appl. Math.*, 14(3):496–501, 1966.

[104] 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.

[105] H. M. Power. Equivalence of Lyapunov matrix equations for continuous and discrete systems. *Electron. Lett.*, 3(2):83, 1967.

[106] H. M. Power. Solution of Lyapunov matrix equation for continuous systems via Schwarz and Routh canonical forms. *Electron. Lett.*, 3(2):81–82, 1967.

[107] D. A. Reed, R. Bajcsy, M. A. Fernandez, J.-M. Griffiths, R. D. Mott, J. Dongarra, C. R. Johnson, A. S. Inouye, W. Miner, M. K. Matzke, et al. *Computational Science: Ensuring America's Competitiveness*. President's Information Technology Advisory Committee, Arlington, VA, 2005. `https://www.nitrd.gov/Pitac/Reports/20050609_computational/computational.pdf`.

[108] S. Richter, L. D. Davis, and E. G. Collins Jr. Efficient computation of the solutions to modified Lyapunov equations. *SIAM J. Matrix Anal. Appl.*, 14(2):420–431, 1993.

[109] J. D. Roberts. Linear model reduction and solution of the algebraic Riccati equation by use of the sign function. *Internat. J. Control*, 32(4):677–687, 1980. (Reprint of Technical Report No. TR-13, CUED/B-Control, Cambridge University, Engineering Department, 1971).

[110] J. E. Roman, C. Campos, E. Romero, and A. Tomas. SLEPc users manual. Technical Report DSIC-II/24/02 - Revision 3.10, D. Sistemes Informàtics i Computació, Universitat Politècnica de València, 2018.

[111] M. Rosenblum. On the operator equation $BX - XA = Q$. *Duke Math. J.*, 23(2):263–269, 1956.

[112] W. E. Roth. The equations $AX - YB = C$ and $AX - XB = C$ in matrices. *Proc. Amer. Math. Soc.*, 3(3):392–396, 1952.

[113] A. Ruberti, A. Isidori, and P. D'Alessandro. *Theory of bilinear dynamical systems*. Springer-Verlag, Vienna, 1972. Course held at the Department for Automation and Information, July 1972. International Centre for Mechanical Sciences, Udine. Courses and Lectures, No. 158.

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

[115] A. Ruhe. Rational Krylov sequence methods for eigenvalue computation. *Linear Algebra Appl.*, 58:391–405, 1984.

[116] A. Ruhe. The rational Krylov algorithm for nonsymmetric eigenvalue problems. III: complex shifts for real matrices. *BIT*, 34(1):165–176, 1994.

[117] D. E. Rutherford. On the solution of the matrix equation $AX + XB = C$. *Nederl. Akad. Wetensch. Proc. Ser. A*, 35:54–59, 1932. Journal is also known as Proceedings of the Royal Netherlands Academy of Arts and Sciences.

[118] Y. Saad. Numerical solution of large Lyapunov equations. Technical Report 89.20, Research Institute for Advanced Computer Science, NASA Ames Research Center, 1989.

[119] Y. Saad. *Iterative methods for sparse linear systems*. SIAM publications, Philadelphia, PA, 2nd edition, 2003.

[120] Y. Saad. *Numerical methods for large eigenvalue problems*. SIAM publications, Philadelphia, PA, 2nd edition, 2011.

[121] Y. Saad and H. A. van der Vorst. Iterative solution of linear systems in the 20th century. *J. Comput. Appl. Math.*, 123(1):1–33, 2000. Numerical Analysis 2000. Vol. III: Linear Algebra.

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

[123] D. E. Shaw, E. D. Lazowska, F. Berman, S. Brobst, R. E. Bryant, M. Dean, D. Estrin, E. W. Felten, S. L. Graham, W. Gropp, A. K. Jones, M. Kearns, P. Kurtz, R. F. Sproull, M. Maxon, et al. *Designing a digital future: Federally funded research and development in networking and information technology*. President's Council of Advisors on Science and Technology, Washington, DC, 2010. `https://obamawhitehouse.archives.gov/sites/default/files/microsites/ostp/pcast-nitrd-report-2010.pdf`.

[124] V. Simoncini. A new iterative method for solving large-scale Lyapunov matrix equations. *SIAM J. Sci. Comput.*, 29(3):1268–1288, 2007.

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

[126] R. A. Smith. Matrix calculations for Liapunov quadratic forms. *J. Differential Equations*, 2(2):208–217, 1966.

[127] R. A. Smith. Matrix equation $XA+BX = C$. *SIAM J. Appl. Math.*, 16(1):198–201, 1968.

[128] G. Starke and W. Niethammer. SOR for $AX - XB = C$. *Linear Algebra Appl.*, 154–156:355–375, 1991.

[129] G. W. Stewart. On the sensitivity of the eigenvalue problem $Ax = \lambda Bx$. *SIAM J. Numer. Anal.*, 9(4):669–686, 1972.

[130] N. Suzuki. On the convergence of Neumann series in Banach space. *Math. Ann.*, 220(2):143–146, 1976.

[131] J. J. Sylvester. Sur l'équation en matrices $px = xq$. *Comptes Rendus de l'Académie des Sciences*, 99(2):67–71 and 115–116, 1884.

[132] L. N. Trefethen. Pseudospectra of linear operators. *SIAM Rev.*, 39(3):383–406, 1997.

[133] L. N. Trefethen and D. Bau III. *Numerical linear algebra*. SIAM publications, Philadelphia, PA, 1997.

[134] L. N. Trefethen and M. Embree. *Spectra and pseudospectra: The behavior of nonnormal matrices and operators*. Princeton Univ. Press, Princeton, NJ, 2005.

[135] P. Upadhyaya, E. Jarlebring, and E. H. Rubensson. A density matrix approach to the convergence of the self-consistent field iteration. *Numer. Algebra Control Optim.*, 11:99–115, 2021.

[136] B. Vandereycken and S. Vandewalle. A Riemannian optimization approach for computing low-rank solutions of Lyapunov equations. *SIAM J. Matrix Anal. Appl.*, 31(5):2553–2579, 2010.

[137] H. Voss. Nonlinear eigenvalue problems. In L. Hogben, editor, *Handbook of Linear Algebra*, number 164 in Discrete Mathematics and Its Applications. CRC Press, Boca Raton, FL, 2nd edition, 2014.

[138] E. L. Wachspress. Iterative solution of the Lyapunov matrix equation. *Appl. Math. Lett.*, 1(1):87–90, 1988.

[139] Q. Wei, N. Dobigeon, and J. Y. Tourneret. Fast fusion of multi-band images based on solving a Sylvester equation. *IEEE Trans. Image Process.*, 24(11):4109–4121, 2015.

[140] H. K. Wimmer. Contour integral solutions of Sylvester-type matrix equations. *Linear Algebra Appl.*, 493:537–543, 2016.

[141] A.-G. Wu, F. Zhu, G.-R. Duan, and Y. Zhang. Solving the generalized Sylvester matrix equation AV+BW=EVF via a Kronecker map. *Appl. Math. Lett.*, 21(10):1069–1073, 2008.

[142] G. Zehfuss. Uber eine gewisse Determinante. *Zeitschrift für Mathematik und Physik*, 3:298–301, 1858.

[143] L. Zhang and J. Lam. On $H_2$ model reduction of bilinear systems. *Automatica J. IFAC*, 38(2):205–216, 2002.

[144] B. Zhou and G.-R. Duan. A new solution to the generalized Sylvester matrix equation $AV - EVF = BW$. *Syst. Control Lett.*, 55(3):193–198, 2006.

## Sources of quotes and comics

[i] In an interview with Barack Obama. *My Next Guest Needs No Introduction*, Netflix, streaming January 12th, 2018

[ii] R. Munroe. Survivorship Bias. *xkcd: A webcomic of romance, sarcasm, math, and language*. `https://xkcd.com/1827/`

[iii] Attributed to von Neumann by Enrico Fermi, in F. Dyson. A meeting with Enrico Fermi. *Nature*, 427(6972):297, 2004.

[iv] R. Munroe. Here to Help. *xkcd: A webcomic of romance, sarcasm, math, and language*. `https://xkcd.com/1831/`

[v] Mimikry, Song: Alderland, Album: Alderland, 2008.

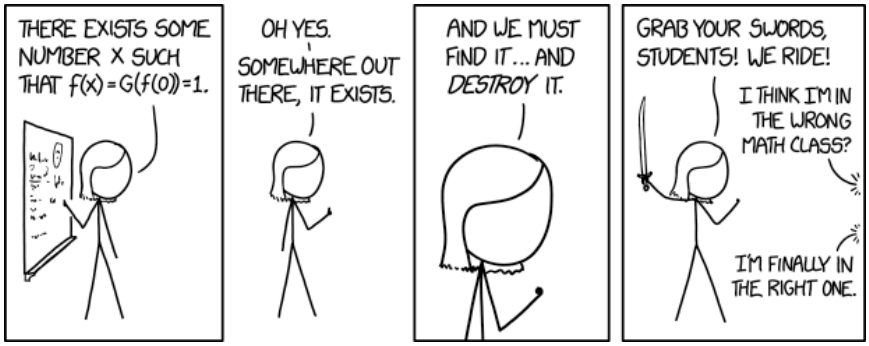
[vi] R. Munroe. Existence proof. *xkcd: A webcomic of romance, sarcasm, math, and language*. `https://xkcd.com/1856/`

[vii] K. Boye. *Kallocain*. Albert Bonniers Förlag, Stockholm, 1940.

# Index

*Först nu kan jag förstå allt det som händer*
*Att Alderland är evigt, enbart människorna byts ut*
*Jag kommer alltid minnas mina vänner*
*Men tiden här för mig har nått sitt slut*
*[. . . ]*
*I Alderland är allting som det ska*

*– Mimikry* [v]




*– xkcd* [vi]

# Part II:
# Research Papers

# Paper D

## Sylvester-based preconditioning for the waveguide eigenvalue problem

# Sylvester-based preconditioning for the waveguide eigenvalue problem

by

Emil Ringh, Giampaolo Mele, Johan Karlsson, Elias Jarlebring

published in *Linear Algebra and its Applications*
Volume 542, Pages 441–463, April 2018

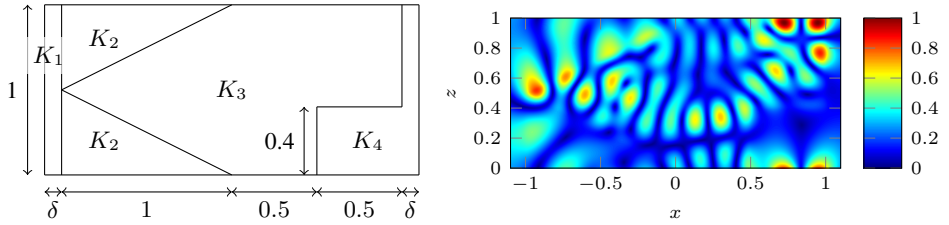
**Abstract**

We consider a nonlinear eigenvalue problem (NEP) arising from absorbing boundary conditions in the study of a partial differential equation (PDE) describing a waveguide. We propose a new computational approach for this large-scale NEP based on residual inverse iteration (Resinv) with preconditioned iterative solves. Similar to many preconditioned iterative methods for discretized PDEs, this approach requires the construction of an accurate and efficient preconditioner. For the waveguide eigenvalue problem, the associated linear system can be formulated as a generalized Sylvester equation $AX + XB + A_1XB_1 + A_2XB_2 + K \circ X = C$, where $\circ$ denotes the Hadamard product. The equation is approximated by a low-rank correction of a Sylvester equation, which we use as a preconditioner. The action of the preconditioner is efficiently computed by using the matrix equation version of the Sherman–Morrison–Woodbury (SMW) formula. We show how the preconditioner can be integrated into Resinv. The results are illustrated by applying the method to large-scale problems.

**Keywords:** Matrix equations, generalized Sylvester equations, PDE-eigenvalue problem, nonlinear eigenvalue problems, preconditioning, iterative methods.

## A.1 Introduction

We are concerned with the study of propagation of waves in a waveguide. The application of two well established techniques (Floquet theory and absorbing boundary conditions) leads to the following characterization of wave propagation in $\mathbb{R}^2$. Details of such a derivation can be found, e.g., in [18, 38].




(a) Geometry of a benchmark waveguide.

(b) The absolute value of an eigenfunction.

Figure A.1: Geometry of the benchmark waveguide and an eigenfunction corresponding to the eigenvalue $\gamma \approx -1.341 - 1.861i$. The same waveguide is used in the numerical examples, Section A.6. The values $K_i$ indicates regions where the wavenumber $\kappa(x, z)$ is constant. For this waveguide $K_1 = \sqrt{2.3}\pi$, $K_2 = 2\sqrt{3}\pi$, $K_3 = 4\sqrt{3}\pi$, $K_4 = \pi$, and $\delta = 0.1$.

The characterization is described by a PDE on a rectangular domain $S_0 := [x_-, x_+] \times [0, 1]$. More precisely, we wish to compute $u : S_0 \to \mathbb{C}$ and $\gamma \in \mathbb{C}$ such that

$$\Delta u(x,z) + 2\gamma u_z(x,z) + (\gamma^2 + \kappa^2(x,z))u(x,z) = 0 \qquad (x,z) \in S_0 \quad \text{(A.1a)}$$

$$u(x,0) = u(x,1) \qquad x \in (x_-, x_+) \quad \text{(A.1b)}$$

$$u_z(x,0) = u_z(x,1) \qquad x \in (x_-, x_+) \quad \text{(A.1c)}$$

$$\mathcal{T}_{-,\gamma}[u(x_-,\cdot)](z) = -u_x(x_-,z) \qquad z \in (0,1) \quad \text{(A.1d)}$$

$$\mathcal{T}_{+,\gamma}[u(x_+,\cdot)](z) = u_x(x_+,z) \qquad z \in (0,1). \quad \text{(A.1e)}$$

The operators $\mathcal{T}_{-,\gamma}$ and $\mathcal{T}_{+,\gamma}$ are the so-called Dirichlet-to-Neumann (DtN) maps, which we specify in Section A.2. The spatially dependent constant $\kappa(x, z)$ is the wavenumber, which in our work is assumed to be piecewise constant. A benchmark example is illustrated in Figure A.1.

Note that (A.1) is a PDE-eigenvalue problem, where the eigenvalue $\gamma$ appears in a nonlinear way in the operator as well as in the boundary conditions, due to the $\gamma$-dependence of the DtNs. The problem (A.1) will be referred to as the waveguide eigenvalue problem (WEP) and we discretize this PDE in a way that allows us to construct an efficient iterative procedure. More precisely, we derive results and methods with a uniform finite-difference (FD) discretization, and also investigate its use in combination with a finite-element method (FEM) discretization. The discretization is presented in Section A.2.2. Due to the nonlinearity in the PDE-eigenvalue problem, the discretized problem is a nonlinear eigenvalue problem (NEP) of the following form: find $(\gamma, v) \in \mathbb{C} \times \mathbb{C}^{n_z n_x + 2n_z} \setminus \{0\}$ such that

$$M(\gamma)v = 0, \quad \text{(A.2)}$$

where $n_x$ and $n_z$ are the number of discretization points in $x$- and $z$-direction respectively.

Over the last few decades, the NEP has been considerably studied in the numerical linear algebra community, and there is a large family of different numerical methods, which we briefly summarize as follows. A number of methods can be seen as flavors of Newton's method, e.g., block Newton methods [21], generalizations of inverse iteration [25, 30] and generalizations of the Jacobi–Davidson method. See PhD thesis [33] for a summary of these methods. A number of approaches are based on numerically computing a contour integral [3, 8], which can be accelerated as described in [42] and references therein. Krylov methods and rational Krylov methods have been generalized in various ways, e.g., the Arnoldi based methods [40, 20], rational Krylov approaches [15, 5, 6]. See the summary papers [24, 30, 41] and the benchmark collection [7] for further literature on methods for nonlinear eigenvalue problems.

Most of these methods involve the solution to the associated linear system of equations

$$M(\sigma)y = r. \tag{A.3}$$

For large-scale problems, the solution to this linear system is often restricting the applicability of the method. In this work we adapt the method called residual inverse iteration (Resinv) which was developed in [25]. Resinv is an iterative method for computing the eigenvalue closest to a given shift $\sigma \in \mathbb{C}$ and it has the attractive feature that the shift $\sigma$ is kept constant throughout the iterations.

The constant shift allows for precomputation, which reduces the computational effort for solving the linear systems (A.3). The standard way to exploit this is to precompute the LU-factorization of $M(\sigma)$. Unfortunately this is not effective for our large-scale problem, due to memory requirements. Instead, we propose to solve (A.3) with a preconditioned iterative method such as GMRES [32] or BiCGStab [39]. Nevertheless, the constant shift and the structure of $M(\sigma)$ allow us to carry out substantial precomputations, related to the preconditioner, in the initialization of Resinv. Other ways to exploit the constant shift, when using Krylov methods to solve repeated linear systems of the type (A.3), is to recycle parts of the invariant subspace of $M(\sigma)$ between different solves, see e.g., [27, 1, 2].

A number of recent approaches exploit that a uniform discretization of a rectangular domain PDE can be expressed as a matrix equation, e.g., using the Sylvester equation or Lyapunov equation. The matrix-equation approach has been used, e.g., in the setting of convection-diffusion equations [26], fractional differential equations [9], PDE-constrained optimization [36], and stochastic differential equations [28]. Inspired by this, we propose a new preconditioner for the WEP based on matrix equations. As a first step, shown in Proposition A.2.2 and Section A.3, the linear system of equations (A.3) is formulated as a matrix equation. In Section A.4, this matrix equation is approximated by a low-rank correction of a Sylvester equation, i.e.,

$$\mathscr{L}(X) + \Pi(X) = C \tag{A.4}$$

where $\mathscr{L}$ is a Sylvester operator and $\Pi$ is a low-rank linear operator of the form $\Pi(X) := \sum_{k=1}^{N} \mathscr{W}_k(X)E_k$, and where $\mathscr{W}_k : \mathbb{C}^{n\times m} \to \mathbb{C}$, $k = 1, \ldots, N$ are linear functionals. We use a matrix equation version of the Sherman–Morrison–Woodbury (SMW) formula

to solve (A.4). In Section A.5 we describe how this can be done in a fast and memory efficient manner using the structures in our problem. A dominating part of the computation is independent of $C$ and can therefore be precomputed. Properties of the approach are illustrated in Section A.6, where we also compare the performance with other approaches.

In this paper we develop specialized techniques for solving the WEP (A.1). The over all framework developed is based on certain general structures of the model problem (A.1), and can be generalized to other problems. Important properties of the model problem are: The dominant operators (the derivatives) only acts in one (spatial) dimension. The problem is defined on a rectangular domain. The PDE is such that a uniform discretization is effective.

The following notation is adopted in this paper. We let $A \circ B$ denote the Hadamard, or element-wise, matrix product between $A$ and $B$, and $A \otimes B$ denotes the Kronecker product. We let $\mathrm{vec}(A) \in \mathbb{C}^{nm}$ denote the vectorization of $A \in \mathbb{C}^{n\times m}$, i.e., the vector obtained by stacking the columns of $A$ on top of each other. The set of eigenvalues of the matrix $A$ is denoted $\mathrm{eig}(A)$. The $n \times n$ identity matrix is denoted $I_n$, and the $i$th column of $I_n$ is denoted $e_i$. The matrix $J_n \in \mathbb{R}^{n\times n}$ denotes the anti-diagonal matrix with ones on the anti-diagonal, that is $[J_n]_{k,\ell} = 1$ if $k = n - \ell + 1$ and 0 otherwise. The column vector consisting of ones is denoted by $\mathbf{1}$.

## A.2 Background and preliminaries

### A.2.1 Problem background

The PDE (A.1) stems from the propagation of waves in a periodic medium. We briefly summarize the derivation and point out the properties needed in our context. See [18, 38] for details.

Consider the Helmholtz's equation

$$\Delta v(x,z) + \kappa(x,z)^2 v(x,z) = 0 \qquad (x,z) \in \mathbb{R}^2, \tag{A.5}$$

where $\kappa(x,z) \in L^\infty(\mathbb{R}^2)$ is the wavenumber. The wavenumber is a 1-periodic function in the $z$-direction which is constant for sufficiently large $|x|$, i.e., $\kappa(x,z+1) = \kappa(x,z)$ for all $x, z$, and there exists real numbers $\xi_-$ and $\xi_+$ such that $\kappa(x,z) = \kappa_-$ for $x < \xi_-$ and $\kappa(x,z) = \kappa_+$ for $x > \xi_+$. We are studying Floquet modes which are solutions corresponding to the ansatz

$$v(x,z) = e^{\gamma z} u(x,z)$$

where $u(x,z+1) = u(x,z)$ in (A.5) and we apply absorbing boundary conditions at $x = x_- \le \xi_-$ and $x = x_+ \ge \xi_+$. From this ansatz we directly identify that $u$ satisfies (A.1a). A more precise analysis (presented in [18]) shows that (A.1d)–(A.1e) also are satisfied where the DtN-maps are defined by

$$\mathcal{T}_{\pm,\gamma}[g](z) := \sum_{k\in\mathbb{Z}} s_{\pm,k}(\gamma) g_k e^{2\pi i k z}, \tag{A.6}$$

where $\{g_k\}_{k\in\mathbb{Z}}$ are the Fourier series coefficients of the function $g(z)$ and $s_k$, for $k \in \mathbb{Z}$, are given by

$$s_{\pm,k}(\gamma) := \operatorname{sign}(\operatorname{Im}(\beta_{\pm,k}(\gamma)))i\sqrt{\beta_{\pm,k}(\gamma)} \tag{A.7a}$$

$$\beta_{\pm,k}(\gamma) := (\gamma + 2\pi i k)^2 + \kappa_{\pm}^2. \tag{A.7b}$$

### A.2.2 Discretization of the WEP

The PDE (A.1) is in this work discretized as follows. We use a uniform FD discretization with $n_x$ and $n_z$ points in $x$- and $z$-direction respectively. The grid consists of the points $x_k = x_- + kh_x$ for $k = 1, 2, \ldots n_x$ where $h_x = (x_+ - x_-)/(n_x + 1)$, and $z_\ell = \ell h_z$ for $\ell = 1, 2, \ldots, n_z$ where $h_z = 1/n_z$.

This FD-discretization leads to the NEP (A.2) described by the following block matrix

$$M(\gamma) := \begin{bmatrix} Q(\gamma) & C_1 \\ C_2^T & P(\gamma) \end{bmatrix}. \tag{A.8}$$

The matrix $Q(\gamma)$ represents the discretization of the interior and the periodic boundary conditions (A.1a)–(A.1c) and $P(\gamma)$ represents the Dirichlet-to-Neumann maps, (A.1d) and (A.1e). The matrix $C_1$ represents the effect of the boundary points to the interior and $C_2^T$ represents the effect of the interior on the boundary constraints, i.e., (A.1d) and (A.1e). The matrix $Q(\gamma) \in \mathbb{C}^{n_x n_z \times n_x n_z}$ is large and sparse, and given by

$$Q(\gamma) := A_0 + \gamma A_1 + \gamma^2 A_2, \tag{A.9}$$

with $A_0 := D_{xx}^T \otimes I_{n_z} + I_{n_x} \otimes D_{zz} + \operatorname{diag}(\operatorname{vec}(K))$, and $A_1 := 2I_{n_x} \otimes D_z$, and $A_2 := I_{n_x n_z}$. Here $D_{xx} \in \mathbb{R}^{n_x \times n_x}$ is the second derivative matrix, and $D_z, D_{zz} \in \mathbb{R}^{n_z \times n_z}$ are the circulant first and second derivative matrices. That is, $D_{xx} = (-2I_{n_x} + Z_{n_x} + Z_{n_x}^T)/h_x^2$, $D_z = (Z_{n_z} + e_1 e_{n_z}^T - Z_{n_z}^T - e_{n_z} e_1^T)/(2h_z)$, and $D_{zz} = (-2I_{n_z} + Z_{n_z} + e_1 e_{n_z}^T + Z_{n_z}^T + e_{n_z} e_1^T)/h_z^2$, where $Z_n \in \mathbb{R}^{n \times n}$ is the shift matrix, defined by $[Z_n]_{k,l} = 1$ if $k - l = 1$ and 0 otherwise.

The matrix $K$ is the discretization of the squared wavenumber, i.e., $[K]_{k,\ell} := \kappa^2(x_\ell, z_k)$. The block $C_1 \in \mathbb{C}^{n_x n_z \times 2n_z}$ is given by

$$C_1 := \frac{1}{h_x^2} \begin{bmatrix} e_1 \otimes I_{n_z} & e_{n_x} \otimes I_{n_z} \end{bmatrix}, \tag{A.10}$$

and the block $C_2^T \in \mathbb{C}^{2n_z \times n_x n_z}$ is given by

$$C_2^T := \begin{bmatrix} d_1 e_1^T \otimes I_{n_z} + d_2 e_2^T \otimes I_{n_z} \\ d_1 e_{n_x}^T \otimes I_{n_z} + d_2 e_{n_x-1}^T \otimes I_{n_z} \end{bmatrix}, \tag{A.11}$$

with $d_1 := \frac{2}{h_x}$, $d_2 := -\frac{1}{2h_x}$. The last block is $P(\gamma) \in \mathbb{C}^{2n_z \times 2n_z}$. To construct this block we truncate the Fourier series expansion of the DtN-maps (A.1d) and (A.1e), i.e., the series

in (A.6). In the truncation we use only the coefficients corresponding to $k = -p, \ldots, p$, and we choose $p$ such that $n_z = 2p + 1$. Then

$$P(\gamma) := \begin{bmatrix} P_-(\gamma) & 0 \\ 0 & P_+(\gamma) \end{bmatrix} = \begin{bmatrix} R\Lambda_-(\gamma)R^{-1} & 0 \\ 0 & R\Lambda_+(\gamma)R^{-1} \end{bmatrix} \tag{A.12}$$

with $\Lambda_\pm(\gamma) := \mathrm{diag}(S_\pm(\gamma))$, where $S_\pm(\gamma) := \left[s_{\pm,-p}(\gamma) + d_0, \ldots, s_{\pm,p}(\gamma) + d_0\right]$, and $d_0 := -\frac{3}{2h_x}$, and where $s_{\pm,k}(\gamma)$ are defined by (A.7a). Moreover, $[R]_{k,\ell} = e^{2\pi i(\ell-p-1)kh_z}$ and $R^* = n_z R^{-1}$.

**Remark A.2.1** (Action of $P(\gamma)$)**.** *Note that $R$ is the Fourier matrix left multiplied with the anti-diagonal matrix having $\left[1, e^{2\pi i p h_z}, e^{2\pi i 2 p h_z}, \ldots, e^{2\pi i (n_z-1) p h_z}\right]$ on its anti-diagonal. Consequently, the action of both $R$ and $R^{-1}$ on a vector can be efficiently calculated with the Fast Fourier Transform (FFT) and from* (A.12) *we conclude that calculating the action of $P_-(\gamma)$, $P_-(\gamma)^{-1}$, $P_+(\gamma)$, and $P_+(\gamma)^{-1}$ on a vector can be done in $\mathcal{O}(n_z \log(n_z))$ operations.*

For future reference, we now also note that when $\gamma$ is in the left-half plane of $\mathbb{C}$ the derivative of $M(\gamma)$ with respect to $\gamma$ is given by

$$M'(\gamma) := \begin{bmatrix} Q'(\gamma) & 0 \\ 0 & P'(\gamma) \end{bmatrix}, \tag{A.13}$$

where $Q'(\gamma) := A_1 + 2\gamma A_2$ and $P'(\gamma) := \mathrm{diag}(R\Lambda'_-(\gamma)R^{-1}, R\Lambda'_+(\gamma)R^{-1})$. The matrices are directly given by $\Lambda'_\pm(\gamma) := \mathrm{diag}(S'_\pm(\gamma))$, where $S'_\pm(\gamma) := \left[s'_{\pm,-p}(\gamma), \ldots, s'_{\pm,p}(\gamma)\right]$, $s'_{\pm,k}(\gamma) := \mathrm{sign}(\mathrm{Im}(\beta_{\pm,k}(\gamma)))i(\gamma + 2\pi i k)/\sqrt{\beta_{\pm,k}(\gamma)}$, and $\beta_{\pm,k}(\gamma)$ are given by (A.7b).

### A.2.3 Residual inverse iteration for the WEP

Our approach is based on the Resinv [25] as a solution method for the NEP (A.2) with $M(\gamma)$ defined by (A.8). Given an approximation to the eigenpair $(\gamma_k, v_k)$, Resinv iteratively computes new approximations in each iteration. In every iteration a new approximation of the eigenvalue $\gamma_{k+1}$ is computed by first solving the nonlinear scalar equation

$$v_k^* M(\gamma_{k+1}) v_k = 0. \tag{A.14}$$

There are different ways of choosing the left vector in (A.14) discussed in the literature [25, 19, 33], but we choose the current approximation of the right eigenvector, as it is presented in the equation. Equation (A.14) is solved with Newton's method in one unknown variable which requires that we calculate the derivative of $v_k^H M(\gamma) v_k$ with respect to $\gamma$. The derivative, for $\gamma$ in the left-half plane of $\mathbb{C}$, can be computed from (A.13). The eigenvector approximation update is done by computing the residual

$$r_k = M(\gamma_{k+1}) v_k, \tag{A.15}$$

and subsequently calculating a correction to the eigenvector by solving

$$\Delta v_k = M(\sigma)^{-1} r_k, \tag{A.16}$$

where $\sigma$ is a fixed shift that is used throughout the whole procedure. Note that since $\sigma$ is a fixed shift, it is possible to make precomputations in relation to solving the linear systems (A.16). The new eigenvector approximation is given by $v_{k+1} = (v_k - \Delta v_k)/\left\|v_k - \Delta v_k\right\|$. The Resinv procedure is summarized in Algorithm A.1.

**Algorithm A.1:** Resinv

**input :** Initial guess of the eigenpair $(\gamma_0, v_0) \in \mathbb{C} \times \mathbb{C}^{n_x n_z + 2n_z}$, with $\|v_0\| = 1$
**output:** An approximation $(\gamma, v) \in \mathbb{C} \times \mathbb{C}^{n_x n_z + 2n_z}$ of $(\gamma_*, v_*) \in \mathbb{C} \times \mathbb{C}^{n_x n_z + 2n_z}$

$\sigma \leftarrow \gamma_0$
**for** $k = 0, 1, 2, \ldots$ **do**
  Compute new approximation of $\gamma_{k+1}$ from (A.14)
  Compute the residual $r_k$ from (A.15)
  Compute the correction $\Delta v_k$ from (A.16)
  $v_{k+1} \leftarrow (v_k - \Delta v_k)/\|v_k - \Delta v_k\|$
$\gamma \leftarrow \gamma_k, \quad v \leftarrow v_k$

Large parts of the computational effort in Algorithm A.1 often consists of the solving of the linear system (A.16) and we present a method that makes the computation feasible for large-scale problems. We use the Schur complement of $M(\sigma)$ with respect to the block $P(\sigma)$,

$$S(\sigma) := Q(\sigma) - C_1 P(\sigma)^{-1} C_2^T, \tag{A.17}$$

to specialize the computation of (A.16) for the WEP. The specialization, which is a direct consequence of the block saddle-point structure in (A.8), is an important step in our algorithm and therefore we present it in the following form.

**Proposition A.2.2** (Schur complement for the WEP)**.** *Let $M(\sigma)$ be as in* (A.8)*, the shift $\sigma \in \mathbb{C}$, and let $S(\sigma)$ be the Schur complement* (A.17)*. Moreover, let $r \in \mathbb{C}^{n_x n_z + 2n_z}$, and let $r_{int}$ be the first $n_x n_z$ elements of $r$ and $r_{ext}$ be the last $2n_z$ elements of $r$. Then*

$$M(\sigma)^{-1} r = \begin{bmatrix} q \\ P(\sigma)^{-1}\left(-C_2^T q + r_{ext}\right) \end{bmatrix} \tag{A.18}$$

*where*

$$q := S(\sigma)^{-1} \tilde{r} \tag{A.19}$$

*and*

$$\tilde{r} := r_{int} - C_1 P(\sigma)^{-1} r_{ext}. \tag{A.20}$$

We use Proposition A.2.2 to solve the linear system in Step 5 in Algorithm A.1. More precisely, we use a preconditioned iterative method to solve (A.19), and the FFT to compute the action $P(\sigma)^{-1}$ (as described in Remark A.2.1). All other operations required for the application of Proposition A.2.2 have negligible computational cost.

## A.3 Matrix equation characterization

In order to construct a good preconditioner for the linear system (A.19) we now formulate it as a matrix equation. Without loss of generality we express (A.19) as $S(\sigma)\mathrm{vec}(X) = \mathrm{vec}(C)$, where $X, C \in \mathbb{C}^{n_z \times n_x}$.

Note that $S(\sigma)$ is defined in (A.17) as the sum of $Q(\sigma)$ and $-C_1 P(\sigma)^{-1} C_2^T$, where $Q(\sigma)$ is described by (A.9). The action of $Q(\sigma)$ can be characterized with matrix equations. By direct application of rules for Kronecker products, see e.g., [17, Section 4.3], it follows that

$$Q(\sigma)\mathrm{vec}(X) = \mathrm{vec}\left((D_{zz} + 2\sigma D_z + \sigma^2 I_{n_z})X + X D_{xx} + K \circ X\right). \tag{A.21}$$

The action of the first two terms of (A.21) can be identified with a Sylvester operator, $\mathscr{L} : \mathbb{C}^{n_z \times n_x} \to \mathbb{C}^{n_z \times n_x}$:

$$\mathscr{L}(X) := AX + XB, \tag{A.22}$$

and hence the action of $Q(\sigma)$ can be viewed as a generalized Sylvester operator. The action corresponding to $S(\sigma)$ can similarly also be constructed as a generalization of the Sylvester operator. We formalize it in the following result, where we also introduce an additional free parameter $\bar{k}$. This parameter is later chosen in such a way that the contribution of the terms corresponding to the Sylvester operator is as large as possible.

**Proposition A.3.1** (Waveguide matrix equation)**.** *Let $X \in \mathbb{C}^{n_z \times n_x}$, let $C \in \mathbb{C}^{n_z \times n_x}$ be a given matrix, and let $S(\sigma)$ be the Schur complement* (A.17)*. Then* $\mathrm{vec}(X)$ *is a solution to* $S(\sigma)\mathrm{vec}(X) = \mathrm{vec}(C)$ *if and only if $X$ is a solution to*

$$AX + XB + (K - \bar{k}\mathbf{1}\mathbf{1}^T) \circ X - P_-(\sigma)^{-1} X E - P_+(\sigma)^{-1} X J_{n_x} E J_{n_x} = C, \tag{A.23}$$

*where $A := D_{zz} + 2\sigma D_z + \sigma^2 I_{n_z} + \bar{k} I_{n_z}$, and $B := D_{xx}$, and $E := \frac{1}{h_x^2}(d_1 e_1 + d_2 e_2) e_1^T$, $d_1$and $d_2$ are given by the discretization, $J_{n_x}$ is the flipped identity as defined in the introduction, and $\bar{k}$ is a free parameter.*

*Proof.* We have that $S(\sigma) = Q(\sigma) - C_1 P(\sigma)^{-1} C_2^T$. The equivalent matrix equation formulation for $Q(\sigma)$ is found apparent from (A.21). The conclusion follows from the calculation

$$\begin{aligned}
&C_1 P(\sigma)^{-1} C_2^T \\
&= \frac{1}{h_x^2}\begin{bmatrix} e_1 \otimes I_{n_z} & e_{n_x} \otimes I_{n_z} \end{bmatrix} \begin{bmatrix} P_-(\sigma)^{-1} & 0 \\ 0 & P_+(\sigma)^{-1} \end{bmatrix} \begin{bmatrix} d_1 e_1^T \otimes I_{n_z} + d_2 e_2^T \otimes I_{n_z} \\ d_1 e_{n_x}^T \otimes I_{n_z} + d_2 e_{n_x-1}^T \otimes I_{n_z} \end{bmatrix} \\
&= e_1 \left( \frac{d_1}{h_x^2} e_1^T + \frac{d_2}{h_x^2} e_2^T \right) \otimes P_-(\sigma)^{-1} + e_{n_x} \left( \frac{d_1}{h_x^2} e_{n_x}^T + \frac{d_2}{h_x^2} e_{n_x-1}^T \right) \otimes P_+(\sigma)^{-1},
\end{aligned}$$

and rules for Kronecker products [17, Section 4.3]. □

## A.4 The Sylvester SMW structure and application to the WEP

### A.4.1 Sylvester-type SMW-structure

Our computational procedure is based on the explicit formula for the inverse of a matrix with a low-rank correction, the Sherman–Morrison–Woodbury (SMW) formula [14, Equation (2.1.4)]. We use the formulation

$$(L + UV^T)^{-1}c = L^{-1}(c - UW^{-1}V^TL^{-1}c) \tag{A.24}$$

where $U, V \in \mathbb{C}^{n\times N}$ and

$$W := I + V^TL^{-1}U \in \mathbb{C}^{N\times N}. \tag{A.25}$$

In order to apply the SMW-formula to equations of the form (A.23), we need a particular matrix equation version of the SMW-formula. The adaption of SMW-formulas to matrix equations has been examined previously in the literature [10, 22, 29]. Our formulation is based on a specialization of [10, Lemma 3.1] that is set up to minimize the memory requirements (as we further discuss in Remark A.4.2).

We select the $L$-matrix in (A.24) as the vectorization of a Sylvester operator (A.22), which is invertible if $\operatorname{eig}(A) \cap \operatorname{eig}(-B) = \emptyset$, see e.g., [17, Theorem 4.4.6]. We make this specific choice since the solution to the Sylvester equation in our case can be computed efficiently. More precisely, the specific structure present in our context can be exploited, as we further describe in Section A.5.3.

In our approach we consider a rank $N$ correction of the Sylvester operator, which can be expressed as a linear operator $\Pi$ of the form

$$\Pi(X) := \sum_{k=1}^{N} \mathscr{W}_k(X)E_k. \tag{A.26}$$

In this setting the matrix $W$ in (A.25) can be expressed in terms of evaluations of the functionals $\mathscr{W}_1, \ldots, \mathscr{W}_N$. This use of the SMW-result is formalized in the following result.

**Theorem A.4.1** (Sylvester-type SMW-structure)**.** *Let $A \in \mathbb{C}^{n\times n}$, $B \in \mathbb{C}^{m\times m}$, and $C \in \mathbb{C}^{n\times m}$ and suppose $\operatorname{eig}(A) \cap \operatorname{eig}(-B) = \emptyset$. Moreover, let the matrices $E_k \in \mathbb{C}^{n\times m}$ and linear functionals $\mathscr{W}_k : \mathbb{C}^{n\times m} \to \mathbb{C}$ be given for $k = 1, 2, \ldots, N$ and define $\Pi : \mathbb{C}^{n\times m} \to \mathbb{C}^{n\times m}$ by* (A.26)*. Assume that there exists a unique solution to the equation*

$$\mathscr{L}(X) + \Pi(X) = C. \tag{A.27}$$

*where $\mathscr{L}$ is the Sylvester operator defined analogous to* (A.22)*. Moreover, let*

$$G := \mathscr{L}^{-1}(C), \qquad \text{and} \tag{A.28a}$$
$$F_k := \mathscr{L}^{-1}(E_k) \qquad \text{for } k = 1, 2, \ldots, N, \tag{A.28b}$$

*and define*

$$W := \begin{bmatrix} 1+\mathscr{W}_1(F_1) & \mathscr{W}_1(F_2) & \dots & \mathscr{W}_1(F_N) \\ \mathscr{W}_2(F_1) & 1+\mathscr{W}_2(F_2) & \dots & \mathscr{W}_2(F_N) \\ \vdots & \vdots & \ddots & \vdots \\ \mathscr{W}_N(F_1) & \mathscr{W}_N(F_2) & \dots & 1+\mathscr{W}_N(F_N) \end{bmatrix}, \quad g := \begin{bmatrix} \mathscr{W}_1(G) \\ \mathscr{W}_2(G) \\ \vdots \\ \mathscr{W}_N(G) \end{bmatrix}. \tag{A.29}$$

*Then the solution to* (A.27) *is given by*

$$X = \mathscr{L}^{-1}\left(C - \sum_{k=1}^{N} \alpha_k E_k\right), \tag{A.30}$$

*where* $a^T = \begin{bmatrix}\alpha_1, \dots, \alpha_N\end{bmatrix}$ *is the unique solution to the system of equations*

$$Wa = g. \tag{A.31}$$

*Proof.* In order to invoke [10, Lemma 3.1] we note that the linear functionals $\mathscr{W}_k$ can be parametrized as $\mathscr{W}_k(X) = \operatorname{vec}(W_k)^T \operatorname{vec}(X)$, for $W_k \in \mathbb{C}^{n\times m}$. Moreover, if we define the matrices $P_1 := [\operatorname{vec}(E_1), \dots, \operatorname{vec}(E_N)]$ and $P_2 := [\operatorname{vec}(W_1), \dots, \operatorname{vec}(W_N)]^T$, then the conclusion (A.30)–(A.31) follows from direct reformulation of [10, Equation (6)]. □

**Remark A.4.2** (Variants of Theorem A.4.1)**.** *Note that, due to the linearity of* $\mathscr{L}^{-1}$*, the solution in* (A.30) *can be equivalently expressed as*

$$X = G - \sum_{k=1}^{N} \alpha_k F_k. \tag{A.32}$$

*Moreover, since* $G, F_1, \dots, F_N$ *can be treated as known,* $X$ *can be computed directly from* (A.32) *without the action of* $\mathscr{L}^{-1}$*. Hence, an approach based on* (A.32) *requires less computational effort than an approach based on* (A.30) *in general. However, in our case,* (A.32) *is not advantageous since it requires more memory resources as we further discuss in Section A.5.2.*

### A.4.2 SMW-structure approximation of the waveguide matrix equation

We saw in the previous section that the matrix equation SMW-formula can be applied to sums of a Sylvester operator $\mathscr{L}$ and the operator $\Pi$ in (A.26). Note that any linear matrix-operator $\Phi(X)$ can be expressed in the form (A.26), by selecting the functionals as $\mathscr{W}_k(X) = e_j^T X e_\ell$ and the matrices $E_k = \Phi(e_j e_\ell^T)$, where $j = 1, \dots, n$, and $\ell = 1, \dots, m$, and $k = j + (\ell - 1)n$ such that $k \in \{1, \dots, nm\}$ and $N = nm$. Unfortunately, such a construction is not practical since Theorem A.4.1 is not computationally attractive for large values of $N$.

Particularly for our problem, equation (A.23) can be efficiently solved if the last terms in (A.23), i.e.,

$$\Phi(X) := (K - \bar{k}\mathbf{1}\mathbf{1}^T) \circ X - P_-(\sigma)^{-1} X E - P_+(\sigma)^{-1} X J E J, \tag{A.33}$$

can be expressed as a low-rank operator $\Pi$ of the form (A.26). Then the solution to (A.23) can be directly computed with Theorem A.4.1. In general, $\Phi$ can only be expressed as an operator $\Pi$ of large rank $N$. Nevertheless, the ranks of the last two operators in (A.33) are bounded by $2n_z$ respectively, since $E$ only has two nonzero elements. Moreover, the first term in (A.33) has low rank for instance if the elements of $K$ equals a constant, $\bar{k}$, except for a few indices. In the continuous formulation, this corresponds to the wavenumber being constant in most parts of the domain.

Although $\Phi$ is in general not a low-rank operator and can therefore only be expressed in the form of (A.26) with a large $N$, we now introduce a low-rank approximation of $\Phi$, called $\Pi$, with $N \ll n_x n_z$. Our construction exploits the structure in $\Phi$ and allows for a representation of $\Pi$ with both low rank $N$ and structured matrices $E_k$.

The construction of $\Pi$ is based on a Galerkin approximation of the solution $X$. More specifically we consider approximations in a vector space $\mathcal{V} \subset \mathbb{C}^{n_z \times n_x}$, with a basis $V_1, \ldots, V_N$, which is assumed to be orthogonal with respect to the trace inner product $\langle X, Y \rangle = \mathrm{Tr}(Y^H X)$. We take the approximation of $X \in \mathbb{C}^{n_z \times n_x}$ from this space and let $\tilde{X} \in \mathcal{V}$ be the best approximation (in the induced trace norm). Equivalently we can impose the Galerkin condition on $X$,

$$\langle X - \tilde{X}, V_k \rangle = 0, \quad \text{for } k = 1, \ldots, N,$$

which leads to the formula $\tilde{X} := \sum_{k=1}^{N} \frac{\langle X, V_k \rangle}{\langle V_k, V_k \rangle} V_k$. Based on this approximation, we construct $\Pi$ as an approximation of $\Phi$ by setting $\Pi(X) := \Phi(\tilde{X})$, i.e.,

$$\Pi(X) := \Phi\left( \sum_{k=1}^{N} \frac{\langle X, V_k \rangle}{\langle V_k, V_k \rangle} V_k \right) = \sum_{k=1}^{N} \frac{\langle X, V_k \rangle}{\langle V_k, V_k \rangle} \Phi(V_k).$$

If we define $\mathscr{W}_k(X) := \frac{\langle X, V_k \rangle}{\langle V_k, V_k \rangle}$ and $E_k := \Phi(V_k)$, then $\Pi$ is of the form (A.26). More precisely, for our structure in the WEP, (A.33), we have

$$E_k := (K - \bar{k}\mathbf{1}\mathbf{1}^T) \circ V_k - P_-(\sigma)^{-1} V_k E - P_+(\sigma)^{-1} V_k J E J. \tag{A.34}$$

As can be expected from a Galerkin approach, the approximation is exact for any $X \in \mathcal{V}$, since by construction $\Phi(V_k) = \Pi(V_k)$, $k = 1, \ldots, N$.

In theory, the construction can be done for any appropriate vector space. For reasons of structure exploitation, we select $V_k$, $k = 1, \ldots, N$, as indicator functions in rectangular regions, as shown in Figure A.2. We then select $N_x$ and $N_z$ intervals in $x$- and $z$-direction respectively, hence $N = N_x N_z$. In this case the matrices $V_{p+(q-1)N_z}$, $p = 1, \ldots, N_z$ and $q = 1, \ldots, N_x$, take the value 1 in the corresponding rectangular region and zero outside, and $\mathscr{W}_{p+(q-1)N_z}$ is the functional taking the mean over that region. More precisely,

$$\begin{aligned} [V_{p+(q-1)N_z}]_{r,s} &= \begin{cases} 1 & \text{if the point } (r,s) \text{ belongs to the region } (p,q) \\ 0 & \text{otherwise} \end{cases} \\ \mathscr{W}_{p+(q-1)N_z}(X) &= \frac{\sum_{r,s} [X \circ V_{p+(q-1)N_z}]_{r,s}}{\sum_{r,s} [V_{p+(q-1)N_z}]_{r,s}}. \end{aligned} \tag{A.35}$$

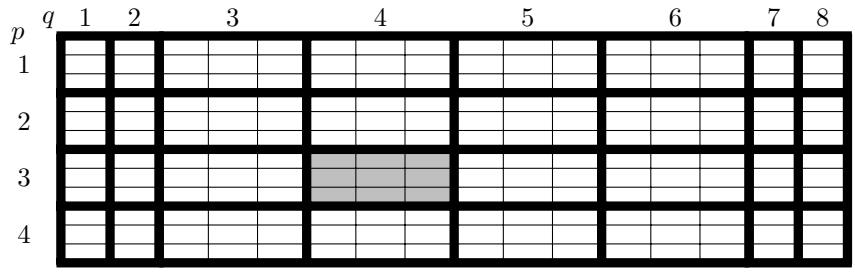



Figure A.2: The fine grid in the figure is an illustration of the grid on which we discretize the PDE, and the coarse grid shows the regions in which we take basis vectors $V_k$ as constant. In this example the weight matrix $V_{3+(4-1)N_z}$ will have value 1 for $(k, \ell)$ in the light gray area and 0 on all other elements. Note that the coarse grid for the SMW approximation is finer towards the boundaries in $x$-direction. This is done in order to better capture the effect of the DtN-maps.

Note that the grid resolution in Figure A.2 is finer near the boundary. This is done in order to improve the approximation of the DtN-maps, which are localized in the boundary region. The localization can be seen in the structure of the $E$-matrix in Proposition A.3.1, specifically by noting that the two boundary terms in (A.23) only depend on the first and last two columns of $X$. More precisely, $E$ stems from a one-sided, second-order, finite-difference approximation of the derivative in (A.1d) at $x_-$, i.e.,

$$u_x(x_0, z_\ell) \approx d_0 u(x_0, z_\ell) + d_1 u(x_1, z_\ell) + d_2 u(x_2, z_\ell), \tag{A.36}$$

for $\ell = 1, \ldots, n_z$. However, the approximation generated by the Galerkin approach with basis as in (A.35) and a uniform grid (without grid refinement at the boundary) would result in the approximation

$$u_x(x_0, z_\ell) \approx d_0 u(x_0, z_\ell) + \frac{(d_1 + d_2)}{\mathcal{N}_x} \sum_{k=1}^{\mathcal{N}_x} u(x_k, z_\ell), \tag{A.37}$$

where $\mathcal{N}_x$ is the number of discretization points, in $x$-direction, contained in the region at the boundary ($\mathcal{N}_x = n_x/N_x$ for a uniform grid). Equation (A.37) is not an accurate derivative approximation, and the grid refinement is added to capture the approximation (A.36). Note that in both cases, (A.36) and (A.37), the Galerkin approach with basis (A.35) would average the approximations in $z$-direction. The reasoning is analogous for (A.1e) at $x_+$. Hence, the boundary refinement as shown in Figure A.2 is not primarily to improve the approximation of $X$ with $\tilde{X} \in \mathcal{V}$, but rather to improve the approximation properties of the operator. The choice of grid refinement is supported by computational experiments presented in Section A.6.

**Remark A.4.3** (Other approximations)**.** *Note that the approximation described above is only an illustration of an approximation procedure and there exist many variations. Our construction can be interpreted as a low-rank approximation with operator composition $\Pi = \Phi\mathcal{P}$, where $\mathcal{P}$ is a linear low-rank operator. We use a Galerkin type approach where $\mathcal{P}$ is a projection onto the space $\mathcal{V}$. Although we have used a specific Galerkin space $\mathcal{V}$ and basis functions (with exploitable structure for this problem) others can also be used. Another option is to use a truncated SVD-based approach, e.g., similar to the rank-revealing procedure proposed in [10, Algorithm 3.2]. Such an approach would aim to give a good approximation of the operator for all $X \in \mathbb{C}^{n_z \times n_x}$. In contrast to this our approach is designed for a specific class of $X$, namely those which are smooth in the sense that they can be approximated by the discretization of a piecewise constant function, i.e., $X \approx \tilde{X} \in \mathcal{V}$. Further options are to use smoothing as, e.g., in multi-grid methods and other domain decomposition methods [12]. In order to use these approaches in the framework here described, further focused research would be required. In our setting, the structure of* (A.34) *allows us to reduce the memory requirements as we describe in the next section.*

## A.5 Structure exploitation and specialization of Resinv

### A.5.1 SMW-preconditioned Resinv

The application of Resinv to the WEP, described in Section A.2.3, requires an efficient solution to the linear system $M(\sigma)^{-1}r$ in equation (A.16). Since we want to solve this linear system iteratively, we need an effective preconditioner. We use the approximation technique presented in Section A.4.2 as a preconditioner. As a consequence of the fact that the shift is kept constant in Resinv, the Sylvester operator defined in (A.22), with $A := D_{zz} + 2\sigma D_z + \sigma^2 I_{n_z} + \bar{k} I_{n_z}$ and $B := D_{xx}$, is also constant. Therefore the matrices $F_1, \ldots, F_N$ in (A.28b) and the $W$-matrix in (A.29) are constant throughout the whole procedure. Hence, the $W$-matrix can be precomputed before initiating Resinv. Moreover, as mentioned in Remark A.4.2, this formulation of the SMW-formula, does not require the storage of the matrices $F_1, \ldots, F_N$, once $W$ has been computed. In fact only one $F$-matrix needs to be stored at a time, since the columns of $W$ can be computed column-wise. The precomputation can also be trivially parallelized since the columns of $W$ are independent. This construction is summarized in Algorithm A.2.

An important feature of the algorithm is that $N$ can be treated as a parameter. Using a large $N$ implies more computational work in the precomputation phase, i.e., Steps 2–5 of Algorithm A.2, since many Sylvester equations need to be solved. However, the quality of the preconditioner is better and we expect the iterative method to convergence in fewer iterations for large $N$. More precisely, less computation is required for the iterative solves[1] in Step 10. Hence, $N$ parameterizes a trade-off between computation time in the initialization and in the iterative solves. As is illustrated in Section A.6, the best choice of $N$ in terms of total computation time is a nontrivial problem, although many choices of $N$ lead to a competitive algorithm.

[1] This holds only if $N \ll n_x n_z$ as computation of (A.31) otherwise is a substantial part of Step 10.

**Algorithm A.2:** Resinv for WEP with preconditioned Schur reformulation

**input :** Initial guess of the eigenpair $(\gamma_0, v_0) \in \mathbb{C} \times \mathbb{C}^{n_x n_z + 2n_z}$, with $\|v_0\| = 1$
**output:** An approximation $(\gamma, v) \in \mathbb{C} \times \mathbb{C}^{n_x n_z + 2n_z}$ of $(\gamma_*, v_*) \in \mathbb{C} \times \mathbb{C}^{n_x n_z + 2n_z}$

$\sigma \leftarrow \gamma_0$
**for** $k = 1, 2, \ldots, N$ **do**
  Compute $F_k$ from (A.28b) with $E_k$ from (A.34)
  Compute the $k$th column of $W$ as described in (A.29)
LU-factorize $W$
**for** $k = 0, 1, 2, \ldots$ **do**
  Compute new approximation of $\gamma_{k+1}$ from (A.14)
  Compute the residual $r_k$ from (A.15)
  Compute $\tilde{r}_k$ from $r_k$ with (A.20), using the sparsity of $C_1$ and the structure of $P(\sigma)^{-1}$ according to Remark A.2.1
  Compute $q$ from the linear system (A.19) with a preconditioned iterative solver, where the preconditioner is applied to a vector $c$ as:
    - Set $C$ such that $\mathrm{vec}(C) = c$ and compute $G$ from (A.28a). Use $G$ to compute $g$ from (A.29)
    - Compute $\{\alpha_k\}_{k=1}^N$ by solving the linear system (A.31) with the pre-factorized matrix $W$
    - Form the right-hand side of (A.30) and solve the Sylvester equation.Vectorize the solution matrix $X$
  Compute the correction $\Delta v_k$ from $q$ using (A.18)
  $v_{k+1} \leftarrow (v_k - \Delta v_k)/\|v_k - \Delta v_k\|$
$\gamma \leftarrow \gamma_k, \quad v \leftarrow v_k$

In order to further improve performance we use a result regarding residual inverse iteration in [37]. More precisely, [37, Theorem 9] states that the linear solves in Resinv can be terminated in a way that preserves the property that the convergence factor is proportional to the shift–eigenvalue distance. It is proposed to use the tolerance $\tau$ satisfying

$$\|M(\gamma^{k+1})v_k - M(\sigma)\Delta v_k\| \leq \tau \|M(\gamma^{k+1})v_k\| \tag{A.38}$$

and $\tau = \mathcal{O}(|\gamma_* - \sigma|)$. Although we solve the linear system (A.19) inexactly, the error propagates linearly to (A.18), and hence the tolerance (A.38) is natural also in our setting.

### A.5.2 Storage improvements

The particular choice of SMW-formulation is due to an observation in computational experiments, that memory is a restricting aspect in our approach. We have therefore selected the SMW-formulation in order to reduce memory requirement at the cost of an increased

computation time. Our approach requires the computation of the solution to two Sylvester equations per iteration, i.e, equation (A.28a) and (A.30). The solution $X$ in (A.30) could be computed by forming a linear combination of $G, F_1, \ldots, F_N$, as in (A.32) but would require the storage of $F_1, \ldots, F_N$ which are full matrices in general. In contrast to this, the matrices $E_1, \ldots, E_N$ have a structure that can be exploited. More precisely, the matrix $C - \sum_{k=1}^{N} \alpha_k E_k$ required in (A.30) can be computed efficiently by exploiting the structure of $V_k$ and $E_k$ defined in (A.34), significantly reducing memory requirements.

### A.5.3 Circulant structure exploitation for Sylvester equation

In order to use the suggested preconditioner we need to solve a number of Sylvester equations, with the Sylvester operator defined by (A.22). There are many methods available in the literature, both direct methods such as the Bartels–Stewart algorithm [4] as well as iterative methods. See [34] for a recent survey of available methods. However, our Sylvester equation has a particular structure which can be exploited further. The approach is based on the implicit diagonalization of the coefficient matrices, from which a closed form expression is available. Consider the equation $AX + XB = C$ where $A$ and $B$ are diagonalizable. Then the solution $X$ is given by

$$X = VYW^{-1}, \quad \text{with} \quad [Y]_{p,q} = \frac{[V^{-1}CW]_{p,q}}{[\Lambda_A]_p + [\Lambda_B]_q}, \tag{A.39}$$

where $A = V\Lambda_A V^{-1}$, and $B = W\Lambda_B W^{-1}$. In the general case, the application of (A.39) is expected to be expensive and numerically unstable. For the waveguide matrix equation (A.23) the matrix $A$ is circulant, as it stems from the discretization with periodic boundary conditions. In particular it is diagonalized by the Fourier matrix whose action can be computed by the FFT; the eigenvalues are also readily available in $\mathcal{O}(n_z \log(n_z))$ operations using the FFT [14, Theorem 4.8.2]. The other matrix $B = D_{xx}$ is well studied and has both known eigenvalues and eigenvectors, the action of the latter can be computed in an efficient and stable way using the relation between Sine-/Cosine-transforms and the FFT [11, Lemma 6.1] [14, Section 4.8]. The solution to the Sylvester operator in (A.22), i.e.,

$$(D_{zz} + 2\sigma D_z + (\sigma^2 + \bar{k})I_{n_z})X + XD_{xx} = C, \tag{A.40}$$

can hence be computed by using (A.39) since the action of $V$, $V^{-1}$, $W$, and $W^{-1}$ can be computed efficiently and accurately using the FFT, and the diagonals of $\Lambda_A$ and $\Lambda_B$ are available. This exploitation of the FFT leads to a computation complexity that is $\mathcal{O}(n_x n_z \log(n_x n_z))$ for the inversion of (A.40), cf. [11, Section 6.7].

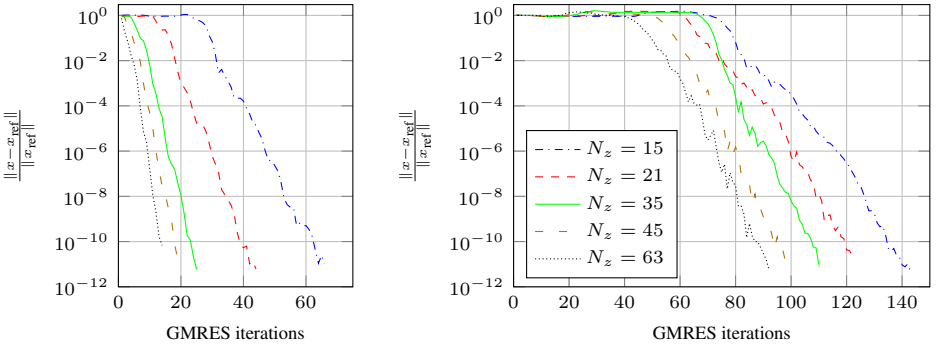
## A.6 Numerical simulations

In order to illustrate properties of our approach, we now show the result of simulations carried out in MATLAB on a desktop computer.[2] Source code for the simulations are pro-

[2] Intel quad-core i5-5250U CPU 1.60 GHz × 4, with 16 GB RAM using MATLAB 2015a.

vided online to improve reproducibility.[3] We use the waveguide illustrated in Figure A.1a, and also described in [18, Section 5.2]. This waveguide has many eigenvalues, oscillatory eigenfunctions, and a large discontinuity in the wavenumber, which is constructed to be representative of a realistic situation. The free parameter in (A.23) is set to $\bar{k} = \text{mean}(K)$. Moreover, the size of the problem is denoted by $n$ and defined as $n := n_x n_z + 2n_z$, and the parametrization of the preconditioner, $N$, is defined by $N := N_x N_z$. For implementation convenience we select $n_x = n_z + 4$ and $N_x = N_z + 4$.

We first illustrate the quality of the preconditioner (without incorporation into Resinv). The relative error as function of GMRES-iteration is visualized in Figure A.3. We clearly see that the required number of iterations decreases with $N$, which is expected since the SMW-approximation error is smaller for larger $N$. Moreover, we see that a small $N$ normally generates a long transient phase. The advantage of selecting a finer grid close to the boundary, as shown in Figure A.2, is clear from the fact that the convergence in Figure A.3a is faster than the convergence in Figure A.3b. In simulations with additional layers of grid refinement close to the boundary ($N_x = N_z + 8$, and $n_x = n_z + 8$) no substantial increase in convergence speed compared to Figure A.3a was observed.




(a) Basis matrices $V_k$ chosen as in Figure A.2, where $N_x = N_z + 4$.

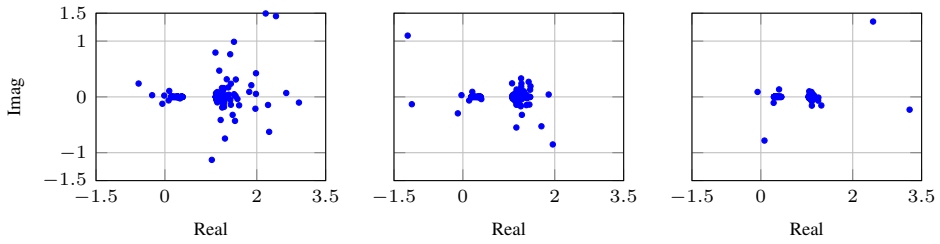
(b) Basis matrices $V_k$ chosen in a uniform way (cf. Figure A.2). Note that here $n_x = n_z$, and $N_x = N_z$.

Figure A.3: GMRES convergence for solving $S(\sigma)x = c$ for different coarse grids in the SMW-approximation. The discretization is $n_z = 945$ and the error is measured as the relative error compared to a reference solution $x_{\text{ref}}$, computed with the same methods but to a higher accuracy.

Although no theoretical bounds on the eigenvalues are derived for the preconditioned system, we provide numerical simulations of the eigenvalues of the preconditioned system in Figure A.4. Since, due to the dimension of the problem, it is infeasible to compute all eigenvalues, and since clustering is observed for medium size problems, we compute the 250 eigenvalues of largest magnitude of the preconditioned system shifted with $-0.75I$.

[3] URL: `https://www.math.kth.se/~eringh/software/wep/wep_code`

The computation is done with `eigs`. As seen in Figure A.4, the eigenvalues exhibit a higher degree of clustering when $N_z$ is increased, which indicates faster convergence for iterative methods.




(a) $N_z = 15$. (b) $N_z = 21$. (c) $N_z = 35$.

Figure A.4: Eigenvalues of the preconditioned systems for different parametrization $N_z$ of the coarse grids in the SMW-approximation. The dots in each plot are the 250 eigenvalues, representing the outer part of the spectrum. The discretization is $n_z = 945$, and the coarse grid is chosen as in Figure A.2. Compare with convergence observed in Figure A.3a.

Algorithm A.2 is applied to this benchmark problem, and Figure A.5 shows the number of required GMRES-iterations for one iteration of Resinv. As expected from the approximation properties of the preconditioner, a larger value $N$ implies fewer iterations. We observe an increase in the number of required GMRES-iterations with increasing problem size. The increase is however rather slow.

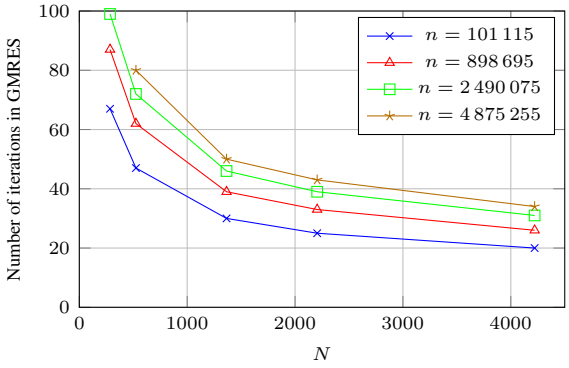
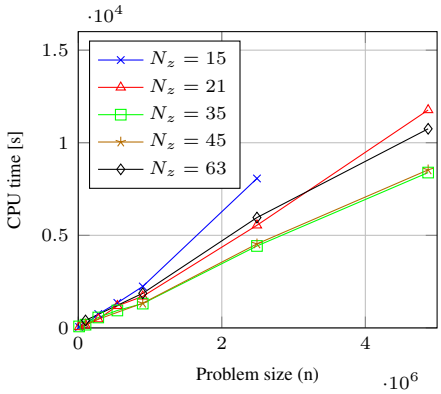


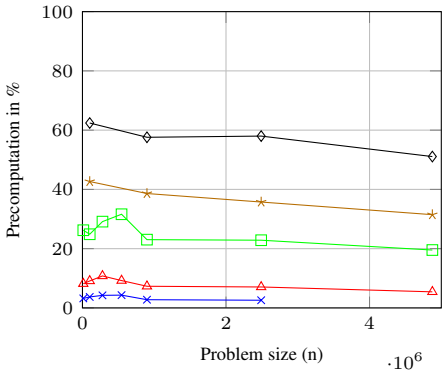

Figure A.5: Illustration of the increase in GMRES iterations. Evaluations for parameter values $N_z \in \{15, 21, 35, 45, 63\}$, where $N_x = N_z + 4$, and $N = N_x N_z$.

The trade-off between computation time in the initiation and in the linear solves is illustrated for different values of $N$ in Figure A.6a. For this particular problem (and computing

environment) the best choice is $N_z = 35$. Note however, that several other choices such as $N_z = 45$ and $N_z = 63$ are almost as good. In the profiling illustration in Figure A.6b, we see, as expected, that increasing $N$, shifts computational effort into to the precalculation phase, and that the computational effort required for the precalculation and the other parts of the algorithm are of the same order of magnitude.




(a) Time in seconds for solving the problem.



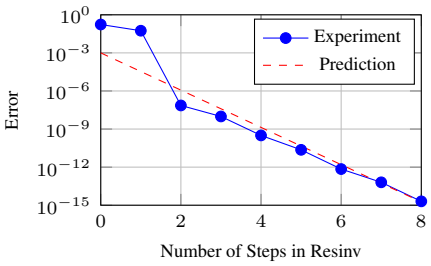

(b) Percentage of time spent on precalculating the (SMW) $W$-matrix from (A.29).

Figure A.6: Time for the complete method described in Algorithm A.2, as a function of problem size. Problem size is the total size of the problem as presented in (A.2), that is $n_x n_z + 2n_z$, where the GMRES-tolerance is select such that linear system is solved to full precision. This is plotted for different $N$ values $N = N_x N_z$, and $N_x = N_z + 4$.

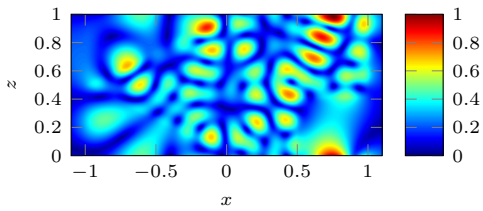
For the WEP we measure the error of the approximation with an estimate of the relative residual norm

$$R(v,\gamma) = \frac{\|M(\gamma)v\|_2}{\sum_{k=0}^{2} |\gamma|^k \|A_k\|_1 + \|C_1\|_1 + \|C_2^T\|_1 + 2|d_0| + \sum_{k=-p}^{p} (|s_{+,k}(\gamma)| + |s_{-,k}(\gamma)|)} \tag{A.41}$$

analogous to estimates for other NEPs [23, 16]. Algorithm A.2 is applied with the GMRES-termination criterion (A.38) set to $\tau = 10^{-3}$, and the error is visualized in Figure A.7a. We use $\sigma = -0.5 - 0.4i$, such that the algorithm converges to the eigenvalue $\gamma \approx -0.523 - 0.375i$. As expected from the GMRES-termination criterion (A.38), we maintain linear convergence and a convergence factor which is in the order of magnitude of the shift–eigenvalue distance. The corresponding computed eigenfunction is visualized in Figure A.7b. The computation time for different parameter values are given in Table A.1. A comparison between Table A.1 and Figure A.6 shows that this affects the best choice of $N_z$, since the termination criterion decreases the cost for the linear solves. The best performance in terms of CPU-time is for these examples achieved when the precomputation time is about 30% of the total computation time.




(a) Error in relative residual norm (A.41). The prediction corresponds to convergence factor $|\gamma - \sigma|$, i.e., a constant times $|\gamma - \sigma|^k$, where $k$ is the number of steps in Resinv.




(b) Absolute value of the eigenfunction.

Figure A.7: Illustration of convergence of Resinv and the corresponding eigenfunction to eigenvalue $\gamma \approx -0.523 - 0.375i$. In this simulation we use the shift $\sigma = -0.5 - 0.4i$, the discretization $n_z = 2835$, and the preconditioner parameter $N_z = 21$.

For very large problems, GMRES is not advantageous due to memory requirements. Dealing with this using restarted GMRES was not successful due to long transient phases, similar to those observed in Figure A.3. GMRES requires more memory than BiCGStab, and the largest problem we manage to solve is computed with BiCGStab. However, GMRES was in general slightly faster in simulations with enough memory. For large problems with GMRES, the size of the Krylov space needs to be carefully adjusted to stay within the available RAM. For most simulations the maximum size of the Krylov space is 100 vectors, but for $n \approx 16 \cdot 10^6$ only 30 vectors is used for $N_z = 35$ and $N_z = 21$. However, for the case $N_z = 15$ no size of the Krylov subspace is found that is sufficiently large to converge to the tolerance, but small enough to stay within the available RAM. We compare our approach with GMRES combined with a preconditioner based on incomplete LU-factorization (ILU) [31, Chapter 10]. Unfortunately, in our experiments ILU required considerable memory resources and we were not able to use it to solve very large problems.

**Remark A.6.1** (Recycling BiCGStab)**.** *We also test the recycling BiCGStab as described in [1, Algorithm 2]. The invariant subspace is computed as the dominant 15 right eigenvectors of the shifted preconditioned system.*[4] *For a problem with $n_z = 945$, we observe that the number of iterations needed for convergence is 50, 28, and 16, for recycling BiCGStab; and 79, 44, and 24 for BiCGStab; where $N_z$ is 15, 21, and 35, respectively. Consequently, recycling reduces the number of iterations needed. However, a direct usage of recycling requires more memory, and for our class of large-scale problems, we have observed that memory consumption is a limiting factor. Hence, using recycling BiCGStab partially defeats the purpose of changing from GMRES to BiCGStab, as in the discussion above.*

[4] The computation is done with `eigs`. The shift is $-0.75I$ (cf. Figure A.4). The bi-orthogonality is enforced in the same way as described in [2, Section 4.3].

| | GMRES | | | BiCGStab | | | ILU (GMRES) | |
|---|---|---|---|---|---|---|---|---|
| $n$ | $N_z = 15$ | $N_z = 21$ | $N_z = 35$ | $N_z = 15$ | $N_z = 21$ | $N_z = 35$ | $\epsilon = 10^{-5}$ | $\epsilon = 10^{-6}$ |
| 101 115 | 63 | 53 | 79 | 85 | 66 | 91 | 78 | 74 |
| 278 775 | 201 | 169 | 241 | 324 | 216 | 246 | 504 | 474 |
| 898 695 | 542 | 405 | 509 | 757 | 520 | 570 | – | – |
| 2 490 075 | 1 674 | 1 129 | 1 652 | 2 391 | 1 668 | 1 796 | – | – |
| 4 875 255 | 3 530 | 2 435 | 3 218 | 4 653 | 3 199 | 3 276 | – | – |
| 8 054 235 | 6 732 | 4 675 | 6 658 | 10 808 | 7 001 | 7 528 | – | – |
| 16 793 595 | – | 11 171 | 17 116 | 23 047 | 14 865 | 15 983 | – | – |

Table A.1: Illustration of CPU-time in seconds for a set of different methods, for different values of the parameter $N_z$ where $N = N_x N_z$ and $N_x = N_z + 4$. The ILU-dropping tolerance is denoted $\epsilon$. The dash, i.e., $-$, denotes simulations which could not be executed due to insufficient RAM.

**Remark A.6.2** (Applicability to FEM-discretization)**.** *A FEM-discretization of this problem was presented in [18]. Our preconditioner can also be applied to solve the discretization with FEM, by using the FD-preconditioner. The observed convergence is similar to the previously observed convergence in the FD-case (cf. Figure A.3a). For instance, when our method is applied to a discretization with* $n_z = 945$ *the number of iterations needed for convergence in GMRES is* 67*,* 44*, and* 25 *iterations for the FD-problem; and similarly* 67*,* 44*, and* 25 *for the FEM-problem; where* $N_z$ *is* 15*,* 21*, and* 35 *respectively.*

## A.7 Concluding remarks and outlook

We have presented a new computational procedure specialized for the WEP (A.1), based on combining the method for NEPs called Resinv and an iterative method for linear systems. The preconditioner for the iterative method is based on an approximation leading to a structure which can be exploited with a matrix-equation version of SMW.

There are many options for constructing the SMW approximation. For instance, the space used in the Galerkin approximation could be selected in a number of ways. Such a construction would necessarily need to use sparsity or other matrix structures in order to solve large-scale problems.

We have focused on one particular method for NEPs: Resinv. One of the crucial features is that a linear system corresponding to $M(\sigma)$ needs to be solved many times (for a constant shift). The Resinv method is not the only method that uses the solution to many linear systems with a fixed shift. This is also the case for the nonlinear Arnoldi method [40] and the tensor infinite Arnoldi method [18]. However, inexact solves in Arnoldi-type methods are sometimes problematic [35], and further research would be required in order to reliably and efficiently use our preconditioner for these methods.

Although our approximation is justified with a Galerkin approach, we have not provided any theoretical convergence analysis. The application of standard proof-techniques for such an analysis, e.g., involving eigenvalues and spectral condition numbers have not led to a clear characterization of the error. Therefore, we believe that a convergence anal-

ysis would require use of the regularity of the eigenfunction, similar to what is used in multi-grid methods [12], which is certainly beyond the scope of this paper.

Finally, we wish to point out that several results in this paper may be of interest also for other problems and other NEPs, e.g., [13]. As mentioned in the introduction, the matrix-equation approach has been used for PDEs on a rectangular domains that are discretized on uniform grids [26, 36, 9, 28]. These techniques lead to problems on the form $\mathscr{L}(X) + \Phi(X) = C$, in which case the SMW-Galerkin approach presented could be natural to try as a preconditioner to the corresponding linear system. Furthermore, many NEPs arise naturally from PDEs with artificial boundary conditions. Most artificial boundary conditions has freedom regarding selection of boundary. We can therefore select a rectangular domain, i.e., similar to the framework considered in this paper.

## Acknowledgment

The authors wish to thank Per Enqvist (KTH) and Tobias Damm (TU Kaiserslautern) for comments and discussions in the early developments of this result. This research is supported by the Swedish Research Council under Grant No. 621-2013-4640.

## References

[1] K. Ahuja, P. Benner, E. de Sturler, and L. Feng. Recycling BiCGSTAB with an application to parametric model order reduction. *SIAM J. Sci. Comput.*, 37(5):S429–S446, 2015.

[2] K. Ahuja, E. de Sturler, S. Gugercin, and E. R. Chang. Recycling BiCG with an application to model reduction. *SIAM J. Sci. Comput.*, 34(4):A1925–A1949, 2012.

[3] J. Asakura, T. Sakurai, H. Tadano, T. Ikegami, and K. Kimura. A numerical method for nonlinear eigenvalue problems using contour integrals. *JSIAM Letters*, 1:52–55, 2009.

[4] R. H. Bartels and G. W. Stewart. Algorithm 432: Solution of the matrix equation $AX + XB = C$. *Comm. ACM*, 15:820–826, 1972.

[5] 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.

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

[7] 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):1–28, 2013.

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

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

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

[11] J. Demmel. *Applied numerical linear algebra*. SIAM publications, Philadelphia, PA, 1997.

[12] V. Dolean, P. Jolivet, and F. Nataf. *An Introduction to Domain Decomposition Methods*. SIAM publications, Philadelphia, PA, 2015.

[13] 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.

[14] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins Univ. Press, Baltimore, MD, 4th edition, 2013.

[15] 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.

[16] N. J. Higham, R. Li, and F. Tisseur. Backward error of polynomial eigenproblems solved by linearization. *SIAM J. Matrix Anal. Appl.*, 29(4):1218–1241, 2008.

[17] R. A. Horn and C. .R Johnson. *Topics in Matrix Analysis*. Cambridge Univ. Press, Cambridge, UK, 1st edition, 1994.

[18] E. Jarlebring, G. Mele, and O. Runborg. The waveguide eigenvalue problem and the tensor infinite Arnoldi method. *SIAM J. Sci. Comput.*, 39(3):A1062–A1088, 2017.

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

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

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

[22] I. Kuzmanović and N. Truhar. Sherman–Morrison–Woodbury formula for Sylvester and T-Sylvester equations with applications. *Int. J. Comput. Math.*, 90(2):306–324, 2013.

[23] B.-S. Liao, Z. Bai, L.-Q. Lee, and K. Ko. Solving large scale nonlinear eigenvalue problems in next-generation accelerator design. Technical Report SLAC-PUB-12137, Stanford University, 2006.

[24] V. Mehrmann and H. Voss. Nonlinear eigenvalue problems: A challenge for modern eigenvalue methods. *GAMM-Mitt.*, 27:121–152, 2004.

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

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

[27] M. L. Parks, E. de Sturler, G. Mackey, D. D. Johnson, and S. Maiti. Recycling Krylov subspaces for sequences of linear systems. *SIAM J. Sci. Comput.*, 28(5):1651–1674, 2006.

[28] 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.

[29] S. Richter, L. D. Davis, and E. G. Collins Jr. Efficient computation of the solutions to modified Lyapunov equations. *SIAM J. Matrix Anal. Appl.*, 14(2):420–431, 1993.

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

[31] Y. Saad. *Iterative methods for sparse linear systems*. SIAM publications, Philadelphia, PA, 2nd edition, 2003.

[32] Y. Saad and M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 7:856–869, 1986.

[33] K. Schreiber. *Nonlinear Eigenvalue Problems: Newton-type Methods and Nonlinear Rayleigh Functionals*. PhD thesis, TU Berlin, 2008.

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

[35] V. Simoncini and D. B. Szyld. Theory of inexact Krylov subspace methods and applications to scientific computing. *SIAM J. Sci. Comput.*, 25(2):454–477, 2003.

[36] M. Stoll and T. Breiten. A low-rank in time approach to PDE-constrained optimization. *SIAM J. Sci. Comput.*, 37(1):B1–B29, 2015.

[37] 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, 2012.

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

[39] H. A. van der Vorst. BI-CGSTAB: A fast and smoothly converging variant of BI-CG for the solution of nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 13(2):631–644, 1992.

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

[41] H. Voss. Nonlinear eigenvalue problems. In L. Hogben, editor, *Handbook of Linear Algebra*, number 164 in Discrete Mathematics and Its Applications. CRC Press, Boca Raton, FL, 2nd edition, 2014.

[42] J. Xiao, C. Zhang, T.-M. Huang, and T. Sakurai. Solving large-scale nonlinear eigenvalue problems by rational interpolation and resolvent sampling based Rayleigh-Ritz method. *Internat. J. Numer. Methods Engrg.*, 2016.

# Paper B

## Krylov methods for low-rank commuting generalized Sylvester equations

# Krylov methods for low-rank commuting generalized Sylvester equations

by

Elias Jarlebring, Giampaolo Mele, Davide Palitta, Emil Ringh

published in *Numerical Linear Algebra with Applications*
Volume 25, Issue 6, Pages e2176, December 2018

**Abstract**

We consider generalizations of the Sylvester matrix equation, consisting of the sum of a Sylvester operator and a linear operator $\Pi$ with a particular structure. More precisely, the commutator of the matrix coefficients of the operator $\Pi$ and the Sylvester operator coefficients are assumed to be matrices with low rank. We show (under certain additional conditions) low-rank approximability of this problem, i.e., the solution to this matrix equation can be approximated with a low-rank matrix. Projection methods have successfully been used to solve other matrix equations with low-rank approximability. We propose a new projection method for this class of matrix equations. The choice of subspace is a crucial ingredient for any projection method for matrix equations. Our method is based on an adaption and extension of the extended Krylov subspace method for Sylvester equations. A constructive choice of the starting vector/block is derived from the low-rank commutators. We illustrate the effectiveness of our method by solving large-scale matrix equations arising from applications in control theory and the discretization of PDEs. The advantages of our approach in comparison to other methods are also illustrated.

**Keywords:** Generalized Sylvester equation, low-rank commutation, extended Krylov subspace, iterative solvers, matrix equation.

## B.1 Introduction

Let $\mathscr{L} : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$ denote the *Sylvester operator* associated with the matrices $A, B \in \mathbb{R}^{n\times n}$, i.e.,

$$\mathscr{L}(X) := AX + XB^T, \tag{B.1}$$

and let $\Pi : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$ denote the matrix operator defined by

$$\Pi(X) := \sum_{i=1}^{m} N_i X M_i^T, \tag{B.2}$$

where $m \ll n$. The matrices $A$ and $B$ are assumed to be large, sparse, and nonsingular, the operator $\mathscr{L}$ is assumed to be invertible, i.e., the spectra of $A$ and $-B$ are disjoint [40, Section 7.2]. Given $C_1, C_2 \in \mathbb{R}^{n\times r}$ with $r \ll n$, our paper concerns the problem of computing $X \in \mathbb{R}^{n\times n}$ such that

$$\mathscr{L}(X) + \Pi(X) = C_1 C_2^T. \tag{B.3}$$

This equation is sometimes (e.g. [9]) referred to as the *generalized Sylvester equation*.

Let $\mathrm{com}(A, B) := AB - BA$ denote the *commutator* of two matrices. The structure of the operator $\Pi$ is assumed to be such that the commutator of the Sylvester coefficients and the coefficients defining the operator $\Pi$ have low rank. In other words, 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

$$\mathrm{com}(A, N_i) = AN_i - N_i A = U_i \tilde{U}_i^T, \tag{B.4a}$$
$$\mathrm{com}(B, M_i) = BM_i - M_i B = Q_i \tilde{Q}_i^T, \tag{B.4b}$$

for $i = 1, \ldots, m$. The property (B.4), which we refer to as *low-rank commutation*, is in this framework a generalization of the concept of commuting matrices. The case of pure commutation, i.e., when the right-hand side of (B.4) is zero, which occurs for instance when $N_i = f_i(A), M_i = g_i(B)$ where $f_i, g_i$ are polynomials or analytic functions, is analyzed in [30] and [8].

A recent successful method class for matrix equations defined by large and sparse matrices, are based on projection, typically called *projection methods* [39, 17, 8]. We propose a new projection method for (B.3) under the low-rank commutation assumption (B.4).

Projection methods are typically derived from an assumption on the decay of the singular values of the solution. More precisely, a necessary condition for the successful application of a projection method is low-rank approximability, i.e., the solution can be approximated by a low-rank matrix. We characterize the low-rank approximability of the solution to (B.3) under the condition that the Sylvester operator $\mathscr{L}$ has a low-rank approximability property and that $\rho(\mathscr{L}^{-1}\,\Pi) < 1$. The low-rank approximability theory is presented in Section B.2. The function $\rho(\cdot)$ denotes the (operator) spectral radius, i.e., $\rho(\mathscr{L}) := \sup\{|\lambda| \,|\, \lambda \in \sigma(\mathscr{L})\}$, where $\sigma(\cdot)$ is the set of eigenvalues.

The choice of the subspace is an important ingredient in any projection method. We propose a particular choice of projection spaces by identifying certain features of the solution to (B.3) based on our characterization of low-rank approximability and the low-rank commutation properties (B.4). More precisely we use an extended Krylov subspace with an appropriate choice of the starting block. We present and analyze an expansion of the framework of the extended Krylov subspace method for Sylvester equation (K-PIK) [39, 15] to the generalized Sylvester equation (Section B.3).

Linear matrix equations of the form (B.3) arise in different applications. For example, the *generalized Lyapunov equation*, which corresponds to the special case where $B = A$, $M_i = N_i$ and $C_1 = C_2$, arises in model order reduction of bilinear and stochastic systems, see e.g. [9, 16, 8] and references therein. Many problems arising from the discretization of PDEs can be formulated as generalized Sylvester equations [37, 35, 33]. Low-rank approximability for matrix equations has been investigated in different settings: for Sylvester equations [22, 1, 21], generalized Lyapunov equations with low-rank correction [8] and more in general for linear systems with tensor product structure [29, 21].

The so-called low-rank methods, which projection methods belong to, directly compute a low-rank approximation to the solution of (B.3). Many algorithms have been developed for the Sylvester equation: projection methods [39, 17], low-rank ADI [11, 10], sign function method [4, 5], Riemannian optimization methods [28, 42] and many more. See the thorough presentation in [40]. For large-scale generalized Sylvester equations, fewer numerical methods are available in the literature. Moreover, they are often designed only for solving the generalized Lyapunov equation although they may be adapted to solve the generalized Sylvester equation. In [8], the authors propose a bilinear ADI (BilADI) method which naturally extends the low-rank ADI algorithm for standard Lyapunov problems to generalized Lyapunov equations. A non-stationary iterative method is derived in [38], and in [27] a greedy low-rank technique is presented. In principle, it is always possible to consider the $n^2 \times n^2$ linear system which stems from equation (B.3) by Kronecker transformations. There are specific methods for solving linear systems with tensor product structure, see [27, 28, 2] and references therein. These problems can also be solved employing one of the many methods for linear systems presented in the literature. In particular, matrix-equation oriented versions of iterative methods for linear systems, together with preconditioning techniques, are present in literature. See, e.g., [8, Section 5], [14, 29, 31]. To our knowledge, the low-rank commutativity properties (B.4) have not been considered in the literature in the context of methods for matrix equations.

The paper is structured as follows: In Section B.2 we use a Neumann series (cf. [19, 30, 36, 44]) with hypothesis $\rho(\mathscr{L}^{-1}\,\Pi) < 1$ to characterize the low-rank approximability of the solution to (B.3). In Section B.3 we further characterize approximation properties of the solution to (B.3) by exploiting the low-rank commutation feature of the coefficients (B.4). We use this characterization in the derivation of an effective projection space. In Section B.3.4 we present an efficient procedure for solving small-scale generalized Sylvester equations (B.3). Numerical examples that illustrate the effectiveness of our strategy are reported in Section B.4. Our conclusions are given in Section B.5.

We use the following notation. The vectorization $\mathrm{vec}(A)$ is the vector obtained by stacking the columns of the matrix $A$ on top of one another. We denote by $\|\cdot\|_F$ the Frobenius norm, whereas $\|\cdot\|$ is any submultiplicative matrix norm. Moreover, $\|\mathscr{L}\| := \sup_{\|A\|=1} \|\mathscr{L}(A)\|$, for a generic linear and continuous operator $\mathscr{L} : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$. The identity and the zero matrices are respectively denoted by $I$ and $O$. We denote by $e_i$ the $i$th vector of the canonical basis of $\mathbb{R}^n$ while $\otimes$ corresponds to the Kronecker product. The matrix obtained by stacking the matrices $A_1, \ldots, A_n$ next to each other is denoted by $\big[A_1, \ldots, A_n\big]$. Lastly, $\mathrm{range}(A)$ is the vector space generated by the columns of the matrix $A$ and $\mathrm{span}(\mathcal{A})$ is the vector space generated by the vectors in the set $\mathcal{A}$.

## B.2 Representation and approximation of the solution

### B.2.1 Representation as Neumann series expansion

The following theorem gives sufficient conditions for the existence of a representation of the solution to a generalized Sylvester equation (B.3) as a convergent series. This will be needed for the low-rank approximability characterization in the following section, as well as in the derivation of a method for small generalized Sylvester equations (further described in Section B.3.4).

**Theorem B.2.1** (Solution as a Neumann series)**.** *Let $\mathscr{L}, \Pi : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$ be linear operators such that $\mathscr{L}$ is invertible, $\rho(\mathscr{L}^{-1}\,\Pi) < 1$ and let $C \in \mathbb{R}^{n\times n}$. The unique solution of the equation $\mathscr{L}(X) + \Pi(X) = C$ can be represented as*

$$X = \sum_{j=0}^{\infty} Y_j, \tag{B.5}$$

*where*

$$\begin{cases} Y_0 & := \ \mathscr{L}^{-1}(C)\,, \\ Y_{j+1} & := -\mathscr{L}^{-1}(\Pi(Y_j))\,, \qquad\qquad j \geq 0. \end{cases} \tag{B.6}$$

*Proof.* By using the invertibility of $\mathscr{L}$ we have $X = (I + \mathscr{L}^{-1}\,\Pi)^{-1}\,\mathscr{L}^{-1}(C)$ and with the assumption $\rho(\mathscr{L}^{-1}\,\Pi) < 1$ we can express the operator $(I + \mathscr{L}^{-1}\,\Pi)^{-1}$ as a convergent Neumann series (for operators as, e.g., in [26, Example 4.5]). In particular, we obtain

$$X = \sum_{j=0}^{\infty} (-1)^j \left(\mathscr{L}^{-1}\,\Pi\right)^j \mathscr{L}^{-1}(C)\,.$$

The relation (B.5) follows by defining $Y_j := (-1)^j \left(\mathscr{L}^{-1}\,\Pi\right)^j \mathscr{L}^{-1}(C)$. By induction it follows that the relations (B.6) are fulfilled. □

**Remark B.2.2.** *Theorem B.2.1 can be used to construct an approximation to the solution of $\mathscr{L}(X) + \Pi(X) = C$ by truncating the series* (B.5) *analogous to the general form in [26, (4.23)]. In particular, let*

$$X^{(\ell)} := \sum_{j=0}^{\ell} Y_j, \tag{B.7}$$

*where $Y_j$ are given by* (B.6)*. The truncation error can be bounded as follows:*

$$\|X - X^{(\ell)}\| \leq \|\mathscr{L}^{-1}(C)\| \frac{\rho(\mathscr{L}^{-1}\,\Pi)^{\ell+1}}{1 - \rho(\mathscr{L}^{-1}\,\Pi)}.$$

If $\mathscr{L}$ and $\Pi$ are respectively the operators (B.1) and (B.2) that define the generalized Sylvester equation (B.3), then the truncated Neumann series (B.7) can be efficiently computed for small scale problems. In particular, this approach can be used in the derivation of a numerical method for solving small scale generalized Sylvester equations as illustrated in Section B.3.4.

### B.2.2 Low-rank approximability

We now use the result in the previous section to show that the solution to (B.3) can be often approximated by a low-rank matrix. We base the reasoning on low-rank approximability properties of $\mathscr{L}$. Our result requires the explicit use of certain conditions on the spectrum of matrix coefficients of $\mathscr{L}$. Under these specific conditions, the solution to a Sylvester equation with low-rank right-hand side can be approximated by a low-rank matrix, see [40, Section 4.1]. In this sense, we can extend several results concerning the low-rank approximability of the solution to Sylvester equations to the case of generalized Sylvester equations under the assumption $\rho(\mathscr{L}^{-1}\,\Pi) < 1$. More precisely, the truncated Neumann series (B.7) is obtained by summing the solutions to the Sylvester equations (B.6). Note that, under the low-rank approximability assumption of $\mathscr{L}$, the right-hand sides of the Sylvester equations (B.6) are low-rank matrices since we assume that $C$ is a low-rank matrix and $m \ll n$. We formalize this argument and present a new characterization of the low-rank approximability of the solution to (B.3) by adapting one of the most commonly used low-rank approximability result for Sylvester equations [21].

We now briefly recall some results presented in [21], for our purposes. Suppose that the matrix coefficients representing $\mathscr{L}$ are such that $\sigma(A) \cup \sigma(B) \subset \mathbb{C}_-$. Let $M \in \mathbb{C}^{n\times n}$ be such that $\sigma(M) \subset \mathbb{C}_-$, then its inverse can be expressed as $M^{-1} = \int_0^\infty \exp(tM)dt$. The integral can be approximated with the following quadrature formula

$$M^{-1} = \int_0^\infty \exp(tM)dt \approx \sum_{j=-k}^{k} w_j \exp(t_j M), \tag{B.8}$$

where the weights $w_j$ and nodes $t_j$ are given in [21, Lemma 5]. More precisely, we have an explicit formula for the approximation error

$$\left\| \int_0^\infty \exp(tM)dt - \sum_{j=-k}^{k} w_j \exp(t_j M) \right\| \leq K e^{-\pi\sqrt{k}}, \tag{B.9}$$

where $K$ is a constant that only depends on the spectrum of $M$. The solution to the Sylvester equation $\mathscr{L}(X) = C$ can be explicitly expressed as $\mathrm{vec}(X) = (I \otimes A + B \otimes I)^{-1}\mathrm{vec}(C)$. The solution to this linear system can be approximated by using (B.8) and approximating the inverse of $I \otimes A + B \otimes I$. Let $\mathscr{L}_{\mathrm{k}}^{-1} : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$ be the linear operator such that $\mathscr{L}_{\mathrm{k}}^{-1}(C)$ corresponds to the approximation (B.8). More precisely, the

operator $\mathscr{L}_{\mathrm{k}}^{\text{-}1}$ satisfies

$$\mathrm{vec}(\mathscr{L}_{\mathrm{k}}^{\text{-}1}(C)) = \sum_{j=-k}^{k} w_j \Big( \exp(t_j B) \otimes \exp(t_j A) \Big) \mathrm{vec}(C).$$

By using the properties of the Kronecker product, it can be explicitly expressed as

$$\mathscr{L}_{\mathrm{k}}^{\text{-}1}(C) = \sum_{j=-k}^{k} w_j \exp(t_j A) C \exp(t_j B^T). \tag{B.10}$$

In terms of operators, the error bound (B.9) is $\| \mathscr{L}^{\text{-}1} - \mathscr{L}_{\mathrm{k}}^{\text{-}1} \| \leq K e^{-\pi\sqrt{k}}$. The result of the above discussion is summarized in the following remark, which directly follows from (B.10) or [21, Lemma 7], [8, Lemma 2].

**Remark B.2.3.** *The solution to the Sylvester equation $\mathscr{L}(X) = C$ can be approximated by $\bar{X} = \mathscr{L}_{\mathrm{k}}^{\text{-}1}(C)$ where $\|X - \bar{X}\| \leq \|C\| K e^{-\pi\sqrt{k}}$, $\mathrm{rank}(\bar{X}) \leq (2k+1)r$, $K$ is a constant that depends on the spectrum of $\mathscr{L}$ and $r$ is the rank of $C$.*

The following theorem concerns the low-rank approximability of the solution to (B.3). More precisely, it provides a generalization of Remark B.2.3 to the case of generalized Sylvester equations by using the Neumann series characterization in Theorem B.2.1.

**Theorem B.2.4** (Low-rank approximability)**.** *Let $\mathscr{L}$ be the Sylvester operator* (B.1)*, $\Pi$ the linear operator* (B.2)*, $C_1, C_2 \in \mathbb{R}^{n \times r}$ and $k$ a positive integer. Let $X^{(\ell)}$ be the truncated Neumann series* (B.7)*. Then there exists a matrix $\bar{X}^{(\ell)}$ such that*

$$\mathrm{rank}(\bar{X}^{(\ell)}) \leq (2k+1)r + \sum_{j=1}^{\ell} (2k+1)^{j+1} m^j r, \tag{B.11}$$

*and*

$$\left\| X^{(\ell)} - \bar{X}^{(\ell)} \right\| \leq \bar{K} e^{-\pi\sqrt{k}}, \tag{B.12}$$

*where $\bar{K}$ is a constant that does not depend on $k$ and only depends on $\mathscr{L}$ and $\ell$.*

*Proof.* Let $\mathscr{L}_k$ be the operator (B.10) and consider the sequence

$$\begin{cases} \bar{Y}_0 & := \ \mathscr{L}_{\mathrm{k}}^{\text{-}1}(C_1 C_2^T), \\ \bar{Y}_{j+1} & := -\mathscr{L}_{\mathrm{k}}^{\text{-}1}(\Pi(\bar{Y}_j)), \quad j \geq 0. \end{cases} \tag{B.13}$$

Define $\beta := \| \mathscr{L}^{\text{-}1} \Pi \|$ and $\beta_k := \| \mathscr{L}_{\mathrm{k}}^{\text{-}1} \Pi \|$. By using Remark B.2.3 we have

$$\begin{aligned} \|Y_{j+1} - \bar{Y}_{j+1}\| &\leq \| \mathscr{L}^{\text{-}1}(\Pi(Y_j)) - \mathscr{L}^{\text{-}1}(\Pi(\bar{Y}_j)) \| + \| \mathscr{L}^{\text{-}1}(\Pi(\bar{Y}_j)) - \mathscr{L}_{\mathrm{k}}^{\text{-}1}(\Pi(\bar{Y}_j)) \| \\ &\leq \beta \|Y_j - \bar{Y}_j\| + K e^{-\pi\sqrt{k}} \|\Pi\| \|\bar{Y}_j\|. \end{aligned}$$

From the above expression, a simple recursive argument shows that

$$\|Y_{j+1} - \bar{Y}_{j+1}\| \leq \beta^{j+1}\|Y_0 - \bar{Y}_0\| + Ke^{-\pi\sqrt{k}}\|\Pi\| \sum_{t=0}^{j} \beta^{j-t}\|\bar{Y}_t\|. \tag{B.14}$$

Using the submultiplicativity of the operator norm, we see that it holds that $\|\bar{Y}_j\| = \|\mathscr{L}_{\mathrm{k}}^{-1}(\Pi(\bar{Y}_{j-1}))\| \leq \beta_k\|\bar{Y}_{j-1}\|$. In particular $\|\bar{Y}_j\| \leq \beta_k^j\|\mathscr{L}_{\mathrm{k}}^{-1}\|\|C_1C_2^T\|$, and therefore, by using Remark B.2.3, from (B.14) it follows that

$$\|Y_{j+1} - \bar{Y}_{j+1}\| \leq \|C_1C_2^T\|K\left(\beta^{j+1} + \|\Pi\|\|\mathscr{L}_{\mathrm{k}}^{-1}\| \sum_{t=0}^{j} \beta^{j-t}\beta_k^t\right) e^{-\pi\sqrt{k}}. \tag{B.15}$$

Since $\mathscr{L}_{\mathrm{k}}^{-1}$ converges to $\mathscr{L}^{-1}$, and by using the continuity of the operators, we have that $\|\mathscr{L}_{\mathrm{k}}^{-1}\|$ and $\beta_k$ are bounded by a constant independent of $k$. Therefore from (B.15) it follows that there exists a constant $K_{j+1}$ independent of $k$ such that $\|Y_{j+1} - \bar{Y}_{j+1}\| \leq K_{j+1}e^{-\pi\sqrt{k}}$. The relation (B.12) follows by defining $\bar{X}^{(\ell)} := \sum_{j=0}^{\ell} \bar{Y}_j$ and observing

$$\|X^{(\ell)} - \bar{X}^{(\ell)}\| \leq \sum_{j=0}^{\ell} \|Y_j - \bar{Y}_j\| \leq e^{-\pi\sqrt{k}} \sum_{j=0}^{\ell} K_j = \bar{K}e^{-\pi\sqrt{k}},$$

where $\bar{K} := \sum_{j=0}^{\ell} K_j$. The upper-bound (B.11) follows by Remark B.2.3 iteratively applied to (B.13). □

We want to point out that, although Theorem B.2.4 provides an explicit procedure for constructing an approximation to the solution of (B.3), we later consider a different class of methods. Theorem B.2.4 has only theoretical interest and it is used to motivate the employment of low-rank methods in the solution of (B.3). Moreover, in the numerical simulations (Section B.4), we have observed a decay in the singular values of the solution to (B.3) that it is faster than the one predicted by Theorem B.2.4.

## B.3 Structure exploiting Krylov methods

### B.3.1 Extended Krylov subspace method

In this section we derive a method for (B.3) that belongs to the class called projection methods. We briefly summarize the adaption of the projection method approach in our setting. Projection methods for matrix equations are iterative algorithms based on constructing two sequences of nested subspaces of $\mathbb{R}^n$, i.e., $\mathcal{K}_{k-1} \subset \mathcal{K}_k$ and $\mathcal{H}_{k-1} \subset \mathcal{H}_k$. Justified by the low-rank approximability of the solution, projection methods construct approximations (of the solution to (B.3)) of the form

$$X_k = \mathcal{V}_k Z_k \mathcal{W}_k^T, \tag{B.16}$$

where $\mathcal{V}_k$ and $\mathcal{W}_k$ are matrices with orthonormal columns representing respectively an orthonormal basis of $\mathcal{K}_k$ and $\mathcal{H}_k$. Note that low-rank approximability (in the sense illustrated in, e.g., Theorem B.2.4) is a necessary condition for the success of an approximation of the type (B.16).

The matrix $Z_k$ can be obtained by imposing the Galerkin orthogonality condition, namely the residual

$$\mathcal{R}_k := AX_k + X_kB^T + \sum_{i=1}^{m} N_iX_kM_i^T - C_1C_2^T, \tag{B.17}$$

is such that $\mathcal{V}_k^T\mathcal{R}_k\mathcal{W}_k = 0$. This condition is equivalent to $Z_k$ satisfying the following small and dense generalized Sylvester equation, usually referred to as the *projected problem,*

$$T_kZ_k + Z_kH_k^T + \sum_{i=1}^{m} G_{k,i}Z_kF_{k,i}^T = E_{k,1}E_{k,2}^T, \tag{B.18}$$

where,

$$T_k := \mathcal{V}_k^TA\mathcal{V}_k, \qquad H_k := \mathcal{W}_k^TB\mathcal{W}_k, \qquad E_{k,1} = \mathcal{V}_k^TC_1, \quad E_{k,2} = \mathcal{W}_k^TC_2, \tag{B.19a}$$

$$G_{k,i} := \mathcal{V}_k^TN_i\mathcal{V}_k, \quad F_{k,i} := \mathcal{W}_k^TM_i\mathcal{W}_k, \quad i = 1, \dots, m. \tag{B.19b}$$

The iterative procedure consists in expanding the spaces $\mathcal{K}_k$ and $\mathcal{H}_k$ until the norm of the residual matrix $\mathcal{R}_k$ (B.17) is sufficiently small.

A projection method is efficient only if the subspaces $\mathcal{K}_k$ and $\mathcal{H}_k$ are selected in a way that the projected matrix (B.16) is a good low-rank approximation to the solution without the dimensions of the spaces being large. One of the most popular choices of subspace is the extended Krylov subspace (although certainly not the only choice [24, 17]). Extended Krylov subspaces form the basis of the method called Krylov-plus-inverted Krylov (K-PIK) [39, 15]. For our purposes it is natural to define extended Krylov subspaces with the notation of block Krylov subspaces, cf. [23, Section 6]. Given an invertible matrix $A \in \mathbb{R}^{n\times n}$ and $C \in \mathbb{R}^{n\times r}$, the extended block Krylov subspace can be defined as the sum of two vector spaces, more precisely $E\mathcal{B}_k^{\square}(A, C) := \mathcal{B}_k^{\square}(A, C) + \mathcal{B}_k^{\square}(A^{-1}, A^{-1}C)$, where

$$\mathcal{B}_k^{\square}(A, C) := \operatorname{span}\left(\left\{p(A)Cw \mid \deg(p) \le k,\ w \in \mathbb{R}^{r\times r}\right\}\right) \subseteq \mathbb{C}^{n\times r},$$

denotes the block Krylov subspace, $p \in \mathbb{R}[x]$ is a polynomial, and $\deg(\cdot)$ is the degree function. This space can be also represented as

$$\mathcal{B}_k^{\square}(A, C) = \underbrace{\mathcal{B}_k(A, C) \times \cdots \times \mathcal{B}_k(A, C)}_{r \text{ times}}$$

where $\mathcal{B}_k(A, C) := \operatorname{span}\left(\{p(A)Cw \mid \deg(p) \le k,\ w \in \mathbb{R}^r\}\right) \subseteq \mathbb{C}^n$.

The extended Krylov subspace method is a projection method where we have spaces $\mathcal{K}_k = E\mathcal{B}_k(A, \bar{C}_1)$, $\mathcal{H}_k = E\mathcal{B}_k(B, \bar{C}_2)$; and $\bar{C}_1$, $\bar{C}_2$ are called the starting blocks, which we will show how to select in our setting in Sections B.3.2 and B.3.3. The procedure is summarized in Algorithm B.1 where the matrices $L$ and $R$ are the low-rank factors of (B.16), i.e., they are such that $X_k = LR^T$. Notice that, in the case of generalized Lyapunov equations, the new blocks $V_k$ and $W_k$ are equal (hence also the basis matrices $\mathcal{V}_k$ and $\mathcal{W}_k$) and Algorithm B.1 can be optimized accordingly.

**Algorithm B.1:** Extended Krylov subspace method for generalized Sylvester eqns.

**input :** Matrix coeff.: $A, B, N_1 \ldots, N_m, M_1, \ldots, M_m \in \mathbb{R}^{n\times n}$ , $C_1, C_2 \in \mathbb{R}^{n\times r}$
Starting blocks: $\bar{C}_1 \in \mathbb{R}^{n\times \bar{r}_1}$ and $\bar{C}_2 \in \mathbb{R}^{n\times \bar{r}_2}$
Maximum number of iterations: $d$

**output:** Low-rank factors: $L, R$

Set $V_1 = \texttt{orth}\left(\left[\bar{C}_1, A^{-1}\bar{C}_1\right]\right)$, $W_2 = \texttt{orth}\left(\left[\bar{C}_2, B^{-1}\bar{C}_2\right]\right)$, $\mathcal{V}_0 = \mathcal{W}_0 = \emptyset$
**for** $k = 1, 2, \ldots, d$ **do**
  $\mathcal{V}_k = \left[\mathcal{V}_{k-1}, V_k\right]$ and $\mathcal{W}_k = \left[\mathcal{W}_{k-1}, W_k\right]$
  Compute $T_k, H_k, E_{k,1}, E_{k,2}, G_{k,i}, F_{k,i}$ according to (B.19a)–(B.19b)
  Solve the *projected problem* (B.18)
  Compute $\|\mathcal{R}_k\|_F$ according to (B.21)
  **if** $\|\mathcal{R}_k\|_F \leq \texttt{tol}$ **then**
    Break
  Set $V_k^{(1)}$: first $\bar{r}_1$ columns of $V_k$; Set $V_k^{(2)}$: last $\bar{r}_1$ columns of $V_k$
  Set $W_k^{(1)}$: first $\bar{r}_2$ columns of $W_k$; Set $W_k^{(2)}$: last $\bar{r}_2$ columns of $W_k$
  $V'_{k+1} = \left[AV_k^{(1)}, A^{-1}V_k^{(2)}\right]$ and $W'_{k+1} = \left[BW_k^{(1)}, B^{-1}W_k^{(2)}\right]$
  $\widehat{V}_{k+1} \leftarrow$ block-orthogonalize $V'_{k+1}$ w.r.t. $\mathcal{V}_k$
  $\widehat{W}_{k+1} \leftarrow$ block-orthogonalize $W'_{k+1}$ w.r.t. $\mathcal{W}_k$
  $V_{k+1} = \texttt{orth}(\widehat{V}_{k+1})$ and $W_{k+1} = \texttt{orth}(\widehat{W}_{k+1})$
Compute the decomposition $Z_k = \widehat{L}\widehat{R}^T$
Return $L = \mathcal{V}_k\widehat{L}$ and $R = \mathcal{W}_k\widehat{R}$

**Remark B.3.1.** *The output of Algorithm B.1 represents the factorization $X_k = LR^T$. Under the condition that $\|\mathcal{R}_k\|$ is small, $X_k$ is an approximation of the solution to* (B.3) *such that* $\operatorname{rank}(X_k) \leq 2\min(\bar{r}_1, \bar{r}_2)k$, *where* $\bar{r}_1 = \operatorname{rank}(\bar{C}_1)$ *and* $\bar{r}_2 = \operatorname{rank}(\bar{C}_2)$. *By construction* $\operatorname{range}(L) \subseteq E\mathcal{B}_k(A, \bar{C}_1)$ *and* $\operatorname{range}(R) \subseteq E\mathcal{B}_k(B, \bar{C}_2)$. *As it has been shown, e.g., in [39, 15], an orthonormal basis of $E\mathcal{B}_k$ can computed by means of the block Arnoldi procedure. Moreover, for the case of the Sylvester equation, i.e., $m = 0$, Algorithm B.1 can be effectively applied with starting blocks $\bar{C}_1 = C_1$ and $\bar{C}_2 = C_2$.*

A breakdown in Algorithm B.1 may occur in two situations. During the generation of the basis of the extended Krylov subspaces, (numerical) loss of orthogonality may occur in Steps 9–11. This issue is present already for the Sylvester equation [39, 15] and we refer to [23] for a presentation of safeguard strategies that may mitigate the problem. We assume that the bases $\mathcal{V}_k$ and $\mathcal{W}_k$ have full rank. The other situation where a breakdown may occur is in Step 4. It may happen that the projected problem (B.18) is not solvable. For the Sylvester equation the solvability of the projected problem is guaranteed by the condition that the field of values of $A$ and $-B$ are disjoint [40, Section 4.4.1]. We extend this result, which provides a way to verify the applicability of the method (without carrying out the method). As illustrated in the following proposition, for the generalized Sylvester equation we need an additional condition. Instead of using the field of values, it is natural to phrase this condition in terms of the ratio field of values (defined in, e.g., [18]).

**Proposition B.3.2.** *Consider the generalized Sylvester equation* (B.3) *and assume that the field of values of $A$ and $-B$ are disjoint, and that the ratio field of values of $\sum_{i=1}^{m} M_i \otimes N_i$ and $B \otimes I + I \otimes A$, i.e.,*

$$R\left(\sum_{i=1}^{m} M_i \otimes N_i, B \otimes I + I \otimes A\right) := \left\{ \frac{y^H \left(\sum_{i=1}^{m} M_i \otimes N_i\right) y}{y^H \left(B \otimes I + I \otimes A\right) y} \;\middle|\; y \in \mathbb{C}^{n^2} \setminus \{0\} \right\},$$

*is strictly contained in the open unit disk. Then the projected problem* (B.18) *has a unique solution.*

*Proof.* Let $\mathscr{L}_{proj}(Z) := T_k Z + Z H_k^T$ and $\Pi_{proj}(Z) := \sum_{i=1}^{m} G_{k,i} Z F_{k,i}^T$. The projected problem (B.18) is equivalently written as $\mathscr{L}_{proj}(Z_k) + \Pi_{proj}(Z_k) = E_{k,1} E_{k,2}^T$. Since $A$ and $-B$ have disjoint fields of values, $\mathscr{L}_{proj}$ is invertible [40, Section 4.4.1]. From Theorem B.2.1 we know that there exists a unique solution $Z_k$ to (B.18) if $\rho\left(\mathscr{L}_{proj}^{-1} \Pi_{proj}\right) < 1$. This condition is equivalent to $|\lambda| < 1$, where $(\lambda, v) \in \mathbb{C} \times \mathbb{C}^{(kr)^2} \setminus \{0\}$ is any eigenpair of the following generalized eigenvalue problem

$$\left(\sum_{i=1}^{m} F_{k,i} \otimes G_{k,i}\right) v = \lambda (H_k \otimes I + I \otimes T_k) v. \tag{B.20}$$

Using the properties of the Kronecker product, equation (B.20) can be written as

$$\sum_{i=1}^{m} (W_k^T \otimes V_k^T)\,(M_i \otimes N_i)\,(W_k \otimes V_k)\, v = \lambda (W_k^T \otimes V_k^T)\,(B \otimes I + I \otimes A)\,(W_k \otimes V_k)\, v.$$

By multiplying the above equation from the left with $v^H$ we have that

$$|\lambda| = \left| \frac{x^H \left(\sum_{i=1}^{m} M_i \otimes N_i\right) x}{x^H \left(B \otimes I + I \otimes A\right) x} \right|, \quad x := (W_k \otimes V_k)\, v.$$

By using that $R\left(\sum_{i=1}^{m} M_i \otimes N_i, B \otimes I + I \otimes A\right)$ is strictly contained in the open unit disk we conclude that $|\lambda| < 1$. □

**Observation B.3.3.** *The computation of the matrices $T_k$, $H_k$ (Step 3) and the orthogonalization of the new blocks $V_{k+1}, W_{k+1}$ (Steps 9–11) can be efficiently performed as in [39, Section 3] where a modified Gram–Schmidt method is employed in the orthogonalization. The matrices $G_{k,i}$ and $F_{k,i}$ (Step 3) can be computed by extending the matrices $G_{k-1,i}$ and $F_{k-1,i}$ with a block-column and a block-row. Moreover, the matrix $X_k$ is never explicitly formed. In particular, the Frobenius norm of the residual* (B.17) *can be computed as*

$$\|\mathcal{R}_k\|_F^2 = \|\tau_{k+1}(e_k \otimes I_{2r})^T Z_k\|_F^2 + \|Z_k(e_k \otimes I_{2r})^T h_{k+1}^T\|_F^2. \tag{B.21}$$

*This follows by replacing in* (B.17) *the following Arnoldi-like relations [41, equation (4)]*

$$A\mathcal{V}_k = \mathcal{V}_k T_k + V_{k+1}\tau_{k+1}(e_k \otimes I_{2r})^T, \quad B\mathcal{W}_k = \mathcal{W}_k H_k + W_{k+1}h_{k+1}(e_k \otimes I_{2r})^T.$$

**Remark B.3.4.** *Algorithm B.1 is a block method in the sense that it uses a block Arnoldi procedure to generate the basis matrices $\mathcal{V}_k$ and $\mathcal{W}_k$. The basis generated by Algorithm B.1 can hence be interpreted in a block sense and related to the block Krylov spaces $E\mathcal{B}_k^{\square}(A, \bar{C}_1)$ and $E\mathcal{B}_k^{\square}(B, \bar{C}_2)$, cf. [20, Section 2]. However, in the framework of projection methods for matrix equations, the span of the columns of $\mathcal{V}_k$ and $\mathcal{W}_k$ are often considered as the projection spaces. Moreover, the columns of $\mathcal{V}_k$ and $\mathcal{W}_k$ are respectively a basis for $E\mathcal{B}_k(A, \bar{C}_1)$ and $E\mathcal{B}_k(B, \bar{C}_2)$. In particular, each column of $L$ and $R$ is respectively in the space $E\mathcal{B}_k(A, \bar{C}_1)$ and $E\mathcal{B}_k(B, \bar{C}_2)$. This is equivalently expressed as* $\mathrm{range}(L) \subseteq E\mathcal{B}_k(A, \bar{C}_1)$ *and* $\mathrm{range}(R) \subseteq E\mathcal{B}_k(B, \bar{C}_2)$. *Therefore, in the following analysis, we derive and use properties of the spaces $E\mathcal{B}_k(A, \bar{C}_1)$ and $E\mathcal{B}_k(B, \bar{C}_2)$.*

### B.3.2 Krylov subspace and low-rank commuting matrices

Algorithm B.1 is efficient only if the starting blocks $\bar{C}_1$ and $\bar{C}_2$ are low-rank matrices and if the subspaces $E\mathcal{B}_k(A, \bar{C}_1)$ and $E\mathcal{B}_k(B, \bar{C}_2)$ have good approximation properties. Our approach consists in applying Algorithm B.1 directly to the generalized Sylvester equation (B.3). Therefore, we now derive certain approximation properties of the solution to (B.3) that naturally suggest a proper choice of the starting blocks. The low-rank of the starting blocks will rely on the low-rank commutation property of the coefficients (B.4). Our reasoning can be described as follows:

- The solution to the generalized Sylvester equation (B.3) can be represented as a converging Neumann series (B.5). By truncating this series, $X^{(\ell)}$ gives an approximation to the solution to (B.3), in the sense of Remark B.2.2.
- The coefficients of the Neumann series (B.6) 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 Algorithm B.1 (as in Remark B.3.1) applied to each Sylvester equation (B.6).
- The matrix $\tilde{X}^{(\ell)} = \sum_{j=0}^{\ell} \tilde{Y}_j$, that can be viewed as an approximation to the solution of (B.3), can be factorized as $\tilde{X}^{(\ell)} = LR^T$ such that $\mathrm{range}(L) \subseteq E\mathcal{B}_k(A, \bar{C}_1)$

and $\mathrm{range}(R) \subseteq E\mathcal{B}_k(B, \bar{C}_2)$. We give a characterization and a procedure for computing $\bar{C}_1$ and $\bar{C}_2$. One condition for these matrices to be low-rank concerns the commutators (B.4) being low-rank. These two matrices will be used as starting blocks in Algorithm B.1.

Although the above reasoning is based on solving a sequence of Sylvester equations, our approach consists of applying Algorithm B.1, only one time, directly to the generalized Sylvester equation (B.3).

We first need a technical result which shows that, if the commutator of two matrices has low rank, then the corresponding commutator, where one matrix is taken to a given power, has also low rank. The rank increases at most linearly with respect to the power of the matrix. The precise statement is presented in the following lemma.

**Lemma B.3.5.** *Suppose that $A$ and $N$ are matrices such that $\mathrm{com}(A, N) = U\tilde{U}^T$. Then,*

$$\mathrm{com}(A^j, N) = \sum_{k=0}^{j-1} A^k U \tilde{U}^T A^{j-k-1}.$$

*Proof.* The proof is by induction. The basis of induction is trivially verified for $j = 1$. Assume that the claim is valid for $j$, then the induction step follows by observing that

$$\mathrm{com}(A^{j+1}, N) = A^{j+1}N - NA^{j+1} = A^j U\tilde{U}^T + (A^j N - NA^j)A,$$

and applying the induction hypothesis on $A^j N - NA^j$. □

As pointed out in Remark B.3.1, $C_1$ and $C_2$ are natural starting blocks for the Sylvester equation. If we apply this result to the sequence of Sylvester equations in Theorem B.2.1, with $\mathscr{L}$ and $\Pi$ defined as (B.1)–(B.2), we obtain subspaces with a particular structure. For example, the approximation $L_0 R_0^T$ to $Y_0$ provided by Algorithm B.1 is such that $\mathrm{range}(L_0) \subseteq E\mathcal{B}_k(A, C_1)$ and $\mathrm{range}(R_0) \subseteq E\mathcal{B}_k(B, C_2)$. Since $Y_0$ is contained in the right-hand side of the definition of $Y_1$, in order to compute an approximation of $Y_1$, we should consider the subspaces $N_i \cdot E\mathcal{B}_k(A, C_1)$ and $M_i \cdot E\mathcal{B}_k(B, C_2)$ for $i = 1, \ldots, m$. By using the low-rank commutation property (B.4) such subspaces can be characterized by the following result.

**Theorem B.3.6.** *Assume that $A \in \mathbb{R}^{n\times n}$ is nonsingular and let $N \in \mathbb{R}^{n\times n}$ such that $\mathrm{com}(A, N) = U\tilde{U}^T$ with $U, \tilde{U} \in \mathbb{R}^{n\times s}$. Let $C \in \mathbb{R}^{n\times r}$, then*

$$N \cdot E\mathcal{B}_k(A, C) \subseteq E\mathcal{B}_k(A, \big[NC, U\big]).$$

*Proof.* Let $Np(A)Cw + Nq(A^{-1})Cv$ be a generator of $N \cdot E\mathcal{B}_k(A, C)$, where $p(x) = \sum_{j=0}^{k} \alpha_j x^j$ and $q(x) = \sum_{j=0}^{k} \beta_j x^j$. Then, with a direct usage of Lemma B.3.5, the vector $Np(A)Cw$ can be expressed as an element of $E\mathcal{B}_k(A, \big[NC, U\big])$ in the following way

$$Np(A)Cw = N\sum_{j=0}^{k} \alpha_j A^j Cw = p(A)NCw - \sum_{j=0}^{k}\sum_{\ell=0}^{j-1} \alpha_j A^\ell U \left(\tilde{U}^T A^{j-1-\ell} Cw\right).$$

We can show that $Nq(A^{-1})Cv$ belongs to the subspace $E\mathcal{B}_k(A, \left[NC, U\right])$ with the same procedure and by using that $\mathrm{com}(A^{-1}, N) = -(A^{-1}U)(A^{-T}\tilde{U})^T$. □

In order to ease the notation and improve conciseness of the results that follow, we introduce the following multivariate generalization of the Krylov subspace for more matrices

$$\mathcal{G}_d(N_1, \ldots, N_m; U) := \mathrm{span}\left(\{p(N_1, \ldots, N_m)Uz | \deg(p) \leq d, z \in \mathbb{R}^r\}\right),$$

where $U \in \mathbb{R}^{n\times r}$ and $p$ is a non-commutative multivariate polynomial in the free algebra $\mathbb{R} < x_1, \ldots, x_N >$ (in the sense of [12, Chapter 10]).

**Observation B.3.7.** *Note that $\mathcal{G}_d(N_1, \ldots, N_m; U)$ is the space generated by the columns of the matrices obtained multiplying (in any order) $s \leq d$ matrices $N_i$ and the matrix $U$. In particular this space can be equivalently characterized as*

$$\mathcal{G}_d(N_1, \ldots, N_m; U) = \mathrm{span}\left(\{N_{i_1} \cdots N_{i_s} Uz \mid 1 \leq i_j \leq m, 0 \leq s \leq d, z \in \mathbb{R}^r\}\right).$$

*This definition generalizes the definition of the standard Krylov subspace in the sense that $\mathcal{G}_d(N; U) = \mathcal{B}_d(N, U)$.*

The solution strategy for (B.3) outlined at the beginning of this subsection is formalized in the following theorem. In order to state the theorem we need the result of the application of the extended Krylov method to the (standard) Sylvester equations of the form

$$A\mathcal{Y} + \mathcal{Y}B^T = C_1 C_2^T, \tag{B.22a}$$

$$A\mathcal{Y} + \mathcal{Y}B^T = -\sum_{i=1}^{m} (N_i L_j)(M_i R_j)^T, \tag{B.22b}$$

as described in [39, 15]. As already stated in Remark B.3.1, this is identical to applying Algorithm B.1 with $m = 0$.

**Theorem B.3.8.** *Consider the generalized Sylvester equation* (B.3)*, with coefficients commuting according to* (B.4)*. Let $\tilde{Y}_0 = L_0 R_0^T$ be the result of Algorithm B.1 applied to the (standard) Sylvester equation* (B.22a) *with starting blocks $\bar{C}_1 = C_1$ and $\bar{C}_2 = C_2$. Moreover, for $j = 0, \ldots, \ell - 1$, let $\tilde{Y}_{j+1} = L_{j+1} R_{j+1}^T$ be the result of Algorithm B.1 applied to the Sylvester equation* (B.22b) *with starting blocks $\bar{C}_1 = \left[N_1 L_j, \ldots, N_m L_j\right]$ and $\bar{C}_2 = \left[M_1 R_j, \ldots, M_m R_j\right]$. Let $\tilde{X}^{(\ell)}$ be the approximation of the truncated Neumann series* (B.7) *given by*

$$\tilde{X}^{(\ell)} := \sum_{j=0}^{\ell} \tilde{Y}_j.$$

*Then, there exist matrices $L, R, \hat{C}_1^{(\ell)}, \hat{C}_2^{(\ell)}$ such that $\mathrm{range}(L) \subseteq E\mathcal{B}_{(\ell+1)d}(A, \hat{C}_1^{(\ell)})$ and $\mathrm{range}(R) \subseteq E\mathcal{B}_{(\ell+1)d}(B, \hat{C}_2^{(\ell)})$ and*

$$\tilde{X}^{(\ell)} = LR^T,$$

*where*

$$\operatorname{range}(\hat{C}_1^{(\ell)}) \subseteq \mathcal{G}_\ell(N_1,\dots,N_m;C_1) + \mathcal{G}_{\ell-1}(N_1,\dots,N_m;U), \tag{B.23a}$$

$$\operatorname{range}(\hat{C}_2^{(\ell)}) \subseteq \mathcal{G}_\ell(M_1,\dots,M_m;C_2) + \mathcal{G}_{\ell-1}(M_1,\dots,M_m;Q), \tag{B.23b}$$

*and* $U := \begin{bmatrix} U_1,\dots,U_m \end{bmatrix}$, $Q := \begin{bmatrix} Q_1,\dots,Q_m \end{bmatrix}$.

*Proof.* We start the proof by showing that for $j = 0,\dots,\ell$, there exists a matrix $S_j$ such that $\operatorname{range}(L_j) \subseteq E\mathcal{B}_{(j+1)d}(A,S_j)$ and

$$\operatorname{range}(S_j) \subseteq \tag{B.24}$$
$$\operatorname{span}\left(\left\{\left(\prod_{k=1}^{j} N_{i_k}\right) C_1 w + p(N_1,\dots,N_m)Uz \;\middle|\; w\in\mathbb{R}^r, z\in\mathbb{R}^s, 1\le i_k\le m, \deg(p)\le j-1\right\}\right),$$

where $s = \sum_{i=1}^m s_i$ and $s_i$ denotes the number of columns of $U_i$. We prove this claim by induction. The basis of induction is trivially verified for $j = 0$ with $S_0 := C_1$ and using Remark B.3.1. We now assume that the claim is valid for some $j$ and perform the induction step. Remark B.3.1 implies that $\operatorname{range}(L_{j+1}) \subseteq E\mathcal{B}_d(A, \begin{bmatrix} N_1L_j,\dots,N_mL_j \end{bmatrix})$. From Theorem B.3.6 and the induction hypothesis we have that

$$\operatorname{range}(N_iL_j) \subseteq E\mathcal{B}_{(j+1)d}(A, \begin{bmatrix} N_i, S_jU_i \end{bmatrix})$$

for any $i = 1,\dots,m$. Therefore we have that

$$\operatorname{range}(L_{j+1}) \subseteq E\mathcal{B}_{(j+2)d}(A, \begin{bmatrix} N_1S_j,\dots,N_mS_j,U \end{bmatrix}).$$

We define $S_{j+1} := \begin{bmatrix} N_1S_j,\dots,N_mS_j,U \end{bmatrix}$ which concludes the induction.

From (B.24) we now obtain the relation

$$\operatorname{range}(\begin{bmatrix} S_0,\dots,S_j \end{bmatrix}) \subseteq \mathcal{G}_j(N_1,\dots,N_m;C_1) + \mathcal{G}_{j-1}(N_1,\dots,N_m;U),$$

that directly implies (B.23a) by setting $\hat{C}_1^{(\ell)} := \begin{bmatrix} S_0,\dots,S_\ell \end{bmatrix}$. Equation (B.23b) follows from completely analogous reasoning. The final conclusion follows by defining $L := \begin{bmatrix} L_0,\dots,L_\ell \end{bmatrix}$ and $R := \begin{bmatrix} R_0,\dots,R_\ell \end{bmatrix}$. □

The main message of the previous theorem can be summarized as follows: The low-rank factors of the approximation of $X^{(\ell)}$ (B.7) obtained by solving the Sylvester equations (B.6) with K-PIK [39, 15] (that is equivalent to Algorithm B.1 as discussed in Remark B.3.1), are contained in an extended Krylov subspace with a specific choice of the starting blocks. In particular the starting blocks are selected as $\bar{C}_1 = \hat{C}_1^{(\ell)}$, $\bar{C}_2 = \hat{C}_2^{(\ell)}$ where $\hat{C}_1^{(\ell)}$ and $\hat{C}_2^{(\ell)}$ fulfill (B.23a)–(B.23b). Our approach consists in applying Algorithm B.1 directly to the generalized Sylvester equation (B.3) with this choice of the starting blocks.

A practical procedure that generates starting blocks that fulfill (B.23) consists in selecting $\bar{C}_1$ and $\bar{C}_2$ such that their columns are a basis of the subspaces $\mathcal{G}_\ell(N_1,\dots,N_m;C_1) +$

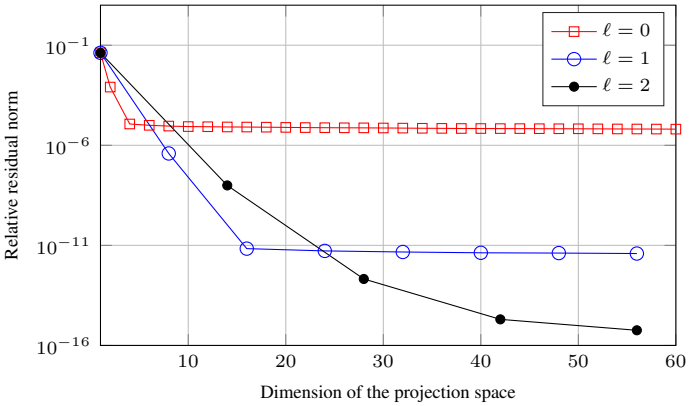



Figure B.1: Convergence history of Algorithm B.1 applied to a (randomly generated) generalized Lyapunov equation $AX + XA + NXN = cc^T$ with $A$ circulant and $N$ sum of a circulant matrix plus a rank one $uu^T$ correction. The algorithm is tested for the stating blocks $\hat{C}^{(\ell)}$ with $\ell = 0, 1, 2$ selected according to Theorem B.3.6, i.e., $\hat{C}^{(0)} = c$, $\hat{C}^{(1)} = \big[c, Nc, u\big]$, $\hat{C}^{(2)} = \big[c, Nc, N^2c, u, Nu\big]$.

$\mathcal{G}_{\ell-1}(N_1, \dots, N_m; U)$ and $\mathcal{G}_\ell(M_1, \dots, M_m; C_2) + \mathcal{G}_{\ell-1}(M_1, \dots, M_m; Q)$ respectively. A basis of such spaces can be computed by using Observation B.3.7. For example a basis of $\mathcal{G}_2(N_1, N_2; U)$ can be obtained from the columns of the matrix

$$\big[U,\ N_1U,\ N_2U,\ N_1N_2U,\ N_2N_1U,\ N_1^2U,\ N_2^2U\big]\,.$$

**Observation B.3.9.** *The choice of the starting blocks involves the parameter $\ell$. In theory, a suitable choice of $\ell$ could be derived by using Remark B.2.2. However, this is not always possible since the quantity $\rho(\mathscr{L}^{-1}\,\Pi)$ is, in many cases, not known and computationally demanding to compute/estimate. The choice of $\ell$ is a trade-off between accuracy and efficiency. The starting blocks $\hat{C}_1^{(\ell)}$ and $\hat{C}_2^{(\ell)}$, for large $\ell$, provide spaces $E\mathcal{B}_k(A, \bar{C}_1)$ and $E\mathcal{B}_k(B, \bar{C}_2)$ with better approximation features, but with potentially higher dimensions. See Figure B.1. This leads to an increment in the computational cost of the whole procedure but to a more accurate approximation to the solution to* (B.3).

Our approach is computationally attractive only if the starting blocks $\bar{C}_1 = \hat{C}_1^{(\ell)}$ and $\bar{C}_2 = \hat{C}_2^{(\ell)}$ have low rank. This condition is fulfilled if the commutators (B.4) are low-rank matrices, see Observation B.3.7. Under this assumption, the advantages of the proposed method can be summarized as follows: Algorithm B.1 generates only one pair of extended Krylov subspaces with given starting blocks. There are other methods based on generating several projection subspaces (with the same coefficient matrix), e.g., a direct computation of $\tilde{X}^{(\ell)}$ or [38, 16] and [7, Section 5.3]. An advantage of our approach with respect to these methods, consists in avoiding redundancy in the approximation spaces. In particular, if several Krylov subspaces with the same coefficient matrix are generated independently

of each other, they may have a nontrivial intersection or in general they may have similar approximation properties. From a computational point of view, this means that considerable efforts are wasted to breed similar information.

In certain cases the dimension of the subspaces $\mathcal{G}_\ell$ is bounded for all the $\ell$, i.e., there exist matrices $\bar{C}_1 \in \mathbb{R}^{n\times \bar{r}_1}$ and $\bar{C}_2 \in \mathbb{R}^{n\times \bar{r}_2}$ such that $\operatorname{range}(\hat{C}_1^{(\ell)}) \subseteq \operatorname{range}(\bar{C}_1)$ and $\operatorname{range}(\hat{C}_2^{(\ell)}) \subseteq \operatorname{range}(\bar{C}_2)$ for all $\ell$. This condition is satisfied, e.g., if the matrix coefficients $N_i$, $M_i$ are nilpotent/idempotent or in general if they have low degree minimal polynomials. Therefore, it is possible to select the starting blocks such that Algorithm B.1 provides an approximation of $X^{(\ell)}$ for all $\ell$, i.e., the full series (B.5) is approximated. These situations naturally appear in applications, see the numerical example in Section B.4.3.

### B.3.3 Krylov subspace method and low-rank matrices

Our numerical method can be improved for the following special case. We now consider a generalized Sylvester equation (B.3) where $N_i = \mathcal{U}_i\tilde{\mathcal{U}}_i^T$ and $M_i = \mathcal{Q}_i\tilde{\mathcal{Q}}_i^T$ are low-rank matrices. Obviously, the commutators $\operatorname{com}(A, N_i)$ and $\operatorname{com}(B, M_i)$ also have low rank and the theory and the procedure presented in the previous section cover this case. However, the solution to (B.3) can be further characterized and an efficient (and different) choice of the starting blocks $\bar{C}_1, \bar{C}_2$ can be derived. The assumption $\rho(\mathscr{L}^{-1}\,\Pi) < 1$ is no longer needed in order to justify the low-rank approximability. This property can be illustrated with a Sherman–Morrison–Woodbury argument as proposed in [8]. The following proposition shows that, the generalized Sylvester equation (B.3) can be implicitly written as a Sylvester equation with right-hand side involving the columns of the matrices $\mathcal{U}_i$ and $\mathcal{Q}_i$ for $i = 1, \ldots, m$. By using Remark B.3.1 this leads to the following natural choice of starting blocks: $\bar{C}_1 = \big[C_1, \mathcal{U}_1, \ldots, \mathcal{U}_m\big]$ and $\bar{C}_2 = \big[C_2, \mathcal{Q}_1, \ldots, \mathcal{Q}_m\big]$.

**Proposition B.3.10.** *Consider the generalized Sylvester equation* (B.3)*. Assume that $\mathscr{L}$ is invertible, and that $N_i = \mathcal{U}_i\tilde{\mathcal{U}}_i^T$ and $M_i = \mathcal{Q}_i\tilde{\mathcal{Q}}_i^T$ are such that $\mathcal{U}_i, \tilde{\mathcal{U}}_i \in \mathbb{R}^{n\times s_i}$ and $\mathcal{Q}_i, \tilde{\mathcal{Q}}_i \in \mathbb{R}^{n\times t_i}$. Then there exist $\alpha_{i,j,\ell} \in \mathbb{R}$ for $j = 1, \ldots, s_i$, $\ell = 1, \ldots, t_i$, and $i = 1, \ldots, m$, such that*

$$AX + XB^T = C_1C_2^T - \sum_{i,j,\ell} \alpha_{i,j,\ell}\,\mathcal{U}_i^{(j)}\,{\mathcal{Q}_i^{(\ell)}}^T,$$

*where $\mathcal{U}_i^{(j)}$ is the $j$th column of $\mathcal{U}_i$, and $\mathcal{Q}_i^{(\ell)}$ is the $\ell$th column of $\mathcal{Q}_i$.*

*Proof.* The proof follows by [37, Theorem 4.1], or [16, Proposition 3.3], or [36, Theorem 2.2]. □

### B.3.4 Solving the projected problem

In order to apply Algorithm B.1 we need to solve the projected problem in Step 4. The projected problem has to be solved in every iteration and efficiency is therefore required in practice. For completeness we now derive a procedure to solve the projected problem

based on the Neumann series expansion derived in Section B.2.1, although this is certainly not the only option. The derivation is based on the following observations. The projected problem is a small generalized Sylvester equation (B.3), and the computation of $X^{(\ell)}$ in (B.7) requires solving $\ell + 1$ Sylvester equations (B.6). Since the Sylvester equations (B.6) are defined by the same coefficients, they can be simultaneously reduced to triangular form (cf., [36, Section 3]) as follows:

$$U_A \widetilde{Y}_0 + \widetilde{Y}_0 U_B^T = \widetilde{C}_1 \widetilde{C}_2^T, \tag{B.25a}$$

$$U_A \widetilde{Y}_{j+1} + \widetilde{Y}_{j+1} U_B^T = -\sum_{i=1}^{m} \widetilde{N}_i \widetilde{Y}_j \widetilde{M}_i^T, \qquad j = 0, \ldots, \ell - 1, \tag{B.25b}$$

where we have defined

$$\widetilde{C}_1 := Q_A^T C_1, \quad \widetilde{C}_2 := Q_B^T C_2, \quad \widetilde{N}_i := Q_A^T N_i Q_A, \quad \widetilde{M}_i := Q_B^T M_i Q_B, \tag{B.26}$$

and $A = Q_A U_A Q_A^T$ and $B = Q_B U_B Q_B^T$ denote the Schur decompositions. The Sylvester equations (B.25) with triangular coefficients can be efficiently solved with backward substitution as in the Bartels–Stewart algorithm [3] and it holds that $X^{(\ell)} = Q_A \left( \sum_{j=0}^{\ell} \widetilde{Y}_j \right) Q_B^T$. The Frobenius norm of the residual $\mathcal{R}^{(\ell)} := AX^{(\ell)} + X^{(\ell)} B^T + \sum_{i=1}^{m} N_i X^{(\ell)} M_i^T - C_1 C_2^T$ can be computed without explicitly constructing $X^{(\ell)}$ as follows:

$$\|\mathcal{R}^{(\ell)}\|_F = \left\| \sum_{i=1}^{m} \widetilde{N}_i \widetilde{Y}_\ell \widetilde{M}_i^T \right\|_F. \tag{B.27}$$

The previous relation follows by simply using the properties of the Frobenius norm (invariance under orthogonal transformations) and the relations (B.25).

In conclusion, the following iterative procedure can be used to approximate the solution to (B.3): The matrices (B.26) are precomputed, then the Sylvester equations in triangular form (B.25) are solved until the residual of the Neumann series (B.27) is sufficiently small. The approximation $X^{(\ell)}$ is not computed during the iteration, but only constructed after the iteration has completed. The procedure is summarized in Algorithm B.2.

## B.4 Numerical examples

We now illustrate our approach with several examples. In the first two examples, we compare our approach with two different methods for generalized Lyapunov equations: BilADI [8] and GLEK [38]. The results are generally in favor of our approach, since the other methods are less specialized to the specific structure. However, they have a wider applicable problem domain. Two variants of BilADI are considered. In the first variant we select the Wachspress shifts, see e.g., [43], computed with the software available on

**Algorithm B.2:** Neumann series approach for (B.3).

**input :** Matrix coefficients: $A, B, N_1 \ldots, N_m, M_1, \ldots, M_m, C_1, C_2$
**output:** Truncated Neumann series $X^{(\ell)}$

1 Compute the Schur decompositions $A = Q_A U_A Q_A^T$, $B = Q_B U_B Q_B^T$
2 Compute $\widetilde{C}_1$, $\widetilde{C}_2$, $\widetilde{N}_i$ $\widetilde{M}_i$ for all $i = 1, \ldots, m$ according to (B.26)
3 Solve $U_A \widetilde{Y}_0 + \widetilde{Y}_0 U_B^T = \widetilde{C}_1 \widetilde{C}_2^T$ and set $\widetilde{X} = \widetilde{Y}_0$
**for** $j = 0, 1, \ldots$ *till convergence* **do**
4 Solve $U_A \widetilde{Y}_{j+1} + \widetilde{Y}_{j+1} U_B^T = -\sum_{i=1}^m \widetilde{N}_i \widetilde{Y}_j \widetilde{M}_i^T$ and set $\widetilde{X} = \widetilde{X} + \widetilde{Y}_{j+1}$
5 Compute $\|\mathcal{R}^{(j+1)}\|_F = \|\sum_{i=1}^m \widetilde{N}_i \widetilde{Y}_{j+1} \widetilde{M}_i^T\|_F$
**if** $\|\mathcal{R}^{(j+1)}\|_F \leq \texttt{tol}$ **then**
6 Set $\ell = j + 1$
7 Break
8 Return $X^{(\ell)} = Q_A \widetilde{X} Q_B^T$

Saak's web page[1]. In the second variant $\mathcal{H}_2$-optimal shifts [7] are used. The GLEK code is available at the web page of Simoncini[2]. This algorithm requires fine-tune of several thresholds. We selected `tol_inexact`$= 10^{-2}$ while the default setting is used for all the other thresholds. The implementation of our approach is based on the modification of K-PIK [39, 15] for generalized Sylvester equation as described in Algorithm B.1. The projected problems, computed in Step 4, are solved with the procedure described in the Section B.3.4. A MATLAB implementation of Algorithm B.1 is available online[3].

In all the methods that we test, the stopping criterion is based on the relative residual norm and the algorithms are stopped when it reaches `tol` $= 10^{-6}$. We compare: number of iterations, memory requirements, rank of the computed approximation, number of linear solves (involving the matrices $A$ and $B$ potentially shifted) and total execution CPU-times.

As memory requirement (denoted Mem. in the following tables) we consider the number of vectors of length $n$ stored during the solution process. In particular, for Algorithm B.1 it consists of the dimension of the approximation space. In GLEK, a sequence of extended Krylov subspaces is generated and the memory requirement corresponds to the dimension of the largest space in the sequence. For the bilinear ADI approach the memory requirement consists of the number of columns of the low-rank factor of the solution. For GLEK, we just report the number of outer iterations. The CPU–times reported for BilADI do no take into account the time for the shifts computation. For the linear solves, the LU-factors are precomputed and reused in the algorithms. All results were obtained with MATLAB R2015a on a computer with two 2 GHz processors and 128 GB of RAM.

[1] `https://www.mpi-magdeburg.mpg.de/1694482/wachspress`
[2] `https://www.dm.unibo.it/~simoncin/software.html`
[3] `https://www.dm.unibo.it/~davide.palitta3`

### B.4.1 A multiple input multiple output system (MIMO)

The time invariant multiple input multiple output (MIMO) bilinear system described in [32, Example 2] yields the following generalized Lyapunov equation

$$AX + XA^T + \gamma^2 \sum_{i=1}^{2} N_i X N_i^T = CC^T, \tag{B.28}$$

where $\gamma \in \mathbb{R}$, $\gamma > 0$, $A = \text{tridiag}(2, -5, 2)$, $N_1 = \text{tridiag}(3, 0, -3)$ and $N_2 = -N_1 + I$. We consider $C \in \mathbb{R}^{n\times 2}$ being a normalized random matrix. In the context of bilinear systems, the solution to (B.28), referred to as *Gramian*, is used for computing energy estimates and reachability of the states. The number $\gamma$ is a scaling parameter selected in order to ensure the solvability of the problem (B.28) and the positive definiteness of the solution, namely $\rho(\mathscr{L}^{-1}\,\Pi) < 1$. This parameter corresponds to rescaling the input of the underlying problem with a possible reduction in the region where energy estimates hold. Therefore, it is preferable not to employ very small values of $\gamma$. See [9] for detailed discussions.

For this problem the commutators have low rank. More precisely, $\text{com}(A, N_1) = -\text{com}(A, N_2) = U\tilde{U}^T$, with $U = 2\sqrt{3}\,\big[e_1, e_n\big]$ and $\tilde{U} = 2\sqrt{3}\,\big[e_1, -e_n\big]$. As proposed in Section B.3.2 we use Algorithm B.1 with starting blocks $\bar{C}_1 = \bar{C}_2 = \big[C, N_1C, U\big]$ since $\text{range}(C_1^{(1)}) = \text{range}(\big[C, N_1C, N_2C, U\big]) = \text{range}\,\big(\big[C, N_1C, U\big]\big)$. Table B.1 illustrates the performances of our approach and the other low-rank methods, GLEK and the BilADI, as $\gamma$ varies. We notice that, the number of linear solves that our projection method requires

| | $\gamma$ | Its. | Mem. | rank($X$) | Lin. solves | CPU time |
|---|---|---|---|---|---|---|
| BilADI (4 Wach.) | 1/6 | 10 | 55 | 55 | 320 | 51.26 |
| BilADI (8 $\mathcal{H}_2$-opt.) | 1/6 | 10 | 55 | 55 | 320 | 51.54 |
| GLEK | 1/6 | 9 | 151 | 34 | 644 | 14.17 |
| Algorithm B.1 | 1/6 | 6 | 72 | 60 | 36 | 3.77 |
| BilADI (4 Wach.) | 1/5 | 14 | 71 | 71 | 588 | 55.15 |
| BilADI (8 $\mathcal{H}_2$-opt.) | 1/5 | 14 | 69 | 69 | 586 | 54.31 |
| GLEK | 1/5 | 12 | 173 | 39 | 1016 | 22.06 |
| Algorithm B.1 | 1/5 | 6 | 72 | 61 | 36 | 4.23 |
| BilADI (4 Wach.) | 1/4 | 24 | 89 | 89 | 1454 | 67.61 |
| BilADI (8 $\mathcal{H}_2$-opt.) | 1/4 | 23 | 89 | 89 | 1371 | 66.83 |
| GLEK | 1/4 | 21 | 218 | 50 | 2348 | 51.49 |
| Algorithm B.1 | 1/4 | 8 | 96 | 81 | 48 | 6.72 |

Table B.1: MIMO example. Comparison of low-rank methods for $n = 50000$.

is always much less than for the other methods. Moreover, it seems that moderate variations of $\gamma$, that correspond to variations of $\rho(\mathscr{L}^{-1}\,\Pi)$, have a smaller influence on the number of iterations in our method compared to the other algorithms.

### B.4.2 A low-rank problem

We now consider the following generalized Lyapunov equation

$$AX + XA^T + UV^T XVU^T = cc^T, \tag{B.29}$$

where $A = n^2\text{tridiag}(1, -2, 1)$ and $U, V \in \mathbb{R}^{n\times m}$, $c \in \mathbb{R}^n$ have random entries and unit norm. We use Algorithm B.1, and as proposed in Section B.3.3, we select $\bar{C}_1 = \bar{C}_2 = \begin{bmatrix} c, U \end{bmatrix}$ as starting blocks. In Table B.2 we report the results of the comparison to the other methods for $m = 1$. We notice that our approach requires the lowest number

| | $n$ | Its. | Mem. | rank($X$) | Lin. solves | CPU time |
|---|---|---|---|---|---|---|
| BilADI (4 Wach.) | 10000 | 60 | 57 | 57 | 2462 | 4.25 |
| BilADI (8 $\mathcal{H}_2$-opt.) | 10000 | 42 | 55 | 55 | 1420 | 2.54 |
| GLEK | 10000 | 4 | 240 | 28 | 310 | 3.10 |
| Algorithm B.1 | 10000 | 46 | 184 | 49 | 92 | 2.77 |
| BilADI (4 Wach.) | 50000 | 327 | 61 | 61 | 18673 | 315.56 |
| BilADI (8 $\mathcal{H}_2$-opt.) | 50000 | 96 | 61 | 61 | 4580 | 81.47 |
| GLEK | 50000 | 4 | 454 | 28 | 565 | 24.78 |
| Algorithm B.1 | 50000 | 78 | 312 | 47 | 156 | 21.09 |
| BilADI (4 Wach.) | 100000 | - | - | - | - | - |
| BilADI (8 $\mathcal{H}_2$-opt.) | 100000 | 84 | 65 | 65 | 4058 | 174.04 |
| GLEK | 100000 | 4 | 457 | 29 | 631 | 66.77 |
| Algorithm B.1 | 100000 | 97 | 388 | 44 | 194 | 55.58 |

Table B.2: Comparison of low-rank methods applied to (B.29) varying $n$ with $m = 1$.

of linear solves. The ADI approaches demand the lowest storage because of the column compression strategy performed at each iteration. However, due to the large number of linear solves, these methods are slower compared to our approach. For large-scale problems the BilADI method with 4 Wachspress shifts does not converge in $500$ iterations. GLEK provides the solution with the smallest rank.

We now consider (B.29) for $m > 1$. Notice that, in equation (B.3), this corresponds to have the operator $\Pi$ (B.2) defined by the sum of $m$ terms of rank 1. In particular, we apply Algorithm B.1 to equation (B.29) for $m = 5, 10, 15$. The results are collected in Table B.3. The number of iterations needed decreases as $m$ increases. However, since the rank of the starting block increases with $m$, the dimension of the approximation space increases, and thus the number of linear solves. As a result, the computation time increases with $m$.

If we replace the matrix $A$ with $A/n^2$ in equation (B.29), neither BilADI nor GLEK converge since the Lyapunov operator is no longer dominant, i.e., $\rho(\mathscr{L}^{-1}\,\Pi) > 1$. However, our algorithm still converges, and for $n = 10000$, $m = 1$, it provides a solution $X$ in $46$ iterations with $\text{rank}(X) = 184$. In this case, the projected problems cannot be solved with the approach described in the Section B.3.4. However, since the projected problems are also of the form (B.29), they can be solved with a Sherman–Morrison–Woodbury ap-

| $n$ | $m$ | Its. | Mem. | rank($X$) | Lin. solves | CPU time |
|---|---|---|---|---|---|---|
| 10000 | 5 | 33 | 396 | 50 | 198 | 9.38 |
| 10000 | 10 | 27 | 594 | 48 | 297 | 19.87 |
| 10000 | 15 | 24 | 768 | 44 | 384 | 27.35 |
| 50000 | 5 | 55 | 660 | 43 | 330 | 54.87 |
| 50000 | 10 | 45 | 990 | 41 | 495 | 117.26 |
| 50000 | 15 | 40 | 1280 | 42 | 640 | 245.87 |
| 100000 | 5 | 68 | 816 | 43 | 408 | 133.72 |
| 100000 | 10 | 56 | 1232 | 41 | 616 | 332.68 |
| 100000 | 15 | 50 | 1600 | 44 | 800 | 743.86 |

Table B.3: Algorithm B.1 applied to (B.29) varying $n$ and $m$.

proach for matrix equation [16, 36, 37]. In this case we used the method presented in [16, Section 3].

### B.4.3 Inhomogeneous Helmholtz equation

In the last example, we analyze the complexity of Algorithm B.1 when solving a large-scale generalized Sylvester equation stemming from a finite difference discretization of a PDE. More precisely, we consider the following inhomogeneous Helmholtz equation

$$\begin{cases} -\Delta u(x,y) + \kappa(x,y)u(x,y) = f(x,y), \quad (x,y) \in [0,1] \times \mathbb{R}, \\ u(0,y) = u(1,y) = 0, \\ u(x,y+1) = u(x,y). \end{cases} \tag{B.30}$$

The boundary conditions are periodic in the $y$-direction and homogeneous-Dirichlet in the $x$-direction. The wavenumber $\kappa(x,y)$ and the forcing term $f(x,y)$ are 1-periodic functions in the $y$-direction. In particular they are respectively the periodic extensions of the scaled indicator functions $\chi_{[1/2,1]^2}$ and $100\chi_{[1/4,1/2]^2}$. The discretization of equation (B.30) with the finite difference method, using $n$ nodes multiple of 4, leads to the following generalized Sylvester equation

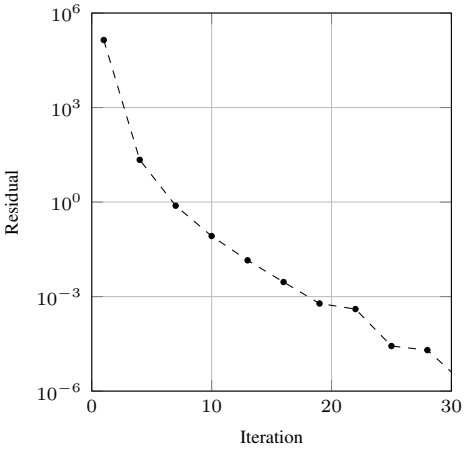
$$AX + XB^T + NXN^T = CC^T, \tag{B.31}$$

where $B = -\text{tridiag}(1,-2,1)/h^2$ and $A = B - \begin{bmatrix} e_1, e_n \end{bmatrix} \begin{bmatrix} e_n, e_1 \end{bmatrix}^T / h^2$, with $h = 1/(n-1)$ being the mesh-size, and

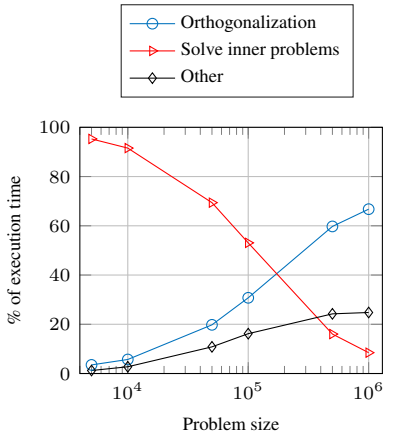
$$N = \begin{bmatrix} O_{n/2} & O_{n/2} \\ O_{n/2} & I_{n/2} \end{bmatrix} \in \mathbb{R}^{n \times n}, \quad C = \begin{bmatrix} c_1, \ldots, c_n \end{bmatrix}^T, \quad c_i = \begin{cases} 10, \text{ if } i \in [n/4, n/2], \\ 0, \text{ otherwise.} \end{cases}$$

A direct computation shows that $\text{com}(A,N) = U\tilde{U}^T$ and $\text{com}(B,N) = Q\tilde{Q}^T$ where

$$U = n \begin{bmatrix} e_{n/2+1}, e_{n/2}, e_1, e_n \end{bmatrix}, \qquad \tilde{U} = n \begin{bmatrix} e_{n/2}, -e_{n/2+1}, -e_n, e_1 \end{bmatrix},$$
$$Q = n \begin{bmatrix} e_{n/2+1}, e_{n/2} \end{bmatrix}, \qquad \tilde{Q} = n \begin{bmatrix} e_{n/2}, -e_{n/2+1} \end{bmatrix}.$$




(a) Residual norm history for problem of size $n = 10000$.




(b) CPU time (percent) of the main parts of Algorithm B.1 with $d = 30$. Total time (in seconds) are 23.71, 25.41, 34.33, 48.19, 163.34, and 315.74 respectively.

Figure B.2: Simulations for the inhomogeneous Helmholtz equation.

Algorithm B.1 is not applicable to equation (B.31) since the matrix $A$ is singular. However, in our approach it is possible to shift the Sylvester operator. In particular we can rewrite equation (B.31) as

$$(A + I)X + XB^T + NXN^T - X = CC^T.$$

It is now possible to apply Algorithm B.1 since $A + I$ is nonsingular. For this problem it holds $N^2 = N$ and then $\mathcal{G}_\ell(N, I; C) = \text{range}(\big[C, NC\big])$ for all $\ell \geq 1$. We note that $\text{com}(A + I, N) = \text{com}(A, N)$, and that $NC = 0$ and $\text{range}(\big[U, NU\big]) = \text{range}(U)$. Hence, according to Theorem B.3.6 we select $\bar{C}_1 = \big[C, U\big]$ and $\bar{C}_2 = \big[C, Q\big]$ as starting blocks. Notice that, with this choice, Algorithm B.1 provides an approximation of $X^{(\ell)}$ for every $\ell \geq 0$. We fix the number of iterations $d = 30$ in Algorithm B.1, and we vary the problem size $n$. In Figure B.2b we report the percentages of the overall execution time devoted to the orthogonalization procedure (Steps 9–11), to the solution of the inner problems (Step 4) and to the remaining steps of the algorithm. We can see that for very large problems, most of the computational effort is dedicated to the orthogonalization procedure. See Figure B.2a for an illustration of the converge history for the problem of size $n = 10000$.

## B.5 Conclusions and outlook

The method that we have proposed for solving (B.3) is directly based on the low-rank commutation feature of the matrix coefficients (B.4). We have applied and adapted our

procedure to problems in control theory and discretization of PDEs that naturally present this property. The structured matrices that present this feature are already analyzed in literature although, to our knowledge, this was never exploited in the setting of Krylov-like methods for matrix equations. Low-rank commuting matrices are usually studied with the *displacement operators*. More precisely, for a given matrix $Z$, the displacement operator is defined as $F(A) := AZ - ZA$. For many specific choices of the matrix $Z$, e.g., Jordan block, circulant, etc., it is possible to characterize the displacement operator and describe the matrices that are low-rank commuting with $Z$. See, e.g., [25, 6], [13, Chap. 2, Sec. 11] and references therein. The theory concerning the displacement operator may potentially be used to classify the problems that can be solved with our approach.

The approach we have pursued in this paper is based on the extended Krylov subspace method. However, it seems to be possible to extend this to the rational Krylov subspace method [17] since, the commutator $\mathrm{com}(A, N)$ is invariant under translations of the matrix $A$. Further research is needed to characterize the spaces and study efficient shift-selection strategies.

In each iteration of Algorithm B.1 the residual can be computed without explicitly constructing the current approximation of the solution but only using the solution of the projected problem. It may be possible to compute the residual norm even without explicitly solving the projected problems as proposed in [34] for Lyapunov and Sylvester equations with symmetric matrix coefficients.

In conclusion, we wish to point out that the low-rank approximability characterization may be of use outside of the scope of projection methods. For instance, the Riemannian optimization methods are designed to compute the best rank $k$ approximation (in the sense of, e.g., [28, 42]) to the solution of the matrix equation. This approach is effective only if $k$ is small, i.e., the solution is approximable by a low-rank matrix, for which we have provided sufficient conditions.

## Acknowledgment

We wish to thank Tobias Breiten (University of Graz) for kindly providing the code which helped us to implement BilADI [8] used in Section B.4. We also thank Stephen D. Shank (Temple University) for providing us the GLEK code before its on-line publication.

This research commenced during a visit of the third author to the KTH Royal Institute of Technology. The warm hospitality received is greatly appreciated. The work of the third author is partially supported by INdAM-GNCS under the 2017 Project "Metodi numerici avanzati per equazioni e funzioni di matrici con struttura". The other authors gratefully acknowledge the support of the Swedish Research Council under Grant No. 621-2013-4640.

## References

[1] 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.

[2] J. Ballani and L. Grasedyck. A projection method to solve linear systems in tensor format. *Numer. Linear Algebra Appl.*, 20(1):27–43, 2013.

[3] R. H. Bartels and G. W. Stewart. Algorithm 432: Solution of the matrix equation $AX + XB = C$. *Comm. ACM*, 15:820–826, 1972.

[4] U. Baur. Low rank solution of data-sparse Sylvester equations. *Numer. Linear Algebra Appl.*, 15(9):837–851, 2008.

[5] U. Baur and P. Benner. Factorized solution of Lyapunov equations based on hierarchical matrix arithmetic. *Computing*, 78(3):211–234, 2006.

[6] B. Beckermann and A. Townsend. On the singular values of matrices with displacement structure. *SIAM J. Matrix Anal. Appl.*, 38(4):1227–1248, 2017.

[7] P. Benner and T. Breiten. Interpolation-based $\mathcal{H}_2$-model reduction of bilinear control systems. *SIAM J. Matrix Anal. Appl.*, 33(3):859–885, 2012.

[8] 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.

[9] 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.

[10] P. Benner and P. Kürschner. Computing real low-rank solutions of Sylvester equations by the factored ADI method. *Comput. Math. Appl.*, 67(9):1656–1672, 2014.

[11] P. Benner, R. C. Li, and N. Truhar. On the ADI method for Sylvester equations. *J. Comput. Appl. Math.*, 233(4):1035–1045, 2009.

[12] J. Berstel and C. Reutenauer. *Noncommutative rational series with applications*, volume 137 of *Graduate Texts in Mathematics*. Cambridge Univ. Press, Cambridge, UK, 2011.

[13] D. Bini and V. Y. Pan. *Polynomial and matrix computations: fundamental algorithms*. Birkhäuser, Cambridge, MA, 1994.

[14] A. Bouhamidi and K. Jbilou. A note on the numerical approximate solutions for generalized Sylvester matrix equations with applications. *Appl. Math. Comput.*, 206(2):687–694, 2008.

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

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

[17] V. Druskin and V. Simoncini. Adaptive rational Krylov subspaces for large-scale dynamical systems. *Syst. Control Lett.*, 60(8):546–560, 2011.

[18] E. Einstein, C. R. Johnson, B. Lins, and I. Spitkovsky. The ratio field of values. *Linear Algebra Appl.*, 434(4):1119–1136, 2011.

[19] G. Flagg and S. Gugercin. Multipoint Volterra series interpolation and $\mathcal{H}_2$ optimal model reduction of bilinear systems. *SIAM J. Matrix Anal. Appl.*, 36(2):549–579, 2015.

[20] A. Frommer, K. Lund, and D. B. Szyld. Block Krylov subspace methods for functions of matrices. *Electron. Trans. Numer. Anal.*, 47:100–126, 2017.

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

[22] L. Grasedyck. Existence of a low rank or $\mathcal{H}$-matrix approximant to the solution of a Sylvester equation. *Numer. Linear Algebra Appl.*, 11(4):371–389, 2004.

[23] M. H. Gutknecht. Block Krylov space methods for linear systems with multiple right-hand sides: An introduction. In *Modern Mathematical Models, Methods and Algorithms for Real World Systems*, pages 420–447. Anamaya, 2007.

[24] I. M. Jaimoukha and E. M. Kasenally. Krylov subspace methods for solving large Lyapunov equations. *SIAM J. Numer. Anal.*, 31(1):227–251, 1994.

[25] T. Kailath and A. H. Sayed. Displacement structure: theory and applications. *SIAM Rev.*, 37(3):297–386, 1995.

[26] T. Kato. *Perturbation Theory for Linear Operators*. Classics in Mathematics. Springer-Verlag, Berlin Heidelberg, 2nd edition, 1995.

[27] D. Kressner and P. Sirković. Truncated low-rank methods for solving general linear matrix equations. *Numer. Linear Algebra Appl.*, 22(3):564–583, 2015.

[28] D. Kressner, M. Steinlechner, and B. Vandereycken. Preconditioned low–rank Riemannian optimization for linear systems with tensor product structure. *SIAM J. Sci. Comput.*, 38(4):A2018–A2044, 2016.

[29] D. Kressner and C. Tobler. Krylov subspace methods for linear systems with tensor product structure. *SIAM J. Matrix Anal. Appl.*, 31(4):1688–1714, 2010.

[30] P. Lancaster. Explicit solutions of linear matrix equations. *SIAM Rev.*, 12(4):544–566, 1970.

[31] Z. Y. Li, B. Zhou, Y. Wang, and G. R. Duan. Numerical solution to linear matrix equation by finite steps iteration. *IET Control Theory Appl.*, 31(1):227–251, 1994.

[32] 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.

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

[34] D. Palitta and V. Simoncini. Computationally enhanced projection methods for symmetric Sylvester and Lyapunov equations. *J. Comput. Appl. Math.*, 330:648–659, 2018.

[35] 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.

[36] S. Richter, L. D. Davis, and E. G. Collins Jr. Efficient computation of the solutions to modified Lyapunov equations. *SIAM J. Matrix Anal. Appl.*, 14(2):420–431, 1993.

[37] E. Ringh, G. Mele, J. Karlsson, and E. Jarlebring. Sylvester-based preconditioning for the waveguide eigenvalue problem. *Linear Algebra Appl.*, 542:441–463, 2018. Proceedings of the 20th ILAS Conference, Leuven, Belgium 2016.

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

[39] V. Simoncini. A new iterative method for solving large-scale Lyapunov matrix equations. *SIAM J. Sci. Comput.*, 29(3):1268–1288, 2007.

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

[41] V. Simoncini and L. Knizhnerman. A new investigation of the extended Krylov subspace method for matrix function evaluations. *Numer. Linear Algebra Appl.*, 17(4):615–638, 2010.

[42] B. Vandereycken and S. Vandewalle. A Riemannian optimization approach for computing low-rank solutions of Lyapunov equations. *SIAM J. Matrix Anal. Appl.*, 31(5):2553–2579, 2010.

[43] E. Wachspress. *The ADI model problem*. Springer-Verlag, New York, NY, 2013.

[44] L. Zhang and J. Lam. On $H_2$ model reduction of bilinear systems. *Automatica J. IFAC*, 38(2):205–216, 2002.

Paper C

# Residual-based iterations for the generalized Lyapunov equation

# Residual-based iterations for the generalized Lyapunov equation

by

Tobias Breiten, Emil Ringh

published in *BIT Numerical Mathematics*
Volume 59, Pages 823-852, December 2019

**Abstract**

This paper treats iterative solution methods for the generalized Lyapunov equation. Specifically, a residual-based generalized rational-Krylov-type subspace is proposed. Furthermore, the existing theoretical justification for the alternating linear scheme (ALS) is extended from the stable Lyapunov equation to the stable generalized Lyapunov equation. Further insights are gained by connecting the energy-norm minimization in ALS to the theory of $\mathcal{H}_2$-optimality of an associated bilinear control system. Moreover, it is shown that the ALS-based iteration can be understood as iteratively constructing rank-1 model reduction subspaces for bilinear control systems associated with the residual. Similar to the ALS-based iteration, the fixed-point iteration can also be seen as a residual-based method minimizing an upper bound of the associated energy norm.

**Keywords:** Generalized Lyapunov equation, $\mathcal{H}_2$-optimal model reduction, bilinear control systems, alternating linear scheme, projection methods, matrix equations, rational Krylov

## C.1 Introduction

This paper concerns iterative ways to compute approximate solutions to what has become known as the *generalized Lyapunov equation*,

$$\mathscr{L}(X) + \Pi(X) + BB^T = 0, \tag{C.1}$$

where $X \in \mathbb{R}^{n\times n}$ is unknown, $B \in \mathbb{R}^{n\times r}$ is given, and the operators $\mathscr{L}, \Pi : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$ are defined as

$$\mathscr{L}(X) := AX + XA^T \tag{C.2}$$

$$\Pi(X) := \sum_{i=1}^{m} N_i X N_i^T, \tag{C.3}$$

with $A, N_i \in \mathbb{R}^{n\times n}$ for $i = 1, \ldots, m$ given. The operator $\mathscr{L}$ is commonly known as the *Lyapunov operator*, and $\Pi$ is sometimes called a *correction*. We further assume that $A$ is *stable*, i.e., $A$ has all its eigenvalues in the left-half plane, which implies that $\mathscr{L}$ is invertible [23, Theorem 4.4.6]. Moreover, we assume that $\rho(\mathscr{L}^{-1}\,\Pi) < 1$, where $\rho$ denotes the (operator) spectral radius. The assumption on the spectral radius implies that (C.1) has a unique solution [24, Theorem 2.1]. Furthermore, the definition of $\Pi$ in (C.3) implies that it is non-negative, in the sense that $\Pi(X)$ is positive semidefinite when $X$ is positive semidefinite. Thus, one can assert that, for all positive definite right-hand sides, the unique solution $X$ is indeed positive definite [9, Theorem 3.9][12, Theorem 4.1]. Under these assumptions we prove that the alternating linear scheme (ALS) presented by Kressner and Sirković in [25] computes search directions which at each step fulfill a first order necessary condition for being $\mathcal{H}_2$-optimal. Moreover, we show an equivalence between the bilinear iterative rational Krylov (BIRKA) method [5, 19] and the ALS-iteration for the generalized Lyapunov equation. The established equivalence leads to that the ALS-iteration for the generalized Lyapunov equation can be understood as iteratively computing model reduction spaces of dimension 1 for a sequence of bilinear control systems associated with the residual of the generalized Lyapunov equation (Section C.3). We also present a residual-based generalized rational-Krylov-type subspace adapted for solving the generalized Lyapunov equation (Section C.5). A further result regards the fixed-point iteration, a residual-based iteration which we show minimizes an upper bound of the energy norm (Section C.4).

The standard Lyapunov equation, $AX + XA^T + BB^T = 0$, has been well studied for a long time and considerable research effort has been, and is still, put into finding efficient algorithms for computing the solution and approximations thereof. For large and sparse problems it is typical to look for low-rank approximations since algorithms can be adapted to exploit the low-rank format, reducing computational effort and storage requirement. One such algorithm is the Riemannian optimization method from [40] which computes a low-rank approximation by minimizing an associated cost function over the manifold of rank-$k$ matrices, where $k \ll n$. The Lyapunov equation has a close connection to control theory. Hence, methods such as the iterative rational Krylov algorithm (IRKA) [21, 18], which computes subspaces for locally $\mathcal{H}_2$-optimal reduced order linear systems, provide good approximation spaces for low-rank approximations. Related research is presented in a series of papers [13, 14, 15], where Druskin and co-authors develop a strategy to choose shifts for the rational Krylov subspace for efficient subspace reduction when solving PDEs [13, 14], as well as for model reduction of linear single-input-single-output (SISO) systems and solutions to Lyapunov equations [15]. Instead of computing full spaces iteratively with a method such as IRKA, the idea is to construct an infinite sequence with asymptotically optimal convergence speed [13]. Then the subspace can be dynamically extended as needed, until required precision is achieved. The idea is also further developed by using tangential directions, proving especially useful for situations where the right-hand side is not of particularly low rank [16], e.g., multiple-input-multiple-output (MIMO) systems. For a more complete overview of results and techniques for Lyapunov equations see the review article [38].

The generalized Lyapunov equation has received increased attention over the past decade. Results on low-rank approximability have emerged [6, 24]. More precisely, sim-

ilarly to the standard Lyapunov equation one can in certain cases when the right-hand side $B$ is of low rank, $r \ll n$, expect the singular values of the solution to decay rapidly even for the generalized Lyapunov equation. The result [6, Theorem 1] is applicable when the matrices $N_i$ for $i = 1, \ldots, m$ have low rank, and the result [24, Theorem 2] when $\rho(\mathscr{L}^{-1}\,\Pi) < 1$. Examples of algorithms exploiting low-rank structures are a Bilinear ADI method [6], specializations of Krylov methods for matrix equations [24], as well as greedy low-rank methods [25], and exploitations of the fixed-point iteration [37]. Through the connection with bilinear control systems there is an extension of IRKA, known as bilinear iterative rational Krylov (BIRKA) [5, 19]. There are also methods based on Lyapunov and ADI-preconditioned GMRES and BiCGStab [12], and in general for problems with tensor product structure [26]. In the context of stochastic steady-state diffusion equations, rational Krylov subspace methods for generalized Sylvester equations have also been analyzed in [32]. The suggested search space is based on a union of rational Krylov subspaces, as well as combinations of rational functions, generated by the coefficient matrices defining the generalized Sylvester operator. We also mention that for the case when the correction $\Pi$ has low operator-rank, there is a specialization of the Sherman–Morrison–Woodbury formula to the linear matrix equation; see [33] or [12, Section 3]. The result has been exploited in works such as [6, 34, 28]. Recently, the generalized Lyapunov equation has also been considered on an infinite-dimensional Hilbert space, see [4]. In particular, the authors show ([4, Proposition 1.1]) that the Gramians solving the generalized linear operator equations can be approximated by truncated Gramians that are associated to a sequence of standard operator Lyapunov equations.

## C.2 Preliminaries

### C.2.1 Generalized matrix equations and approximations

We recall some basic definitions and results that will be used later in the paper. In general we will think of $\hat{X}_k \in \mathbb{R}^{n\times n}$ as an approximation of the solution to (C.1), where $k$ is typically an iteration count. Connected with an approximation $\hat{X}_k$ is the corresponding *error*

$$X_k^{\mathrm{e}} := X - \hat{X}_k, \tag{C.4}$$

where $X$ is the exact solution to (C.1), and the *residual*,

$$\mathcal{R}_k := \mathscr{L}(\hat{X}_k) + \Pi(\hat{X}_k) + BB^T. \tag{C.5}$$

The goal is to find an $\hat{X}_k$ such that $\|X_k^{\mathrm{e}}\|$ is small for some norm. Since $\|X_k^{\mathrm{e}}\|$ is usually not available in practice, one instead aims at a small residual norm $\|\mathcal{R}_k\|$. To discuss projection methods and make the results precise, we make the following (standard) definition.

**Definition C.2.1** (The Galerkin approximation)**.** Let $\mathcal{K}_k \subseteq \mathbb{R}^n$ be an $n_k \leq n$ dimensional subspace for $k = 0, 1, \ldots$, and let $\mathcal{V}_k \in \mathbb{R}^{n\times n_k}$ be a matrix containing an orthogonal basis

of $\mathcal{K}_k$. We call $\hat{X}_k$ the *Galerkin approximation* to (C.1), in $\mathcal{K}_k$, if $\hat{X}_k = \mathcal{V}_k\, Y_k\, \mathcal{V}_k^T$ and $Y_k$ is determined by the condition

$$\mathcal{V}_k^T \left( \mathscr{L}(\hat{X}_k) + \Pi(\hat{X}_k) + BB^T \right) \mathcal{V}_k = 0. \tag{C.6}$$

For the generalized Lyapunov equation there are certain sufficient conditions for the Galerkin approximation to exist and be unique, e.g., the criteria in [9, Theorem 3.9],[12, Theorem 4.1] or [24, Proposition 3.2]. Related to the Galerkin approximation there is also the (standard) definition of the Galerkin residual.

**Definition C.2.2** (The Galerkin residual)**.** We call $\mathcal{R}_k$ from (C.5) the *Galerkin residual* if $\hat{X}_k$ is the Galerkin approximation.

The condition (C.6) is known as both the *projected problem* and the *Galerkin condition*, and it states that $\mathcal{V}_k^T\, \mathcal{R}_k\, \mathcal{V}_k = 0$ for the Galerkin residual. Some of the results and arguments presented below are valid for a (generic) residual and others, more specialized, only for the Galerkin residual. However, it will be clear from context and the Galerkin residual will always be referenced as such.

The following fundamental result from linear algebra will be important for us. The specialization for the Lyapunov equation was presented already by Smith in [39]. For generalized matrix equations cf. [12, Section 4.2], and [25, Algorithm 2]; and an analogy for the algebraic Riccati equation in [29].

**Proposition C.2.3** (Residual equation)**.** *Consider equation* (C.1)*. Let $\hat{X}_k$ be an approximation of the solution, $\mathcal{R}_k$ be the residual* (C.5)*, and $X_k^e$ be the error* (C.4)*. Then*

$$\mathscr{L}(X_k^e) + \Pi(X_k^e) + \mathcal{R}_k = 0.$$

One strategy for computing updates to the current approximation is to compute approximations of the error. Proposition C.2.3 allows such iterations by connecting the error with the known, or computable, quantities $\mathscr{L}$, $\Pi$ and $\mathcal{R}_k$. The idea is well established in the literature and is, e.g., analogous to the defect correction method [29] and the RADI method [8] for the algebraic Riccati equation, as well as the iterative improvement [20, Section 3.5.3] for a general linear system. For future reference we also need the following basic definition.

**Definition C.2.4** (Symmetric generalized Lyapunov equation)**.** The generalized Lyapunov equation

$$AX + XA^T + \sum_{i=1}^{m} N_i X N_i^T + BB^T = 0, \tag{C.7}$$

is called *symmetric* if $A = A^T$ and $N_i = N_i^T$ for $i = 1, \ldots, m$.

### C.2.2 Bilinear systems

We recall some control theoretic concepts for *bilinear control systems* of the form

$$\Sigma \begin{cases} \dot{x}(t) = Ax(t) + \sum_{i=1}^{m} N_i x(t) w_i(t) + Bu(t) \\ y(t) = Cx(t), \end{cases} \tag{C.8}$$

with $A, N_i \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times r}$ and $C \in \mathbb{R}^{p\times n}$ and control inputs $u(t) \in \mathbb{R}^r$ and $w(t) \in \mathbb{R}^m$.

**Remark C.2.5.** *Note that the bilinear system* (C.8) *differs from the notation frequently used in the literature, e.g., [1, 2, 5, 9, 12, 19, 41]. The formulation* (C.8) *is convenient since it allows for $m \neq r$. However, the system $\Sigma$ can be put into the usual form by considering the input vector $\begin{bmatrix} w(t)^T, & u(t)^T \end{bmatrix}^T$, adding $m$ zero-columns to the beginning of $B$, i.e., $\begin{bmatrix} 0, & B \end{bmatrix}$, and considering the matrices $N_{m+1} = 0, \ldots, N_{m+r} = 0$. The system $\Sigma$ can also be compared to systems from applications, e.g., [30, Equation (2)].*

As in [2], for a MIMO bilinear system (C.8), we define the controllability and observability Gramians as follows.

**Definition C.2.6** (Bilinear Gramians [2])**.** Consider the bilinear system (C.8) with $A$ stable. Moreover, let $P_1(t_1) := e^{At_1}B$, $P_j(t_1, \ldots, t_j) := e^{At_j}\begin{bmatrix} N_1 P_{j-1}, \ldots, N_m P_{j-1} \end{bmatrix}$ for $j = 2, 3, \ldots$, $Q_1(t_1) := Ce^{At_1}$, and $Q_j(t_1, \ldots, t_j) := \begin{bmatrix} N_1^T Q_{j-1}^T, \ldots, N_m^T Q_{j-1}^T \end{bmatrix}^T e^{At_j}$ for $j = 2, 3, \ldots$. We define the controllability and observability Gramian respectively as

$$P := \sum_{j=1}^{\infty} \int_0^{\infty} \cdots \int_0^{\infty} P_j P_j^T dt_1 \ldots dt_j$$
$$Q := \sum_{j=1}^{\infty} \int_0^{\infty} \cdots \int_0^{\infty} Q_j^T Q_j dt_1 \ldots dt_j.$$

It is possible that the generalized Gramians from Definition C.2.6 do not exist; sufficient conditions are given in, e.g., [41, Theorem 2]. However, if the Gramians exist we also know that they satisfy the following matrix equations

$$\begin{aligned} AP + PA^T + \sum_{i=1}^{m} N_i P N_i^T + BB^T &= 0 \\ A^T Q + QA + \sum_{i=1}^{m} N_i^T Q N_i + C^T C &= 0. \end{aligned} \tag{C.9}$$

In relation to the generalized controllability and observability Gramians, one might also define a generalized cross Gramian similar to the SISO case discussed in [36]. Consider

the symmetric generalized Lyapunov equation (C.7), and an approximation $\hat{X}_k$ with related error $X_k^{\mathrm{e}}$, and residual $\mathcal{R}_k$. One can easily verify that for the auxiliary system

$$\Sigma^{\mathrm{e}} = \begin{cases} \dot{x}(t) = Ax(t) + \sum_{i=1}^{m} N_i x(t) w_i(t) + B_{\mathcal{R}_k} u(t) \\ y(t) = C_{\mathcal{R}_k} x(t), \end{cases}$$

with $B_{\mathcal{R}_k} = US^{1/2}$ and $C_{\mathcal{R}_k} = S^{1/2}V^T$, where $\mathcal{R}_k = USV^T$ is a singular value decomposition of $\mathcal{R}_k$, the associated cross Gramian coincides with the error $X_k^{\mathrm{e}}$. In the special case where $\mathcal{R}_k = \mathcal{R}_k^T \succeq 0$, it is easy to show the following result.

**Proposition C.2.7.** *Consider the symmetric generalized Lyapunov equation* (C.7)*. Let $\hat{X}_k$ be an approximation such that the residual $\mathcal{R}_k = \mathcal{R}_k^T \succeq 0$. Then one can choose $B_{\mathcal{R}_k} = C_{\mathcal{R}_k}^T$ and the error $X_k^e$ is the controllability and observability Gramian of the system $\Sigma^e$.*

For what follows, we recall the definition of the $\mathcal{H}_2$-norm for bilinear systems that was introduced by Zhang and Lam in [41].

**Definition C.2.8** ([41], Bilinear $\mathcal{H}_2$-norm)**.** Consider the bilinear system $\Sigma$ from (C.8). We define the $\mathcal{H}_2$-norm of $\Sigma$ as

$$\|\Sigma\|_{\mathcal{H}_2}^2 := \operatorname{Tr}\left( \sum_{j=1}^{\infty} \int_0^{\infty} \cdots \int_0^{\infty} \sum_{\ell_1,\cdots,\ell_{j-1}=1}^{m} \sum_{\ell_j=1}^{r} g_j^{(\ell_1,\ldots,\ell_j)} (g_j^{(\ell_1,\ldots,\ell_j)})^T \mathrm{d}s_1 \cdots \mathrm{d}s_j \right),$$

with $g_j^{(\ell_1,\ldots,\ell_j)}(s_1,\ldots,s_j) := Ce^{As_j} N_{\ell_1} e^{As_{j-1}} N_{\ell_2} \cdots e^{As_1} b_{\ell_j}$.

It has been shown [41, Theorem 6] that if the Gramians from Definition C.2.6 exist, then

$$\|\Sigma\|_{\mathcal{H}_2}^2 = \operatorname{Tr}\left(CPC^T\right) = \operatorname{Tr}\left(B^T Q B\right).$$

## C.3 ALS and $\mathcal{H}_2$-optimal model reduction for bilinear systems

In this section, we discuss a low-rank approximation method proposed by Kressner and Sirković in [25]. We show that several results can be generalized from the case of the standard Lyapunov equation to the more general form (C.1). Moreover, we show that in the symmetric case the method allows for an interpretation in terms of $\mathcal{H}_2$-optimal model reduction for bilinear control systems. With this in mind, we assume that we have a symmetric generalized Lyapunov equation (C.7). If additionally $A \prec 0$ and $\rho(\mathscr{L}^{-1}\,\Pi) < 1$, then the operator $\mathcal{M}(X) := -\mathscr{L}(X) - \Pi(X)$ is positive definite and allows us to define a weighted inner product via

$$\langle \cdot, \cdot \rangle_{\mathcal{M}} \colon \mathbb{R}^{n\times n} \times \mathbb{R}^{n\times n} \to \mathbb{R}$$
$$\langle X, Y \rangle_{\mathcal{M}} = \langle X, \mathcal{M}(Y) \rangle = \operatorname{Tr}\left(X^T \mathcal{M}(Y)\right),$$

with a corresponding induced $\mathcal{M}$-norm, also known as *energy norm*,

$$\|X\|_{\mathcal{M}}^2 = \langle X, X\rangle_{\mathcal{M}}.$$

### C.3.1 ALS for the generalized Lyapunov equation

In [25], it is suggested to construct iterative approximations $\hat{X}_k$ by rank-1 updates that are locally optimal with respect to the $\mathcal{M}$-norm. To be more precise, assume that $X$ is a solution to the symmetric Lyapunov equation (C.7), i.e., $AX + XA + \sum_{i=1}^m N_i X N_i + BB^T = 0$. Given an approximation $\hat{X}_k$, we consider the minimization problem

$$\min_{v,w\in\mathbb{R}^n} \|X - \hat{X}_k - vw^T\|_{\mathcal{M}}^2 = \langle X - \hat{X}_k - vw^T, X - \hat{X}_k - vw^T\rangle_{\mathcal{M}}.$$

Since the minimization involves the constant term $\|X - \hat{X}_k\|_{\mathcal{M}}^2$, it suffices to focus on

$$J(v,w) := \langle vw^T, vw^T\rangle_{\mathcal{M}} - 2\,\mathrm{Tr}\left(wv^T\mathcal{R}_k\right), \tag{C.10}$$

where $\mathcal{R}_k$ is the current residual, i.e., (C.5). Locally optimal vectors $v_k$ and $w_k$ are then (approximately) determined via an *alternating linear scheme* (ALS). The main step is to fix one of the two vectors, e.g., $v$ and then minimize the strictly convex objective function to obtain an update for $w$. A pseudocode is given in Algorithm C.1.

**Algorithm C.1:** ALS for the generalized Lyapunov equation [25, Algorithm 1]

**input :** System: $A, N_1, \ldots, N_m \in \mathbb{R}^{n\times n}$ $\mathcal{R}_k \in \mathbb{R}^{n\times n}$, `tol`
Initial guess: $v, w \in \mathbb{R}^n$
**output:** Approximation vectors: $v^{\mathrm{a}}, w^{\mathrm{a}} \in \mathbb{R}^n$

**while** *Change in* $(\frac{v^TAv}{\|v\|^2} + \frac{w^TA^Tw}{\|w\|^2})/2$ *larger than* `tol` **do**

1. Normalize $w = w/\|w\|$
2. $\hat{A}_1 = A + I(w^TAw) + \sum_{i=1}^m N_i(w^TN_iw)$
3. Solve $\hat{A}_1 v = -\mathcal{R}_k w$
4. Normalize $v = v/\|v\|$
5. $\hat{A}_2 = A + I(v^TAv) + \sum_{i=1}^m (v^TN_iv)N_i$
6. Solve $\hat{A}_2 w = -\mathcal{R}_k^T v$

7. Normalize such that $\|w\| = \|v\|$

**return** $v^{\mathrm{a}} = v$, $w^{\mathrm{a}} = w$

In view of Proposition C.2.3 the ALS-based approach for computing new subspace extensions can be seen as searching for an approximation to $X_k^{\mathrm{e}}$ of the form $v_k w_k^T$ by iterating $(\mathscr{L}(v_kw_k^T) + \Pi(v_kw_k^T) + \mathcal{R}_k)w_k = 0$ when determining $v_k$ and $v_k^T(\mathscr{L}(v_kw_k^T) + \Pi(v_kw_k^T) + \mathcal{R}_k) = 0$ when determining $w_k$. This is to say that the error is approximated by a rank-1 matrix, and at convergence this would result in the new residual, $\mathcal{R}_{k+1}$, being left-orthogonal to $v_k$ and right-orthogonal to $w_k$. In the symmetric case, local minimizers of

(C.10) are necessarily symmetric positive semidefinite. This yields the following extension of [25, Lemma 2.3].

**Lemma C.3.1.** *Consider the symmetric generalized Lyapunov equation* (C.7) *and assume that* $A \prec 0$*,* $\rho(\mathscr{L}^{-1}\,\Pi) < 1$*, and* $\mathcal{R}_k = \mathcal{R}_k^T \succeq 0$*. Let* $J$ *be as in* (C.10)*. Then every local minimum* $(v_*, w_*)$ *of* $J$ *is such that* $v_* w_*^T$ *is symmetric positive semidefinite.*

*Proof.* The proof naturally follows along the lines of [25, Lemma 2.3], and hence without loss of generality we assume that $v_* \neq 0$ , $w_* \neq 0$, and $\|v_*\| = \|w_*\|$. Thus, $v_* w_*^T$ is positive semidefinite if and only if $v_* = w_*$. The proof is by contradiction and we assume that $v_* \neq w_*$. Then, since $J(v_*, w)$ is strictly convex in $w$ and $J(v, w_*)$ is strictly convex in $v$, it follows that

$$2J(v_*, w_*) < J(v_*, v_*) + J(w_*, w_*).$$

Simplifying the left-hand side we get

$$2J(v_*, w_*) = -2v_*^T\,\mathscr{L}(v_* w_*^T)w_* - 2v_*^T\Pi(v_* w_*^T)w_* - 4v_*^T\mathcal{R}_k w_*,$$

and similarly the right-hand side gives

$$\begin{aligned} J(v_*, v_*) + J(w_*, w_*) = &- v_*^T\,\mathscr{L}(v_* v_*^T)v_* - v_*^T\Pi(v_* v_*^T)v_* - 2v_*^T\mathcal{R}_k v_* \\ &- w_*^T\,\mathscr{L}(w_* w_*^T)w_* - w_*^T\Pi(w_* w_*^T)w_* - 2w_*^T\mathcal{R}_k w_*. \end{aligned}$$

Collecting the terms involving the $\mathscr{L}$-operator we observe that

$$\begin{aligned} -2v_*^T\,\mathscr{L}(v_* w_*^T)w_* + v_*^T\,\mathscr{L}(v_* v_*^T)v_* + w_*^T\,\mathscr{L}(w_* w_*^T)w_* = \\ 2(v_*^T v_*)(w_*^T A w_* - v_*^T A v_*) + 2w_*^T w_*(v_*^T A v_* - w_*^T A w_*) = 0, \end{aligned}$$

Thus, by collecting the terms involving the $\Pi$-operator to the left, and the residual to the right, the inequality reduces to

$$-2v_*^T\Pi(v_* w_*^T)w_* + v_*^T\Pi(v_* v_*^T)v_* + w_*^T\Pi(w_* w_*^T)w_* < -2(v_* - w_*)^T\mathcal{R}_k(v_* - w_*).$$

The argument is now concluded by showing that

$$-2v_*^T\Pi(v_* w_*^T)w_* + v_*^T\Pi(v_* v_*^T)v_* + w_*^T\Pi(w_* w_*^T)w_* \geq 0,$$

since this implies that $-2(v_* - w_*)^T\mathcal{R}_k(v_* - w_*) > 0$ in contradiction to the positive semidefiniteness of $\mathcal{R}_k$. We can without loss of generality consider $m = 1$, i.e., only one $N$-matrix, since the following argument can be applied to all terms in the sum independently. We observe that

$$-2v_*^T N v_* w_*^T N w_* + v_*^T N v_* v_*^T N v_* + w_*^T N w_* w_*^T N w_* = (v_*^T N v_* - w_*^T N w_*)^2 \geq 0,$$

which shows the desired inequality and thus concludes the proof. □

Algorithm C.1 and the argument in Lemma C.3.1 are based on a residual. However, if $\hat{X}_k = 0$, then $\mathcal{R}_k = BB^T$, and hence the result is applicable directly to any symmetric generalized Lyapunov equation. The focus on the residual in the previous results is natural since it leads to the following extension of [25, Theorem 2.4] to the case of the symmetric generalized Lyapunov equation.

**Theorem C.3.2.** *Consider the symmetric generalized Lyapunov equation* (C.7) *with the additional assumptions that* $A \prec 0$ *and* $\rho(\mathscr{L}^{-1}\,\Pi) < 1$. *Moreover, consider the sequence of approximations constructed as*

$$\begin{aligned} \hat{X}_0 &= 0 \\ \hat{X}_{k+1} &= \hat{X}_k + v_{k+1}v_{k+1}^T, \qquad k = 0, 1, \ldots, \end{aligned} \tag{C.11}$$

*where* $v_{k+1}$ *is a locally optimal vector computed with ALS (Algorithm C.1). Then* $\mathcal{R}_{k+1} = \mathcal{R}_{k+1}^T \succeq 0$ *for all* $k \geq -1$.

*Proof.* We show the assertion by induction. It clearly holds that $\mathcal{R}_0 = \mathcal{R}_0^T \succeq 0$. Now assume that this is the case for some $k$. From Lemma C.3.1 the local minimizers of (C.10) are symmetric and hence $\hat{X}_{k+1}$ is reasonably defined in (C.11). Moreover, since $\hat{X}_{k+1}$ and the operators in (C.1) are symmetric it follows that $\mathcal{R}_{k+1}$ is symmetric. Thus, what is left to show is that $\mathcal{R}_{k+1} \succeq 0$, which is true if and only if $y^T\mathcal{R}_{k+1}y \geq 0$ for all $y \in \mathbb{R}^n$. Hence, take an arbitrary $y \in \mathbb{R}^n$ and consider $y^T\mathcal{R}_{k+1}y$. We derive properties similar to [25, equations (12)-(14)]:

Since $(v_{k+1}, v_{k+1})$ is a (local) minimizer of $J(v, w)$, it also follows that $v_{k+1}$ is a (global) minimizer of the (convex) cost function

$$J_w(v) := J(v, w) = \langle vw^T, vw^T\rangle_{\mathcal{M}} - 2\,\mathrm{Tr}(wv^T\mathcal{R}_k),$$

where $w = v_{k+1}$. Note that the gradient $\nabla_v J_w$ of $J_w$ with respect to $v$ is given by

$$(\nabla_v J_w)_i = 2\langle e_i w^T, vw^T\rangle_{\mathcal{M}} - 2e_i^T\mathcal{R}_k w.$$

Due to the optimality of $v_{k+1}$ with respect to $J_{v_{k+1}}$, first order optimality conditions then imply that

$$-Av_{k+1}v_{k+1}^Tv_{k+1} - v_{k+1}v_{k+1}^TAv_{k+1} - \sum_{i=1}^m N_iv_{k+1}v_{k+1}^TN_iv_{k+1} = \mathcal{R}_kv_{k+1}. \tag{C.12}$$

Striking this equality with $v_{k+1}^T$ from the left implies that

$$2v_{k+1}^TAv_{k+1}\|v_{k+1}\|^2 = -v_{k+1}^T\mathcal{R}_kv_{k+1} - \sum_{i=1}^m (v_{k+1}^TN_iv_{k+1})^2. \tag{C.13}$$

Based on (C.12) and its transpose, and by exploiting the symmetry of the involved matrices, we can write the residual as

$$\begin{aligned} y^T\mathcal{R}_{k+1}y &= y^T\mathcal{R}_k y + y^T\left(Av_{k+1}v_{k+1}^T + v_{k+1}v_{k+1}^TA + \sum_{i=1}^m N_iv_{k+1}v_{k+1}^TN_i\right)y \\ &= y^T\mathcal{R}_k y + \sum_{i=1}^m y^TN_iv_{k+1}v_{k+1}^TN_iy + \frac{1}{\|v_{k+1}\|^2}y^T(U_{k+1}+U_{k+1}^T)y, \end{aligned}$$

with $U_{k+1} := -\mathcal{R}_k v_{k+1}v_{k+1}^T - (v_{k+1}^TAv_{k+1})v_{k+1}v_{k+1}^T - \sum_{i=1}^m N_iv_{k+1}z_{i,k+1}^T$, and where $z_{i,k+1} := (v_{k+1}^TN_iv_{k+1})v_{k+1}$. We rearrange, identify the term $-2(v_{k+1}^TAv_{k+1})v_{k+1}v_{k+1}^T$ and insert (C.13) to get

$$\begin{aligned} y^T\mathcal{R}_{k+1}y &= y^T\mathcal{R}_k y \\ &\quad + \frac{1}{\|v_{k+1}\|^2}y^T\left(-\mathcal{R}_kv_{k+1}v_{k+1}^T - v_{k+1}v_{k+1}^T\mathcal{R}_k + \frac{1}{\|v_{k+1}\|^2}v_{k+1}^T\mathcal{R}_kv_{k+1}v_{k+1}v_{k+1}^T\right)y \\ &\quad + \frac{1}{\|v_{k+1}\|^2}y^T\left(\sum_{i=1}^m N_iv_{k+1}v_{k+1}^TN_i\|v_{k+1}\|^2 + \frac{1}{\|v_{k+1}\|^2}\sum_{i=1}^m z_{i,k+1}z_{i,k+1}^T\right)y \\ &\quad + \frac{1}{\|v_{k+1}\|^2}y^T\left(-\sum_{i=1}^m N_iv_{k+1}z_{i,k+1}^T - \sum_{i=1}^m z_{i,k+1}v_{k+1}^TN_i\right)y \\ &= y^T\mathcal{R}_k y + \frac{1}{\|v_{k+1}\|^2}\left(-2(y^T\mathcal{R}_kv_{k+1})(v_{k+1}^Ty) + \frac{1}{\|v_{k+1}\|^2}(v_{k+1}^T\mathcal{R}_kv_{k+1})(v_{k+1}^Ty)^2\right) \\ &\quad + \frac{1}{\|v_{k+1}\|^2}\Big(\sum_{i=1}^m(y^TN_iv_{k+1})^2\|v_{k+1}\|^2 + \frac{1}{\|v_{k+1}\|^2}(z_{i,k+1}^Ty)^2 - 2(y^TN_iv_{k+1})(z_{i,k+1}^Ty)\Big) \\ &= (y - v_{k+1}\frac{v_{k+1}^Ty}{\|v_{k+1}\|^2})^T\mathcal{R}_k(y - v_{k+1}\frac{v_{k+1}^Ty}{\|v_{k+1}\|^2}) \\ &\quad + \frac{1}{\|v_{k+1}\|^2}\sum_{i=1}^m\left(\|v_{k+1}\|(y^TN_iv_{k+1}) - \frac{1}{\|v_{k+1}\|}(z_{i,k+1}^Ty)\right)^2 \geq 0. \end{aligned}$$

This asserts the inductive step and hence concludes the proof. □

**Corollary C.3.3.** *The iteration* (C.11) *produces an increasing sequence of approximations* $0 = \hat{X}_0 \preceq \hat{X}_1 \preceq \cdots \preceq X$.

### C.3.2 $\mathcal{H}_2$-optimal model reduction for symmetric state space systems

For the standard Lyapunov equation it has been shown, in [7], that minimization of the energy norm induced by the Lyapunov operator (see [40]) is related to $\mathcal{H}_2$-optimal model reduction for linear control systems. We show that a similar conclusion can be drawn for the minimization of the cost functional (C.10) and $\mathcal{H}_2$-optimal model reduction for

symmetric bilinear control systems. In this regard, let us briefly summarize the most important concepts from bilinear model reduction. Given a bilinear system $\Sigma$ as in (C.8) with $\dim(\Sigma) = n$, the goal of model reduction is to construct a surrogate model $\widehat{\Sigma}$ of the form

$$\widehat{\Sigma}\colon \begin{cases} \dot{\widehat{x}}(t) = \hat{A}\widehat{x}(t) + \sum_{i=1}^{m} \hat{N}_i \widehat{x}(t) w_i(t) + \hat{B}u(t) \\ \widehat{y}(t) = \hat{C}\widehat{x}(t), \end{cases} \tag{C.14}$$

with $\hat{A}, \hat{N}_i \in \mathbb{R}^{k\times k}, \hat{B} \in \mathbb{R}^{k\times r}, \hat{C} \in \mathbb{R}^{r\times k}$ and control inputs $u(t) \in \mathbb{R}^r$ and $w(t) \in \mathbb{R}^m$. In particular, the reduced system should satisfy $k \ll n$ and $\widehat{y}(t) \approx y(t)$ in some norm. In [5, 19] the authors have suggested an algorithm, BIRKA, that iteratively tries to compute a reduced model satisfying first order necessary conditions for $\mathcal{H}_2$-optimality, for the bilinear $\mathcal{H}_2$-norm as defined in Definition C.2.8. A corresponding pseudocode is given in Algorithm C.2.

**Algorithm C.2:** BIRKA [5, Algorithm 2] and [19, Algorithm 5]

**input :** System: $A, N_1, \ldots, N_m \in \mathbb{R}^{n\times n}$ $B \in \mathbb{R}^{n\times r}$, $C \in \mathbb{R}^{r\times n}$, `tol`
Initial guess: $\tilde{V}, \tilde{W} \in \mathbb{R}^{n\times k}$

**output:** Approximation spaces: $\tilde{V}^{\text{opt}}, \tilde{W}^{\text{opt}} \in \mathbb{R}^{n\times k}$ satisfying necessary conditions for $\mathcal{H}_2$-optimality

**while** *Change in eigenvalues of* $\tilde{A}$ *larger than* `tol` **do**

1. Update guess $\tilde{A} = (\tilde{W}^T\tilde{V})^{-1}\tilde{W}^T A\tilde{V}$, $\tilde{N}_1 = (\tilde{W}^T\tilde{V})^{-1}\tilde{W}^T N_1\tilde{V}, \ldots,$ $\tilde{N}_m = (\tilde{W}^T\tilde{V})^{-1}\tilde{W}^T N_m\tilde{V}$, $\tilde{B} = (\tilde{W}^T\tilde{V})^{-1}\tilde{W}^T B$, $\tilde{C} = C\tilde{V}$
2. Decompose $R\tilde{\Lambda}R^{-1} = \tilde{A}$
3. Compute $\hat{B} = R^{-1}\tilde{B}$, $\hat{C} = \tilde{C}R$, $\hat{N}_1 = R^{-1}\tilde{N}_1 R, \ldots, \hat{N}_m = R^{-1}\tilde{N}_m R$
4. Solve $\tilde{V}\tilde{\Lambda} + A\tilde{V} + \sum_{i=1}^m N_i\tilde{V}\hat{N}_i^T + B\hat{B}^T = 0$ for $\tilde{V}$
5. Solve $\tilde{W}\tilde{\Lambda} + A^T\tilde{W} + \sum_{i=1}^m N_i^T\tilde{W}\hat{N}_i + C^T\hat{C} = 0$ for $\tilde{W}$
6. Orthogonalize $\tilde{V} = \texttt{orth}(\tilde{V})$, $\tilde{W} = \texttt{orth}(\tilde{W})$

**return** $\tilde{V}^{\text{opt}} = \tilde{V}$, $\tilde{W}^{\text{opt}} = \tilde{W}$

To establish the connection we introduce the following generalizations of the operator $\mathcal{M}$:

$$\widetilde{\mathcal{M}}\colon \mathbb{R}^{n\times k} \to \mathbb{R}^{n\times k}, \quad \widetilde{\mathcal{M}}(X) := -AX - X\hat{A}^T - \sum_{i=1}^m N_i X \hat{N}_i^T,$$

$$\widehat{\mathcal{M}}\colon \mathbb{R}^{k\times k} \to \mathbb{R}^{k\times k}, \quad \widehat{\mathcal{M}}(X) := -\hat{A}X - X\hat{A}^T - \sum_{i=1}^m \hat{N}_i X \hat{N}_i^T,$$

where $\hat{A} = V^T AV$, $\hat{N}_i = V^T N_i V$ for $i = 1, \ldots, m$, and $V \in \mathbb{R}^{n\times k}$ is orthogonal. Our first result is concerned with the invertibility of the operators $\widetilde{\mathcal{M}}$ and $\widehat{\mathcal{M}}$, respectively.

**Proposition C.3.4.** *If $\sigma(\mathcal{M}) = -\sigma(\mathscr{L} + \Pi) \subset \mathbb{C}_+$, then $\sigma(\widetilde{\mathcal{M}}) \subset \mathbb{C}_+$ and $\sigma(\widehat{\mathcal{M}}) \subset \mathbb{C}_+$.*

*Proof.* Note that $\sigma(\widetilde{\mathcal{M}})$ is determined by the eigenvalues of the matrix

$$\widetilde{\mathbf{M}} := -I \otimes A - \hat{A} \otimes I - \sum_{i=1}^{m} \hat{N}_i \otimes N_i. \tag{C.15}$$

Similarly, we obtain $\sigma(\mathcal{M})$ by computing the eigenvalues of the matrix

$$\mathbf{M} := -I \otimes A - A \otimes I - \sum_{i=1}^{m} N_i \otimes N_i. \tag{C.16}$$

Since $A$ and $N_i$ are assumed to be symmetric, we conclude that $\mathbf{M} = \mathbf{M}^T \succ 0$. Let us then define the orthogonal matrix $\mathbf{V} = V \otimes I$. It follows that $\widetilde{\mathbf{M}} = \mathbf{V}^T \mathbf{M} \mathbf{V}$ and, consequently, $\widetilde{\mathbf{M}} = \widetilde{\mathbf{M}}^T \succ 0$. A similar argument with $\mathbf{V} = V \otimes V$ can be applied to show the second assertion. □

Given a reduced bilinear system, we naturally obtain an approximate solution to the generalized Lyapunov equation. Moreover, the error with respect to the $\mathcal{M}$-inner product is given by the $\mathcal{H}_2$-norms of the original and reduced system, respectively.

**Proposition C.3.5.** *Let $\Sigma$ denote a bilinear system* (C.8) *and let $A = A^T \prec 0, N_i = N_i^T$ for $i = 1, \ldots, m$, and $B = C^T$. Assume that $\rho(\mathscr{L}^{-1}\,\Pi) < 1$. Given an orthogonal $V \in \mathbb{R}^{n \times k}, k < n$, define $\widehat{\Sigma}$, the reduced bilinear system* (C.14)*, via $\hat{A} = V^T A V, \hat{N}_i = V^T N_i V$ and $\hat{B} = V^T B = \hat{C}^T$. Let $X$ be the solution to $\mathcal{M}(X) = BB^T$, and let $\hat{X}$ be the solution to $\widehat{\mathcal{M}}(\hat{X}) = \hat{B}\hat{B}^T$. Then*

$$\|X - V\hat{X}V^T\|_{\mathcal{M}}^2 = \|\Sigma\|_{\mathcal{H}_2}^2 - \|\widehat{\Sigma}\|_{\mathcal{H}_2}^2.$$

*Proof.* By assumption it holds that $\mathcal{M}$ and $\widehat{\mathcal{M}}$ are invertible and the controllability Gramians $X$ and $\hat{X}$ exist. We observe the relations $\|X\|_{\mathcal{M}}^2 = \operatorname{Tr}(XBB^T) = \|\Sigma\|_{\mathcal{H}_2}^2$ and $\langle V\hat{X}V^T, X\rangle_{\mathcal{M}} = \operatorname{Tr}(V\hat{X}V^T BB^T) = \|\widehat{\Sigma}\|_{\mathcal{H}_2}^2$. Moreover, for the reduced system we obtain

$$\widehat{\mathcal{M}}(\hat{X}) = -V^T(AV\hat{X}V^T + V\hat{X}V^T A + \sum_{i=1}^{m} N_i V\hat{X}V^T N_i)V = V^T\mathcal{M}(V\hat{X}V^T)V,$$

which implies that $\|V\hat{X}V^T\|_{\mathcal{M}}^2 = \operatorname{Tr}(\hat{X}\widehat{\mathcal{M}}(\hat{X})) = \|\widehat{\Sigma}\|_{\mathcal{H}_2}^2$. Hence, we obtain

$$\|X - V\hat{X}V^T\|_{\mathcal{M}}^2 = \|X\|_{\mathcal{M}}^2 + \|V\hat{X}V^T\|_{\mathcal{M}}^2 - 2\langle V\hat{X}V^T, X\rangle_{\mathcal{M}} = \|\Sigma\|_{\mathcal{H}_2}^2 - \|\widehat{\Sigma}\|_{\mathcal{H}_2}^2.$$

□

Extending the results from [7], we obtain a lower bound for the previous terms by the $\mathcal{H}_2$-norm of the error system $\Sigma - \widehat{\Sigma}$.

**Proposition C.3.6.** *Let $\Sigma$ denote a bilinear system* (C.8) *and let $A = A^T \prec 0, N_i = N_i^T$ for $i = 1, \ldots, m$, and $B = C^T$. Assume that $\rho(\mathscr{L}^{-1}\,\Pi) < 1$. Given an orthogonal $V \in \mathbb{R}^{n\times k}, k < n$, define $\widehat{\Sigma}$, the reduced bilinear system* (C.14)*, via $\hat{A} = V^T A V, \hat{N}_i = V^T N_i V$ and $\hat{B} = V^T B = \hat{C}^T$. Then, it holds*

$$\|\Sigma - \widehat{\Sigma}\|_{\mathcal{H}_2}^2 \leq \|\Sigma\|_{\mathcal{H}_2}^2 - \|\widehat{\Sigma}\|_{\mathcal{H}_2}^2,$$

*with equality if $\widehat{\Sigma}$ is locally $\mathcal{H}_2$-optimal.*

*Proof.* The proof follows by arguments similar to those used in [7, Lemma 3.1]. By definition of the $\mathcal{H}_2$-norm for bilinear systems

$$\|\Sigma - \widehat{\Sigma}\|_{\mathcal{H}_2}^2 = \operatorname{Tr}(\begin{bmatrix} B^T & -\hat{B}^T \end{bmatrix} X_e \begin{bmatrix} B \\ -\hat{B} \end{bmatrix}),$$

where $X_e = \begin{bmatrix} X & Y \\ Y^T & \hat{X} \end{bmatrix}$ is the solution of

$$\begin{bmatrix} A & 0 \\ 0 & \hat{A} \end{bmatrix} X_e + X_e \begin{bmatrix} A & 0 \\ 0 & \hat{A} \end{bmatrix} + \sum_{i=1}^{m} \begin{bmatrix} N_i & 0 \\ 0 & \hat{N}_i \end{bmatrix} X_e \begin{bmatrix} N_i & 0 \\ 0 & \hat{N}_i \end{bmatrix} + \begin{bmatrix} B \\ \hat{B} \end{bmatrix} \begin{bmatrix} B^T & \hat{B}^T \end{bmatrix} = 0.$$

Analyzing the block structure of $X_e$, adding and subtracting $\|\widehat{\Sigma}\|_{\mathcal{H}_2}^2 = \operatorname{Tr}(\hat{B}^T \hat{X} \hat{B})$, we find the equivalent expression

$$\|\Sigma - \widehat{\Sigma}\|_{\mathcal{H}_2}^2 = \|\Sigma\|_{\mathcal{H}_2}^2 - \|\widehat{\Sigma}\|_{\mathcal{H}_2}^2 - 2\left(\operatorname{Tr}(B^T Y \hat{B}) - \operatorname{Tr}(\hat{B}^T \hat{X} \hat{B})\right).$$

We claim that $\operatorname{Tr}(B^T Y \hat{B}) - \operatorname{Tr}(\hat{B}^T \hat{X} \hat{B}) \geq 0$ which then shows the first assertion. In fact, $Y$ and $\hat{X}$ are the solutions of $\widetilde{\mathcal{M}}(Y) = B\hat{B}^T$ and $\widehat{\mathcal{M}}(\hat{X}) = \hat{B}\hat{B}^T$. With the operators introduced in (C.15) and (C.16), we obtain

$$\begin{aligned} \operatorname{Tr}(B^T Y \hat{B}) - \operatorname{Tr}(\hat{B}^T \hat{X} \hat{B}) &= \widetilde{\mathbf{b}}^T \operatorname{vec}(Y) - \widehat{\mathbf{b}}^T \operatorname{vec}(\hat{X}) = \widetilde{\mathbf{b}}^T \widetilde{\mathbf{M}}^{-1} \widetilde{\mathbf{b}}^T - \widehat{\mathbf{b}}^T \widehat{\mathbf{M}}^{-1} \widehat{\mathbf{b}} \\ &= \widetilde{\mathbf{b}}^T \left( \widetilde{\mathbf{M}}^{-1} - \mathbf{V}(\mathbf{V}^T \widetilde{\mathbf{M}} \mathbf{V})^{-1} \mathbf{V}^T \right) \widetilde{\mathbf{b}}, \end{aligned}$$

where $\widetilde{\mathbf{b}} = \operatorname{vec}(B\hat{B}^T)$ and $\widehat{\mathbf{b}} = \operatorname{vec}(\hat{B}\hat{B}^T)$. As in [7, Lemma 3.1], it follows that the previous expression contains the Schur complement of $\widetilde{\mathbf{M}}^{-1}$ in $\mathbf{S} = \begin{bmatrix} \mathbf{V}^T \widetilde{\mathbf{M}} \mathbf{V} & \mathbf{V}^T \\ \mathbf{V} & \widetilde{\mathbf{M}}^{-1} \end{bmatrix}$ which can be shown to be positive semidefinite. We omit the details and refer to [7].

Assume now that $\widehat{\Sigma}$ is locally $\mathcal{H}_2$-optimal. From [41], we have the following first-order necessary optimality conditions

$$\begin{aligned} Y^T Z + \hat{X} \hat{Z} = 0, \quad Z^T N_i Y + \hat{X} \hat{N}_i \hat{Z} = 0, \;\; i = 1, \ldots, m, \\ Z^T B + \hat{Z} \hat{B} = 0, \quad C Y + \hat{C} \hat{X} = 0, \end{aligned}$$

where $Y, \hat{X}$ are as before and $Z, \hat{Z}$ satisfy

$$A^T Z + Z\hat{A} + \sum_{i=1}^{m} N_i^T Z \hat{N}_i - C^T \hat{C} = 0, \qquad \hat{A}^T \hat{Z} + \hat{Z}\hat{A} + \sum_{i=1}^{m} \hat{N}_i^T \hat{Z} \hat{N}_i + \hat{C}^T \hat{C} = 0.$$

From the symmetry of $A, \hat{A}, N_i$ and $\hat{N}_i$ as well as the fact that $B = C^T$ and $\hat{B} = \hat{C}^T$, we conclude that $\hat{Z} = \hat{X}$ and $Z = -Y$. Hence, from the optimality conditions, we obtain

$$0 = Z^T B + \hat{Z}\hat{B} = -Y^T B + \hat{X}\hat{B}$$

which in particular implies that

$$\operatorname{Tr}(B^T Y \hat{B}) - \operatorname{Tr}(\hat{B}^T \hat{X} \hat{B}) = \operatorname{Tr}(\hat{B}^T (Y^T B - \hat{X}\hat{B})) = 0.$$

This shows the second assertion. □

As a consequence of Propositions C.3.5 and C.3.6, we obtain the following result.

**Theorem C.3.7.** *Let $\Sigma$ denote a bilinear system* (C.8) *and let $A = A^T \prec 0, N_i = N_i^T$ for $i = 1, \ldots, m$ and $B = C^T$. Assume that $\rho(\mathscr{L}^{-1}\Pi) < 1$. Given an orthogonal $V \in \mathbb{R}^{n\times k}, k < n$, define $\widehat{\Sigma}$, the reduced bilinear system* (C.14)*, via $\hat{A} = V^T A V, \hat{N}_i = V^T N_i V$ and $\hat{B} = V^T B = \hat{C}^T$. Assume that $\hat{X}$ solves $\widehat{\mathcal{M}}(\hat{X}) = \hat{B}\hat{B}^T$. If $\widehat{\Sigma}$ is locally $\mathcal{H}_2$-optimal, then $V\hat{X}V^T$ is locally optimal with respect to the $\mathcal{M}$-norm.*

### C.3.3 Equivalence of ALS and rank-1 BIRKA

So far we have shown that a subspace producing a locally $\mathcal{H}_2$-optimal model reduction is also a subspace for which the Galerkin approximation is locally optimal in the $\mathcal{M}$-norm. In this part we, algorithmically, establish an equivalence between BIRKA and ALS. More precisely, for the symmetric case the equivalence is between BIRKA applied with the target model reduction subspace of dimension 1 for (C.8), and ALS applied to (C.1). The proof is based on the following lemmas.

**Lemma C.3.8.** *Consider using BIRKA (Algorithm C.2) with $k = 1$, i.e., both the initial guesses and the output are vectors. Then $\tilde{A} \in \mathbb{R}$ is a scalar and hence we can take $\tilde{\Lambda} = \tilde{A}$ and $R = 1$ in Step 2. Thus, $\hat{B} = \tilde{B}$, $\hat{C} = \tilde{C}$, $\hat{N}_1 = \tilde{N}_1, \ldots, \hat{N}_m = \tilde{N}_m$, and hence Steps 2–3 are redundant. Moreover, since $\tilde{V}$ and $\tilde{W}$ are vectors, Step 6, is redundant.*

*Proof.* The result follows from direct computation. □

When speaking about *redundant* steps and operations we mean that the entities assigned in that step are exactly equal to another, existing, entity. In such a situation the algorithm can be rewritten, by simply changing the notation, in a way that skips the redundant step and still produces the same result.

**Lemma C.3.9.** *Consider the symmetric generalized Lyapunov equation* (C.7) *and let* $v, w \in \mathbb{R}^n$ *be two given vectors. Let* $v_{\text{BIRKA}}, w_{\text{BIRKA}} \in \mathbb{R}^n$ *be the approximations obtained by applying BIRKA (Algorithm C.2) to* (C.1) *with* $C = B^T$ *and initial guesses* $v$ *and* $w$*. If* $v = w$*, then* $v_{\text{BIRKA}} = w_{\text{BIRKA}}$*.*

*Proof.* By induction it suffices to show that if $\tilde{V} = \tilde{W}$ at the beginning of a loop, the same holds at the end of the loop. Thus, assume $\tilde{V} = \tilde{W}$. Then $\tilde{N}_i = (\tilde{W}^T\tilde{V})^{-1}\tilde{W}^T N_i \tilde{V} = \tilde{V}^T N_i \tilde{V}/\|V\|^2 = \tilde{V}^T N_i^T \tilde{V}/\|V\|^2 = \tilde{N}_i^T$ for $i = 1, \ldots, m$, and $\tilde{C} = C\tilde{V} = B^T\tilde{W} = \tilde{B}^T$. By Lemma C.3.8 we do not need to consider Steps 2–3. We can now conclude that Step 4 and Step 5 are equal, and thus at the end of the iteration we still have $\tilde{V} = \tilde{W}$. □

**Lemma C.3.10.** *Consider the symmetric generalized Lyapunov equation* (C.7) *and let* $v, w \in \mathbb{R}^n$ *be two given vectors. Let* $v_{\text{ALS}}, w_{\text{ALS}} \in \mathbb{R}^n$ *be the approximations obtained by applying the ALS algorithm (Algorithm C.1) to* (C.1) *with initial guesses* $v$ *and* $w$*. If* $v = w$*, then* $v_{\text{ALS}} = w_{\text{ALS}}$*.*

*Proof.* Similar to the proof of Lemma C.3.9 it is enough to show that if $v = w$ at the beginning of a loop then it also holds at the end of the loop. Hence, we assume that $v = w$. Then by direct calculations $\hat{A}_1 = \hat{A}_2$. Moreover, by assumption $\mathcal{R}_k = \mathcal{R}_k^T$. Thus, Step 3 and Step 6 are equal, and hence at the end of the iteration we still have that $v = w$. □

**Theorem C.3.11.** *Consider the symmetric generalized Lyapunov equation* (C.7) *and let* $v \in \mathbb{R}^n$ *be a given vector. Let* $v_{\text{BIRKA}} \in \mathbb{R}^n$ *be the approximation obtained by applying BIRKA (Algorithm C.2) to* (C.1) *with* $C = B^T$ *and initial guess* $v$*. Moreover, let* $v_{\text{ALS}} \in \mathbb{R}^n$ *be the approximation obtained by applying the ALS algorithm (Algorithm C.1) to* (C.1) *with initial guess* $v$*. Then* $v_{\text{BIRKA}} = v_{\text{ALS}}$*.*

*Proof.* First, Lemma C.3.9 and Lemma C.3.10 makes it reasonable to assess the algorithms with only a single initial guess as well as a single output. Moreover, Step 5 in BIRKA as well as Steps 2–4 in ALS are redundant. Furthermore, it follows from Lemma C.3.8 that in this situation Steps 2, 3, and 6 of BIRKA are also redundant. Hence, we need to compare the procedure consisting of Steps 1 and 4 from BIRKA, with the procedure consisting of Steps 1, 5, and 6 from ALS. It can be observed that the computations are equivalent and thus the asserted equality holds if they stop after an equal amount of iterations. We hence consider the stopping criteria and note that they are the same, since $(v^T A^T v + v^T A v)/2\|v\|^2 = v^T A v/\|v\|^2 = \tilde{A} \in \mathbb{R}$. □

**Corollary C.3.12.** *Theorem C.3.2 is applicable with ALS changed to BIRKA, using subspaces of dimension 1.*

**Remark C.3.13.** *Note that ALS can be generalized such that the optimization is computing rank-*$\ell$ *corrections, see [25, Remark 2.2]. With similar arguments as above, one can show that for symmetric systems this can equivalently be achieved by BIRKA. From a theoretical point of view, this will yield more accurate approximations. However, the computational complexity increases quickly since each ALS or BIRKA step then requires solving a generalized Sylvester equation of dimension* $n \times \ell$*.*

## C.4 Fixed-point iteration and approximative $\mathcal{M}$-norm minimization

In the previous section we showed that the ALS-based iteration (C.11) locally minimizes the error in the $\mathcal{M}$-norm with rank-1 updates. In contrast we here show that the fixed-point iteration minimizes an upper bound for the $\mathcal{M}$-norm, but with no rank constraint on the minimizer.

Recall the fixed-point iteration for the generalized Lyapunov equation (C.1),

$$\mathscr{L}(\hat{X}_{k+1}) = -\Pi(\hat{X}_k) - BB^T, \qquad k = 0, 1, \ldots, \tag{C.17}$$

with $\hat{X}_0 = 0$. Under the assumption that $\rho(\mathscr{L}^{-1}\,\Pi) < 1$ the iteration is a convergent splitting and has been presented in, e.g., [12, Equation (12)], [41, Equation (12)], and [37, Equation (4)]. The fixed-point iteration is a residual-based iteration since (C.17) is known to be equivalent to

$$\hat{X}_{k+1} = \hat{X}_k - \mathscr{L}^{-1}(\mathcal{R}_k), \qquad k = 0, 1, \ldots, \tag{C.18}$$

with $\hat{X}_0 = 0$. To relate the fixed-point iteration to the $\mathcal{M}$-norm minimization problem we consider the problem

$$\min_{\substack{\Delta \\ \Delta = \Delta^T \succeq 0}} \|X - \hat{X}_k - \Delta\|_{\mathcal{M}}^2.$$

The minimization is restricted to symmetric positive semidefinite matrices since we know that the solution $X = X^T \succeq 0$. Hence, it is desired to have that if $\hat{X}_k = \hat{X}_k^T \succeq 0$, then the new iterate $\hat{X}_{k+1} = \hat{X}_k + \Delta$ also fulfills $\hat{X}_{k+1} = \hat{X}_{k+1}^T \succeq 0$; specifically then $\hat{X}_{k+1} \succeq \hat{X}_k$. Proposition C.2.3 gives us the solution in just one step. However, the computation is as difficult as the original problem and hence the goal is to minimize an upper bound on the error. As before we disregard the constant term $\|X - \hat{X}_k\|_{\mathcal{M}}^2$ in the minimization and consider

$$\begin{aligned}\min_{\substack{\Delta \\ \Delta = \Delta^T \succeq 0}} \langle \Delta, \Delta \rangle_{\mathcal{M}} - 2\operatorname{Tr}(\Delta^T \mathcal{R}_k) &= \min_{\substack{\Delta \\ \Delta = \Delta^T \succeq 0}} \operatorname{Tr}(\Delta^T(-\mathscr{L}(\Delta) - 2\mathcal{R}_k) - \Delta^T \Pi(\Delta)) \\ &\leq \min_{\substack{\Delta \\ \Delta = \Delta^T \succeq 0}} \operatorname{Tr}(\Delta^T(-\mathscr{L}(\Delta) - 2\mathcal{R}_k)),\end{aligned}$$

where the inequality is a consequence of the linearity of the trace and the positive semidefiniteness of $\Delta^T$ and $\Pi(\Delta)$. Hence, the trace is non-negative [31]. The last expression is minimized by $\Delta = -\mathscr{L}^{-1}(\mathcal{R}_k)$, if $\mathcal{R}_k$ is symmetric and positive definite. The latter is asserted in the following theorem.

**Theorem C.4.1.** *Consider the symmetric generalized Lyapunov equation* (C.7) *with the additional assumptions that $A \prec 0$ and $\rho(\mathscr{L}^{-1}\,\Pi) < 1$. Moreover, consider the sequence of approximations constructed by* (C.18) *where $\mathcal{R}_k$ is the residual associated with $\hat{X}_k$. Then $\hat{X}_k = \hat{X}_k^T \succeq 0$ and $\mathcal{R}_k = \mathcal{R}_k^T \succeq 0$, for all $k \geq 0$.*

*Proof.* The proof is by induction and similar to that of Theorem C.3.2. It holds that $X_0 = X_0^T \succeq 0$ and $\mathcal{R}_0 = \mathcal{R}_0^T \succeq 0$. Now assume that this is the case for some $k$. Then $\Delta = -\mathscr{L}^{-1}(\mathcal{R}_k)$ is symmetric and positive semidefinite, and hence $\hat{X}_{k+1}$ is symmetric and positive semidefinite. Moreover, since $\hat{X}_{k+1}$ and the operators in (C.1) are symmetric it follows that $\mathcal{R}_{k+1}$ is symmetric. Thus, what is left to show is $\mathcal{R}_{k+1} \succeq 0$, which is true if and only if $y^T \mathcal{R}_{k+1} y \geq 0$ for all $y \in \mathbb{R}^n$. Hence, take an arbitrary $y \in \mathbb{R}^n$ and consider

$$y^T \mathcal{R}_{k+1} y = y^T \mathcal{R}_k y + y^T \left(\mathscr{L}(\Delta) + \Pi(\Delta)\right) y = y^T \left(\Pi(\Delta)\right) y \geq 0.$$

The last inequality holds since $\Delta$ is symmetric and positive semidefinite and $\Pi$ is a symmetric operator. □

**Corollary C.4.2.** *The fixed-point iteration* (C.17) *produces an increasing sequence of approximations* $0 = \hat{X}_0 \preceq \hat{X}_1 \preceq \cdots \preceq X$.

**Remark C.4.3.** *One could consider creating a subspace iteration from* (C.18)*, by computing a few singular vectors of* $\mathscr{L}^{-1}(\mathcal{R}_k)$ *and adding these to the basis. The method seems to have nice convergence properties per iteration in the symmetric case, but not in the non-symmetric case. However, the (naive) computations are prohibitively expensive. See [37] for a computationally more efficient way of exploiting the fixed-point iteration.*

## C.5 A residual-based rational Krylov generalization

A viable technique for designing iterative methods for the generalized Lyapunov equation seems to be working with the residual; see the discussion in connection to Proposition C.2.3, and in Sections C.3 and C.4. In [25, Section 4] it is suggested that, so called, preconditioned residuals can be used to expand the search space. It is further suggested that one such preconditioner could be a one-step-ADI preconditioner $P_{\text{ADI}}^{-1} = (A - \sigma I)^{-1} \otimes (A - \sigma I)^{-1}$, for a suitable choice of the shift. We present a method along those lines, and show that it can be seen as a generalization of the rational Krylov subspace method.

### C.5.1 Suggested search space

For the generalized Lyapunov equation (C.1), we suggest the following search space:

$$\mathcal{K}_k := \text{range}\{B, (A - \sigma_1 I)^{-1} u_0, (A - \sigma_2 I)^{-1} u_1, \ldots, (A - \sigma_k I)^{-1} u_{k-1}\}, \quad \text{(C.19)}$$

where $u_{k-1}$ is the most dominant left singular vector of the Galerkin residual $\mathcal{R}_{k-1}$ of $\mathcal{K}_{k-1}$, and $\{\sigma_\ell\}_{\ell=1}^k$ is a sequence of shifts that needs to be chosen. In analogy with the discussion in [15], we suggest that the shifts are chosen according to the largest approximation error along the current direction. More precisely,

$$\sigma_k := \arg\max_{\sigma \in [\sigma_{\min}, \sigma_{\max}]} \left( \left\| u_{k-1} - (A - \sigma I)\, \mathcal{V}_{k-1} (\hat{A}_{k-1} - \sigma I)^{-1}\, \mathcal{V}_{k-1}^T\, u_{k-1} \right\| \right), \quad \text{(C.20)}$$

where $\mathcal{V}_{k-1}$ is a matrix with orthogonal columns containing a basis of $\mathcal{K}_{k-1}$, the matrix $\hat{A}_{k-1} = \mathcal{V}_{k-1}^T A \mathcal{V}_{k-1}$, and $[\sigma_{\min}, \sigma_{\max}]$ is a search interval. Typically for a stable matrix $A$ we let $\sigma_{\min}$ be the negative real part of the eigenvalue of $A$ with largest real part (closest to 0). Correspondingly we let $\sigma_{\max}$ the negative real part of the eigenvalue of $A$ with smallest real part. Equations (C.19) and (C.20) can be straightforwardly incorporated in a Galerkin method for the generalized Lyapunov equation; the pseudocode is presented in Algorithm C.3.

**Algorithm C.3:** Residual-based rational-Krylov-type solver

**input :** $A, N_1, \ldots, N_m \in \mathbb{R}^{n\times n}$ $B \in \mathbb{R}^{n\times r}$, `tol`
**output:** $\hat{X}$

1 $\mathcal{V}_0 = \emptyset$, $v_1 = \texttt{orth}(B)$
**for** $k = 1, 2, \ldots$ *until convergence* **do**
2 $\quad \mathcal{V}_k = \left[\mathcal{V}_{k-1}, v_k\right]$
3 $\quad$ Compute the projected matrices: $\hat{A}_k = \mathcal{V}_k^T A \mathcal{V}_k$, and $\hat{N}_{i,k} = \mathcal{V}_k^T N_i \mathcal{V}_k$ for $i = 1, 2, \ldots, m$, and $\hat{B}_k = \mathcal{V}_k^T B$
4 $\quad$ Solve the projected problem: $\hat{A}_k Y_k + Y_k \hat{A}_k^T + \sum_{i=1}^m \hat{N}_{i,k} Y_j \hat{N}_{i,k}^T + \hat{B}_k \hat{B}_k^T = 0$
5 $\quad$ Construct the (Galerkin) approximation: $\hat{X}_k = \mathcal{V}_k Y_k \mathcal{V}_k^T$
6 $\quad$ Compute the residual: $\mathcal{R}_k = A\hat{X}_k + \hat{X}_k A^T + \sum_{i=1}^m N_i \hat{X}_k N_i^T + BB^T$
7 $\quad$ **if** $\|\mathcal{R}_k\| < \texttt{tol}$ **then** Break
8 $\quad$ $u_k \leftarrow$ the most dominant left singular vector of $\mathcal{R}_k$
9 $\quad$ Select shift $\sigma_{k+1}$ according to (C.20)
10 $\quad$ $v_{k+1} = (A - \sigma_{k+1} I)^{-1} u_k$
11 $\quad$ $v_{k+1} \leftarrow$ orthogonalize $v_{k+1}$ with respect to $\mathcal{V}_k$
12 **return** $\hat{X} = \hat{X}_k$

**Remark C.5.1.** *In practice the computation of the left singular vector can typically be done approximatively in an iterative fashion. This would also remove the need of computing the approximative solution $\hat{X}_k$ in Step 5 and the residual in Step 6 explicitly, since the matrix vector product can be implemented as $\mathcal{R}_k v = A \mathcal{V}_k Y_k \mathcal{V}_k^T v + \mathcal{V}_k Y_k \mathcal{V}_k^T A^T v + \sum_{i=1}^m N_i \mathcal{V}_k Y_k \mathcal{V}_k^T N_i^T v + BB^T v$. However, such computations may introduce inexactness which can present a difficulty in a subspace method.*

**Remark C.5.2.** *Heuristically the dynamic shift-search in Step 9 can be changed to an analogue of [15, (2.4) and (2.2)]. We suggest*

$$\sigma_k := \arg\max_{\sigma \in \partial S} \frac{1}{|\tau_{k-1}(z)|}, \tag{C.21}$$

*where $S$ approximates the mirrored spectrum of $A$ and $\partial S$ is the boundary of $S$, and*

$$\tau_{k-1}(z) := \frac{\prod_{j=1}^{\dim(\mathcal{K}_{k-1})} z - \lambda_j^{(k-1)}}{\prod_{\ell=1}^{k-1} z - \sigma_\ell},$$

*with $\lambda_j^{(k-1)}$ being the Ritz values of $\hat{A}_{k-1}$. Typically $S$ is approximated at each step using the convex hull of the Ritz values of $\hat{A}_{k-1}$. It has been observed efficient in experiments since the maximization of* (C.21) *is computationally faster compared to* (C.20)*. See Section C.6 for a practical comparison of convergence properties.*

**Remark C.5.3.** *The steps 8–9 in Algorithm C.3 can be changed for a tangential-direction approach according to [16]. One practical way, although a heuristic, is to do the shift search according to either* (C.20) *or* (C.21)*, and then compute the principal direction(s) according to [16, Section 3], i.e., through a singular value decomposition of $\mathcal{R}_{k-1} - (A - \sigma_k I)\,\mathcal{V}_{k-1}(\hat{A}_{k-1} - \sigma_k I)^{-1}\,\mathcal{V}_{k-1}^T\,\mathcal{R}_{k-1}$. It has been observed in experiments that such an approach tends to speed up the convergence, in terms of computation time, since the computation of the residual is costly.*

**Remark C.5.4.** *It is (sometimes) desirable to allow for complex conjugate shifts $\sigma_k$ and $\bar{\sigma}_k$, although, for reasons of computations and model interpretation one wants to keep the basis real. This goal is achievable using the same idea as in [35]. More precisely, one can utilize the relation*

$$\begin{aligned}&\operatorname{range}\left\{(A-\sigma_k I)^{-1}u_{k-1},\,(A-\bar{\sigma}_k I)^{-1}u_{k-1}\right\} = \\ &\operatorname{range}\left\{\operatorname{Re}((A-\sigma_k I)^{-1}u_{k-1}),\,\operatorname{Im}((A-\sigma_k I)^{-1}u_{k-1})\right\}.\end{aligned}$$

*Although it requires two shifts to be used together with the vector $u_{k-1}$.*

### C.5.2 Analogies to the linear case

To give further insight to the suggested subspace in (C.19), we draw parallels with the (standard) rational Krylov subspace for the (standard) *Lyapunov equation*,

$$\mathscr{L}(X) + BB^T = 0, \tag{C.22}$$

where $\mathscr{L}$ is defined by (C.2) and $B \in \mathbb{R}^{n\times r}$. The idea is to show that the suggested space (C.19), reduces to something well known in this case. As a technical note we observe that Definitions C.2.1 and C.2.2 are analogous for the (standard) Lyapunov equation (C.22), but with $\Pi = 0$. The reasoning in this section can be compared to that of [3, Section 2]. To prove the main result of this section we need the following lemma.

**Lemma C.5.5.** *Let $A \in \mathbb{R}^{n\times n}$ and $\sigma_a \in \mathbb{R}$ be any scalar such that $(A - \sigma_a I)$ is nonsingular. Moreover, let $\mathcal{V} \in \mathbb{R}^{n\times k}$, $k \leq n$, be orthogonal, i.e., $\mathcal{V}^T\mathcal{V} = I$, and let $\mathcal{R} \in \mathbb{R}^{n\times n}$ be such that $\operatorname{range}((A-\sigma_a I)^{-1}\mathcal{R}) \subseteq \operatorname{range}(\mathcal{V})$. Then $\mathcal{R} = (A - \sigma_a I)\,\mathcal{V}(\mathcal{V}^T A\,\mathcal{V} - \sigma_a I)^{-1}\,\mathcal{V}^T\,\mathcal{R}$.*

*Proof.* We introduce the notation $S := (A - \sigma_a I)$ and $\hat{S} := (V^T A V - \sigma_a I)$. To prove the statement we consider the right-hand side of the asserted equality,

$$S\,\mathcal{V}\,\hat{S}^{-1}\,\mathcal{V}^T\,\mathcal{R} = S\,\mathcal{V}\,\hat{S}^{-1}\,\mathcal{V}^T\,SS^{-1}\mathcal{R} = S\,\mathcal{V}\,\hat{S}^{-1}\,\mathcal{V}^T\,S\,\mathcal{V}\,\mathcal{V}^T\,S^{-1}\mathcal{R},$$

where the second equality follows from the assumption $\mathrm{range}(S^{-1}\mathcal{R}) \subseteq \mathrm{range}(\mathcal{V})$. By observing that $\hat{S}^{-1}\,\mathcal{V}^T\,S\,\mathcal{V} = I$ the expression can be further simplified as

$$S\,\mathcal{V}\,\hat{S}^{-1}\,\mathcal{V}^T\,\mathcal{R} = S\,\mathcal{V}\,\mathcal{V}^T\,S^{-1}\mathcal{R} = SS^{-1}\mathcal{R} = \mathcal{R},$$

where, again, the second equality follows from $\mathrm{range}(S^{-1}\mathcal{R}) \subseteq \mathrm{range}(\mathcal{V})$. □

**Theorem C.5.6.** *Let $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times r}$, and let $\{\sigma_\ell\}_{\ell=1}^{k+1}$ be a sequence of shifts such that $A - \sigma_\ell I$ is nonsingular for $\ell = 1, \ldots, k+1$. Define the space $\mathcal{K}_k := \mathrm{range}\{B, (A - \sigma_1 I)^{-1}B, \ldots, \prod_{\ell=1}^{k}(A - \sigma_\ell I)^{-1}B\}$, and $\mathcal{K}_{k+1}$ analogously. Let $\mathcal{V}_k$ be an orthogonal basis of $\mathcal{K}_k$, $\mathcal{V}_{k+1}$ an orthogonal basis of $\mathcal{K}_{k+1}$, and let $v_{k+1} \in \mathbb{R}^{n\times r}$ be such that $\mathcal{V}_{k+1} = \begin{bmatrix}\mathcal{V}_k, & v_{k+1}\end{bmatrix}$.*[1] *Moreover, let $\mathcal{R}_k \in \mathbb{R}^{n\times n}$ be the Galerkin residual with respect to* (C.22)*. Then each column of $(A - \sigma_{k+1}I)^{-1}\mathcal{R}_k$ is in $\mathrm{range}(V_{k+1})$, i.e., $\mathrm{range}((A - \sigma_{k+1}I)^{-1}\mathcal{R}_k) \subseteq \mathrm{range}(V_{k+1})$. Furthermore, if $\mathrm{range}((A - \sigma_{k+1}I)^{-1}\mathcal{R}_k) \subseteq \mathrm{range}(V_k)$, then $\mathcal{R}_k = 0$.*

*Proof.* We introduce the notation $S_{k+1} := (A - \sigma_{k+1}I)$ and $\hat{S}_{k+1} := (V^T A V - \sigma_{k+1}I)$.

From existing results on rational Krylov subspaces, see, e.g., [27, Proposition 2.2], there exists $\alpha \in \mathbb{R}^{r\times n}$ such that

$$\begin{aligned}\mathcal{R}_k &= AV_kY_kV_k^T + V_kY_kV_k^TA^T + BB^T \\ &= \sigma_{k+1}v_{k+1}\alpha - (I - V_kV_k)Av_{k+1}\alpha + V_kT_kY_kV_k^T + V_kY_kV_k^TA^T + V_kV_k^TBB^T \\ &= -S_{k+1}v_{k+1}\alpha + V_k\beta\end{aligned}$$

for a suitable $\beta \in \mathbb{R}^{(k+1)r\times n}$. This shows the first claim.

To prove the second claim we assume that $\mathrm{range}(S_{k+1}^{-1}\mathcal{R}_k) \subseteq \mathcal{K}_k = \mathrm{range}(\mathcal{V}_k)$. Under this assumption we can use Lemma C.5.5 and the fact that $\mathcal{R}_k = \mathcal{R}_k^T$ to get

$$\mathcal{R}_k = S_{k+1}\,\mathcal{V}_k\,\hat{S}_{k+1}^{-1}\,\mathcal{V}_k^T\,\mathcal{R}_k = S_{k+1}\,\mathcal{V}_k\,\hat{S}_{k+1}^{-1}\,\mathcal{V}_k^T\,\mathcal{R}_k\,\mathcal{V}_k\,\hat{S}_{k+1}^{-1}\,\mathcal{V}_k^T\,S_{k+1} = 0,$$

since $\mathcal{R}_k$ is the Galerkin residual and thus $\mathcal{V}_k^T\,\mathcal{R}_k\,\mathcal{V}_k = 0$. □

**Remark C.5.7.** *The interpretation of Theorem C.5.6 is easiest in the case when $B = b \in \mathbb{R}^n$. Consider the two spaces $\mathcal{K}_k := \mathrm{range}\{b, (A - \sigma_1 I)^{-1}b, \ldots, \prod_{\ell=1}^{k}(A - \sigma_\ell I)^{-1}b\}$ and $\hat{\mathcal{K}}_k := \mathrm{range}\{\mathcal{R}_{-1}, (A - \sigma_1 I)^{-1}\mathcal{R}_0, \ldots, (A - \sigma_k I)^{-1}\mathcal{R}_{k-1}\}$, where $\mathcal{R}_{-1} = b$ and $\mathcal{R}_j$ is the Galerkin residual in space $\mathcal{K}_j$, with $j = 0, 1, \ldots, k-1$ . Then for all relevant cases, i.e., $\mathcal{R}_j \neq 0$ for $j = -1, 0, \ldots, k-1$, we have that $\mathcal{K}_k = \hat{\mathcal{K}}_k$. In this sense the suggested subspace in* (C.19) *can be seen as a natural generalization of a rational Krylov subspace for linear matrix equations.*

[1] Here we have, implicitly, assumed that the dimension of the $\mathcal{K}_{k+1}$ is $n \times (k+2)r$, i.e., all the columns in the definition of the space are linearly independent.

## C.6 Numerical examples

We now numerically compare different methods discussed in the paper. All algorithms are treated in a subspace fashion[2] and we compare practically achieved approximation properties as a function of subspace dimension. Since the paper focuses on the symmetric problem we use Galerkin projection in the tested methods, except BIRKA. However, to (numerically) investigate the domain of application we test the methods on problems with varying degree of symmetry.

For small and moderate sized problems there are algorithms for computing the full solution, see [24, Algorithm 2], cf. [41, equation (12)]. Although costly, this nevertheless allows for inspection of the *relative error*, i.e.,

$$\|X - \hat{X}_k\|/\|X\|.$$

Moreover, it also allows comparison with the (in the Frobenius norm) optimal low-rank approximation based on the SVD.

We summarize some of the implementation details. Specifically, BIRKA is implemented as described in Algorithm C.2, with a maximum allowed number of iterations equal to 100. Convergence tolerance is implemented as relative norm difference of the vector of sorted eigenvalues and was set to $10^{-3}$. Each subspace is computed independently from a random initial guess. We emphasize that the method based on ALS is a subspace method, and not an iteratively updated method as described in (C.11). Because of the structure of the generalized Lyapunov equation, the solution is symmetric even if the coefficient matrices are not, we use a symmetric version of ALS even for the non-symmetric examples. More precisely, a symmetrized version of Algorithm C.1, cf. Lemma C.3.10, was used as an inner iteration, the resulting vector was used in an outer iteration to expand the search space, and the approximation was found using Galerkin projection. The maximum allowed number of iterations in ALS (inner iteration) was set to 20, and the tolerance to $10^{-2}$. With reference to [25] we note that preconditioned residuals were not used, although it may accelerate convergence. Regarding the rational-Krylov-type methods we compare the following methods, which we give short labels for the legends further down:

- A: $\mathcal{K}_k$ as in (C.19), according to Algorithm C.3
- B: Algorithm C.3 but with tangential directions according to Remark C.5.3, though with shifts according to (C.20)
- C: Algorithm C.3 but with shifts according to (C.21)
- D: Algorithm C.3 but with tangential directions according to Remark C.5.3 and shifts according to (C.21)
- E: Standard rational Krylov. More precisely, similar to Algorithm C.3, but instead of using $u_{k-1}$ we use the right-hand side $B$ in both (C.19) and (C.20)

[2]The technique of turning an iterative method, such as, e.g., ALS, into a subspace method is known as *Galerkin acceleration*. The idea is nicely explained in [25, Section 3].

- F: $\mathcal{K}_k$ as in (C.19), but with on-beforehand-prescribed shifts given as the recycling of mirrored eigenvalues from a size-10-BIRKA (convergence tolerance set to $10^{-3}$). Mirrored eigenvalues are potentially complex, with positive real part, and taken in ascending order according to the real parts.

For methods C, D, and F the shifts may be complex-valued, and the complex arithmetic is avoided by creating the space in accordance with Remark C.5.4. For methods A–E, the shift-search-boundaries were set to $\sigma_{\text{max}} = -1.01 \cdot \min_{\lambda \in \sigma(A)} \lambda$ and $\sigma_{\text{min}} = -0.99 \cdot \max_{\lambda \in \sigma(A)} \lambda$, as to slightly enlarge the region. For methods A, B, and E, the shifts are taken as approximations to (C.20). The approximation is computed by discretizing the interval $[\sigma_{\text{min}}, \sigma_{\text{max}}]$ in 30 equidistant points and comparing the value of the target function. Orthogonalization of the basis is implemented using MATLAB built-in QR factorization and keeping vectors only if the corresponding diagonal element in R is large enough. Implementations for the methods A-F are available online.[3] The simulations were done in MATLAB R2018a (9.4.0.813654) on a computer with four 1.6 GHz processors and 16 GB of RAM.

We test the algorithms on three different problems. All examples are bilinear control systems and we approximate the associated controllability Gramian, as in (C.9). The examples all have stable Lyapunov operators. The first example is symmetric, the second is non-symmetric but symmetrizable, and the third example is non-symmetric.

### C.6.1 Heat equation

The first example is motivated by an optimal control problem for selective cooling of steel profiles, see [17]. In this example, the state variable $w$ models the evolution of a temperature and is described by a two-dimensional heat equation,

$$\frac{\partial}{\partial t} w(x, y, t) = \Delta w(x, y, t) \qquad (x, y, t) \in (0, 1) \times (0, 1) \times (0, T),$$

where a control $u(t)$ enters bilinearly from the left through a Robin condition,

$$-\frac{\partial}{\partial x} w(0, y, t) = 0.5(w(0, y, t) - 1)u(t) \qquad (y, t) \in (0, 1) \times (0, T).$$

The control can be interpreted as the spraying intensity of a cooling fluid. The other spatial boundaries satisfy homogeneous Dirichlet conditions, and at $t = 0$ an initial temperature profile is specified. The equation is discretized in space using centered finite difference, which yields a bilinear system with $A \in \mathbb{R}^{5041 \times 5041}$, $B \in \mathbb{R}^{5041}$, $m = 1$, and $N_1 = N \in \mathbb{R}^{5041 \times 5041}$. It can be further noted that, $A = A^T \prec 0$ and $N = N^T$, and hence the theory of $\mathcal{H}_2$-optimality and the definition of the $\mathcal{M}$-norm is applicable.

We compare different methods discussed in the paper, both the relative residual norm and the relative error. For readability the plots have been split in different figures. Hence, in Figure C.1 we compare across different classes of methods, and in Figure C.2 we compare between different flavors the rational-Krylov-type methods. It can be observed, see

[3] https://people.kth.se/~eringh/software/res_rat_Kry_type/

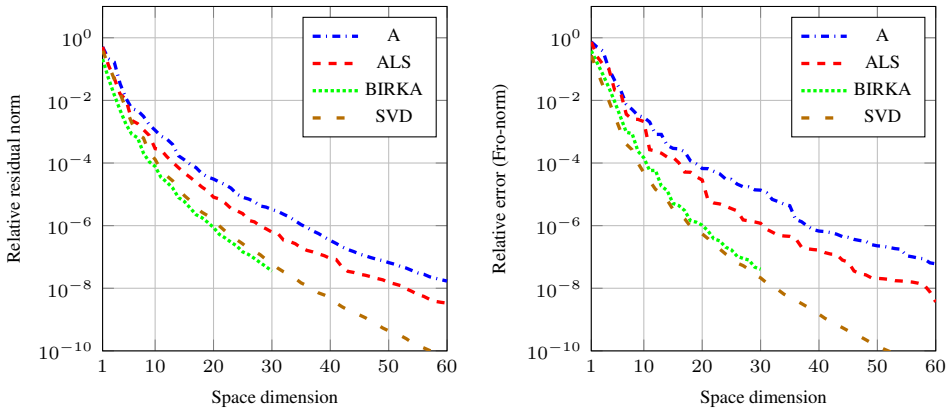



Figure C.1: Cross-algorithm comparison for the heat equation. Relative residual norm (left), relative error in Frobenius norm (right).

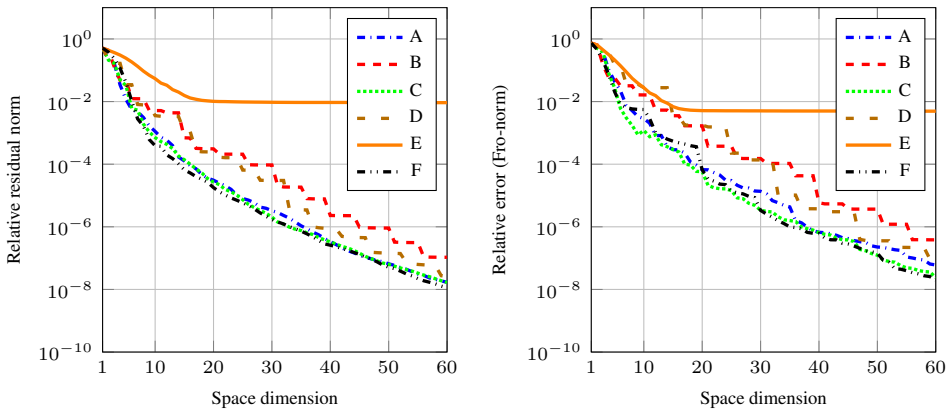



Figure C.2: Rational-Krylov-type method comparison for the heat equation. Relative residual norm (left), relative error in Frobenius norm (right). Compare with Figure C.1 as the lines for method A are the same in the two figures. For a description of the labels, see the beginning of this section.

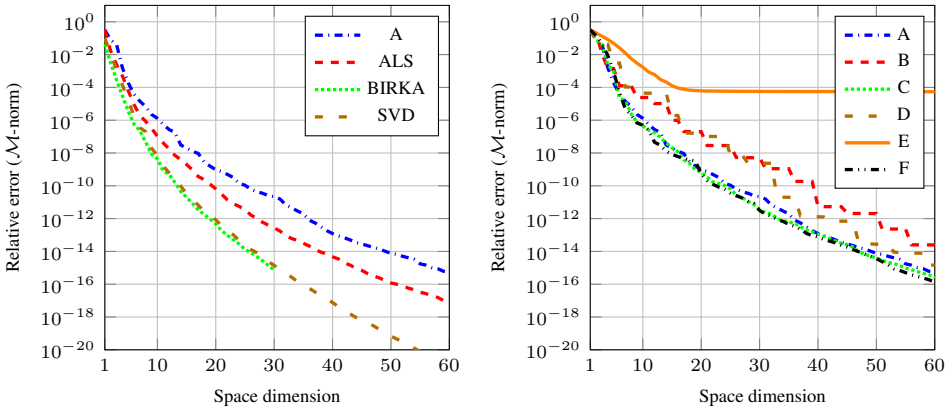



Figure C.3: The cross-algorithm comparison (left) and rational-Krylov-type methods (right) for the heat equation. The relative error is measured in the $\mathcal{M}$-norm. The lines for method A are the same in the two plots.

Figure C.1, that for this example BIRKA has extremely good performance, even outperforming the SVD in relative residual norm. Nevertheless, the larger BIRKA subspaces can be rather costly to compute. In comparison ALS shows good performance compared to the rational-Krylov-type subspace, and is rather cheap to compute. When comparing the different rational-Krylov-type methods, see Figure C.2, we see that standard rational-Krylov (E) has the problem that the convergence stagnates. The methods A, C, and F have similar performance. In comparison, B and D are only slightly worse in the error per subspace dimension comparison but are practically sometimes faster to compute.

Since the $\mathcal{M}$-norm is defined for this example we compare the relative error also in this norm, see Figure C.3. The trend is similar as in the Frobenius norm, although in general the error is smaller and BIRKA has best performance, even compared to the SVD.

### C.6.2 1D Fokker–Planck

The second example is from quantum physics, where a one-dimensional Fokker–Planck equation is used to describe the evolution of a probability density function, $\rho$, of a particle affected by a potential. Parts of the potential can be manipulated by a so-called optical tweezer, which constitutes the control. For further details of the problem see [22]. More precisely we consider

$$\begin{aligned}
\frac{\partial}{\partial t}\rho(x,t) &= \nu\frac{\partial^2}{\partial x^2}\rho(x,t) + \frac{\partial}{\partial x}\left(\rho(x,t)\frac{\partial}{\partial x}V(x,t)\right) && (x,t) \in (-6,6)\times(0,T)\\
\rho(x,0) &= \rho_0(x) && x \in (-6,6)\\
\nu\frac{\partial}{\partial x}\rho(x,t) &= -\rho(x,t)\frac{\partial}{\partial x}V(x,t) && (x,t) \in \{-6,6\}\times(0,T),
\end{aligned}$$

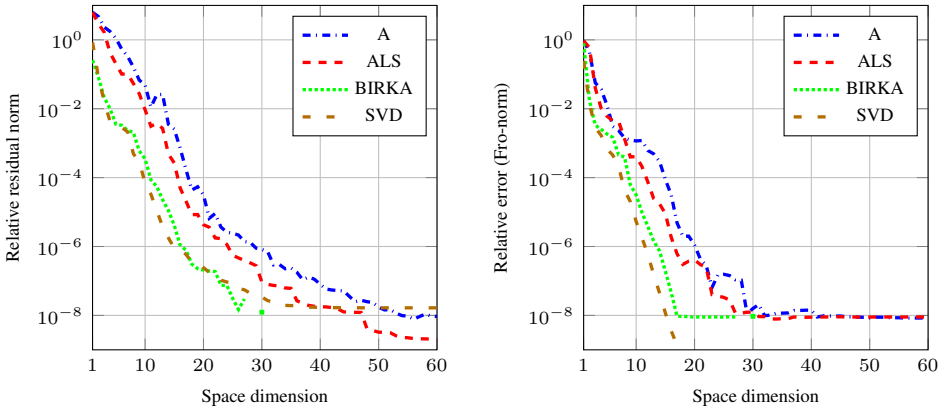



Figure C.4: Cross-algorithm comparison for 1D Fokker–Planck equation. Relative residual norm (left) and relative error (right).

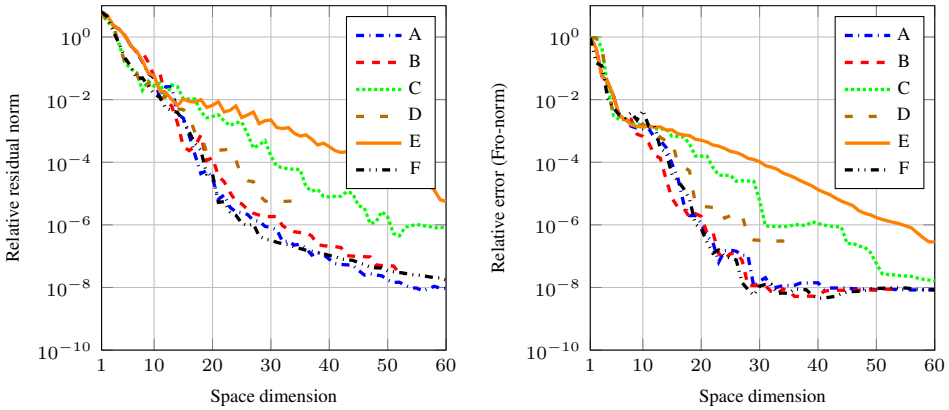



Figure C.5: Rational-Krylov-type method comparison for 1D Fokker–Planck equation. Compare with Figure C.4 as the lines for method A are the same in the two figures. For a description of the labels, see the beginning of this section.

where the potential is $V(x,t) = W(x) + \alpha(x)u(t)$, with the ground (fixed) potential being $W(x) = \left(\left((0.5x^2 - 15)x^2 + 199\right)x^2 + 28x + 50\right)/200$, and $\alpha(x)$ is an approximately linear control shape function; for more details see [11]. In a weighted inner product, the dynamics can be described by self-adjoint operators. However, here we employ an upwinding type finite difference scheme with $5000$ grid points, leading to a non-symmetric system. As has been pointed out in [22], the system matrix $A$ is not asymptotically stable due to a simple zero eigenvalue associated with the stationary probability distribution. Using a projection-based decoupling, it is however possible to work with an asymptotically stable system of dimension $n = 4999$. Similar to the first example, the control variable is a scalar and, consequently, we only obtain a single bilinear coupling matrix $N_1 = N$. Since the system is non-symmetric, the operator $\mathcal{M}$ is generally indefinite and hence we make no comparisons in the $\mathcal{M}$-norm.

The plots in Figures C.4 and C.5 are analogous to the plots in Figures C.1 and C.2 respectively. However, for this example the direct solver stagnated at a relative residual of about $10^{-8}$, which can be seen in the stagnation of the SVD approximation in the left of Figure C.4. As a result, the comparisons of relative error performance, the right of Figures C.4 and C.5, show an artificial stagnation. At a certain level the convergence stagnates since it measures the discrepancy between the method approximations and the inexact reference solution, rather than the true error of the method approximations. Nevertheless we believe the comparisons to be fair more or less up to to the point of stagnation, which is justified by the relative residual plots showing similar behavior. However, the relative residual indicates stagnation around $10^{-8}$ for the other methods as well, although not quite as clear as for the SVD.

From Figure C.4 we see the BIRKA performs well for this example. However, the subspaces of dimension 28 and 29 did not converge in a 100 iterations and hence for clarity these are left out of the plots. This illustrates a drawback of the method. The performance difference between ALS and the rational-Krylov-type method is slightly smaller compared to the previous example. Among the rational-Krylov-type methods A, B, and F seems to have similar performance, whereas C is clearly worse. Method E is competitive for about 10 iterations and then the convergence is significantly slower. However, method D ends up with an insufficient subspace.

### C.6.3 Burgers' equation

In the third example we consider an approximation to the one-dimensional viscous Burgers' equation

$$\begin{aligned} \frac{\partial}{\partial t}w(x,t) + w(x,t)\frac{\partial}{\partial x}w(x,t) &= \nu\frac{\partial^2}{\partial x^2}w(x,t) && (x,t) \in (0,1)\times(0,T) \\ w(x,0) &= w_0(x) && x \in (0,1) \end{aligned}$$

where $\nu = 0.1$ is constant. The spatial boundary conditions are Dirichlet conditions. More specifically, $w(1,t) = 0$ and $w(0,t) = u(t)$, where $u(t)$ is an applied control input. The solution $w(x,t)$ can be interpreted as a velocity and the equation occurs in, e.g., modeling

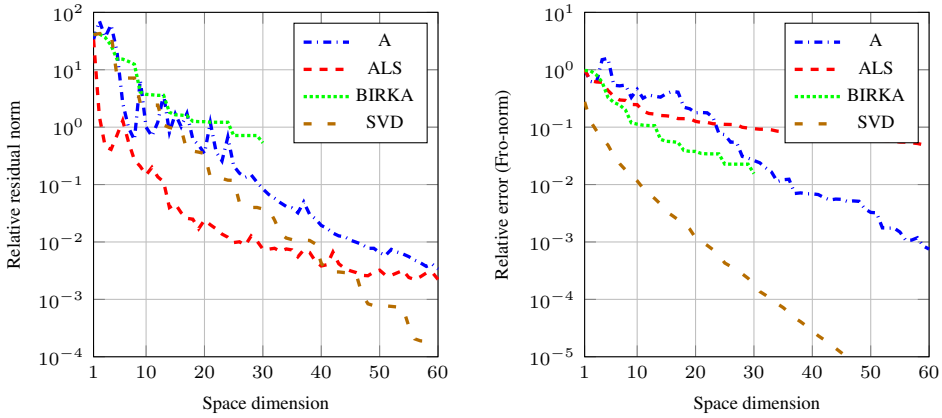



Figure C.6: Cross-algorithm comparison for Burgers' equation. Relative residual norm (left) and relative error (right).

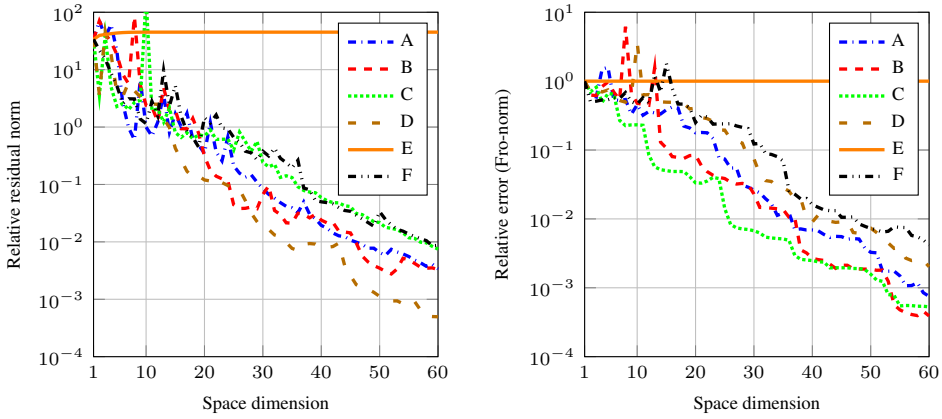



Figure C.7: Rational-Krylov-type method comparison for Burgers' equation. Compare with Figure C.6 as the lines for method A are the same in the two figures. For a description of the labels, see the beginning of this section.

of gas or traffic flow. The problem is discretized in space using centered finite differences with 71 uniformly distributed grid points. Using a second order Carleman bilinearization, we obtain a bilinear control system approximation with $A, N \in \mathbb{R}^{5112\times5112}$ and $B \in \mathbb{R}^{5112}$; see [10] for further details. Note that in this case $A$ is an asymptotically stable but non-symmetric matrix. To ensure the positive semidefiniteness of the Gramian, we scale the control matrices $N$ and $B$ with a factor $\alpha = 0.25$. We emphasize that the control law is scaled proportionally with $\frac{1}{\alpha}$ such that the dynamics remain unchanged, for further discussion see [9, Section 3.4].

The comparison is similar to the previous examples and the Figures C.6 and C.7 are analogous to the respective Figures C.1 and C.2. The problem is difficult in the sense that the singular values of the solution decay slowly. Moreover, the direct method stagnates at a relative residual norm of $5 \cdot 10^{-6}$. This is, however, less visible compared to the previous example since in general the convergence is slower.

For this example the performance of BIRKA is not significantly better than other methods, which is not surprising since the theoretical justifications for the method are not valid. ALS shows faster convergence in relative residual norm but slower convergence in relative error, as well as indications of stagnation. However, the theoretical justifications for ALS are also not valid for this example and the result is in line with the results in [25]. Regarding the rational-Krylov-type methods it seems as if method D and B has the best performance. However, method E does not provide a useful subspace for this example.

### C.6.4 Execution time experiment

We conclude the numerical examples with a small experiment comparing the execution time of different methods considered. The problems are the same as above, i.e., the heat equation, the 1D Fokker–Planck equation, and the Burger's equation. For all these we generate a BIRKA subspace of dimension 30, an ALS subspace of dimension 60, and a subspace of type A of dimension 60. The approximation properties of these spaces are similar for the heat equation, see Figure C.1. The cumulative CPU time as a function of iteration count, in the respective method, is plotted in Figure C.8. Note that for ALS and method A the iteration count corresponds to increasing the dimension of the subspace with one, since the right-hand side is rank one. However, for BIRKA the dimension of the subspace is fixed on beforehand and hence there is an irregular number of iterations, corresponding to the convergence of the fixed-point problem rather than the size of the subspace. It was, for example, mentioned above that the BIRKA iterations for subspaces of dimension 28 and 29, for the Fokker–Planck equation, did not converge to the specified tolerance in the allowed 100 iterations.

In this situation, and for the chosen parameters, BIRKA is faster for the heat equation, and slower for the Fokker–Planck equation. In the case of the Burger's equation it seems as if BIRKA is faster. However, if we take the approximation properties into account we find, by looking at Figure C.6, that a more fair comparison with method A is to consider the latter only up to iteration 30. Moreover, fixing the subspace dimension, rather than the tolerance, is (likely) advantageous for BIRKA.

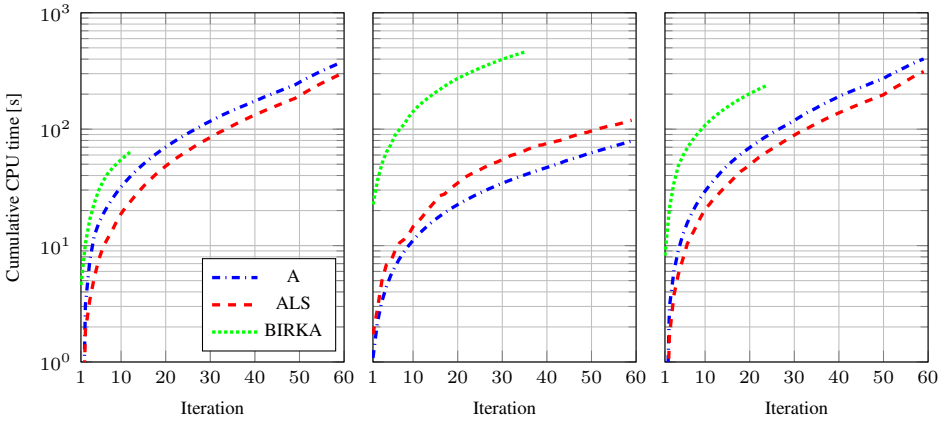



Figure C.8: Cumulative time as function of number of iterations. The plots are for the: Heat equation (left), Fokker–Planck (middle), and Burgers' equation (right).

## C.7 Conclusions and outlooks

We have proposed a rational-Krylov-type subspace for solving the generalized Lyapunov equation. Simulations indicate competitive performance, at least in the non-symmetric case where optimality statements for the other methods are no longer valid. Simulations show that methods A and F perform well for all three examples. The ALS iteration, as well as results from the literature, cf. [1], seems to indicate that subspaces of the type $(A - \sigma I - \mu N_i)^{-1}B$ could be useful. Although we have not been able to exploit this efficiently. Another generalization of the rational Krylov subspace, for general linear matrix equations, is presented in [32]. It is suggested to use subspaces of the type $(A - \sigma I)^{-1}v$, and $(N_i - \sigma I)^{-1}v$, where $v$ is a vector from the previous space. We see that more research is needed to understand the theoretical aspects of the suggested, and related, spaces.

Common for all methods studied is that they use the current residual in the iterations. Computing the residual can in itself be costly for a truly large scale problem, although approximate dominant directions can be computed in an iterative fashion, resulting in an inner–outer-type iteration. However, more research is needed to understand the consequences of such inexact subspaces.

## Acknowledgments

We wish to thank the anonymous referees, who's comments helped improve the manuscript. The authors also wish to thank Elias Jarlebring (KTH) for support and discussions.

This research project was started when the second author visited the first author at the Karl–Franzens-Universität in Graz; the kind hospitality was greatly appreciated. The visit was made possible due to the generous support from the European Model Reduction Network (COST action TD1307, STSM grant 38025).

## References

[1] M. Ahmad, U. Baur, and P. Benner. Implicit Volterra series interpolation for model reduction of bilinear systems. *J. Comput. Appl. Math.*, 316(Supplement C):15–28, 2017.

[2] S. A. Al-Baiyat and M. Bettayeb. A new model reduction scheme for k-power bilinear systems. In *Proceedings of 32nd IEEE Conference on Decision and Control*, pages 22–27, 1993.

[3] S. Baars, J. P. Viebahn, T. E. Mulder, C. Kuehn, F. W. Wubs, and H. A. Dijkstra. Continuation of probability density functions using a generalized Lyapunov approach. *J. Comput. Phys.*, 336:627–643, 2017.

[4] S. Becker and C. Hartmann. Infinite-dimensional bilinear and stochastic balanced truncation with explicit error bounds. *Math. Control Signals Systems*, 31(2):1–37, 2019.

[5] P. Benner and T. Breiten. Interpolation-based $\mathcal{H}_2$-model reduction of bilinear control systems. *SIAM J. Matrix Anal. Appl.*, 33(3):859–885, 2012.

[6] 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.

[7] P. Benner and T. Breiten. On optimality of approximate low rank solutions of large-scale matrix equations. *Syst. Control Lett.*, 67:55–64, 2014.

[8] P. Benner, Z. Bujanović, P. Kürschner, and J Saak. RADI: a low-rank ADI-type algorithm for large scale algebraic Riccati equations. *Numer. Math.*, 138(2):301–330, 2018.

[9] 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.

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

[11] T. Breiten, K. Kunisch, and L. Pfeiffer. Numerical study of polynomial feedback laws for a bilinear control problem. *Math. Control Relat. Fields*, 8(3&4):557–582, 2018.

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

[13] V. Druskin, L. Knizhnerman, and M. Zaslavsky. Solution of large scale evolutionary problems using rational Krylov subspaces with optimized shifts. *SIAM J. Sci. Comput.*, 31(5):3760–3780, 2009.

[14] V. Druskin, C. Lieberman, and M. Zaslavsky. On adaptive choice of shifts in rational Krylov subspace reduction of evolutionary problems. *SIAM J. Sci. Comput.*, 32(5):2485–2496, 2010.

[15] V. Druskin and V. Simoncini. Adaptive rational Krylov subspaces for large-scale dynamical systems. *Syst. Control Lett.*, 60(8):546–560, 2011.

[16] V. Druskin, V. Simoncini, and M. Zaslavsky. Adaptive tangential interpolation in rational Krylov subspaces for MIMO dynamical systems. *SIAM J. Matrix Anal. Appl.*, 35(2):476–498, 2014.

[17] K. Eppler and F. Tröltzsch. Fast optimization methods in the selective cooling of steel. In M. Grötschel, S. O. Krumke, and J. Rambau, editors, *Online Optimization of Large Scale Systems*, pages 185–204. Springer-Verlag, Berlin Heidelberg, 2001.

[18] G. Flagg, C. Beattie, and S. Gugercin. Convergence of the iterative rational Krylov algorithm. *Syst. Control Lett.*, 61(6):688–691, 2012.

[19] G. Flagg and S. Gugercin. Multipoint Volterra series interpolation and $\mathcal{H}_2$ optimal model reduction of bilinear systems. *SIAM J. Matrix Anal. Appl.*, 36(2):549–579, 2015.

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

[21] S. Gugercin, A. C. Antoulas, and C. Beattie. $\mathcal{H}_2$ model reduction for large-scale linear dynamical systems. *SIAM J. Matrix Anal. Appl.*, 30(2):609–638, 2008.

[22] C. Hartmann, B. Schäfer-Bung, and A. Thöns-Zueva. Balanced averaging of bilinear systems with applications to stochastic control. *SIAM J. Control Optim.*, 51(3):2356–2378, 2013.

[23] R. A. Horn and C. .R Johnson. *Topics in Matrix Analysis*. Cambridge Univ. Press, Cambridge, UK, 1st edition, 1994.

[24] E. Jarlebring, G. Mele, D. Palitta, and E. Ringh. Krylov methods for low-rank commuting generalized Sylvester equations. *Numer. Linear Algebra Appl.*, 25(6):e2176, 2018.

[25] D. Kressner and P. Sirković. Truncated low-rank methods for solving general linear matrix equations. *Numer. Linear Algebra Appl.*, 22(3):564–583, 2015.

[26] D. Kressner and C. Tobler. Krylov subspace methods for linear systems with tensor product structure. *SIAM J. Matrix Anal. Appl.*, 31(4):1688–1714, 2010.

[27] Y. Lin and V. Simoncini. Minimal residual methods for large scale Lyapunov equations. *Appl. Numer. Math.*, 72:52–71, 2013.

[28] S. Massei, D. Palitta, and L. Robol. Solving rank-structured Sylvester and Lyapunov equations. *SIAM J. Matrix Anal. Appl.*, 39(4):1564–1590, 2018.

[29] V. Mehrmann and E. Tan. Defect correction method for the solution of algebraic Riccati equations. *IEEE Trans. Autom. Control*, 33(7):695–698, 1988.

[30] R. R. Mohler and W. J. Kolodziej. An overview of bilinear system theory and applications. *IEEE Transactions on Systems, Man, and Cybernetics*, 10(10):683–688, 1980.

[31] H. Neudecker. A matrix trace inequality. *J. Math. Anal. Appl.*, 166(1):302–303, 1992.

[32] 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.

[33] S. Richter, L. D. Davis, and E. G. Collins Jr. Efficient computation of the solutions to modified Lyapunov equations. *SIAM J. Matrix Anal. Appl.*, 14(2):420–431, 1993.

[34] E. Ringh, G. Mele, J. Karlsson, and E. Jarlebring. Sylvester-based preconditioning for the waveguide eigenvalue problem. *Linear Algebra Appl.*, 542:441–463, 2018. Proceedings of the 20th ILAS Conference, Leuven, Belgium 2016.

[35] A. Ruhe. The rational Krylov algorithm for nonsymmetric eigenvalue problems. III: complex shifts for real matrices. *BIT*, 34(1):165–176, 1994.

[36] H. R. Shaker and M. Tahavori. Control configuration selection for bilinear systems via generalised Hankel interaction index array. *Internat. J. Control*, 88(1):30–37, 2015.

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

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

[39] R. A. Smith. Matrix equation $XA + BX = C$. *SIAM J. Appl. Math.*, 16(1):198–201, 1968.

[40] B. Vandereycken and S. Vandewalle. A Riemannian optimization approach for computing low-rank solutions of Lyapunov equations. *SIAM J. Matrix Anal. Appl.*, 31(5):2553–2579, 2010.

[41] L. Zhang and J. Lam. On $H_2$ model reduction of bilinear systems. *Automatica J. IFAC*, 38(2):205–216, 2002.

# Paper D

## Nonlinearizing two-parameter eigenvalue problems

# Nonlinearizing two-parameter eigenvalue problems

by

Emil Ringh, Elias Jarlebring

accepted for publication in
*SIAM Journal on Matrix Analysis and Applications*, 2021

**Abstract**

We investigate a technique to transform a linear two-parameter eigenvalue problem, into a nonlinear eigenvalue problem (NEP). The transformation stems from an elimination of one of the equations in the two-parameter eigenvalue problem, by considering it as a (standard) generalized eigenvalue problem. We characterize the equivalence between the original and the nonlinearized problem theoretically and show how to use the transformation computationally. Special cases of the transformation can be interpreted as a reversed companion linearization for polynomial eigenvalue problems, as well as a reversed (less known) linearization technique for certain algebraic eigenvalue problems with square-root terms. Moreover, by exploiting the structure of the NEP we present algorithm specializations for NEP-methods, although the technique also allows general solution methods for NEPs to be directly applied. The nonlinearization is illustrated in examples and simulations, with focus on problems where the eliminated equation is of much smaller size than the other two-parameter eigenvalue equation. This situation arises naturally in domain decomposition techniques. A general error analysis is also carried out under the assumption that a backward stable eigensolver is used to solve the eliminated problem, leading to the conclusion that the error is benign in this situation.

**Keywords:** two-parameter eigenvalue problem, nonlinear eigenvalue problem, multiparameter eigenvalue problem, iterative algorithms, implicit function theorem

## D.1 Introduction

This paper concerns the *two-parameter eigenvalue problem*: Determine nontrivial quadruplets $(\lambda, x, \mu, y) \in \mathbb{C} \times \mathbb{C}^n \times \mathbb{C} \times \mathbb{C}^m$ such that

$$0 = A_1 x + \lambda A_2 x + \mu A_3 x \tag{D.1a}$$
$$0 = B_1 y + \lambda B_2 y + \mu B_3 y, \tag{D.1b}$$

where $A_1, A_2, A_3 \in \mathbb{C}^{n\times n}$, and $B_1, B_2, B_3 \in \mathbb{C}^{m\times m}$. More specifically, with nontrivial we mean that the eigenvectors should be nonzero, i.e., $y \neq 0$ and $x \neq 0$. We denote the corresponding functions $A(\lambda,\mu) := A_1 + \lambda A_2 + \mu A_3$ and $B(\lambda,\mu) := B_1 + \lambda B_2 + \mu B_3$. This problem has been extensively studied in the literature, see, e.g., the fundamental work of Atkinson [2], and the summary of recent developments below. We assume that $m \ll n$ and that $A_1$, $A_2$ and $A_3$ are large and sparse matrices, although several theoretical contributions of this paper are valid without this assumption.

The main idea of our approach can be described as follows. We view (D.1b) as a parameterized generalized linear eigenvalue problem, where $\lambda$ is the parameter. Due to perturbation theory for eigenvalue problems, there is a family of continuous functions $\{g_i(\lambda)\}$, defined by the eigenvalues of (D.1b), where $\mu$ is the eigenvalue, of a *generalized eigenvalue problem* (GEP). More formally, for a fixed value of $\lambda \in \mathbb{C}$ the functions $g_i(\lambda) \in \mathbb{C}$ and $\phi_i(\lambda) \in \mathbb{C}^m$ can be defined, as the solution to

$$0 = (B_1 + \lambda B_2 + g_i(\lambda)B_3)\phi_i(\lambda) \tag{D.2a}$$
$$1 = c^T\phi_i(\lambda), \tag{D.2b}$$

for a given vector $c \in \mathbb{C}^m$. We explicitly introduced the normalization condition (D.2b), to uniquely define a corresponding eigenvector. The condition (D.2b) is not a restriction of generality except for the rare situation that the eigenvector is orthogonal to $c$. We prefer this condition over the standard Euclidean normalization, since the right-hand side of (D.2b) is an analytic function.

By insertion of $\mu = g_i(\lambda)$ into (D.1a), we see that a solution to (D.1) will satisfy

$$M(\lambda)x = (A_1 + \lambda A_2 + g_i(\lambda)A_3)x = 0. \tag{D.3}$$

Note that we have now eliminated $\mu$ and (D.1b), at the cost of the introduction of a nonlinear function into the eigenvalue problem. The problem $M(\lambda)x = 0$ is called a *nonlinear eigenvalue problem* (NEP). In our setting it is rather a family of NEPs, since we have a different nonlinearity for each function $g_1, \ldots, g_m$. The study of NEPs is a mature field within numerical linear algebra, and there are considerable theoretical results, as well as algorithms and software for NEPs which aim to find a selection of solutions. Note that the NEP-solvers in general only compute a subset of the eigenvalues and therefore our approach is mainly for situations where particular $(\lambda,\mu)$-values are of interest, e.g., close to a target.

We provide a theoretical characterization of the elimination procedure in Section D.2. The characterization shows that the functions are locally analytic (and not necessarily entire functions), everywhere except for certain points, which are explicitly described. Section D.3 contains new methods for (D.1) derived from NEP-methods designed for problems with local analyticity. Analysis of the conditioning of the structured perturbations corresponding to the elimination are provided in Section D.4. We provide software for the simulations, both for MATLAB and for Julia [6]. The Julia software is implemented using the data structures of the NEP-PACK software package [20], including adaption of theory for how to compute derivatives and projections. This provides new ways to solve (D.1), using the large number of NEP-solvers available in NEP-PACK. Some contributions

are also converse, i.e., we provide insight to NEPs based on the equivalence with two-parameter eigenvalue problems. For instance, in Sections D.2.2–D.2.3 we show how to transform certain NEPs with square-root nonlinearities to two-parameter eigenvalue problems. This in turn (using the operator determinants described below) allows us to transform the problem to a standard generalized eigenvalue problem, similar to companion linearization techniques for polynomial and rational eigenvalue problems.

We now summarize the NEP-results relevant for our approach. For a broad overview see the summary papers [38, 30, 50, 10], as well as the benchmark collection [4], and software packages with NEP-solvers [37, 12, 13, 20]. There are considerable theoretical works available for the NEP, in particular for polynomial eigenvalue problems. Techniques to transform polynomial NEPs to standard eigenvalue problems (known as linearization) have been completely characterized in a number of works, e.g., [27, 28] and [33]. We relate our approach to this type of linearization in Section D.2.2. In our derivation, we make explicit use of the implicit function theorem applied to the NEP. This has been done in the context of sensitivity analysis, leading to eigenvector free formulas for conditioning [1]. There are a number of algorithms available for NEPs, of which many seem to be applicable to (D.3). More specifically, we characterize the specialization of residual inverse iteration [34], which forms the basis of more recent methods such as the nonlinear Arnoldi method [49]. We also show how the infinite Arnoldi method [23] can be adapted to (D.3).

In Section D.5.2 we illustrate how two-parameter eigenvalue problems of this type can arise by the separation of domains of a boundary value problem (BVP). The domains are decoupled in a way that the discretization leads to a two-parameter eigenvalue problem. In this context, the elimination corresponds to an elimination of one of the domains. The elimination of an outer domain, in a way that directly leads to NEPs, by introduction of artificial boundary conditions is the origin of several standard NEPs in the literature, e.g., [44] and the electromagnetic cavity model in [48].

Relevant results for two-parameter eigenvalue problems can be summarized as follows. Many results for two-parameter eigenvalue problems are phrased in the more general setting of *multiparameter eigenvalue problems*. There are a number of recent efficient algorithms available, e.g., based on the Jacobi–Davidson approach [15, 17], including subspace methods in [16]. A number of generalizations of inverse iteration are derived in [36]. Our approach is based on an eigenvalue parameterization viewpoint. Eigenvalue parameterization and continuation techniques (but with an additional parameter) have been studied, e.g., in [35].

One of the most fundamental properties of two-parameter eigenvalue problems is the fact that solutions are given by the solution to a larger linear (generalized) eigenvalue problem. This is also often used in the numerical algorithms mentioned above, and to our knowledge first proposed as a numerical method in [40]. More precisely, we associate with (D.1) the *operator determinants*

$$\Delta_0 = B_2 \otimes A_3 - B_3 \otimes A_2 \tag{D.4}$$

$$\Delta_1 = B_3 \otimes A_1 - B_1 \otimes A_3 \tag{D.5}$$

$$\Delta_2 = B_1 \otimes A_2 - B_2 \otimes A_1, \tag{D.6}$$

where $\otimes$ denotes the Kronecker product. The solutions to (D.1) are (under certain assumptions) equivalent to the solutions to the two generalized eigenvalue problems

$$\Delta_1 z = \lambda \Delta_0 z \tag{D.7a}$$
$$\Delta_2 z = \mu \Delta_0 z \tag{D.7b}$$

where $z = y \otimes x$. In practice, the application of a general purpose eigenvalue solver on one of the GEPs in (D.7) yields an accurate solution for small systems. The Sylvester-like structure of the operator determinants is exploited in [29] with applications in, e.g., detection of a Hopf bifurcation. The equivalence between (D.7) and (D.1) holds under nonsingularity assumption; in particular the problem is singular if $A_3$ and $B_3$ both are singular; $A_2$ and $B_2$ both are singular; $A_2$ and $A_3$ have intersecting null-spaces; or $B_2$ and $B_3$ have intersecting null-spaces. See [2] for a precise characterization, and [24, 18] for more recent formulations.

The following matrix is often used in theory for eigenvalue multiplicity and eigenvalue conditioning, and will be needed throughout the paper. We denote

$$C_0 := \begin{bmatrix} v^H A_2 x & v^H A_3 x \\ w^H B_2 y & w^H B_3 y \end{bmatrix}, \tag{D.8}$$

where $v$ and $w$ are left eigenvectors associated with (D.1a) and (D.1b), respectively. In particular, for an (algebraically) simple eigenvalue of the two-parameter eigenvalue problem (D.1), the matrix $C_0$ is nonsingular; see [24, Lemma 3], [15, Lemma 1.1], and [18, Lemma 1]. For a simple eigenvalue, the normwise condition number for the two-parameter eigenvalue problem is expressed as a special induced matrix norm of $C_0^{-1}$, see [18, Section 4].

## D.2 Nonlinearization

### D.2.1 Existence and equivalence

The elimination of the $B$-equation (D.1b) in the two-parameter eigenvalue problem can be explicitly characterized as we describe next. Note that when $\lambda$ is viewed as a parameter, the second equation in the two-parameter eigenvalue problem is a GEP corresponding to the pencil $(-(B_1 + \lambda B_2), B_3)$:

$$-(B_1 + \lambda B_2)y = \mu B_3 y. \tag{D.9}$$

In the algorithms section, we will use GEP-eigensolvers to compute $\mu$. In order to describe when this GEP leads to an analytic well-defined parameterization we introduce the normalization $c^T y = 1$ for theoretical purposes.

The idea is based on viewing the GEP (D.9) and the normalization condition as a set of nonlinear equations in the variables $y$, $\lambda$ and $\mu$. Conditions on the existence of a parameterization is in our first result expressed in terms of the partial Jacobian, with respect to the

variables $y$ and $\mu$, of this nonlinear function. The Jacobian is

$$J(\lambda, \mu, y) := \begin{bmatrix} B(\lambda, \mu) & B_3 y \\ c^T & 0 \end{bmatrix}. \tag{D.10}$$

In the theorem that directly follows this lemma, we show how the singularity of the Jacobian is directly related to the multiplicity of the eigenvalue of the GEP (D.9).

**Lemma D.2.1** (Existence of implicit functions)**.** *Let $\lambda \in \mathbb{C}$ be given. Suppose the pencil associated with the GEP* (D.9) *is regular. Let $(\mu_i, y_i) \in \mathbb{C} \times \mathbb{C}^m$ be an eigenpair of the GEP normalized such that $c^T y_i = 1$. Moreover, assume that $J(\lambda, \mu_i, y_i)$ as given by* (D.10) *is nonsingular. Then, there exists a domain $\Omega_i \subset \mathbb{C}$, and functions analytic in this domain $g_i : \Omega_i \to \mathbb{C}$ and $\phi_i : \Omega_i \to \mathbb{C}^m$ such that*

$$-(B_1 + sB_2)\phi_i(s) = g_i(s)B_3\phi_i(s), \text{ for all } s \in \Omega_i,$$

*where $\Omega_i$ is a neighborhood of $\lambda$, and $g_i(\lambda) = \mu_i$, $\phi_i(\lambda) = y_i$.*

*Proof.* Consider the analytic function $f : \mathbb{C}^{m+2} \to \mathbb{C}^{m+1}$ given by

$$f(\lambda, \mu, y) := \begin{bmatrix} B(\lambda, \mu)y \\ c^T y - 1 \end{bmatrix}. \tag{D.11}$$

Then, as noted above, $J = \partial f / \partial(y, \mu)$. Since $f(\lambda, \mu_i, y_i) = 0$ and $J(\lambda, \mu_i, y_i)$ is nonsingular, the result follows from the complex implicit function theorem [8, Theorem I.7.6]. □

**Theorem D.2.2** ($J$-singularity)**.** *Let $\lambda \in \mathbb{C}$ be given. Assume that the pencil associated with the GEP* (D.9) *is regular. Let $(\mu_i, y_i) \in \mathbb{C} \times \mathbb{C}^m$ be an eigenpair of the GEP normalized such that $c^T y_i = 1$. Then $J(\lambda, \mu_i, y_i)$ defined in* (D.10) *is singular if and only if $\mu_i$ is a non-simple eigenvalue of the GEP.*

*Proof.* We start by proving that $J(\lambda, \mu_i, y_i)$ is singular implies that $\mu_i$ is a non-simple eigenvalue . Assume that $J(\lambda, \mu_i, y_i)$ is singular. Then there exists a nontrivial vector $\begin{bmatrix} \xi^T & \alpha \end{bmatrix}^T \in \mathbb{C}^{m+1}$ such that $J(\lambda, \mu_i, y_i) \begin{bmatrix} \xi^T & \alpha \end{bmatrix}^T = 0$. The first row gives

$$B(\lambda, \mu_i)\xi + B_3 y_i \alpha = 0, \tag{D.12}$$

and the second row gives

$$c^T \xi = 0. \tag{D.13}$$

The cases $\alpha = 0$ and $\alpha \neq 0$ are investigated separately. Assume that $\alpha = 0$, then $\xi \neq 0$ and thus (D.12) implies that $\xi$ is an eigenvector to the GEP. However, (D.13) implies that $\xi$ is not a scaling of $y_i$, hence, $\mu_i$ is not simple. Assume that $\alpha \neq 0$. Note that since the pencil is regular and $\mu_i \in \mathbb{C}$ we have that $B_3 y_i \neq 0$. Then by rescaling equation (D.12) with $1/\alpha$ we see that there exists a Jordan chain of length at least two, hence, $\mu_i$ is not simple.

To prove the converse, assume that $\mu_i$ is a non-simple eigenvalue (semisimple or non-semisimple). Choose a vector $u \in \mathbb{C}^m$ such that in the semisimple case $u$ is a second eigenvector to $\mu_i$, with any normalization; and in the non-semisimple case $u$ is the second vector of a Jordan chain of length at least two, corresponding to $\mu_i$. Let $\xi := u - (c^T u)y_i$, and note that $\xi \neq 0$ in the semisimple case. By inserting the definition of $\xi$ into equations (D.12) and (D.13) and utilizing the definition of $u$, we can see that the vectors $\begin{bmatrix}\xi^T & 0\end{bmatrix}^T$ and $\begin{bmatrix}\xi^T & 1\end{bmatrix}^T$ are nontrivial vectors in the kernel of $J(\lambda, \mu_i, y_i)$ for the semisimple and non-semisimple case, respectively. Hence, $J(\lambda, \mu_i, y_i)$ is singular. In the latter case, $Bu + B_3 y_i = 0$ since that is the Jordan chain defining the chosen $u$. □

Under the same conditions that the implicit functions exist we have the following equivalence between the solutions to the NEP (D.3) and the solutions to the two-parameter eigenvalue problem (D.1).

**Theorem D.2.3** (Equivalence)**.** *Assume the quadruplet $(\lambda, x, \mu, y) \in \mathbb{C} \times \mathbb{C}^n \times \mathbb{C} \times \mathbb{C}^m$ is such that $c^T y = 1$, the pencil associated with the GEP* (D.9) *is regular, and $J(\lambda, \mu, y)$ defined in* (D.10) *is nonsingular. Then, $(\lambda, x, \mu, y)$ is a solution to* (D.1) *if and only if $(\lambda, x)$ is a solution to the NEP* (D.3) *for one pair of functions $(g_i(\lambda), \phi_i(\lambda)) = (\mu, y)$ which satisfies* (D.2)*, where $g_i$ and $\phi_i$ are the functions defined in Lemma D.2.1.*

*Proof.* To prove the forward implication direction suppose $(\lambda, x, \mu, y)$ is a solution to (D.1). From Lemma D.2.1, there are functions $g$ and $\phi$ such that $g(\lambda) = \mu$ and $\phi(\lambda) = y$. Therefore, (D.3) is satisfied for that pair $(g(\lambda), \phi(\lambda))$.

To prove the backward implication direction suppose $(\lambda, x)$ is a solution to (D.3) for a given pair $(g(\lambda), \phi(\lambda))$. Then $(\lambda, x, \mu, y) = (\lambda, x, g(\lambda), \phi(\lambda))$ is a solution to (D.1). More precisely, (D.1a) is satisfied since (D.3) is, and (D.1b) is satisfied due to (D.2). □

The theorems above can be further interpreted as follows. A direct consequence of Lemma D.2.1 and Theorem D.2.2 is that, if the pencil is regular then the simple eigenvalues of the GEP (D.9) are analytic in a region around the point $\lambda$. Hence, in this sense, there exists a nonlinearization. We now further discuss the assumptions in the theory.

**Remark D.2.4** (Theory assumptions)**.** *Note that the problem* (D.9) *is a GEP, whose properties are independent of normalization, and for every eigenpair of the GEP there exists a vector $c$ not orthogonal to the eigenvector. The assumption $c^T y \neq 0$ is therefore not a restriction of generality.*

*If eigenvalues of the GEP have multiplicity greater than one, the theory does not predict an analytic nonlinearization. Moreover, from perturbation theory we know that there are algebraic multivalued functions which can have branch point singularities. Hence, at such a point, there are corresponding nonlinear functions and a (multivalued) nonlinearization exists in this sense. We have restricted the theory to simple eigenvalues for simplicity.*

*The theorems are based on the assumption that the pencil is regular for a fixed $\lambda \in \mathbb{C}$. The pencil can be singular, e.g., if $B_1 + \lambda B_2$ and $B_3$ have intersecting null-spaces. If the pencil is singular, then for each $\mu$ there exists a nonzero vector $y \in \mathbb{C}^m$ such that $(B_1 + \lambda B_2 + \mu B_3)y = 0$. Since all values of $\mu$ satisfies the GEP, the equation does not*

*define $\mu$ as a function of $\lambda$. The assumption is thus a limitation of the method since it cannot directly be applied to these eigenvalues. However, situations with a singular pencil may often be approached directly, and it is possible that a whole set of eigenvalues for the two-parameter eigenvalue problem can be found by solving* (D.1a) *for $\mu$, while keeping $\lambda$ fixed. The situation can be exemplified by the extreme case where $B_3 = 0$. Then values of $\lambda$ can be determined by* (D.1b)*, independently of $\mu$, and the latter can then be found by solving* (D.1a) *with the corresponding fixed $\lambda$-values. In this case the $\lambda$-values are exactly those that make the pencil $(-(B_1 + \lambda B_2), 0)$ singular.*

### D.2.2 Nonlinearizations leading to quadratic eigenvalue problems

We first illustrate the theory in the previous section with an implicitly defined function which can be derived explicitly. Consider the two-parameter eigenvalue problem

$$0 = A_1 x + \lambda A_2 x + \mu A_3 x \tag{D.14a}$$

$$0 = \left( \begin{bmatrix} 0 & 0 \\ 0 & -1 \end{bmatrix} + \lambda \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} + \mu \begin{bmatrix} -1 & 0 \\ 0 & 0 \end{bmatrix} \right) y, \tag{D.14b}$$

for general matrices $A_1$, $A_2$ and $A_3$. The second row in (D.14b) implies that the elements in the vector $y^T = \begin{bmatrix} y_1 & y_2 \end{bmatrix}$ are related by $y_2 = \lambda y_1$. The first row in (D.14b) becomes $\lambda^2 y_1 - \mu y_1 = 0$. Hence, since $y_1 \neq 0$, we have $\mu = \lambda^2$ and (D.14a) becomes

$$0 = A_1 x + \lambda A_2 x + \lambda^2 A_3 x. \tag{D.15}$$

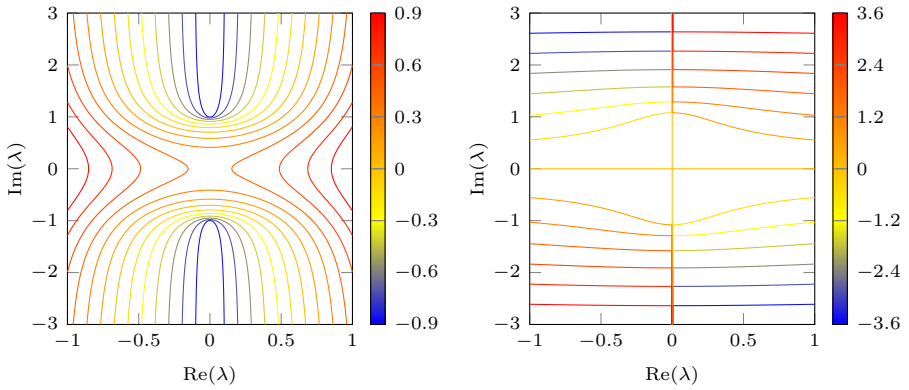
This problem is commonly known as the quadratic eigenvalue problem, which has been extensively studied in the literature [45]. The example shows that the two-parameter eigenvalue problem (D.14) can be nonlinearized to a quadratic eigenvalue problem. Moreover, the determinant operator equation (D.7a) leads to the equation

$$\begin{bmatrix} -A_1 & 0 \\ 0 & A_3 \end{bmatrix} z = \lambda \begin{bmatrix} A_2 & A_3 \\ A_3 & 0 \end{bmatrix} z,$$

which is a particular companion linearization of (D.15). (It is in fact a symmetry preserving linearization [45, Section 3.4].) Many of the linearizations of polynomial eigenvalue problems given in [28] can be obtained in a similar fashion. Since, the second equation (D.1b) can be expressed as $\det(B(\lambda, \mu)) = 0$, which is a bivariate polynomial, this example is consistent with the bivariate viewpoint of companion linearizations in [33]. Some higher-degree polynomials can be constructed analogously to above, e.g., the polynomial eigenvalue problem $A_1 + \lambda A_2 + \lambda^m A_3$. However, the general higher-degree polynomial eigenvalue problem does not seem to fit into the class of two-parameter eigenvalue problems.

### D.2.3 Nonlinearization leading to algebraic functions

The previous example can be modified in a way that it leads to algebraic functions, which is also the generic situation. Nontrivial solutions to (D.1b) satisfy $\det(B(\lambda, \mu)) = 0$, which




(a) $\mathrm{Re}(g_+(\lambda))$ (b) $\mathrm{Im}(g_+(\lambda))$

Figure D.1: The square-root nonlinearity illustrated in the example in Section D.2.3, with $a = 3$, $b = 2$, $c = -1$, $d = -2$, $e = 2$ and $f = 1$. We observe a square-root singularity at $\lambda = \pm\sqrt{-3/2}$ which are the roots of $p(\lambda)$.

is a bivariate polynomial. Therefore, the functions $g_i(\lambda)$ are roots of a polynomial, where the coefficients are polynomials in $\lambda$, i.e., $g_i$ are algebraic functions. The generic situation can be seen from the case where $m = 2$:

$$0 = (A_1 + \lambda A_2 + \mu A_3)x \tag{D.16a}$$

$$0 = \left(\begin{bmatrix} a & b \\ c & d \end{bmatrix} + \lambda \begin{bmatrix} 0 & e \\ f & 0 \end{bmatrix} + \mu \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right) y. \tag{D.16b}$$

We obtain that $\mu$ is the root of a polynomial, where the coefficients depend on $\lambda$, i.e.,

$$0 = (\mu + a)(\mu + d) - (c + \lambda f)(b + \lambda e).$$

The explicit solutions to this quadratic equation are given by

$$\mu = g_\pm(\lambda) = -\frac{a+d}{2} \pm \sqrt{\frac{(a+d)^2}{4} - ad + (b + \lambda e)(c + \lambda f)}.$$

We see by insertion of $\mu = g_\pm$ into (D.16a) that the nonlinearization of (D.16) is a NEP with an algebraic nonlinearity. The function $g_+$ is illustrated in Figure D.1.

Several general conclusions can be made from this example. Note that the variables $a, b, c, d, e, f$ can be used for fitting of any function $\sqrt{p(\lambda)}$ where $p$ is a polynomial of degree two. Therefore, we can now reverse the nonlinearization, and for the trivial case $a = d = 0$ we directly obtain the following characterization.

**Lemma D.2.5** (Two-parameterization of an algebraic NEP)**.** *Suppose* $p(\lambda) = (b+\lambda e)(c+\lambda f)$ *is given, and let* $a = d = 0$*. If* $(\lambda, x)$ *is a solution to the NEP*

$$(A_1 + \lambda A_2 + \sqrt{p(\lambda)}A_3)x, \tag{D.17}$$

*then* $(\lambda, x, \mu, y)$ *satisfies the two-parameter eigenvalue problem equation* (D.16) *with* $\mu := \sqrt{p(\lambda)}$ *and*

$$y := \begin{bmatrix} \sqrt{b+\lambda e} \\ -\sqrt{c+\lambda f} \end{bmatrix}.$$

A further consequence of the lemma is that problems of the type (D.17) can be linearized to a GEP using the determinant operators (D.7). More precisely, the combination of Lemma D.2.5 and (D.7) shows that (D.17) can be solved by computing solutions to

$$\begin{bmatrix} A_1 & -bA_3 \\ -cA_3 & A_1 \end{bmatrix} z = \lambda \begin{bmatrix} -A_2 & eA_3 \\ fA_3 & -A_2 \end{bmatrix} z.$$

The fact that algebraic NEPs can be linearized was already pointed out in the conference presentation [39], using techniques not involving two-parameter eigenvalue problems.

Also note that the functions $g_i(\lambda)$ have branch-point singularities. This is the generic situation and we can therefore never expect that the nonlinearizations are entire functions in general. The singularities restrict the performance of many methods, as we will see in the simulations. The implications of singularities in practice is well-known in quantum chemistry, where parameterized eigenvalue problems is a fundamental tool and the singularities are referred to as intruder states [11, Chapter 14]. In that context, methods for computing the closest singularity (which limits the performance of the method) are given in [21, 25].

## D.3 Algorithm specializations

### D.3.1 Derivative based algorithms

Many NEP-algorithms are based on derivatives of $M$. We will now illustrate how to efficiently and reliably access the derivatives of the NEP stemming from a nonlinearization of a two-parameter eigenvalue problem. As a representative first situation we consider the augmented Newton method; see [38, 47]. It can be derived by an elimination of the correction equation in Newton's method, and leads to separate eigenvalue and eigenvector update formulas expressed as

$$x_{k+1} = \alpha_k M(\lambda_k)^{-1} M'(\lambda_k) x_k \tag{D.18a}$$
$$\alpha_k^{-1} = d^T M(\lambda_k)^{-1} M'(\lambda_k) x_k \tag{D.18b}$$

and $\lambda_{k+1} = \lambda_k - \alpha_k$, where $d \in \mathbb{C}^n$ is a normalization vector. In an implementation, one takes advantage of the fact that the same linear system appears twice, and only needs to be computed once. The iteration has appeared in many variations with different names, e.g., inverse iteration [41] and Newton's method [46].

In order to apply (D.18) we clearly need the derivative of $M$ defined in (D.3), which can be obtained directly if we can compute the derivative of the implicitly defined function $g_i$. Note that the functions $g_i(\lambda)$ (as well as the auxiliary vector $\phi_i(\lambda)$) can be evaluated by solving the GEP (D.9), and normalizing according to $c^T y_i = 1$. Since the functions are analytic in general, their respective derivatives exist. They can be computed according to the following result, which gives a recursion that can compute the $k$th derivative by solving $k$ linear systems of dimension $(m+1)\times(m+1)$. The adaption of the theorem and (D.18) into an algorithm results in Algorithm D.1.

**Theorem D.3.1** (Explicit recursive form for derivatives)**.** *Let $\lambda \in \mathbb{C}$ be given. Assume that the pencil associated with the GEP* (D.9) *is regular, and that $(\mu_i, y_i) \in \mathbb{C}\times\mathbb{C}^m$ is a solution to the GEP with $y_i$ normalized as $c^T y_i = 1$. Moreover, assume that $J(\lambda, \mu_i, y_i)$ is invertible, where $J$ is defined in* (D.10)*. Let $g_i$ and $\phi_i$ be the functions defined in Lemma D.2.1, then the $k$th derivative, $k = 1, 2, \ldots$, of $g_i$ and $\phi_i$ are given by*

$$\begin{bmatrix} \phi_i^{(k)}(\lambda) \\ g_i^{(k)}(\lambda) \end{bmatrix} = J(\lambda, \mu_i, y_i)^{-1} \begin{bmatrix} -b_k \\ 0 \end{bmatrix}, \tag{D.19}$$

*where*

$$b_k = kB_2\phi_i^{(k-1)}(\lambda) + \sum_{j=1}^{k-1} \binom{k}{j} g_i^{(k-j)}(\lambda) B_3 \phi_i^{(j)}(\lambda).$$

*Proof.* We again consider the analytic function $f$ given by (D.11). By Lemma D.2.1 we know that $g_i$ and $\phi_i$ are analytic around $\lambda$, and that $f(\lambda, g_i(\lambda), \phi_i(\lambda)) = 0$ in a neighborhood of $\lambda$. Taking the $k$th implicit derivative with respect to $\lambda$ gives

$$0 = \frac{d^k}{d\lambda^k} \begin{bmatrix} B_1\phi_i(\lambda) \\ c^T\phi_i(\lambda) - 1 \end{bmatrix} + \frac{d^k}{d\lambda^k} \begin{bmatrix} \lambda B_2\phi_i(\lambda) \\ 0 \end{bmatrix} + \frac{d^k}{d\lambda^k} \begin{bmatrix} g_i(\lambda) B_3\phi_i(\lambda) \\ 0 \end{bmatrix}.$$

The first term is found directly as

$$\frac{d^k}{d\lambda^k} \begin{bmatrix} B_1\phi_i(\lambda) \\ c^T\phi_i(\lambda) - 1 \end{bmatrix} = \begin{bmatrix} B_1\phi_i^{(k)}(\lambda) \\ c^T\phi_i^{(k)}(\lambda) \end{bmatrix}.$$

The second and third term can be calculated, by using Leibniz derivation rule for products, to be

$$\frac{d^k}{d\lambda^k} \begin{bmatrix} \lambda B_2\phi_i(\lambda) \\ 0 \end{bmatrix} = \begin{bmatrix} \lambda B_2\phi_i^{(k)}(\lambda) \\ 0 \end{bmatrix} + \binom{k}{k-1} \begin{bmatrix} B_2\phi_i^{(k-1)}(\lambda) \\ 0 \end{bmatrix},$$

and

$$\frac{d^k}{d\lambda^k} \begin{bmatrix} g_i(\lambda) B_3\phi_i(\lambda) \\ 0 \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^{k-1} \binom{k}{j} g_i^{(k-j)}(\lambda) B_3\phi_i^{(j)}(\lambda) \\ 0 \end{bmatrix} + \begin{bmatrix} g_i^{(k)}(\lambda) B_3\phi_i(\lambda) \\ 0 \end{bmatrix} + \begin{bmatrix} g_i(\lambda) B_3\phi_i^{(k)}(\lambda) \\ 0 \end{bmatrix}.$$

We emphasize the recursion: All derivatives up to order $k-1$ can be considered known since these do not depend on the higher derivatives. Collecting the known terms in the right-hand side gives the result. □

**Remark D.3.2.** *As a special case of Theorem D.3.1, for $k = 1$, we find that $g_i'(\lambda) = -\frac{w_i^H B_2 y_i}{w_i^H B_3 y_i}$ where $w_i$ is the corresponding left eigenvector to the eigenpair $(\mu_i, y_i)$. It follows from multiplying the first block-row of equation system* (D.19) *from the left with $w_i^H$. The result is a special case of well known perturbation analysis for generalized eigenvalue problems [14, Theorem 2.5]. In our case $g'(\lambda)$ is the perturbation of the eigenvalue $\mu$ with respect to $\lambda$ in the GEP* (D.9)*. More precisely, a perturbation of the matrix $-(B_1 + \lambda B_2)$ with the structured perturbation $\varepsilon B_2$.*

*Specifically, the closed form of $g_i'(\lambda)$ means that the derivative of the NEP* (D.3) *can be written in closed form, as*

$$M'(\lambda) = A_2 - \frac{w_i^H B_2 y_i}{w_i^H B_3 y_i} A_3.$$

*For methods only requiring the first derivative of $M(\lambda)$, the above expression can be used instead of* (D.19)*. However, that requires the computations of the left eigenvector of the GEP. We will need the expression for theoretical purposes in Section D.4.*

**Algorithm D.1:** Augmented Newton method for nonlinearized two-parameter eigenvalue problem

**input :** Starting values $\lambda_0 \in \mathbb{C}$ and $x_0 \in \mathbb{C}^n$
**output:** Approximations of eigenpairs of (D.3)

1 **for** $k = 1, 2, 3, \ldots,$ **do**
2 Compute $g_i(\lambda_k) := \mu$ from the GEP (D.9) with $c$-normalized eigenvector $y \in \mathbb{C}^m$
3 **if** $\|A(\lambda_k, \mu)x_k\| \leq TOL$ **then**
4 **break**
5 Compute $g_i'(\lambda_k)$ by computing $b_1 = B_2 y$ and solving the linear system of equations (D.19)
6 Compute $u = M(\lambda_k)^{-1} M'(\lambda_k) x_k$ by using the results in Steps 2–5
7 Compute $\alpha_k = (d^T u)^{-1}$
8 Compute $x_{k+1} = \alpha_k y$
9 Compute $\lambda_{k+1} = \lambda_k - \alpha_k^{-1}$

The family of methods in [23, 22, 32] (flavors of the infinite Arnoldi method) also

requires derivative information. These methods require computation of quantities such as

$$\begin{aligned} z_0 &= M(\sigma)^{-1}(M'(\sigma)x_1 + \cdots + M^{(p)}(\sigma)x_p) \\ &= M(\sigma)^{-1}(A_1x_1 + A_2\sum_{j=1}^{p} g^{(j)}(\sigma)x_j), \end{aligned}$$

where $x_1, \ldots, x_p$ are given vectors. The computation requires higher derivatives of $g_i$. However, $\sigma$ is unchanged throughout the iteration and therefore the matrix in the linear system for derivative computation (D.19) is unchanged. Hence, all needed derivatives can be computed by solving an additional linear system. If $m \ll n$, this will in general not be computationally demanding. We also note that these fixed-shift methods choose a branch $g_i$ in the initial solution of the GEP (D.9), and then stay on that branch.

**Remark D.3.3.** *There are in general $m$ different branches of the nonlinearization. Each branch corresponds to a different eigenvalue, $g_i(\lambda) := \mu_i$, $i = 1, \ldots, m$, of the GEP* (D.9)*. In a method, at each evaluation of $g_i(\lambda)$, a branch needs to be chosen. Practical ways to choose a branch are:*

*(i). Closest to the previous value, i.e.,* $\arg\min_i\{|g_i(\lambda_k) - g_i(\lambda_{k-1})|\}$.

*(ii). Closest to a specific target value $\mu_*$, i.e.,* $\arg\min_i\{|g_i(\lambda_k) - \mu_*|\}$.

*(iii). To minimize the residual norm of the NEP* (D.3)*, i.e.,* $\arg\min_i\{\|(A_1 + \lambda_k A_2 + g_i(\lambda_k)A_3)x_k\|\}$.

*We use option (iii) in the simulations in Section D.5.*

Note that the way the iterate $\lambda_k$ is updated depends on the NEP-algorithm and is in that sense independent of how the branch $g_i(\lambda_k)$ is chosen. Hence, the resulting method can either aim for eigenvalues $(\lambda, \mu)$ close to some joint target $(\lambda_*, \mu_*)$, or close to some target $\lambda_*$.

### D.3.2 Projection methods

Many NEP-algorithms require the computation of a projected problem

$$W^T M(\lambda) V z = 0 \tag{D.20}$$

where $V, W \in \mathbb{C}^{n\times p}$ are orthogonal matrices. The problem (D.20) is again a NEP, but of smaller size. This can be viewed as a Petrov–Galerkin projection of the spaces spanned by the columns of $V$ and $W$. The projection is sometimes called subspace acceleration (or the nonlinear Rayleigh–Ritz procedure), since it is often used to improve properties of a more basic algorithm, e.g., the nonlinear Arnoldi method [49], Jacobi–Davidson methods [7, 5], block preconditioned harmonic projection methods [51], the infinite Lanczos method [31], and many more.

In order to give access to these methods, we need to provide a way to solve (D.20) for our nonlinearized problem. Fortunately, the projected problem stemming from the nonlinearized two-parameter eigenvalue problem, i.e.,

$$(W^T A_1 V + \lambda W^T A_2 V + g_i(\lambda) W^T A_3 V)z = 0, \tag{D.21}$$

has a structure which suggests straightforward methods for the projected problem. This is because the projected NEP has the same structure as the nonlinearized two-parameter eigenvalue problem, and can therefore be lifted back to a two-parameter eigenvalue problem, but now of much smaller size. We can then use general methods for two-parameter eigenvalue problems. This is directly observed from the fact that (D.21) is the nonlinearization of a two-parameter eigenvalue problem with projected $A$-matrices. It is made more precise in the following result.

**Corollary D.3.4** (Projected nonlinearized problem)**.** *Assume the quadruplet* $(\lambda, z, \mu, y) \in \mathbb{C} \times \mathbb{C}^p \times \mathbb{C} \times \mathbb{C}^m$ *is such that* $c^T y = 1$*, the pencil associated with the GEP* (D.9) *is regular, and* $J(\lambda, \mu, y)$ *defined in* (D.10) *is nonsingular. Then,* $(\lambda, z, \mu, y)$ *is a solution to the two-parameter eigenvalue problem*

$$\begin{aligned} 0 &= W^T A_1 V z + \lambda W^T A_2 V z + \mu W^T A_3 V z && \text{(D.22a)} \\ 0 &= B_1 y + \lambda B_2 y + \mu B_3 y && \text{(D.22b)} \end{aligned}$$

*if and only if* $(\lambda, z)$ *is a solution to* (D.21) *for one pair of functions* $(g(\lambda), \phi(\lambda)) = (\mu, y)$ *which satisfies* (D.2)*.*

*Proof.* This follows directly from the application of Theorem D.2.3 on the projected problem (D.22) and the NEP (D.21). □

If the projection space is small $p \ll n$, and $m \ll n$, we may even solve the two-parameter eigenvalue problem using the operator determinant eigenvalue equations (D.7) or [15, Algorithm 2.3].

The situation $p = 1$ implies that the projected problem is a scalar problem, and reduces to the so-called Rayleigh functional. There are several methods based on the Rayleigh functional, e.g., residual inverse iteration [34], and variational principle based approaches such as [43] and references therein. The fact that the projected problem is scalar and linear allows us to eliminate one of the variables, and we find that the other variable is a solution to the generalized eigenvalue problem. The following corollary specifies the formulas more precisely when $\mu$ is eliminated, and the adaption of the result into the residual inverse iteration is given in Algorithm D.2.

**Corollary D.3.5.** *Assume that* $w^T A_3 v \neq 0$*. A solution,* $\lambda$*,* $\mu \in \mathbb{C}$ *and* $y \in \mathbb{C}^m$*, to the projected NEP* (D.21) *with* $p = 1$ *can be characterized as follows. The tuple* $(\lambda, y)$ *is a solution to the GEP*

$$((w^T A_3 v)B_1 - (w^T A_1 v)B_3)y = \lambda((w^T A_2 v)B_3 - (w^T A_3 v)B_2)y, \tag{D.23}$$

*and $\mu$ is given by*

$$\mu = -\frac{w^T A_1 v + \lambda w^T A_2 v}{w^T A_3 v}. \tag{D.24}$$

*Proof.* This is derived from a special case of Corollary D.3.4 where $p = 1$. Assuming that $w^T A_3 v \neq 0$, the relation (D.22a) with $W = w$ and $V = v$ can be solved for $\mu$ resulting in the relation (D.24). By inserting this relation into (D.22b) we obtain the GEP (D.23). □

**Algorithm D.2:** Resinv for nonlinearized two-parameter eigenvalue problem

**input :** Approximate eigenvector $x_0 \in \mathbb{C}^n$, shift $\sigma \in \mathbb{C}$, right Rayleigh functional vector $w \in \mathbb{C}^n$

**output:** Approximations of eigenpairs of (D.3)

1 Compute $M(\sigma)$ and factorize
**while** *not converged* **do**
2 Compute $\lambda_{k+1} = \lambda$ by solving the GEP (D.23) for $v = x_k$
3 Compute $\mu$ from (D.24) with $v = x_k$
4 Compute $z := M(\lambda_{k+1})x_k = A_0 x_k + \lambda_{k+1} A_1 x_k + \mu A_2 x_k$
5 Compute correction $u_{k+1} = x_k - M(\sigma)^{-1} z$ using the factorization computed in Step 1
6 Normalize $x_{k+1} = u_{k+1}/\|u_{k+1}\|$

**Remark D.3.6.** *In Corollary D.3.5 we have assumed that $w^T A_3 v \neq 0$. There is an analogous formula that can be used when $w^T A_2 v \neq 0$, and $\lambda$ is eliminated. Then $(\mu, y)$ is a solution to the GEP*

$$((w^T A_2 v)B_1 - (w^T A_1 v)B_2)y = \mu((w^T A_3 v)B_2 - (w^T A_2 v)B_3)y,$$

*and $\lambda$ is given by*

$$\lambda = -\frac{w^T A_1 v + \mu w^T A_3 v}{w^T A_2 v}.$$

*For completeness we also consider the assumption $w^T A_2 v = w^T A_3 v = 0$. There are two cases: First, if $w^T A_1 v \neq 0$, then there is no solution to the projected problem* (D.21)*. Second, if $w^T A_1 v = 0$, then equation* (D.21) *is satisfied for any value $\lambda$. Hence, for any fixed $\lambda$, $(\mu, y)$ can be taken as any solution to the GEP* (D.22b)*.*

**Remark D.3.7.** *In Step 2 of Algorithm D.2 a specific value has to be selected for $\lambda_{k+1}$, and there are in general $m$ different values to choose from. The situation is inherent to the algorithm and the literature suggests to choose the value closest to the previous iterate, i.e., $\arg\min_i\{|\lambda_{k+1}^{(i)} - \lambda_k|\}$, see, e.g., [34, 10]. This is the strategy used in the simulations in Section D.5.*

## D.4 Conditioning and accuracy

In order to characterize when the elimination procedure works well, we now analyze how the technique behaves subject to perturbations. As a consequence of this we can directly conclude how backward stable computation of $g$ influences the accuracy (Section D.4.2).[1]

### D.4.1 Conditioning as a nonlinear eigenvalue problem

Standard results for the condition number of NEPs can be used to analyze perturbations with respect to the $A$-matrices. More precisely, for $\lambda \in \mathbb{C}$ we define

$$\kappa_A(\lambda) := \limsup_{\varepsilon \to 0} \left\{ \frac{|\Delta\lambda|}{\varepsilon} : \|\Delta A_j\| \leq \varepsilon \alpha_j, j = 1,2,3 \right\},$$

where $\alpha_j$ are scalars for $j = 1,2,3$, and $\Delta\lambda$ is such that

$$0 = (A_1 + \Delta A_1 + (\lambda + \Delta\lambda)(A_2 + \Delta A_2) + g(\lambda + \Delta\lambda)(A_3 + \Delta A_3))(x + \Delta x), \tag{D.25}$$

where additionally we require that $\|\Delta x\| \to 0$ and $|\Delta\lambda| \to 0$ as $\varepsilon \to 0$, cf. [14, p. 499] Then we know (see, e.g., [1]) that

$$\kappa_A(\lambda) = \|v\|\|x\| \frac{\alpha_1 + |\lambda|\alpha_2 + |g(\lambda)|\alpha_3}{|v^H M'(\lambda) x|}, \tag{D.26}$$

where $v, x$ are the corresponding left and right eigenvectors. In the following we will establish how this formula is modified when we also consider perturbations in the $B$-matrices. Note that this implies that the function $g$ is also perturbed and we cannot directly use the standard result. We therefore define, for $\lambda \in \mathbb{C}$, the condition number

$$\kappa(\lambda) := \limsup_{\varepsilon \to 0} \left\{ \frac{|\Delta\lambda|}{\varepsilon} : \|\Delta A_j\| \leq \varepsilon \alpha_j, j = 1,2,3 \text{ and } \|\Delta B_j\| \leq \varepsilon \beta_j, j = 1,2,3 \right\},$$

where $\beta_j$ are scalars for $j = 1,2,3$, and $\Delta\lambda$ fulfills (D.25) but with a perturbed $g$, i.e., $\mu + \Delta\mu = g(\lambda + \Delta\lambda)$, such that

$$0 = (B_1 + \Delta B_1 + (\lambda + \Delta\lambda)(B_2 + \Delta B_2) + (\mu + \Delta\mu)(B_3 + \Delta B_3))(y + \Delta y) \tag{D.27a}$$

$$1 = c^T(y + \Delta y), \tag{D.27b}$$

where additionally we require that $\|\Delta y\| \to 0$ and $|\Delta\mu| \to 0$ as $\varepsilon \to 0$. The definitions can be used both for *absolute* and *relative condition* numbers by setting $\alpha_j = \beta_j = 1$ or $\alpha_j = \|A_j\|$, $\beta_j = \|B_j\|$ for $j = 1,2,3$, respectively.

[1] For notational convenience the $i$ index on $g_i$ is dropped in this section.

As an intermediate step we first consider the perturbation of $\mu \in \mathbb{C}$ subject to perturbations in the $B$-matrices and fixed perturbations in $\lambda \in \mathbb{C}$ by analyzing

$$\kappa_g(\lambda) := \limsup_{\varepsilon \to 0} \left\{ \frac{|\Delta\mu|}{\varepsilon} : |\Delta\lambda| \leq \varepsilon\gamma \text{ and } \|\Delta B_j\| \leq \varepsilon\beta_j, j = 1, 2, 3 \right\},$$

where $\gamma$ is a scalar, and $\Delta\mu$ satisfies (D.27) for a given $\lambda$ . The following result shows that $\kappa_g$ can be expressed as a sum of perturbations associated with the $B$-matrices and perturbations associated with $\lambda$.

**Lemma D.4.1.** *Let $\lambda \in \mathbb{C}$ be given. Suppose the pencil associated with the GEP* (D.9) *is regular, and that $g(\lambda) = \mu \in \mathbb{C}$ is a simple eigenvalue of the GEP with $w$ and $y$ being corresponding left and right eigenvectors, respectively. Then,*

$$\kappa_g(\lambda) = \kappa_{g,B}(\lambda) + \kappa_{g,\lambda}(\lambda),$$

*where*

$$\kappa_{g,B}(\lambda) = \|w\|\|y\| \frac{\beta_1 + |\lambda|\beta_2 + |g(\lambda)|\beta_3}{|w^H B_3 y|} \qquad \textit{and} \qquad \kappa_{g,\lambda}(\lambda) = \gamma \frac{|w^H B_2 y|}{|w^H B_3 y|}.$$

*Proof.* Since $\mu$ is a simple eigenvalue of the GEP (D.9), the eigenvalue and eigenvector are analytic, and therefore $\Delta y = \mathcal{O}(\varepsilon)$ when all the perturbations are $\mathcal{O}(\varepsilon)$. Moreover, since $\mu$ is a simple finite eigenvalue, then $w^H B_3 y \neq 0$. By collecting all the higher order terms the perturbed GEP (D.27a) can thus be written as

$$\left(\Delta B_1 + \lambda\Delta B_2 + \Delta\lambda B_2 + \mu\Delta B_3 + \Delta\mu B_3\right) y + B(\lambda, \mu)\Delta y = \mathcal{O}(\varepsilon^2).$$

Multiplying with $w^H$ from the left, solving for $\Delta\mu$, and dividing with $\varepsilon$ gives that

$$\frac{\Delta\mu}{\varepsilon} = -\frac{w^H\Delta B_1 y + \lambda w^H \Delta B_2 y + \Delta\lambda w^H B_2 y + \mu w^H \Delta B_3 y}{\varepsilon w^H B_3 y} + \mathcal{O}(\varepsilon). \tag{D.28}$$

An upper bound is thus found as

$$\frac{\Delta\mu}{\varepsilon} \leq \|w\|\|y\| \frac{\beta_1 + |\lambda|\beta_2 + |\mu|\beta_3}{|w^H B_3 y|} + \gamma \frac{|w^H B_2 y|}{|w^H B_3 y|} + \mathcal{O}(\varepsilon).$$

It remains to show that the bound can be attained. This follows from considering $\hat{B} = wy^H/\|w\|\|y\|$, and inserting

$$\Delta B_1 = -\varepsilon\beta_1\hat{B} \qquad \Delta B_2 = -\varepsilon\frac{\overline{\lambda}}{|\lambda|}\beta_2\hat{B}$$

$$\Delta B_3 = -\varepsilon\frac{\overline{g(\lambda)}}{|g(\lambda)|}\beta_3\hat{B} \qquad \Delta\lambda = -\varepsilon\frac{\overline{w^H B_2 y}}{|w^H B_2 y|}\frac{|w^H B_3 y|}{\overline{w^H B_3 y}}\gamma,$$

into (D.28). □

Using the intermediate result we can now show that the condition number $\kappa(\lambda)$ is the sum of the standard condition number of NEPs and a term representing perturbations in $g$ generated by perturbations in the $B$-matrices, i.e., $\kappa_{g,B}(\lambda)$.

**Theorem D.4.2.** *Let $\lambda \in \mathbb{C}$ be a simple eigenvalue of the NEP* (D.3) *with $v$ and $x$ being corresponding left and right eigenvectors, respectively. Moreover, for this $\lambda$, suppose the pencil associated with the GEP* (D.9) *is regular, and $g(\lambda) = \mu \in \mathbb{C}$ is a simple eigenvalue of the GEP with $w$ and $y$ being corresponding left and right eigenvectors, respectively.* [2] *Then,*

$$\kappa(\lambda) = \kappa_A(\lambda) + \kappa_{g,B}(\lambda)\frac{|v^H A_3 x|}{|v^H M'(\lambda) x|},$$

*where $\kappa_A(\lambda)$ is given by* (D.26).

*Proof.* Recall the assumptions that the NEP (D.3), i.e., $M$, is analytic, that $\lambda$ is a simple eigenvalue of the NEP, and that $\mu$ is a simple eigenvalue of the GEP (D.9). Hence, the eigenvalues and eigenvectors are analytic, and therefore $\Delta x = \mathcal{O}(\varepsilon)$ when all the perturbations are $\mathcal{O}(\varepsilon)$. Moreover, we note that it also implies that $v^H M'(\lambda) x \neq 0$ and $w^H B_3 y \neq 0$. By using that $g(\lambda + \Delta\lambda) = g(\lambda) + \Delta\mu$ and collecting all the higher order terms, the perturbed NEP (D.25) can therefore be written as

$$(\Delta A_1 + \lambda \Delta A_2 + \Delta\lambda A_2 + g(\lambda)\Delta A_3 + \Delta\mu A_3)x + M(\lambda)\Delta x = \mathcal{O}(\varepsilon^2).$$

Multiplying with $v^H$ from the left, expanding $\Delta\mu$ according to (D.28), solving for $\Delta\lambda$, and dividing with $\varepsilon$, gives that

$$\frac{\Delta\lambda}{\varepsilon} = -\frac{v^H \Delta A_1 x + \lambda v^H \Delta A_2 x + g(\lambda) v^H \Delta A_3 x + \theta_{g,B}(\lambda) v^H A_3 x}{\varepsilon v^H \left(A_2 - \frac{w^H B_2 y}{w^H B_3 y} A_3\right) x} + \mathcal{O}(\varepsilon), \quad \text{(D.29)}$$

where $\theta_{g,B}(\lambda) := -(w^H \Delta B_1 y + \lambda w^H \Delta B_2 y + g(\lambda) w^H \Delta B_3 y)/(w^H B_3 y)$. Based on Remark D.3.2 we observe that the denominator of (D.29) is equal to $\varepsilon v^H M'(\lambda) x$. An upper bound is therefore

$$\frac{\Delta\lambda}{\varepsilon} \leq \|v\|\|x\|\frac{\alpha_1 + |\lambda|\alpha_2 + |g(\lambda)|\alpha_3}{|v^H M'(\lambda) x|} + \kappa_{g,B}(\lambda)\frac{|v^H A_3 x|}{|v^H M'(\lambda) x|} + \mathcal{O}(\varepsilon).$$

It remains to show that the bound can be attained. Similar to the proof of Lemma D.4.1, this follows from considering $\hat{B} = wy^H/\|w\|\|y\|$ and $\hat{A} = vx^H/\|v\|\|x\|$, and inserting

$$\Delta B_1 = \varepsilon\beta_1\hat{B} \qquad \Delta B_2 = \varepsilon\frac{\overline{\lambda}}{|\lambda|}\beta_2\hat{B} \qquad \Delta B_3 = \varepsilon\frac{\overline{g(\lambda)}}{|g(\lambda)|}\beta_3\hat{B}$$

$$\Delta A_1 = -\varepsilon\alpha_1\hat{A} \qquad \Delta A_2 = -\varepsilon\frac{\overline{\lambda}}{|\lambda|}\alpha_2\hat{A} \qquad \Delta A_3 = -\varepsilon\frac{\overline{g(\lambda)}}{|g(\lambda)|}\alpha_3\hat{A},$$

into (D.29). □

[2] This corresponds to $(\lambda, \mu)$ being a simple eigenvalue to the two-parameter eigenvalue problem.

### D.4.2 Backward stable computation of $g$

The nonlinearization is based on solving a GEP to evaluate the function $g(\lambda)$. We analyze the effects on the accuracy in the computed $\lambda$ when the GEP is solved numerically with a backward stable method. The analysis assumes the two triplets $(\lambda, x, v) \in \mathbb{C} \times \mathbb{C}^n \times \mathbb{C}^n$ and $(\mu, y, w) \in \mathbb{C} \times \mathbb{C}^m \times \mathbb{C}^m$ are such that $\lambda$ is a simple eigenvalue of the NEP (D.3), the pencil associated with the GEP (D.9) is regular, $\mu$ is a simple eigenvalue of the GEP, and $v, w$ and $x, y$ are corresponding left and right eigenvectors, respectively.

From the assumption that the GEP (D.9) is solved by a backward stable method we know that $\mu$ can be characterized as the exact solution to a nearby problem. More precisely, $\mu$ solves

$$(C_1 + \Delta C_1)y = \mu(C_2 + \Delta C_2)y,$$

where $C_1 = -(B_1 + \lambda B_2)$, $C_2 = B_3$, with perturbations, $\Delta C_1$ and $\Delta C_2$, that are proportional to the errors in our GEP solver. Specifically, there are non-negative $\beta_1, \beta_3 \in \mathbb{R}$ such that $\|\Delta C_1\| = \beta_1\varepsilon$ and $\|\Delta C_2\| = \beta_3\varepsilon$. Thus, the perturbation in $g$ is precisely captured by $\kappa_{g,B}(\lambda)$ from Lemma D.4.1, with $\beta_2 = 0$ and $\beta_1$ and $\beta_3$ given above, i.e., by the specific choice of GEP solver. Hence, by application of Theorem D.4.2 with $\alpha_j = 0$ for $j = 1, 2, 3$ we can conclude that the forward error in $\lambda$, induced by the inexact but backward stable computation of $g(\lambda)$ is bounded by

$$|\Delta\lambda| \leq \|w\|\|y\| \frac{\beta_1 + |g(\lambda)|\beta_3}{|w^H B_3 y|} \frac{|v^H A_3 x|}{|v^H M'(\lambda) x|}\varepsilon + \mathcal{O}(\varepsilon^2). \tag{D.30}$$

Without loss of generality we now assume that $\|x\| = \|v\| = \|y\| = \|w\| = 1$.

The upper bound (D.30) is related to the condition number for multiparameter eigenvalue problems as follows. As mentioned in the introduction, the condition number for the two-parameter eigenvalue problem can be directly expressed with the inverse of $C_0$ defined in (D.8). First note that our assumptions imply that $C_0$ is invertible.

**Lemma D.4.3.** *Under the conditions of Theorem D.4.2 the matrix $C_0$ is nonsingular, where $C_0$ is defined in* (D.8)*.*

*Proof.* By using the expression for $M'(\lambda)$ from Remark D.3.2 we thus have

$$(w^H B_3 y)(v^H M'(\lambda) x) = (v^H A_2 x)(w^H B_3 y) - (v^H A_3 x)(w^H B_2 y) = \det(C_0). \tag{D.31}$$

Since the eigenvalues $\lambda$ and $\mu$ are simple we know that $w^H B_3 y \neq 0$ and $v^H M'(\lambda) x \neq 0$. Hence, $\det(C_0) \neq 0$. □

From (D.31) we can conclude that the bound (D.30) on $|\Delta\lambda|$ can be written as

$$|\Delta\lambda| \leq (\beta_1 + |g(\lambda)|\beta_3) \frac{|v^H A_3 x|}{|\det(C_0)|}\varepsilon + \mathcal{O}(\varepsilon^2). \tag{D.32}$$

Moreover, for a nonsingular $C_0$ it is shown in [18, Theorem 6] that the condition number of the two-parameter eigenvalue is

$$K = \|C_0^{-1}\|_\theta,$$

where the $\theta$-norm, i.e., $\|\cdot\|_\theta$, is an induced norm defined in [18, Equation (5)].[3] In our case we can explicitly bound the condition number by using bounds following directly from the definition of the $\theta$-norm:

$$\|C_0^{-1}\|_\theta = \frac{1}{|\det(C_0)|}\left\|\begin{bmatrix} w^H B_3 y & -v^H A_3 x \\ -w^H B_2 y & v^H A_2 x \end{bmatrix}\right\|_\theta \geq \frac{1}{|\det(C_0)|}\left\|\begin{bmatrix} 0 & -v^H A_3 x \\ 0 & 0 \end{bmatrix}\right\|_\theta = \frac{|v^H A_3 x||\theta_2|}{|\det(C_0)|}.$$

The parameter $\theta_2$ is the second component of the $\theta$-vector used in the definition of the $\theta$-norm. Hence, the bound in (D.32) can be further bounded by

$$|\Delta\lambda| \leq K\frac{\beta_1 + |g(\lambda)|\beta_3}{|\theta_2|}\varepsilon + \mathcal{O}(\varepsilon^2). \tag{D.33}$$

The typical choices of $\theta$ corresponding to the absolute, respectively, relative condition number of the two-parameter eigenvalue problem are $|\theta_2| = 1 + |\lambda| + |g(\lambda)|$ and $|\theta_2| = \|B_1\| + |\lambda|\|B_2\| + |g(\lambda)|\|B_3\|$. From the bounds in (D.33) we therefore conclude: The error generated by a backward stable method is benign for well conditioned two-parameter eigenvalue problems.

## D.5 Simulations

### D.5.1 Random example

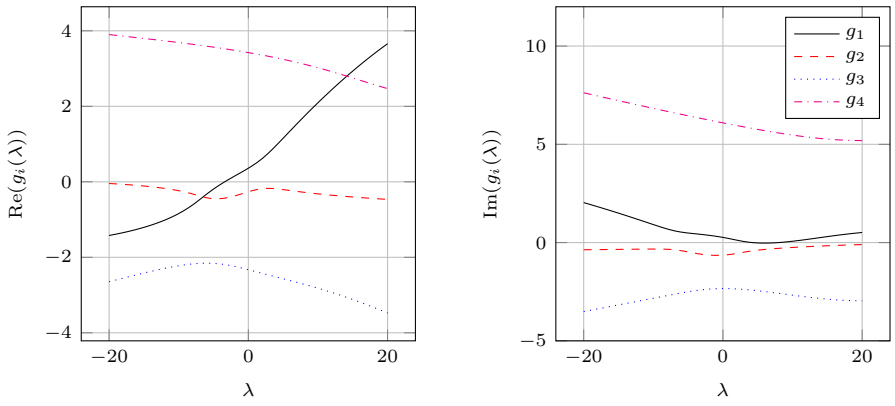
We generate an example similar to the example in [15], but with $m \ll n$. More precisely, we let

$$A_i = \alpha_i V_{A_i} F_i U_{A_i}, \quad B_i = \beta_i V_{B_i} G_i U_{B_i}, \quad i = 1, 2, 3$$

where $n = 5000$ and $m = 20$. The matrices $V_{A_i}$, $U_{A_i}$, $V_{B_i}$, $U_{B_i}$ have randomly normal distributed elements and $F_i$, $G_i$ are diagonal matrices with randomly normal distributed diagonal elements. The scalars $\alpha_i$ and $\beta_i$ were selected such that the eigenvalues closest to the origin were of order of magnitude one in modulus ($\alpha_1 = \beta_1 = 1$, $\alpha_2 = \beta_2 = 1/500$, $\alpha_3 = \beta_3 = 1/50$). The simulations were carried out using the Julia language [6] (version 1.1.0), but implementations of the algorithms are available online for both Julia and MATLAB.[4]

[3] For a given vector $\theta \in \mathbb{R}^n$ with positive entries, the $\theta$-norm of a matrix $C \in \mathbb{C}^{n\times n}$ is defined as $\|C\|_\theta := \max\{\|Cz\| : z \in \mathbb{C}^n, |z_k| = \theta_k \text{ for } k = 1, 2, \ldots, n\}$.

[4] The matrices and the simulations are provided online for reproducibility: `https://www.math.kth.se/~eliasj/src/nonlinearization`. The simulations were carried out using Ubuntu Linux, Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz, 16 GB of RAM.



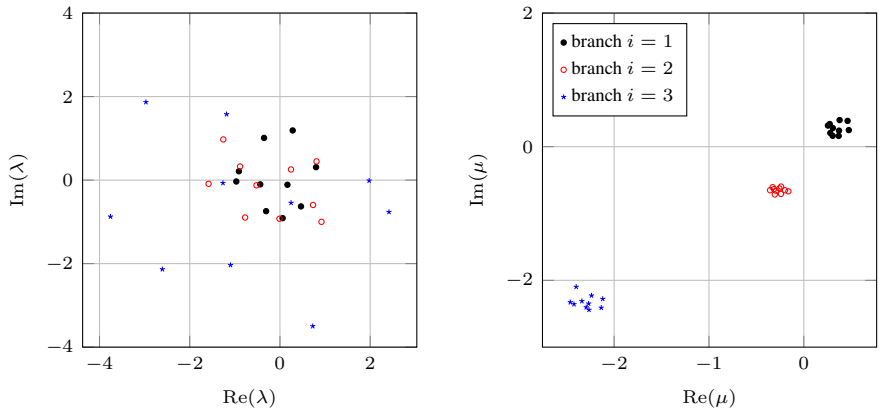

(a) Real (b) Imag

Figure D.2: The functions $g_i(\lambda)$, $i = 1, \ldots, 4$ closest to the origin, for $\lambda \in [-20, 20]$. All functions are analytic in the considered interval.

Since $m = 20$, we in general obtain 20 different functions $g_1, \ldots, g_{20}$, which we order by magnitude in the origin, each corresponding to a different NEP. Some of the nonlinear functions $g_i$ are visualized in Figure D.2. The solutions closest to the origin, for the NEPs corresponding to the functions $g_1, g_2, g_3$, are given in Figure D.3. The solutions are computed with the tensor infinite Arnoldi method. In theory, if the equivalence as described in Theorem D.2.3 holds in the desired point, the solution of the two-parameter eigenvalue problem closest to the origin could be found by computing solutions for all the 20 NEPs corresponding to $g_1, \ldots, g_{20}$.

We illustrate our algorithms and compare with several other single-vector state-of-the-art algorithms in [36]. As starting values we use $\lambda_0 = 0.15 + 0.1i$ and $\mu_0 = 35 + 0.25i$, and a starting vector with an elementwise absolute error (from a nearby solution) less than $0.05$. The iteration history of Algorithm D.1, in terms of residual norm (with eigenvectors normalized with respect to the 2-norm), is given in Figure D.4. We observe an asymptotic fast convergence for Algorithm D.1, which is expected since the solution point is analytic and simple. The error is measured at Step 3 in Algorithm D.1 which implies that by construction, the error in the $B$-equation is (numerically) zero. This is a property of the elimination in our approach. We compare (with the same starting values) with the inverse iteration Newton approach proposed in [36]. Note that this method is designed for more general problems, and not specifically our situation where $m \ll n$ and also multiparameter nonlinear problems.[5] In our implementation of [36, InvIter] we use two LU-factorizations to handle the multiple linear solves per iteration. For the above example that was faster in

[5]For multiparameter linear problems [36, InvIter] is equivalent to the tensor Rayleigh quotient iteration from [35, 16].




(a) Solutions $\lambda$

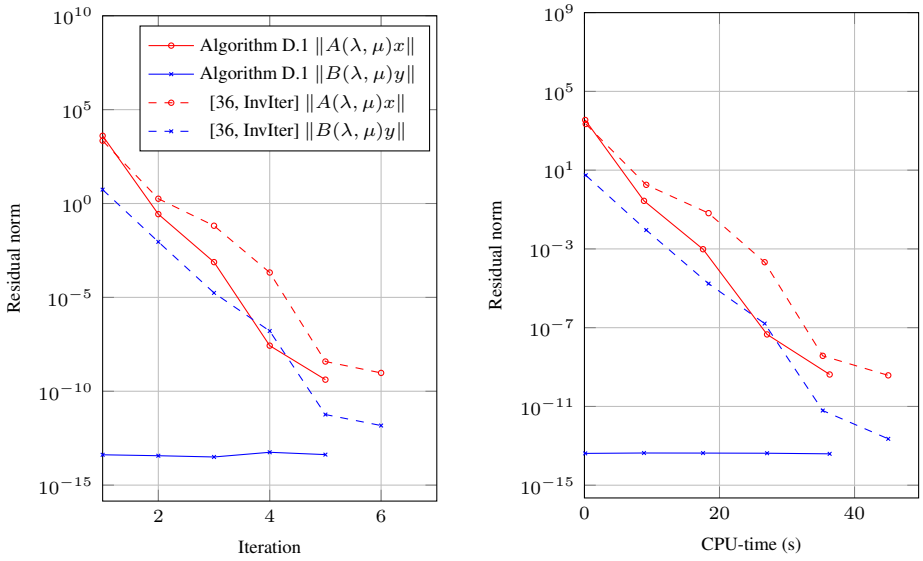
(b) Solutions $\mu$

Figure D.3: Solutions corresponding to $g_i$ where $i = 1, 2, 3$.

terms of execution time compared to direct solves, although that might not be the case for a larger and sparse problem. The comparison between the two algorithms as a function of iteration is inconclusive, as can be seen in Figure D.4a. However, in terms of CPU-time Algorithm D.1 is somewhat faster, as can seen in Figure D.4b.

The convergence of our adaption of residual inverse iteration (Algorithm D.2) initiated in the same way (except the starting vector is chosen as a vector of ones) is illustrated in Figure D.5. We clearly see the expected linear convergence, since it is equivalent to residual inverse iteration for NEPs and the convergence theory in [34, Section 3–4] is directly applicable. We compare with a proposed generalization of residual inverse iteration [36, InvIter], again noting that it has a much wider applicability domain than our approach. In this case, our method has a smaller convergence factor, intuitively motivated by the fact that we solve the $B$-equation exactly.

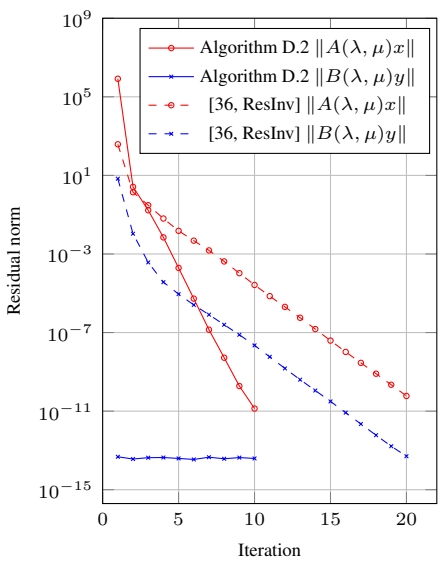
The problem can also be solved with the tensor infinite Arnoldi method [22]. More specifically, we use the implementation of the method available in the Julia package NEPPACK [20] (version 1.0.2). By directly using Theorem D.3.1 we can compute the 60 first derivatives. The convergence of the first ten eigenvalues are visualized in Figure D.6, for two branches. The solutions are visualized in Figure D.3.



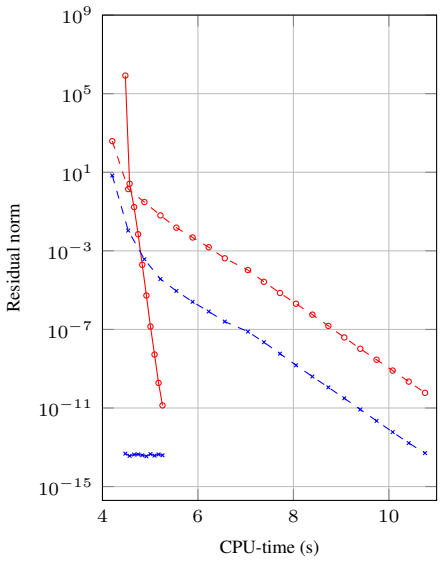

(a) Residual norm vs. iteration

(b) Residual norm vs. CPU-time

Figure D.4: Visualization of the convergence of Algorithm D.1 and [36, Algorithm 1 (InvIter)] for the problem in Section D.5.1.

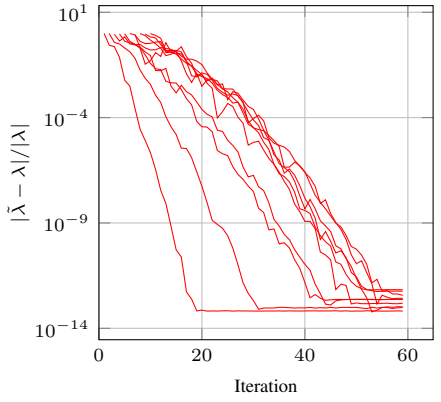



(a) Residual norm vs. iteration

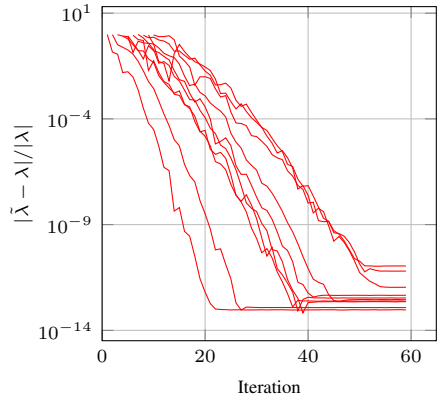



(b) Residual norm vs. CPU-time

Figure D.5: Visualization of the convergence of Algorithm D.2 and [36, Algorithm 2 (ResIter)] for the problem in Section D.5.1.




(a) Branch 1: $M(\lambda) = A_0 + \lambda A_1 + g_1(\lambda)A_2$




(b) Branch 2: $M(\lambda) = A_0 + \lambda A_1 + g_2(\lambda)A_2$

Figure D.6: Visualization of convergence of the tensor infinite Arnoldi method for the problem in Section D.5.1, for $g_1$ and $g_2$. The error is measured as relative error where the computed value $\lambda$ is compared to a precomputed reference solution $\hat{\lambda}$.

### D.5.2 Domain decomposition example

We consider a BVP-eigenvalue problem, which we separate into two domains in a way that it leads to a two-parameter eigenvalue problem. Similar techniques and analysis is found in, e.g., [9], [3, Chapter 2], and [19, Experiment 4], where it is common to force the solution to have roots within the considered interval.

Consider the Helmholtz eigenvalue problem defined in the domain $x \in [x_0, x_2]$,

$$\begin{aligned} u''(x) + \kappa^2(x)u(x) &= \lambda u(x) \text{ for } x \in [x_0, x_2] && \text{(D.34a)} \\ u(x_0) &= 0 && \text{(D.34b)} \\ u'(x_2) &= 0, && \text{(D.34c)} \end{aligned}$$

with a wavenumber $\kappa$ which is discontinuous in one part of the domain and smooth in another, as in Figure D.7. We take a point $x_1$ such that $\kappa$ is smooth for $x > x_1$, assume that the solution is nonzero in the interface point $x = x_1$, and define

$$\mu := \frac{u'(x_1)}{u(x_1)}.$$

This means we have two separate boundary value problems for the two domains:

$$\begin{aligned} u_1''(x) + \kappa^2(x)u_1(x) &= \lambda u_1(x), \quad x_0 \le x \le x_1 && \text{(D.35a)} \\ u_1(x_0) &= 0 && \text{(D.35b)} \\ u_1'(x_1) - \mu u_1(x_1) &= 0 && \text{(D.35c)} \end{aligned}$$

and

$$\begin{aligned} u_2''(x) + \kappa^2(x)u_2(x) &= \lambda u_2(x), \quad x_1 \le x \le x_2 && \text{(D.36a)} \\ u_2'(x_1) - \mu u_2(x_1) &= 0 && \text{(D.36b)} \\ u_2'(x_2) &= 0. && \text{(D.36c)} \end{aligned}$$

These are standard linear BVPs (with robin boundary conditions) and the uniqueness of these BVPs implies an equivalence with the original BVP (D.34). See [26] and references therein for domain decomposition methods for PDE eigenvalue problems.

The wavenumber is given as in Figure D.7, i.e., it is discontinuous at several points in $[x_0, x_1]$ and with a high frequency decaying sine-curve in $[x_1, x_2]$, representing a inhomogeneous periodic medium. We invoke different discretizations in the two domains, for the following reasons. Since $\kappa$ is discontinuous in $[x_0, x_1]$ spectral discretization in that domain will not be considerably faster than a finite difference approximation. We therefore use a uniform second order finite difference for (D.35) to obtain sparse matrices and one sided second order finite different scheme for the boundary condition. A spectral discretization is used for $[x_1, x_2]$ where the wavenumber is smooth. Since $\mu$ appears linearly in the boundary condition, the discretization leads to a two-parameter eigenvalue problem of the type (D.1). In our setting $A_1, A_2, A_3$ are large and sparse, and $B_1, B_2, B_3$ are full matrices of smaller size. We use the discretization parameters such that $n = 10^6$

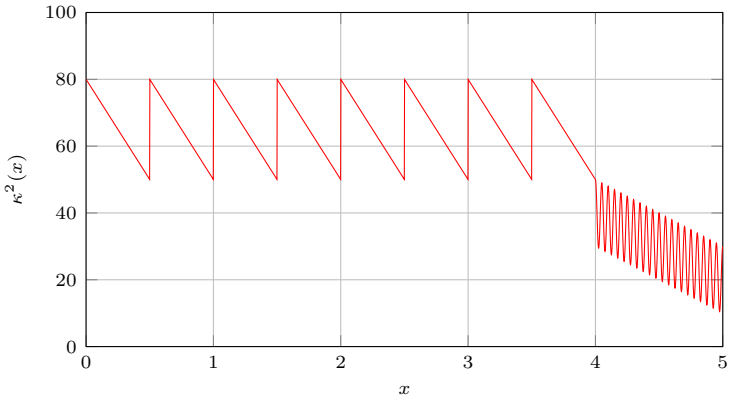



Figure D.7: The wavenumber for the example in Section D.5.2. The wavenumber is sinusoidal with high frequency in the interval $[4, 5]$, and discontinuous in $\frac{1}{2}, \frac{2}{2}, \frac{3}{2}, \ldots, \frac{7}{2}$.

and $m = 30$, and $x_0 = 0$, $x_1 = 4$ and $x_2 = 5$. In order to make the measurement of error easier, we use left diagonal scaling of the problem such that the diagonal elements of $A(1.0, 1.0)$ and $B(1.0, 1.0)$ are equal to one.

The eigenvalues and some corresponding eigenfunctions are plotted in Figure D.8 and Figure D.9. In this one-dimensional case, the structure of the problem implies that $B_3$ is a rank-one matrix. Hence, the GEP (D.9) only has one finite solution. The nonlinear function $g_1$ of this problem is given in Figure D.10. Clearly the function has singularities for some real $\lambda$-values. The convergence of Algorithm D.1 and Algorithm D.2 are again compared to [36] in Figure D.11. We again conclude that both our approaches are competitive, although not always faster in terms of iterations, but our approach is generally faster in terms of CPU-time. We note that the closed-form solution of $g_1$ is not exploited in these simulations. The algorithms are initiated with approximate rounded eigenvectors and eigenvalues close to the eigenvalue $\lambda \approx 18$. We note that our methods do not require a starting value for $\mu$ (in contrast to [36]) which is an attractive feature from an application point of view, since the value $\mu = u'(x_1)/u(x_1)$ is artificially introduced parameter and may not be easy to estimate.

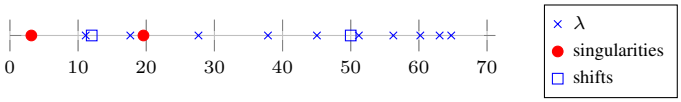



Figure D.8: Computed eigenvalues, singularities, and the shifts used in the infinite Arnoldi method.

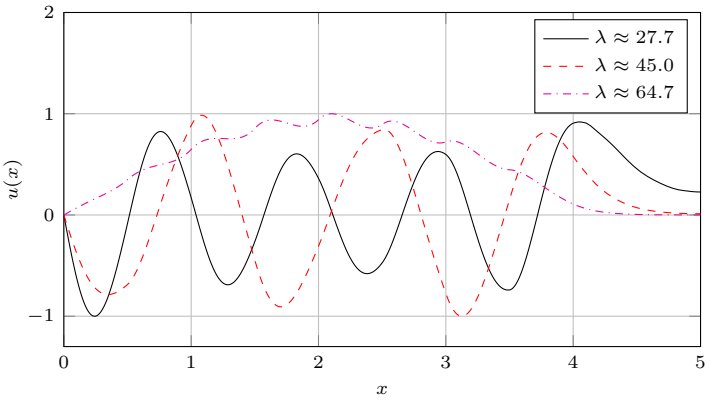



Figure D.9: Some computed eigenfunctions of (D.34)

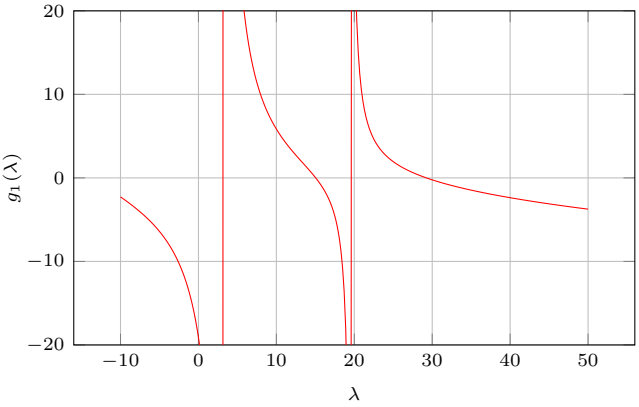
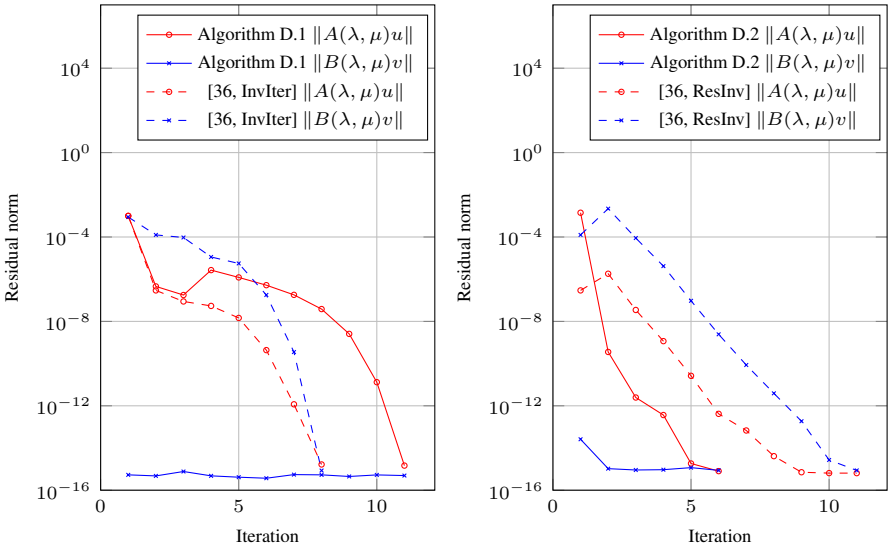


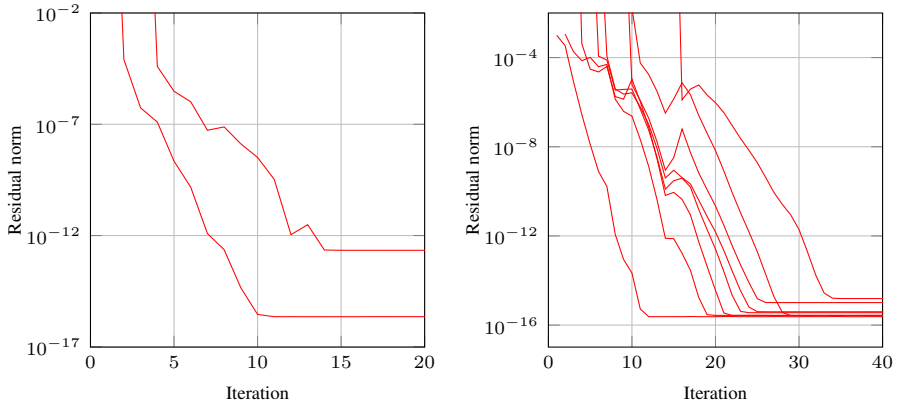

Figure D.10: The nonlinear function $g_1$ in the simulation in Section D.5.2.

We apply the tensor infinite Arnoldi method also for this problem. Since this family of methods is based on a power series expansion, one can only expect to be able to compute eigenvalues on the same side of the singularities as the shift. We therefore run the algorithm several times for different shifts, and select the shifts far away from the singularities, as described in Figure D.8. The convergence of the two runs are illustrated in Figure D.12. Note that the convergence corresponding to one eigenvalue for the shift $\sigma = 12$ stagnates. This is because the eigenvalue is close to the singularity, and therefore difficult to compute, as can be seen in Figure D.8.




(a) Newton-type algorithms (b) Fixed shift methods

Figure D.11: Visualization of the convergence of the proposed algorithms and two algorithms in [36] applied to the domain decomposition example in Section D.5.2.




(a) Shift $\sigma = 12.0$ (b) Shift $\sigma = 50.0$

Figure D.12: Convergence history for two different shifts

## D.6 Conclusions and outlook

We have presented a general framework to approach two-parameter eigenvalue problems, by nonlinearization to NEPs. Several steps in this technique seem to be generalizable (but beyond the scope of the paper), e.g., to general multiparameter eigenvalue problems essentially by successive application of the elimination. One such elimination leads to a nonlinear two-parameter eigenvalue problem as considered, e.g., in [36].

Our paper uses the assumption $m \ll n$ and that $A_1$, $A_2$ and $A_3$ are large and sparse. We made this assumption mostly for convenience, since this allows us to apply a general purpose method for the parameterized eigenvalue problem (D.9). If, on the other hand, we wish to solve two-parameter eigenvalue problems where these assumptions are not satisfied, the ideas may still be useful. The GEP (D.9) may for instance be approached with structured algorithms (exploiting sparsity, low-rank properties and symmetry), or iterative methods for the GEP, where early termination is coupled with the NEP-solver.

The generated nonlinear functions $g_i$ are algebraic functions, and can therefore contain singularities (e.g., branch point singularities as characterized in Section D.2). These can be problematic in the numerical method, and therefore it would be useful with transformations that remove singularities. Linearization which do not lead to singularities have been established for rational eigenvalue problems [42].

The problem in Section D.5.2 is such that we obtain one large and sparse parameterized matrix $A(\lambda, \mu)$ which is coupled with a small and dense system. The setting matches the assumptions of the paper and is a representative example of cases where the behavior is different in the two physical domains. The example may be generalizable, to other coupled physical systems where the modeling in one domain leads to a much smaller matrix, e.g., using domain decomposition with more physical dimensions. Note however that the presented methods seem mostly computationally attractive if the discretization of one domain is much smaller. If we apply the same technique to domains of equal size, other generic two-parameter eigenvalue methods (such as those in [36]) may be more effective.

## Acknowledgment

We thank Olof Runborg, KTH Royal Institute of Technology, for discussions regarding the Helmholtz eigenvalue problem.

## References

[1] R. Alam and S. Safique Ahmad. Sensitivity analysis of nonlinear eigenproblems. *SIAM J. Matrix Anal. Appl.*, 40(2):672–695, 2019.

[2] F. Atkinson. *Multiparameter Eigenvalue Problems. Volume I: Matrices and compact operators*. Academic press, New York, NY, 1972.

[3] F. V. Atkinson and A. B. Mingarelli. *Multiparameter Eigenvalue Problems: Sturm-Liouville theory*. CRC Press, Boca Raton, FL, 2011.

[4] 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):1–28, 2013.

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

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

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

[8] K. Fritzsche and H. Grauert. *From holomorphic functions to complex manifolds*, volume 213 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, NY, 2002.

[9] M. Greguš, F. Neuman, and F. M. Arscott. Three-point boundary value problems in differential equations. *J. Lond. Math. Soc. (2)*, 2(3):429–436, 1971.

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

[11] T. Helgaker, P. Jørgensen, and J. Olsen. *Molecular Electronic-Structure Theory*. John Wiley & Sons, Chichester, UK, 2000.

[12] V. Hernandez, J. E. Roman, and V. Vidal. SLEPc: Scalable Library for Eigenvalue Problem Computations. *Lect. Notes Comput. Sci.*, 2565:377–391, 2003.

[13] 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.

[14] D. J. Higham and N. J. Higham. Structured backward error and condition of generalized eigenvalue problems. *SIAM J. Matrix Anal. Appl.*, 20(2):493–512, 1998.

[15] M. E. Hochstenbach, T. Košir, and B. Plestenjak. A Jacobi–Davidson type method for the two-parameter eigenvalue problem. *SIAM J. Matrix Anal. Appl.*, 26(2):477–497, 2005.

[16] M. E. Hochstenbach, K. Meerbergen, E. Mengi, and B. Plestenjak. Subspace methods for three-parameter eigenvalue problems. *Numer. Linear Algebra Appl.*, page e2240, 2019.

[17] M. E. Hochstenbach and B. Plestenjak. A Jacobi–Davidson type method for a right definite two-parameter eigenvalue problem. *SIAM J. Matrix Anal. Appl.*, 24(2):392–410, 2002.

[18] M. E. Hochstenbach and B. Plestenjak. Backward error, condition numbers, and pseudospectra for the multiparameter eigenvalue problem. *Linear Algebra Appl.*, 375:63–81, 2003.

[19] M. E. Hochstenbach and B. Plestenjak. Computing several eigenvalues of nonlinear eigenvalue problems by selection. *Calcolo*, 57(2):1–25, 2020.

[20] E. Jarlebring, M. Bennedich, G. Mele, E. Ringh, and P. Upadhyaya. NEP-PACK: A Julia package for nonlinear eigenproblems, 2018. https://github.com/nep-pack.

[21] E. Jarlebring, S. Kvaal, and W. Michiels. Computing all pairs $(\lambda, \mu)$ such that $\lambda$ is a double eigenvalue of $A + \mu B$. *SIAM J. Matrix Anal. Appl.*, 32(3):902–927, 2011.

[22] E. Jarlebring, G. Mele, and O. Runborg. The waveguide eigenvalue problem and the tensor infinite Arnoldi method. *SIAM J. Sci. Comput.*, 39(3):A1062–A1088, 2017.

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

[24] T. Košir. Finite-dimensional multiparameter spectral theory: The nonderogatory case. *Linear Algebra Appl.*, 212:45–70, 1994.

[25] S. Kvaal, E. Jarlebring, and W. Michiels. Computing singularities of perturbation series. *Phys. Rev. A*, 83(3):032505, 2011.

[26] S. H. Lui. Domain decomposition methods for eigenvalue problems. *J. Comput. Appl. Math.*, 117(1):17–34, 2000.

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

[28] S. Mackey, N. Mackey, C. Mehl, and V. Mehrmann. Vector spaces of linearizations for matrix polynomials. *SIAM J. Matrix Anal. Appl.*, 28:971–1004, 2006.

[29] K. Meerbergen and B. Plestenjak. A Sylvester–Arnoldi type method for the generalized eigenvalue problem with two-by-two operator determinants. *Numer. Linear Algebra Appl.*, 22(6):1131–1146, 2015.

[30] V. Mehrmann and H. Voss. Nonlinear eigenvalue problems: A challenge for modern eigenvalue methods. *GAMM-Mitt.*, 27:121–152, 2004.

[31] G. Mele. The infinite Lanczos method for symmetric nonlinear eigenvalue problems. Technical report, KTH Royal Institute of Technology, 2018. arXiv:1812.07557.

[32] G. Mele and E. Jarlebring. On restarting the tensor infinite Arnoldi method. *BIT*, 58(1):133–162, 2018.

[33] Y. Nakatsukasa, V. Noferini, and A. Townsend. Vector spaces of linearizations for matrix polynomials: A bivariate polynomial approach. *SIAM J. Matrix Anal. Appl.*, 38(1):1–29, 2017.

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

[35] B. Plestenjak. A continuation method for a right definite two-parameter eigenvalue problem. *SIAM J. Matrix Anal. Appl.*, 21(4):1163–1184, 2000.

[36] B. Plestenjak. Numerical methods for nonlinear two-parameter eigenvalue problems. *BIT*, 56(1):241–262, 2016.

[37] J. E. Roman, C. Campos, E. Romero, and A. Tomas. SLEPc users manual. Technical Report DSIC-II/24/02 - Revision 3.10, D. Sistemes Informàtics i Computació, Universitat Politècnica de València, 2018.

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

[39] M. Shao. Conquering algebraic nonlinearity in nonlinear eigenvalue problems. Presentation at SIAM ALA, Hong Kong. Joint work with Z. Bai, W. Gao, and X. Huang, 2018.

[40] T. Slivnik and G. Tomšič. A numerical method for the solution of two-parameter eigenvalue problems. *J. Comput. Appl. Math.*, 15(1):109–115, 1986.

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

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

[43] D. Szyld and F. Xue. Preconditioned eigensolvers for large-scale nonlinear Hermitian eigenproblems with variational characterizations. I. Extreme eigenvalues. *Math. Comp.*, 85:2887–2918, 2016.

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

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

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

[47] H. Unger. Nichtlineare behandlung von eigenwertaufgaben. *Z. Angew. Math. Mech.*, 30:281–282, 1950. English translation: `https://www.math.tu-dresden.de/~schwetli/Unger.html`.

[48] R. Van Beeumen, O. Marques, E. G. Ng, C. Yang, Z. Bai, L. Ge, O. Kononenko, Z. Li, C.-K. Ng, and L. Xiao. Computing resonant modes of accelerator cavities by solving nonlinear eigenvalue problems via rational approximation. *J. Comput. Phys.*, 374:1031–1043, 2018.

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

[50] H. Voss. Nonlinear eigenvalue problems. In L. Hogben, editor, *Handbook of Linear Algebra*, number 164 in Discrete Mathematics and Its Applications. CRC Press, Boca Raton, FL, 2nd edition, 2014.

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

*Från oss går det levande ut.*
*I er går det livlösa in.*

*– Karin Boye* [vii]