

Low-rank methods for parameter-dependent eigenvalue problems and matrix equations

THÈSE N° 7137 (2016)

PRÉSENTÉE LE 2 SEPTEMBRE 2016

À LA FACULTÉ DES SCIENCES DE BASE

ALGORITHMES NUMÉRIQUES ET CALCUL HAUTE PERFORMANCE - CHAIRE CADMOS
PROGRAMME DOCTORAL EN MATHÉMATIQUES

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Petar SIRKOVIĆ

acceptée sur proposition du jury:

Prof. F. Nobile, président du jury
Prof. D. Kressner, directeur de thèse
Prof. Z. Drmac, rapporteur
Prof. E. Mengi, rapporteur
Prof. J. Hesthaven, rapporteur



ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2016

Acknowledgements

This thesis is based on my research conducted from September 2012 to May 2016 at EPFL and during a research stay at Virginia Tech University in March 2014.

First of all, I would like to thank my thesis advisor, Prof. Daniel Kressner, for giving me this opportunity, guiding me through my PhD studies with a lot of helpful and constructive comments and suggestions, as well as teaching me how to write scientific texts. I have learned a lot from him, both scientifically as well as about the academic community in general.

Furthermore, I would like to thank the members of my PhD committee, Prof. Zlatko Drmac, Prof. Jan Hesthaven, Prof. Fabio Nobile and in particular Prof. Emre Mengi for his detailed and constructive comments on the thesis.

Likewise, I would like to express my gratitude to Prof. Mark Embree, and Prof. Serkan Gugercin for their hospitality and help during my stay at Virginia Tech, as well as for the many encouraging and insightful scientific discussions.

Financial support by the Swiss National Science Foundation under the SNSF research module *A Reduced Basis Approach to Large-Scale Pseudospectra Computations* within the ProDoc *Efficient Numerical Methods for Partial Differential Equations* is thankfully acknowledged.

I would also like to thank all of my colleagues, fellow PhD students at EPFL, and in particular my officemates Agnieszka, Ana, Cedric, Christine, Michael, for all the discussions and laughs, both work and non-work related. I am especially grateful to Annick, Christine, and Jonas for being real friends who were there for me in times of need, and Michael for being an older "scientific" brother and paving the way for me. Moreover, I want to thank all of my friends in Lausanne, in particular Alberto, Amos, Maja, Marie, Marko, Momchil, and Viljami, as well as all the friends in Zagreb, for all the support and fun we had together during the past years, which has recharged my emotional batteries on numerous occasions.

Finally, I want to thank my family for their unconditional support and love throughout the years, Katarina for showing me that true love is just as mathematics, based on understanding which makes things simple, rather than complicated, and in particular Ana for keeping up with me on a daily basis and being there for me in some of the hardest moments of my life.

Lausanne, July 2016

P. S.

Abstract

The focus of this thesis is on developing efficient algorithms for two important problems arising in model reduction, estimation of the smallest eigenvalue for a parameter-dependent Hermitian matrix and solving large-scale linear matrix equations, by extracting and exploiting underlying low-rank properties.

Availability of reliable and efficient algorithms for estimating the smallest eigenvalue of a parameter-dependent Hermitian matrix $A(\mu)$ for many parameter values μ is important in a variety of applications. Most notably, it plays a crucial role in *a posteriori* estimation of reduced basis methods for parametrized partial differential equations. We propose a novel subspace approach, which builds upon the current state-of-the-art approach, the Successive Constraint Method (SCM), and improves it by additionally incorporating the sampled smallest eigenvectors and implicitly exploiting their smoothness properties. Like SCM, our approach also provides rigorous lower and upper bounds for the smallest eigenvalues on the parameter domain D . We present theoretical and experimental evidence to demonstrate that our approach represents a significant improvement over SCM in the sense that the bounds are often much tighter, at a negligible additional cost. We have successfully applied the approach to computation of the coercivity and the inf-sup constants, as well as computation of ε -pseudospectra.

Solving an $m \times n$ linear matrix equation $A_1 X B_1^T + \dots + A_K X B_K^T = C$ as an $mn \times mn$ linear system, typically limits the feasible values of m, n to a few hundreds at most. We propose a new approach, which exploits the fact that the solution X can often be well approximated by a low-rank matrix, and computes it by combining greedy low-rank techniques with Galerkin projection as well as preconditioned gradients. This can be implemented in a way where only linear systems of size $m \times m$ and $n \times n$ need to be solved. Moreover, these linear systems inherit the sparsity of the coefficient matrices, which allows to address linear matrix equations as large as $m = n = O(10^5)$. Numerical experiments demonstrate that the proposed methods perform well for generalized Lyapunov equations, as well as for the standard Lyapunov equations.

Finally, we combine the ideas used for addressing matrix equations and parameter-dependent eigenvalue problems, and propose a low-rank reduced basis approach for solving parameter-dependent Lyapunov equations.

Keywords: parameter-dependent problems, Hermitian eigenvalue problem, linear matrix equations, low-rank structure, subspace acceleration, greedy low-rank, pseudospectra computation, reduced basis method, Lyapunov equation

Zusammenfassung

Der Fokus dieser Arbeit liegt auf der Entwicklung effizienter Algorithmen für zwei wichtige Probleme im Bereich der Modellreduktion: der Schätzung des kleinsten Eigenwertes für eine parameterabhängige hermitesche Matrix sowie der Lösung von grossskaligen linearen Matrixgleichungen durch die Ausnutzung zugrunde liegender Niedrigrangeigenschaften.

Die Verfügbarkeit von verlässlichen und effizienten Algorithmen zur Schätzung des kleinsten Eigenwertes einer parameterabhängigen hermiteschen Matrix $A(\mu)$ für viele Parameterwerte μ aus einem Gebiet $D \subset \mathbb{R}^P$ ist für eine Vielzahl von Anwendungen von Bedeutung. Insbesondere spielen diese eine wichtige Rolle in *a posteriori* Abschätzungen von Reduzierte-Basis-Methoden für parametrisierte partielle Differentialgleichungen. Wir schlagen hier einen neuen Zugang über Unterräume vor, der auf der Successive Constraint Method (SCM) aufbaut und diese durch die zusätzliche Berücksichtigung von kleinsten Eigenvektoren aus einer Stichprobe verbessert, wobei gleichzeitig deren Glattheitseigenschaften ausgenutzt werden. Wie bei SCM ergeben sich aus unserem Zugang untere und obere Schranken für den kleinsten Eigenwert auf dem Parametergebiet D . Mittels einer theoretischen Analyse und durch numerische Experimente zeigen wir, dass unser Zugang eine signifikante Verbesserung zur SCM darstellt, der unter geringfügig höheren Kosten oft weitaus engere Schranken ermöglicht. Wir haben unseren neuen Zugang sowohl erfolgreich auf die Berechnung der Koerzitivitäts- und der inf-sup-Konstante wie auch auf die Berechnung von ϵ -Pseudospektren angewendet.

Die Lösung einer $m \times n$ linearen Matrixgleichung $A_1 X B_1^T + \dots + A_K X B_K^T = C$ als lineares System der Grösse $mn \times mn$ beschränkt die möglichen Werte für m, n oft auf maximal einige Hunderte. Wir schlagen hier einen neuen Zugang vor, der ausnutzt, dass die Lösung X oft sehr gut durch eine Niedrigrangmatrix approximiert werden kann. Wir berechnen diese Approximation durch eine Kombination von greedy Niedrigrangtechniken, einer Galerkin-Projektion und vorkonditionierten Gradienten. Unser Zugang ist so implementiert, dass nur lineare Systeme der Grösse $m \times m$ und $n \times n$ gelöst werden müssen. Ausserdem erben diese linearen Systeme die Dünnbesetztheit der Koeffizientenmatrizen, so dass auch Matrixgleichungen der Grösse $m = n = O(10^5)$ gelöst werden können. Unsere numerischen Experimente zeigen, dass sich die vorgeschlagene Methode sehr gut für verallgemeinerte Lyapunov-Gleichungen wie auch für den Standardfall nutzen lässt.

Zuletzt kombinieren wir unsere Ideen zu Matrixgleichungen und parameterabhängigen Eigenwertproblemen und schlagen eine Niedrigrang-Reduzierte-Basis-Methode zur Lösung parameterabhängiger Lyapunov-Gleichungen vor.

Acknowledgements

Stichwörter: parameterabhängige Probleme, hermitesches Eigenwertproblem, lineare Matrixgleichungen, Niedrigrangstruktur, Unterraum-Beschleunigung, greedy Niedrigrang, Berechnung von Pseudospektren, Reduzierte-Basis-Methode, Lyapunov-Gleichung

Contents

Acknowledgements	i
Abstract (English/Deutsch)	iii
1 Introduction	1
1.1 Parameter dependent eigenvalue problems	1
1.1.1 Applications	2
1.2 Linear matrix equations	3
1.2.1 Applications	4
1.2.2 Parameter-dependent matrix equations	5
1.3 Contributions of this thesis	5
2 Preliminaries	9
2.1 Eigenvalue problems	9
2.1.1 Numerical range	10
2.1.2 Hermitian eigenvalue problem	10
2.1.3 Computing extremal eigenvalues of a Hermitian matrix using the Lanczos method	11
2.1.4 Analyticity of eigenvalue decomposition	16
2.1.5 Perturbation analysis	19
2.2 Lyapunov equations	20
2.2.1 Low-rank solutions of Lyapunov equations	21
2.2.2 Solving large-scale Lyapunov equations	21
2.2.3 Lyapunov equation for Gramians of linear control systems	23
2.3 Reduced basis method	24
2.3.1 Model problem	25
2.3.2 Finite element discretization	26
2.3.3 Offline phase	28
2.3.4 Online phase	28
2.3.5 Error estimation	28
3 Low-rank approach for parameter dependent Hermitian eigenvalue problem	31
3.1 Successive constraint method	32
3.1.1 Linear optimization problem for $\lambda_{\min}(A(\mu))$	33
3.1.2 Bounding box	33

Contents

3.1.3	SCM bounds for $\lambda_{\min}(A(\mu))$	35
3.1.4	Error estimates and sampling strategy	36
3.1.5	Computational complexity	36
3.1.6	Interpolation results	37
3.2	Subspace acceleration	41
3.2.1	Subspace approach for upper bounds	41
3.2.2	Subspace approach for lower bounds	42
3.2.3	Error estimates and sampling strategy	46
3.2.4	Interpolation properties	46
3.2.5	A priori convergence estimates	49
3.2.6	Geometric interpretation	54
3.3	Heuristic variants	55
3.4	Algorithm	57
3.4.1	Computational details	57
3.4.2	Parameter value selection	59
3.4.3	Computational complexity	60
3.5	Applications and numerical examples	62
3.5.1	Random matrices	62
3.5.2	Estimation of the coercivity constant	63
3.5.3	Estimation of the inf-sup constant	66
3.6	Conclusion	69
4	Low-rank approach to pseudospectra computation	71
4.1	Existing approaches	73
4.1.1	Projection-based approaches	74
4.2	Subspace acceleration	76
4.2.1	Computation of $\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell)$ and $\lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell)$	77
4.2.2	Error estimates and sampling	80
4.2.3	Interpolation properties	80
4.3	Algorithm	82
4.3.1	Implementation details	82
4.3.2	Parameter value selection	87
4.3.3	Algorithm and computational complexity.	88
4.4	Numerical experiments	88
4.4.1	Comparison with other approaches	89
4.4.2	Dense matrices	90
4.4.3	Sparse matrices	91
4.5	Conclusion	95
5	Greedy low-rank approach to linear matrix equations	97
5.1	Greedy rank-1 approach	99
5.1.1	Symmetric positive definite case	99
5.1.2	Symmetric indefinite and nonsymmetric cases	102
5.1.3	Numerical example	103

5.1.4 Symmetry in the solution	103
5.2 Galerkin projection	107
5.2.1 Numerical example	109
5.3 Preconditioning	110
5.3.1 Preconditioners	111
5.3.2 Numerical example	112
5.4 Numerical experiments	112
5.4.1 Generalized Lyapunov equations	113
5.4.2 Lyapunov equation with right-hand sides having a singular value decay	117
5.4.3 Detailed numerical study of components of Algorithm 9	119
5.5 Conclusion	121
6 Low-rank approach to parameter dependent symmetric Lyapunov equations	123
6.1 Parametric model reduction	124
6.2 Reduced basis method for Lyapunov equations	126
6.2.1 Low-rank structure in the offline phase	127
6.2.2 Low-rank structure in the online phase	127
6.2.3 Low-rank structure in the error estimator	128
6.3 Algorithm	129
6.3.1 Computational details	131
6.3.2 Computational complexity	132
6.4 Numerical examples	133
6.5 Conclusion	134
7 Conclusion	137
Bibliography	139
Curriculum Vitae	151

1 Introduction

This thesis is concerned with extracting and exploiting the low-rank structure in two problems arising in model reduction. More specifically, we aim at developing efficient algorithms for estimating the smallest eigenvalues of a parameter-dependent matrix as well as solving large-scale matrix equations.

In the first part of the thesis, we focus on parameter-dependent eigenvalue problems, present several applications, and discuss various approaches to address them. In Chapter 3, we present a novel subspace-accelerated approach for parameter-dependent Hermitian eigenvalue problem and show how it can be used for estimating coercivity and inf-sup constants. This approach is further optimized for computation of ε -pseudospectra in Chapter 4.

The second part of the thesis is concerned with solving linear matrix equations admitting a low-rank solution. In Chapter 5, we present a greedy low-rank approach for solving general linear matrix equations. Furthermore, in Chapter 6, we combine the low-rank methods for matrix equations with techniques for estimating the smallest eigenvalue from the first part of the thesis in order to address parameter-dependent symmetric Lyapunov equations.

In the remainder of this chapter we introduce and motivate each of the problems by presenting few common applications.

1.1 Parameter dependent eigenvalue problems

Suppose we are given a Hermitian matrix $A(\mu) \in \mathbb{C}^{n \times n}$ depending on a parameter $\mu \in D$, where D is a compact subset of \mathbb{R}^d , and we are interested in computing its smallest eigenvalue

$$\lambda_{\min}(A(\mu)), \quad \mu \in D, \tag{1.1}$$

for *many* different values of μ . In the large-scale setting, say when $n > 1000$, computation of the smallest eigenvalue $\lambda_{\min}(A(\mu))$ using a standard eigensolver, such as the Lanczos method, is computationally affordable only for a few values of μ but becomes infeasible for larger numbers (e.g., thousands) of parameter values.

Without any further assumptions on the dependence of $A(\mu)$ on μ , addressing (1.1) is computationally very difficult, especially when d is large. Assuming regularity in $A(\mu)$ does not necessarily help; even when $A(\mu)$ depends analytically on μ , the smallest eigenvalue is not necessarily analytic in μ . In fact, $\lambda_{\min}(A(\mu))$ does inherit analyticity as long as it remains simple, but, at the eigenvalue crossings, $\lambda_{\min}(A(\mu))$ is only Lipschitz continuous. For larger values of d , even $d > 1$, keeping track of eigenvalue crossings is usually not computationally feasible and, thus, methods for solving (1.1) can exploit the piecewise regularity only implicitly.

Computationally efficient approaches for (1.1) can be derived by additionally assuming that $A(\mu)$ admits an affine linear decomposition with respect to μ : there exist $Q \in \mathbb{N}$, $Q \ll n^2$, Hermitian matrices A_1, \dots, A_Q , and functions $\theta_1, \dots, \theta_Q : D \rightarrow \mathbb{R}$ such that

$$A(\mu) = \theta_1(\mu)A_1 + \dots + \theta_Q(\mu)A_Q, \quad \forall \mu \in D. \quad (1.2)$$

This assumption is commonly found in the literature when addressing parameter-dependent problems. The current state-of-the-art approach, the so-called Successive Constraint Method (SCM) [HRSP07], samples values of $\lambda_{\min}(A(\mu))$ for carefully chosen parameter values inside D , and uses (1.2) together with linear programming techniques to provide rigorous bounds for the smallest eigenvalues on whole D . In the first part of the thesis we present a new subspace approach for (1.1) which builds upon SCM by additionally incorporating the sampled smallest eigenvectors and implicitly exploiting their smoothness properties. Like SCM, our approach provides both upper and lower bounds for $\lambda_{\min}(A(\mu))$. We present theoretical and experimental evidence that the bounds produced by the subspace approach represent a significant improvement over SCM in the sense that the bounds are often much tighter, at a negligible additional computational cost.

1.1.1 Applications

Eigenvalue problems of the form (1.1) often arise in model order reduction techniques, such as the reduced basis method (RBM). Successful application of RBM to a parameter-dependent symmetric elliptic partial differential equation (PDE), depends on the availability of reliable *a posteriori* error estimates, which can be attained by estimating the coercivity constant of the underlying PDE, see, e.g. [RHP08]. It can be easily shown that, in a discretized setting, estimation of the coercivity constant is equivalent to (1.1). Moreover, for more general types of PDEs, instead of the coercivity constants, *a posteriori* error estimation requires estimates for the inf-sup constants [HKC⁺10]. After discretization, this comes down to estimating the smallest singular values of a parameter-dependent matrix $A(\mu)$, or equivalently, estimating the smallest eigenvalue of the Hermitian matrix $A(\mu)^* A(\mu)$, again an eigenvalue problem of the form (1.1).

Another important application of (1.1) is in the computation of ε -pseudospectra [TE05, Part IX], which requires computation of the smallest singular values of $zI - A$ on a portion of the complex plane. Since singular values of $zI - A$ are also eigenvalues of $(zI - A)^*(zI - A)$, this is equivalent to solving the following parameter-dependent Hermitian eigenvalue problem,

which also admits affine linear decomposition w.r.t. x and y :

$$\begin{aligned}\sigma_{\min}((x + iy)I - A)^2 &= \lambda_{\min}(((x + iy)I + A)^*((x + iy)I + A)) \\ &= \lambda_{\min}(A^*A - x(A + A^*) - yi(A^* - A) + (x^2 + y^2)I).\end{aligned}$$

Other applications of parameter-dependent eigenvalue problems include computation of other spectral and pseudospectral properties, e.g. the numerical range [Joh78, Uhl14], the method of particular solutions [BT05], and the eigenvalue analysis of waveguides [EHS09]. The related problem of optimizing the extremal eigenvalue(s) of a parameter-dependent Hermitian matrix appears in a large variety of applications: one-parameter optimization problems play a critical role in the design of numerical methods [RSS01] and robust control [LO96]; multi-parameter optimization problems arise from semidefinite programming [HR00] and graph partitioning [KM06, GBS08].

A problem class closely connected to (1.1) are the stochastic eigenvalues problems, where A depends on a random field. They can easily be turned into parameter-dependent eigenvalue problems using the (truncated) Karhunen-Loève expansions, see [AS12, HKL15] for examples. Several approaches have been proposed to address stochastic eigenvalues problems, including adaptive sparse grid collocation and stochastic inverse iteration, see [HKL15, MG14]. While these approaches carry over to the setting of (1.1), they do not provide reliable lower and upper bounds for $\lambda_{\min}(A(\mu))$ unlike SCM or our proposed approach.

1.2 Linear matrix equations

We are interested in solving large-scale linear matrix equations of the form

$$\sum_{q=1}^Q A_q X B_q^T = C, \tag{1.3}$$

for given coefficient matrices $A_1, \dots, A_Q \in \mathbb{R}^{m \times m}$, $B_1, \dots, B_Q \in \mathbb{R}^{n \times n}$, $C \in \mathbb{R}^{m \times n}$. Vectorization of the matrix equation (1.3) turns it into an equivalent linear system

$$\sum_{q=1}^Q (B_q \otimes A_q) \text{vec}(X) =: \mathcal{A} \text{vec}(X) = \text{vec}(C). \tag{1.4}$$

The most straightforward approach to solving (1.3) is by computing $X \in \mathbb{R}^{m \times n}$ as the solution of (1.4), which typically limits the feasible values of m, n to a few hundreds at most. We aim to find a method that would allow us to address (1.3) as large as m, n of the order of 10^5 , by exploiting additional structure, such as the fact that X can often be well approximated using a low-rank matrix or that the coefficient matrices are often very sparse.

For the case $Q = 2$, the matrix equation (1.3) has been well understood. It reduces to the

so called *generalized Sylvester equation* and can be solved in $\mathcal{O}(n^3)$ using the generalized Bartels-Stewart algorithm [BS72, GLAM92]. It includes the standard Sylvester equation $A_1 X + X B_2^T = C$ and the *Lyapunov equation* $A_1 X + X A_1^T = -C$, with C symmetric positive definite, as particularly important special cases. For larger values of n and m , a number of specialized approaches have been developed that rely on the assumption that X can be well approximated using a low-rank matrix, and attempt to compute the low-rank factors directly, resulting in significant computational and storage savings. It has been shown in [GHK03, Pen00, ASZ02] that such a low-rank approximation of X exist for Lyapunov equations when the right-hand side C is of low-rank.

None of these established methods for Lyapunov and Sylvester equations directly generalizes to the case $Q > 2$. The existing work for $Q > 2$ has mostly addressed some special cases of (1.3), with emphasis on the *generalized Lyapunov equation*

$$\mathcal{L}(X) + \mathcal{N}(X) = AX + XA^T + \sum_{q=1}^Q N_q X N_q^T = -DD^T, \quad (1.5)$$

with $\mathcal{L} : X \mapsto AX + XA^T$, which appears to be the most frequently encountered instance of (1.3) with $Q > 2$. It typically arises in connection with bilinear dynamical systems and stochastic control. By extending the results for the Lyapunov case, singular value decay bounds for the solution of (1.5) have been established in [BB13, Mer12] under various conditions on A and N_q . Existing approaches that exploit the low-rank approximability of the solution include a fixed point iteration [Dam08] based on splitting $\mathcal{L}(X) + \mathcal{N}(X) = -DD^T$, which converges when \mathcal{L} is the dominant part of (1.5), as well as an approach [BB13] that combines a standard iterative solver, such as CG or BiCGstab, with preconditioning and low-rank truncation of the iterates.

In the second part of this thesis, we develop low-rank methods for solving the general linear matrix equations (1.3). The core idea of our approach is motivated by a class of methods proposed in [AMCK06, Nou10] for solving Fokker-Planck equations and stochastic partial differential equations. More precisely, we subsequently refine the current approximation to the solution X by adding a rank-1 correction, which is chosen as a minimizer of a certain target functional, making the approach a greedy algorithm. Additionally, we propose two techniques for accelerating convergence: including information from the preconditioned residual, similar to the techniques considered in [DS14], and performing Galerkin projection.

1.2.1 Applications

The most prominent application of Sylvester, Lyapunov and generalized Lyapunov equations is in the control theory. In particular, it can be shown that the reachability and the controllability Gramians of linear

$$x'(t) = Ax(t) + Bu(t), \quad y(t) = Cx(t), \quad x(0) = x_0, \quad (1.6)$$

and bilinear control systems

$$x'(t) = Ax(t) + \sum_{q=1}^Q N_q u_q x(t) + Bx(t), \quad y(t) = Cx(t), \quad x(0) = x_0,$$

are the solutions of the corresponding Lyapunov and generalized Lyapunov equations (see e.g. [BD11]), respectively. Gramians play a crucial role in model order reduction for dynamical systems. They can be used to identify the states that are both hard to reach and hard to observe, i.e. the states that can be neglected without significantly influencing the system's transfer behavior. This idea is implemented in a popular model order reduction technique, also known as the balanced truncation [Moo81].

1.2.2 Parameter-dependent matrix equations

In applications, the dynamical systems considered above frequently depend on a number of parameters. For example, the dynamical system (1.6) dependent on d real parameters $(\mu^{(1)}, \dots, \mu^{(d)}) = \mu \in D \subset \mathbb{R}^d$ can be written in the following way

$$x'(t; \mu) = A(\mu)x(t; \mu) + B(\mu)u(t; \mu), \quad y(t; \mu) = C(\mu)x(t; \mu), \quad x(0; \mu) = x_0, \quad \forall \mu \in D.$$

Design, control and optimization of such dynamical systems often require repeated model evaluations for *many* different parameter values. As explained above, using the balanced truncation algorithm to construct reduced-order models in such a setting would require computing (approximate) solution of the following large-scale parameter-dependent Lyapunov equation

$$A(\mu)P(\mu) + P(\mu)A(\mu)^T = -B(\mu)B(\mu)^T \quad (1.7)$$

for each of these parameter values. In case of general parameter-dependence in $A(\mu)$ and $B(\mu)$, this is computationally too expensive, and usually not feasible. However, if $A(\mu)$ and $B(\mu)$ additionally admit affine linear decomposition w.r.t μ (1.2), the reduced basis method can be used to accelerate this procedure. After sampling solutions of (1.7) for few values of μ , the RBM provides accurate approximate solutions to (1.7) on whole D at almost negligible additional computational cost. We present a specialized version of the reduced basis method that exploits Kronecker product structure in (1.7) as well as low-rank approximability of $X(\mu)$.

1.3 Contributions of this thesis

Chapter 2. We review some basic definitions and concepts related to Hermitian eigenvalue problems, Lyapunov equations and the reduced basis method that will be used in the subsequent chapters.

Chapter 3. We derive a new subspace approach for computing extremal eigenvalues of parameter-dependent Hermitian eigenvalue problems. The content of this chapter is mostly based on ideas discussed in [SK16, Sir16].

While presenting an overview of the current state-of-the-art approach, the Successive Constraint Method (SCM), we discuss possible alternatives for the bounding box set used in SCM. Additionally, we partly explain a numerically observed phenomenon from the literature, that the SCM upper bounds converge faster than the SCM lower bounds, by proving that the SCM upper bounds interpolate the derivatives of the smallest eigenvalues and showing by counterexample that the same does not hold for the SCM lower bounds. Finally, we prove that the SCM lower bounds cannot be improved without taking into account additional information about $A(\mu)$.

We derive our subspace approach as an extension of SCM where we allow sampling of more than one smallest eigenpair per sampling point. We show that the proposed subspace upper bounds can be efficiently computed by solving a small dense eigenvalue problem. As one of the key results, we demonstrate that the proposed subspace lower bounds can be computed at a negligible additional cost using linear programming techniques, and a perturbation argument that combines the computed values of subspace upper bounds, SCM lower bounds and eigenvalue residuals.

We show that the subspace bounds are always at least as good as the SCM bounds. Moreover, we prove that not only the subspace upper bounds interpolate the derivatives of $\lambda_{\min}(A(\mu))$, but also the subspace lower bounds, which indicates that we can expect locally second order convergence of the proposed bounds. Furthermore, we show exponential convergence in a special case when $d = 1$, $A(\mu)$ is analytic in μ , and the smallest eigenvalue stays simple on D . For a specific case of linear parameter dependence in $A(\mu)$, we prove that the subspace lower bounds are always at least as good approximation to $\lambda_{\min}(A(\mu))$ as the linear interpolation of the sampled values.

By precomputing the projected matrices similarly as in the reduced basis method, we obtain an efficient implementation of our approach where the evaluation of the subspace bounds for fixed $\mu \in D$ has a computational complexity independent of n . The performance is further optimized by incorporating the "saturation assumption". We demonstrate on a number of numerical examples that our approach, as implemented, significantly outperforms SCM both in terms of iterations and the total computational time.

In addition to the content presented in [SK16, Sir16], we discuss in Section 3.4.2 the impact of the number of sampled eigenvectors per sample point and the size of the training set on the performance of our approach. Furthermore, in Section 3.5.3 we include an example from the literature showing that our subspace approach can for specific cases be successfully applied to computation of inf-sup constants.

Chapter 4. We build upon the proposed subspace approach, i.e. Algorithm 3, presented in Chapter 3 and develop a new projection-based approach for pseudospectra computation. The content of this chapter is mostly based on [Sir16].

We demonstrate that Algorithm 3 can be extended to pseudospectra computation. More precisely, we show that solving the parameter-dependent singular value problem $\sigma_{\min}(zI - A)$ is equivalent to solving Hermitian eigenvalue problem linearly depending on two real parameters $\lambda_{\min}(A^*A - x(A + A^*) - yi(A^* - A))$. However, in order to make Algorithm 3 computationally efficient in this setting, we take into account the particular problem structure as well as the demands for high absolute accuracy, and make a number of modifications: we avoid often numerically unstable computation of the bounding box, accelerate the computation of the SCM lower bounds by using the simplex method with updating, and make the residual computation more robust. Finally, we accelerate the approach using a "warm start" strategy by *a priori* insertion eigenvalues of A inside D into the sample set. Moreover, we show that the interpolation properties and the *a priori* convergence results from Chapter 3 naturally extend to the singular value case.

We test our implementation on a number of examples from the literature, and compare its performance with few other projection-based approaches. The results indicate that our approach is particularly suited to the computation of pseudospectra around isolated parts of the spectrum.

In addition to the content presented in [Sir16], in Section 4.1.1 we discuss the possibility of using two-sided projections for approximating the smallest singular values. We include a simple example which clearly indicates that such approaches are not stable.

Chapter 5. We develop a greedy low-rank approach for solving general linear matrix equations. The content of this chapter is mostly based on the ideas discussed in [KS15].

We derive a basic greedy rank-1 strategy for updating the current approximate solution by adding a rank-1 correction, which is chosen to minimize the error, either in the energy norm, if \mathcal{A} is symmetric positive definite, or in the norm induced by $\mathcal{A}^T \mathcal{A}$. A local minimizer of the target functional can be efficiently computed using the alternating linear scheme (ALS). We show that the approach can analogously be extended to work with rank- r corrections. For the special case of symmetric Lyapunov equations, we prove that this algorithm preserves symmetry and converges to the exact solution monotonically in the Löwner ordering of positive semidefinite matrices.

We further improve convergence of our greedy low-rank approach by adding information from the residual preconditioned by one step of sign function iteration for Lyapunov equations and by performing Galerkin projection on the subspaces spanned by all previous correction terms. As the computational cost of the Galerkin projection grows very rapidly with the rank of the subspaces, we limit this effect by performing low-rank truncation of the correction subspaces.

We test our approach on a number of large-scale examples available in the literature. The results indicate that our approach is competitive with other available approaches, especially when the imposed limit on the subspace size is not reached.

Chapter 6. We develop a low-rank approach for solving parameter-dependent symmetric Lyapunov equations. The content of this chapter is mostly based on [KSSS14].

We use the idea of the reduced basis method (RBM) to address (1.7), but instead of constructing a subspace out of vectorized solutions $\text{vec}(P(\mu_1)), \dots, \text{vec}(P(\mu_M)) \in \mathbb{R}^{n^2 \times 1}$, we consider their low-rank Cholesky factors $P(\mu_i) = L(\mu_i)L(\mu_i)^T$ and collect them into a subspace $U_M \subset \mathbb{R}^n$

$$\mathcal{U}_M = \text{range}([L(\mu_1), \dots, L(\mu_M)]).$$

The approximate solution is then computed using Galerkin projection of (1.7) onto $\mathcal{U}_M \otimes \mathcal{U}_M$, and requires solution of a small-scale Lyapunov equation. Compared to the straightforward use of RBM, not only is our approach more accurate, but it also guarantees that the approximate solution is positive semidefinite – a very important property in model order reduction applications. We define *a posteriori* error estimates similarly like in RBM. However, by estimating the error in the Frobenius norm, we can avoid directly estimating the smallest eigenvalue of $n^2 \times n^2$ matrix $\mathcal{A}(\mu) = I \otimes A(\mu) + A(\mu) \otimes I$. Instead, a reliable lower bound for $\lambda_{\min}(\mathcal{A}(\mu))$ can be constructed using properties of the Kronecker product, and efficiently computed by estimating the smallest eigenvalue of $n \times n$ matrix $A(\mu)$. To further optimize the performance of our approach, we use the saturation assumption, which significantly reduces the number of error estimate computations throughout the iterations.

2 Preliminaries

In this chapter we recall the notation and some basic results for the two main topics of this thesis: eigenvalue problems and matrix equations.

In Section 2.1, we first provide basic definitions for eigenvalue problems. We present an overview of the Lanczos method, a popular method for computing extremal eigenvalues of a symmetric or Hermitian matrix. Furthermore, we shortly discuss and present some important results on the analyticity of eigenvalues and eigenvalue perturbation theory.

In Section 2.2, we consider Lyapunov equations, one of the most important examples of a linear matrix equation. We discuss some important properties of the solution, such as definiteness and low-rank approximability of solutions, and present an approach for solving large-scale Lyapunov equations that exploits these properties. Furthermore, we discuss some important applications of Lyapunov equations.

In Section 2.3, we present an overview of the reduced basis method for symmetric coercive parameter-dependent partial differential equations. For the considered model problem, we describe an efficient implementation of the offline and the online phase and discuss the choice of norm and how it influences *a posteriori* error estimation.

2.1 Eigenvalue problems

Given a matrix $A \in \mathbb{C}^{n \times n}$, we say that $\lambda \in \mathbb{C}$ is an *eigenvalue* of A if there exists a nonzero vector $v \in \mathbb{C}^{n \times 1}$ such that

$$(A - \lambda I)v = 0.$$

Such a vector v is called an *eigenvector* of A associated to the eigenvalue λ and (λ, v) an *eigenpair* of A . Since $A - \lambda I$ is singular if and only if $\det(A - \lambda I) = 0$, we can equivalently define eigenvalues of A as the roots of the *characteristic polynomial*

$$\kappa_A(\lambda) := \det(A - \lambda I) = 0.$$

It follows directly that A has n , not necessarily distinct, eigenvalues $\lambda_1, \dots, \lambda_n \in \mathbb{C}$. The set of all eigenvalues of A is called the *spectrum* of A , and it is denoted with $\lambda(A)$. The eigenvalue λ_i is said to be *simple* if its corresponding multiplicity in the characteristic polynomial is one or, equivalently, if $\lambda_i \neq \lambda_j$, for $j \neq i$.

Eigenvalues are sensitive to small changes in the matrix A . However, if we know that $\tilde{\lambda}$ is an eigenvalue of $A + E$, where E is small in norm, the following theorem [Saa92] proves that $\tilde{\lambda}$ is "close" to $\lambda(A)$.

Theorem 2.1 (Bauer-Fike). *If $\tilde{\lambda}$ is an eigenvalue of $A + E \in \mathbb{C}^{n \times n}$ and there exists invertible $X \in \mathbb{R}^{n \times n}$ such that $XAX^{-1} = D = \text{diag}(\lambda_1, \dots, \lambda_n)$, then*

$$\min_{\lambda \in \lambda(A)} |\lambda - \tilde{\lambda}| \leq \kappa(X) \|E\|_2,$$

where $\kappa(X)$ is the matrix condition number defined as

$$\kappa(X) = \|X\|_2 \|X^{-1}\|_2.$$

2.1.1 Numerical range

A rough estimate for the spectrum $\lambda(A)$ can be obtained by computing what is called the *numerical range* and denoted with

$$W(A) := \text{Im}(R_A) \subset \mathbb{C},$$

where R_A is the *Rayleigh-Ritz quotient* associated to the matrix A

$$R_A(v) = \frac{v^* A v}{v^* v}. \tag{2.1}$$

The inclusion $\lambda(A) \subset W(A)$ follows immediately from the fact that $R_A(v) = \lambda$ for an eigenpair (λ, v) . Furthermore, it was shown by Hausdorff and Toeplitz [Hau19, Toe18] that $W(A)$ is a compact and convex subset of the complex plane.

2.1.2 Hermitian eigenvalue problem

A matrix $A \in \mathbb{C}^{n \times n}$ is called *Hermitian* if it equals its conjugate transpose

$$A = A^*.$$

Additional properties hold for eigenvalues and eigenvectors of Hermitian matrices, some of which we recall in the following:

- All eigenvalues are real

$$\lambda_i \in \mathbb{R}, \quad \forall i = 1, \dots, n.$$

- If $\lambda_i \neq \lambda_j$, then the corresponding eigenvectors v_i and v_j are orthogonal

$$v_j^* v_i = 0.$$

- A is diagonalizable, i.e. there exists a diagonal matrix $D = \text{diag}(\lambda_1, \dots, \lambda_n) \in \mathbb{R}^{n \times n}$ and a unitary matrix $V = [v_1, \dots, v_n] \in \mathbb{C}^{n \times n}$ ($V^* V = V V^* = I$) containing the associated eigenvectors as columns, such that

$$A = V D V^* = \sum_{i=1}^n \lambda_i v_i v_i^*. \quad (2.2)$$

Having only real eigenvalues, we can impose an ordering on the eigenvalues of a Hermitian matrix A . Without loss of generality, we have $\lambda_{\min}(A) = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_{n-1} \leq \lambda_n = \lambda_{\max}(A)$. Of particular interest in many applications, such as checking definiteness, computing condition number, etc., are the extremal eigenvalues λ_1 and λ_n . Since $A = A^*$, we have $R_A(v) = R_A(v)^*$, and bearing in mind that $W(A)$ is convex, this immediately implies that $W(A)$ is a real line segment. In fact, using the eigenvalue decomposition of A (2.2), we obtain the minimax characterisation of extremal eigenvalues

$$\lambda_{\min}(A) = \min_{v \in \mathbb{C}^n} R_A(v) \leq \frac{\sum_{i=1}^n \lambda_i (v^* v_i)^2}{v^* v} \leq \max_{v \in \mathbb{C}^n} R_A(v) = \lambda_{\max}(A), \quad (2.3)$$

which proves that $W(A) = [\lambda_1, \lambda_n] \subset \mathbb{R}$. By slightly modifying (2.3), we also can characterise the other eigenvalues of A in the following way:

$$\lambda_k = \min_{\substack{\mathcal{U} \subset \mathbb{R}^n \\ \dim(\mathcal{U})=k}} \max_{v \in \mathcal{U}} R_A(v) = \max_{\substack{\mathcal{U} \subset \mathbb{R}^n \\ \dim(\mathcal{U})=n-k+1}} \min_{v \in \mathcal{U}} R_A(v). \quad (2.4)$$

2.1.3 Computing extremal eigenvalues of a Hermitian matrix using the Lanczos method

In the following, we present a short summary of the Lanczos method and its convergence properties. The presentation is largely based upon [Saa92, Kre14].

Let $A \in \mathbb{C}^{n \times n}$ Hermitian, x a random starting vector and $k \in \mathbb{N}$. In the power method where the dominant eigenvector is approximated simply by repeatedly applying A to x k -times. In comparison, in the Lanczos method the dominant eigenvector of A is approximated inside the Krylov subspace

$$\mathcal{K}_k(A, x) = \{x, Ax, A^2 x, \dots, A^{k-1} x\}.$$

An approximation (μ, u) to the dominant eigenpair of A is chosen inside $\mathbb{R} \times \mathcal{K}_k(A, x)$, by additionally imposing the Galerkin condition

$$Au - \mu u \perp \mathcal{K}_k(A, x). \quad (2.5)$$

Chapter 2. Preliminaries

Given an orthonormal basis U_k for $\mathcal{K}_k(A, x)$, $u \in \mathcal{K}_k(A, x)$ can be written as $u = U_k w$ for some $w \in \mathbb{C}^k$. Furthermore, (2.5) is equivalent to

$$U_k^* A U_k w - \mu w = 0 \iff U_k^* A U_k w = \mu w,$$

making the eigenpairs $(\mu_1, w_1), \dots, (\mu_k, w_k)$ of $U_k^* A U_k$ possible choices for (μ, w) . Usually, the eigenvalues $\mu_1 \leq \dots \leq \mu_k$ are called the Ritz values and vectors $U w_1, \dots, U w_k$ are called the Ritz vectors. In the Lanczos method, the extremal Ritz pairs $(\mu_1, U w_1)$ and $(\mu_k, U w_k)$ are used as approximations to the extremal eigenpairs of A .

An orthonormal basis U_k for $\mathcal{K}_k(A, x)$ and the projected matrix $H_k = U_k^* A U_k$ can be efficiently computed using the Arnoldi algorithm. It can be shown that this leads to H_k tridiagonal:

$$H_k = \begin{pmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \beta_2 & \alpha_3 & \ddots & \\ & & \ddots & \ddots & \beta_{k-1} \\ & & & \beta_{k-1} & \alpha_k \end{pmatrix},$$

making it very simple to compute, as demonstrated in Algorithm 1.

Algorithm 1 Lanczos method

Input: Hermitian matrix $A \in \mathbb{C}^{n \times n}$, starting vector $x \neq 0$, $k \in \mathbb{N}$.

Output: Orthonormal basis $U = [u_1, \dots, u_k]$ of $\mathcal{K}_k(A, x)$.

- 1: $u_1 = x / \|x\|_2$
 - 2: **for** $i = 1, \dots, k-1$ **do**
 - 3: $w = A u_i$
 - 4: $\alpha_i = u_i^* w$
 - 5: $\tilde{u}_{i+1} = w - \alpha_i u_i$
 - 6: $\beta_i = \|\tilde{u}_{i+1}\|_2$
 - 7: $u_{i+1} = \tilde{u}_{i+1} / \beta_i$
 - 8: **end for**
-

As k gets larger, the Krylov subspace $\mathcal{K}_k(A, x)$ contains increasingly better approximations of the extremal eigenpairs. Having in mind that

$$\mathcal{K}_k(A, x) = \{p(A)x : p \text{ polynomial of degree smaller than } k\},$$

the convergence of the Lanczos method can be quantified in following way [Saa92, Lemma 6.1]

$$\tan \angle(v_i, \mathcal{K}_k(A, x)) \leq \min_{p \in \mathbb{P}_{k-1}, p(\lambda_i)=1} \|p(A)\|_2 \tan \angle(v_i, x),$$

where (λ_i, v_i) is an eigenpair of A . Since A is diagonalizable (by being Hermitian), this result

can be further simplified to

$$\tan \angle(v_i, \mathcal{K}_k(A, x)) \leq \min_{p \in \mathbb{P}^{k-1}} \max_{\lambda \in \tilde{\Lambda}_i} \frac{|p(\lambda)|}{|p(\lambda_i)|} \tan \angle(v_i, x), \quad (2.6)$$

with $\tilde{\Lambda}_i = \{\lambda_1, \dots, \lambda_{i-1}, \lambda_{i+1}, \dots, \lambda_n\}$.

In order to estimate the convergence rate from (2.6), we need to consider polynomials p that have a large value at λ_i and small value at the rest of the spectrum, which immediately motivates the use of Chebyshev polynomials. For the special case when $i = 1$, we can set p to be the Chebyshev polynomial of the order $k - 1$ on $[\lambda_2, \lambda_n]$:

$$p(\lambda) := T_{k-1}((2\lambda - \lambda_2 - \lambda_n)/(\lambda_n - \lambda_2)),$$

with T_k the standard Chebyshev polynomial on $[-1, 1]$ of the order $k - 1$. For $|x| > 1$, T_k can be bounded from below by $\frac{1}{2}|x|^{k-1}$, which allows us to bound $|p(\lambda)|/|p(\lambda_i)|$ in (2.6) in the following way:

$$\begin{aligned} \frac{|p(\lambda)|}{|p(\lambda_1)|} &\leq \frac{1}{T_{k-1}((2\lambda_1 - \lambda_2 - \lambda_n)/(\lambda_n - \lambda_2))} = \frac{1}{T_{k-1}(-1 - 2(\lambda_2 - \lambda_1)/(\lambda_n - \lambda_2))} \\ &\leq \frac{1}{T_{k-1}(-1 - 2(\lambda_2 - \lambda_1)/(\lambda_n - \lambda_1))} \leq \frac{1}{2} \left(1 + 2(\lambda_2 - \lambda_1)/(\lambda_n - \lambda_1)\right)^{1-k}, \end{aligned}$$

which proves the exponential convergence of $\tan \angle(v_1, \mathcal{K}_k(A, x))$ to zero with the rate dependent on the relative gap $\frac{\lambda_2 - \lambda_1}{\lambda_n - \lambda_1}$:

$$\tan \angle(v_1, \mathcal{K}_k(A, x)) \leq \frac{1}{2} \left(1 + 2(\lambda_2 - \lambda_1)/(\lambda_n - \lambda_1)\right)^{1-k} \tan \angle(v_1, x). \quad (2.7)$$

Similarly, for the case $i = n$, we obtain the exponential convergence with the rate dependent on the relative gap $\frac{\lambda_n - \lambda_{n-1}}{\lambda_n - \lambda_1}$:

$$\tan \angle(v_n, \mathcal{K}_k(A, x)) \leq \frac{1}{2} \left(1 + 2(\lambda_n - \lambda_{n-1})/(\lambda_n - \lambda_1)\right)^{1-k} \tan \angle(v_n, x). \quad (2.8)$$

The bounds (2.7) and (2.8) quantify the approximation quality of the extremal eigenvectors inside $\mathcal{K}_k(A, x)$ but do not say anything about the convergence of the extremal Ritz values μ_1 and μ_k to λ_1 and λ_n , respectively. These errors can be bounded using the minimax characterisation of extremal eigenvalues (2.4), similarly as in [Saa92, Theorem 6.4], in the following

way

$$\mu_1 - \lambda_1 \leq \frac{\lambda_n - \lambda_1}{(1 + 2(\lambda_2 - \lambda_1)/(\lambda_n - \lambda_1))^{k-1}} \tan \angle(v_1, x) \quad (2.9)$$

$$\lambda_n - \mu_k \leq \frac{\lambda_n - \lambda_1}{(1 + 2(\lambda_n - \lambda_{n-1})/(\lambda_n - \lambda_1))^{k-1}} \tan \angle(v_n, x). \quad (2.10)$$

Remark 2.2. Suppose that (λ_n, v_n) is the dominant eigenpair of A ($|\lambda_n| > |\lambda_1|$). Similarly as in the power method, if the starting vector x is chosen to be orthogonal to v_n , then the largest Ritz value μ_k converges to λ_{n-1} instead of λ_n . In practice, this can be avoided by taking a random starting vector.

Remark 2.3. Let $\varepsilon_{\text{tol}} > 0$ and suppose we are interested in computing ℓ largest eigenvalues of A . In practice, we usually stop the execution of the Lanczos method when the eigenvalue residual R becomes small enough

$$\|R\|_2 := \|AU_\ell - U_\ell\Lambda_\ell\|_2 < \varepsilon_{\text{tol}},$$

where $\Lambda_\ell \in \mathbb{R}^{\ell \times \ell}$ is a diagonal matrix containing the ℓ largest Ritz values and $U_\ell \in \mathbb{R}^{n \times \ell}$ contains the corresponding Ritz vectors.

Remark 2.4. The Lanczos method can also be used for computing the extremal eigenvalues of a generalized symmetric eigenvalue problem

$$Av = \lambda Mv, \quad (2.11)$$

where $A \in \mathbb{C}^{n \times n}$ is Hermitian and $M \in \mathbb{C}^{n \times n}$ is symmetric positive definite matrix. By computing the Cholesky decomposition of $M = LL^*$, we can transform (2.11) into a standard symmetric eigenvalue problem of the following form

$$L^{-1}AL^{-*}w = \lambda w,$$

where $w = L^*v$. When implementing this approach, it is preferable to keep the matrix $L^{-1}AL^{-*}$ in the factorized form, since forming it explicitly would destroy the underlying sparsity pattern.

Remark 2.5. Suppose we are interested in computing the smallest eigenvalue of a symmetric positive definite matrix $A \in \mathbb{R}^{n \times n}$ coming from a spatial discretization of a partial differential equation (PDE) with n degrees of freedom. Even though 2.7 proves the exponential convergence of the Lanczos method, the observed convergence in practice is often very slow for large values of n . This usually happens when the spectrum of the underlying PDE is unbounded, which results in relative gaps $\frac{\lambda_2 - \lambda_1}{\lambda_n - \lambda_1}$ that approach 0 as $n \rightarrow +\infty$. Instead, it is preferred to use the inverse Lanczos method, where, instead of $\mathcal{K}_k(A, x)$, we construct the Krylov subspaces $\mathcal{K}_k(A^{-1}, x)$ of A^{-1} . Since computing the smallest eigenvalues of A is equivalent to computing the largest

eigenvalues of A^{-1} , we can bound the error in the inverse Lanczos method in the following way:

$$\begin{aligned}
 \tan \angle(v_1, \mathcal{K}_k(A^{-1}, x)) &\leq \frac{1}{2} \left(1 + 2 \left(\frac{1}{\lambda_1} - \frac{1}{\lambda_2}\right) / \left(\frac{1}{\lambda_1} - \frac{1}{\lambda_n}\right)\right)^{1-k} \tan \angle(v_1, x) \\
 &= \frac{1}{2} \left(1 + 2 \frac{\lambda_n(\lambda_2 - \lambda_1)}{\lambda_2(\lambda_n - \lambda_1)}\right)^{1-k} \tan \angle(v_1, x) \\
 &= \frac{1}{2} \left(1 + 2 \left(1 - \frac{\lambda_1(\lambda_n - \lambda_2)}{\lambda_2(\lambda_n - \lambda_1)}\right)\right)^{1-k} \tan \angle(v_1, x) \\
 &\leq \frac{1}{2} \left(1 + 2 \left(1 - \frac{\lambda_1}{\lambda_2}\right)\right)^{1-k} \tan \angle(v_1, x) \\
 &= \frac{1}{2} \left(1 + 2 \left(\frac{\lambda_2 - \lambda_1}{\lambda_2}\right)\right)^{1-k} \tan \angle(v_1, x). \tag{2.12}
 \end{aligned}$$

We can see that in the case of the inverse Lanczos method, the convergence rate depends only on the relative gap between λ_1 and λ_2 , which, in the case of PDE discretizations, converges to the relative gap between the first and second eigenvalues of the PDE eigenvalue problem as $n \rightarrow \infty$, and hence it stays bounded away from zero if these two eigenvalues are different. A simple example of this phenomenon is shown in Figure 2.1, where we present the convergence rates of both the Lanczos and the inverse Lanczos method when applied to computing the smallest eigenvalue of the 1D Laplacian with Dirichlet boundary conditions for different values of n .

It is important to note that using $\mathcal{K}_k(A, x)$ or $\mathcal{K}_k(A^{-1}, x)$ makes a pronounced difference from a computational point of view, since constructing the latter involves solving linear systems with A , while forming $\mathcal{K}_k(A, x)$ requires just matrix-vector multiplications. However, in certain cases, the construction of $\mathcal{K}_k(A^{-1}, x)$ can be made more efficient by precomputing the LU decomposition of A at the start.

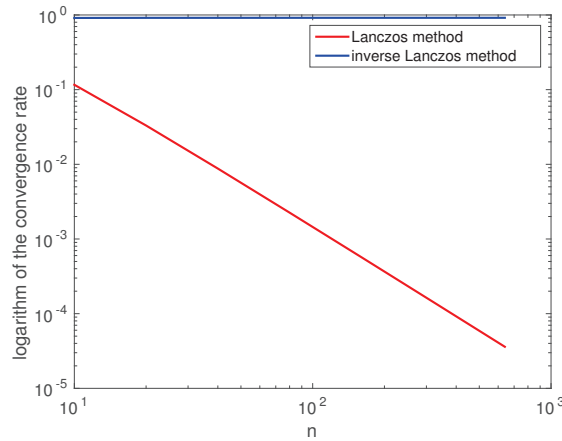


Figure 2.1: Logarithms of the convergence rates (2.7) and (2.12) for computing the smallest eigenvalue of the 1D Laplacian using the Lanczos method and the inverse Lanczos method, respectively.

2.1.4 Analyticity of eigenvalue decomposition

Given the eigenvalue decomposition of A , a question which naturally arises is how the eigenvalues and the eigenvectors change when A is perturbed. Let us consider a family of matrices depending on a parameter μ :

$$A(\mu) : D \rightarrow \mathbb{C}^{n \times n},$$

where D is an open subset of \mathbb{R} . If $A(\mu)$ is analytic w.r.t. $\mu \in D$, a well known result from eigenvalue perturbation theory [Kat95, Chapter 2, Theorem 1.8] states that there exist analytic functions $\lambda_1(\mu), \dots, \lambda_n(\mu)$ and $v_1(\mu), \dots, v_n(\mu)$ on D with only finitely many algebraic singularities describing dependence of eigenvalues and eigenvectors w.r.t. μ , respectively. In fact, the singularities of eigenvalues and eigenvectors occur in the eigenvalue crossings, i.e. $\mu \in D$ such that there exist $i \neq j$ and $\lambda_i(\mu) = \lambda_j(\mu)$. However, it is important to emphasize that not all eigenvalue crossing are necessarily singularities of the eigenvalue or the eigenvector functions.

In the specific case when $A(\mu)$ is a Hermitian family, the following theorem shows existence of an analytic extension of the eigenvalues and the eigenvectors functions in a ball around $\mu_0 \in D$ even if it is not a simple eigenvalue.

Theorem 2.6 (Theorem 1, [Rel69]). *Let $A(\mu)$ be a family of Hermitian matrices on D . For a fixed $\mu_0 \in D$ let λ_0 be an eigenvalue of multiplicity m of $A(\mu_0)$. Then the following holds:*

- *There exist m (not necessarily distinct) complex-valued functions analytic around $z = \mu_0$, denoted by $\lambda_1(\cdot), \dots, \lambda_m(\cdot)$, such that $\lambda_j(\mu_0) = \lambda_0$, $j = 1, \dots, m$ and $\lambda_j(z)$, $j = 1, \dots, m$ are eigenvalues of the analytic extension $A(z)$ near $z = \mu$.*
- *There are no other eigenvalues of $A(z)$ near $\lambda_j(z)$, $j = 1, \dots, m$.*
- *There are m complex-analytic \mathbb{C}^n -functions $v_1(\cdot), \dots, v_m(\cdot)$ such that near $z = \mu$, $v_1(z), \dots, v_m(z)$ are the eigenvectors of $A(z)$.*
- *For $z \in \mathbb{R}$, we have $v_i(z)^* v_j(z) = \delta_{ij}$.*

Theorem 2.6 does not *a priori* provide a radius $r_{\mu_0}(\lambda_0)$ of the analyticity ball for general Hermitian family $A(\mu)$. As shown in the following example, it depends on potential singularities in the eigenvalue and the eigenvector functions, and thus also on the parametric dependence in A .

Example 2.7. *Consider $A : \mathbb{R} \rightarrow \mathbb{C}^{2 \times 2}$ defined in the following way*

$$A(\mu) = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} + \mu \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

The eigenvalues of $A(\mu)$ are given as $\pm \sqrt{1 + \mu^2}$. By extending A to the complex plane, $A(z)$ has eigenvalues $\lambda_{\pm}(z) = \pm \sqrt{1 + z^2}$, complex analytic functions in z for $|z| < 1$. Even though $A(z)$ is

analytic in z , the eigenvalue mappings $\lambda_{\pm}(z)$ have singularities for $z = \pm i$. Note that $A(z)$ is Hermitian only for $\text{Im}z = 0$.

If $A(\mu)$ is an analytic Hermitian family on $B(\mu_0, R)$ defined with

$$A(\mu) = A_0 + \mu A_1 + \mu^2 A_2 + \dots, \quad |\mu - \mu_0| < R,$$

such that $\|A_n\|_2 \leq ac^{n-1}, \forall n \in \mathbb{N}$, for some $a > 0$ and $0 \leq c < \infty$, then a result in [Bau85, Section 8.1.3] allows us to bound $r_{\mu_0}(\lambda)$ from below in the following way:

$$\left(c + \frac{2a}{d}\right)^{-1} \leq r_{\mu_0}(\lambda_0),$$

where $d = \text{dist}(\lambda_0, \lambda(A_0) \setminus \{\lambda_0\})$. In particular, for $A(\mu) = A_0 + \mu A_1$, we have $c = 0$, which results in

$$\frac{d}{2a} \leq r_{\mu_0}(\lambda_0). \tag{2.13}$$

As the lower bound 2.13 matches the radius of analyticity around 0 in Example 2.7, we see that this lower bound is sharp. Another example of this lower bound in practice can be seen in Example 2.8.

Example 2.8. Let $\delta > 0$, and let us consider $A : [-1, 1] \rightarrow \mathbb{C}^{4 \times 4}$ defined by $A(\mu) = A_0 + \mu A_1$, where

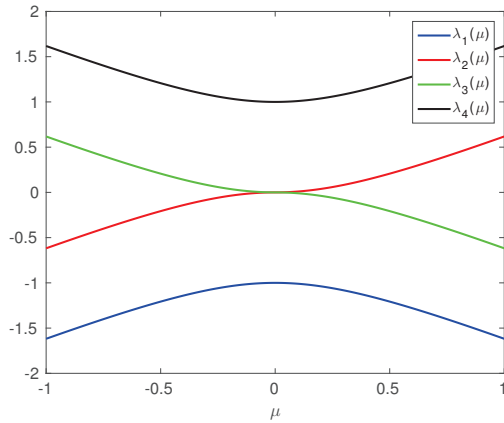
$$A_0 = \begin{bmatrix} \delta & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\delta \end{bmatrix}, \quad A_1 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Since A_0 has a double zero eigenvalue with a gap of δ to the rest of the spectrum and $\|A_1\|_2 = 1$, we obtain a lower bound of $\frac{\delta}{2}$ for the analyticity radius of the double eigenvalue at $\mu = 0$. For general $\mu \in [-1, 1]$ the eigenvalues of $A(\mu)$ are given by the following formula

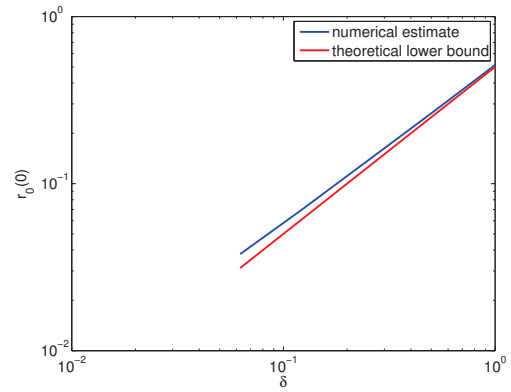
$$\lambda_{1,2,3,4}(\mu) = \pm \frac{\delta}{2} \pm \sqrt{\left(\frac{\delta}{2}\right)^2 + \mu^2},$$

and, as shown in Figure 2.2a. We can see that there are no other eigenvalue crossings besides the one at $\mu = 0$.

For a function $f : \mathbb{R} \rightarrow \mathbb{R}^4$ on $[-1, 1]$ that admits an analytic extension on the Bernstein ellipse E_R around $[-1, 1]$ (foci ± 1 , sum of half-axes equal to R), the interpolation error of f with N Chebyshev nodes on $[-1, 1]$ can be bounded by $C \frac{R+R^{-1}}{(R^N+R^{-N})(R+R^{-1}-2)}$, for some $C > 0$, see e.g. [MH03]. By estimating the convergence rate of the Chebyshev interpolation for v_2 and v_3 (eigenvectors corresponding to the middle two eigenvalues λ_2 and λ_3), we can estimate R_{\max} , the maximal value of R such that E_R is contained in the analyticity domain. Assuming that the estimate for R_{\max} is correct, we can compute a numerical estimate for the analyticity radius at $\mu = 0$, since the length of the shorter half-axis $\frac{R_{\max}-R_{\max}^{-1}}{2}$ is bounded by $r_0(0)$. In Figure 2.2b, a comparison of



(a) Eigenvalues of $A(\mu)$ for $\mu \in [-1, 1]$ in Example 2.8.



(b) Comparison of theoretical lower bounds and numerical estimates for $r_0(0)$ for different values of δ in Example 2.8.

the theoretical lower bounds on the analyticity radius and the numerical estimates for different values of δ is shown. We can see that the numerical estimates and theoretical lower bounds are approximately equal and that, as expected, the analyticity radius $r_{\mu_0}(\lambda_0)$ is proportional to the eigenvalue gap δ .

Multivariate case

In the multivariate case, when $D \subset \mathbb{R}^d$, the previous results do not extend and the eigenvalues and the eigenvectors of $A(\mu)$ are not necessarily analytic functions, as can be seen from the following example.

Example 2.9. Consider $A: \mathbb{R}^2 \rightarrow \mathbb{C}^{2 \times 2}$ defined in the following way

$$A(x, y) = x \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} + y \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

Then $A(x_1, x_2)$ is analytic in both x and y and Hermitian for real x and y . However, its eigenvalues $\lambda_{\pm}(x, y) = \pm \sqrt{x^2 + y^2}$ are not real-analytic with respect to (x, y) in any neighborhood around zero.

Even though the eigenvalues and the eigenvectors are not necessarily analytic w.r.t. μ , in practice we usually observe them to be highly regular functions. This behavior was studied in [AS12] for the following special case of linear parameter dependence

$$A(\mu) = A_1 + \mu^{(1)} A_2 + \dots + \mu^{(d)} A_{d+1}, \quad \forall \mu = (\mu^{(1)}, \dots, \mu^{(d)}) \in D \subset \mathbb{R}^d,$$

where the authors show that an eigenpair $(\lambda(\mu), v(\mu))$ can be extended to jointly complex-analytic function in \mathbb{C}^d as long as it remains simple (separated from the rest of the spectrum).

2.1.5 Perturbation analysis

In the following we discuss eigenvalue sensitivity from a more practical point of view. We present two perturbation results which help us answer the following questions: Given a k -dimensional subspace $\mathcal{U} \subset \mathbb{R}^n$, does it approximate an invariant subspace of a Hermitian matrix A ? If yes, can we and how accurately can we compute approximate eigenvalues of A ? Is there a way to determine which eigenvalues of A are approximated?

Let $U \in \mathbb{C}^{n \times k}$ be an orthonormal basis for \mathcal{U} . To measure the distance of \mathcal{U} to an invariant subspace of A , we first need to compute approximate eigenvalues. As will be shown in the following, good candidates are the eigenvalues of the projected matrix U^*AU . As will be discussed in Remark 2.12, we may assume without loss of generality that U^*AU equals a diagonal matrix containing its eigenvalues $\Lambda_{\mathcal{U}} = \text{diag}(\lambda_{\mathcal{U}}^{(1)}, \dots, \lambda_{\mathcal{U}}^{(k)})$. Given U and $\Lambda_{\mathcal{U}}$, similarly as in the Lanczos method, distance from an invariant subspace can now be measured by computing the eigenvalue residual $R \in \mathbb{C}^{n \times k}$

$$R = AU - U\Lambda_{\mathcal{U}}. \quad (2.14)$$

For $i \in \{1, \dots, n\}$, as $\Lambda_{\mathcal{U}}$ is diagonal, (2.14) implies $|Au_i - \lambda_{\mathcal{U}}^{(i)}u_i| < \|R\|_2$, $i = 1, \dots, k$, where u_i is the i -th column of the matrix U . It is easily shown that $\lambda_{\mathcal{U}}^{(i)}$ is an exact eigenvalue of a perturbed Hermitian matrix, as in the setting of Theorem 2.1.

Corollary 2.10. *With notation as above, there exists $\lambda_i \in \sigma(A)$ such that*

$$|\lambda_i - \lambda_{\mathcal{U}}^{(i)}| < \|R\|_2.$$

Corollary 2.10 is a direct consequence of Theorem 2.1, taking into account that the corresponding matrix X for A is unitary and thus $\kappa(X) = 1$, and it justifies the choice of $\lambda_{\mathcal{U}}^{(i)}$ as approximate eigenvalues, since it proves that there exists an exact eigenvalue in a neighborhood of radius $\|R\|_2$ around $\lambda_{\mathcal{U}}^{(i)}$. However, it is still unclear if it is the only eigenvalue in that neighborhood, or which of n eigenvalues it approximates.

Provided that $\lambda((U_{\perp})^*AU_{\perp})$ is also available, where U_{\perp} is an orthonormal basis of \mathcal{U}^{\perp} , these questions are easily answered using the following perturbation result by Li and Li [LL05], where we set $H_1 = U^*AU$ and $H_2 = (U_{\perp})^*AU_{\perp}$. Assuming that $\lambda(H_1)$ and $\lambda(H_2)$ are sufficiently separated, it allows us to identify which of the eigenvalues of the original matrix are approximated by $\Lambda_{\mathcal{U}} = \text{diag}(\lambda_{\mathcal{U}}^{(1)}, \dots, \lambda_{\mathcal{U}}^{(k)})$.

Theorem 2.11 (Theorem 2, [LL05]). *Let*

$$A = \begin{bmatrix} H_1 & R \\ R^* & H_2 \end{bmatrix}, \text{ and } \tilde{A} = \begin{bmatrix} H_1 & 0 \\ 0 & H_2 \end{bmatrix}$$

be Hermitian matrices with eigenvalues

$$\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n \text{ and } \tilde{\lambda}_1 \leq \tilde{\lambda}_2 \leq \dots \leq \tilde{\lambda}_n,$$

respectively. Define the eigenvalue gaps

$$\begin{aligned}\eta_i &= \begin{cases} \text{dist}(\tilde{\lambda}_i, \lambda(H_2)) & \text{if } \tilde{\lambda}_i \in \lambda(H_1) \\ \text{dist}(\tilde{\lambda}_i, \lambda(H_1)) & \text{if } \tilde{\lambda}_i \in \lambda(H_2) \end{cases} \\ \eta &= \text{dist}(\lambda(H_1), \lambda(H_2)).\end{aligned}$$

Then for $i = 1, \dots, n$, we have

$$|\lambda_i - \tilde{\lambda}_i| \leq \frac{2\|R\|_2^2}{\eta_i + \sqrt{\eta_i^2 + 4\|R\|_2^2}} \quad (2.15)$$

$$\leq \frac{2\|R\|_2^2}{\eta + \sqrt{\eta^2 + 4\|R\|_2^2}}. \quad (2.16)$$

Remark 2.12. Let $A \in \mathbb{C}^{n \times n}$ be a Hermitian matrix and \mathcal{U} a k -dimensional subspace in \mathbb{R}^n . It is easy to show there exists an orthonormal basis $U \in \mathbb{R}^{n \times k}$ for \mathcal{U} such that U^*AU is diagonal. Let $\tilde{U} \in \mathbb{R}^{n \times k}$ be any orthonormal basis for \mathcal{U} . Since the projected matrix $\tilde{U}^*A\tilde{U}$ is also Hermitian, it admits an eigenvalue decomposition of the form $W\Lambda W^* = \tilde{U}^*A\tilde{U}$, with $\Lambda \in \mathbb{R}^{k \times k}$ diagonal and $W \in \mathbb{C}^{k \times k}$ unitary. It is now easy to see that $U := \tilde{U}W$ has the sought properties, it is an orthonormal basis for \mathcal{U} , and $(\tilde{U}W)^*A(\tilde{U}W) = \Lambda$ equals to a diagonal matrix.

2.2 Lyapunov equations

Given $n \in \mathbb{N}$, $A \in \mathbb{R}^{n \times n}$, and $B \in \mathbb{R}^{n \times m}$, we consider the following $n \times n$ Lyapunov equation

$$AX + XA^T = -BB^T. \quad (2.17)$$

In the vectorized form, (2.17) is equivalent to the following linear system

$$\mathcal{A} \text{vec}(X) = (I \otimes A + A \otimes I) \text{vec}(X) = -(B \otimes B) \text{vec}(I_m), \quad (2.18)$$

where $\mathcal{A} = I \otimes A + A \otimes I$. Using the spectral properties of the Kronecker sum, we obtain that (2.17) has a unique solution if and only if $\lambda_i + \lambda_j \neq 0, \forall \lambda_i, \lambda_j \in \lambda(A)$, which we assume in the following. By transposing the whole equation (2.17), we see that both X and X^T are solutions, implying that the solution is necessarily symmetric. Furthermore, if $A \in \mathbb{R}^{n \times n}$ is stable (i.e. the spectrum lies in the left half of the complex plane $\lambda(A) \subset \mathbb{C}_-$), the solution X can be represented in the following way

$$X = \int_0^\infty e^{A\tau} BB^T e^{A^T\tau} d\tau,$$

which immediately yields that X is necessarily positive semidefinite.

In the following, we discuss the conditions that ensure low-rank structure in X and present an approach that exploits this to efficiently solve large-scale Lyapunov equations. Finally, we describe the role of Lyapunov equations in model reduction of linear dynamical systems with control, one of the most important applications.

2.2.1 Low-rank solutions of Lyapunov equations

We have shown that if A is stable, (2.17) has a unique positive semidefinite solution X . Additionally, it has been shown in [Sab06, KT10] that if A is symmetric, then X exhibits a singular value decay if $m \ll n$. More precisely, there exists a matrix $X_k \in \mathbb{R}^{n \times n}$ of rank km such that

$$\|X - X_k\|_F \leq \frac{8\|B\|_F}{\lambda_{\max}(A)} \exp\left(\frac{-k\pi^2}{\log(8\kappa(A))}\right),$$

where $\kappa(A)$ is the condition number of A . This implies that X has an exponential eigenvalue decay:

$$\lambda_k(X) \lesssim \gamma^k, \text{ with } \gamma = \exp\left(\frac{-\pi^2}{m \log(8\kappa(A))}\right),$$

where $\lambda_k(X)$ denotes the k -th largest eigenvalue of X . We see that the decay rate deteriorates and vanishes as $\kappa(A) \rightarrow \infty$. This issue has been resolved in [GK14], where the authors show for certain situations that as $\kappa(A) \rightarrow \infty$ the eigenvalue decay becomes exponential with respect to \sqrt{k} , instead of k :

$$\lambda_k(X) \lesssim \gamma^{\sqrt{k}}, \text{ with } \gamma = \exp(-\pi/\sqrt{2m}).$$

2.2.2 Solving large-scale Lyapunov equations

For $n \lesssim 5000$, a classical approach to solving (2.17) is using a direct method, such as the Bartels-Stewart algorithm [BS72], which requires $\mathcal{O}(n^3)$ operations. For larger values of n , these methods are not computationally feasible as they require the Schur decomposition of A . Instead, various iterative approaches have been proposed, that achieve computational advantage by exploiting sparsity in A and the low-rank structure in the solution. In the following, we follow [Pen00] and describe one of the most popular approaches, the alternating direction implicit (ADI) iteration.

In the ADI method, the solution X is generated as a limit of the iterates X_i , defined in the following way:

$$\begin{aligned} (A + p_i I)X_{i-1/2} &= -BB^T - X_{i-1}(A^T - p_i I), \\ (A + p_i I)X_i^T &= -BB^T - X_{i-1/2}^T(A^T - p_i I), \end{aligned}$$

with $X_0 = 0$ and shift parameters $p_1, p_2, \dots \in \mathbb{C}_-$. This pair of half-iterations is equivalent to the

iteration step

$$X_i = (A - p_i I)(A + p_i I)^{-1} X_{i-1} (A^T - p_i I)(A^T + p_i I)^{-1} - 2p_i (A + p_i I)^{-1} B B^T (A^T + p_i I)^{-1}. \quad (2.19)$$

It can be shown that the errors $E_i = X - X_i$ satisfy the following expression

$$E_i = (r_i(A) r_i(-A)^{-1}) E_0 (r_i(A) r_i(-A)^{-1})^T,$$

where r_i is the polynomial $r_i(x) = (x - p_1 I) \cdots (x - p_2 I) \cdots (x - p_i I)$. Thus, to ensure convergence, the shifts p_1, p_2, \dots need to be chosen in a way that will guarantee $r_i(A) r_i(-A)^{-1} \approx 0$. Assuming that A is diagonalizable, minimizing the spectral radius of $r_i(A) r_i(-A)^{-1}$ leads to the following ADI minimax problem

$$\{p_1, \dots, p_i\} = \operatorname{argmin}_{p_1, \dots, p_i \in \mathbb{C}_-} \max_{x \in \lambda(A)} \frac{|r_i(x)|}{|r_i(-x)|}, \quad (2.20)$$

which indicates criteria for choosing the shifts. As the spectrum $\lambda(A)$ is usually not available, in practice, (2.20) is often relaxed by replacing $\lambda(A)$ with E (compact subset of \mathbb{C} such that $\lambda(A) \subset E$):

$$\{p_1, \dots, p_i\} = \operatorname{argmin}_{p_1, \dots, p_i \in \mathbb{C}_-} \max_{x \in E} \frac{|r_i(x)|}{|r_i(-x)|}. \quad (2.21)$$

The relaxed ADI minimax problem has been solved exactly (see [Wac63]) only for the case of symmetric A . For the general case, several heuristic strategies for choosing close to optimal shifts have been proposed, see, e.g. [Pen00, Wac88, FG13].

The ADI method can be implemented in a way that exploits positive definiteness in X as well as the low-rank structure in X described in Section 2.2.1. In the low-rank version of the ADI method (LR-ADI), the iterates are substituted by their Cholesky decompositions $X_i = Z_i Z_i^T$, while the iteration step (2.19) can be written in the following way

$$Z_i = [(A - p_i I)(A + p_i I)^{-1} Z_{i-1} \sqrt{-2p_i} (A + p_i I)^{-1} B],$$

with $Z_1 = \sqrt{-2p_1} (A + p_1 I)^{-1} B$. A drawback of LR-ADI is that the memory requirements and the computational cost per iteration are increasing with each iteration, since the low-rank factor Z_i is enlarged by m in each iteration ($\operatorname{rank}(Z_i) \leq mi$, where $m = \operatorname{rank}(B)$). However, in practice, LR-ADI is an efficient method since the required number of iterations is usually low. Furthermore, the effect of this drawback can be further reduced by performing low-rank truncation of the iterates.

Other popular methods for solving large-scale Lyapunov equations include the Rational Krylov projection method [HR92] and the extended Arnoldi method [Sim07]. In these methods, the approximate solution of the original Lyapunov equation is computed by projecting (2.17) onto k -dimensional (rational) Krylov subspaces. Solving the projected problem is equivalent to solving a small-scale $k \times k$ Lyapunov equation which can be solved efficiently using the

Bartels-Stewart algorithm, since, in practice, we usually have $k \ll n$. Projection techniques can also be used to accelerate the convergence of the ADI method. For example, in [BLT09], the Galerkin projection onto subspace $V_k \otimes V_k$, where V_k is an orthonormal basis for the column space of the current ADI iterate $\mathcal{V}_k = \text{range}(Z_k)$, is used for computing an approximate solution of the form $\tilde{X} = V_k R_k V_k^*$.

Remark 2.13. *As shown in [HS95, KPT14], Krylov subspace methods for solving Lyapunov equations can be effectively preconditioned with a few steps of the ADI method. For example, one step of the ADI method with a single shift p defines the following preconditioner for (2.18)*

$$\mathcal{P}_{\text{ADI}}^{-1} = (A - pI)^{-1} \otimes (A - pI). \quad (2.22)$$

Finding the optimal shift p in (2.22) is equivalent to solving (2.21) with $i = 1$. As shown in [Sta91], for the case of a symmetric A , the optimal shift p equals $\sqrt{\lambda_{\max}(A)\lambda_{\min}(A)}$.

In a similar fashion, it is possible to derive a preconditioner for (2.18) based on the first ℓ steps of the sign function iteration for Lyapunov equations [KPT14]. In particular, for $\ell = 1$, this gives rise to the following preconditioner

$$\mathcal{P}_{\text{sign}}^{-1} = \frac{1}{2c} (I \otimes I + c^2 A^{-1} \otimes A^{-1}), \quad (2.23)$$

with the scaling factor $c = \sqrt{\frac{\|A\|_2}{\|A^{-1}\|_2}}$, which can be approximated using $\|M\|_2 \approx \sqrt{\|M\|_1 \|M\|_\infty}$, see, e.g., [SB08]. Other known choices of preconditioners for (2.18) include the classical Jacobi and SSOR preconditioning [HS95].

Remark 2.14. *The ADI method can be extended to address generalized Lyapunov equations of the form*

$$AXE^T + EXA^T = -BB^T,$$

where $A, E \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, with E symmetric positive definite and $\lambda E - A$ a stable pencil. Similarly as in LR-ADI, this extension can be formulated in terms of the low-rank Cholesky factors Z_i , which is also known as the generalized low-rank ADI [Sty08].

2.2.3 Lyapunov equation for Gramians of linear control systems

Suppose we are given the following continuous linear time-invariant dynamical system with control

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t), \end{aligned}$$

with system matrices $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{\ell \times n}$, state vector $x(t) \in \mathbb{R}^n$, input control vector $u(t) \in \mathbb{R}^m$ and output function $y(t) \in \mathbb{R}^\ell$. Furthermore, we assume that A is stable ($\text{Im}(\sigma(A)) \subset (-\infty, 0)$). The quantity of interest is usually the input-to-output mapping of the

given dynamical system, which can be very difficult to compute for very large values of n . To address this problem, we aim to find a reduced-order model

$$\begin{aligned}\tilde{x}(t) &= \tilde{A}\tilde{x}(t) + \tilde{B}u(t), \\ \tilde{y}(t) &= \tilde{C}\tilde{x}(t),\end{aligned}$$

with $\tilde{A} \in \mathbb{R}^{k \times k}$, $\tilde{B} \in \mathbb{R}^{k \times m}$, $\tilde{C} \in \mathbb{R}^{\ell \times k}$, $\tilde{x}(t) \in \mathbb{R}^k$, $\tilde{y}(t) \in \mathbb{R}^\ell$ and $k \ll n$.

Ideally, when reducing the state space, we would like to remove states that are either

- hard to reach: input energy to guide the system into the state is very high;
- hard to observe: output energy generated from system being in the state is very low.

This idea is implemented in the balanced truncation algorithm [Moo81, PS82], which preserves stability of the dynamical system and provides computable error bounds. In order to provide the reduced model, the balanced truncation algorithm relies upon computation of the controllability Gramian P and the observability Gramian Q which are defined as the unique symmetric positive semidefinite solutions $P, Q \in \mathbb{R}^{n \times n}$ of the following Lyapunov equations:

$$\begin{aligned}AP + PA^T &= -BB^T, \\ A^TQ + QA &= -C^TC.\end{aligned}$$

Given the Cholesky decompositions of the computed Gramians $P = P_C^T P_C$ and $Q = Q_C^T Q_C$, the optimal projection bases $W, V \in \mathbb{R}^{n \times k}$ are extracted as the dominant left and right singular vectors of $P_C Q_C^T$, respectively, while the resulting reduced-order model is constructed as follows

$$\tilde{A} = W^T A V, \quad \tilde{B} = W^T B, \quad \tilde{C} = C V, \quad \tilde{x}(t) = V x(t), \quad \text{and} \quad \tilde{y}(t) = \tilde{C} \tilde{x}(t).$$

2.3 Reduced basis method

The reduced basis (RB) method provides a framework for the solution of parameter-dependent PDEs [RHP08]. It consists of an *offline* phase, where solutions of the PDEs are solved for suitably chosen parameter values and their solutions are collected in a (low-dimensional) subspace. In the subsequent *online* phase, approximate solutions are computed inside this subspace using a Galerkin projection approach. This may speed up the solution process dramatically, especially if the PDE needs to be solved for many parameter values. *A posteriori* error analysis is an important part of the RB method to ensure its reliability.

In this section, we present a short summary of the RB method for parameter-dependent symmetric elliptic coercive partial differential equation (PDE), which is largely based on the survey paper [RHP08]. For more details see also [HRS16, QMN16].

2.3.1 Model problem

Given $\mu \in D \subset \mathbb{R}^d$, we are interested in computing the solution $u(\mu) \in X(\Omega)$ (or an output quantity $l(u(\mu))$) of the following parametrized PDE given in its weak formulation

$$a(u(\mu), v; \mu) = f(v), \quad \forall v \in X, \quad (2.24)$$

where the parameter domain D is a compact subset of \mathbb{R}^d , $a(\cdot, \cdot; \mu) : X \times X \rightarrow \mathbb{R}$ is a symmetric bilinear form for all $\mu \in D$, X is a Hilbert space of functions on Ω .

As we consider second-order elliptic partial differential equations, we have $H_0^1(\Omega) \subset X \subset H^1(\Omega)$. Furthermore, we assume that $a(\cdot, \cdot; \mu)$ is continuous and coercive for all $\mu \in D$ w.r.t. the inner product and the induced norm on $H^1(\Omega)$. As this implies that $a(\cdot, \cdot; \mu)$ defines the energy inner product and the induced energy norm on X for all $\mu \in D$, instead of $(\cdot, \cdot)_{H^1(\Omega)}$ and $\|\cdot\|_{H^1(\Omega)}$, we equip X with a scalar product and an equivalent norm that is more suitable for *a posteriori* error estimation:

$$\begin{aligned} (u, v)_X &= a(u, v; \bar{\mu}) + \tau(u, v)_{L^2(\Omega)} \\ \|u\|_X &= \sqrt{(u, u)_X}, \end{aligned} \quad (2.25)$$

where $\bar{\mu}$ is a specifically chosen reference parameter value in D and $\tau > 0$. The choice of τ is discussed in more detail in Remark 2.16. This allows us to define the coercivity and the continuity constants $\alpha(\mu)$ and $\gamma(\mu)$, respectively, in the following way

$$\begin{aligned} \alpha(\mu) &= \inf_{u \in X \setminus \{0\}} \frac{a(u, u; \mu)}{\|u\|_X^2} \\ \gamma(\mu) &= \sup_{u \in X \setminus \{0\}} \sup_{v \in X \setminus \{0\}} \frac{a(u, v; \mu)}{\|u\|_X \|v\|_X}. \end{aligned}$$

Our hypothesis on coercivity and continuity of a can now be precisely stated as follows:

$$\begin{aligned} \exists \alpha_0 > 0: \quad & \alpha(\mu) > \alpha_0, \quad \forall \mu \in D; \\ \exists \gamma_0 < +\infty: \quad & \gamma(\mu) < \gamma_0, \quad \forall \mu \in D. \end{aligned} \quad (2.26)$$

Finally, we assume that a admits an affine linear decomposition w.r.t. μ : there exists $Q \in \mathbb{N}$, (smooth) functions $\theta_1, \dots, \theta_Q : D \rightarrow \mathbb{R}$, μ -independent symmetric bilinear forms $a_1, \dots, a_Q : X \times X \rightarrow \mathbb{R}$ such that

$$a(u, v; \mu) = \theta_1(\mu) a_1(u, v) + \dots + \theta_Q(\mu) a_Q(u, v), \quad \forall \mu \in D. \quad (2.27)$$

As we will see in the following sections, this assumption is crucial for the efficient implementa-

tion of the RB method.

Remark 2.15. *Assumption (2.27) is commonly found in literature concerning parameter-dependent PDEs, linear systems and eigenvalue problems, since deriving computationally efficient algorithms is usually not possible in case of a general parametric dependence. It holds with small Q for $a(\cdot, \cdot; \mu)$ arising from a number of important applications such as:*

- *parameter-dependent PDE with parametrized coefficients on disjoint subdomains or parametrized geometry (e.g. see [RHP08]),*
- *truncated Karhunen-Loeve (KL) or Polynomial Chaos (PC) expansion of a random field (see e.g. [AS12]).*

In a discretized setting, (2.27) is equivalent to approximating a parameter-dependent Hermitian matrix $A(\mu) \in \mathbb{R}^{n \times n}$ with $Q \ll n^2$ constant Hermitian matrices A_1, \dots, A_Q and corresponding functions $\theta_1, \dots, \theta_Q : D \rightarrow \mathbb{R}$ in the following way:

$$A(\mu) = \theta_1(\mu)A_1 + \dots + \theta_Q(\mu)A_Q, \quad \forall \mu \in D.$$

Even when $a(\cdot, \cdot; \mu)$ and $A(\mu)$ do not a priori admit an affine linear decomposition, in certain cases it may still be possible to approximate them very well by a short affine linear decomposition. For example, $a(\cdot, \cdot; \mu)$ can be approximated using the Empirical Interpolation Method [BMNP04] and $A(\mu)$ using its discrete version the Matrix Discrete Empirical Interpolation Method [NMA15].

An important specific case of affine linear dependence is when $a(\cdot, \cdot; \mu)$ is an affine function in μ , which has been studied in more detail in [AS12, CCDS13]. It can be easily shown that in this case the functions θ_i are linear functions in μ and $a(\cdot, \cdot; \mu)$ admits the following decomposition:

$$a(\cdot, \cdot; \mu) = a_1(\cdot, \cdot) + \mu^{(1)} a_2(\cdot, \cdot) + \dots + \mu^{(d)} a_{d+1}(\cdot, \cdot), \quad (2.28)$$

where $\mu^{(i)}$ denotes the i -th component of μ . In fact, any $a(\cdot, \cdot; \mu)$ satisfying Assumption 2.27 can be represented in the form (2.28) by setting $D := \text{Im}(\theta) \subset \mathbb{R}^Q$. Some examples of $a(\cdot, \cdot; \mu)$ satisfying (2.28) are the cookie example from [KT11] and the thermal block example from [HNPR10].

2.3.2 Finite element discretization

A conforming finite element discretization on Ω with n degrees of freedom of (2.24) leads to the following parameter-dependent linear system

$$A(\mu)x(\mu) = (\theta_1(\mu)A_1 + \dots + \theta_Q(\mu)A_Q)x(\mu) = b, \quad \forall \mu \in D, \quad (2.29)$$

with constant Hermitian matrices $A_1, \dots, A_Q \in \mathbb{R}^{n \times n}$. Furthermore, since we are using a conforming finite element discretization, i.e. the finite element space is a subspace of X , the

discretized coercivity constant $\alpha^n(\mu)$ satisfies

$$0 < \alpha(\mu) = \inf_{u \in X \setminus \{0\}} \frac{a(u, u; \mu)}{\|u\|_X^2} \leq \inf_{v \in \mathbb{R}^n \setminus \{0\}} \frac{v^T A(\mu) v}{v^T X v} = \alpha^n(\mu), \quad \forall \mu \in D,$$

with $X = A(\bar{\mu}) + \tau M \in \mathbb{R}^{n \times n}$ denotes the discretization of the scalar product (2.25), where $M \in \mathbb{R}^{n \times n}$ is the mass matrix. As a result, we have that $A(\mu)$ is positive definite for all $\mu \in D$. Similarly, we can obtain that the discretized continuity constant satisfies $\gamma^n(\mu) < \gamma(\mu), \forall \mu \in D$.

Remark 2.16. *Computing the discretized coercivity constant $\alpha^n(\mu)$ is equivalent to computing the smallest eigenvalue of the following generalized eigenvalue problem*

$$A(\mu) v = \lambda X v, \tag{2.30}$$

which can be done, for example, using the Lanczos method, see Remark 2.4.

If we denote with $\lambda_1 \leq \dots \leq \lambda_n$ the eigenvalues of $A(\bar{\mu}) v = \lambda M v$, it can be easily shown that $\frac{\lambda_i}{\lambda_i + \tau}, i = 1, \dots, n$, is an eigenvalue of the eigenvalue problem

$$A(\bar{\mu}) v = \lambda X v. \tag{2.31}$$

Clearly, the value of constant $\tau > 0$ influences the eigenvalue distribution in (2.31). Applying the Lanczos method to (2.31) leads to the following convergence rate in (2.7):

$$1 + 2 \frac{(\lambda_2 - \lambda_1)(\lambda_n + \tau)}{(\lambda_n - \lambda_1)(\lambda_2 + \tau)}. \tag{2.32}$$

As described in Remark 2.5, in case of PDE discretizations, the spectrum of the underlying PDE eigenvalue problem is often unbounded which leads to $\frac{\lambda_n + \tau}{\lambda_n - \lambda_1} \approx 1$ as $n \rightarrow \infty$. In practice, the constant τ is often chosen to be equal to λ_1 . When inserted in (2.32), this leads to a convergence rate of

$$1 + 2 \frac{\lambda_2 - \lambda_1}{2\lambda_1}, \tag{2.33}$$

which is independent of the discretization size for sufficiently large values of n and approximately equal to that of the inverse Lanczos method. Therefore, as described in Remark 2.4, the eigenvalue problem (2.31) can be solved efficiently by computing the Cholesky decomposition of X and without inverting $A(\bar{\mu})$.

The discussion above provides only the convergence rate (2.33) for the Lanczos method when solving (2.31). However, we hope that, whenever μ is close to $\bar{\mu}$, the convergence rates of the Lanczos method for solving (2.30) are not going to be significantly different. Note that $\bar{\mu}$ is often set to be the "central" point of the parameter domain D .

2.3.3 Offline phase

In the offline phase of the RB method, we subsequently select M parameter samples $\mu_1, \dots, \mu_M \in \Xi$, where $\Xi \subset D$ is a training set of finitely many (typically a few thousand) parameter values, and construct the subspaces $\mathcal{V}_1 \subseteq \dots \subseteq \mathcal{V}_M \subset \mathbb{R}^N$. The parameter samples are selected by a greedy strategy aiming at to minimizing an estimate of the error. Assuming that the first k samples have been processed, the $(k+1)$ -th step of this procedure consists of selecting

$$\mu_{k+1} = \operatorname{argmax}\{\Delta_k(\mu) : \mu \in \Xi\}, \quad (2.34)$$

where $\Delta(\mu)$ is an error estimator, see Section 2.3.5 below. Then, by solving (2.29) with $\mu = \mu_{k+1}$, we obtain $x(\mu_{k+1})$ and the subspace \mathcal{V}_k is extended to a new subspace

$$\begin{aligned} \mathcal{V}_{k+1} &= \mathcal{V}_k + \operatorname{span}\{x(\mu_{k+1})\} \\ &= \operatorname{span}\{x(\mu_1), x(\mu_2), \dots, x(\mu_{k+1})\}. \end{aligned}$$

2.3.4 Online phase

In the online phase, assuming $\dim(\mathcal{V}_M) = M$ and that an orthonormal basis $V \in \mathbb{R}^{N \times M}$ of \mathcal{V}_M is available, we compute using Galerkin projection an approximate solution of the linear system (2.29) for an arbitrary parameter value $\mu \in D$ as $\tilde{x}(\mu) = Vy(\mu)$, where $y(\mu)$ is the solution of the compressed linear system

$$(V^T A(\mu) V)y(\mu) = V^T b. \quad (2.35)$$

Since M is usually small, a standard direct solver for linear systems can be used to solve (2.35). To setup the linear system (2.35) efficiently, we use the affine linear decomposition of $A(\mu)$ (2.27) to obtain

$$V^T A(\mu) V = \theta_1(\mu) V^T A_1 V + \dots + \theta_Q(\mu) V^T A_Q V.$$

Having precomputed matrices $V^T A_q V$ for $q = 1, \dots, Q$ then allows us to attain a complexity of $O(QM^2 + M^3)$ for constructing and solving (2.35).

The accuracy of the approximate solution $\tilde{x}(\mu)$ can be quantified using the error estimator $\Delta(\mu)$ described in the next subsection.

2.3.5 Error estimation

Given an approximate solution of the form $\tilde{x}(\mu) = Vy(\mu)$ for a fixed parameter sample $\mu \in D$, the norm of the residual $r(\mu) = b - A(\mu)\tilde{x}(\mu)$ can be computed from

$$\begin{aligned} \|r(\mu)\|_X^2 &= (b - A(\mu)Vy(\mu))^T X (b - A(\mu)Vy(\mu)) \\ &= b^T X b - 2b^T X A(\mu)Vy(\mu) + y(\mu)^T V^T A(\mu)^T X A(\mu)Vy(\mu). \end{aligned}$$

For $A(\mu)$ as in (2.29), we have

$$\begin{aligned} \|r(\mu)\|_X^2 &= b^T X b - 2 \sum_{q=1}^Q \theta_q(\mu) b^T X A_q V y(\mu) \\ &+ \sum_{q_1, q_2=1}^Q \theta_{q_1}(\mu) \theta_{q_2}(\mu) y(\mu)^T V^T A_{q_1}^T X A_{q_2} V y(\mu). \end{aligned} \quad (2.36)$$

If we precompute and store the parameter-independent quantities $b^T X A_q V$ and $V^T A_{q_1}^T X A_{q_2} V$ for $q, q_1, q_2 = 1, \dots, Q$, then $\|r(\mu)\|_X$ can be computed in $O(Q^2 M^2)$ operations. The difference to the true solution $x(\mu)$ can then be estimated as

$$\|x(\mu) - \tilde{x}(\mu)\|_X \leq \frac{\|r(\mu)\|_X}{\lambda_{\min}(A(\mu), X)} \leq \frac{\|r(\mu)\|_X}{\lambda_{\text{LB}}(\mu)} =: \Delta(\mu), \quad (2.37)$$

where $\lambda_{\min}(A(\mu), X)$ denotes the smallest eigenvalue of the generalized eigenvalue problem $A(\mu)v = \lambda Xv$ and $\lambda_{\text{LB}}(\mu) > 0$ is a lower bound for $\lambda_{\min}(A(\mu), X)$. Effective and reliable nonnegative bounds on $\lambda_{\min}(A(\mu), X)$ can be efficiently computed, for example, using the Successive Constraint Method (SCM) [HRSP07], which will be described in more detail in Section 3.1.

The error estimator $\Delta_k(\mu)$ used in (2.34) to guide the sampling strategy in the offline phase is defined in an analogous way, with V replaced by a basis V_k of \mathcal{V}_k .

Remark 2.17. *We have seen in Section 2.3.3 that the next parameter sample μ_{k+1} is computed as the maximizer of the error estimate $\Delta_k(\mu)$ on Ξ . In every iteration, this requires recomputing $\tilde{x}(\mu)$ and $\|r(\mu)\|_X$ on the whole training set Ξ , which can become computationally quite expensive. Instead, as explained in [HSZ14], we can optimize the search for μ_{k+1} by using the error estimates from the previous iteration. As $k \rightarrow \infty$, the error estimates (2.37) converge to 0. Even though the convergence is not monotonic, it is reasonable to assume what is known as the saturation assumption: there exists $C_{\text{sat}} > 0$ such that the following holds*

$$\Delta_\ell(\mu) < C_{\text{sat}} \Delta_k(\mu), \quad \forall \ell > k, \forall \mu \in D. \quad (2.38)$$

We assume that the elements in Ξ are sorted descendingly according to the error estimate (2.37) from the previous iteration, and look for μ_{k+1} by iterating over Ξ . We sequentially recompute $\tilde{x}(\mu)$ and $\|r(\mu)\|_X$ and keep track of the current maximum error estimate Δ_{max} as well as the point $\mu_{\text{max}} \in \Xi$ where it was attained. Reaching a point $\mu \in \Xi$ such that $C_{\text{sat}} \Delta_k(\mu) < \Delta_{\text{max}}$, allows us to skip all the remaining elements of Ξ , and simply set $\Delta_{k+1}(\mu) = \Delta_k(\mu)$, since (2.38) ensures that their error estimates will be smaller than the current maximum Δ_{max} .

Remark 2.18. *The reduced basis method can also be applied to more general PDEs which do not satisfy the coercivity assumption (2.26), but instead satisfy the inf-sup condition*

$$\exists \beta_0 > 0: \beta(\mu) = \inf_{u \in X \setminus \{0\}} \sup_{v \in X \setminus \{0\}} \frac{a(u, v; \mu)}{\|u\|_X \|v\|_X} \geq \beta, \quad \forall \mu \in D.$$

As explained in [HRS16, Section 6.4], it is often assumed that also the finite element discretization

Chapter 2. Preliminaries

of $a(\cdot, \cdot; \mu)$ satisfies the inf-sup condition

$$\exists \beta_0^n > 0 : \beta^n(\mu) = \inf_{u \in \mathbb{R}^n} \sup_{v \in \mathbb{R}^n} \frac{u^T A(\mu) v}{\sqrt{u^T X u} \sqrt{v^T X v}}.$$

Having the inf-sup condition instead of the coercivity assumption requires the use of slightly different a posteriori error estimates [HKC⁺ 10]:

$$\|x(\mu) - \tilde{x}(\mu)\|_X \leq \frac{\|r(\mu)\|_X}{\beta^n(\mu)} \leq \frac{\|r(\mu)\|_X}{\beta_{\text{LB}}(\mu)},$$

where $\beta_{\text{LB}}(\mu)$ is a nonnegative lower bound for $\beta^n(\mu)$. Such lower bounds can be efficiently computed using, for example, the natural-norm SCM [HKC⁺ 10].

3 Low-rank approach for parameter dependent Hermitian eigenvalue problem

This chapter is concerned with methods for approximating the smallest eigenvalue of $A(\mu)$

$$\lambda_{\min}(A(\mu)), \quad \mu \in D, \quad (3.1)$$

for *many* different values of $\mu \in D$, where $A : D \rightarrow \mathbb{C}^{n \times n}$ is a matrix-valued function on a compact subset $D \subset \mathbb{R}^d$ such that $A(\mu)$ is Hermitian for every $\mu \in D$. For simplicity, we assume that D is a hyperrectangle in \mathbb{R}^d . We consider a large-scale setting, where applying a standard eigensolver, such as the Lanczos method [BDD⁺00], is computationally feasible only for a few values of μ but would become too expensive for many (e.g., thousand) parameter values.

As discussed in Section 2.1.4, if A depends analytically on μ then the smallest eigenvalue inherits analyticity *if* $\lambda_{\min}(A(\mu))$ remains simple [Kat95]. As shown in [AS12], this can be used to approximate $\lambda_{\min}(A(\mu))$ very well by high-order Legendre polynomials (for $d = 1$) or sparse tensor products of Legendre polynomials (for $d > 1$ if D is a hypercube). Requiring $\lambda_{\min}(A(\mu))$ to stay simple on the whole of D is, however, a rather strong condition. In general, there are eigenvalue crossings at which $\lambda_{\min}(A(\mu))$ is Lipschitz continuous only. For larger d , keeping track of eigenvalue crossings explicitly appears to be a rather daunting task and we therefore aim at a method for solving (3.1) that benefits only implicitly from piecewise regularity.

One of the simplest approaches to address (3.1) is to use Gershgorin's theorem [Joh89] for estimating the smallest eigenvalue, but the accuracy of the resulting estimate is usually insufficient and limits the scope of applications severely. Without any further assumptions on the dependence of $A(\mu)$ on μ , it is usually not possible to improve on this simple approach, since the smallest eigenvalue of $A(\mu)$ is computationally intractable, especially when d is large. However, several more sophisticated approaches are available if $A(\mu)$ also admits an affine linear decomposition with respect to μ :

$$A(\mu) = \theta_1(\mu)A_1 + \cdots + \theta_Q(\mu)A_Q, \quad \forall \mu \in D, \quad (3.2)$$

with $Q \ll n^2$, Hermitian matrices $A_1, \dots, A_Q \in \mathbb{C}^{n \times n}$, and functions $\theta_1, \dots, \theta_Q : D \rightarrow \mathbb{R}$. For example, eigenvalue perturbation analysis can be used to locally approximate the smallest

eigenvalues [NVP05, VRP02]. Currently, the most commonly used approach is the Successive Constraint Method (SCM; see [HRSP07]), probably due to its generality and relative simplicity. Variants of SCM for computing smallest singular values can be found in [SVH⁺06, HKC⁺10], while an extension of SCM to non-linear problems and alternative heuristic strategies have been proposed in [MN15]. Various subspace approaches based on additional conditions on the parameter dependencies have been proposed in [MMO⁺00, PRV⁺02]. In the context of eigenvalue optimization problems, subspace acceleration has been discussed in [DVM12, KV14, MYK14].

We present a subspace-accelerated variant of SCM which can be summarized as follows. Given M parameter samples $\mu_1, \mu_2, \dots, \mu_M$, we consider the subspace \mathcal{V} containing the eigenvectors belonging to one or several smallest eigenvalues of $A(\mu_i)$ for $i = 1, \dots, M$. The smallest Ritz value of $A(\mu)$ with respect to \mathcal{V} immediately yields an upper bound for $\lambda_{\min}(A(\mu))$. A lower bound is obtained by combining this upper bound with a perturbation argument, which requires knowledge on the involved eigenvalue gap. We show that this gap can be estimated by adapting the linear programming approach used in SCM for computing the lower bounds. Having both the upper and the lower bounds for $\lambda_{\min}(A(\mu))$ enables the definition of an error estimate that drives the greedy strategy for selecting the next parameter sample μ_{M+1} . The whole procedure is stopped once the error estimate is uniformly small on D or, rather, on a surrogate of D . The considered numerical experiments indicate that our subspace approach significantly accelerates convergence compared to SCM.

The rest of this chapter is largely based on [SK16] and is organized as follows. In Section 3.1, we first give an overview of SCM. Additionally, we discuss its interpolation properties and point out a limitation on the quality of the lower bounds that can possibly be attained when solely using the information taken into account by SCM. In Section 3.2, we present our novel subspace-accelerated approach for solving (3.1). Furthermore, we show that the new approach has better interpolation properties than SCM and present *a priori* convergence estimates. Motivated by the fast convergence of the upper bounds in the novel approach, in Section 3.3, we also introduce residual-based lower bounds which are less reliable but sometimes converge much faster. In Section 3.4, we present the full algorithm and discuss implementational details, while in Section 3.5 we discuss various applications of the approach and present the accompanying numerical experiments.

3.1 Successive constraint method

In the following, we recall the Successive Constraint Method (SCM) from [HRSP07] and derive new theoretical properties. The basic idea of SCM is to exploit (3.2) in order to construct reduced-order models for (3.1) that allow efficient evaluation of lower and upper bounds for $\lambda_{\min}(A(\mu))$.

3.1.1 Linear optimization problem for $\lambda_{\min}(A(\mu))$

Assumption 3.2, together with the characterization of the smallest eigenvalue as the minimal value of the Rayleigh quotient (2.1), allows us to obtain the following expression:

$$\begin{aligned}\lambda_{\min}(A(\mu)) &= \min_{\substack{u \in \mathbb{C}^n \\ u \neq 0}} \frac{u^* A(\mu) u}{u^* u} = \min_{\substack{u \in \mathbb{C}^n \\ u \neq 0}} \sum_{q=1}^Q \theta_q(\mu) \frac{u^* A_q u}{u^* u} \\ &= \min_{\substack{u \in \mathbb{C}^n \\ u \neq 0}} \theta(\mu)^T R(u) = \min_{y \in \mathcal{Y}} \theta(\mu)^T y,\end{aligned}\tag{3.3}$$

where we have defined the vector-valued functions $\theta : D \rightarrow \mathbb{R}^Q$, $R : \mathbb{C}^n \setminus \{0\} \rightarrow \mathbb{R}^Q$ as

$$\theta(\mu) := [\theta_1(\mu), \dots, \theta_Q(\mu)]^T, \quad R(u) := \left[\frac{u^* A_1 u}{u^* u}, \dots, \frac{u^* A_Q u}{u^* u} \right]^T,\tag{3.4}$$

and set $\mathcal{Y} := \text{im}(R)$. It follows from (3.3) that the computation of $\lambda_{\min}(A(\mu))$ is equivalent to optimizing the linear functional $\theta(\mu)$ over \mathcal{Y} . The constraint set \mathcal{Y} is called the *joint numerical range* of matrices A_1, \dots, A_Q , which is generally not convex; see [GJK04]. Thus, standard optimization techniques cannot be used to reliably solve (3.3). To circumvent this, in SCM, the set \mathcal{Y} is approximated from above and from below using convex polyhedra, which, in turn allows for the use of standard linear programming (LP) techniques to yield lower and upper bounds for $\lambda_{\min}(A(\mu))$.

3.1.2 Bounding box

As explained above, in order to compute a lower bound for $\lambda_{\min}(A(\mu))$, we need to construct a convex polyhedron containing \mathcal{Y} . More precisely, we need to find constraint matrices $C \in \mathbb{R}^{m \times Q}$ and $b \in \mathbb{R}^m$ such that the following linear program is bounded for all $\theta \in \Theta := \{\theta(\mu) : \mu \in D\}$

$$\begin{aligned}\min_{y \in \mathbb{R}^Q} \quad & \theta^T y \\ \text{s.t.} \quad & Cy \geq b\end{aligned}\tag{3.5}$$

and \mathcal{Y} is contained in its feasible set. The dual linear program of (3.5) has the following form

$$\begin{aligned}\max_{z \in \mathbb{R}^m} \quad & b^T z \\ \text{s.t.} \quad & C^T z = \theta \\ & z \geq 0\end{aligned}\tag{3.6}$$

To ensure that (3.5) is bounded $\forall \theta \in \Theta$, it is sufficient to show that (3.6) is feasible for every $\theta \in \Theta$, i.e. that for every μ , the coefficient vector $\theta(\mu)$ can be represented as a non-negative linear combination of the individual constraints (rows of the matrix C). The following lemma will help us generate suitable constraints in (3.5).

Lemma 3.1. *Let $\theta \in \mathbb{R}^Q$, $A_1, \dots, A_Q \in \mathbb{C}^{n \times n}$ Hermitian matrices and let \mathcal{Y} be their joint numeri-*

Chapter 3. Low-rank approach for parameter dependent Hermitian eigenvalue problem

cal range. If $\lambda_{\min}(\theta_1 A_1 + \dots + \theta_Q A_Q) = \lambda_\theta$, then

$$\theta^T y \geq \lambda_\theta, \quad \forall y \in \mathcal{Y}. \quad (3.7)$$

Moreover, there exists $y_\theta \in \mathcal{Y}$ such that $\theta^T y_\theta = \lambda_\theta$, i.e. the hyperplane $\{y : \theta^T y = \lambda_\theta\}$ is tangential to \mathcal{Y} .

Proof. Let $y \in \mathcal{Y}$ and let $v \in \mathbb{R}^n$ such that $R(v) = y$. Then, by the minimax characterization of eigenvalues (2.3), we have

$$\lambda_\theta \leq \frac{v^* (\theta_1 A_1 + \dots + \theta_Q A_Q) v}{v^* v} = \theta^T R(v) = \theta^T y,$$

which proves (3.7) since y was an arbitrary element of \mathcal{Y} . Furthermore, if we denote with v_θ an eigenvector corresponding to λ_θ , then

$$\lambda_\theta = \frac{v_\theta^* (\theta_1 A_1 + \dots + \theta_Q A_Q) v_\theta}{v_\theta^* v_\theta} = \theta^T R(v_\theta).$$

Since $R(v_\theta) \in \mathcal{Y}$, this completes the proof. \square

Using Lemma 3.1 $2Q$ times, once for each of the signed canonical basis vectors $\pm e_1, \dots, \pm e_Q$ as θ , yields $2Q$ linear constraints on \mathcal{Y} . We assemble these constraints into the constraint matrix C and the vector b . Clearly, this choice of C ensures (3.6) to be feasible, as each vector $\theta(\mu) \in \mathbb{R}^Q$ can always be represented as a non-negative linear combination of canonical basis vector $\pm e_1, \dots, \pm e_Q$. As discussed above, this is equivalent to (3.5) being bounded, which is exactly what we wanted to achieve.

Furthermore, it is clear that the constraints obtained using Lemma 3.1 are of the following type:

$$\lambda_{\min}(A_q) \leq y_q \leq \lambda_{\max}(A_q), \quad \forall q = 1, \dots, Q.$$

By putting them together, we obtain the bounding box \mathcal{B} for \mathcal{Y} :

$$\mathcal{B} := [\lambda_{\min}(A_1), \lambda_{\max}(A_1)] \times \dots \times [\lambda_{\min}(A_Q), \lambda_{\max}(A_Q)] \subseteq \mathbb{R}^Q. \quad (3.8)$$

Remark 3.2. Computing solution of $2Q$ original-sized eigenvalue problem in (3.8) is computationally not cheap. Moreover, in practice the bounding box constraints in (3.5) usually provide only a crude approximation to $\lambda_{\min}(A(\mu))$, since $\theta(\mu)$ is not necessarily close to any of the canonical basis vectors $\pm e_q$. Therefore, it is worth considering possible alternatives to using \mathcal{B} .

Suppose we are given a Q -dimensional hypercube \mathcal{D} such that $\{\theta(\mu) : \mu \in D\} \subset \mathcal{D}$. As before, using Lemma 3.1 2^Q times, once for each vertex of \mathcal{D} as θ , yields 2^Q linear constraints on \mathcal{Y} . By definition, these constraints span all vectors $\theta \in \mathcal{D}$. Assembling them into constraints matrices C

and b clearly leads to a feasible dual linear program (3.6), making this a possible alternative to using \mathcal{B} . In practice, given $\mu \in D$, we can expect this set of constraints to provide a significantly more accurate approximation to $\lambda_{\min}(A(\mu))$ than \mathcal{B} , as the vertices of \mathcal{D} are generally closer to the objective functions $\theta(\mu)$ than any of the canonical basis vectors.

In general, such a hypercube \mathcal{D} is usually not available, whereas constructing the alternative set of constraints requires solving 2^Q original-sized eigenvalue problems, which is usually computationally too expensive. However, when $A(\mu)$ is affine in μ , as in (2.28), \mathcal{D} is directly available as $\{1\} \times D$. Furthermore, if Q is also small, say $Q \leq 3$, solving 2^Q eigenvalue problems is not significantly more computationally expensive than solving $2Q$, making this approach a preferred alternative to using \mathcal{B} due to improved approximation quality.

3.1.3 SCM bounds for $\lambda_{\min}(A(\mu))$

Given the sample set \mathcal{S} containing M parameter values $\mathcal{S} = \{\mu_1, \dots, \mu_M\} \subset D$, let us suppose we have computed the corresponding eigenpairs $(\lambda_1, v_1), \dots, (\lambda_M, v_M)$, that is, λ_i is the smallest eigenvalue of $A(\mu_i)$ with eigenvector $v_i \in \mathbb{C}^n$. We now describe how SCM uses this information to approximate the set \mathcal{Y} defined above.

Clearly,

$$\mathcal{Y}_{\text{UB}}(\mathcal{S}) := \{R(v_i) : i = 1, \dots, M\} \quad (3.9)$$

is a subset of \mathcal{Y} . Optimizing (3.4) over $\mathcal{Y}_{\text{UB}}(\mathcal{S})$ instead of \mathcal{Y} thus yields an *upper bound* for $\lambda_{\min}(A(\mu))$. Note that this is equivalent to optimizing over the convex hull of $\mathcal{Y}_{\text{UB}}(\mathcal{S})$, since a solution of the LP can always be attained at a vertex of the convex polyhedron.

To get a lower bound, we use the bounding box \mathcal{B} defined in (3.8). However, as previously mentioned, \mathcal{B} alone is often a too crude approximation to \mathcal{Y} and we further refine it using the sampled eigenvalues. Each sampled eigenvalue λ_i contributes to one additional constraint, using Lemma 3.1, resulting in

$$\mathcal{Y}_{\text{LB}}(\mathcal{S}) := \{y \in \mathcal{B} : \theta(\mu_i)^T y \geq \lambda_i, i = 1, \dots, M\}.$$

The property $\mathcal{Y} \subset \mathcal{Y}_{\text{LB}}(\mathcal{S})$ follows from the minimax characterization of eigenvalues (2.3): every $y = R(u_y) \in \mathcal{Y}$ satisfies $\theta(\mu_i)^T y = u_y^* A(\mu_i) u_y / u_y^* u_y \geq \min_u u^* A(\mu_i) u / u^* u = \lambda_i$. As shown in Lemma 3.1, this implies that the convex polyhedron $\mathcal{Y}_{\text{LB}}(\mathcal{S})$ is tangential to \mathcal{Y} at $R(v_1), \dots, R(v_M)$.

With the sets defined above, we let

$$\lambda_{\text{UB}}(\mu; \mathcal{S}) := \min_{y \in \mathcal{Y}_{\text{UB}}(\mathcal{S})} \theta(\mu)^T y, \quad (3.10)$$

$$\lambda_{\text{LB}}(\mu; \mathcal{S}) := \min_{y \in \mathcal{Y}_{\text{LB}}(\mathcal{S})} \theta(\mu)^T y. \quad (3.11)$$

Since $\mathcal{Y}_{\text{UB}}(\mathcal{S}) \subseteq \mathcal{Y} \subseteq \mathcal{Y}_{\text{LB}}(\mathcal{S})$, it follows that

$$\lambda_{\text{LB}}(\mu; \mathcal{S}) \leq \lambda_{\min}(A(\mu)) \leq \lambda_{\text{UB}}(\mu; \mathcal{S})$$

for every $\mu \in D$. While the evaluation of $\lambda_{\text{UB}}(\mu; \mathcal{S})$ is trivial, the evaluation of $\lambda_{\text{LB}}(\mu; \mathcal{S})$ requires the solution of an LP; see Figure 3.1 for an illustration.

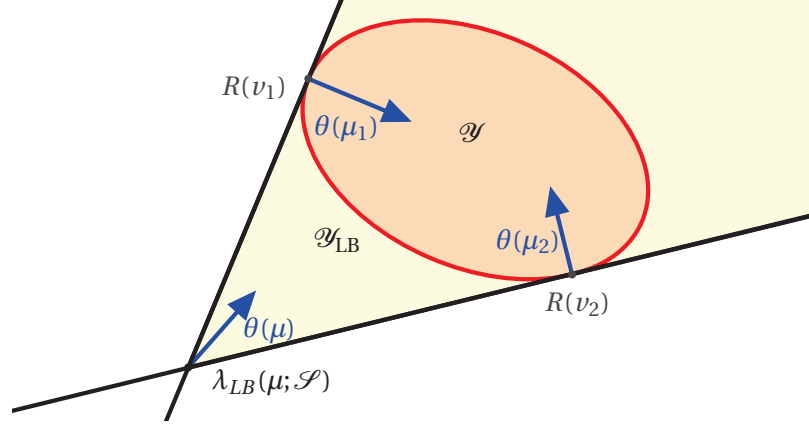


Figure 3.1: Illustration of the LP defining the lower bound $\lambda_{\text{LB}}(\mu; \mathcal{S})$ for $Q = 2$ and $M = 2$.

3.1.4 Error estimates and sampling strategy

Assessing the quality of the bounds (3.10)–(3.11) on the entire, usually continuous parameter domain D is, in general, an infeasible task. A common strategy in SCM, we substitute D by a training set $\Xi \subset D$ that contains finitely many (usually, a few thousand) parameter samples. We then measure the quality of the bounds by estimating the largest relative difference:

$$\max_{\mu \in \Xi} \frac{\lambda_{\text{UB}}(\mu; \mathcal{S}) - \lambda_{\text{LB}}(\mu; \mathcal{S})}{|\lambda_{\text{UB}}(\mu; \mathcal{S})|}. \quad (3.12)$$

If (3.12) is not sufficiently small, SCM enlarges \mathcal{S} by a parameter that attains the maximum in (3.12) and recomputes the bounds $\lambda_{\text{UB}}(\mu; \mathcal{S})$ and $\lambda_{\text{LB}}(\mu; \mathcal{S})$. The resulting greedy sampling strategy is summarized in Algorithm 2.

3.1.5 Computational complexity

Let us briefly summarize the computations performed by SCM. The bounding box \mathcal{B} for \mathcal{Y} needs to be determined initially by computing the smallest and the largest eigenvalues of A_1, \dots, A_Q . Since each iteration requires the computation of the smallest eigenpair (λ_i, v_i) of $A(\mu_i)$, this amounts to solving $2Q + M$ eigenproblems of size $n \times n$ in total. Verifying the accuracy of the current approximation on Ξ and selecting the next parameter sample requires computing $\lambda_{\text{UB}}(\mu; \mathcal{S})$ and $\lambda_{\text{LB}}(\mu; \mathcal{S})$ for all $\mu \in \Xi$. In total, this amounts to solving $M|\Xi|$ LP

Algorithm 2 Successive Constraint Method

Input: Training set Ξ , affine linear decomposition such that $A(\mu) = \theta_1(\mu)A_1 + \dots + \theta_Q(\mu)A_Q$ is Hermitian for every $\mu \in \Xi$. Relative error tolerance ε_{SCM} .

Output: Set $\mathcal{S} \subset \Xi$ with corresponding eigenpairs (λ_i, v_i) , such that

$$\frac{\lambda_{\text{UB}}(\mu; \mathcal{S}) - \lambda_{\text{LB}}(\mu; \mathcal{S})}{|\lambda_{\text{UB}}(\mu; \mathcal{S})|} < \varepsilon_{\text{SCM}} \text{ for every } \mu \in \Xi.$$

1: Compute $\lambda_{\min}(A_q), \lambda_{\max}(A_q)$ for $q = 1, \dots, Q$, defining \mathcal{B} according to (3.8).

2: $M = 0, \mathcal{S} = \emptyset$

3: **while** $\max_{\mu \in \Xi} \frac{\lambda_{\text{UB}}(\mu; \mathcal{S}) - \lambda_{\text{LB}}(\mu; \mathcal{S})}{|\lambda_{\text{UB}}(\mu; \mathcal{S})|} > \varepsilon_{\text{SCM}}$ **do**

4: $\mu_{M+1} \leftarrow \operatorname{argmax}_{\mu \in \Xi} \frac{\lambda_{\text{UB}}(\mu; \mathcal{S}) - \lambda_{\text{LB}}(\mu; \mathcal{S})}{|\lambda_{\text{UB}}(\mu; \mathcal{S})|}$

5: $\mathcal{S} \leftarrow \mathcal{S} \cup \mu_{M+1}$

6: Recompute $\lambda_{\text{UB}}(\mu; \mathcal{S})$ and $\lambda_{\text{LB}}(\mu; \mathcal{S})$. according to (3.10)–(3.11).

7: $M \leftarrow M + 1$

8: **end while**

problems with Q variables and at most $2Q + M$ constraints.

3.1.6 Interpolation results

In this section we study interpolation properties of the SCM bounds $\lambda_{\text{UB}}(\mu; \mathcal{S})$ and $\lambda_{\text{SLB}}(\mu; \mathcal{S})$, which can be used to provide *a priori* convergence estimates in the vicinity of the sampled points, similarly as in the case of eigenvalue optimization [KMMM15].

As also discussed in [HRSP07], it is immediate to see that the bounds produced by SCM coincide with $\lambda_{\min}(A(\mu))$ for all $\mu \in \mathcal{S}$. The following theorem shows that the upper bounds also interpolate the derivatives of $\lambda_{\min}(A(\mu))$ on \mathcal{S} .

Theorem 3.3. *Let $\mathcal{S} \subset D$ be finite and consider the upper bound $\lambda_{\text{UB}}(\mu; \mathcal{S})$ defined in (3.10). Given $\mu_i \in \mathcal{S}$ in the interior of D , assume that $\theta_1, \dots, \theta_Q : D \rightarrow \mathbb{R}$ are differentiable at μ_i and that $\lambda_i = \lambda_{\min}(A(\mu_i))$ is a simple eigenvalue of $A(\mu_i)$. Then*

$$\nabla \lambda_{\text{UB}}(\mu_i; \mathcal{S}) = \nabla \lambda_{\min}(A(\mu_i)),$$

with the gradient ∇ with respect to μ .

Proof. Let v_i be an eigenvector associated with λ_i such that $\|v_i\|_2 = 1$ and set $y_i := R(v_i) \in \mathcal{Y}_{\text{UB}}(\mathcal{S})$. By definition (3.10), the relation

$$\lambda_{\text{UB}}(\mu; \mathcal{S}) = \min_{y \in \mathcal{Y}_{\text{UB}}(\mathcal{S})} \theta(\mu)^T y = \theta(\mu)^T y_i \tag{3.13}$$

holds for $\mu = \mu_i$.

We will first prove that y_i is the unique minimizer. Let us suppose the contrary, there exist $j \in \{1, \dots, M\}$ such that $R(v_j) = y_j \neq y_i$ and $\theta(\mu_i)^T y_j = \theta(\mu_i)^T y_i = \lambda_i$, which also gives that

Chapter 3. Low-rank approach for parameter dependent Hermitian eigenvalue problem

$v_i \neq v_j$. However, by the minimax characterization of eigenvalues (2.3), this implies that both (λ_i, v_i) and (λ_i, v_j) are eigenpairs of $A(\mu_i)$ which contradicts the fact that λ_i is a simple eigenvalue. Thus, y_i is the unique element of $\mathcal{Y}_{\text{UB}}(\mathcal{S})$ such that equality holds in (3.13) for $\mu = \mu_i$.

Furthermore, this implies that the following inequalities

$$\theta(\mu)^T (y_i - y_j) < 0, \quad \forall y_j \in \mathcal{Y}_{\text{UB}}(\mathcal{S}) \text{ such that } y_j \neq y_i, \quad (3.14)$$

hold for $\mu = \mu_i$. Since $|\mathcal{Y}_{\text{UB}}(\mathcal{S})| < \infty$ and $\theta(\mu)$ is continuous at μ_i , there exists an open neighborhood $\Omega \subset D$ around μ_i on which (3.14) is fulfilled. Therefore, y_i is the unique minimizer of (3.10) for all $\mu \in \Omega$, and (3.13) holds on Ω . Consequently,

$$\frac{\partial}{\partial \mu^{(p)}} \lambda_{\text{UB}}(\mu_i; \mathcal{S}) = \frac{\partial}{\partial \mu^{(p)}} \theta(\mu_i)^T y_i,$$

where $\mu^{(p)}$ denotes the p -th entry of μ for $p = 1, \dots, d$.

On the other hand, the well-known expression for the derivative of a simple eigenvalue [Lan64] gives

$$\begin{aligned} \frac{\partial}{\partial \mu^{(p)}} \lambda_{\min}(A(\mu_i)) &= v_i^* \frac{\partial}{\partial \mu^{(p)}} A(\mu_i) v_i = v_i^* \left(\sum_{q=1}^Q \frac{\partial}{\partial \mu^{(p)}} \theta_q(\mu_i) A_q \right) v_i \\ &= \sum_{q=1}^Q \frac{\partial}{\partial \mu^{(p)}} \theta_q(\mu_i) v_i^* A_q v_i = \frac{\partial}{\partial \mu^{(p)}} \theta(\mu_i)^T y_i, \end{aligned}$$

which completes the proof. □

As the following example shows, the result of Theorem 3.3 does not extend to the lower bounds produced by SCM.

Example 3.4. For $\mu \in D := [0, \pi]$, let

$$A(\mu) = \cos(\mu) A_1 + \sin(\mu) A_2 = \cos(\mu) \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} + \sin(\mu) \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}.$$

It can be shown that \mathcal{Y} , the joint numerical range of A_1 and A_2 , equals the unit circle around 0. Consider the sample set $\mathcal{S} = \{\mu_1, \mu_2, \mu_3\} = \{0, \frac{\pi}{2}, \pi\}$, with

$$\lambda_{\min}(A(\mu_1)) = \lambda_{\min}(A(\mu_2)) = \lambda_{\min}(A(\mu_3)) = -1.$$

The resulting lower bound set $\mathcal{Y}_{\text{LB}}(\mathcal{S})$ is the half-infinite box shown in Figure 3.2. When minimizing $\theta(\mu)^T y$ for $y \in \mathcal{Y}_{\text{LB}}(\mathcal{S})$, the minimum is attained at the vertex $(-1, -1)$ for $\mu \in$

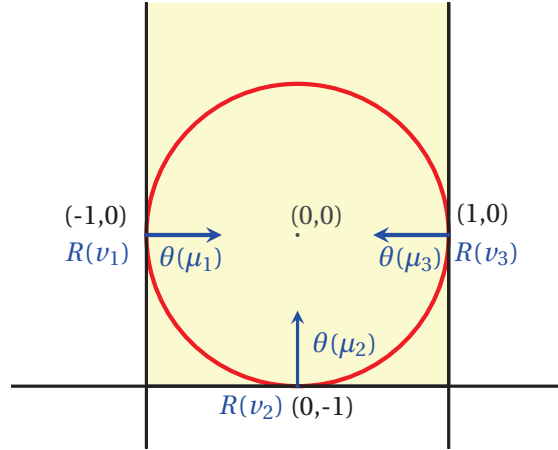


Figure 3.2: Joint numerical range \mathcal{Y} (red circle) and lower bound set $\mathcal{Y}_{\text{LB}}(\mathcal{S})$ (yellow area) for the setting described in Example 3.4.

$[\pi/4, \pi/2]$ and at the vertex $(1, -1)$ for $\mu \in [\pi/2, 3\pi/4]$. Hence,

$$\lambda_{\text{LB}}(\mu) = \begin{cases} -\cos \mu - \sin \mu, & \text{for } \mu \in [\pi/4, \pi/2], \\ \cos \mu - \sin \mu, & \text{for } \mu \in [\pi/2, 3\pi/4], \end{cases}$$

yielding the following one-sided derivatives at $\mu = \pi/2$:

$$\begin{aligned} \lambda'_{\text{LB}}((\pi/2)^-) &= \sin(\pi/2) - \cos(\pi/2) = 1, \\ \lambda'_{\text{LB}}((\pi/2)^+) &= -\sin(\pi/2) - \cos(\pi/2) = -1. \end{aligned}$$

In contrast, the exact eigenvalue is differentiable at $\pi/2$. Moreover, $\lambda'_{\min}(A(\pi/2)) = 0$ is different from both one-sided derivatives of $\lambda_{\text{LB}}(\mu)$.

As will be discussed in more detail in Section 3.2.5, interpolation properties offer an indication of the approximation quality in the vicinity of sampled points. Thus, considering Theorem 3.3 and Example 3.4, we can anticipate the lower bounds produced by SCM to be asymptotically less accurate than the upper bounds, which has been confirmed by numerical experiments in Section 3.5. This phenomenon has already been observed numerically in [HKC⁺10], implying a need to find more accurate lower bounds. However, the following theorem indicates that such an improvement is not possible without taking additional information on $A(\mu)$ into account.

Theorem 3.5. *Let $\mathcal{S} = \{\mu_1, \dots, \mu_M\} \subseteq D$ and consider the lower bounds $\lambda_{\text{LB}}(\mu; \mathcal{S})$ defined in (3.11) for a Hermitian matrix-valued function $A(\mu)$ in affine linear decomposition (3.2). Let $\tilde{\mu} \in D$. If $2Q + M < n$ then there exist matrices $\bar{A}_1, \dots, \bar{A}_Q \in \mathbb{C}^{n \times n}$, defining $\bar{A}(\mu) = \theta_1(\mu)\bar{A}_1 + \dots + \theta_Q(\mu)\bar{A}_Q$ and $\bar{\mathcal{B}}$ analogously as in (3.8), such that $\bar{\mathcal{B}} = \bar{\mathcal{B}}$ and*

$$\lambda_{\min}(\bar{A}(\tilde{\mu})) = \lambda_{\text{LB}}(\tilde{\mu}; \mathcal{S}), \quad \lambda_{\min}(\bar{A}(\mu_i)) = \lambda_{\min}(A(\mu_i)) \quad (3.15)$$

Chapter 3. Low-rank approach for parameter dependent Hermitian eigenvalue problem

hold for $i = 1, \dots, M$.

Proof. Let the columns of $V \in \mathbb{C}^{n \times (M+2Q)}$ and $V_\perp \in \mathbb{C}^{n \times (n-M-2Q)}$ form orthonormal bases of $\mathcal{V} = \text{span}\{v_1, \dots, v_M, w_1, \dots, w_Q, w^1, \dots, w^Q\}$ and \mathcal{V}^\perp , respectively, where each v_i denotes an eigenvector associated with $\lambda_{\min}(A(\mu_i))$, while each w_i and w^i denote eigenvectors associated with $\lambda_{\min}(A_i)$ and $\lambda_{\max}(A_i)$, respectively. Moreover, let $y_{\tilde{\mu}} \in \mathcal{Y}_{\text{LB}}(\mathcal{S}) \subset \mathbb{R}^Q$ denote a minimizer of (3.11) for $\tilde{\mu}$, that is, $\lambda_{\text{LB}}(\tilde{\mu}; \mathcal{S}) = \theta(\tilde{\mu})^T y_{\tilde{\mu}}$. The rest of the proof consists of showing that the matrices defined by

$$\bar{A}_q := VV^* A_q VV^* + y_{\tilde{\mu}, q} V_\perp V_\perp^*, \quad q \in \{1, \dots, Q\},$$

satisfy $\mathcal{B} = \overline{\mathcal{B}}$ and (3.15).

Given a vector $u \in \mathbb{C}^n$ of unit norm, we can write $u = u_\mathcal{V} + u_\perp$ with $u_\mathcal{V} \in \mathcal{V}$ and $u_\perp \in \mathcal{V}^\perp$. We have that $u^* \bar{A}_q u = u_\mathcal{V}^* \bar{A}_q u_\mathcal{V} + y_{\tilde{\mu}, q} \|u_\perp\|^2$, $\forall q = 1, \dots, Q$, where $y_{\tilde{\mu}, q} \in [\lambda_{\min}(A_q), \lambda_{\max}(A_q)]$, and $u_\mathcal{V}^* \bar{A}_q u_\mathcal{V} \in [\|u_\mathcal{V}\|^2 \lambda_{\min}(A_q), \|u_\mathcal{V}\|^2 \lambda_{\max}(A_q)]$. Consequently,

$$\lambda_{\min}(A_q) \leq u^* \bar{A}_q u \leq \lambda_{\max}(A_q).$$

Since equality is attained for $u = w_q$ and $u = w^q$, we obtain $\mathcal{B} = \overline{\mathcal{B}}$.

To show (3.15) we first note that

$$u^* \bar{A}(\mu) u = \sum_{q=1}^Q \theta_q(\mu) (u_\mathcal{V}^* A_q u_\mathcal{V} + y_{\tilde{\mu}, q} \|u_\perp\|^2) = u_\mathcal{V}^* A(\mu) u_\mathcal{V} + \theta(\mu)^T y_{\tilde{\mu}} \|u_\perp\|^2 \quad (3.16)$$

for any $\mu \in D$. For $\mu = \mu_i$, this yields

$$u^* \bar{A}(\mu_i) u \geq \lambda_{\min}(A(\mu_i)) \|u_\mathcal{V}\|_2^2 + \lambda_{\min}(A(\mu_i)) \|u_\perp\|_2^2 = \lambda_{\min}(A(\mu_i)) \|u\|_2^2,$$

where we have used that $y_{\tilde{\mu}} \in \mathcal{Y}_{\text{LB}}(\mathcal{S})$ implies $\theta(\mu_i)^T y_{\tilde{\mu}} \geq \lambda_{\min}(A(\mu_i))$. Since equality is attained for $u = v_i$, this establishes the second equality in (3.15).

Concerning the first equality in (3.15), we first note that the definition of $\lambda_{\text{LB}}(\tilde{\mu}; \mathcal{S})$ implies

$$u_\mathcal{V}^* A(\tilde{\mu}) u_\mathcal{V} = \theta(\tilde{\mu})^T R(u_\mathcal{V}) \|u_\mathcal{V}\|_2^2 \geq \lambda_{\text{LB}}(\tilde{\mu}; \mathcal{S}) \|u_\mathcal{V}\|_2^2.$$

Inserted into (3.16) for $\mu = \tilde{\mu}$, this yields

$$u^* \bar{A}(\tilde{\mu}) u \geq \lambda_{\text{LB}}(\tilde{\mu}; \mathcal{S}) \|u_\mathcal{V}\|_2^2 + \lambda_{\text{LB}}(\tilde{\mu}; \mathcal{S}) \|u_\perp\|_2^2 = \lambda_{\text{LB}}(\tilde{\mu}; \mathcal{S}) \|u\|_2^2.$$

Since equality is attained by any $u \in \mathcal{V}^\perp$, this shows the first equality in (3.15) and thus completes the proof. \square

By definition the lower bounds in (3.11) depend only on $\theta(\mu)$, the bounding box \mathcal{B} and the eigenvalues at μ_i . Hence, the lower bounds for $\bar{A}(\mu)$, the Hermitian matrix-valued function

constructed in Theorem 3.5, are identical with those for $A(\mu)$. For $\bar{A}(\mu)$, the lower bound $\lambda_{\text{LB}}(\tilde{\mu}; \mathcal{S})$ coincides with the exact eigenvalue at an arbitrary fixed $\tilde{\mu} \in D$. Hence, additional knowledge, beyond the eigenvalues at μ_i , needs to be incorporated to improve the lower bounds.

3.2 Subspace acceleration

In this section, our new subspace approach is presented that takes eigenvector information across different parameter samples into account and offers the flexibility to incorporate eigenvectors for larger eigenvalues as well. The basic idea of this approach is to construct a low-dimensional subspace $\mathcal{V} \subset \mathbb{R}^n$ that approximates $\{v_{\min}(\mu) : \mu \in D\}$ well, where $v_{\min}(\mu)$ denotes the smallest eigenvector of $A(\mu)$.

Given $\mathcal{S} = \{\mu_1, \dots, \mu_M\} \subset D$, suppose that for each sample μ_i we have computed the $\ell \geq 1$ smallest eigenvalues

$$\lambda_i = \lambda_i^{(1)} \leq \lambda_i^{(2)} \leq \dots \leq \lambda_i^{(\ell)}$$

of $A(\mu_i)$ along with an orthonormal basis of associated eigenvectors $v_i^{(1)}, v_i^{(2)}, \dots, v_i^{(\ell)} \in \mathbb{C}^n$. To simplify notation, we assume ℓ is constant for each μ_1, \dots, μ_M , although this is not necessary. The eigenvectors will be collected in the subspace

$$\mathcal{V}(\mathcal{S}, \ell) := \text{span}\{v_1^{(1)}, \dots, v_1^{(\ell)}, v_2^{(1)}, \dots, v_2^{(\ell)}, \dots, v_M^{(1)}, \dots, v_M^{(\ell)}\}. \quad (3.17)$$

In the subsequent two sections, we discuss how the information in $\mathcal{V}(\mathcal{S}, \ell)$ can be used to compute tighter bounds for $\lambda_{\min}(A(\mu))$.

3.2.1 Subspace approach for upper bounds

Given the subspace $\mathcal{V}(\mathcal{S}, \ell)$ from (3.17), we define an upper bound set analogously to (3.9):

$$\mathcal{Y}_{\text{SUB}}(\mathcal{S}, \ell) := \{R(v) : v \in \mathcal{V}(\mathcal{S}, \ell)\}.$$

The corresponding upper bound for $\mu \in D$ is defined as

$$\lambda_{\text{SUB}}(\mu; \mathcal{S}, \ell) := \min_{y \in \mathcal{Y}_{\text{SUB}}(\mathcal{S}, \ell)} \theta(\mu)^T y.$$

Clearly, we have $\mathcal{Y}_{\text{UB}}(\mathcal{S}) \subseteq \mathcal{Y}_{\text{SUB}}(\mathcal{S}, \ell) \subseteq \mathcal{Y}$ and thus

$$\lambda_{\text{UB}}(\mu; \mathcal{S}) \geq \lambda_{\text{SUB}}(\mu; \mathcal{S}, \ell) \geq \lambda_{\min}(A(\mu)).$$

To evaluate $\lambda_{\text{SUB}}(\mu; \mathcal{S}, \ell)$, we first compute an orthonormal basis $V \in \mathbb{C}^{n \times M\ell}$ of $\mathcal{V}(\mathcal{S}, \ell)$ and

obtain

$$\begin{aligned}
 \lambda_{\text{SUB}}(\mu; \mathcal{S}, \ell) &= \min_{v \in \mathcal{V}(\mathcal{S}, \ell)} \theta(\mu)^T R(v) = \min_{\substack{w \in \mathbb{C}^{M\ell} \\ \|w\|_2=1}} \theta(\mu)^T R(Vw) \\
 &= \min_{\substack{w \in \mathbb{C}^{M\ell} \\ \|w\|_2=1}} \theta_1(\mu) w^* V^* A_1 V w + \dots + \theta_Q(\mu) w^* V^* A_Q V w \\
 &= \lambda_{\min}(\theta_1(\mu) V^* A_1 V + \dots + \theta_Q(\mu) V^* A_Q V) = \lambda_{\min}(V^* A(\mu) V). \quad (3.18)
 \end{aligned}$$

Thus, the computation of $\lambda_{\text{SUB}}(\mu; \mathcal{S}, \ell)$ requires the solution of an eigenvalue problem of size $M\ell \times M\ell$, with $M\ell$ usually much smaller than n .

3.2.2 Subspace approach for lower bounds

We will use a perturbation result to turn the upper bound (3.18) into a lower bound $\lambda_{\text{SLB}}(\mu; \mathcal{S}, \ell)$ for $\mu \in D$. For this purpose, we consider for some small integer $r \leq M\ell$ the r smallest eigenvalues

$$\lambda_{\text{SUB}}(\mu; \mathcal{S}, \ell) = \lambda_{\mathcal{V}}^{(1)} \leq \lambda_{\mathcal{V}}^{(2)} \leq \dots \leq \lambda_{\mathcal{V}}^{(r)}$$

of $V^* A(\mu) V$, along with the corresponding eigenvectors $w_1, \dots, w_r \in \mathbb{C}^{M\ell}$. Let

$$\mathcal{W}(\mu) = \text{span}\{w_1, \dots, w_r\}$$

and let $U \in \mathbb{C}^{n \times r}$ be an orthonormal basis of the subspace $\mathcal{U}(\mu)$ spanned by the Ritz vectors:

$$\mathcal{U}(\mu) := V\mathcal{W}(\mu) = \text{span}\{Vw_1, \dots, Vw_r\}.$$

Moreover, let $U_{\perp} \in \mathbb{C}^{n \times (n-r)}$ be an orthonormal basis of $\mathcal{U}^{\perp}(\mu)$ and denote the eigenvalues of $U_{\perp}^* A(\mu) U_{\perp}$ by

$$\lambda_{\mathcal{U}^{\perp}}^{(1)} \leq \lambda_{\mathcal{U}^{\perp}}^{(2)} \leq \dots \leq \lambda_{\mathcal{U}^{\perp}}^{(n-r)}.$$

The transformed matrix

$$[U, U_{\perp}]^* A(\mu) [U, U_{\perp}] = \begin{bmatrix} U^* A(\mu) U & U^* A(\mu) U_{\perp} \\ U_{\perp}^* A(\mu) U & U_{\perp}^* A(\mu) U_{\perp} \end{bmatrix}$$

clearly has the same eigenvalues as $A(\mu)$, while the perturbed matrix

$$\begin{bmatrix} U^* A(\mu) U & 0 \\ 0 & U_{\perp}^* A(\mu) U_{\perp} \end{bmatrix}$$

has the eigenvalues $\{\lambda_{\mathcal{V}}^{(1)}, \dots, \lambda_{\mathcal{V}}^{(r)}\} \cup \{\lambda_{\mathcal{U}^\perp}^{(1)}, \dots, \lambda_{\mathcal{U}^\perp}^{(n-r)}\}$. Applying Theorem 2.11 to this situation yields the error bound

$$|\lambda_{\min}(A(\mu)) - \min(\lambda_{\mathcal{V}}^{(1)}, \lambda_{\mathcal{U}^\perp}^{(1)})| \leq \frac{2\rho^2}{\delta + \sqrt{\delta^2 + 4\rho^2}},$$

with the residual norm

$$\rho := \|U_\perp^* A(\mu) U\|_2 = \|A(\mu) U - U(U^* A(\mu) U)\|_2$$

and the absolute gap $\delta := |\lambda_{\mathcal{V}}^{(1)} - \lambda_{\mathcal{U}^\perp}^{(1)}|$. Rearranging terms thus gives the lower bound

$$f(\lambda_{\mathcal{U}^\perp}^{(1)}) \leq \lambda_{\min}(A(\mu)), \quad \text{with} \quad f(\eta) := \min(\lambda_{\mathcal{V}}^{(1)}, \eta) - \frac{2\rho^2}{|\lambda_{\mathcal{V}}^{(1)} - \eta| + \sqrt{|\lambda_{\mathcal{V}}^{(1)} - \eta|^2 + 4\rho^2}}. \quad (3.19)$$

This lower bound is not practical so far, as it involves the quantity $\lambda_{\mathcal{U}^\perp}^{(1)}$, which would require the solution of a large eigenvalue problem of size $(n-r) \times (n-r)$.

Lemma 3.6. *The function $f : \mathbb{R} \rightarrow \mathbb{R}$ defined in (3.19) is continuous and monotonically increasing.*

Proof. As a composition of continuous functions, the function f is clearly continuous. To prove monotonicity we distinguish two cases. First, let $\eta \geq \lambda_{\mathcal{V}}^{(1)}$. Then

$$f(\eta) = \lambda_{\mathcal{V}}^{(1)} - 2\rho^2 / \left(\eta - \lambda_{\mathcal{V}}^{(1)} + \sqrt{(\eta - \lambda_{\mathcal{V}}^{(1)})^2 + 4\rho^2} \right),$$

which clearly increases as η increases. Now, let $\eta \leq \lambda_{\mathcal{V}}^{(1)}$. Then

$$f(\eta) = \eta - 2\rho^2 / \left(\lambda_{\mathcal{V}}^{(1)} - \eta + \sqrt{(\eta - \lambda_{\mathcal{V}}^{(1)})^2 + 4\rho^2} \right)$$

and

$$f'(\eta) = 1 - \frac{2\rho^2}{\left(\lambda_{\mathcal{V}}^{(1)} - \eta + \sqrt{(\lambda_{\mathcal{V}}^{(1)} - \eta)^2 + 4\rho^2} \right) \sqrt{(\lambda_{\mathcal{V}}^{(1)} - \eta)^2 + 4\rho^2}}.$$

Showing $f'(\eta) \geq 0$, and thus establishing monotonicity, is equivalent to

$$\begin{aligned} (\lambda_{\mathcal{V}}^{(1)} - \eta)^2 + 4\rho^2 + (\lambda_{\mathcal{V}}^{(1)} - \eta) \sqrt{(\lambda_{\mathcal{V}}^{(1)} - \eta)^2 + 4\rho^2} &\geq 2\rho^2 \\ (\lambda_{\mathcal{V}}^{(1)} - \eta) \sqrt{(\lambda_{\mathcal{V}}^{(1)} - \eta)^2 + 4\rho^2} &\geq 0 \geq -(\lambda_{\mathcal{V}}^{(1)} - \eta)^2 - 2\rho^2, \end{aligned}$$

which is trivially satisfied for $\lambda_{\mathcal{V}}^{(1)} \geq \eta$. This completes the proof. \square

Lemma 3.6 implies that $f(\eta)$ remains a lower bound as long as $\eta \leq \lambda_{\mathcal{U}^\perp}^{(1)}$. To summarize, our

Chapter 3. Low-rank approach for parameter dependent Hermitian eigenvalue problem

subspace-accelerated lower bound is defined as

$$\lambda_{\text{SLB}}(\mu; \mathcal{S}, \ell) := \min(\lambda_{\mathcal{V}}^{(1)}, \eta(\mu)) - \frac{2\rho^2}{|\lambda_{\mathcal{V}}^{(1)} - \eta(\mu)| + \sqrt{|\lambda_{\mathcal{V}}^{(1)} - \eta(\mu)|^2 + 4\rho^2}} \quad (3.20)$$

for a lower bound $\eta(\mu)$ of $\lambda_{\mathcal{U}^\perp}^{(1)}$.

Determining a lower bound for $\lambda_{\mathcal{U}^\perp}^{(1)}$

The lower bound for $\lambda_{\mathcal{U}^\perp}^{(1)} = \lambda_{\min}(U_\perp^* A(\mu) U_\perp)$ needed in (3.20) will be determined by adapting the ideas from Section 3.1.3. Let us recall that SCM determines a lower bound for $\lambda_{\min}(A(\mu))$ by solving the LP

$$\lambda_{\text{LB}}(\mu; \mathcal{S}) = \min_{y \in \mathcal{Y}_{\text{LB}}(\mathcal{S})} \theta(\mu)^T y, \quad (3.21)$$

with $\mathcal{Y}_{\text{LB}}(\mathcal{S}) := \{y \in \mathcal{B} : \theta(\mu_i)^T y \geq \lambda_i, i = 1, \dots, M\}$ and the bounding box \mathcal{B} defined in (3.8). To simplify the discussion, we always assume in the following that $\mathcal{Y}_{\text{LB}}(\mathcal{S})$ is a simple polytope with no degenerate facets. Then there exists an optimizer $y_\mu \in \mathbb{R}^Q$ of (3.21) such that there are Q , among $2Q + M$, linearly independent active constraints [MG07]. In other words, y_μ satisfies a linear system

$$\Theta y_\mu = \psi, \quad (3.22)$$

where $\Theta \in \mathbb{R}^{Q \times Q}$ is invertible and each equation corresponds either to a constraint of the form $\theta(\mu_i)^T y_\mu = \lambda_i$ or to a box constraint. In the following, we assume that at least one of the active constraints is a non-box constraint.

Establishing a lower bound for $\lambda_{\mathcal{U}^\perp}^{(1)}$ is equivalent to determining $\eta(\mu)$ such that $\eta(\mu) \leq u_\perp^* A(\mu) u_\perp$ holds for every $u_\perp \in \mathcal{U}^\perp(\mu)$ with $\|u_\perp\|_2 = 1$. The restriction of u_\perp to a lower-dimensional subspace can be used to tighten the non-box constraints in (3.21) and the following lemmas explain how to achieve that.

Lemma 3.7. *Let $D \in \mathbb{C}^{s \times s}$, $W \in \mathbb{C}^{t \times s}$, such that $D = \text{diag}(d_{ii})$ with $d_{ii} < 0, \forall i = 1, \dots, s$ and W is of full-rank. If $s \leq t$, then*

$$\lambda_{\min}(WDW^*) = \lambda_{\min}(W^*WD).$$

Proof. Let $W_\perp \in \mathbb{C}^{t \times t-s}$ denote an orthonormal basis of $\text{range}(W)^\perp$, that is $W^*W_\perp = 0$ and $[WW_\perp]$ is an invertible matrix. The matrix congruence between the matrices $\begin{bmatrix} D & 0 \\ 0 & 0 \end{bmatrix}$ and

$WDW^* = \begin{bmatrix} W & W_\perp \end{bmatrix} \begin{bmatrix} D & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} W & W_\perp \end{bmatrix}^*$ implies that $\lambda_{\min}(WDW^*) < 0$, as well as:

$$\begin{aligned} \lambda_{\min}(WDW^*) &= \lambda_{\min}\left(\begin{bmatrix} W & W_\perp \end{bmatrix} \begin{bmatrix} D & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} W & W_\perp \end{bmatrix}^*\right) \\ &= \lambda_{\min}\left(\begin{bmatrix} W & W_\perp \end{bmatrix}^* \begin{bmatrix} W & W_\perp \end{bmatrix} \begin{bmatrix} D & 0 \\ 0 & 0 \end{bmatrix}\right) = \lambda_{\min}\left(\begin{bmatrix} W^*WD & 0 \\ 0 & 0 \end{bmatrix}\right), \end{aligned}$$

where in the second equality we have used the fact that $\lambda(AB) = \lambda(BA)$ for square matrices A and B . The assertion of the lemma now follows from $\lambda\left(\begin{bmatrix} W^*WD & 0 \\ 0 & 0 \end{bmatrix}\right) = \lambda(W^*WD) \cup \{0\}$ and the fact that $\lambda_{\min}(WDW^*) < 0$. \square

Lemma 3.8. Let $\Lambda_i = \text{diag}(\lambda_i^{(1)}, \dots, \lambda_i^{(\ell)})$ and $V_i = [v_i^{(1)}, \dots, v_i^{(\ell)}]$, with the notation as introduced above. If $n - r \geq r$ then

$$u_\perp^* A(\mu_i) u_\perp \geq \lambda_i + \beta_i(\mu),$$

where $\beta_i(\mu)$ is the smallest eigenvalue of the matrix

$$(\Lambda_i - \lambda_i I_\ell) - V_i^* U U^* V_i (\Lambda_i - \lambda_i^{(\ell+1)} I_\ell).$$

Proof. Using the spectral decomposition of $A(\mu_i)$, the result follows from

$$\begin{aligned} \min_{\substack{u_\perp \in \mathcal{R}^\perp(\mu) \\ \|u_\perp\|_2=1}} u_\perp^* A(\mu_i) u_\perp &\geq \min_{\substack{u_\perp \in \mathcal{R}^\perp(\mu) \\ \|u_\perp\|_2=1}} u_\perp^* V_i \Lambda_i V_i^* u_\perp + \lambda_i^{(\ell+1)} u_\perp^* (I - V_i V_i^*) u_\perp \\ &= \lambda_i^{(\ell+1)} + \min_{\substack{u_\perp \in \mathcal{R}^\perp(\mu) \\ \|u_\perp\|_2=1}} u_\perp^* V_i (\Lambda_i - \lambda_i^{(\ell+1)} I_\ell) V_i^* u_\perp \\ &= \lambda_i^{(\ell+1)} + \lambda_{\min}(U_\perp^* V_i (\Lambda_i - \lambda_i^{(\ell+1)} I_\ell) V_i^* U_\perp) \\ &= \lambda_i^{(\ell+1)} + \lambda_{\min}(V_i^* U_\perp U_\perp^* V_i (\Lambda_i - \lambda_i^{(\ell+1)} I_\ell)) \\ &= \lambda_i^{(\ell+1)} + \lambda_{\min}((I_\ell - V_i^* U U^* V_i) (\Lambda_i - \lambda_i^{(\ell+1)} I_\ell)) \\ &= \lambda_i + \lambda_{\min}((\Lambda_i - \lambda_i I_\ell) - V_i^* U U^* V_i (\Lambda_i - \lambda_i^{(\ell+1)} I_\ell)), \end{aligned} \tag{3.23}$$

where we used in the third equality that the negative eigenvalues of the matrix product $U_\perp^* V_i (\Lambda_i - \lambda_i^{(\ell+1)} I_\ell) V_i^* U_\perp$ do not change under a cyclic permutation of its factors, as proven in Lemma 3.7. \square

Using the values of $\beta_i(\mu)$ defined in Lemma 3.8, we update the right-hand side $\psi \in \mathbb{R}^Q$ in (3.22) as follows: If the k th equation corresponds to a non-box constraint $\theta(\mu_i)^T y = \lambda_i$, we set $\tilde{\psi}_k := \psi_k + \beta_i(\mu) = \lambda_i + \beta_i(\mu)$ and, otherwise, $\tilde{\psi}_k := \psi_k$. Since Θ is invertible, the solution of

the resulting LP

$$\inf_y \theta(\mu)^T y \quad \text{subject to} \quad \Theta y \geq \tilde{\psi}$$

is trivially given by

$$\check{y}_\mu := \Theta^{-1} \tilde{\psi}. \quad (3.24)$$

This finally yields the desired lower bound

$$\eta(\mu) := \theta(\mu)^T \check{y}_\mu \leq \lambda_{\mathcal{U}_\perp}^{(1)} = \lambda_{\min}(U_\perp^* A(\mu) U_\perp).$$

Remark 3.9. *The choice of r , the dimension of the Ritz subspace $\mathcal{U}(\mu)$, requires some consideration. For $r = 0$, $\mathcal{U}_\perp(\mu) = \mathbb{R}^n$ yields no improvement: $\lambda_{\text{SLB}}(\mu; \mathcal{S}, \ell) = \lambda_{\text{LB}}(\mu; \mathcal{S}, \ell)$. Intuitively, choosing $r = 1$ will be most effective when the second smallest eigenvalue of $A(\mu)$ is well separated from the smallest eigenvalue. Otherwise, one may benefit from choosing slightly larger values of r . In practice, we choose r adaptively by taking the maximal value of $\lambda_{\text{SLB}}(\mu; \mathcal{S}, \ell)$ over a few small values of $r = 0, 1, 2, \dots$, which also ensures that $\lambda_{\text{LB}}(\mu; \mathcal{S}, \ell) \leq \lambda_{\text{SLB}}(\mu; \mathcal{S}, \ell)$.*

3.2.3 Error estimates and sampling strategy

Similarly as in SCM, we substitute D by a finite training set $\Xi \subset D$. We use error estimates similar to (3.12), with the SCM bounds replaced by the subspace bounds:

$$\Delta(\mu; \mathcal{S}, \ell) = \frac{\lambda_{\text{SUB}}(\mu; \mathcal{S}, \ell) - \lambda_{\text{SLB}}(\mu; \mathcal{S}, \ell)}{|\lambda_{\text{SUB}}(\mu; \mathcal{S}, \ell)|}. \quad (3.25)$$

We use the same greedy sampling strategy as in SCM. As long as the error estimates (3.25) are not sufficiently small on whole Ξ , we enrich \mathcal{S} by the parameter value that attains the maximum in (3.25).

3.2.4 Interpolation properties

In this section, we study interpolation properties of the proposed subspace bounds $\lambda_{\text{SUB}}(\mu; \mathcal{S}, \ell)$ and $\lambda_{\text{SLB}}(\mu; \mathcal{S}, \ell)$. As will be discussed in the following Section 3.2.5, the significance of these properties is in their use for deriving *a priori* convergence estimates in the vicinity of sample points, which offer an indication of the approximation quality.

By definition, we already know that the bounds from our subspace approach are never worse than the bounds produced by SCM:

$$\lambda_{\text{LB}}(\mu; \mathcal{S}) \leq \lambda_{\text{SLB}}(\mu; \mathcal{S}, \ell) \leq \lambda_{\min}(A(\mu)) \leq \lambda_{\text{SUB}}(\mu; \mathcal{S}, \ell) \leq \lambda_{\text{UB}}(\mu; \mathcal{S}), \quad (3.26)$$

with equality at $\mu = \mu_i \in \mathcal{S}$. Together with Theorem 3.3, these inequalities imply that our

upper bounds also interpolate the derivatives at μ_i . The result was independently obtained by Kangal et al. in [KMMM15, Lemma 2.5].

Corollary 3.10. *For any $\ell \geq 1$ and any $\mu_i \in \mathcal{S}$ that satisfies the assumptions of Theorem 3.3, it holds that*

$$\nabla \lambda_{\text{SUB}}(\mu_i; \mathcal{S}, \ell) = \nabla \lambda_{\min}(A(\mu_i)),$$

with $\lambda_{\text{SUB}}(\mu_i; \mathcal{S}, \ell)$ defined as in (3.18).

Proof. By the assumptions, μ_i is an interior point of D and (3.26) implies that the inequality $\lambda_{\min}(A(\mu)) \leq \lambda_{\text{SUB}}(\mu; \mathcal{S}, \ell) \leq \lambda_{\text{UB}}(\mu; \mathcal{S})$ holds for all μ in a neighbourhood of μ_i . Combined with $\lambda_{\min}(A(\mu_i)) = \lambda_{\text{SUB}}(\mu_i; \mathcal{S}, \ell) = \lambda_{\text{UB}}(\mu_i; \mathcal{S})$, we get

$$\frac{\partial}{\partial \mu^{(q)}} \lambda_{\min}(A(\mu_i)) \leq \frac{\partial}{\partial \mu^{(q)}} \lambda_{\text{SUB}}(\mu_i; \mathcal{S}, \ell) \leq \frac{\partial}{\partial \mu^{(q)}} \lambda_{\text{UB}}(\mu_i; \mathcal{S}),$$

for $q = 1, \dots, d$. The result $\nabla \lambda_{\min}(A(\mu_i)) = \nabla \lambda_{\text{UB}}(\mu_i; \mathcal{S})$ of Theorem 3.3 now yields the desired result. \square

In contrast to SCM, it turns out that the subspace lower bounds also interpolate the derivative of $\lambda_{\min}(A(\mu))$ at $\mu \in \mathcal{S}$. To show this, we need the following lemma.

Lemma 3.11. *Let $\mu_i \in \mathcal{S}$ satisfy the assumptions of Theorem 3.3. For any $\varepsilon > 0$, there is a neighbourhood $\Omega \subseteq D$ around μ_i such that*

$$|\lambda_i - \lambda_{\mathcal{V}}^{(1)}(\mu)| \leq \varepsilon, \tag{3.27}$$

$$\lambda_i^{(2)} - \eta(\mu) \leq \varepsilon, \tag{3.28}$$

hold for all $\mu \in \Omega$, where $\lambda_{\mathcal{V}}^{(1)}(\cdot)$ and $\eta(\cdot)$ are defined as in Section 3.2.2.

Proof. Since $v_i^{(1)} \in \mathcal{V}(\mathcal{S}, \ell)$, we have $\lambda_{\mathcal{V}}^{(1)}(\mu_i) = \lambda_i$ and thus the continuity of the smallest eigenvalue implies that (3.27) holds for all μ in some neighbourhood Ω_1 around μ_i . It remains to prove (3.28).

In the LP (3.11) for determining $\lambda_{\text{LB}}(\mu_i; \mathcal{S})$, which is trivially given by λ_i , the constraint $\theta(\mu_i)^T y = \lambda_i$ is active. Since we assumed that $\mathcal{A}_{\text{LB}}(\mathcal{S})$ is a simple polytope with no degenerate facets, the continuity of $\theta(\mu)$ implies that this constraint remains active in a neighbourhood Ω_2 : $\theta(\mu_i)^T y_\mu = \lambda_i$ for all $\mu \in \Omega_2$, where y_μ is a minimizer of (3.11) for determining $\lambda_{\text{LB}}(\mu; \mathcal{S})$.

Chapter 3. Low-rank approach for parameter dependent Hermitian eigenvalue problem

By (3.23), the value of $\beta_i(\mu)$ defined in Lemma 3.8 satisfies

$$\begin{aligned}\beta_i(\mu) &= \min_{\substack{u_\perp \in \mathcal{U}^\perp(\mu) \\ \|u_\perp\|_2=1}} u_\perp^* V_i \Lambda_i V_i^* u_\perp + \lambda_i^{(\ell+1)} u_\perp^* (I - V_i V_i^*) u_\perp - \lambda_i \\ &\geq \min_{\substack{u_\perp \in \text{span}\{u(\mu)\}^\perp \\ \|u_\perp\|_2=1}} u_\perp^* V_i \Lambda_i V_i^* u_\perp + \lambda_i^{(\ell+1)} u_\perp^* (I - V_i V_i^*) u_\perp - \lambda_i,\end{aligned}$$

where $u(\mu) \in \mathcal{U}(\mu)$ denotes the Ritz vector corresponding to the smallest Ritz value $\lambda_V^{(1)}(\mu)$. Let us now consider the eigenvector $v_i^{(1)}$ belonging to the eigenvalue $\lambda_i = \lambda_{\min}(A(\mu_i))$. By definition, $v_i^{(1)}$ is contained in $\mathcal{U}(\mu_i)$. The simplicity of λ_i implies that the angle between $v_i^{(1)}$ and $u(\mu)$ becomes arbitrarily small as μ approaches μ_i . Therefore, for any $\varepsilon > 0$, there is a neighbourhood Ω_3 of μ_i such that

$$\beta_i(\mu) \geq \min_{\substack{u_\perp \perp v_i^{(1)} \\ \|u_\perp\|_2=1}} u_\perp^* V_i \Lambda_i V_i^* u_\perp + \lambda_i^{(\ell+1)} u_\perp^* (I - V_i V_i^*) u_\perp - \lambda_i - \frac{\varepsilon}{2} = \lambda_i^{(2)} - \lambda_i - \frac{\varepsilon}{2}, \forall \mu \in \Omega_3.$$

In summary, the vector \check{y}_μ defined in (3.24) satisfies

$$\theta(\mu_i)^T \check{y}_\mu = \lambda_i + \beta_i(\mu) \geq \lambda_i + \lambda_i^{(2)} - \lambda_i - \frac{\varepsilon}{2} = \lambda_i^{(2)} - \frac{\varepsilon}{2}. \quad (3.29)$$

By the invertibility of Θ , the vector \check{y}_μ remains bounded in the vicinity of μ_i . Together with the continuity of $\theta(\mu)$, this implies that there is a neighbourhood Ω_4 of μ_i such that

$$|(\theta(\mu) - \theta(\mu_i))^T \check{y}_\mu| \leq \frac{\varepsilon}{2}, \quad \forall \mu \in \Omega_4.$$

Combined with (3.29), this yields

$$\eta(\mu) = \theta(\mu)^T \check{y}_\mu \geq \lambda_i^{(2)} - \varepsilon,$$

which establishes (3.28). Setting $\Omega = \Omega_1 \cap \Omega_2 \cap \Omega_3 \cap \Omega_4$ completes the proof. \square

The following theorem establishes the Hermite interpolation property of the subspace lower bounds.

Theorem 3.12. *Let $\mu_i \in \mathcal{S}$ satisfy the assumptions of Theorem 3.3 and, additionally, suppose that $r \leq \ell$ and $\lambda_i^{(r+1)} > \lambda_i^{(r)}$. Then*

$$\nabla \lambda_{\text{SLB}}(\mu_i; \mathcal{S}, \ell) = \nabla \lambda_{\min}(A(\mu_i)).$$

Proof. By Lemma 3.11 and the simplicity of $\lambda_{\min}(A(\mu_i))$, there is $\delta_0 > 0$ such that $\eta(\mu) \geq \lambda_V^{(1)}(\mu) + \delta_0$ for μ sufficiently close to μ_i . Hence, the subspace lower bound (3.20) is given by

$$\lambda_{\text{SLB}}(\mu; \mathcal{S}, \ell) = \lambda_V^{(1)}(\mu) - \frac{2\rho^2}{\delta + \sqrt{\delta^2 + 4\rho^2}}, \quad (3.30)$$

with $\rho = \|U_{\perp}^* A(\mu) U\|_2$ and $\delta = |\lambda_{\mathcal{V}}^{(1)}(\mu) - \eta(\mu)| \geq \delta_0$. Since $\lambda_{\text{SUB}}(\mu_i; \mathcal{S}, \ell) \equiv \lambda_{\mathcal{V}}^{(1)}(\mu_i)$, by Corollary 3.10, we have

$$\nabla \lambda_{\mathcal{V}}^{(1)}(\mu_i) = \nabla \lambda_{\text{SUB}}(\mu_i; \mathcal{S}, \ell) = \nabla \lambda_{\min}(A(\mu_i)).$$

Since δ is bounded from below, the result follows from (3.30) if the gradient of ρ^2 at $\mu = \mu_i$ is zero. To show the latter, we first observe that the assumptions $\lambda_i^{(r+1)} > \lambda_i^{(r)}$ and $r \leq \ell$ imply that the invariant subspace belonging to the r smallest eigenvalues of $A(\mu_i)$ is simple and contained in V . Let us recall that $\mathcal{W}(\mu)$ is the invariant subspace belonging to the r smallest eigenvalue of $V^* A(\mu) V$. By minimax characterization of eigenvalues 2.4, the gap between $\lambda_i^{(r)}$ and $\lambda_i^{(r+1)}$ implies an equal or larger gap between $r+1$ -th and r -th eigenvalue of $V^* A(\mu_i) V$. This, together with the Lipschitz continuity of $V^* A(\mu) V$ w.r.t. μ , implies [DK70, Theorem 8.2]

$$\sin \Theta(\mathcal{W}(\mu_i), \mathcal{W}(\mu)) = \mathcal{O}(\|\mu - \mu_i\|_2).$$

Since $\mathcal{U}(\mu) = V\mathcal{W}(\mu)$, $\forall \mu \in D$, we have

$$\|\sin \Theta(\mathcal{U}(\mu_i), \mathcal{U}(\mu))\|_2 = \mathcal{O}(\|\mu - \mu_i\|_2).$$

In other words, there is a basis U for $\mathcal{U}(\mu_i)$ such that $U = U_i + \mathcal{O}(\|\mu - \mu_i\|_2)$. Thus,

$$\begin{aligned} U_{\perp}^* A(\mu) U &= U_{\perp}^* A(\mu_i) U_i + \mathcal{O}(\|\mu - \mu_i\|_2) \\ &= U_{\perp}^* (U_i^{\perp} (U_i^{\perp})^* + U_i U_i^*) A(\mu_i) U_i + \mathcal{O}(\|\mu - \mu_i\|_2) \\ &= \mathcal{O}(\|\mu - \mu_i\|_2). \end{aligned}$$

Therefore, we have $\nabla \rho^2 = 0$ at $\mu = \mu_i$, which completes the proof. \square

If $\mu_i \in \mathcal{S}$ satisfies the assumptions of Theorem 3.3, we have that $\lambda_i^{(i)}$ is simple, and thus, Theorem 3.12 holds for the choice $r = 1$, since $\lambda_i^{(1)} < \lambda_i^{(2)}$ and $r = 1 \leq \ell$. By the pinching theorem, this implies that the lower bounds returned by the procedure explained in Remark 3.9 (that is, adaptively choosing r to maximize $\lambda_{\text{SLB}}(\mu; \mathcal{S}, \ell)$) also satisfy the assertion of Theorem 3.12.

3.2.5 A priori convergence estimates

Using the interpolation results from Corollary 3.10 and Theorem 3.12 we obtain the following *a priori* convergence estimates for $\lambda_{\text{SUB}}(\mu; \mathcal{S}, \ell)$ and $\lambda_{\text{SLB}}(\mu; \mathcal{S}, \ell)$.

Theorem 3.13. *Let $\mu_i \in \mathcal{S}$ be such that $\lambda_{\min}(A(\mu_i))$ is simple and let $h > 0$ be such that $\lambda_{\min}(A(\mu))$, $\lambda_{\text{SUB}}(\mu; \mathcal{S}, \ell)$ and $\lambda_{\text{SLB}}(\mu; \mathcal{S}, \ell)$ are twice differentiable on $B(\mu_i, h)$. Then there exist constants $C_1, C_2 > 0$ such that*

$$\begin{aligned} |\lambda_{\text{SUB}}(\mu; \mathcal{S}, \ell) - \lambda_{\min}(A(\mu))| &< C_1 h^2, \\ |\lambda_{\text{SLB}}(\mu; \mathcal{S}, \ell) - \lambda_{\min}(A(\mu))| &< C_2 h^2, \end{aligned}$$

for all $\mu \in B(\mu_i, h)$.

Proof. Let $\mu \in B(\mu_i, h)$. Expanding $\lambda_{\min}(A(\mu))$ and $\lambda_{\text{SUB}}(\mu; \mathcal{S}, \ell)$ around μ_i using a second-order Taylor polynomial expansion and using (3.26) and the results of Corollary 3.10, we obtain

$$\lambda_{\text{SUB}}(\mu; \mathcal{S}, \ell) - \lambda_{\min}(A(\mu)) = \frac{1}{2} (\nabla^2 \lambda_{\text{SUB}}(\tilde{\mu}; \mathcal{S}, \ell) - \nabla^2 \lambda_{\min}(A(\hat{\mu}))) (\mu - \mu_i)^2,$$

for $\tilde{\mu}, \hat{\mu} \in [\mu_i, \mu]$. The first inequality now holds for

$$C_1 = \max_{\tilde{\mu} \in B(\mu_i, h)} \|\nabla^2 \lambda_{\text{SUB}}(\tilde{\mu}; \mathcal{S}, \ell)\|_2 + \max_{\hat{\mu} \in B(\mu_i, h)} \|\nabla^2 \lambda_{\min}(A(\hat{\mu}))\|_2.$$

The second inequality can be shown in the same way using (3.26) and Theorem 3.12. \square

Remark 3.14. *To ensure the differentiability conditions on $\lambda_{\min}(A(\mu))$ and $\lambda_{\text{SUB}}(\mu; \mathcal{S}, \ell)$ needed in the assumptions of Theorem 3.13, it is sufficient that the smallest eigenvalues $\lambda_{\min}(A(\mu))$ and $\lambda_{\text{SUB}}(\mu; \mathcal{S}, \ell)$ stay simple on $B(\mu, h)$, see [KMMM15]. A simple criterion for differentiability of $\lambda_{\text{SLB}}(\mu; \mathcal{S}, \ell)$ is not available, since (3.20) involves $\eta(\mu)$, which depends on the solution of the linear program (3.11) $\lambda_{\text{LB}}(\mu; \mathcal{S})$ and, thus, is not necessarily smooth around μ_i .*

Since $h = \mathcal{O}(M^{1/d})$, the convergence estimates obtained in Theorem 3.13 are of practical importance only for small values of $d = 1, 2, \dots$. However, if $A(\mu)$ is an analytic function in μ , we can expect much faster convergence than the one guaranteed by Theorem 3.13, as explained in the following section and observed for the numerical experiments presented in Section 3.5.

A priori convergence estimates for analytic $A(\mu)$ in the one-parameter case

In the following, we analyse the convergence of our subspace bounds for a special case: We assume that $A(\mu)$ depends analytically on one parameter $\mu \in [-1, 1]$ and, moreover, the eigenvalue $\lambda_{\min}(A(\mu))$ is simple and separated by at least $\delta_0 > 0$ from the rest of the spectrum for all $\mu \in [-1, 1]$.

Let E_R denote the open elliptic disc in the complex plane with foci ± 1 and the sum of its half axes equal to R . Under the above assumptions, there is $R_0 > 1$ such that the (suitably normalized) eigenvector $v(\mu)$ belonging to $\lambda_{\min}(A(\mu))$ admits an analytic extension $v: E_{R_0} \rightarrow \mathbb{C}^n$; see, e.g., [Kat95, RS78]. Note that v can be chosen to have norm 1 on $[-1, 1]$, see [RS78, Theorem XII.4], but this is not the case on E_{R_0} in general. Let $\mathcal{S} = \{\mu_1, \dots, \mu_M\}$ contain the Chebyshev nodes $\mu_i = \cos(\frac{2i-1}{2M}\pi)$ and set $v_i := v(\mu_i)$. The corresponding vector-valued interpolating polynomial is given by

$$p_M(\mu) = \ell_1(\mu)v_1 + \dots + \ell_M(\mu)v_M, \tag{3.31}$$

with the Lagrange polynomials $\ell_1, \dots, \ell_M: [-1, 1] \rightarrow \mathbb{R}$.

3.2. Subspace acceleration

Let $T_M(x) = \cos(M \arccos(x))$ denote M -th Chebyshev polynomial on $[-1, 1]$. The interpolation error can be expressed in the following way [MH03, Lemma 6.6]

$$v(\mu) - p_M(\mu) = \int_{E_R} \frac{T_M(\mu) v(z)}{T_M(z)(z - \mu)} dz,$$

for any $1 < R < R_0$, and bounded by

$$\|v(\mu) - p_M(\mu)\|_2 \leq \int_{E_R} \frac{|T_M(\mu)| \|v(z)\|_2}{|T_M(z)| |z - \mu|} |dz|. \quad (3.32)$$

We can further simplify (3.32) using [MH03, Corollary 6.6A], which yields

$$\max_{\mu \in [-1, 1]} \|v(\mu) - p_M(\mu)\|_2 \leq \frac{(R + R^{-1})C}{(R^{M+1} - R^{-M-1})(R + R^{-1} - 2)}, \quad (3.33)$$

with $C = \sup_{z \in E_R} \|v(z)\|_2$. This result is utilized in the proof of the following theorem, which shows exponential convergence of our subspace bounds.

Theorem 3.15. *Under the setting described above, the subspace lower and upper bounds for $\ell = r = 1$ satisfy*

$$\lambda_{\text{SUB}}(\mu; \mathcal{S}, 1) - \lambda_{\min}(A(\mu)) \leq C_U R^{-2M}, \quad (3.34)$$

$$\lambda_{\min}(A(\mu)) - \lambda_{\text{SLB}}(\mu; \mathcal{S}, 1) \leq C_L R^{-2M}, \quad (3.35)$$

for every $\mu \in [-1, 1]$, with constants C_U, C_L independent of M and μ .

Proof. For $\ell = 1$, the subspace used in our bounds takes the form $\mathcal{V} = \text{span}\{v_1, \dots, v_M\}$. The interpolating polynomial defined in (3.31) clearly satisfies $p_M(\mu) \in \mathcal{V}$, and hence (3.33) yields the following bound on the angle between \mathcal{V} and $v(\mu)$:

$$\min_{\tilde{v} \in \mathcal{V}} \|\tilde{v} - v(\mu)\|_2 \lesssim R^{-M}. \quad (3.36)$$

By approximation results for Ritz values [Saa92, Theorem 4.6, Proposition 4.5] and (3.36), we have

$$\begin{aligned} \lambda_{\text{SUB}}(\mu; \mathcal{S}, 1) - \lambda_{\min}(A(\mu)) &\leq \|A(\mu) - \lambda_{\min}(A(\mu))I\|_2 \left(1 + \frac{\gamma_\mu^2}{\delta_\mu^2}\right) \left(\min_{\tilde{v} \in \mathcal{V}} \|\tilde{v} - v(\mu)\|_2\right)^2 \\ &\lesssim \|A(\mu) - \lambda_{\min}(A(\mu))I\|_2 \left(1 + \frac{\gamma_\mu^2}{\delta_\mu^2}\right) R^{-2M}, \end{aligned}$$

where $\gamma_\mu = \|VV^*A(\mu)(I - VV^*)\|_2$ and δ_μ equals the distance between $\lambda_{\min}(A(\mu))$ and the rest of the spectrum of $A(\mu)$. Since $A(\mu)$ is bounded for all $\mu \in [-1, 1]$, and $\delta_\mu > \delta_0$, this proves inequality (3.34) with a constant C_U independent of M and μ .

Chapter 3. Low-rank approach for parameter dependent Hermitian eigenvalue problem

To prove (3.35), we first note that the arguments from the proof of Theorem 3.12 can be utilized to show that

$$\lambda_{\text{SLB}}(\mu; \mathcal{S}, 1) = \lambda_{\text{SUB}}(\mu; \mathcal{S}, 1) - \frac{2\rho^2}{\delta + \sqrt{\delta^2 + 4\rho^2}},$$

for sufficiently large M , where $\delta > \frac{\delta_0}{2} > 0$. Since $r = 1$, the quantity ρ coincides with the residual of the smallest Ritz vector of $A(\mu)$ with respect to \mathcal{V} . Approximation results for Ritz values [Par98, Theorem 11.7.1] and [Saa92, Theorem 4.6], together with (3.36), yield a bound on ρ

$$\begin{aligned} \rho &\leq \text{spread}(A(\mu)) \sqrt{1 + \frac{\gamma_\mu^2}{\delta_\mu^2} \min_{\tilde{v} \in \mathcal{V}} \|\tilde{v} - v(\mu)\|_2} \\ &\lesssim \text{spread}(A(\mu)) \sqrt{1 + \frac{\gamma_\mu^2}{\delta_\mu^2} R^{-M}}, \end{aligned}$$

with δ_μ and γ_μ as before, and $\text{spread}(A(\mu)) = |\lambda_{\max}(A(\mu)) - \lambda_{\min}(A(\mu))|$. Using similar arguments as for the subspace upper bounds, this proves the second inequality and completes the proof. \square

The maximal value of the exponent R in (3.34)–(3.35) depends on the analyticity radii on $[-1, 1]$, which are, as has been already discussed in Section 2.1.4, connected to the gaps between the smallest and the second smallest eigenvalue, and the variation in $A(\mu)$.

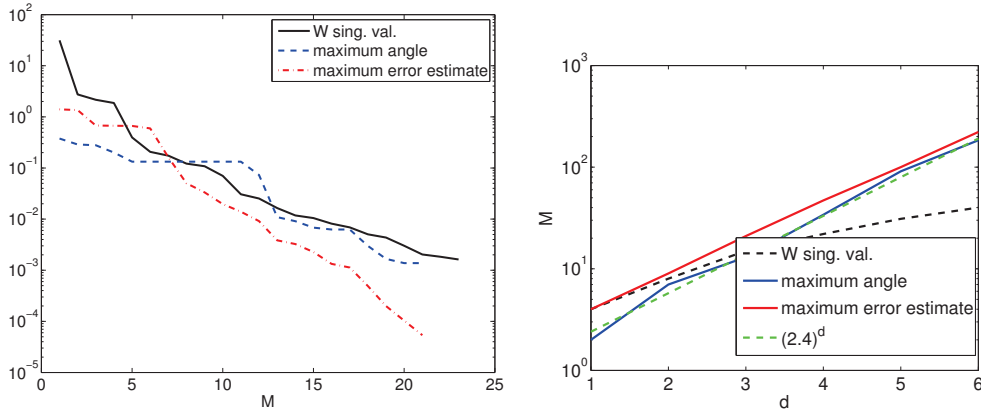
Theorem 3.15 covers only the situation when $\lambda_{\min}(A(\mu))$ stays simple on $[-1, 1]$. If this is not the case, we can expect a subexponential convergence rate which deteriorates with the number of eigenvalue crossings on D , since Theorem 3.15 still applies to intervals in between the eigenvalue crossings.

Remark 3.16. *Theorem 3.15 can be extended to the multiparameter case by using tensorized Chebyshev nodes, following the work in [AS12], which results in subexponential error decay $\mathcal{O}(R^{(-M)^{1/d}})$ that deteriorates as the number of parameters grow. Algorithm 3, when applied to Example 4.9 (with $\delta = 0.1$) from Section 3.5, exhibits similar convergence rate, as presented in Figure 3.3.*

Let $W \in \mathbb{R}^{n \times |\Xi|}$ be the matrix assembled with the vectors $v(\mu)$, defined as above, as columns for each $\mu \in \Xi$, and let two error indicators, the maximum error estimate (3.25) and the maximum angle, be defined as follows

$$\Delta_{\max} := \max_{\mu \in \Xi} \Delta(\mu; \mathcal{S}, \ell), \quad (3.37)$$

$$\alpha_{\max} := \max_{\mu \in \Xi} \angle(v(\mu), \mathcal{V}(\mathcal{S}, \ell)). \quad (3.38)$$



(a) Maximum error estimate Δ_{\max} and the maximum angle α_{\max} w.r.t. M and the singular values of W for $d = 3$. (b) Minimal values of M such that $\Delta_{\max} < 10^{-4}$, $\alpha_{\max} < 10^{-2}$ and $\sigma_{M+1} < 10^{-2}$ w.r.t. d .

Figure 3.3: Convergence of Algorithm 3 for Example 4.9 with $\delta = 0.1$.

First, we compute the smallest values of M such that each of $\sigma_{M+1}(W) < 10^{-2}$, $\Delta_{\max} < 10^{-4}$, $\alpha_{\max} < 10^{-2}$ is fulfilled and present the evolution of the obtained values for M w.r.t. d in Figure 3.3b. We can see that the required number of iterations to attain fixed precision grows exponentially with d , indicating that Algorithm 3 also has a convergence rate $\exp(-M^{1/d})$. Moreover, for $d = 3$, we show the singular value decay of the matrix W as well as the convergence rate of Δ_{\max} and α_{\max} in Figure 3.3a. The results indicate that the sampled subspace $\mathcal{V}(\mathcal{S}, \ell)$ is close to optimal in approximating the dominant left singular vectors of W .

Relation to linear interpolation

Given a set of sampled functions values, a straightforward idea is to approximate the function on the whole domain using the linear interpolation. In the following theorem we show that in the special case when $A(\mu)$ is an affine function in $\mu = (\mu^{(2)}, \dots, \mu^{(Q)})$ as in (2.28):

$$A(\mu) = A_1 + \mu^{(2)} A_2 + \dots + \mu^{(Q)} A_Q,$$

the subspace lower bounds $\lambda_{\text{SLB}}(\mu; \mathcal{S}, \ell)$ are always at least as good approximation to $\lambda_{\min}(A(\mu))$ as the linear interpolation of the sampled values.

Theorem 3.17. *Suppose we are given \mathcal{S} , defined as above, and $\mu \in \text{conv}(\mathcal{S})$. Let $1 \leq i_1 < i_2 < \dots < i_Q \leq M$ such that $\mu \in \text{conv}\{\mu_{i_1}, \dots, \mu_{i_Q}\}$. We define $l(\mu) : \mathbb{R}^Q \rightarrow \mathbb{R}$ to be the linear function interpolating $\lambda_{\min}(A(\mu))$ at $\mu_{i_1}, \dots, \mu_{i_Q}$. Then we have*

$$l(\mu) \leq \lambda_{\text{LB}}(\mu; \mathcal{S}),$$

where $\lambda_{\text{LB}}(\mu; \mathcal{S})$ is defined in (3.11). Additionally, there exist $1 \leq i_1 < i_2 < \dots < i_Q \leq M$ such that $\mu \in \text{conv}\{\mu_{i_1}, \dots, \mu_{i_Q}\}$ and the corresponding function l satisfies $l(\mu) = \lambda_{\text{LB}}(\mu; \mathcal{S})$.

Proof. The dual problem of the lower bound minimization in (3.11) is:

$$\begin{aligned} \lambda_{\text{LB}}(\mu; \mathcal{S}) &= \max && b^T z \\ \text{s.t. } & z^T C &= & [1, \mu^{(2)}, \dots, \mu^{(Q)}]^T, \\ & z &\geq & 0, \end{aligned} \quad (3.39)$$

with C and b the corresponding constraint matrices. We can interpret (3.39) as an optimization problem over all possible representations of μ as a convex combination of the points in \mathcal{S} . Barycentric coordinates of μ on the simplex spanned by $\mu_{i_1}, \mu_{i_2}, \dots, \mu_{i_Q}$ clearly provide an admissible point of (3.39), immediately proving $l(\mu) \leq \lambda_{\text{LB}}(\mu; \mathcal{S})$.

Moreover, there is an optimal point z for (3.39) such that z has only Q non-zero coordinates, as each non-zero coordinate of z corresponds to one the active constraints in the optimal solution of the primal problem. This immediately gives that there exist $1 \leq i_1 < i_2 < \dots < i_Q \leq M$ such that $l(\mu) = \lambda_{\text{LB}}(\mu; \mathcal{S})$. \square

From (3.26), we have $\lambda_{\text{LB}}(\mu; \mathcal{S}) \leq \lambda_{\text{SLB}}(\mu; \mathcal{S}, \ell) \leq \lambda_{\min}(A(\mu))$. Combining this with the results of Theorem 3.17, we get

$$l(\mu) \leq \lambda_{\text{SLB}}(\mu; \mathcal{S}, \ell) \leq \lambda_{\min}(A(\mu)),$$

for all $\mu \in \text{conv}(\mathcal{S})$ and all functions ℓ obtained as linear interpolation of $\lambda_{\min}(A(\mu))$ on a simplex in \mathcal{S} containing μ . This shows that the subspace upper bounds $\lambda_{\text{SLB}}(\mu; \mathcal{S}, \ell)$ are at least as good approximation to $\lambda_{\min}(A(\mu))$ as the one obtained by linearly interpolating the computed values of $\lambda_{\min}(A(\mu))$.

3.2.6 Geometric interpretation

In SCM, the joint numerical range \mathcal{Y} is approximated using the convex polyhedra \mathcal{Y}_{UB} and \mathcal{Y}_{LB} , as shown in Figure 3.1. As the minimum of the linear program (3.11) can always be attained at a vertex of the polyhedron, in order to minimize the error in the SCM lower bounds, the vertices (corners) of \mathcal{Y}_{LB} need to be as close as possible to \mathcal{Y} . In the following, we present numerical evidence indicating that, given the same sample set \mathcal{S} , we are able to "cut the corners" of \mathcal{Y}_{LB} and obtain a better approximation of \mathcal{Y} by using the proposed subspace-accelerated approach.

We consider $\mu \in [-1, 1]$ and $A(\mu) = A_1 + \mu A_2$, with A_1, A_2 random Hermitian matrices. Suppose that $\ell = 1$ and $\mathcal{S} = \{-1, 1\}$. Having computed the smallest eigenpairs in the sample points in \mathcal{S} , we calculate the subspace bounds $\lambda_{\text{SUB}}(\mu; \{-1, 1\}, 1)$ and $\lambda_{\text{SLB}}(\mu; \{-1, 1\}, 1)$ and compare them with the SCM bounds $\lambda_{\text{UB}}(\mu; \{-1, 1\})$ and $\lambda_{\text{LB}}(\mu; \{-1, 1\})$, see Figure 3.4a. The corresponding approximations to \mathcal{Y} are shown in Figure 3.4b, indicating that the subspace-accelerated approach is indeed able to exploit information about the eigenvalue gaps to "cut the corners" of \mathcal{Y}_{LB} . Tighter bounds and a better approximation of \mathcal{Y} can be achieved by either increasing ℓ , as shown in Figures 3.4c and 3.4d for the case $\ell = 2$, or by enriching the sample set \mathcal{S} ,

as presented in Figures 3.4e and 3.4f for the case $\mathcal{S} = \{-1, 0, 1\}$. Moreover, the numerical examples presented in Figures 3.4a, 3.4c and 3.4e serve as a experimental evidence for the theoretical results presented in Sections 3.1.6 and 3.2.4, showing that the SCM upper bound and both subspace bounds interpolate the derivative of $\lambda_{\min}(A(\mu))$ in the sampled points, while the SCM lower bound does not.

3.3 Heuristic variants

As we will see later in the numerical experiments section, the existing approaches, such as SCM, often do not provide satisfactory results, leading to proposals of various heuristic strategies for approximating the smallest eigenvalues of $A(\mu)$. Such approaches (see e.g. [MN15, MMO⁺00]) provide bounds which are usually very easy to compute and, although not rigorous, in practice often very accurate.

In some of the numerical experiments (especially in Example 3.24), our subspace-accelerated version of SCM (Algorithm 3) also exhibits slow convergence, where the subspace lower bounds converge rather slowly in the initial phase of the algorithm, in contrast to the subspace upper bounds. This slow convergence can be viewed as a price that needs to be paid in order to maintain the reliability of the lower bounds. In the following, we propose an alternative that is heuristic (i.e., its reliability is not guaranteed) and is observed to converge faster in the initial phase.

The alternative consists of simply subtracting the residual norm from the upper bound:

$$\lambda_{\text{SUB}}(\mu; \mathcal{S}, \ell) - \|A(\mu)u - \lambda_{\text{SUB}}(\mu; \mathcal{S}, \ell)u\|_2, \quad (3.40)$$

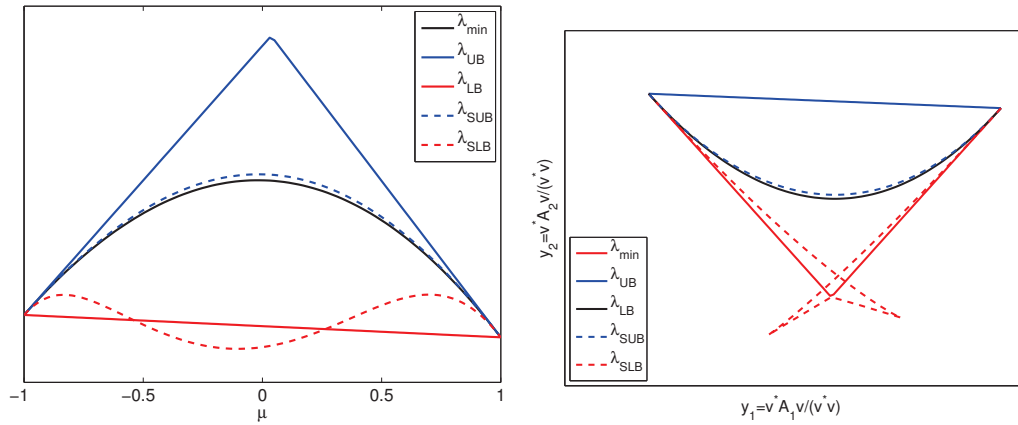
where u with $\|u\|_2 = 1$ is a Ritz vector belonging to the smallest Ritz value $\lambda_{\text{SUB}}(\mu; \mathcal{S}, \ell)$ of $A(\mu)$ with respect to \mathcal{V} . A basic first-order perturbation result for Hermitian matrices [Par98, Theorem 4.5.1] implies that (3.40) constitutes a lower bound for *an* eigenvalue of $A(\mu)$, but not necessarily the smallest one. There is a risk, especially in the very beginning, that (3.40) is actually larger than the smallest eigenvalue, see Section 3.5 for examples. However, in all numerical experiments we have observed that a small number of iterations suffices until (3.40) becomes a lower bound for the smallest eigenvalue.

Remark 3.18. *When using the residual-based lower bound (3.40), it makes sense to also adjust the error measure (3.12) that drives the sampling strategy to*

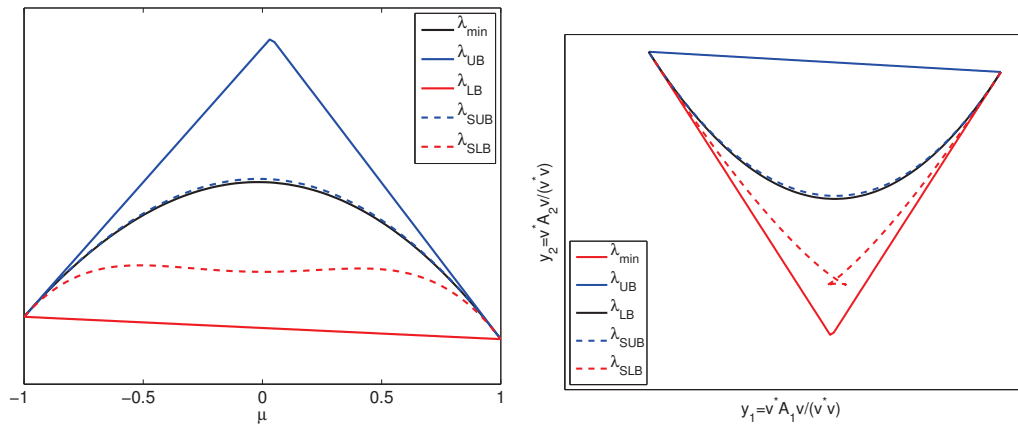
$$\max_{\mu \in \Xi} \frac{\|A(\mu)u - \lambda_{\text{SUB}}(\mu; \mathcal{S}, \ell)u\|_2}{|\lambda_{\text{SUB}}(\mu; \mathcal{S}, \ell)|},$$

and stop the iteration when this error estimate drops below ε_{SCM} .

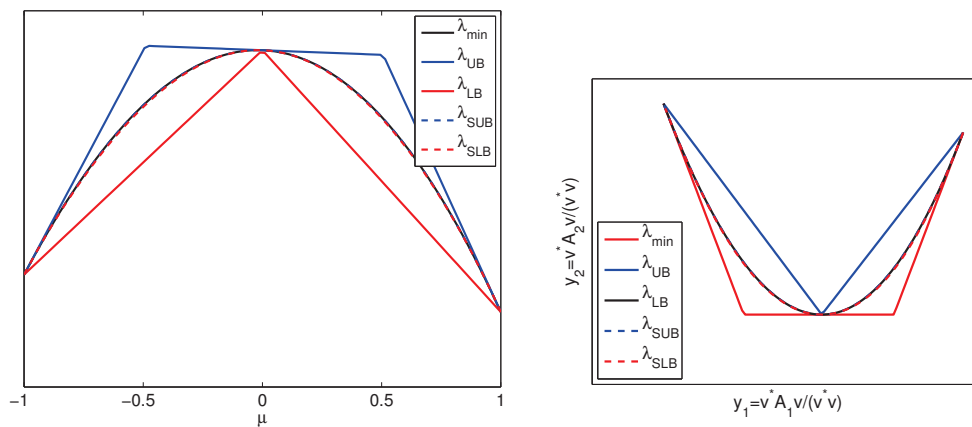
Remark 3.19. *The subspace lower bounds formula (3.20) can also be used to motivate various heuristic approaches. For example, instead of calculating $\eta(\mu)$ rigorously like in Section 3.2.2, we can use the computed smallest Ritz values as an estimate for the eigenvalue gaps and, thus, also to approximate $\eta(\mu)$. Similar reasoning has already been used in [MMO⁺00, FMPV15],*



(a) Comparison of the subspace-accelerated and the SCM bounds for $\lambda_{\min}(A(\mu))$ for $\ell = 1$ and $\mathcal{S} = \{-1, 1\}$. (b) Approximations to \mathcal{Y} for $\ell = 1$ and $\mathcal{S} = \{-1, 1\}$.



(c) Comparison of the subspace-accelerated and the SCM bounds for $\lambda_{\min}(A(\mu))$ for $\ell = 2$ and $\mathcal{S} = \{-1, 1\}$. (d) Approximations to \mathcal{Y} for $\ell = 2$ and $\mathcal{S} = \{-1, 1\}$.



(e) Comparison of the subspace-accelerated and the SCM bounds for $\lambda_{\min}(A(\mu))$ for $\ell = 1$ and $\mathcal{S} = \{-1, 0, 1\}$. (f) Approximations to \mathcal{Y} for $\ell = 1$ and $\mathcal{S} = \{-1, 0, 1\}$.

Figure 3.4: Illustration of the subspace-accelerated approach and comparison to SCM for $Q = 2$ and $M = 2$.

where the proposed lower bounds for the smallest eigenvalues rely on the working assumption that the second smallest computed Ritz value $\lambda_{\mathcal{V}}^{(2)}$ is an accurate approximation to the second smallest eigenvalue of $A(\mu)$.

3.4 Algorithm

In this section, we present a summary, in form of Algorithm 3, of our subspace-accelerated approach introduced in Section 3.2 and discuss its implementation and computational complexity.

Algorithm 3 Subspace-accelerated SCM

Input: Training set $\Xi \subset D$, affine linear decomposition such that $A(\mu) = \theta_1(\mu)A_1 + \dots + \theta_Q(\mu)A_Q$ is Hermitian for every $\mu \in \Xi$. Relative error tolerance ε_{SCM} .

Output: Set $\mathcal{S} \subset \Xi$ with corresponding eigenvalues $\lambda_i^{(j)}$ and an orthonormal eigenvector basis V of $\mathcal{V}(\mathcal{S}, \ell)$, such that $\frac{\lambda_{\text{SUB}}(\mu; \mathcal{S}, \ell) - \lambda_{\text{SLB}}(\mu; \mathcal{S}, \ell)}{|\lambda_{\text{SUB}}(\mu; \mathcal{S}, \ell)|} < \varepsilon_{\text{SCM}}$ for every $\mu \in \Xi$.

1: Compute $\lambda_{\min}(A_q), \lambda_{\max}(A_q)$ for $q = 1, \dots, Q$, defining \mathcal{B} according to (3.8).

2: $M = 0, \mathcal{S} = \emptyset$

3: Set μ_{\max} to be a randomly chosen element of Ξ .

4: **while** $\Delta(\mu_{\max}; \mathcal{S}, \ell) > \varepsilon_{\text{SCM}}$ **do**

5: $\mu_{M+1} \leftarrow \{\mu_{\max}\}$

6: Compute smallest eigenpairs $(\lambda_{M+1}^{(1)}, v_{M+1}^{(1)}), \dots, (\lambda_{M+1}^{(\ell)}, v_{M+1}^{(\ell)})$ of $A(\mu_{M+1})$.

7: $\mathcal{S} \leftarrow \mathcal{S} \cup \mu_{M+1}$

8: Update $V^* A_q V$ and $V^* A_q^* A_{q'} V$ for all $q, q' = 1, \dots, Q$.

9: **for** $\mu \in \Xi$ **do**

10: **if** $C\Delta(\mu; \mathcal{S}, \ell) < \Delta_{\max}$ **then**

11: Exit the for loop.

12: **end if**

13: Recompute $\lambda_{\text{SUB}}(\mu; \mathcal{S}, \ell) = \lambda_{\min}(V^* A(\mu) V)$.

14: Recompute $\rho = \sqrt{\lambda_{\max}(U^* A(\mu)^* A(\mu) U - \Lambda_U^2)}$ according to (3.41).

15: Recompute $y_\mu = \arg \min_{y \in \mathcal{B}_{\text{LB}}(\mathcal{S})} \theta(\mu)^T y$ and updated \check{y}_μ according to (3.24).

16: Recompute $\eta(\mu) \leftarrow \theta(\mu)^T \check{y}_\mu$.

17: Recompute $\lambda_{\text{SLB}}(\mu; \mathcal{S}, \ell)$ according to (3.20).

18: Recompute $\Delta(\mu; \mathcal{S}, \ell)$ according to (3.25) and update Δ_{\max} and μ_{\max} .

19: **end for**

20: $M \leftarrow M + 1$

21: **end while**

3.4.1 Computational details

The efficient implementation of our proposed approach for computing upper and lower bounds for $\lambda_{\min}(A(\mu))$ requires care in order to avoid unnecessary computations. Some implementation details are discussed in the following.

Computation of $\lambda_{\min}(A(\mu))$. For computing a few smallest eigenpairs of a large-scale Her-

Chapter 3. Low-rank approach for parameter dependent Hermitian eigenvalue problem

mitian matrix, it is preferable to use an iterative solver, such as the Lanczos method, presented in Section 2.1.3, or the LOBPCG [Kny01]. In our implementation we use the the MATLAB built-in function `eigs` of the Lanczos method, which is based on ARPACK [LSY98]. As discussed in Remark 2.5, if it is *a priori* known that $A(\mu)$ positive definite, it is often advisable to use the inverse Lanczos method instead, together with the sparse Cholesky factorization.

Computation of $V^* A(\mu) V$ and $U^* A(\mu)^* A(\mu) U$. By the affine linear decomposition (3.2),

$$V^* A(\mu) V = \theta_1(\mu) V^* A_1 V + \dots + \theta_Q(\mu) V^* A_Q V.$$

A standard technique in RBM, we compute and store the $M\ell \times M\ell$ matrices $V^* A_q V$, and update them as new columns are added to V . In turn, the computation of $V^* A(\mu) V$, which is needed to evaluate the upper bound for every $\mu \in \Xi$, becomes negligible as long as $M\ell \ll n$. Similarly, the evaluation of $U^* A(\mu)^* A(\mu) U$ needed for ρ becomes negligible after the precomputation of $V^* A_q^* A_{q'} V$ for all $q, q' = 1, \dots, Q$.

Computation of ρ . The quantity $\rho = \|A(\mu)U - U\Lambda_U\|_2$ with $\Lambda_U = U^* A(\mu)U = \text{diag}(\lambda_{\mathcal{V}}^{(1)}, \dots, \lambda_{\mathcal{V}}^{(r)})$ can be computed by solving an $r \times r$ eigenvalue problem:

$$\begin{aligned} \rho^2 &= \lambda_{\max}((A(\mu)U - U\Lambda_U)^* (A(\mu)U - U\Lambda_U)) \\ &= \lambda_{\max}(U^* A(\mu)^* A(\mu)U - \Lambda_U^2). \end{aligned} \quad (3.41)$$

Note that U and Λ_U both depend on μ .

Computing $\lambda_{\text{LB}}(A(\mu))$. Computationally the most expensive part of computing $\lambda_{\text{SLB}}(A(\mu))$ is solving (3.11). For solving linear programs, in our implementation, we use the interior point method, as it offers the best time complexity in the general case. However, when the dimensionality of the problem is small, $Q \leq 3$, we use the simplex method instead. It becomes a viable alternative, especially due to the fact that the linear program (3.11) changes only slightly from one iteration to the other and the simplex method allows for updating previously computed solutions. For example, if the newly added constraints do not cut off the previously optimal vertex, it will stop immediately.

Computation of the next parameter sample μ_{M+1} . The next parameter sample μ_{M+1} is computed as the maximizer of the error estimate (3.25) on Ξ . In every iteration, this requires recomputing the bounds $\lambda_{\text{SUB}}(\mu; \mathcal{S}, \ell)$ and $\lambda_{\text{SLB}}(\mu; \mathcal{S}, \ell)$ on the whole training set Ξ , which can become computationally quite expensive. Instead, as explained in Remark 2.17, the search for μ_{M+1} (lines 9–19 in Algorithm 3) can be optimized using the saturation assumption, which in the current setting takes the following form: there exists $C_{\text{sat}} > 0$ such that

$$\Delta(\mu; \mathcal{S}^*, \ell) < C_{\text{sat}} \Delta(\mu; \mathcal{S}, \ell), \quad \forall \mathcal{S}^* \supset \mathcal{S}, \forall \mu \in D. \quad (3.42)$$

Using the saturation assumption in the context of parameter-dependent eigenvalue problems has already been proposed in [CHMR09]. As described in Remark 2.17, using the error estimates from the previous iteration together with (3.42) often allows us to skip recomputing $\Delta(\mu; \mathcal{S}, \ell)$ for a number of points in Ξ when searching for the next parameter sample μ_{M+1} . In Algorithm 3, we use the same notation as in Remark 2.17 for Δ_{\max} and μ_{\max} , which are the current maximum error estimate and the point in Ξ where it was attained, respectively.

It is important to note that the saturation assumption (3.42) can be easily proven with $C_{\text{sat}} = 1$ for all $\mu \in D$ such that $\lambda_{\text{SLB}}(\mu; \mathcal{S}, \ell)$ and $\lambda_{\text{SUB}}(\mu; \mathcal{S}, \ell)$ are of the same sign. For example, if $\lambda_{\text{SUB}}(\mu; \mathcal{S}, \ell) > 0$ and $\lambda_{\text{SLB}}(\mu; \mathcal{S}, \ell) > 0$, we have

$$\Delta(\mu; \mathcal{S}, \ell) = 1 - \frac{\lambda_{\text{SLB}}(\mu; \mathcal{S}, \ell)}{\lambda_{\text{SUB}}(\mu; \mathcal{S}, \ell)} < 1.$$

Since both eigenvalue bounds are monotonically improving throughout the iterations, this immediately implies that the error estimates $\Delta(\mu; \mathcal{S}, \ell)$ are monotonically decreasing, thus proving (3.42) with $C_{\text{sat}} = 1$. The discussion is similar for the case when both bounds are negative. In the numerical examples considered in Section 3.5, we can see that $\max_{\mu \in \Xi} \Delta(\mu; \mathcal{S}, \ell) < 1$ usually holds after only a few iterations, implying that the bounds are of the same sign on the whole domain, and making the use of the saturation assumption completely justified.

3.4.2 Parameter value selection

Choice of r . The subspace lower bounds $\lambda_{\text{SLB}}(\mu; \mathcal{S}, \ell)$ clearly depend on the choice of r , number of the smallest Ritz vectors used in the construction of the subspace U . As explained in Remark 3.9, r is chosen adaptively for each $\mu \in \Xi$ by taking the maximal value of $\lambda_{\text{SLB}}(\mu; \mathcal{S}, \ell)$ among a few small values of $r = 0, 1, 2, \dots$

Choice of ℓ . Clearly, a larger choice of ℓ can be expected to lead to better bounds. On the other hand, a larger value of ℓ increases the computational cost. Intuitively, choosing ℓ larger than one appears to be most beneficial when the gap between the smallest and second smallest eigenvalues is small or even vanishes. One could, for example, choose ℓ such that $\lambda_i^{(\ell+1)} - \lambda_i^{(1)}$ exceeds a certain threshold. However, in absence of *a priori* information on eigenvalue gaps, it might be the wisest to simply choose $\ell = 1$ for all μ_i , as shown in Figure 3.5, where we present the convergence rates of Algorithm 3 for different choices of ℓ .

Choice of Ξ . Depending on d , the training set Ξ is either chosen as a tensorized grid in \mathbb{R}^d or as a subset of \mathbb{R}^d containing few thousand randomly selected points. Using a tensorized grid makes sense only if D is a hyperrectangle and is viable only for small values of d , say $d \leq 3$, whereas in Section 3.5, we consider numerical examples with d up to 9. Choosing Ξ as a random subset of D is usually a more efficient option and as such,

is a standard practice in the reduced basis method, see [RHP08, HSZ14]. It is not *a priori* clear how many points exactly to include in Ξ , as it depends on the problem, in particular the dimensionality d . The results presented in Figure 3.6 indicate that having $|\Xi| \approx 10^3$ is usually sufficient to provide reliable results, as further enrichment of Ξ does not influence the number of iterations of Algorithm 3, indicating that the obtained reduced-order model already is good enough on the whole D .

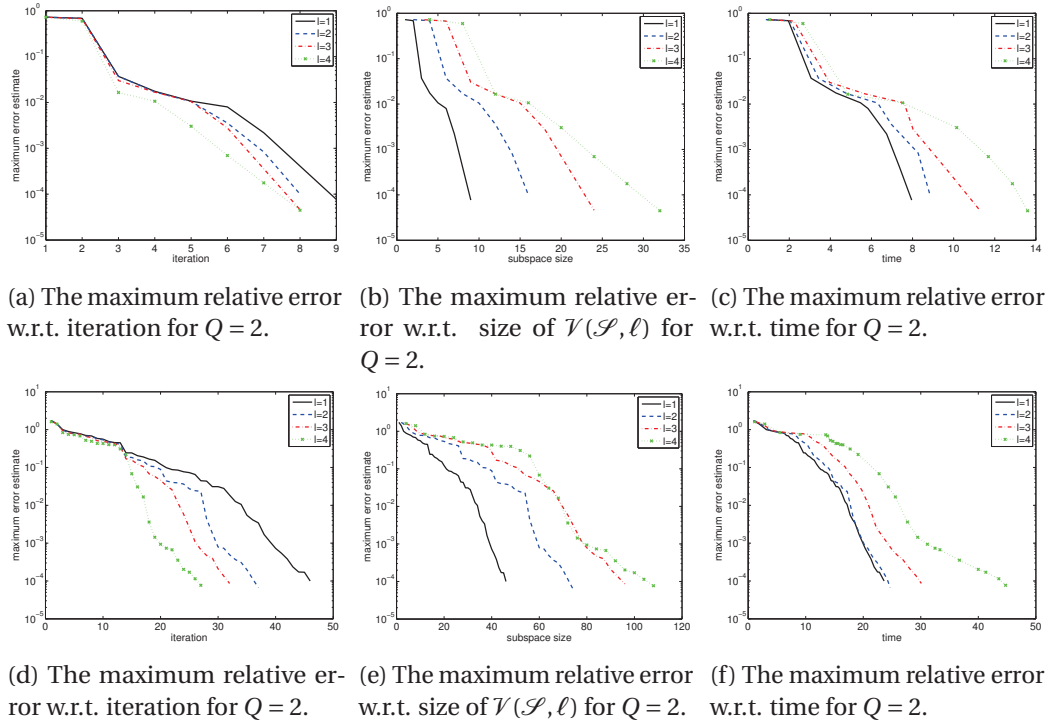


Figure 3.5: Convergence of Algorithm 3 applied to Example 4.9 with $\delta = 0.1$ and $Q = 2, 4$ for different choices of ℓ .

3.4.3 Computational complexity

Algorithm 3 summarizes our proposed procedure for computing subspace lower and upper bounds, taking into account implementational details from Section 3.4.1. Similarly as SCM, the algorithm requires solution of $2Q + M$ eigenvalue problems of size $n \times n$ for determining both the bounding box \mathcal{B} at the start and the smallest $\ell + 1$ eigenpairs in each iteration. Clearly, the latter part will become more expensive than in SCM if $\ell \geq 1$. However, we expect that this increase can be mitigated significantly in practice by the use of block algorithms. More specifically, when using a block eigenvalue solver such as LOBPCG [Kny01] and efficient implementations of block matrix-vector products with the matrix A (and its preconditioner), the computation of ℓ smallest eigenvalues will not be much more expensive as long as ℓ remains modest.

Computing $\lambda_{\text{SUB}}(\mu; \mathcal{S}, \ell)$ and $\lambda_{\text{SLB}}(\mu; \mathcal{S}, \ell)$ for all $\mu \in \Xi$ amounts to solving $M|\Xi|$ eigenprob-

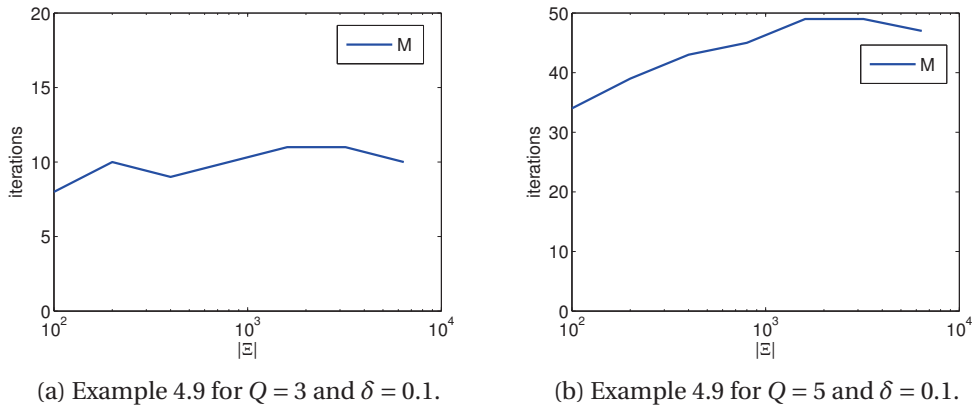


Figure 3.6: The final sample size M in Algorithm 3 applied to Example 4.9 as a function of $|\Xi|$.

lems of size $M\ell \times M\ell$, as well as $M|\Xi|$ LP problems with Q variables and $2Q + M$ constraints. As long as $M\ell \ll n$, these parts will be negligible, and the cost of Algorithms 2 and 3 will be approximately equal. Moreover in practice, as explained in Section 3.4.1, by assuming the saturation assumption, for a fixed $\mu \in \Xi$, the bounds $\lambda_{\text{SUB}}(\mu; \mathcal{S}, \ell)$ and $\lambda_{\text{SLB}}(\mu; \mathcal{S}, \ell)$ do not have to be recomputed in every iteration, but rather only a few times throughout the execution.

Remark 3.20. *As solving a dense $k \times k$ eigenvalue problem has complexity $\mathcal{O}(k^3)$, computing subspace upper bounds get significantly more expensive as M gets larger. It is not a priori clear what is the critical value of M when the subspace-accelerated approach becomes more expensive than SCM or a direct approach, where for each $\mu \in \Xi$ the eigenvalue problem would be solved exactly. This depends on a number of different factors such as:*

- *the ratio between the computational times needed for computing $\lambda_{\min}(A(\mu))$ and solving a small dense eigenvalue problem $V^* A(\mu) V$ for a single value of μ ,*
- *size of the training set Ξ ,*
- *computational savings due to the saturation assumption (3.42).*

This issue can be spotted in Section 4.4, in particular in Example 4.12, where the subspace-accelerated approach is only slightly faster than the direct approach. Similar problems have already been addressed in the reduced basis framework for linear systems using a domain splitting technique, where the algorithm is run on each component of the parameter domain independently, see [EPR10, HDO11]. If needed, such a domain splitting technique could easily be integrated in Algorithm 3 as well. However, it is important to emphasize that the idea of domain splitting is usually most effective when d is not too large and there is an easy way to split D into a few connected components.

3.5 Applications and numerical examples

In this section, we report on the performance of our proposed approach for a number of examples. Algorithms 2 and 3 have been implemented in MATLAB Version 7.14.0.739 (R2012a) and all experiments have been performed on an Intel Xeon CPU E31225 with 4 cores, 3.1 GHz, and 8 GB RAM.

We compare Algorithm 3 with Algorithm 2 by computing the maximum relative error ratio (3.12). Additionally, we compare the convergence of the bounds from Sections 3.1 and 3.2 towards the exact smallest eigenvalues by measuring the absolute error

$$\max_{\mu \in \Xi} |\text{bound}(\mu) - \lambda_{\min}(A(\mu))|, \quad (3.43)$$

for the corresponding bound, both with respect to the number of iterations and with respect to the execution time (in seconds).

When implementing and testing Algorithms 2 and 3, we have made the following choices. We set the relative tolerance to $\varepsilon_{\text{SCM}} = 10^{-4}$, the maximum number of iterations to $M_{\max} = 200$ and the surrogate set Ξ to be a random subset of D containing 1000 elements. The smallest eigenpairs of $A(\mu_i)$ have been computed using the MATLAB built-in function `eigs`, which is based on ARPACK [LSY98], with the tolerance set to 10^{-10} . For solving the linear program (4.6), we have used the MOSEK 7 Matlab toolbox [ApS15] implementation of the interior point method and the simplex method with updating. In all experiments, we have used Algorithm 3 with the number of smallest eigenpairs included in \mathcal{V} set to $\ell = 1$, since this already provided significant improvements over Algorithm 2. In the first five iterations of Algorithm 3 we have worked with the saturation constant set to $C_{\text{sat}} = +\infty$ and $C_{\text{sat}} = 1$ in the following iterations. For choosing r from Section 3.2.2, we have tested all values $r = 0, 1, \dots, Q$, see Remark 3.9.

3.5.1 Random matrices

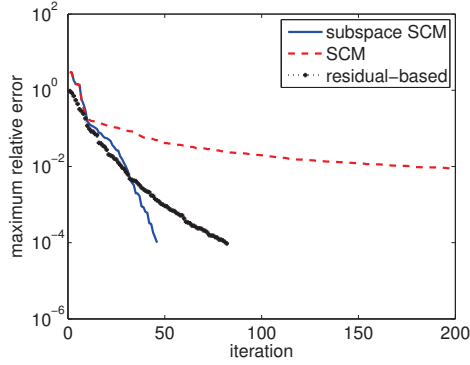
We first consider an academic example, where a random dense Hermitian matrix $A_1 \in \mathbb{C}^{n \times n}$ is perturbed, to a certain extent, by random Hermitian matrices $A_2, \dots, A_Q \in \mathbb{C}^{n \times n}$:

$$A(\mu) = A_1 + \mu_2 A_2 + \dots + \mu_Q A_Q,$$

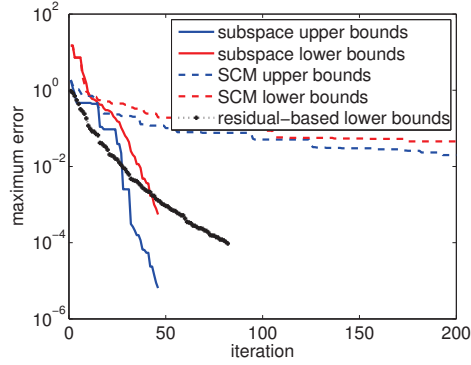
where $\mu = (\mu_2, \dots, \mu_Q) \in D = [0, \delta]^{Q-1}$.

Example 3.21. *We consider $Q = 4$, $n = 1000$, $\delta = 0.2$, with A_1, A_2, A_3, A_4 having real random entries from the unit normal distribution. The performances of both algorithms is shown in Figure 4.3. The convergence of Algorithm 2 flattens after around 25 iterations and does not reach the desired tolerance, while the convergence of Algorithm 3 is much faster and reaches the desired tolerance within 47 iterations. We have also considered an optimized version of Algorithm 2, where we incorporate the optimized strategy for selecting μ_{M+1} based on the saturation assumption, as described in Section 3.4.1. Results of this modification on the performance of Algorithm 2 can be seen in Figure 3.7d. This modification speeds up Algorithm 2 significantly,*

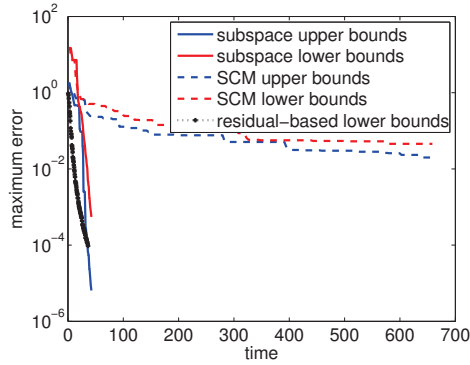
without notably affecting the accuracy. However, Algorithm 3 still outperforms Algorithm 2 both in terms of the computational time and the accuracy attained. Since Algorithm 3 converges quickly, there is no need to even consider the residual-based lower bounds from Section 3.3, but we still include the results in Figure 4.3 for the sake of completeness.



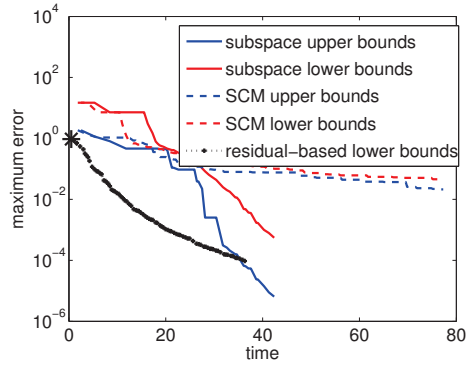
(a) Convergence of the maximum relative error ratio (3.12).



(b) Convergence of the error (3.43) for the bounds w.r.t. iteration.



(c) Convergence of the error (3.43) for the bounds w.r.t. time.



(d) Convergence of the error (3.43) for the bounds w.r.t. time.

Figure 3.7: Convergence plots for Algorithms 2 and 3 applied to Example 3.21.

3.5.2 Estimation of the coercivity constant

As explained in Section 2.3, *a posteriori* error estimation in model order reduction techniques for parametrized PDEs, such as the reduced basis method, requires reliable estimates for the coercivity constant [RHP08] defined as

$$\alpha(\mu) = \inf_{u \in X} \frac{a(u, u; \mu)}{\|u\|_X^2}, \quad (3.44)$$

where $a(\cdot, \cdot, \mu)$ is a coercive symmetric bilinear form on $X \times X$ representing the weak formulation of a PDE on a domain Ω_X and X is a suitable function space. As described in Section 2.3.2,

Chapter 3. Low-rank approach for parameter dependent Hermitian eigenvalue problem

a finite element discretization of (3.44) leads to the minimization problem

$$\alpha^n(\mu) = \inf_{v \in \mathbb{R}^n} \frac{v^T A(\mu) v}{v^T X v}, \quad (3.45)$$

where $A(\mu) \in \mathbb{R}^{n \times n}$ is the matrix discretizing $a(\cdot, \cdot, \mu)$ and $X = A(\bar{\mu}) + \tau M \in \mathbb{R}^{n \times n}$, where $M \in \mathbb{R}^{n \times n}$ is the mass matrix.

Minimizing (3.45) is clearly equivalent to computing the smallest eigenvalue of the generalized eigenvalue problem

$$A(\mu) v = \lambda X v.$$

As in Remark 2.4, we can transform it into a standard eigenvalue problem of the form (3.1) by computing the (sparse) Cholesky factorization $X = LL^T$:

$$L^{-1} A(\mu) L^{-T} w = \lambda w.$$

Hence, the matrices A_i appearing in Assumption 3.2 need to be replaced by

$$L^{-1} A_i L^{-T}, \quad i = 1, \dots, Q.$$

Note that, as described in Remark 2.4, it is often preferable to keep matrices $L^{-1} A_i L^{-T}$ in the factorized form.

In the following, we consider three numerical examples of this type from the rbMIT toolbox [HNPR10]. We only include brief explanations of the examples; more details can be found in [HNPR10] and [PR07].

Example 3.22. *This example concerns a linear elasticity model of a parametrized body (see Figure 3.8a). The parameter μ_1 determines the width of the hole in the body while the parameter μ_2 determines its Poisson's ratio. A discretization of the underlying PDE leads to the matrix $A(\mu) = \sum_{i=1}^Q \theta_i(\mu) A_i$, with $Q = 16$, $\mu = (\mu_1, \mu_2)$ and functions $\theta_i(\mu)$ that arise from the parametrization of the geometry. We choose $n = 2183$ and $D = [-0.1, 0.1] \times [0.2, 0.3]$. As can be seen from Figure 3.8, The results are similar to those presented in Example 3.21, with Algorithm 3 converging in 31 iteration and Algorithm 2 not reaching the desired tolerance.*

Example 3.23. *This example concerns a stationary heat equation on a parametrized domain (see Figure 3.9a). The parameter μ_1 determines the coefficient in the Robin boundary conditions while the parameter μ_2 determines the length of the domain. A discretization of the underlying PDE leads to the matrix $A(\mu) = \sum_{i=1}^Q \theta_i(\mu) A_i$, with $Q = 3$, $\mu = (\mu_1, \mu_2)$ and functions $\theta_i(\mu)$ arising from the parametrization of the geometry and boundary conditions. We choose $n = 1311$ and $D = [0.02, 0.5] \times [2, 8]$. As can be seen from Figure 3.9, the results are similar to those observed in Examples 3.21 and 3.22.*

Example 3.24. *This example concerns a stationary heat equation on a square domain divided into blocks (see Figure 3.10a). In each of the subdomains, one of the parameters μ_1, \dots, μ_9*

3.5. Applications and numerical examples

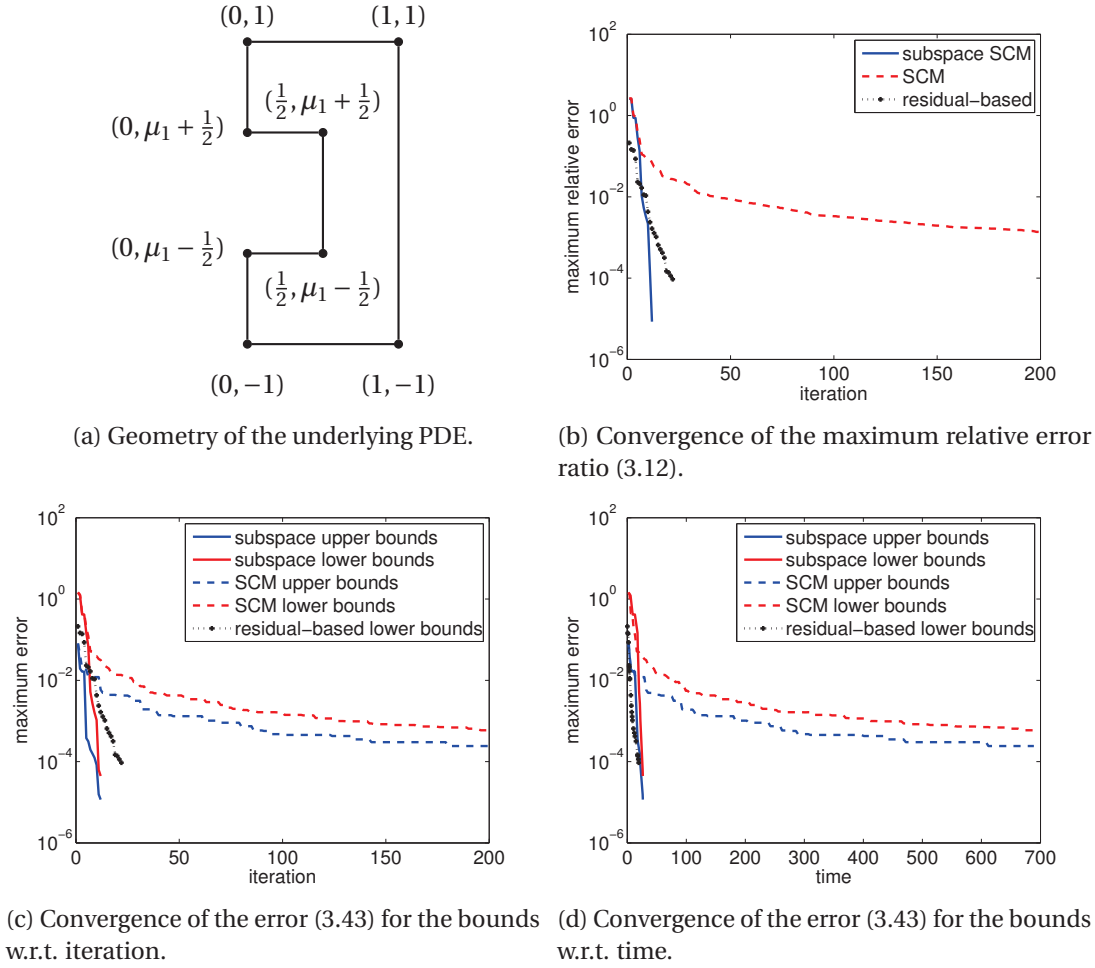
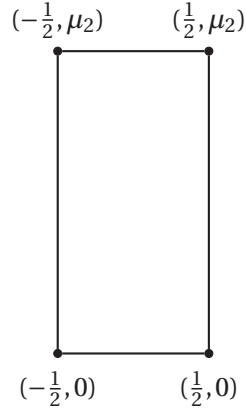


Figure 3.8: Convergence plots for Algorithms 2 and 3 applied to Example 3.22.

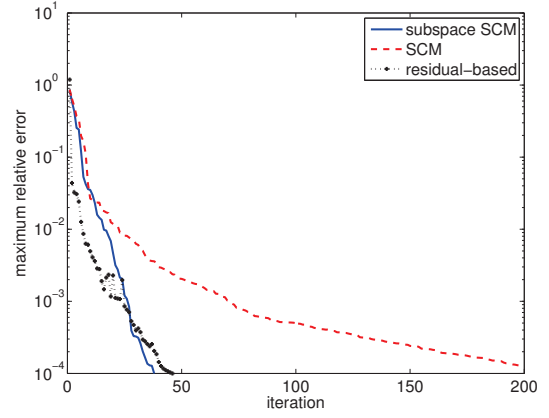
determines a coefficient of the PDE

$$\operatorname{div} \left(\begin{bmatrix} 1 & -\mu_i \\ -\mu_i & 1 \end{bmatrix} \nabla u \right) = 0 \text{ on } \Omega_i, \quad i = 1, \dots, 9.$$

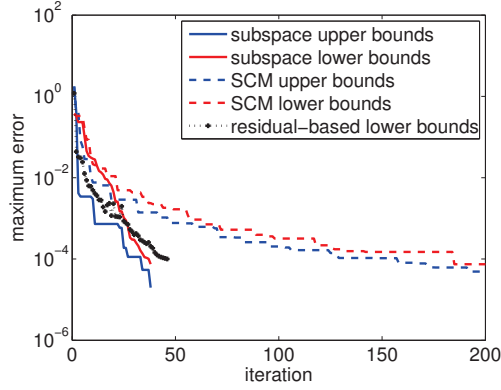
A discretization of the PDE leads to the matrix $A(\mu) = \sum_{i=1}^Q \theta_i(\mu) A_i$, where $Q = 10$, $\mu = (\mu_1, \dots, \mu_9)$ and functions $\theta_i(\mu)$ arising from the parametrization of the PDE coefficients. We choose $n = 1056$ and $D = [0.1, 0.5]^9$. As can be seen in Figure 3.10, the performance of both Algorithms 2 and 3 is not satisfactory, as neither algorithm reaches the desired tolerance, due to the slow convergence of the SCM and subspace lower bounds. However, Algorithm 3 is significantly faster than Algorithm 2 due to the saturation assumption which reduces the number of bound evaluations per iteration. Only the subspace upper bounds converges at a satisfactory rate. In this example, the residual-based lower bounds clearly show their advantage. They become reliable after only 31 iterations.



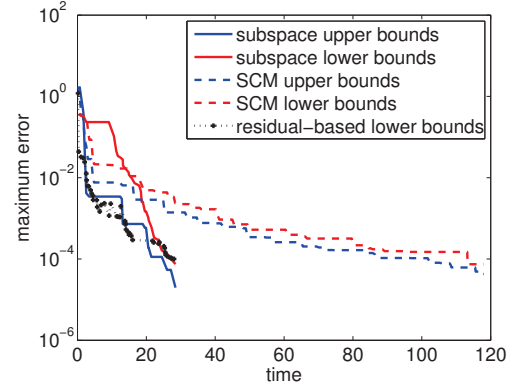
(a) Geometry of the underlying PDE.



(b) Convergence of the maximum relative error ratio (3.12).



(c) Convergence of the error (3.43) for the bounds w.r.t. iteration.



(d) Convergence of the error (3.43) for the bounds w.r.t. time.

Figure 3.9: Convergence plots for Algorithm 2 and 3 applied to Example 3.23.

3.5.3 Estimation of the inf-sup constant

In Section 3.5.2 we have seen that the computation of coercivity constants can be formulated in terms of (3.1). As explained in Remark 2.18, for non-coercive parametrized PDE one may have to resort to the inf-sup constant [HKC⁺10] defined as

$$\beta(\mu) = \inf_{u \in X} \sup_{v \in X} \frac{b(u, v; \mu)}{\|u\|_X \|v\|_X}, \quad (3.46)$$

where $b(\cdot, \cdot, \mu)$ is the bilinear form in the weak formulation of the underlying PDE and X is the accompanying function space with the norm $\|\cdot\|_X$ induced by the scalar product $(\cdot, \cdot)_X$. A finite element discretization of (3.46) leads to the minimization problem

$$\beta^n(\mu) = \inf_{u \in \mathbb{R}^n} \sup_{v \in \mathbb{R}^n} \frac{u^T B(\mu) v}{\sqrt{u^T X u} \sqrt{v^T X v}} = \inf_{x \in \mathbb{R}^n} \sup_{y \in \mathbb{R}^n} \frac{x^T L^{-T} B(\mu) L^{-1} y}{\|x\|_2 \|y\|_2} \quad (3.47)$$

3.5. Applications and numerical examples

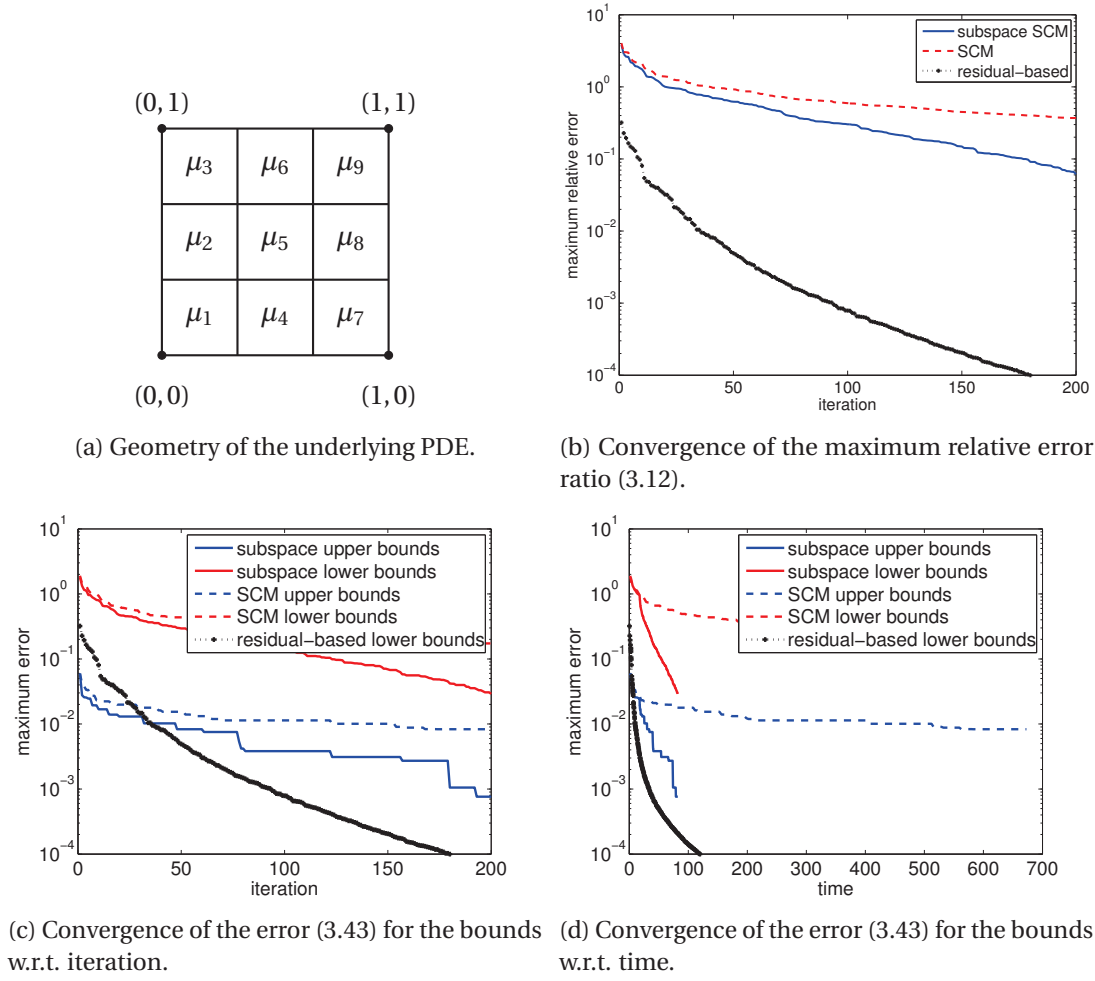


Figure 3.10: Convergence plots for Algorithm 2 and 3 applied to Example 3.24.

where, once again, $B(\mu)$ and $X = LL^T$ are the discretizations of $b(\cdot, \cdot, \mu)$ and $(\cdot, \cdot)_X$, respectively. Minimizing (3.47) is equivalent to solving the singular value problem

$$\sigma_{\min}(L^{-1}B(\mu)L^{-T}),$$

which, in turn, is equivalent to computing

$$\lambda_{\min}(L^{-1}B(\mu)^T X^{-1}B(\mu)L^{-T}), \quad (3.48)$$

since $\sigma_{\min}(B) = \sqrt{\lambda_{\min}(B^T B)}$. The expression (3.48) can be recast in terms of (3.1), with $Q(Q+1)/2$ terms, and with the matrices $A_{i,j}$ and functions $\theta_{ij}(\mu)$ for $1 \leq i < j \leq Q$ defined as

$$\begin{aligned} A_{ij} &= L^{-1}B_i^T X^{-1}B_j L^{-T} + L^{-1}B_j^T X^{-1}B_i L^{-T} \\ \theta_{ij}(\mu) &= \left(1 - \frac{\delta_{ij}}{2}\right) \theta_i(\mu) \theta_j(\mu), \end{aligned}$$

Chapter 3. Low-rank approach for parameter dependent Hermitian eigenvalue problem

where δ_{ij} is the Kronecker delta function. The SCM algorithm has already been applied to (3.48) but only with limited success, since having $Q(Q+1)/2$ terms in the affine decomposition of $A(\mu)$ further increases the computational cost by making the solution of the LP problem (3.11) significantly harder. The faster convergence of the subspace-accelerated approach to (3.48) mitigates this cost to a certain extent.

An illustration of this idea can be seen in the following numerical example, where we apply both Algorithm 2 and 3 to computation of the inf-sup constants of a convection-diffusion operator and compare their respective performances.

Example 3.25. We consider an example from [HKC⁺ 10] concerning a convection-diffusion problem on the unit square $\Omega = [0, 1]^2$ with homogeneous Dirichlet boundary conditions on $\partial\Omega$, with the parameter-dependent bilinear form $b(u, v; \mu)$ defined as follows

$$b(u, v; \mu) = \mu_1 \int_{\Omega} \nabla u \nabla v + \mu_2 \int_{\Omega} x_1 \frac{\partial u}{\partial x_1} v - \int_{\Omega} x_2 \frac{\partial u}{\partial x_2} v, \quad \forall u, v \in X \equiv H_0^1(\Omega) \quad (3.49)$$

and parameter $\mu = (\mu_1, \mu_2)$ inside the parameter domain $D = [0.1, 1] \times [1, 5]$.

Similarly as explained in Section 2.3 we consider a "natural-norm" on X induced by the scalar product

$$(u, v)_X = \int_{\Omega} \nabla u \nabla v + \tau \int_{\Omega} uv,$$

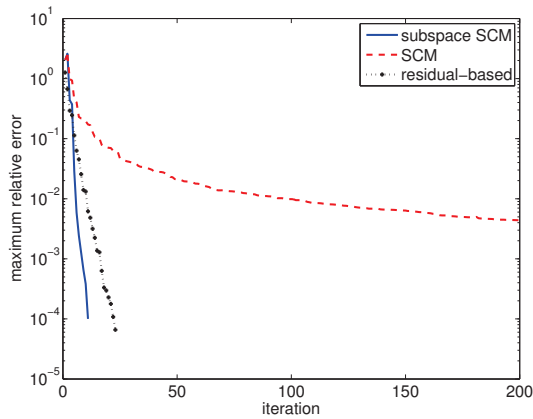
with

$$\tau = \min_{u \in X} \frac{\int_{\Omega} \nabla u \nabla u}{\int_{\Omega} uu}.$$

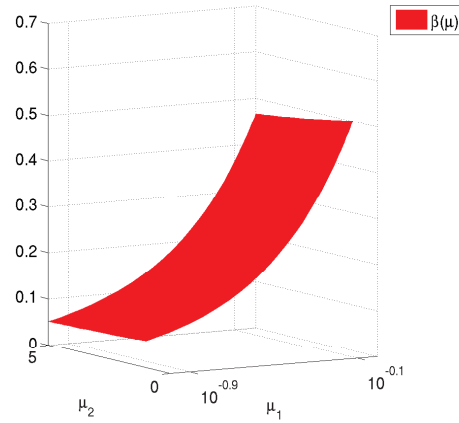
We consider a finite element discretization of $b(u, v; \mu)$ on Ω with $n = 1009$ degrees of freedom, which yields

$$B(\mu) = \mu_1 B_1 + \mu_2 B_2 - B_3,$$

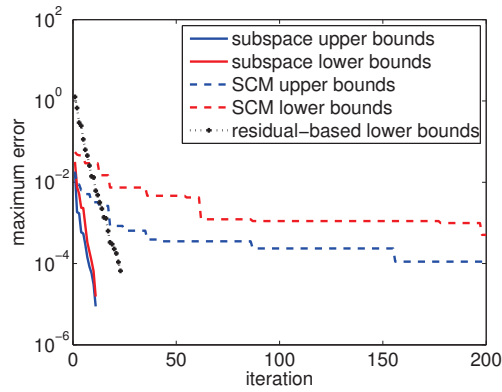
where B_1, B_2, B_3 are the discretizations each of the corresponding integrals in (3.49). Using the procedure explained above, we obtain $A(\mu) = \sum_{q=1}^Q \theta_q(\mu) A_q$ with $Q = 6$ such that $A(\mu) = B(\mu)^T B(\mu)$ and $\lambda_{\min}(A(\mu)) = \sigma_{\min}(B(\mu))^2$. The performances of both algorithms applied to computing $\lambda_{\min}(A(\mu))$ are shown in Figure 3.11, with Figure 3.11b showing the $\beta(\mu)$ surface plot on a 32×32 regular grid on D . Similarly as in the experiments presented in Section 3.5.2, Algorithm 3 converges in only 10 iterations, while Algorithm 2 fails to attain the desired tolerance in 200 iterations.



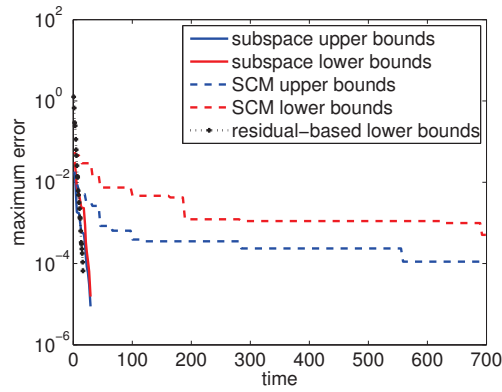
(a) Convergence of the maximum relative error ratio (3.12).



(b) Surface plot of $\beta(\mu)$.



(c) Convergence of the error (3.43) for the bounds w.r.t. iteration.



(d) Convergence of the error (3.43) for the bounds w.r.t. time.

Figure 3.11: Convergence plots for Algorithm 2 and 3 applied to Example 3.25.

3.6 Conclusion

We have proposed a new subspace-accelerated approach, given in Algorithm 3, for solving parameter-dependent Hermitian eigenvalue problem. It builds upon the most commonly used existing approach, SCM, and improves on it by implicitly exploiting regularity in the smallest eigenvectors.

We have shown that the subspace acceleration in Algorithm 3 leads to improved interpolation properties in comparison to SCM, with both subspace bounds interpolating the derivatives of $\lambda_{\min}(A(\mu))$, which has not been the case for the SCM bounds. Moreover, for $A(\mu)$ analytic and $d = 1$, the presented results show that we can expect exponential convergence of Algorithm 3 on intervals where $\lambda_{\min}(A(\mu))$ stays simple. In addition, we have demonstrated that the subspace bounds can be efficiently computed at a per iteration computational cost which is only slightly larger than in SCM.

Chapter 3. Low-rank approach for parameter dependent Hermitian eigenvalue problem

Furthermore, we have shown that the better theoretical properties of Algorithm 3 carry over to numerical experiments. More specifically, when applied to estimation of the coercivity constant, Algorithm 3 presents a significant improvement over SCM, both in terms of iterations and the total computational time, on a number of numerical examples from the literature. Moreover, the proposed approach can be extended to the solution of parameter-dependent singular value problems, as demonstrated in Example 3.25, where it is applied to estimation of the inf-sup constant.

We have observed that for problems with small gaps between the smallest eigenvalues and a large variation in the parameter space, as in Example 3.24, the convergence of the subspace lower bounds may still not be satisfactory. For such cases, we propose a heuristic approach using residual-based lower bounds.

4 Low-rank approach to pseudospectra computation

Let $A \in \mathbb{C}^{n \times n}$ be a non-normal matrix and $\varepsilon > 0$. Effects of perturbations on the spectrum of A

$$\lambda(A) = \{z \in \mathbb{C} : \|(zI - A)^{-1}\|_2 = \infty\}$$

can be studied by computing the so-called ε -pseudospectra:

$$\sigma_\varepsilon(A) := \{z \in \mathbb{C} : \|(zI - A)^{-1}\|_2 > \varepsilon^{-1}\},$$

which can also be seen as sublevel sets of the function

$$g(z) = \|(zI - A)^{-1}\|_2 = \sigma_{\min}(zI - A).$$

By evaluating $g(z)$ on a domain of interest $D \subset \mathbb{C}$ we obtain $\sigma_\varepsilon(A) \cap D$ for all $\varepsilon > 0$. In this chapter, we consider a large-scale setting, where evaluating $g(z)$ exactly using the standard techniques is computationally feasible only for a few values of $z \in \mathbb{C}$. Thus, our goal is to compute an approximation $\tilde{g}(z) \approx g(z)$ on the whole domain D using only few exact computations of $\sigma_{\min}(zI - A)$.

An example of how pseudospectral images look like can be seen in Figure 4.1. Using a coarse grid, as in Figure 4.1a, usually does not capture the full variation of $\|(zI - A)^{-1}\|_2$, making the use of a finer grid, as in Figure 4.1b, necessary. However, as the exact computation of the presented resolvent norms on this finer grid takes approximately 10 hours, there is a clear need for a computationally more efficient way to compute ε -pseudospectra.

Applications of pseudospectra and pseudospectral images include linearized stability analysis in fluid mechanics [Sch07], the convergence analysis and design of iterative methods [BES05, TE05], the asymptotic behavior of matrix functions [Hig08, TE05] and operator theory [BG05, Dav07, DP04, Han08, Tre08]. By definition, pseudospectra can also be used to quantify the effects of perturbations and uncertainties on computed eigenvalues and eigenvectors. A more detailed overview of pseudospectra applications can be found in [TE05].

For pseudospectra computation of a large matrix A , the projection-based approach has been

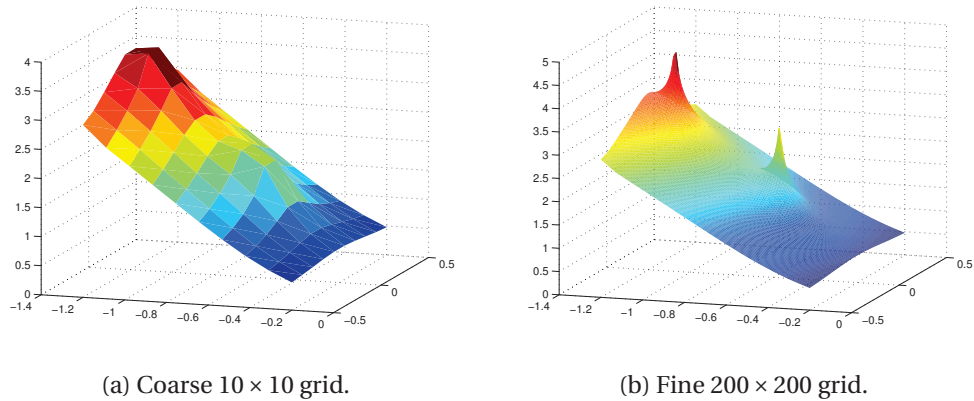


Figure 4.1: Resolvent norms $\log_{10} \|(zI - A)^{-1}\|_2$, for Example 4.11 with $A \in \mathbb{R}^{9512 \times 9512}$, evaluated on $D = [-1.2, -0.2] + [-0.5, 0.5]i$ using the grid-based approach on a rough grid (left) and on a fine grid (right).

proposed, where given a subspace $\mathcal{U} \subset \mathbb{C}^n$ and its orthonormal basis $U \in \mathbb{R}^{n \times k}$, we have the following inclusion

$$\sigma_\varepsilon(A) \supset \sigma_\varepsilon(U, AU),$$

with $\sigma_\varepsilon(G, H) = \{z \in \mathbb{C} : \sigma_{\min}(zG - H) < \varepsilon\}$, for $G, H \in \mathbb{C}^{n \times k}$. Existing choices for the subspace \mathcal{U} include Krylov subspaces [TT96, WT01, SG98] or an invariant subspace containing eigenvectors belonging to a few eigenvalues in or close to the region of interest [RSH93, GS98]. As we will see later, in Section 3.5, both approaches often suffer from slow convergence and lack of means to quantify the obtained accuracy. However, the projection-based approaches have been successfully applied to computation of pseudospectral quantities [KV14, MMMVB15], providing a significant improvement over the previous work [BLO03, GO11].

In this chapter, we propose a new projection-based approach inspired by the subspace-acceleration strategy used in Section 3.2. It is primarily designed to provide highly accurate approximation to ε -pseudospectra in isolated parts of spectrum, that is, regions in the complex plane containing only a few eigenvalues of A . As described in the previous paragraph, given a carefully chosen orthonormal matrix $V \in \mathbb{R}^{n \times k}$, $k \ll n$, $\sigma_{\min}(zV - AV)$ can be used to reconstruct $g(z)$. As will be described in Section 4.1.1, after a preprocessing step, $\sigma_{\min}(zV - AV)$ can be computed in $\mathcal{O}(k^3)$ operations for any $z \in \mathbb{C}$. If $v(z) \in V$, where $v(z)$ is the right singular vector corresponding to $\sigma_{\min}(zI - A)$, then $\sigma_{\min}(zI - A) = \sigma_{\min}(zV - AV)$ and the reconstruction is exact. Clearly, one cannot expect that $v(z) \in \text{span}(V)$ for all $z \in D$, but the goal is to find such V which contains good approximations to $v(z)$ for all $z \in D$.

Recasting $\sigma_{\min}((x + iy)I - A)$, with $z = x + iy$, into a parameter-dependent Hermitian eigenvalue problem, allows us to efficiently obtain such V using the subspace accelerated version of SCM presented in Algorithm 3. Despite the dependence on just two real parameters x and y , the problem remains very challenging due to the need for high absolute accuracy, required in order to attain reasonably good relative accuracy in the vicinity of the eigenvalues.

Moreover, the particular structure of the problem allows for additional improvements, such as incorporating the invariant subspace approach for obtaining a good *a priori* basis and an optimized computation procedure for the lower bounds.

The rest of this chapter is largely based on [Sir16] and is organized as follows. In Section 4.1, we first give an brief overview of existing approaches for pseudospectra computation, in particular the projection-based approaches and discuss the use of two-sided projections. We present our new projection-based approach to pseudospectra computation in Section 4.2. Additionally, we discuss the choice of error estimates, the sampling strategy and the interpolation properties. In Section 4.3, we present the full algorithm together with the complexity analysis, and discuss its efficient implementation, while in Section 4.4, we present a few numerical experiments showing the performance of the proposed approach in comparison to some of the existing methods.

4.1 Existing approaches

In this section, we present short overview of the existing approaches that are commonly used for pseudospectra computation.

A straightforward way to compute pseudospectra and pseudospectral images is using a grid-based approach, where $g(z)$ is computed for a finitely many points z_1, \dots, z_m , typically arranged on a uniformly spaced rectangular grid, requiring $\mathcal{O}(mn^3)$ operations. EigTool [Wri02], the most commonly used software for pseudospectra computation, also uses a grid-based approach. The approach can be made more efficient if a Schur decomposition $A = QTQ^*$ is available, since

$$\sigma_{\min}(zI - A) = \sigma_{\min}(Q(zI - T)Q^*) = \sigma_{\min}(zI - T),$$

where computation of $\sigma_{\min}(zI - T)$ using the inverse Lanczos method requires solution of just two triangular linear system per iteration, resulting in total complexity of $\mathcal{O}(n^3 + mn^2)$ operations [Lui97, Tre99]. However, as T is in general dense, this approach remains computationally infeasible for large values of n due to memory requirements.

For large-scale sparse A , grid-based approach can be made more efficient by using a sparse LU decomposition [BH96, Dav06] of $zI - A$ together with the inverse Lanczos method, which is often faster than computing the full singular value decomposition. Moreover, in case when $\sigma_{\varepsilon}(A)$ is computed only for fixed value of ε , path following techniques may be used, typically requiring fewer evaluations of $g(z)$ than a grid-based approach [BG01, Brü96, MP02]. In addition to the computation-oriented approaches, the asymptotic behavior of ε -pseudospectra has been studied in [GMM⁺15], while *a priori* estimates for pseudospectra using first-order approximations have been derived in [Han15].

4.1.1 Projection-based approaches

As previously mentioned, for large matrices A , projection-based approaches have been proposed. Given a subspace $\mathcal{U} \subset \mathbb{C}^n$, and its orthonormal basis $U \in \mathbb{R}^{N \times k}$, we have the following inequality

$$\sigma_{\min}(zI - A) = \min_{\substack{u \in \mathbb{C}^n \\ \|u\|_2=1}} \|(zI - A)u\|_2 \leq \min_{\substack{u \in \mathcal{U} \\ \|u\|_2=1}} \|(zI - A)u\|_2 = \sigma_{\min}(zU - AU)$$

and the inclusion

$$\sigma_{\varepsilon}(U, AU) \subset \sigma_{\varepsilon}(A),$$

with $\sigma_{\varepsilon}(G, H) = \{z \in \mathbb{C} : \sigma_{\min}(zG - H) < \varepsilon\}$, for $G, H \in \mathbb{C}^{N \times k}$. For a good choice of \mathcal{U} , $\sigma_{\varepsilon}(U, AU)$ may offer a surprisingly accurate approximation to $\sigma_{\varepsilon}(A)$, while being significantly cheaper to compute. After a preprocessing step in which the QR decomposition of the matrix $[U, AU] \in \mathbb{R}^{n \times 2k}$ is computed, for any $i \in \{1, \dots, m\}$, $\sigma_{\min}(z_i U - AU)$ can be computed by solving the following $2m \times m$ small singular value problem:

$$\begin{aligned} \sigma_{\min}(z_i U - AU) &= \sigma_{\min}([U, AU][z_i I_k, -I_k]^T) = \sigma_{\min}(QR[z_i I_k, -I_k]^T) \\ &= \sigma_{\min}(R[z_i I_k, -I_k]^T), \end{aligned}$$

where I_k denotes the $k \times k$ identity matrix, resulting in total complexity of $\mathcal{O}(nk^2 + mk^3)$.

Existing choices for the subspace \mathcal{U} include Krylov subspaces [TT96, WT01, SG98]

$$\mathcal{U} = \mathcal{K}_k(A, b) = \text{span}\{b, Ab, A^2 b, \dots, A^{k-1} b\},$$

with the starting vector $b \in \mathbb{R}^n$, which usually provides a good approximation in the outer parts of the spectrum, and invariant subspaces of A containing eigenvectors belonging to a few eigenvalues in or close to the region of interest [RSH93, GS98], which usually provides a good approximation in isolated parts of the spectrum.

Let $z \in \mathbb{C}$ and let us denote with $v(z)$ the right singular vector corresponding to $\sigma_{\min}(zI - A)$. Then the error of a projection-based approach can be bounded in the following way

$$\begin{aligned} \sigma_{\min}(zI - A) &\leq \sigma_{\min}(zU - AU) = \min_{u \in \mathcal{U}} \|(A - zI)v(z) + (A - zI)(u - v(z))\|_2 \\ &\leq \sigma_{\min}(zI - A) + \delta \|zI - A\|_2, \end{aligned}$$

where $\delta := \text{dist}(v(z), \mathcal{U}) = \min_{u \in \mathcal{U}} \|v(z) - u\|_2$. Unfortunately, $\delta \|zI - A\|_2$ is not practically useful as an error estimate, since it involves $v(z)$, a quantity that is available only by solving the full-size singular value problem, which is exactly what we are trying to avoid. In fact, it is not *a priori* clear why δ should (rapidly) decrease by increasing the size of either the

Krylov or an invariant subspace. This lack of means to quantify the obtained accuracy and the frequently observed slow convergence are two main disadvantages of existing projection-based approaches, as we will see later in Section 4.4.

Two-sided projections

The projection-based approaches we have discussed above use one-sided projections on the subspace \mathcal{U} , which rely on accurately approximating the right singular vector $v(z)$ of $\sigma_{\min}(zI - A)$ to provide reliable pseudospectra estimates. Analogous approaches can be designed based on approximating the left singular vectors corresponding to $\sigma_{\min}(zI - A)$. A question which naturally arises is whether it is possible to combine these approaches and approximate $\sigma_{\min}(zI - A)$ using $\sigma_{\min}(V^*(zI - A)U)$, where U and V are orthonormal bases of the subspaces \mathcal{U} and \mathcal{V} , respectively. Unfortunately, it turns out that this approach is not stable as shown in Example 4.1, i.e. we do not necessarily reconstruct $\sigma_{\min}(A - zI)$ exactly even when \mathcal{U} and \mathcal{V} contain the exact smallest singular vectors of $zI - A$.

Example 4.1. Let $z = 1$, A be the Landau matrix from Example 4.10 with $n = 200$, and u and v the smallest right and left singular vector of $I - A$, respectively. We consider a uniformly spaced grid on $[0.95, 1.05] \times [-0.05, 0.05]$ with seven points in each direction: $\Xi = \{(x_1, y_1), (x_2, y_2), \dots, (x_{25}, y_{25}) = (1, 0), \dots, (x_{49}, y_{49})\}$. Let v_k and u_k be the smallest left and right singular vectors of $(x_k + iy_k)I - A$, respectively. For each $k = 1, \dots, 49$, we construct the subspaces \mathcal{U}_k and \mathcal{V}_k :

$$\mathcal{U}_k = \{u_1, \dots, u_k\}, \quad \mathcal{V}_k = \{v_1, \dots, v_k\},$$

and their orthonormal bases U_k and V_k , respectively, and compute the following quantities:

$$\sigma_{\min}((I - A)U_k), \quad \sigma_{\min}(V_k^*(I - A)), \quad \sigma_{\min}(V_k^*(I - A)U_k), \quad \sigma_{\min}(W_k^*(I - A)U_k)$$

where $W_k = [V_k, E_k] \in \mathbb{R}^{n \times (k+5)}$, with $E_k \in \mathbb{R}^{n \times 5}$ random orthonormal matrix orthogonal to V_k . As expected, $\sigma_{\min}((I - A)U_k)$ and $\sigma_{\min}(V_k^*(I - A))$ converge to $\sigma_{\min}(I - A)$, as the subspaces \mathcal{U}_k and \mathcal{V}_k contain more accurate approximation of u and v , respectively. On the contrary, $\sigma_{\min}(V_k^*(I - A)U_k)$ does not necessarily converge to $\sigma_{\min}(I - A)$, as the error is non-zero (as indicated by the spikes in Figure 4.2), even when $u \in \mathcal{U}_k$ and $v \in \mathcal{V}_k$. This phenomenon can be explained by looking at the variational characterization of the smallest singular values

$$\begin{aligned} \sigma_{\min}(I - A) &= \inf_{u \in \mathbb{R}^n \setminus \{0\}} \sup_{v \in \mathbb{R}^n \setminus \{0\}} \frac{|v^*(I - A)u|}{\|u\| \|v\|} \\ \sigma_{\min}(V_k^*(I - A)U_k) &= \inf_{u \in \mathbb{R}^k \setminus \{0\}} \sup_{v \in \mathbb{R}^k \setminus \{0\}} \frac{|v^* V_k^*(I - A)U_k u|}{\|u\| \|v\|} = \inf_{u \in \mathcal{U}_k \setminus \{0\}} \sup_{v \in \mathcal{V}_k \setminus \{0\}} \frac{|v^*(I - A)u|}{\|u\| \|v\|}, \end{aligned}$$

where we see that by multiplying $I - A$ with both U_k and V_k , we also restrict the majorization domain, and, thus, $\sigma_{\min}(V_k^*(I - A)U_k)$ can end up being much smaller than $\sigma_{\min}(I - A)$ for

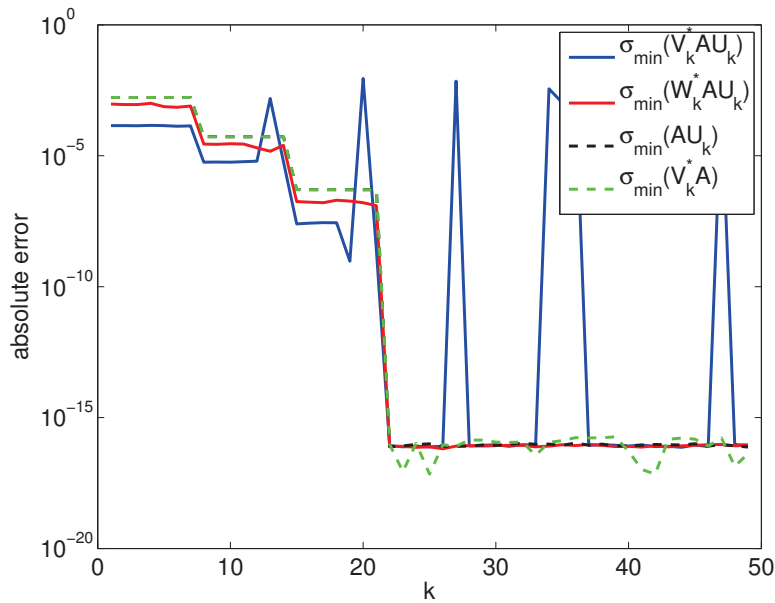


Figure 4.2: Comparison of one-sided versus two-sided projection approach.

some values of k . In certain situations this issue can be resolved by enriching one of the subspaces with a random subspace, as we see that $\sigma_{\min}(W_k^*(I-A)U_k)$ converges to $\sigma_{\min}(I-A)$. However, the use of this idea remains questionable as it is not a priori clear how big E_k should be chosen to guarantee convergence.

Remark 4.2. It can be shown that using two-sided projections for approximating the largest singular value does not suffer from the same loss of stability demonstrated above for the smallest singular value. The largest singular value $\sigma_{\max}(A)$ is the solution of the following maximization problem:

$$\sigma_{\max}(A) = \max_{(u,v) \in (\mathbb{C}^N \setminus \{0\})^2} \frac{|v^* Au|}{\|v\|_2 \|u\|_2}.$$

By restricting u and v to subspaces \mathcal{U} and \mathcal{V} , respectively, we obtain the following inequality:

$$\sigma_{\max}(V^* AU) \leq \sigma_{\max}(A), \quad (4.1)$$

where U and V are corresponding orthonormal bases of \mathcal{U} and \mathcal{V} . Clearly, equality in (4.1) is attained if and only if the subspaces \mathcal{U} and \mathcal{V} contain the dominant left and right singular vectors of A , making this approach stable.

4.2 Subspace acceleration

In this section, we present our new projection-based approach for pseudospectra computation on a domain of interest $D \subset \mathbb{C}$. It is largely based upon the subspace-accelerated approach for parameter-dependent Hermitian eigenvalue problems described in Chapter 3. Without

loss of generality, we assume that D is a rectangle $D = [a, b] + [c, d]i \subset \mathbb{C}$ in the complex plane. Similarly as with parameter-dependent eigenvalues in Chapter 3, assessing the resolvent norms on the whole continuous domain D is computationally infeasible, we follow standard practice in pseudospectra computation [TE05] and substitute D by a finite, but rather fine, uniformly spaced grid $\Xi \subset D$.

For $z = x + iy \in \mathbb{C}$, the computation of $g(z)$ can be viewed as a Hermitian eigenvalue problem depending on the two real parameters x and y :

$$\begin{aligned} g(x + iy)^2 &= \lambda_{\min} \left(((x + yi)I - A)^* ((x + yi)I - A) \right) \\ &= \lambda_{\min} \left(A^* A - x(A + A^*) - yi(A^* - A) + (x^2 + y^2)I \right) \\ &= \lambda_{\min} \left(\widehat{A}(x, y) \right) + x^2 + y^2, \end{aligned} \tag{4.2}$$

where $\widehat{A}(x, y) = A^* A - x(A + A^*) - yi(A^* - A)$. Note that each of the matrices $A^* A$, $A + A^*$ and $i(A^* - A)$ is Hermitian.

Since $\widehat{A}(x, y)$ admits an affine linear decomposition w.r.t. (x, y) , we can use Algorithm 3, and by sampling ℓ smallest eigenpairs of $\widehat{A}(x, y)$ for each (x, y) in the sample set \mathcal{S} , compute both an upper bound $\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell)$ and a lower bound $\lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell)$ for $\lambda_{\min}(\widehat{A}(x, y))$,

Given $\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell)$ and $\lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell)$, (4.2) allows us to bound $\sigma_{\min}(zI - A)$ in the following way:

$$\lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell) + x^2 + y^2 \leq \sigma_{\min}^2(zI - A) \leq \lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell) + x^2 + y^2.$$

By taking the square root, the upper bound $\sigma_{\text{SUB}}(x, y; \mathcal{S}, \ell)$ for $\sigma_{\min}(zI - A)$ can now be defined as

$$\sigma_{\text{SUB}}(x, y; \mathcal{S}, \ell) := \sqrt{\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell) + x^2 + y^2}, \tag{4.3}$$

while the lower bound $\sigma_{\text{SLB}}(x, y; \mathcal{S}, \ell)$ is defined by

$$\sigma_{\text{SLB}}(x, y; \mathcal{S}, \ell) = \sqrt{\max(\lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell) + x^2 + y^2, 0)}, \tag{4.4}$$

keeping in mind the non-negativity of the singular values.

Computation of the bounds $\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell)$ and $\lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell)$, and the choice of appropriate error estimates for driving the sampling procedure is explained in more detail in the following Sections 4.2.1 and 4.2.2, respectively.

4.2.1 Computation of $\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell)$ and $\lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell)$

In the following, we adapt the subspace-accelerated approach from Section 3.2 to the computation of pseudospectra bounds.

Chapter 4. Low-rank approach to pseudospectra computation

Given a sample set $\mathcal{S} = \{(x_1, y_1), \dots, (x_M, y_M)\} \subset D$, suppose that we have computed the $\ell \geq 1$ smallest eigenvalues for each sample $(x_i, y_i) \in \mathcal{S}$:

$$\lambda_i = \lambda_i^{(1)} \leq \lambda_i^{(2)} \leq \dots \leq \lambda_i^{(\ell)}$$

of $\hat{A}(x_i, y_i)$ along with an orthonormal basis of associated eigenvectors $v_i^{(1)}, v_i^{(2)}, \dots, v_i^{(\ell)} \in \mathbb{C}^n$. By collecting these eigenvectors into a subspace

$$\mathcal{V}(\mathcal{S}, \ell) := \text{span}\{v_1^{(1)}, \dots, v_1^{(\ell)}, v_2^{(1)}, \dots, v_2^{(\ell)}, \dots, v_M^{(1)}, \dots, v_M^{(\ell)}\}$$

allows us to use Algorithm 3, presented in Section 3.2, to compute the subspace upper bound $\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell)$. By solving the following $M\ell \times M\ell$ eigenvalue problem

$$V^* \hat{A}(x, y) V w = \lambda_{\mathcal{V}} w,$$

where V denotes an orthonormal basis for $\mathcal{V}(\mathcal{S}, \ell)$, we obtain the smallest $r \leq M\ell$ eigenvalues

$$\lambda_{\mathcal{V}}^{(1)} \leq \lambda_{\mathcal{V}}^{(2)} \leq \dots \leq \lambda_{\mathcal{V}}^{(r)}$$

and the corresponding eigenvectors $w_1, \dots, w_r \in \mathbb{C}^{M\ell}$. By the eigenvalue interlacing property we have

$$\lambda_{\min}(\hat{A}(x, y)) \leq \lambda_{\mathcal{V}}^{(1)},$$

which allows us to define the subspace upper bound for $\lambda_{\min}(\hat{A}(x, y))$ as:

$$\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell) := \lambda_{\mathcal{V}}^{(1)}.$$

In terms of lower bounds for $\lambda_{\min}(\hat{A}(x, y))$, by solving the linear program (3.11), we can easily compute the SCM lower bound $\lambda_{\text{LB}}(x, y; \mathcal{S})$. However, in practice, this lower bound is not always a very accurate approximation to $\lambda_{\min}(\hat{A}(x, y))$. By additionally exploiting the structure in U and gaps among the sampled smallest eigenpairs of $\hat{A}(x, y)$ in \mathcal{S} , as explained in Section 3.2.2, we can calculate a lower bound $\eta(x, y)$ for Ritz values of $\hat{A}(x, y)$ on U_{\perp}

$$\eta(x, y) \leq \lambda_{\min}(U_{\perp}^* \hat{A}(x, y) U_{\perp}),$$

where $U, U_{\perp} \in \mathbb{C}^{n \times r}$ are orthonormal bases for $\{w_1, \dots, w_r\}$ and its orthogonal complement, respectively. As before, $\eta(x, y)$ can be computed by simply solving a linear program similar to (3.11) with updated right-hand side of the constraints, which, in this case, requires solving just one 3×3 linear system. Following the procedure in Section 3.2, combining the Ritz values of $\hat{A}(x, y)$ and $\eta(x, y)$, and using the quadratic residual perturbation bounds from Theorem 2.11, allows us to define the subspace lower bound for $\lambda_{\min}(\hat{A}(x, y))$:

$$\lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell) := \min(\lambda_{\mathcal{V}}^{(1)}, \eta(x, y)) - \frac{2\rho^2}{|\lambda_{\mathcal{V}}^{(1)} - \eta(x, y)| + \sqrt{|\lambda_{\mathcal{V}}^{(1)} - \eta(x, y)|^2 + 4\rho^2}}, \quad (4.5)$$

with the residual norm $\rho = \|U_{\perp}^* \widehat{A}(x, y) U\|_2 = \|\widehat{A}(x, y) U - U(U^* \widehat{A}(x, y) U)\|_2$.

Remark 4.3. *First, it is worth noting that the smallest eigenvectors of $\widehat{A}(x, y)$ coincide with the right singular vectors corresponding to $\sigma_{\min}(zI - A)$. Secondly, our subspace-accelerated approach for computing upper bounds $\sigma_{\text{SUB}}(x, y; \mathcal{S}, \ell)$ can be seen as a special case of the general projection-based approach for the choice of $\mathcal{U} = \mathcal{V}(\mathcal{S}, \ell)$:*

$$\begin{aligned} \min_{u \in \mathcal{V}(\mathcal{S}, \ell)} \|(zI - A)u\|_2 &= \sigma_{\min}((zI - A)V) \\ &= \sqrt{\lambda_{\min}(V^*(zI - A)^*(zI - A)V)} \\ &= \sqrt{\lambda_{\min}(V^* \widehat{A}(x, y) V) + x^2 + y^2} = \sigma_{\text{SUB}}(x, y; \mathcal{S}, \ell), \end{aligned}$$

with $z = x + iy$. In fact, in the invariant subspace approach we construct the subspace by sampling the right singular vector corresponding to $\sigma_{\min}(zI - A)$ for $z \in \lambda(A)$ (they coincide with the eigenvectors for the corresponding $z \in \lambda(A)$), while in our approach we generalize this idea by allowing, both, sampling of the smallest singular vectors for $z \notin \lambda(A)$ as well as sampling of more than one smallest singular vector per sampling point.

Bounding box

As explained in Section 3.2.2, to compute of $\eta(x, y)$ we first need to solve (3.11) and compute $\lambda_{\text{LB}}(x, y; \mathcal{S})$. In this specific setting, we would need to solve the following linear program

$$\begin{aligned} \lambda_{\text{LB}}(x, y; \mathcal{S}) &:= \min_{d \in \mathbb{R}^3} [1, x, y]^T d \\ \text{s.t.} \quad [1, x_i, y_i]^T d &\geq \lambda_i^{(1)}, \quad i = 1, \dots, M \\ d &\in \mathcal{B}, \end{aligned} \tag{4.6}$$

where

$$\begin{aligned} \mathcal{B} &= [\lambda_{\min}(A^* A), \lambda_{\max}(A^* A)] \times [\lambda_{\min}(A + A^*), \lambda_{\max}(A + A^*)] \\ &\quad \times [\lambda_{\min}(i(A^* - A)), \lambda_{\max}(i(A^* - A))]. \end{aligned} \tag{4.7}$$

As explained in Section 3.1.2, the role of \mathcal{B} in (4.6) is to ensure that the solution is finite. However, for the examples considered in Section 4.4, matrices $A^* A$, $A + A^*$ and $i(A^* - A)$ often have very small relative gaps between the extremal eigenvalues and the rest of the spectrum, making the eigenproblems in (4.7) very hard to solve. Moreover, for examples with a mass matrix, such as Example 4.11, computation of the extremal eigenvalues of the matrix $\frac{M^{-1}A + A^T M^{-T}}{2}$ requires inverting a large-scale matrix M , as well as solving a large-scale dense eigenvalue problem, which is often computationally infeasible.

Yet, in this specific application, since $\widehat{A}(x, y)$ is affine in (x, y) and d is only 2, we can avoid computation of \mathcal{B} . Instead, as explained in Remark 3.2, we can *a priori* insert vertices of D

into \mathcal{S} , which requires computation of the ℓ smallest eigenpairs for matrices

$$\widehat{A}(a, c), \widehat{A}(a, d), \widehat{A}(b, c), \widehat{A}(b, d).$$

This modification both reduces the number of full-size eigenvalue problems that need to be solved (\mathcal{B} does not need to be computed anymore) as well as improves the accuracy of the computed SCM lower bounds $\lambda_{\text{LB}}(x, y; \mathcal{S})$.

4.2.2 Error estimates and sampling

As described in the previous section, by sampling the smallest eigenpairs of $\lambda_{\min}(\widehat{A}(x, y))$ on a set of samples \mathcal{S} , we can compute an upper and a lower bound for $\lambda_{\min}(\widehat{A}(x, y))$ on the whole domain D . In our approach, we use a greedy sampling strategy, adding in each iteration to \mathcal{S} a point from Ξ with the largest error estimate. Similarly as in Section 3.2, for $z = x + iy \in D$, given $\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell)$ and $\lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell)$, we define the error estimate $\Delta(x, y; \mathcal{S}, \ell)$, for the Hermitian eigenvalue problem $\lambda_{\min}(\widehat{A}(x, y) + (x^2 + y^2)I)$, in the following way:

$$\begin{aligned} \Delta(x, y; \mathcal{S}, \ell) &= \frac{\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell) + x^2 + y^2 - \lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell) - x^2 - y^2}{\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell) + x^2 + y^2} \\ &= \frac{\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell) - \lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell)}{\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell) + x^2 + y^2}. \end{aligned} \quad (4.8)$$

In each iteration, we compute $\Delta(x, y; \mathcal{S}, \ell)$ for all $(x, y) \in \Xi$, and select the one having the largest error estimate as the next parameter sample point.

4.2.3 Interpolation properties

Using the interpolation results from Section 3.2.4 we obtain that the subspace eigenvalue bounds $\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell)$ and $\lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell)$ interpolate the exact values of $\lambda_{\min}(\widehat{A}(x, y))$:

$$\lambda_{\min}(\widehat{A}(x, y)) = \lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell) = \lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell) \quad \forall (x, y) \in \mathcal{S}. \quad (4.9)$$

Additionally, if $\lambda_{\min}(\widehat{A}(x, y))$ is a simple eigenvalue, the subspace bounds also capture the derivatives

$$\nabla \lambda_{\min}(\widehat{A}(x, y)) = \nabla \lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell) = \nabla \lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell) \quad \forall (x, y) \in \mathcal{S}, \quad (4.10)$$

with the gradient ∇ with respect to (x, y) . These interpolation results easily extend to the singular value bounds $\sigma_{\text{SUB}}(x, y; \mathcal{S}, \ell)$ and $\sigma_{\text{SLB}}(x, y; \mathcal{S}, \ell)$ as can be seen from the following theorem.

Theorem 4.4. *For $z = x + iy \in \mathcal{S}$, the singular value bounds $\sigma_{\text{SUB}}(x, y; \mathcal{S}, \ell)$ and $\sigma_{\text{SLB}}(x, y; \mathcal{S}, \ell)$,*

defined in (4.3) and (4.4), respectively, satisfy

$$\sigma_{\min}(zI - A) = \sigma_{\text{SUB}}(x, y; \mathcal{S}, \ell) = \sigma_{\text{SLB}}(x, y; \mathcal{S}, \ell).$$

Additionally, if $\sigma_{\min}(zI - A)$ is simple and positive, then

$$\nabla \sigma_{\min}(zI - A) = \nabla \sigma_{\text{SUB}}(x, y; \mathcal{S}, \ell) = \nabla \sigma_{\text{SLB}}(x, y; \mathcal{S}, \ell).$$

Proof. The first equality follows directly from (4.9) by taking the square root. Since $\sigma_{\min}(zI - A) > 0$, by differentiating (4.2) we get

$$\nabla \sigma_{\min}(zI - A) = \frac{1}{2\sigma_{\min}(zI - A)} \left(\nabla \lambda_{\min}(\widehat{A}(x, y)) + \begin{bmatrix} 2x \\ 2y \end{bmatrix} \right).$$

Simplicity of $\sigma_{\min}(zI - A)$ implies (4.10), which together with the first equality, gives the second equality. \square

Using Theorem 4.4 we can formulate an analogous theorem to Theorem 3.13 and obtain *a priori* error estimates for $\sigma_{\text{SUB}}(x, y; \mathcal{S}, \ell)$ and $\sigma_{\text{SLB}}(x, y; \mathcal{S}, \ell)$.

Theorem 4.5. *Let $z_{\mathcal{S}} = x_{\mathcal{S}} + iy_{\mathcal{S}}$ such that $\sigma_{\min}(z_{\mathcal{S}}I - A)$ is simple and positive and let $h > 0$ such that $\sigma_{\min}(zI - A)$, $\sigma_{\text{SUB}}(x, y; \mathcal{S}, \ell)$ and $\sigma_{\text{SLB}}(x, y; \mathcal{S}, \ell)$ are twice differentiable on $B(z_{\mathcal{S}}, h)$. Then there exist constants $C_1, C_2 > 0$ such that*

$$\begin{aligned} |\sigma_{\text{SUB}}(x, y; \mathcal{S}, \ell) - \sigma_{\min}(zI - A)| &< C_1 h^2 \\ |\sigma_{\text{SLB}}(x, y; \mathcal{S}, \ell) - \sigma_{\min}(zI - A)| &< C_2 h^2, \end{aligned}$$

for all $z = x + iy \in B(z_{\mathcal{S}}, h)$.

Proof. Let $z = x + iy \in B(z_{\mathcal{S}}, h)$. Expanding $\sigma_{\min}(zI - A)$ and $\sigma_{\text{UB}}(x, y; \mathcal{S}, \ell)$ around $z_{\mathcal{S}}$ using a second-order Taylor polynomial expansion and using the results of Theorem 4.4, we obtain

$$\sigma_{\text{UB}}(x, y; \mathcal{S}, \ell) - \sigma_{\min}(zI - A) = \frac{(z - z_{\mathcal{S}})^2}{2} (\nabla^2 \sigma_{\min}(z_1 I - A) - \nabla^2 \sigma_{\text{UB}}(x_2, y_2; \mathcal{S}, \ell)),$$

for $z_1, z_2 = x_2 + iy_2 \in [z_{\mathcal{S}}, z]$. The first inequality now holds for

$$C_1 = \max_{\tilde{z} \in B(z_{\mathcal{S}}, h)} \|\nabla^2 \sigma_{\min}(\tilde{z}I - A)\|_2 + \max_{\tilde{z} = \tilde{x} + i\tilde{y} \in B(z_{\mathcal{S}}, h)} \|\nabla^2 \sigma_{\text{UB}}(\tilde{x}, \tilde{y}; \mathcal{S}, \ell)\|_2.$$

The second inequality can be shown in the same way. \square

Remark 4.6. *As in Section 3.2.5, to ensure the differentiability conditions on $\sigma_{\min}(zI - A)$ and $\sigma_{\text{SUB}}(x, y; \mathcal{S}, \ell)$ needed in the assumptions of Theorem 4.5, it is sufficient that the smallest singular values $\sigma_{\min}(zI - A)$ and $\sigma_{\text{SUB}}(x, y; \mathcal{S}, \ell)$ stay simple and positive on $B(z, h)$, see [KMMM15].*

A simple criterion for differentiability of $\sigma_{\text{SLB}}(x, y; \mathcal{S}, \ell)$ is not available, since (4.5) involves $\eta(x, y)$, which depends on the solution of the linear program $\lambda_{\text{LB}}(x, y; \mathcal{S})$, which is not necessarily smooth around (x_i, y_i) .

Remark 4.7. The requirement for positivity of σ_{min} in Theorems 4.4 and 4.5 is artificial and can be fixed by using the "signed" singular values as in the case of the analytic SVD [BGBMN91].

In practice, since $\hat{A}(x, y)$ is an analytic function in x and y , as already discussed in Section 3.2.4, we can expect much faster convergence than the one guaranteed by Theorem 4.5. Numerical experiments shown in Section 4.4 support this.

Additionally, as $\hat{A}(x, y)$ is an affine linear function w.r.t x and y , we can apply Theorem 3.17 to show that using the subspace lower bounds $\lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell)$ for approximating $\lambda_{\text{min}}(\hat{A}(x, y))$ is always at least as good as linearly interpolating the computed values of $\lambda_{\text{min}}(\hat{A}(x, y))$.

4.3 Algorithm

In this section we present a summary, in form of Algorithm 4, of our subspace-accelerated approach for pseudospectra computation introduced in Section 4.2 and discuss its implementation and computational complexity.

4.3.1 Implementation details

The efficient implementation of Algorithm 4 requires care in order to avoid unnecessary computations. Some implementation details are discussed in the following.

Initialization of the sample \mathcal{S} . As explained in Section 4.2.1, we initialize \mathcal{S} to contain the vertices of the domain D :

$$\mathcal{S} = \{(a, c), (a, d), (b, c), (b, d)\}.$$

For certain problems, it makes sense to *a priori* add additional points from D to \mathcal{S} . To make the error estimates (4.8) sufficiently small, we require high absolute accuracy in regions around the eigenvalues of A . In numerical experiments we observe that \mathcal{S} eventually always contains many points very close to the exact eigenvalues of A . We use this observation, and combine our approach with the invariant subspace approach, to "warm start" the algorithm by inserting eigenvalues of A inside D into the initial sample. In practice, this is usually enough to ensure high absolute accuracy in the proximity of the eigenvalues of A . Such eigenvalues of A can be efficiently computed by simply computing the eigenvalues closest to the centre of D . However, in order not to make the sample \mathcal{S} too large, we limit the number of the exact eigenvalues included in \mathcal{S} to 20 closest to the center of D , unless stated otherwise.

Computation of $\lambda_{\text{min}}(\hat{A}(x, y))$. As can be seen in (4.2), computing the smallest eigenpairs of

Algorithm 4 Subspace acceleration for pseudospectra computation

Input: $A \in \mathbb{C}^{n \times n}$, uniformly spaced grid Ξ on $D = [a, b] + [c, d]i \subset \mathbb{C}$, ℓ . Relative error tolerance ε_{tol} .

Output: Sample set $\mathcal{S} \subset D$ with corresponding eigenvalues $\lambda_i^{(j)}$ and an eigenvector basis V of $\mathcal{V}(\mathcal{S}, \ell)$ such that $\frac{\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell) - \lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell)}{\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell) + x^2 + y^2} < \varepsilon_{\text{tol}}$ for every $(x, y) \in \Xi$.

- 1: Initialize the sample set $\mathcal{S} = \{(a, c), (a, d), (b, c), (b, d)\} \cup (\lambda(A) \cap D)$.
- 2: Compute the ℓ smallest eigenpairs of $\hat{A}(x, y)$, for all $(x, y) \in \mathcal{S}$.
- 3: Compute an orthonormal basis V for $\mathcal{V}(\mathcal{S}, \ell)$, matrices in the affine linear expansion of $V^* \hat{A}(x, y) V$ and R .
- 4: Compute $\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell)$ and $\lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell)$ for all $(x, y) \in \Xi$.
- 5: $(x_{\text{max}}, y_{\text{max}}) \leftarrow \arg \max_{(x, y) \in \Xi} \Delta(x, y; \mathcal{S}, \ell)$.
- 6: **while** $\Delta(x_{\text{max}}, y_{\text{max}}; \mathcal{S}, \ell) > \varepsilon_{\text{tol}}$ **do**
- 7: $\mathcal{S} \leftarrow \mathcal{S} \cup \{(x_{\text{max}}, y_{\text{max}})\}$.
- 8: Compute the ℓ smallest eigenpairs of $\hat{A}(x_{\text{max}}, y_{\text{max}})$.
- 9: Update the orthonormal basis V for $\mathcal{V}(\mathcal{S}, \ell)$, matrices in the affine linear expansion of $V^* \hat{A}(x, y) V$ and recompute R .
- 10: **for** $(x, y) \in \Xi$ **do**
- 11: **if** $C\Delta(x, y; \mathcal{S}, \ell) < \Delta_{\text{max}}$ **then**
- 12: Exit the for loop.
- 13: **end if**
- 14: Recompute $\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell) = \lambda_{\min}(V^* \hat{A}(x, y) V)$.
- 15: Recompute the residual norm ρ according to (4.11).
- 16: Recompute $\lambda_{\text{LB}}(\mu; \mathcal{S} = \arg \min_{y \in \mathcal{O}_{\text{LB}}(\mathcal{S})} \theta(\mu)^T y$ and updated \check{y}_μ according to (3.24).
- 17: Recompute $y_\mu = \arg \min_{y \in \mathcal{O}_{\text{LB}}(\mathcal{S})} \theta(\mu)^T y$ and updated \check{y}_μ according to (3.24).
- 18: Recompute $\eta(\mu) \leftarrow \theta(\mu)^T \check{y}_\mu$.
- 19: Recompute $\lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell)$ according to (4.5).
- 20: Recompute $\Delta(x, y; \mathcal{S}, \ell)$ according to (4.8) and update Δ_{max} and $(x_{\text{max}}, y_{\text{max}})$.
- 21: **end for**
- 22: **end while**
- 23: Compute $\sigma_{\text{SUB}}(x, y; \mathcal{S}, \ell)$ and $\sigma_{\text{SLB}}(x, y; \mathcal{S}, \ell)$ for all $(x, y) \in \Xi$.

$\widehat{A}(x, y)$ is equivalent to computing the smallest singular values and associated singular vectors of the matrix $zI - A$. However, numerically this is not equivalent. When computing $\lambda_{\min}(\widehat{A}(x, y))$ directly, we are working with a matrix of squared condition number. To avoid that, we solve the singular value problem instead, by computing the smallest eigenpairs of the extended matrix $\begin{bmatrix} 0 & zI - A \\ (zI - A)^* & 0 \end{bmatrix}$ using the inverse Lanczos method. For a dense matrix A , this can be made more efficient by first computing the Schur decomposition of $A = QTQ^T$, see [Lui97, TE05], since $\sigma_{\min}(Q(zI - T)Q^T) = \sigma_{\min}(zI - T)$. In this case, each iteration of the inverse Lanczos method requires solving just two triangular linear systems. For a large-scale sparse matrix A , the inverse Lanczos method can be made more computationally efficient by first computing a sparse LU factorization of $zI - A$. We assume this method to be accurate and efficient for all $(x, y) \in \Xi$.

Computation of the residual norm ρ . Efficient and accurate computation of the residual ρ is very important for the accuracy of the lower bounds. The application of the technique used in Algorithm 3 requires precomputation of matrices in the affine linear expansion of $V^* \widehat{A}(x, y)^* \widehat{A}(x, y) V$, one of which is $V^* (A^* A)^* A^* AV$. We can expect $V^* (A^* A)^* A^* AV$ to be extremely ill-conditioned even for moderate $\kappa(A)$. To avoid this, we pay a slightly higher price and compute in each iteration the QR decomposition of the following $n \times 4M\ell$ matrix

$$QR = [A^* AV, (A + A^*)V, i(A^* - A)V, V].$$

For any $(x, y) \in D$, this allows computation of ρ by solving the following small $4M\ell \times r$ singular value problem

$$\begin{aligned} \rho &= \|U_{\perp}^* \widehat{A}(x, y)U\|_2 = \|\widehat{A}(x, y)U - U(U^* \widehat{A}(x, y)U)\|_2 \\ &= \|\widehat{A}(x, y)VW - VW\Lambda\|_2 \\ &= \|[A^* AV, (A + A^*)V, i(A^* - A)V, V][W^T, -xW^T, -yW^T, -\Lambda W^T]^T\|_2 \\ &= \|R[W^T, -xW^T, -yW^T, -\Lambda W^T]^T\|_2, \end{aligned} \quad (4.11)$$

where $W \in \mathbb{R}^{M\ell \times r}$ is such that $U = VW$ and $\Lambda = \text{diag}(\lambda_{\mathcal{V}}^{(1)}, \lambda_{\mathcal{V}}^{(2)}, \dots, \lambda_{\mathcal{V}}^{(r)})$.

Updating of $\lambda_{\text{LB}}(\widehat{A}(x, y))$. As explained in Section 3.4.1, computationally the most expensive part of computing $\lambda_{\text{SLB}}(\widehat{A}(x, y))$ is solving (4.6). In general, the interior point method is proposed for solving (4.6). However, for this specific application, the simplex method proves to be far superior, since the linear program (4.6) has just three variables. Additionally, as we incrementally build (4.6), the simplex method, unlike the interior-point method, allows us to take advantage of previously computed solutions and just slightly update them to compute the new ones. In practice, we observe that this modification significantly reduces the computational time.

Stopping criterion. Given the prescribed tolerance $\varepsilon_{\text{tol}} > 0$, we stop the execution of Algo-

rithm 4 when

$$\max_{(x,y) \in \Xi} \Delta(x, y; \mathcal{S}, \ell) = \max_{(x,y) \in \Xi} \frac{\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell) - \lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell)}{\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell) + x^2 + y^2} < \varepsilon_{\text{tol}}. \quad (4.12)$$

However, for $(x, y) \in \Xi$ close to an eigenvalue of A , fulfilling (4.12) requires the absolute error $\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell) - \lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell)$ to be very small which can not always be attained due to inexact computation of $\lambda_{\min}(\widehat{A}(x, y))$. To circumvent this issue, we additionally prescribe an absolute tolerance $\varepsilon_{\text{abs}} > 0$ and for points $(x, y) \in \Xi$ satisfying either

$$\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell) - \lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell) < \varepsilon_{\text{abs}}$$

or

$$\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell) + x^2 + y^2 < \varepsilon_{\text{abs}}.$$

In these cases, we set $\lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell)$ to the value of $\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell)$, assuming that $\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell)$ already is a very good approximation to $\sigma_{\min}((x + iy)I - A)$.

Computation of $\lambda_{\min}(\widehat{A}(x, y))$ for $x + iy \in \lambda(A)$.

As mentioned above, we "warm start" our approach by initializing \mathcal{S} to contain the exact eigenvalues of A inside D . However, for $z = x + iy \in \lambda(A)$, the inverse Lanczos method can not be directly applied since $(zI - A)^{-1}$ is not defined. Knowing that the smallest singular value is 0, it is possible to extract the non-singular part of $zI - A$, by deflating the directions of the smallest singular vectors, and compute the subsequent singular values and vectors.

Without loss of generality, we can assume that $x = y = 0$. Furthermore, we assume that zero is a simple eigenvalue of A . Suppose v_1 and u_1 are the left and the right singular vectors of A corresponding to the singular value zero:

$$v_1^* A = 0, \quad A u_1 = 0. \quad (4.13)$$

By Remark 2.2, we know that the Lanczos method will converge to the second largest eigenvector, if the starting vector is orthogonal to the dominant eigenvector. Thus, when computing the subsequent left (right) singular vectors using the Lanczos method, we need to choose an initial starting vector which is orthogonal to v_1 (u_1). However, in order to successfully apply the Lanczos method to this setting, we need to be able to efficiently solve the following linear systems

$$\text{for a given } v \in \{v_1\}^\perp \quad \text{find } u \in \{u_1\}^\perp \quad \text{s.t.} \quad A u = v, \quad (4.14)$$

$$\text{for a given } u \in \{u_1\}^\perp \quad \text{find } v \in \{v_1\}^\perp \quad \text{s.t.} \quad A^* v = u, \quad (4.15)$$

which are similar to the correction equation in Jacobi-Davidson SVD [Hoc01]. In the following

Lemma 4.8, we describe the procedure for solving (4.14), while (4.15) can be addressed in a similar way.

Lemma 4.8. *Let A have a simple zero singular value with corresponding singular vectors v_1, u_1 , and an LU decomposition $PAQ = LU$, where P, Q are permutation matrices, L has a unit diagonal, and the last row of U is zero. Let $v \in \mathbb{C}^n$ be such that $v^* v_1 = 0$ and let $H = I - 2ww^T / \|w\|^2$, $w \in \mathbb{C}^n$ be the Householder reflector such that $Hu_1 = e_1$. Then, there exist $\gamma \in \mathbb{C}$ and an invertible matrix $\tilde{U} \in \mathbb{C}^{(n-1) \times (n-1)}$ such that*

$$UQ^T H = UQ^T - \gamma UQ^T e_1 w^T \quad (4.16)$$

$$= \begin{bmatrix} 0 & \tilde{U} \\ 0 & 0 \end{bmatrix}. \quad (4.17)$$

Moreover, solving (4.14) is equivalent to solving the following linear system for $u \in \{u_1\}^\perp$

$$\begin{bmatrix} 0 & \tilde{U} \\ 0 & 0 \end{bmatrix} Hu = L^{-1} P v = \begin{bmatrix} \tilde{v} \\ 0 \end{bmatrix}, \quad (4.18)$$

with $\tilde{v} \in \mathbb{C}^{n-1}$, whose solution u is given as

$$u = H \begin{bmatrix} 0 \\ \tilde{U}^{-1} \tilde{v} \end{bmatrix}. \quad (4.19)$$

Proof. From the fact that the last row of U is zero, we have $e_n = (0, \dots, 0, 1) \in \ker(U^*)$, and thus also

$$e_n^* UQ^T H = 0. \quad (4.20)$$

As $P^T L$ is invertible, $0 = Au_1 = P^T LUQ^T u_1$ implies that

$$UQ^T u_1 = UQ^T H e_1 = 0. \quad (4.21)$$

which, when combined with (4.20) and the simplicity of the zero singular value, yields (4.17). Moreover, since the vector w in the definition of H is given as $w = u_1 + \text{sign}((u_1)_1) e_1$, the identity (4.21) also implies (4.16) with $\gamma = 2 \frac{\text{sign}((u_1)_1)}{\|w\|^2}$.

Since $H^2 = I$ and $P^T L$ is invertible, (4.14) can be equivalently written as

$$P^T LUQ^T H H u = P^T L \begin{bmatrix} 0 & \tilde{U} \\ 0 & 0 \end{bmatrix} H u = v \iff \begin{bmatrix} 0 & \tilde{U} \\ 0 & 0 \end{bmatrix} H u = L^{-1} P v.$$

Thus, to prove (4.18), it is sufficient to show that there exists $\tilde{v} \in \mathbb{C}^{n-1}$ such that

$$L^{-1} P v = \begin{bmatrix} \tilde{v} \\ 0 \end{bmatrix}. \quad (4.22)$$

As $v_1^* A = v_1^* P^T L U Q^T = 0$, we have that $(v_1^* P^T L)^* \in \ker(U^*) = \text{span}\{e_n\}$, or equivalently since $P^T L$ is invertible:

$$v_1^* P^T L = \alpha e_n^* \Rightarrow v_1^* = \alpha e_n^* L^{-1} P,$$

for some nonzero $\alpha \in \mathbb{C}$. Since $v_1^* v = 0$, this implies

$$v_1^* v = \alpha e_n^* L^{-1} P v = 0 \Rightarrow e_n^* L^{-1} P v = 0, \quad (4.23)$$

which immediately yields (4.22), and proves (4.18). Clearly, u as in (4.19) is a good candidate for the solution of (4.18). In fact, it can be easily verified that this choice of u also satisfies the orthogonality condition in (4.14):

$$u_1^* u = u_1^* H \begin{bmatrix} 0 \\ \tilde{U}^{-1} \tilde{v} \end{bmatrix} = e_1^* \begin{bmatrix} 0 \\ \tilde{U}^{-1} \tilde{v} \end{bmatrix} = 0,$$

which proves (4.19) and concludes the proof. \square

As indicated in (4.16), multiplying U with $Q^T H$ usually does not destroy the underlying sparsity pattern, i.e. \tilde{U} has approximately equal number of nonzero elements as U . This was also the case in all of the numerical examples considered in Section 4.4, and thus, in the actual implementation, the computation of u was further accelerated by computing a sparse LU decomposition of \tilde{U} . Full procedure for the solution of (4.14) is summarized in Algorithm 5.

Algorithm 5 Solving the deflated linear system (4.14).

Input: A vector $v \in \{v_1\}^\perp$, LU decompositions $A - zI = P^T L U Q^T$ and $\tilde{U} = \hat{P}^T \hat{L} \hat{U} \hat{Q}^T$, with \tilde{U} defined as in (4.17), the singular vectors v_1 and u_1 corresponding to $\sigma_{\min}(zI - A)$ and a Householder reflector H such that $Hu_1 = e_1$.

Output: Vector $u \in \{u_1\}^\perp$ such that $(A - zI)u = v$.

- 1: Compute $\hat{v} = L^{-1} P v$ by solving a triangular linear system.
 - 2: Extract the first $n - 1$ components of \hat{v} into \tilde{v} .
 - 3: Compute $\hat{u} = \hat{Q} \hat{U}^{-1} \hat{L}^{-1} \hat{P} \tilde{v}$ by solving two triangular linear systems.
 - 4: Compute $u = H \begin{bmatrix} 0 \\ \hat{u} \end{bmatrix}$.
-

4.3.2 Parameter value selection

Choice of Ξ . As previously discussed in Sections 3.4.2 and 4.2, for $d = 2$, it is reasonable to choose $\Xi \subset \mathbb{C}$ as a finite uniform grid on D . The complexity and the quantity of features in the underlying pseudospectral image dictate the required resolution and thus also the number of grid points needed in each of the directions. In practice, for the numerical examples considered in Section 4.4, we have used Ξ to be a 100×100 uniform grid on D .

Choice of ℓ . Using a larger value of ℓ , number of the smallest eigenvectors included in $\mathcal{V}(\mathcal{S}, \ell)$ per sample point, leads to better bounds, but on the other hand, it increases the compu-

tational cost. As explained in Section 3.4.2, given eigenvalue gaps between few smallest eigenvalues, ℓ should be chosen to maximize the eigenvalue gap $\lambda_i^{(\ell+1)} - \lambda_i^{(\ell)}$. In the absence of *a priori* information on eigenvalue gaps, the experiments in Chapter 3 indicated the choice of $\ell = 1$ to be optimal. However, in the examples presented in Section 4.4 we observe the gap between the first few smallest eigenvalues to be very small and the performance of Algorithm 4 improved significantly by using a slightly larger value for ℓ . In our implementation, we have used $\ell = 6$ for all $(x, y) \in D$, as this usually ensured the eigenvalue gap $\lambda_i^{(\ell+1)} - \lambda_i^{(1)}$ to be sufficiently large for our approach to provide satisfying convergence.

4.3.3 Algorithm and computational complexity.

Algorithm 4 summarizes our proposed approach explained in the previous sections, taking into account implementational details from Section 4.3.1. The algorithm requires solution of M singular value problems of size $n \times n$ for computing the exact smallest singular values and vectors of $zI - A$, one for each $z \in \mathcal{S}$. Computing $\lambda_{\text{SUB}}(\mu; \mathcal{S}, \ell)$ and $\lambda_{\text{SLB}}(\mu; \mathcal{S}, \ell)$ for all $(x, y) \in \Xi$ in every iteration amounts to solving at most $M|\Xi|$ eigenproblems of size at most $M\ell \times M\ell$, as well as at most $M|\Xi|$ LP problems with 3 variables and up to M constraints. As long as $M\ell \ll n$, these parts will be negligible, and the computational cost of Algorithm 4 will be dominated by the cost of computing the exact singular values and vectors. Moreover, as explained in Section 3.4.1, by the saturation assumption, we do not have to recompute the bounds $\lambda_{\text{SUB}}(\mu; \mathcal{S}, \ell)$ and $\lambda_{\text{SLB}}(\mu; \mathcal{S}, \ell)$ for all $(x, y) \in \Xi$ in every iteration. In practice, the bounds for specific $(x, y) \in \Xi$ are recomputed only a few times throughout the execution.

4.4 Numerical experiments

In this section, we report on the performance of our proposed approach for a number of large-scale examples available in the literature and compare it with some of the existing approaches discussed in Section 4.1. Algorithm 4 has been implemented and tested in the same computing environment already described in Section 3.5.

When implementing and testing Algorithm 4, we have made the following choices. Unless stated otherwise, we set the error tolerance ε_{tol} to 0.1, the absolute tolerance ε_{abs} to 10^{-8} , the maximum number of sampled points to $M_{\text{max}} = 100$ and Ξ to be 100×100 uniformly spaced grid on D . The smallest singular values and the corresponding singular vectors of $zI - A$ have been computed, as explained in Section 4.3.1, using the MATLAB built-in function `eigs`, which is based on ARPACK [LSY98], with the tolerance set to 10^{-10} . For solving the linear program (4.6), we have used MOSEK 7 Matlab toolbox [ApS15] implementation of the simplex method with updating. In all experiments, we have used Algorithm 4 with the number of smallest eigenpairs included in $\mathcal{V}(\mathcal{S}, \ell)$ set to $\ell = 6$. In the first three iterations we have worked with the saturation constant set to $C_{\text{sat}} = +\infty$ and $C_{\text{sat}} = 1$ in the following iterations. For choosing r from Section 3.2, we have tested all values $r = 0, 1, \dots, 3\ell$, as explained in

Section 4.3.1.

4.4.1 Comparison with other approaches

As can be seen in Examples 4.9 – 4.14, in terms of computational time, Algorithm 4 is significantly faster than the grid-based approach, while providing satisfying accuracy. However, in some examples, especially Example 4.12, the speedup is not great, even though Algorithm 4 solves the full-size singular value problem only a couple of times. As the subspace $\mathcal{V}(\mathcal{S}, \ell)$ gets larger, the amount of time spent in computing the subspace upper bounds $\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell)$ is no more negligible in comparison to exact computation of $\sigma_{\min}(zI - A)$. As already discussed in Remark (3.20), this issue can be addressed by splitting the parameter domain and solving a few smaller problems. However, how exactly to incorporate this idea into Algorithm 4 remains an open question and may be interesting for future research.

Additionally, we compare the performance of Algorithm 4 against two other projection-based approaches presented in Section 4.1, namely the Krylov subspace approach and the invariant subspace approach. On a smaller 30×30 uniformly spaced grid $\tilde{\Xi}$, we compute the exact smallest singular values $\sigma_{\min}(zI - A)$, as well as the approximations $\sigma_{\text{kry}}(x, y; k)$ and $\sigma_{\text{inv}}(x, y; k)$ for few values of $k \in \mathbb{N}$, where

$$\begin{aligned}\sigma_{\text{kry}}(x, y; k) &= \sigma_{\min}(zU_k^{\text{kry}} - AU_k^{\text{kry}}) \\ \sigma_{\text{inv}}(x, y; k) &= \sigma_{\min}(zU_k^{\text{inv}} - AU_k^{\text{inv}}),\end{aligned}$$

with U_k^{arn} and U_k^{inv} the k -dimensional Krylov subspace of matrix A and the k -dimensional invariant subspace spanned by the eigenvectors corresponding to eigenvalues closest to D , respectively. In Figures 4.3c – 4.8c, we present the convergence rates towards the exact values of $\sigma_{\min}((x + iy)I - A)$:

$$\max_{(x, y) \in \tilde{\Xi}} \frac{\sigma_{\text{kry}}(x, y; k)^2 - \sigma_{\min}((x + iy)I - A)^2}{\sigma_{\min}((x + iy)I - A)^2}, \text{ and} \quad (4.24)$$

$$\max_{(x, y) \in \tilde{\Xi}} \frac{\sigma_{\text{inv}}(x, y; k)^2 - \sigma_{\min}((x + iy)I - A)^2}{\sigma_{\min}((x + iy)I - A)^2} \quad (4.25)$$

w.r.t. the subspace size k and compare them to the corresponding convergence rates for the computed subspace bounds $\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell)$ and $\lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell)$:

$$\max_{(x, y) \in \tilde{\Xi}} \frac{\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell) - \sigma_{\min}((x + iy)I - A)}{\sigma_{\min}((x + iy)I - A)} \quad (4.26)$$

$$\max_{(x, y) \in \tilde{\Xi}} \frac{\lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell) - \sigma_{\min}((x + iy)I - A)}{\sigma_{\min}((x + iy)I - A)} \quad (4.27)$$

w.r.t. the dimensionality of the subspace $\mathcal{V}(\mathcal{S}, \ell)$. We can observe that the convergences of $\sigma_{\text{krv}}(x, y; k)^2$ and $\sigma_{\text{inv}}(x, y; k)^2$ usually flatten after first few iterations, while the subspace upper bounds $\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell)$ provide a very accurate approximation to $\sigma_{\min}((x + iy)I - A)^2$ after only a few iterations. The corresponding relative error 4.26 is often very small already at the beginning due to the "warm start" strategy and this very fast convergence to the exact values could be used as a motivation for deriving a heuristic version of our approach. Eventually, when Algorithm 4 finishes, we usually observe that even the subspace lower bounds $\lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell)$ provide a more accurate approximation than the other two approaches.

4.4.2 Dense matrices

We first consider two moderately sized dense matrices A ($n \leq 5000$), such that it is still possible to compute their Schur decomposition $A = QTQ^*$. We compute approximate pseudospectra $\sigma_\varepsilon(T)$, and compare the results obtained using Algorithm 4 and other approaches for pseudospectra computation. For more details, see Examples 4.9 and 4.10.

Example 4.9. We consider the example `random_demo.m` from `EigTool` [Wri02], where $A \in \mathbb{R}^{n \times n}$ is a random matrix whose entries are drawn from the normal distribution with mean 0 and variance $1/N$. As $N \rightarrow \infty$, spectral abscissa of A converges to 1. We choose $N = 2000$ and set $D = [0.95, 1.05] + [-0.05, 0.05]i$ to be a region in the complex plane around the right-most part of the spectrum. The observed matrix A has four eigenvalues inside D . The spectrum of A (blue dots) in the region around D (red square) is shown in Figure 4.3a, whereas in Figure 4.3b we can see the convergence of the maximum error estimate in Algorithm 4 w.r.t. iteration. The Algorithm 4 reaches the desired tolerance in 26 iterations with the computational time of 1613 seconds, while the exact computation using a grid-based approach would take around 22000 seconds. In Figure 4.3e we see the computed ε -pseudospectra for $\varepsilon = 10^{-1}, 10^{-2}$, while in Figure 4.3d the surface plot of $\sigma_{\text{SUB}}(x, y; \mathcal{S}, \ell)$ is presented. We see that with prescribed tolerance ε_{tol} , the upper and the lower bounds for ε -pseudospectra almost completely overlap. The convergence of the maximum relative error for $\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell), \lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell), \sigma_{\text{krv}}(x, y; k)^2, \sigma_{\text{inv}}(x, y; k)^2$ w.r.t. the subspace size is shown in Figure 4.3c.

Example 4.10. We consider the example `landau_demo.m` from `EigTool` [Wri02], with matrix A representing an integral equation from laser theory [Lan78]. We choose $N = 4000$ and $D = [0.8, 1.2] + [-0.2, 0.2]i$, a region in the complex plane around the right-most part of the spectrum. There are five eigenvalues of A inside D which we initially include in \mathcal{S} . The spectrum of A (blue dots) in the region around D (red square) is shown in Figure 4.4a, whereas in Figure 4.4b we can see the convergence of the maximum error estimate in Algorithm 4 w.r.t. to iteration. The Algorithm 4 reaches the desired tolerance in only 4 iterations with the computational time of 637 seconds, while the exact computation using a grid-based approach would take around 80000 seconds. In Figure 4.4e, we see the computed ε -pseudospectra for $\varepsilon = 10^{-1}, 10^{-2}$. while in Figure 4.4d the surface plot of $\sigma_{\text{SUB}}(x, y; \mathcal{S}, \ell)$ is presented. The convergence of the maximum relative error for $\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell), \lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell), \sigma_{\text{krv}}(x, y; k)^2, \sigma_{\text{inv}}(x, y; k)^2$ w.r.t. the subspace size is shown in Figure 4.4c.

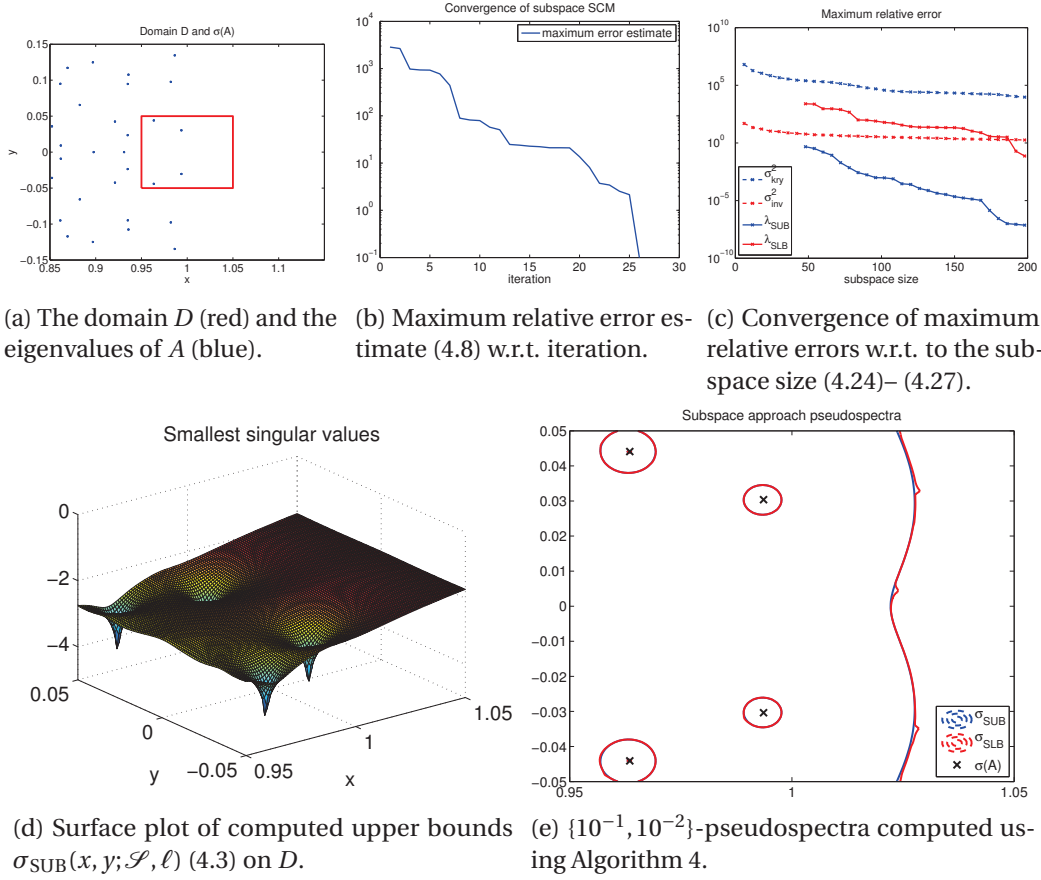


Figure 4.3: Application of Algorithm 4 to Example 4.9.

4.4.3 Sparse matrices

For a large-scale sparse matrix A , computing the Schur decomposition of A is rarely possible and almost never justified. We consider four large sparse matrices A and compute approximate pseudospectra $\sigma_\varepsilon(A)$, and compare the results obtained using Algorithm 4 with other approaches for pseudospectra computation. As explained in Section 4.3.1, we use the sparse LU decomposition of A to speed up the computation of $\lambda_{\min}(\hat{A}(x, y))$. For more details, see Examples 4.11 – 4.14.

Example 4.11. *This example arises in fluid dynamics, as a model of a flow over obstacle, with the Reynolds number equal to 100, linearized around steady state, using Q2-Q1 mixed finite elements using IFISS [ERS07]. We are given matrices A and M of size $N = 9512$ representing finite elements discretizations of the operator and the mass matrix, respectively. We compute pseudospectra of the matrix $M^{-1}A$ in $D = [-1.2, -0.2] + [-0.5, 0.5]i$, a region in the complex plane around the right-most part of the spectrum. There are three eigenvalues of A inside D which we initially include in \mathcal{S} . The spectrum of A (blue dots) in the region around D (red square) is shown in Figure 4.5a, whereas in Figure 4.5b we can see the convergence of the maximum error estimate in Algorithm 4 w.r.t. to iteration. The Algorithm 4 reaches the desired tolerance in 36 iterations with the computational time of 2355 seconds, while the exact*

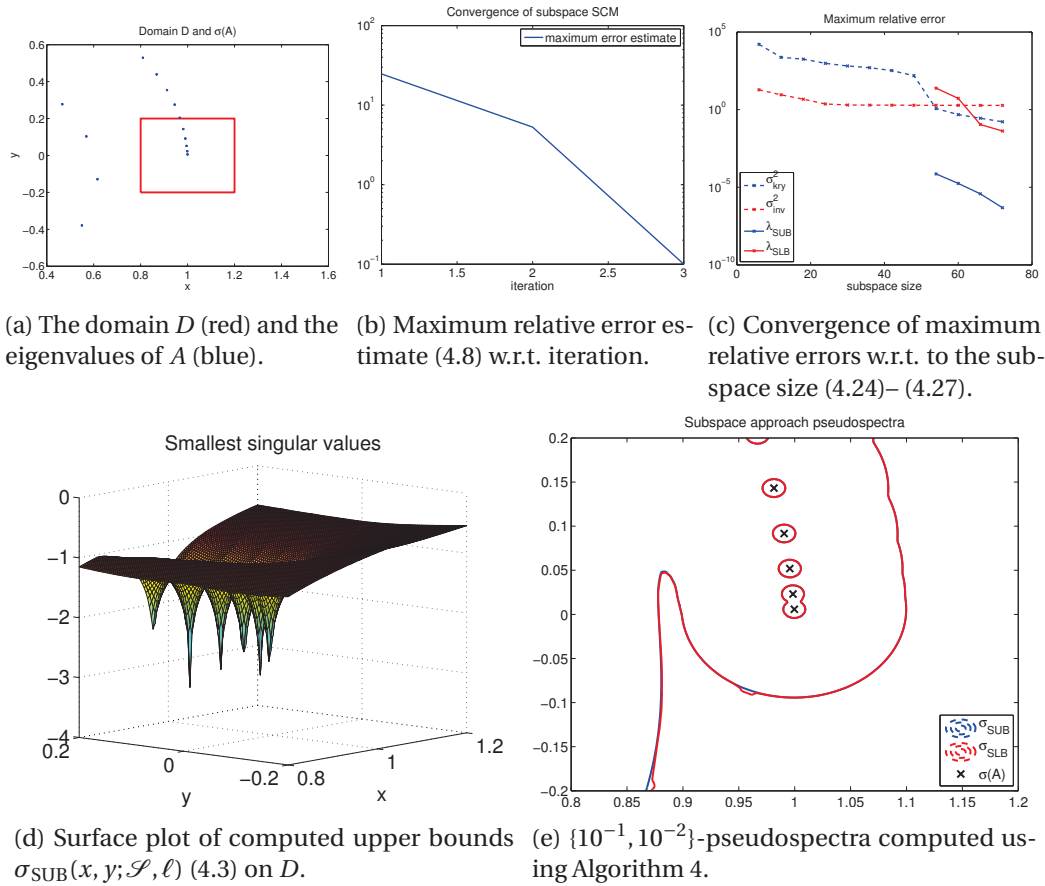


Figure 4.4: Application of Algorithm 4 to Example 4.10.

computation using a grid-based approach would take around 11000 seconds. In Figure 4.5e, we see the computed ε -pseudospectra for $\varepsilon = 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}$. while in Figure 4.5d the surface plot of $\sigma_{\text{SUB}}(x, y; \mathcal{S}, \ell)$ is presented. The convergence of the maximum relative error for $\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell), \lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell), \sigma_{\text{krv}}(x, y; k)^2, \sigma_{\text{inv}}(x, y; k)^2$ w.r.t. the subspace size is shown in Figure 4.5c.

Example 4.12. We consider the MATPDE example from the Matrix Market [MMa07] collection of non-Hermitian eigenvalue problems, where the matrix A is a five-point central finite difference discretization of the two-dimensional variable-coefficient linear elliptic equation. Size of the matrix A is $N = 2961$ and we choose $D = [0, 0.1] + [0, 0.1]i$, region in the complex plane around the left-most part of the spectrum. In this region there are six eigenvalues of A which we initially include in \mathcal{S} . The spectrum of A (blue dots) in the region around D (red square) is shown in Figure 4.6a, whereas in Figure 4.6b we can see the convergence of the maximum error estimate in Algorithm 4 w.r.t. to iteration. The Algorithm 4 reaches the desired tolerance in 14 iterations with the computational time of 686 seconds, while the exact computation using a grid-based approach would take around 877 seconds. In Figure 4.6e, we see the computed ε -pseudospectra for $\varepsilon = 10^{-2}, 10^{-3}, 10^{-4}$. while in Figure 4.6d the surface plot of $\sigma_{\text{SUB}}(x, y; \mathcal{S}, \ell)$ is presented. The convergence of the maximum relative error for

4.4. Numerical experiments

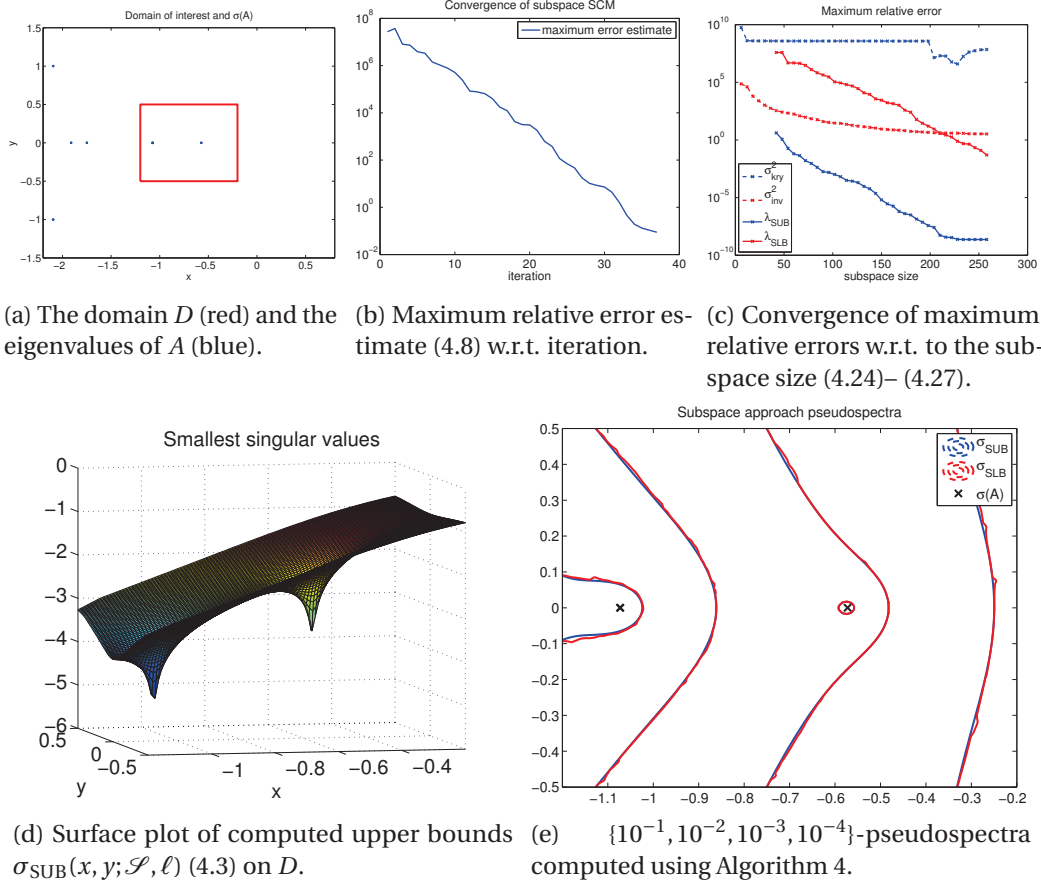


Figure 4.5: Application of Algorithm 4 to Example 4.11.

$\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell), \lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell), \sigma_{\text{Kry}}(x, y; k)^2, \sigma_{\text{inv}}(x, y; k)^2$ w.r.t. the subspace size is shown in Figure 4.6c.

Example 4.13. We consider the BRUSSEL example from the Matrix Market [MMa07] collection of non-Hermitian eigenvalue problems, where the matrix A arises in chemical engineering as a discretization of a 2D reaction-diffusion model. Size of the matrix A is $N = 3200$ and we choose $D = [-0.5, 0.5] + [1.5, 2.5]i$, region in the complex plane around the right-most part of the spectrum. In this region there are three eigenvalues of A which we initially include in \mathcal{S} . In this example, we observe that the subspace containing the sampled smallest singular vectors $\mathcal{V}(\mathcal{S}, \ell)$ can be well approximated by a subspace containing lot less than $M\ell$ vectors. Instead of simply using the QR decomposition like in other examples, here we compute the orthonormal basis V using the truncated singular value decomposition with the tolerance set to 10^{-10} . The spectrum of A (blue dots) in the region around D (red square) is shown in Figure 4.7a, whereas in Figure 4.7b we can see the convergence of the maximum error estimate in Algorithm 4 w.r.t. to iteration. The Algorithm 4 reaches the desired tolerance in 35 iterations with the computational time of 365 seconds, while the exact computation using a grid-based approach would take around 1580 seconds. In Figure 4.7e, we see the computed ε -pseudospectra for $\varepsilon = 10^{-1}, 10^{-2}$. while in Figure 4.7d the surface plot of $\sigma_{\text{SUB}}(x, y; \mathcal{S}, \ell)$ is presented. The convergence

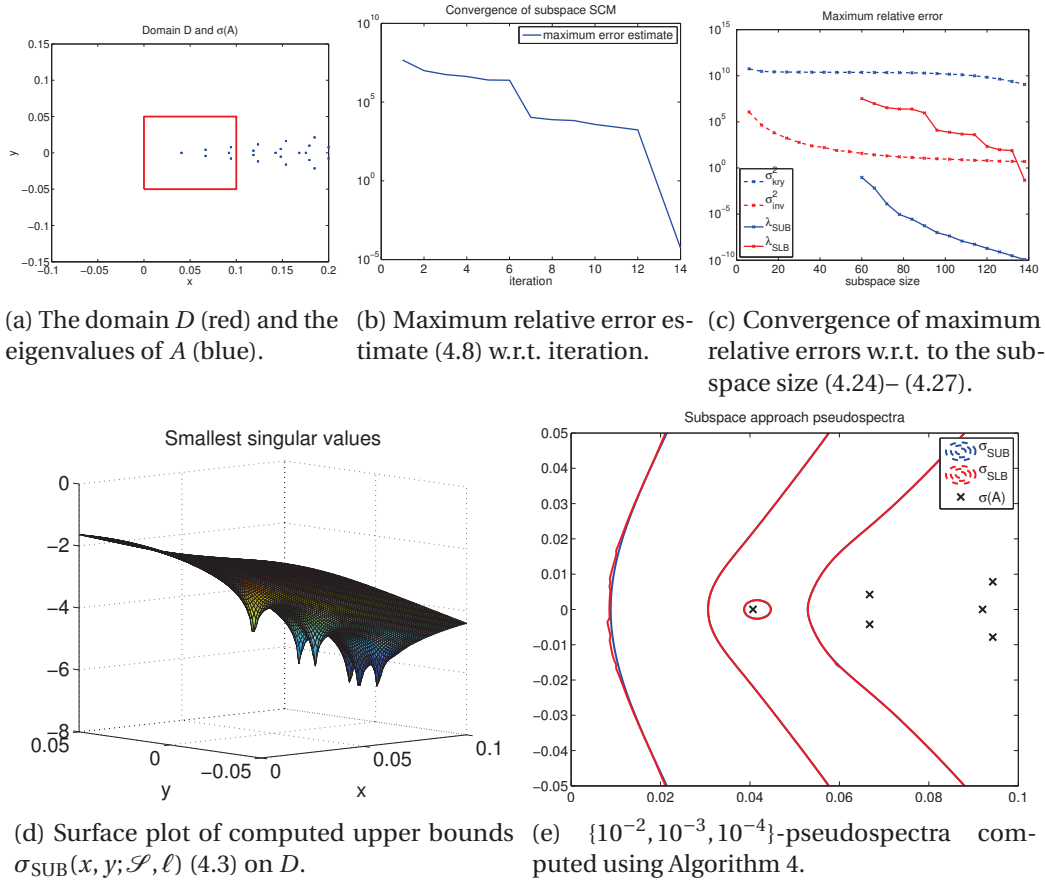


Figure 4.6: Application of Algorithm 4 to Example 4.12.

of the maximum relative error for $\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell)$, $\lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell)$, $\sigma_{\text{kry}}(x, y; k)^2$, $\sigma_{\text{inv}}(x, y; k)^2$ w.r.t. the subspace size is shown in Figure 4.7c.

Example 4.14. We consider the H2plus example from the Matrix Market [MMa07] collection of non-Hermitian eigenvalue problems, where the matrix A arises in quantum chemistry as a discretization of a model for H_2^+ in an electromagnetic field. Size of the matrix A is $N = 2534$ and we choose $D = [2.5, 3.5] + [-0.5, 0.5]i$, region in the complex plane around the right-most part of the spectrum. In this region there are six eigenvalues of A which we initially include in \mathcal{S} . The spectrum of A (blue dots) in the region around D (red square) is shown in Figure 4.8a, whereas in Figure 4.8b we can see the convergence of the maximum error estimate in Algorithm 4 w.r.t. to iteration. The Algorithm 4 reaches the desired tolerance in only 5 iterations with the computational time of 191 seconds, while the exact computation using a grid-based approach would take around 8000 seconds. In Figure 4.8e, we see the computed ε -pseudospectra for $\varepsilon = 3 \cdot 10^{-1}, 10^{-1}, 3 \cdot 10^{-2}, 10^{-2}$. while in Figure 4.8d the surface plot of $\sigma_{\text{SUB}}(x, y; \mathcal{S}, \ell)$ is presented. The convergence of the maximum relative error for $\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell)$, $\lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell)$, $\sigma_{\text{kry}}(x, y; k)^2$, $\sigma_{\text{inv}}(x, y; k)^2$ w.r.t. the subspace size is shown in Figure 4.8c. The relative error for $\sigma_{\text{inv}}(x, y; k)^2$ increases for larger values of k due to the fact that not all eigenvectors included in the invariant subspace have converged.

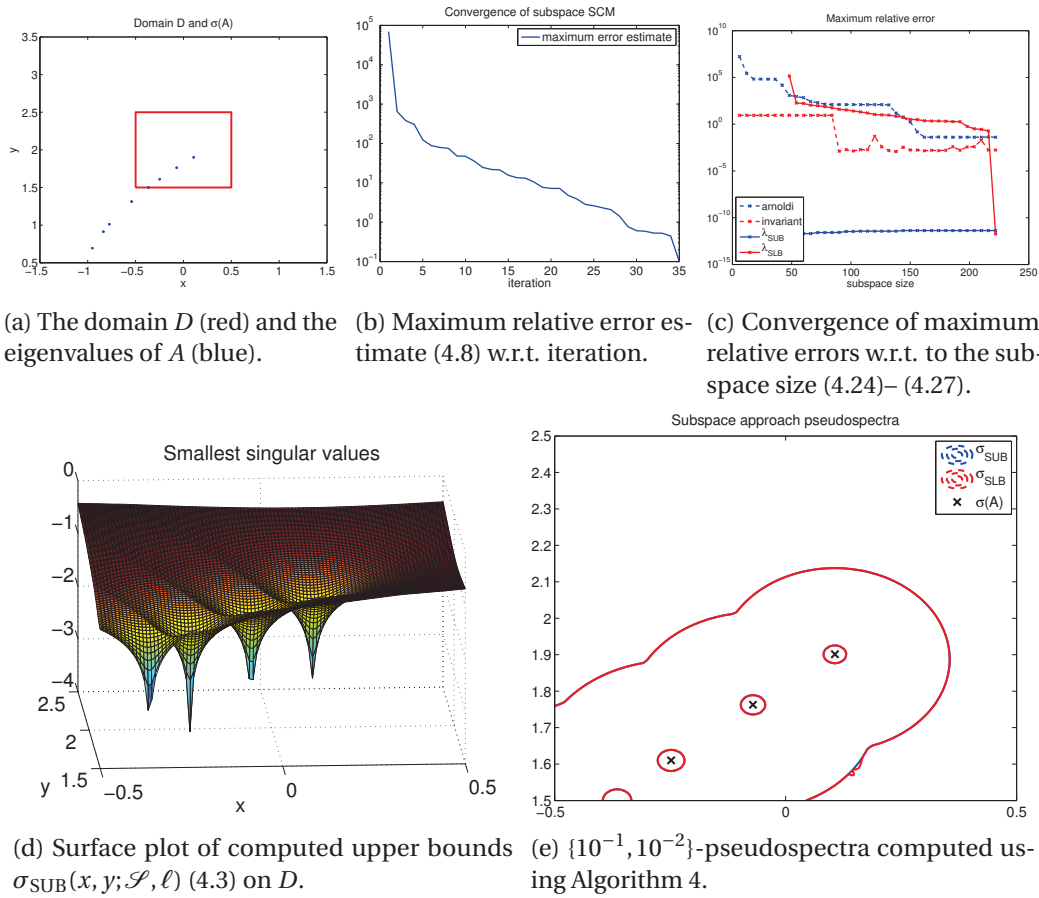


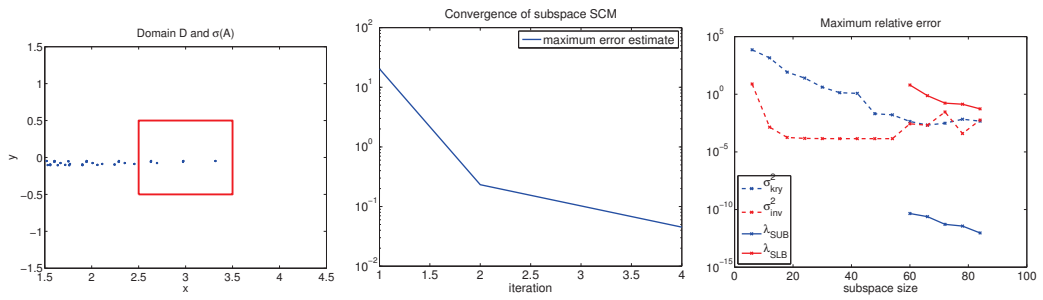
Figure 4.7: Application of Algorithm 4 to Example 4.13.

4.5 Conclusion

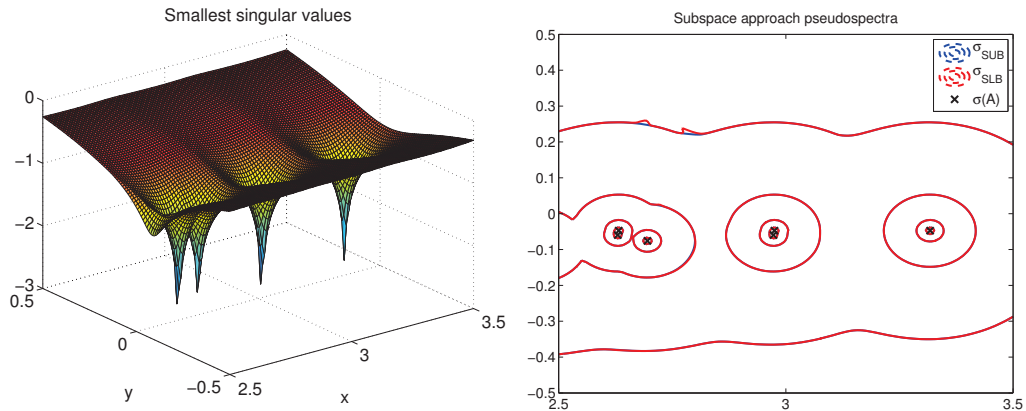
We have proposed a novel projection-based approach inspired by the greedy sampling strategies, given in Algorithm 4. It is primarily designed to provide highly accurate approximations of ε -pseudospectra in isolated parts of the spectrum, containing only few eigenvalues of A .

We have shown that the singular value problem $\sigma_{\min}(zI - A)$ can be recasted into a Hermitian eigenvalue problem linearly depending on two real parameters. The presented approach uses this characterization, and builds upon the subspace-accelerated approach for approximating the smallest eigenvalues of a parameter-dependent Hermitian matrix presented in Algorithm 3 and discussed in Chapter 3. Taking into account the particular problem structure and demands for high absolute accuracy, we have modified Algorithm 3 in order to make our approach computationally efficient and competitive. In particular, we have made the approach more numerically stable, accelerated the computation of the lower bounds, as well as introduced a "warm start" strategy. Additionally, we have extended the interpolation results from Chapter 3 to the proposed singular value bounds, allowing us to provide *a priori* error estimates.

Moreover, we have compared the performance of our approach to few other existing ap-



(a) The domain D (red) and the eigenvalues of A (blue). (b) Maximum relative error estimate (4.8) w.r.t. iteration. (c) Convergence of maximum relative errors w.r.t. to the subspace size (4.24)–(4.27).



(d) Surface plot of computed upper bounds $\sigma_{\text{SUB}}(x, y; \mathcal{S}, \ell)$ (4.3) on D . (e) $\{3 \cdot 10^{-1}, 10^{-1}, 3 \cdot 10^{-2}, 10^{-2}\}$ -pseudospectra computed using Algorithm 4.

Figure 4.8: Application of Algorithm 4 to Example 4.14.

proaches on a number of examples discussed in the literature. For larger values of n , our approach is significantly faster than the grid-based approach, while providing satisfactory accuracy. In comparison to the other projection-based approaches, our approach provides higher relative accuracy w.r.t. to the subspace size, especially in the proposed upper bounds $\sigma_{\text{SUB}}(x, y; \mathcal{S}, \ell)$, as well as the rigorous error estimates. Moreover, to our knowledge it is the first approach which provides certified upper bounds for ε -pseudospectra, enabling localization of eigenvalues.

5 Greedy low-rank approach to linear matrix equations

We consider the numerical solution of large-scale linear matrix equations of the form

$$\sum_{q=1}^Q A_q X B_q^T = C, \quad (5.1)$$

for given coefficient matrices $A_1, \dots, A_Q \in \mathbb{R}^{m \times m}$, $B_1, \dots, B_Q \in \mathbb{R}^{n \times n}$, $C \in \mathbb{R}^{m \times n}$. Equation can also be seen as a linear system

$$\sum_{q=1}^Q (B_q \otimes A_q) \text{vec}(X) =: \mathcal{A} \text{vec}(X) = \text{vec}(C). \quad (5.2)$$

The matrix equation (5.1) is uniquely solvable if and only if $\mathcal{A} \in \mathbb{R}^{mn \times mn}$ is invertible, which will be assumed throughout this chapter.

For $Q = 2$, the matrix equation (5.1) reduces to the so called *generalized Sylvester equation*, within which particularly important special cases are the standard Sylvester equation $A_1 X + X B_2^T = C$ and the *Lyapunov equation* $A_1 X + X A_1^T = -C$, with C symmetric positive definite. The efficient numerical solution of Lyapunov and Sylvester equations has been studied intensively during the last decades, and significant progress has been made; we refer to [BS13, Sim13] for recent surveys. In particular, a number of approaches have been developed for $Q = 2$ that avoid the explicit computation and storage of the $m \times n$ matrix X . Such methods attempt to compute a low-rank approximation to X and store only the low-rank factors. As already discussed in Section 2.2.2 for the case of Lyapunov equations, one popular approach which implements this idea is the ADI method, which has also been extended to solving the Sylvester equation in [BLT09]. Of course, this requires that X can be well approximated by a low-rank matrix at the first place, that is, that the singular values of X have a strong decay. As already discussed in Section 2.2.1, such a decay has been shown for Lyapunov equations with a low-rank right-hand side C .

However, none of the established methods for Lyapunov and Sylvester equations generalizes to the case $Q > 2$. In fact, the recent survey paper by Simoncini [Sim13] states: *The efficient numerical solution to ... [reference to equation (5.1)] thus represents the next frontier for linear*

matrix equations . . . Among the existing work addressing $Q > 2$, particular attention has been paid to the *generalized Lyapunov equation*

$$AX + XA^T + \sum_{q=1}^Q N_q X N_q^T = -DD^T. \quad (5.3)$$

In fact, this appears to be the most frequently encountered instance of (5.1) for $Q > 2$ and typically arises in connection with bilinear dynamical systems. By extending results for the Lyapunov case, singular value decay bounds for X have been established in [BB13, Mer12], under various conditions on A and N_q .

As already discussed in Remark 2.13, iterative methods for solving Lyapunov equations can be successfully preconditioned with, for example, few steps of the ADI method. In a similar fashion, the ADI preconditioning has been used in the fixed point iteration proposed by Damm [Dam08] for solving (5.3). The iteration is based on the splitting $\mathcal{L}(X) + \mathcal{N}(X) = -DD^T$ of (5.3) with the Lyapunov operator $\mathcal{L} : X \mapsto AX + XA^T$. This iteration converges if \mathcal{L} is the dominant part of (5.3), that is, the spectral radius of $\mathcal{L}^{-1}\mathcal{N}$ is smaller than 1.

A rather different approach by Benner and Breiten [BB13] treats (5.3) as an $n^2 \times n^2$ linear system in the entries of X . Based on ideas from [EB10, KT11], a standard iterative solver, such as CG or BiCGstab, is combined with low-rank truncation of the iterates. This approach requires the availability of a preconditioner to ensure fast convergence. There is evidence [KT11] that fast convergence is crucial to avoid an excessive growth of the *numerical* ranks during intermediate iterations. Natural candidates for preconditioners are \mathcal{L} or approximations thereof, such as one iteration of the ADI method, especially if \mathcal{L} is the dominant part. Numerical experiments reported in [BB13] demonstrate that this approach performs remarkably well.

In this chapter, we develop a framework of low-rank methods for addressing the general linear matrix equation (5.1). Our approach is very much inspired by a class of methods proposed in [AMCK06, Nou10] for solving Fokker-Planck equations and stochastic partial differential equations, see [CAC10] for a survey of recent developments. The basic idea is to subsequently refine the current approximation to the solution X by adding a rank-1 correction. This correction is chosen to minimize a certain target functional, which renders the approach a greedy algorithm. As we will see, this basic approach may require further improvement to perform well for a larger range of applications. We will discuss two techniques for improving convergence: adding information from the preconditioned residual, similar to the techniques considered in [DS14], and performing Galerkin projection.

The rest of this chapter is largely based on [KS15] and is organized as follows. In Section 5.1, we explain the basic algorithm using greedy rank-1 updates. For the special case of stable symmetric Lyapunov equations, this algorithm is shown to preserve symmetry of the solution. As shown in Section 5.2, the performance of this basic algorithm is improved by using Galerkin projections. In Section 5.3, we discuss the incorporation of preconditioners into the method. Finally, a variety of numerical experiments is presented in Section 5.4.

5.1 Greedy rank-1 approach

In this section, we describe the basic greedy rank-1 strategy for approximating the solution X of (5.1). Starting from the zero initial guess $X_0 = 0$, a sequence of approximations X_1, X_2, X_3, \dots with $\text{rank}(X_j) \leq j$ is constructed as follows. Given the current approximation X_j , the next approximation takes the form

$$X_{j+1} = X_j + u_{j+1}v_{j+1}^T, \quad (5.4)$$

where the rank-1 correction $u_{j+1}v_{j+1}^T$ is chosen to minimize the approximation error. If the system matrix \mathcal{A} defined in (5.2) is symmetric positive definite, we may use the energy norm induced by \mathcal{A} to measure the error. Otherwise, we will use the residual norm. In the following, we will discuss details for these two choices. For notational convenience, we will identify the matrix representation $\mathcal{A} \in \mathbb{R}^{mn \times mn}$ with the corresponding linear operator

$$\mathcal{A} : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}^{m \times n}, \quad \mathcal{A} : X \mapsto \sum_{q=1}^Q A_q X B_q^T.$$

5.1.1 Symmetric positive definite case

Let us assume that \mathcal{A} is symmetric positive definite. Then the linear operator \mathcal{A} induces the scalar product $\langle Y, Z \rangle_{\mathcal{A}} = \text{tr}(Y^T \mathcal{A}(Z))$ on $\mathbb{R}^{m \times n}$ along with the corresponding norm $\|Y\|_{\mathcal{A}} = \sqrt{\langle Y, Y \rangle_{\mathcal{A}}}$. We choose the correction $u_{j+1}v_{j+1}^T$ in (5.4) such that the approximation error measured in this norm is as small as possible. This yields the minimization problem

$$\begin{aligned} \min_{u,v} \|X - X_j - uv^T\|_{\mathcal{A}}^2 &= \min_{u,v} \langle X - X_j - uv^T, X - X_j - uv^T \rangle_{\mathcal{A}} \\ &= \|X - X_j\|_{\mathcal{A}}^2 + \min_{u,v} \langle uv^T, uv^T \rangle_{\mathcal{A}} - 2 \text{tr}(vu^T \mathcal{A}(X - X_j)) \\ &= \|X - X_j\|_{\mathcal{A}}^2 + \min_{u,v} \langle uv^T, uv^T \rangle_{\mathcal{A}} - 2 \text{tr}(vu^T C_j), \end{aligned}$$

where we set $C_j := \mathcal{A}(X - X_j) = C - \mathcal{A}(X_j)$ and X is the solution (5.1). Ignoring the constant term, we thus obtain $u_{j+1}v_{j+1}^T$ from the minimization of the functional

$$J(u, v) := \langle uv^T, uv^T \rangle_{\mathcal{A}} - 2 \text{tr}(vu^T C_j). \quad (5.5)$$

Note that J is convex in each of the two vectors u, v but it is not jointly convex. This setting is well suited for the alternating linear scheme (ALS), see [OR00]. Note that a minor complication arises from the non-uniqueness in the representation of uv^T by the factors u, v : $J(u, v) = J(\lambda u, \lambda^{-1} v)$ for any $\lambda \neq 0$. In ALS, this can be easily addressed by normalizing the factor that is currently not optimized.

In the first half-iteration of ALS, we consider v with $\|v\|_2 = 1$ to be fixed and optimize for u :

$$\begin{aligned}
 \hat{u} &= \arg \min_u J(u, v) = \arg \min_u \langle uv^T, uv^T \rangle_{\mathcal{A}} - 2 \operatorname{tr}(vu^T C_j) \\
 &= \arg \min_u \sum_{q=1}^Q \operatorname{tr}(vu^T A_q uv^T B_q^T) - 2 \operatorname{tr}(vu^T C_j) \\
 &= \arg \min_u \sum_{q=1}^Q (u^T A_q u)(v^T B_q v) - 2u^T C_j v.
 \end{aligned} \tag{5.6}$$

The matrix

$$\hat{A} := \sum_{q=1}^Q (v^T B_q v) A_q \tag{5.7}$$

amounts to $(v^T \otimes I)_{\mathcal{A}}(v \otimes I)$ and thus inherits the positive definiteness from \mathcal{A} . Therefore, the solution of the unconstrained linear-quadratic optimization problem (5.6) is given by the solution of the linear system $\hat{A}\hat{u} = C_j v$.

In the second half-iteration of ALS, we fix the normalized $u \leftarrow \hat{u}/\|\hat{u}\|_2$ and optimize for v . By the same arguments, the minimizer \hat{v} is given by the solution of the linear system $\hat{B}\hat{v} = C_j^T u$, with

$$\hat{B} := \sum_{q=1}^Q (u^T A_q u) B_q. \tag{5.8}$$

The described procedure is summarized in Algorithm 6.

Algorithm 6 ALS for minimizing (5.5).

Choose random vectors u, v such that $\|v\|_2 = 1$.
while not converged **do**
 Solve linear system $\hat{A}\hat{u} = C_j v$ with \hat{A} defined in (5.7).
 Normalize $u \leftarrow \hat{u}/\|\hat{u}\|_2$.
 Solve linear system $\hat{B}\hat{v} = C_j^T u$ with \hat{B} defined in (5.8).
 Normalize $v \leftarrow \hat{v}/\|\hat{v}\|_2$.
end while

We refer to [OR00] concerning the convergence of Algorithm 6 to a local minimum of (5.5), which is not necessarily the global minimum. Let us emphasize, however, that in our setting there is no need to let Algorithm 6 converge to high accuracy and we stop it after a few iterations.

Remark 5.1. *The system matrices \hat{A} and \hat{B} in (5.7)–(5.8) are linear combinations of the coefficient matrices A_1, \dots, A_Q and B_1, \dots, B_Q , respectively. They therefore inherit the sparsity of these matrices, which allows to use a sparse direct solver [Dav06] for solving the linear systems in*

Algorithm 6. In the special case of a Lyapunov equation $AX + XA^T = C$, we have

$$\widehat{A} = A + (v^T Av)I, \quad \widehat{B} = A + (u^T Au)I.$$

Remark 5.2. Similar to the discussion in [Nou08], the procedure above can be extended to work with rank- r corrections UV^T , where $U \in \mathbb{R}^{m \times r}$ and $V \in \mathbb{R}^{n \times r}$, instead of rank-1 corrections. As before, if X_j is the current approximate solution and $C_j = C - \mathcal{A}(X_j)$, the rank- r correction $U_{j+1}V_{j+1}^T$ is computed by minimizing the following functional on $\mathbb{R}^{n \times r} \times \mathbb{R}^{m \times r}$:

$$J(U, V) = \langle UV^T, UV^T \rangle_{\mathcal{A}} - 2\text{tr}(VU^T C_j). \quad (5.9)$$

The first half-step of ALS for (5.9) then consists of fixing V (normalized to have orthonormal columns) and optimizing for U . The resulting linear system takes the form of a linear operator equation $\widehat{\mathcal{A}}(\widehat{U}) = C_j V$ with

$$\widehat{\mathcal{A}}: \mathbb{R}^{m \times r} \rightarrow \mathbb{R}^{m \times r}, \quad \widehat{\mathcal{A}}: Y \mapsto \sum_{q=1}^Q A_q Y (V^T B_q V)^T. \quad (5.10)$$

For the special case of a Lyapunov equation, we have $\widehat{\mathcal{A}}: Y \mapsto AY + Y(V^T AV)^T$. After computing a Schur decomposition of the $r \times r$ matrix $V^T AV$, the linear operator equation $\widehat{\mathcal{A}}(\widehat{U}) = C_j V$ decouples into r linear systems, see, e.g., [Sim13, Sec. 4.3].

For $Q > 2$, such a decoupling is usually impossible and one therefore has to solve an $mr \times mr$ linear system for the matrix representation $\widehat{\mathcal{A}} = \sum_{q=1}^Q V^T B_q V \otimes A_q$. The unfavorable sparsity pattern and the size of $\widehat{\mathcal{A}}$ make the application of a sparse direct solver to this linear system expensive, see [BB12] for a related discussion.

Combining Algorithm 6 with the basic iteration (5.4) for rank-1 updates leads to Algorithm 7.

Algorithm 7 Greedy rank-1 updates.

Input: Matrices $A_1, \dots, A_Q, B_1, \dots, B_Q, C$ defining a symmetric positive definite linear matrix equation (5.1), number of updates R .

Output: Rank- R approximation X_R to the solution of (5.1).

$$X_0 = 0$$

$$C_0 = C$$

for $j = 0, 1, \dots, R - 1$ **do**

 Apply Algorithm 6 with right-hand side C_j to determine rank-1 correction $u_{j+1}v_{j+1}^T$.

$$X_{j+1} \leftarrow X_j + u_{j+1}v_{j+1}^T$$

$$C_{j+1} \leftarrow C_j - \sum_{q=1}^Q A_q u_{j+1}v_{j+1}^T B_q^T$$

end for

Assuming that a fixed number of inner iterations in Algorithm 6 is used, Algorithm 7 requires the solution of $2R$ linear systems of size $m \times m$ or $n \times n$. According to Remark 5.1, these linear systems inherit the sparsity from the coefficient matrices. Note that X_R is not

stored explicitly, but in terms of its low-rank factors $[u_1, \dots, u_R] \in \mathbb{R}^{m \times R}$, $[v_1, \dots, v_R] \in \mathbb{R}^{n \times R}$. Similarly, the updated right-hand side C_j is stored implicitly, as a sum of the matrix C and j rank- Q correction terms. Note that we only need to perform matrix-vector multiplications with C_j and C_j^T . To perform this efficiently, it is sufficient that C is sparse or has moderate rank. For example, if C has rank $R_C \ll \min\{m, n\}$ and is given in factorized form, a matrix-vector multiplication with C_j can be performed in $O((m+n)r)$ operations with $r = R_C + QR$. However, in contrast to many algorithms for large-scale matrix equations [Sim13], it is not necessary that C is of (very) low rank, see Section 5.4.2 for an example.

5.1.2 Symmetric indefinite and nonsymmetric cases

In the case when \mathcal{A} is not symmetric positive definite, we use the residual norm to measure the error. Applying the derivation of Section 5.1.1 to the normal equation leads to the minimization of the functional

$$J(u, v) := \langle uv^T, uv^T \rangle_{\mathcal{A}^T \mathcal{A}} - 2 \operatorname{tr}(vu^T \mathcal{A}^T(C_j)) \quad (5.11)$$

for determining the best rank-1 correction. The symmetric positive definite linear operator $\mathcal{A}^T \mathcal{A}$ has the form

$$\mathcal{A}^T \mathcal{A} : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}^{m \times n}, \quad \mathcal{A}^T \mathcal{A} : X \mapsto \sum_{q_1, q_2=1}^Q A_{q_1}^T A_{q_2} X B_{q_2}^T B_{q_1}.$$

As before, we use ALS to address the minimization of (5.11). The first half-iteration takes the form

$$\begin{aligned} \tilde{u} &= \operatorname{argmin}_u J(u, v) = \langle uv^T, uv^T \rangle_{\mathcal{A}^T \mathcal{A}} - 2 \operatorname{tr}(vu^T \mathcal{A}^T(C_j)) \\ &= \operatorname{argmin}_u \sum_{q_1=1}^Q \sum_{q_2=1}^Q (u^T A_{q_1}^T A_{q_2} u)(v^T B_{q_2}^T B_{q_1} v) - 2 \sum_{q_1=1}^Q (u^T A_{q_1}^T C_j B_{q_1} v). \end{aligned} \quad (5.12)$$

The matrix

$$\tilde{A} := \sum_{q_1=1}^Q \sum_{q_2=1}^Q (v^T B_{q_2}^T B_{q_1} v) A_{q_1}^T A_{q_2}$$

amounts to $(v^T \otimes I) \mathcal{A}^T \mathcal{A} (v \otimes I)$ and thus inherits the positive definiteness from $\mathcal{A}^T \mathcal{A}$. Therefore, the solution of the unconstrained linear-quadratic optimization problem (5.12) is given by the solution of the linear system $\tilde{A} \tilde{u} = \sum_{q=1}^Q A_q^T C_j B_q v$.

In the second half-iteration of ALS, we fix the normalized $u \leftarrow \tilde{u} / \|\tilde{u}\|_2$ and optimize for v . By the same arguments, the minimizer \hat{v} is given by the solution of the linear system $\tilde{B} \hat{v} =$

$\sum_{q=1}^Q (A_q^T C_j B_q)^T u$, with

$$\tilde{B} := \sum_{q_1=1}^Q \sum_{q_2=1}^Q (u^T A_{q_2}^T A_{q_1} u) B_{q_1}^T B_{q_2}.$$

Using the described procedure instead of Algorithm 6 in Algorithm 7 then yields the basic greedy rank-1 algorithm for indefinite and nonsymmetric \mathcal{A} .

5.1.3 Numerical example

The approach described in Section 5.1.2 considers $\mathcal{A}^T \mathcal{A}$ instead of \mathcal{A} . This squares the condition number, which is well known to slow down convergence of classical iterative methods for solving linear systems. Our greedy low-rank methods are no exception.

To illustrate this point, we consider a generalized Lyapunov equation

$$AX + XA^T + N_1 X N_1^T = -DD^T \quad (5.13)$$

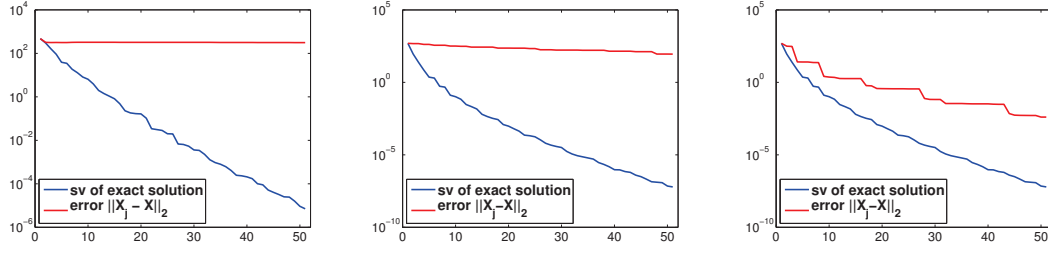
from the discretization of a 2D heat equation with bilinear boundary control, see Example 5.10 below for more details. We have used 50 discretization points in each direction, resulting in matrices of size $n = 2500$. The corresponding $n^2 \times n^2$ system matrix \mathcal{A} is symmetric, but not positive definite; it has one negative eigenvalue.

The bottom curves in the plots of Figure 5.1 show the singular values of the exact solution X for (5.13). Since the $(j+1)$ th singular value is the 2-norm error of the *best* rank- j approximation to X , the singular values represent a lower bound for the error of the iterates obtained from any greedy rank-1 algorithm. As can be seen in Figure 5.1a, Algorithm 7 based on the residual norm converges quite slowly or may even stagnate. We now modify (5.13), by dividing the matrices N_i by 2. In turn, the matrix \mathcal{A} becomes definite. As seen in Figure 5.1b, the convergence of Algorithm 7 based on the residual norm does not benefit from this modification. However, the positive definiteness allows us to use the energy norm, which significantly speeds up convergence, see Figure 5.1c. Although the error curve is still not close to the best possible convergence predicted by the singular values, this clearly shows that it is preferable to use the energy norm formulation whenever possible. However, in the indefinite case, further improvements are needed to attain satisfactory convergence.

5.1.4 Symmetry in the solution

In most of the conducted numerical experiments, we observed that ALS, Algorithm 6, converges to a symmetric solution for symmetric right-hand sides. In the following we show this property for the special case of symmetric Lyapunov equations.

In order to prove the symmetry in the solution, we first need to address the fact that the low-rank representation of the iterates in the rank- r ALS is not unique, since for any invertible



(a) Indefinite case: greedy low-rank based on residual norm (b) Definite case: greedy low-rank based on residual norm (c) Definite case: greedy low-rank based on energy norm

Figure 5.1: Convergence of basic greedy rank-1 algorithm for the generalized Lyapunov equation (5.13) arising from the discretization of 2D heat equation with bilinear boundary control.

$R \in \mathbb{R}^{r \times r}$ we have

$$UV^T = (UR)(VR^{-T})^T, \quad \forall U, V \in \mathbb{R}^{n \times r}. \quad (5.14)$$

In the following Lemma 5.3 and Theorem 5.4 we show that UV^T is symmetric positive semidefinite if and only if U and V can be chosen so that $U = V$, i.e. there exists invertible $R \in \mathbb{R}^{r \times r}$ such that $UR = VR^{-T}$.

Lemma 5.3. *Suppose we are given $U, V \in \mathbb{R}^{n \times r}$ such that $\text{rank}(U) = \text{rank}(V) = r$. There exists an invertible matrix $R \in \mathbb{R}^{r \times r}$ such that the matrices $\tilde{U} := UR, \tilde{V} := VR^{-T} \in \mathbb{R}^{n \times k}$ satisfy $\tilde{U}^T \tilde{U} = \tilde{V}^T \tilde{V}$.*

Proof. The proof follows the idea used in the balanced truncation algorithm [ASZ02] for balancing the Gramians. Let $C_U^T C_U$ and $C_V^T C_V$ be the Cholesky decompositions of the matrices $U^T U$ and $V^T V$, respectively. By computing the singular value decomposition of the matrix $C_U C_V^T$ we obtain

$$C_U C_V^T = U_\Sigma \Sigma V_\Sigma^T.$$

By setting $R := C_U^{-1} U_\Sigma \Sigma^{1/2}$ we obtain the following expressions for \tilde{U} and \tilde{V} :

$$\begin{aligned} \tilde{U} &= UR = UC_U^{-1} U_\Sigma \Sigma^{1/2}, \\ \tilde{V} &= VR^{-T} = VC_U^T U_\Sigma^T \Sigma^{-1/2}. \end{aligned}$$

We can now easily verify that \tilde{U} and \tilde{V} satisfy the statement of the lemma

$$\begin{aligned}
 \tilde{U}^T \tilde{U} &= \Sigma^{1/2} U_\Sigma^T C_U^{-T} U^T U C_U^{-1} U_\Sigma \Sigma^{1/2} \\
 &= \Sigma^{1/2} U_\Sigma^T C_U^{-T} C_U^T C_U C_U^{-1} U_\Sigma \Sigma^{1/2} = \Sigma, \\
 \tilde{V}^T \tilde{V} &= \Sigma^{-1/2} U_\Sigma^{-1} C_U V^T V C_U^T U_\Sigma^{-T} \Sigma^{-1/2} \\
 &= \Sigma^{-1/2} U_\Sigma^{-1} C_U C_V^T C_V C_U^T U_\Sigma^{-T} \Sigma^{-1/2} \\
 &= \Sigma^{-1/2} U_\Sigma^{-1} U_\Sigma \Sigma V_\Sigma^T V_\Sigma \Sigma U_\Sigma^T U_\Sigma^{-T} \Sigma^{-1/2} = \Sigma.
 \end{aligned}$$

□

Theorem 5.4. *Let $U, V \in \mathbb{R}^{n \times r}$, both of full column rank, such that $U^T U = V^T V$. If UV^* is symmetric positive semidefinite, then $U = V$.*

Proof. Let $U = Q_U R_U$ and $V = Q_V R_V$ be the QR decompositions of the matrices U and V , respectively. By construction, we have

$$R_U^T R_U = U^T U = V^T V = R_V^T R_V.$$

Since both R_U and R_V are upper triangular and the Cholesky decomposition of $U^T U$ is unique, we have that $R_U = R_V$. Given the eigenvalue decomposition of $R_U R_U^T = U_\Sigma \Sigma U_\Sigma^T$, we obtain the SVD decomposition of UV^T :

$$UV^T = Q_U R_U R_U^T Q_V^T = (Q_U U_\Sigma) \Sigma (Q_V U_\Sigma)^T. \quad (5.15)$$

Since UV^T is symmetric positive semidefinite, (5.15) is also an eigenvalue decomposition, which immediately gives $Q_U = Q_V$ and proves the theorem. □

We can now prove that every local minimum of (5.9) for the Lyapunov equation with symmetric positive definite A and positive semidefinite right-hand side C is necessarily symmetric positive semidefinite.

Lemma 5.5. *Let us consider the Lyapunov equation $AX + XA = C$, where A is symmetric positive definite and C is symmetric positive semidefinite. Let (U_*, V_*) , where $U_*, V_* \in \mathbb{R}^{n \times r}$ both have full column rank, be a local minimum of the corresponding rank- r functional $J(U, V) := \langle UV^T, UV^T \rangle_{\mathcal{A}} - 2 \operatorname{tr}(VU^T C)$. Then, the matrix $U_* (V_*)^T$ is symmetric positive semidefinite.*

Proof. Let (U_*, V_*) be a local minimum of $J(U, V)$. As seen in (5.14),

$$J(U, V) = \operatorname{tr}(V^T A V U^T U + U^T A U V^T V) - 2 \operatorname{tr}(U^T C V)$$

is invariant under rescaling: $J(U, V) = J(UR, VR^{-T})$ for every invertible $R \in \mathbb{R}^{r \times r}$. Hence, by Lemma 5.3, we may assume w.l.o.g. that $U_*^T U_* = V_*^T V_*$. Under these restrictions, Theorem 5.4 proves that $U_* V_*^T$ is symmetric positive semidefinite if and only if $U_* = V_*$.

In contradiction to the statement of the lemma, let us suppose that $U_* \neq V_*$. Since $f_{U_*}(V) := J(U_*, V)$ is strictly convex and its unique minimum is given by V_* . In particular, this implies $J(U_*, V_*) < J(U_*, U_*)$. Analogously, $J(U_*, V_*) < J(V_*, V_*)$. Adding these two inequalities, one gets

$$\begin{aligned} & 2\operatorname{tr}(U_*^T U_* V_*^T A V_* + V_*^T V_* U_*^T A U_*) - 4\operatorname{tr}(U_*^T C V_*) \\ & < 2\operatorname{tr}(U_*^T U_* U_*^T A U_* + V_*^T V_* V_*^T A V_*) - 2\operatorname{tr}(U_*^T C U_*) - 2\operatorname{tr}(V_*^T C V_*). \end{aligned}$$

Since we have $U_*^T U_* = V_*^T V_*$, this is equivalent to

$$\begin{aligned} -2\operatorname{tr}(U_*^T C V_*) & < -\operatorname{tr}(U_*^T C U_*) - \operatorname{tr}(V_*^T C V_*) \\ \Leftrightarrow 0 & < -\operatorname{tr}((U_* - V_*)^T C (U_* - V_*)), \end{aligned}$$

which leads to a contradiction, since C is positive semidefinite. \square

Remark 5.6. *The assumption in Lemma 5.5 that both U_* and V_* have full column rank is not restrictive. For $U_* V_*^T$ of rank- ℓ , with $\ell < r$, we can always find $\tilde{U}_*, \tilde{V}_* \in \mathbb{R}^{n \times \ell}$ of full rank, such that $\tilde{U}_* \tilde{V}_*^T = U_* V_*^T$ which will again be a local minimum of a reduced rank- ℓ ALS functional.*

We can use Lemma 5.5 to prove the following theorem, which establishes that in this special case, Algorithm 7 converges monotonically from below to the exact solution, providing always symmetric positive semidefinite approximate solutions. This is important, since in some applications positive definiteness of the solution is further exploited.

Theorem 5.7. *Let us consider the Lyapunov equation*

$$AX + XA = C \tag{5.16}$$

where A is symmetric positive definite and C is symmetric positive semidefinite. Assuming that Algorithm 6 always converges to a local minimum, the application of Algorithm 7 to (5.16) results in a monotonically increasing (in the Löwner ordering; see [Sio68]) sequence of approximations

$$0 = X_0 \leq X_1 \leq \dots \leq X_R \leq \dots \leq X. \tag{5.17}$$

Proof. We will prove (5.17) by induction. Initially, we have that $X_0 = 0$ and $C_0 = C$ are both symmetric positive semidefinite. Suppose that after j iterations of Algorithm (7) the approximate solution X_j and the corresponding updated right-hand side $C_j = C - AX_j - X_j A$ are both symmetric positive semidefinite. The next greedy rank-1 update $U_{j+1} V_{j+1}^T$ is a local minimizer of (5.9) for the updated equation $A(X - X_j) + (X - X_j)A = C_j$. Lemma 5.5 yields $U_{j+1} = V_{j+1}$ due to the positive (semi)definiteness of both A and C_j . In turn, the new approximate solution $X_{j+1} = X_j + U_{j+1} U_{j+1}^T \geq X_j$ is also symmetric positive semidefinite, while the updated right-hand side now has the form

$$C_{j+1} = C_j - A U_{j+1} U_{j+1}^T - U_{j+1} U_{j+1}^T A.$$

This also implies that in an ALS half-iteration with $V = U_{j+1}$ fixed, U_{j+1} is the solution of (5.10), providing the following equivalent expressions it satisfies

$$\begin{aligned} U_{j+1}U_{j+1}^T AU_{j+1} + AU_{j+1}W &= C_j U_{j+1} \\ &\Downarrow \\ (C_j U_{j+1} - U_{j+1}U_{j+1}^T AU_{j+1})W^{-1} &= AU_{j+1}, \end{aligned} \quad (5.18)$$

with $W = U_{j+1}^T U_{j+1}$. This also implies

$$\begin{aligned} U_{j+1}^T AU_{j+1}W + WU_{j+1}^T AU_{j+1} &= U_{j+1}^T C_j U_{j+1} \\ &\Downarrow \\ U_{j+1}^T AU_{j+1}W^{-1} + W^{-1}U_{j+1}^T AU_{j+1} &= W^{-1}U_{j+1}^T C_j U_{j+1}W^{-1}. \end{aligned} \quad (5.19)$$

The positive semidefiniteness of C_{j+1} now follows from

$$\begin{aligned} y^T C_{j+1} y &= y^T C_j y - y^T (C_j U_{j+1} - U_{j+1}U_{j+1}^T AU_{j+1})W^{-1}U_{j+1}^T y \\ &\quad - y^T U_{j+1}W^{-1}(C_j U_{j+1} - U_{j+1}U_{j+1}^T AU_{j+1})^T y \\ &= y^T C_j y - y^T C_j U_{j+1}W^{-1}U_{j+1}^T y - y^T U_{j+1}W^{-1}U_{j+1}^T C_j y \\ &\quad + y^T U_{j+1}U_{j+1}^T AU_{j+1}W^{-1}U_{j+1}^T y + y^T U_{j+1}W^{-1}U_{j+1}^T AU_{j+1}U_{j+1}^T y \\ &= y^T C_j y - y^T C_j U_{j+1}W^{-1}U_{j+1}^T y - y^T U_{j+1}W^{-1}U_{j+1}^T C_j y \\ &\quad + y^T U_{j+1}W^{-1}U_{j+1}^T C_j U_{j+1}W^{-1}U_{j+1}^T y \\ &= (y - U_{j+1}W^{-1}U_{j+1}^T y)^T C_j (y - U_{j+1}W^{-1}U_{j+1}^T y) \geq 0, \end{aligned}$$

where we have used (5.18) in the first equality and (5.19) in the third equality. This proves the induction step and finishes the proof. \square

Theorem 5.7 itself is of limited practical relevance, as it requires the availability of exact local minima. In practice, we stop Algorithm 6 (very) early and only obtain approximate local minima. The result of Theorem 5.7 may then still be used as a theoretical justification for choosing the subspaces U and V equal, resulting in computational savings in our main algorithm, Algorithm 9 below.

5.2 Galerkin projection

In this section, we combine greedy rank-1 updates with Galerkin projection, similarly to the techniques presented in Section 2.2.2.

After R iterations of Algorithm 7 the approximate solution takes the form

$$X_R = \sum_{j=1}^R u_j v_j^T.$$

Following the idea for accelerating the ADI method using Galerkin projection in Section 2.2.2, we consider the column space $\mathcal{U} = \text{span}(\{u_1, \dots, u_R\})$ and the row space $\mathcal{V} = \text{span}(\{v_1, \dots, v_R\})$ of X_R ($X_R \in \mathcal{U} \otimes \mathcal{V}$ by construction), and hope to obtain an improved approximation to X by choosing the best approximation from $\mathcal{V} \otimes \mathcal{U}$. For this purpose, let the columns of $U, V \in \mathbb{R}^{n \times R}$ form orthonormal bases of \mathcal{U} and \mathcal{V} , respectively. Then every element in $\mathcal{V} \otimes \mathcal{U}$ takes the form UYV^T for some $R \times R$ matrix Y .

If \mathcal{A} is symmetric positive definite, we arrive at the minimization problem

$$\begin{aligned} & \min_{Z \in \mathcal{V} \otimes \mathcal{U}} \|X - Z\|_{\mathcal{A}}^2 \\ &= \min_{Z \in \mathcal{V} \otimes \mathcal{U}} \text{tr}(X^T C) + \langle Z, Z \rangle_{\mathcal{A}} - 2 \text{tr}(Z^T C), \\ &= \min_{Y \in \mathbb{R}^{R \times R}} \text{tr}(X^T C) + \langle UYV^T, UYV^T \rangle_{\mathcal{A}} - 2 \text{tr}(VY^T U^T C) \\ &= \min_{Y \in \mathbb{R}^{R \times R}} \text{tr}(X^T C) + \text{vec}(Y)^T (V \otimes U)^T \mathcal{A} (V \otimes U) \text{vec}(Y) - 2 \text{vec}(Y)^T (V \otimes U)^T \text{vec}(C). \end{aligned}$$

This minimization problem is strictly convex and has the unique solution Y_R given by the solution of the linear system

$$\sum_{q=1}^Q (V^T \otimes U^T) (B_q \otimes A_q) (V \otimes U) \text{vec}(Y_R) = (V^T \otimes U^T) \text{vec}(C). \quad (5.20)$$

This can be viewed as a Galerkin projection of the original equation (5.1) onto the subspace $\mathcal{V} \otimes \mathcal{U}$.

If \mathcal{A} is not symmetric positive definite, minimizing the residual yields Y_R as the solution of the linear system

$$\sum_{q_1=1}^Q \sum_{q_2=1}^Q (V^T \otimes U^T) (B_{q_1} \otimes A_{q_1})^T (B_{q_2} \otimes A_{q_2}) (V \otimes U) \text{vec}(Y_R) = \sum_{q_1=1}^Q (V^T \otimes U^T) (B_{q_1} \otimes A_{q_1})^T \text{vec}(C). \quad (5.21)$$

Combining greedy rank-1 updates, Algorithm 7, with Galerkin projection yields Algorithm 8.

Remark 5.8. Both, (5.20) and (5.21), amount to solving a dense linear system of size $R^2 \times R^2$. This is performed by an LU decomposition, which requires $\mathcal{O}(R^6)$ operations and thus limits R to moderate values, say $R \leq 100$. A notable exception occurs for (5.20) when $Q = 2$. Then (5.20) is a generalized Sylvester equation and can be solved with $\mathcal{O}(R^3)$ operations [GLAM92]. For the general case, one may be able to exploit the Kronecker structure (5.20) and (5.21) by using the

Algorithm 8 Greedy rank-1 updates with Galerkin projection.

Input: Matrices $A_1, \dots, A_Q, B_1, \dots, B_Q, C$ defining a linear matrix equation (5.1), number of updates R .

Output: Rank- R approximation X_R to the solution of (5.1).

$$X_0 = 0$$

$$C_0 = C$$

for $j = 0, 1, \dots, R - 1$ **do**

Apply Algorithm 6 with right-hand side C_j to determine rank-1 correction $u_{j+1}v_{j+1}^T$.

Orthonormalize u_{j+1} w.r.t. U and append to U .

Orthonormalize v_{j+1} w.r.t. V and append to V .

$Y_{j+1} \leftarrow$ solution of the Galerkin equation (5.20) or (5.21)

$$X_{j+1} \leftarrow UY_{j+1}V^T$$

$$C_{j+1} \leftarrow C - \sum_{q=1}^Q A_q X_{j+1} B_q^T$$

end for

preconditioned conjugate gradient method. This, however, requires the availability of a good preconditioner.

5.2.1 Numerical example

We reconsider the example from Section 5.1.3, with $n = 400$ (20 discretization points in each direction) and $n = 2500$ (50 discretization points in each direction). In both cases, the corresponding operator \mathcal{A} is indefinite, and therefore the residual based formulation needs to be used. Figure 5.2 shows the convergence improvement obtained from the use of Galerkin projection. Clearly, a significant improvement sets in much earlier for $n = 400$ than for $n = 2500$.

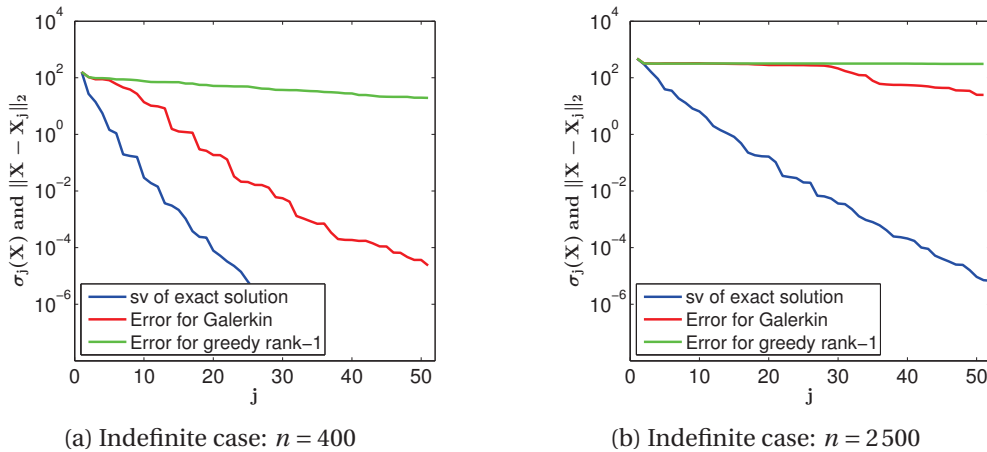


Figure 5.2: Convergence of error $\|X_j - X\|_2$ for Algorithm 8 vs. the basic greedy rank-1 algorithm applied to the generalized Lyapunov equation (5.13).

5.3 Preconditioning

The example from Section 5.2.1 shows that the use of greedy low-rank techniques and Galerkin projection is not sufficient to attain quick convergence for ill-conditioned problems. It is sometimes possible to construct an efficient preconditioner \mathcal{P} for a general linear matrix equation $\mathcal{A}(X) = C$. For example, suitable preconditioners for the generalized Lyapunov equation (5.3) can often be obtained from preconditioners for the Lyapunov operator $X \mapsto AX + XA^T$. The usual way of using preconditioners when solving linear systems consists of replacing $\mathcal{A}(X) = C$ by the preconditioned equation $\mathcal{P}^{-1}(\mathcal{A}(X)) = \mathcal{P}^{-1}(C)$. This, however, bears a major disadvantage: Assuming that \mathcal{P}^{-1} can be represented by a sum of L Kronecker products, the composition $\mathcal{P}^{-1} \circ \mathcal{A}$ is a sum of $Q \cdot L$ (instead of Q) Kronecker products. This significantly increases the cost of Algorithms 7 and 8.

In this section, we therefore suggest a different way of incorporating preconditioners, inspired by the Alternating minimal energy method (AMEn) from [DS14]. In AMEn, a low-rank approximation of the residual is used to enrich the subspaces in the Galerkin projection. Our approach follows the same idea, but uses a preconditioned residual instead of the residual. In turn, information from 1 step of the preconditioned Richardson iteration is injected into the subspaces.

The preconditioned residual in step $j + 1$ of Algorithm 8 is given by $\mathcal{P}^{-1}(C_j)$, with $C_j = C - \sum_{q=1}^Q A_q X_j B_q^T$. Of course, this matrix is not computed explicitly but represented in terms of its low-rank factors, exploiting the fact that C_j itself is given in terms of low-rank factors and \mathcal{P}^{-1} is a short sum of Kronecker products. Still, the rank of $\mathcal{P}^{-1}(C_j)$ is usually quite high and needs to be truncated further. As we will discuss in Remark 5.9 below, from a theoretical point of view it would be desirable to truncate $\mathcal{P}^{-1}(C_j)$ within a (small) prescribed accuracy. However, this may require a large rank and, thus, quickly lead to impractically large dimensions of the subspaces U and V . Following [DS14], we therefore truncate $\mathcal{P}^{-1}(C_j)$ to fixed rank, say rank 5. The matrices containing the corresponding dominant left and right singular vectors are denoted by U_{res} and V_{res} , respectively. These vectors are added to U and V before performing the Galerkin projection. In effect, the dimension of the subspaces spanned by U and V grows more quickly compared to Algorithm 8. In particular, the solution of the linear systems (5.20) or (5.21) becomes rapidly expensive, see Remark 5.8. To diminish this effect, we perform another low-rank truncation after every Galerkin projection. This requires the computation of an SVD of the (small) matrix Y_{j+1} . If possible, the tolerance for performing this truncation should be kept small, say $\text{tol} = 10^{-10}$, as it ultimately determines the accuracy of the approximate solution.

Remark 5.9. Assuming that the truncation of the preconditioned residual $\mathcal{P}^{-1}(C_j)$ is performed within a prescribed accuracy, the optimality properties of the Galerkin projection imply that Algorithm 9 converges at least as fast as the inexact steepest descent method applied to the preconditioned linear system $\mathcal{P}^{-1}(\mathcal{A}(X)) = \mathcal{P}^{-1}(C)$. As explained in more detail in [DS14, Sec 4.2], this implies linear convergence with a rate determined by the condition number of $\mathcal{P}^{-1} \circ \mathcal{A}$

Algorithm 9 Greedy rank-1 updates with Galerkin projection and preconditioned residuals.

Input: Matrices $A_1, \dots, A_Q, B_1, \dots, B_Q, C$ defining a linear matrix equation (5.1), number of updates R .

Output: Low-rank approximation X_R to the solution of (5.1).

$$X_0 = 0$$

$$C_0 = C$$

for $j = 0, 1, \dots, R - 1$ **do**

 Apply Algorithm 6 with right-hand side C_j to determine rank-1 correction $u_{j+1}v_{j+1}^T$.

 Compute approximate left/right dominant singular vectors $U_{\text{res}}, V_{\text{res}}$ of $\mathcal{P}^{-1}(C_j)$.

 Orthonormalize $[u_{j+1}, U_{\text{res}}]$ w.r.t. U and append to U .

 Orthonormalize $[v_{j+1}, V_{\text{res}}]$ w.r.t. V and append to V .

$Y_{j+1} \leftarrow$ solution of the Galerkin equation (5.20) or (5.21).

 Truncate Y_{j+1} to lower rank.

$$X_{j+1} \leftarrow UY_{j+1}V^T$$

$$C_{j+1} \leftarrow C - \sum_{q=1}^Q A_q X_{j+1} B_q^T$$

end for

and the truncation level.

5.3.1 Preconditioners

It remains to discuss examples of effective preconditioners for which \mathcal{P}^{-1} is represented as a short sum of Kronecker products. As mentioned above, we can use a preconditioners for the Lyapunov operator $X \mapsto AX + XA^T$ in the case of a generalized Lyapunov equation (5.3). As discussed in [KPT14], such preconditioners can be derived from iterative methods for solving Lyapunov equations. For our setting we consider the following two, which are presented in more detail in Remark 2.13:

1. One step of the ADI method with a single shift p

$$\mathcal{P}_{\text{ADI}}^{-1} = (A - pI)^{-1} \otimes (A - pI)^{-1}.$$

Suitable choices for p are discussed in, e.g., [BS13]. For the case of a symmetric A , the optimal p equals $\sqrt{\lambda_{\max}(A)\lambda_{\min}(A)}$.

2. One step of the sign function iteration for Lyapunov equations gives rise to the preconditioner

$$\mathcal{P}_{\text{sign}}^{-1} = \frac{1}{2c}(I \otimes I + c^2 A^{-1} \otimes A^{-1}), \quad (5.22)$$

with the scaling factor $c = \sqrt{\frac{\|A\|_2}{\|A^{-1}\|_2}}$.

The application of $\mathcal{P}_{\text{ADI}}^{-1}$ and $\mathcal{P}_{\text{sign}}^{-1}$ to a matrix of rank ℓ requires the solution of 2ℓ linear systems with the (shifted) matrix A . To optimize this step, the LU factors are computed only

once and reused in every iteration.

5.3.2 Numerical example

Figure 5.3 shows the convergence of Algorithm 9 for the example from Sections 5.1.3 and 5.2.1 for $n = 2500$. We used the preconditioner $\mathcal{P}_{\text{sign}}^{-1}$ from (5.22). The convergence, compared to Algorithm 5.2, clearly improves, to the extent that the method becomes practical for this example. This comes at the expense of a faster increase of the rank, which makes the Galerkin projection more expensive. To limit this increase, we apply a more aggressive truncation strategy and cap the rank at 50. This procedure is explained in more detail in Section 5.4 below.

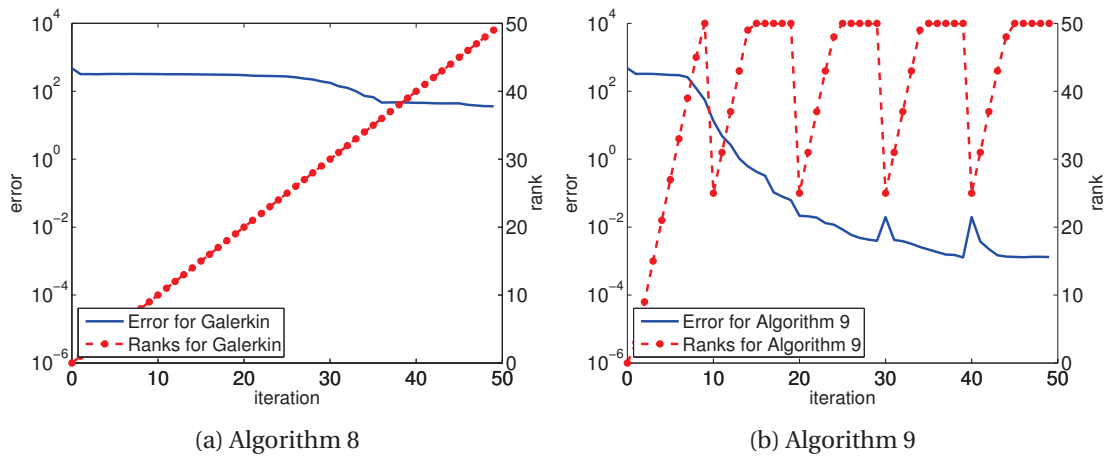


Figure 5.3: Convergence of error $\|X_j - X\|_2$ and ranks of X_j for Algorithms 8 and 9 applied to the generalized Lyapunov equation (5.13).

5.4 Numerical experiments

In this section, we first report on the performance of Algorithm 9 for a number of large-scale examples available in the literature and then we perform a more detailed study of the impact of the individual parts of our algorithm on its performance. Algorithm 9 has been implemented and tested in the same computing environment already described in Section 3.5.

Unless stated otherwise, we have made the following choices in the implementation of Algorithm 9:

ALS iterations. The number of ALS iterations (see Algorithm 6) in the greedy rank-1 procedure is fixed to 5.

Preconditioner. The sign function based preconditioner $\mathcal{P}_{\text{sign}}^{-1}$ from (5.22) is used.

Truncation of residual. The preconditioned residual $\mathcal{P}^{-1}(C_j)$ is replaced by its best rank-5 approximation. This truncation is performed by combining QR decompositions with an

SVD, exploiting the fact that the rank of $\mathcal{P}^{-1}(C_j)$ is not full but given by the product of $\text{rank}(C_j)$ with the Kronecker rank of \mathcal{P}^{-1} (which is 2 for $\mathcal{P}_{\text{sign}}^{-1}$).

Truncation of iterates. As explained in Section 5.3, we truncate Y_{j+1} to lower rank such that all singular values below the relative tolerance $\text{tol} = 10^{-10}$ are neglected and the maximal rank maxrank is never exceeded. This strategy bears the risk that little new information can be added once maxrank is reached. To avoid this, we have implemented a restart strategy when this happens: Every 10 iterations the current approximation is truncated more aggressively to rank $0.6 \times \text{maxrank}$.

In all experiments below, we measure the convergence of Algorithm 9 by computing the relative residual norm

$$\|C - \mathcal{A}(X_j)\|_F / \|C\|_F.$$

5.4.1 Generalized Lyapunov equations

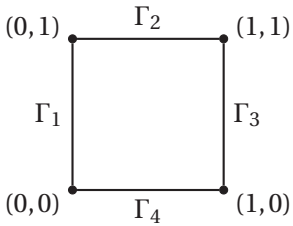
Generalized Lyapunov equations typically arise from bilinear control problems of the form

$$\dot{x}(t) = Ax(t) + \sum_{q=1}^Q N_q x(t) u_q(t) + Du(t), \quad x(0) = x_0, \quad (5.23)$$

with the state vector $x(t) \in \mathbb{R}^n$ and the control $u(t) \in \mathbb{R}^\ell$. The controllability Gramian [BD11] for (5.23) plays an important role in model reduction of bilinear systems and is given by the solution of the generalized Lyapunov equation (5.3).

In the following, we consider two examples of bilinear control systems, a bilinear boundary control problem and the Carleman bilinearization of an RC circuit.

Example 5.10. Following [BB13, Dam08], we consider the heat equation on the unit square with bilinear boundary control: where $\Gamma_1, \Gamma_2, \Gamma_3, \Gamma_4$ are the boundaries of $]0, 1[^2$. After a stan-

$$\begin{aligned} \frac{\partial}{\partial t} z &= \Delta z && \text{in }]0, 1[^2, \\ \vec{n} \cdot \nabla z &= 0.5 \cdot u \cdot (z - 1) && \text{on } \Gamma_1, \\ z &= 0 && \text{on } \Gamma_2, \Gamma_3, \Gamma_4, \end{aligned}$$


dard finite difference discretization, the controllability Gramian is obtained as the solution of the generalized Lyapunov equation

$$AX + XA^T + N_1 X N_1^T = -DD^T, \quad (5.24)$$

where $A \in \mathbb{R}^{n \times n}$ is the discretization of the 2D Laplace operator. The matrices N_1, D arise from the Neumann boundary control on Γ_1 and therefore have $O(\sqrt{n})$ nonzero columns. The

corresponding $n^2 \times n^2$ system matrix $\mathcal{A} = I \otimes A + A \otimes I + N_1 \otimes N_1$ turns out to be symmetric, but indefinite; most of its eigenvalues are negative and only a few are positive.

The convergence of Algorithm 9 for $n = 10\,000$ and the maximal rank $\text{maxrank} = 90$ is shown in Figure 5.5. The execution time spent per iteration significantly increases as the size of

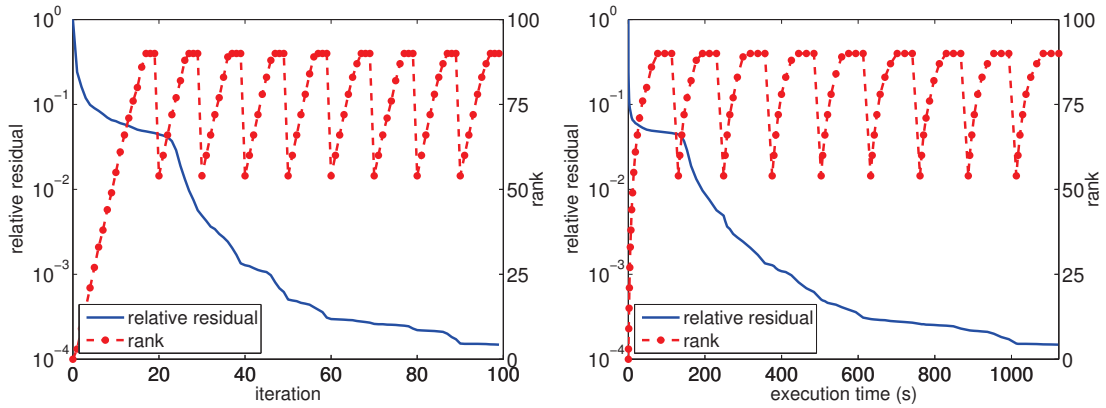


Figure 5.5: Convergence of relative residual norm for Algorithm 9 applied to Example 5.10 (indefinite case).

the subspaces \mathcal{U} and \mathcal{V} grows, mainly due to the increased cost of constructing and solving the Galerkin system (5.21) and partly due to the orthogonalization that has to be performed. When increasing n further, we would need to work with even larger values of maxrank to attain reasonable convergence.

Inspired by the experiments in [BB13], we consider a slight modification of this example, dividing the matrices N_i by 2. In turn, the matrix \mathcal{A} becomes definite and Algorithm 9 can be based on the energy norm. Also, the singular value decay of X appears to improve. Figure 5.6 shows the obtained results for $n = 250\,000$. Even though n is larger than in Figure 5.5, Algorithm 9 converges significantly faster and attains a higher accuracy with the same maximal rank.

For both examples, the convergence of Algorithm 9 is clearly sublinear. This appears to be typical for algorithms based on greedy low-rank strategies, see, e.g., [CEL11].

Compared to the results for $n = 562\,500$ reported in [BB13] for the preconditioned CG with low-rank truncation, our algorithm seems to perform slightly worse in terms of attainable accuracy vs. the rank of the approximate solution. \diamond

Example 5.11. This example is taken from [BS06] and concerns a scalable RC ladder with n_0 resistors described by

$$v_t = f(v) + bu(t), \tag{5.25}$$

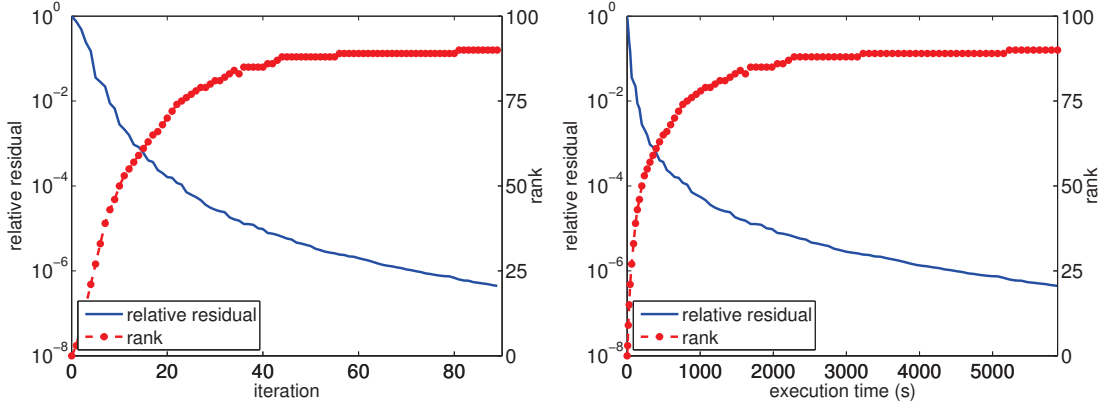


Figure 5.6: Convergence of relative residual norm for Algorithm 9 applied to Example 5.10 (definite case).

where

$$f(v) = \begin{pmatrix} -g(v_1) - g(v_1 - v_2) \\ g(v_1 - v_2) - g(v_2 - v_3) \\ \vdots \\ g(v_{n_0-1} - v_{n_0}) \end{pmatrix}, \quad g(v) = \exp(40v) + v - 1.$$

Using Carleman bilinearization, the nonlinear control problem (5.25) can be approximated by a bilinear control problem of the form (5.23). In turn, we obtain a generalized Lyapunov equation

$$AX + XA^T + NXN^T = -DD^T$$

with $X \in \mathbb{R}^{(n_0+n_0^2) \times (n_0+n_0^2)}$ and

$$A = \begin{bmatrix} A_0 & A_1 \\ 0 & I \otimes A_0 + A_0 \otimes I \end{bmatrix},$$

and A_0 is a tridiagonal matrix and A_1 arises from the coupling of first and second order terms.

According to our experiments, it is beneficial for this example to skip the greedy rank-1 procedure entirely and only include information from the preconditioned residual in U and V . The resulting convergence for $n_0 = 500$, that is $n = 250500$, and $\text{maxrank} = 70$ is displayed in Figure 5.7. The algorithm converges quickly to an accuracy below 10^{-3} , after which the convergence slows down due to imposed limit on the subspace size.

For reference, we also include the results for a modification discussed in [BB13], where the matrix N is divided by 2. Figure 5.8 shows nearly the same convergence behavior. Compared to the results reported in [BB13], the convergence of our algorithm is significantly faster until the imposed limit on the subspace size is reached. \diamond

Chapter 5. Greedy low-rank approach to linear matrix equations

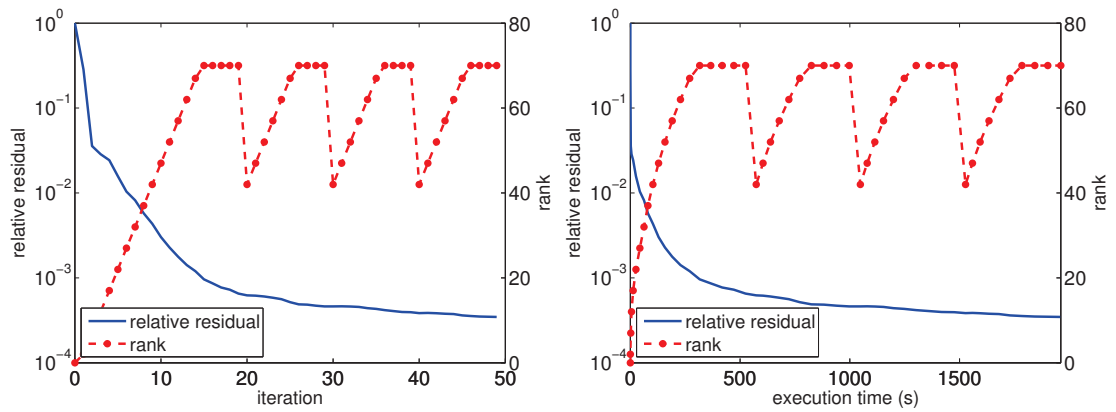


Figure 5.7: Convergence of relative residual norm for Algorithm 9 (without greedy rank-1) applied to Example 5.11.

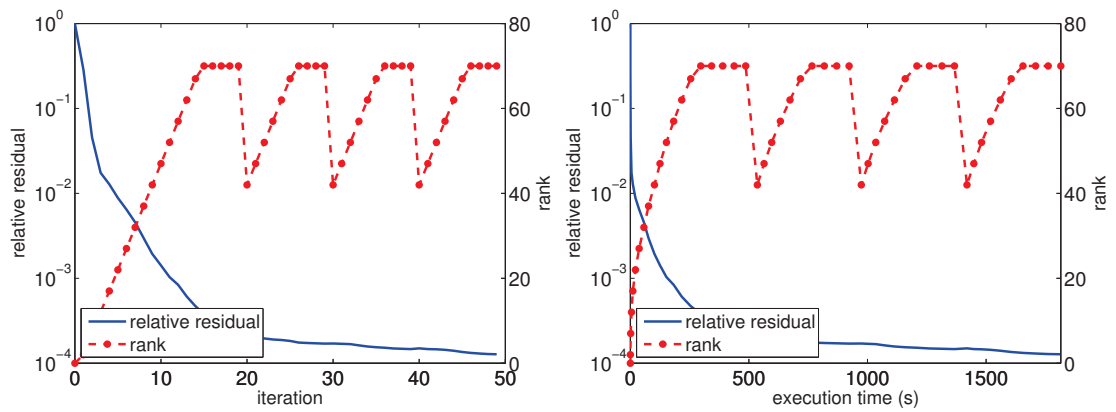


Figure 5.8: Convergence of relative residual norm for Algorithm 9 (without greedy rank-1) applied to Example 5.11 with N replaced by $N/2$.

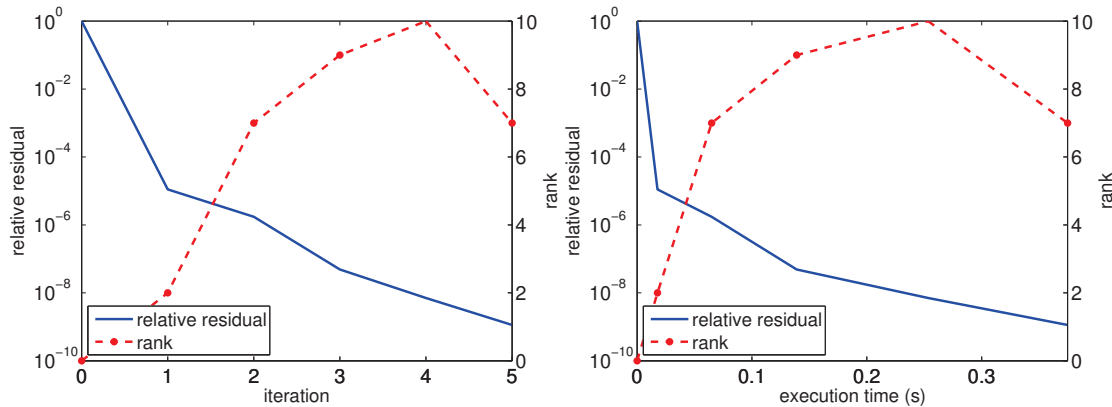


Figure 5.9: Convergence of relative residual norm for Algorithm 9 applied to Example 5.12.

The following example is concerned with a stochastic control problem.

Example 5.12. This example is taken from [BB13] and arises from the control of a dragged Brownian particle, whose motion is described by the Fokker-Planck equation. We refer to [HSBTZ13] for a detailed explanation of this example. After discretization, the resulting generalized Lyapunov equation has size $n = 10000$ and is of the form (5.24). The matrix N_1 is sparse and has full rank 10000, while the right-hand side has rank 1.

As can be seen in Figure 5.9, Algorithm 9 converges quickly for this example and requires less than 0.5 seconds to attain an accuracy below 10^{-8} . According to [BB13, Table 1], the preconditioned BiCG with low-rank truncation requires around 10 seconds for the same example in a computing environment that is comparable to ours. \diamond

5.4.2 Lyapunov equation with right-hand sides having a singular value decay

As mentioned in the introduction, one of the most important special cases of (5.1) is the Lyapunov equation

$$AX + XA^T = C, \quad (5.26)$$

where $A, C \in \mathbb{R}^{n \times n}$. There are numerous numerical methods that specifically target (5.26), see [BS13, Sim13]. For large-scale equations, most existing strategies crucially depend on a low-rank right-hand side, that is

$$C = -DD^T, \quad \text{with } D \in \mathbb{R}^{n \times \ell}, \quad \ell \ll n.$$

In particular this is the case for methods that make use of Krylov subspaces based on A and D . The dimension of these subspaces grows proportionally with ℓ , rendering these techniques impractical for larger values of ℓ .

In contrast, Algorithm 8 does not require such a low-rank assumption on the right-hand side

to perform efficiently; we only need to be able to perform fast matrix-vector multiplications with C . Of course, Algorithm 8 can only attain reasonable convergence if the solution X has a strong singular value decay. For this purpose, it is not necessary that C has low rank. As the following example demonstrates, it sometimes suffices that C has a (possibly weaker) singular value decay.

Example 5.13. Consider the 2D Poisson equation on the unit square:

$$\begin{aligned} \Delta u(\xi) &= f(\xi), & \xi \in \Omega &=]-1, 1[^2 \\ u(\xi) &= 0 & \xi \in \partial\Omega. \end{aligned}$$

The standard finite difference discretization with n grid points in each coordinate yields an $n^2 \times n^2$ linear system of the form

$$(L \otimes I + I \otimes L) \text{vec}(X) = \text{vec}(F),$$

where L is the discretization of the 1D Laplace operator and F contains the values of f at the grid points. This is equivalent to the Lyapunov equation

$$LX + XL^T = F.$$

In our experiments, we have used $f(\xi_1, \xi_2) = \exp((\xi_1^p + \xi_2^p)^{\frac{1}{p}})$ with $p = 10$ and $n = 40000$. This results in a matrix F with a relatively slow singular value decay. There are several established techniques to multiply with such a matrix F implicitly and efficiently. For simplicity, we have used ACA (Adaptive Cross Approximation [BR03]) to replace F with a matrix of rank $\ell = 92$, which corresponds to an error indicator of 9.7×10^{-8} in ACA. The resulting convergence of Algorithm 8 (with 3 ALS iterations in Algorithm 6) is shown in Figure 5.10. The observed

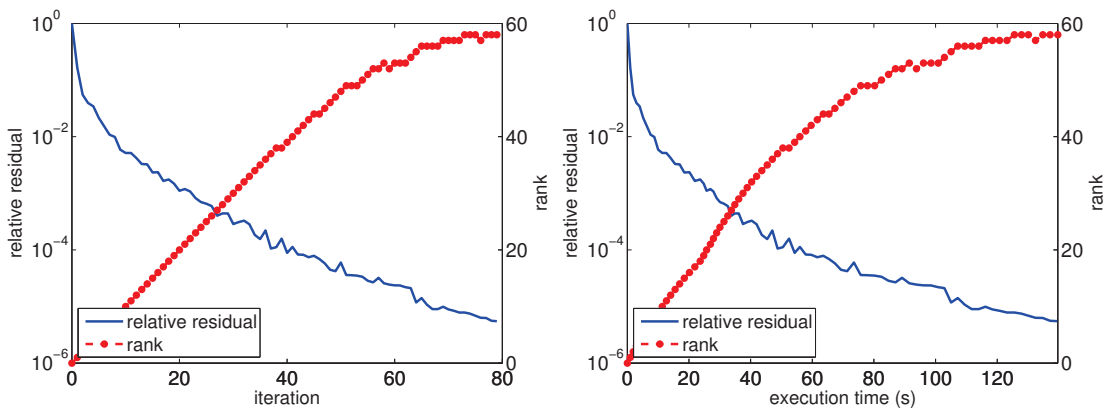


Figure 5.10: Convergence behavior for the Lyapunov equation arising from 2D Poisson equation with non-low-rank righthand side.

execution times are very small compared to majority of other examples, due to the fact that each iteration only requires the solution of $n \times n$ tridiagonal linear systems and a small-scale Sylvester equation. \diamond

5.4.3 Detailed numerical study of components of Algorithm 9

The purpose of the following numerical experiments is to consider the different parts of Algorithm 9 separately and assess their impact on its performance.

Greedy rank-1 updates vs. preconditioned residuals

In Algorithm 9, the bases U and V are enriched by adding information from the greedy rank-1 update and the preconditioned residual. The purpose of the following three experiments is to assess the impact of these two enrichment steps separately. Except for the third experiment, we always truncate the preconditioned residuals to rank 1, so that only 1 vector is added to each basis, equilibrating with the enrichment gained from the greedy rank-1 update. Truncating to rank 1 instead of a higher rank also has the advantage that it enables us to essentially turn off low-rank truncation (we only truncate singular values below 10^{-14}).

Example 5.14. We first consider a synthetic, well-conditioned example of (5.1) for $Q = 3$ and $n = 3000$. The coefficient matrices are given by

$$A_i = \frac{R_i + R_i^T}{2} + \frac{n}{8} I_n, \quad B_i = \frac{S_i + S_i^T}{2} + \frac{n}{8} I_n, \quad i = 1, \dots, 3,$$

and $C = e_1 e_1^T$, where the matrices R_i and S_i are random matrices generated with the Matlab function `randn`. No preconditioner is used; $\mathcal{P} = I$. From the numerical results shown in Figure 5.11, it can be concluded, for this particular example, that the enrichment gained from greedy rank-1 updates is much more significant compared to the residuals. However, the approach using just the residuals is significantly faster due to the fact that the residual does not require the solution of (dense) linear systems unlike for the greedy rank-1 updates. \diamond

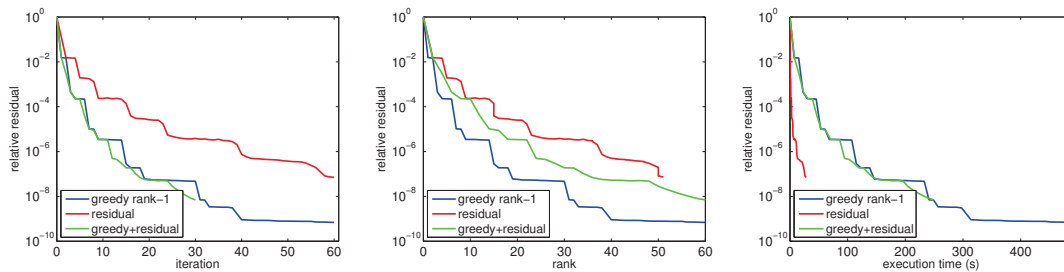


Figure 5.11: Convergence of Algorithm 9 with different enrichment strategies for Example 5.14.

Example 5.15. We consider the generalized Lyapunov equation from Example 5.10 for a

modest size, $n = 2500$, and use the standard sign-function based preconditioner. As can be seen in Figure 5.12, greedy rank-1 updates are still more important than preconditioned residuals but, in contrast to Example 5.14, combining both approaches yields a significant convergence improvement. \diamond

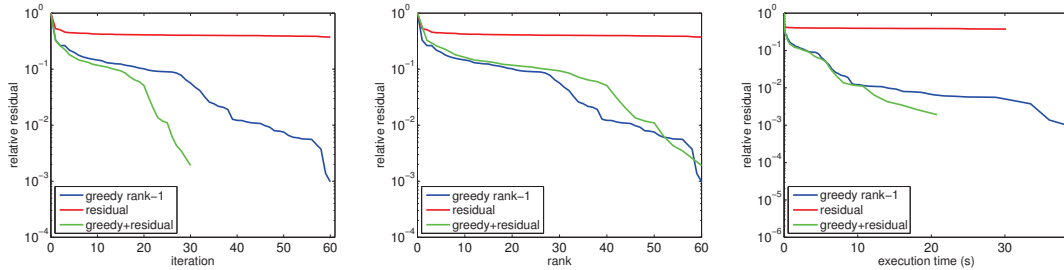


Figure 5.12: Convergence of Algorithm 9 with different enrichment strategies for Example 5.15.

Example 5.16. We consider the generalized Lyapunov equation from Example 5.11 with $n = 2550$, and the standard preconditioner. In contrast to the previous example, we have observed that truncating the preconditioned residuals to rank 2 instead of rank 1 has a non-negligible impact on the convergence. To illustrate this effect, we have used rank-2 truncation when only the preconditioned residuals are included in the enrichment (red curves in Figure 5.13). It turns out that this yields better convergence and requires less time compared to combining greedy rank-1 updates and rank-1 truncated preconditioned residuals (green curves in Figure 5.13), let alone using only greedy rank-1 updates (blue curves in Figure 5.13). \diamond

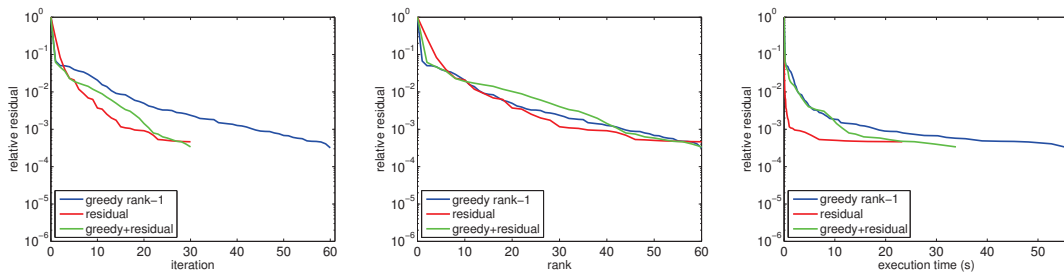


Figure 5.13: Convergence of Algorithm 9 with different enrichment strategies for Example 5.16.

From the three experiments above, no clear conclusion can be drawn. For some examples the greedy rank-1 updates constitute the most important parts of the subspaces, while for others the preconditioned residuals become more important.

Low-rank truncation

As explained in the beginning of this section, the size of the bases U and V is kept under control by a low-rank truncation of the current approximation. All singular values below the relative tolerance $\text{tol} = 10^{-10}$ are neglected and the rank is limited to the maximal rank maxrank . The purpose of the following experiment is to assess the impact of the latter truncation criterion on the overall performance. To limit the overall rank growth, we always truncate the preconditioned residuals to rank 1. Two vectors are added to each basis in each iteration, one from the greedy rank-1 update and one from the (preconditioned) residual. We compare the implementation of Algorithm 9 with $\text{maxrank} = \infty$ (we only truncate singular values below 10^{-12}) and with $\text{maxrank} = 45$, combined with restarting every 16 iterations.

Example 5.17. We consider the generalized Lyapunov equation from Example 5.10 with $n = 1600$, and the standard preconditioner. From the numerical results shown in Figure 5.14, we observe the expected effect that truncation slows down convergence. On the other hand, it can be clearly seen from Figure 5.14c that the implementation with truncation produces good results in significantly smaller amounts of time. \diamond

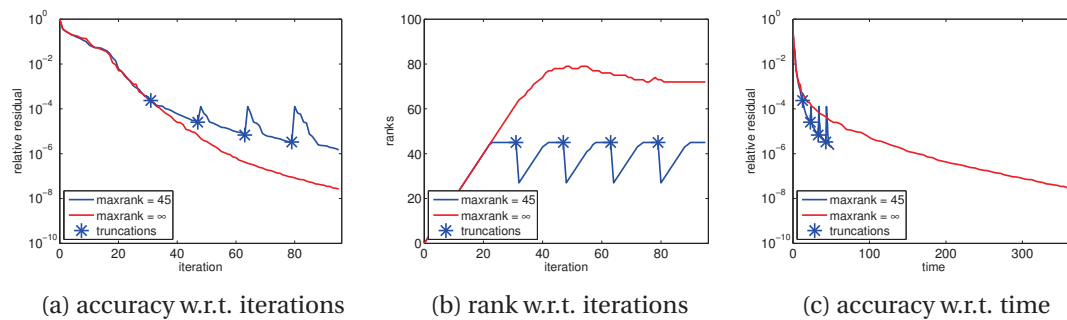


Figure 5.14: Convergence of Algorithm 9 with truncation based on $\text{maxrank} = 45$ vs. $\text{maxrank} = \infty$ for Example 5.17.

5.5 Conclusion

We have developed a novel low-rank approach for addressing large-scale linear matrix equations, based on greedy low-rank updates, Galerkin projection and residual preconditioning.

In principle, greedy low-rank methods are applicable to any linear matrix equation whose solution can be well approximated with a low-rank matrix. However, in practice, these methods need to be combined with Galerkin projection and preconditioning strategies in order to ensure satisfactory convergence speed for a wider range of applications. We have further improved the performance of our approach by incorporating low-rank truncation and a restarting strategy.

The resulting solver, Algorithm 9, is demonstrated to perform quite well for problems that have been discussed earlier in the literature, especially those for which the imposed limit on

Chapter 5. Greedy low-rank approach to linear matrix equations

the subspace size is not reached. For more challenging problems that feature larger ranks, the need for constructing and solving the Galerkin systems (5.20)–(5.21) may become a bottleneck. One way to overcome this is to stop our method when a certain rank is reached, and use the approximate result as an initial guess for the iterative methods discussed in [BB13].

6 Low-rank approach to parameter dependent symmetric Lyapunov equations

Suppose we are given a large-scale control system, similar to the one considered in Section 2.2.3, where the system matrices $E, A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$ and $C \in \mathbb{R}^{l \times n}$ additionally depend on a parameter $\mu \in D \subset \mathbb{R}^d$:

$$\begin{aligned} E(\mu)\dot{x}(t) &= A(\mu)x(t) + B(\mu)u(t), \\ y(t) &= C(\mu)x(t). \end{aligned} \tag{6.1}$$

Such systems frequently arise from the spatial discretization of partial differential equations (PDEs), in which the parameters are used to model the variation of geometries and material properties to be optimized. As discussed in Section 2.2.3, simulation of (6.1), for a fixed $\mu \in D$, can be accelerated by computing a reduced-order model. However, in a parameterized setting, it is often essential to enable fast simulations for many different parameter values in D , and computing a reduced-order model for each of them separately is usually computationally too costly.

In the last two decades, several model reduction approaches have been developed for parameterized systems. These include multivariate moment matching based on (rational) Krylov subspaces, interpolation-based techniques, and the reduced basis method. None of these methods can be considered as optimal. In this chapter, we develop a model reduction approach based on the balanced truncation, presented in more detail in Section 2.2.3. As previously discussed, computation of the reduced-order model for a fixed $\mu \in D$ using the balanced truncation, requires computation of Gramians $P(\mu)$ and $Q(\mu)$, which are defined as the unique symmetric positive semidefinite solutions of the Lyapunov equations

$$A(\mu)P(\mu)E(\mu)^T + E(\mu)P(\mu)A(\mu)^T = -B(\mu)B(\mu)^T, \tag{6.2}$$

$$A(\mu)^T Q(\mu)E(\mu) + E(\mu)^T Q(\mu)A(\mu) = -C(\mu)^T C(\mu), \tag{6.3}$$

respectively, provided that $E(\mu)$ is nonsingular and all eigenvalues of the matrix pencil $A(\mu) - \lambda E(\mu)$ have negative real part. Thus, in order to determine the reduced-order models for many parameter values, we need to compute (approximate) solutions of the Lyapunov equations

(6.2) and (6.3) for all these parameters. In this chapter, we present a low-rank reduced basis method for approximating solutions to parameter-dependent Lyapunov equations on the whole parameter domain.

The rest of this chapter is largely based on [KSSS14] and is organized as follows. In Section 6.1, we first give a short overview of parametric model reduction. In Section 6.2, we discuss using the reduced basis method for solving Lyapunov equations and present our low-rank reduced basis approach for solving (6.2). In particular, we discuss the choice of the solution subspace and the *a posteriori* error estimates. In Section 6.3, we present the full algorithm and discuss implementational details, while in Section 6.4 we present the performance of the approach on a few numerical experiments.

6.1 Parametric model reduction

In this section, we present a short overview, largely based on the recent survey paper [BGW15], of the most important challenges arising in the parametric model reduction, along with some of the approaches to overcome them.

As explained in Section 2.2.3, for a fixed value of $\mu \in D$, a common approach to calculating a reduced model for (6.1) is to compute low-dimensional subspaces $\mathcal{V}(\mu), \mathcal{W}(\mu)$ and their bases $V(\mu), W(\mu) \in \mathbb{R}^{n \times r}$ using a model reduction technique, such as the balanced truncation, and project the system matrices onto $V(\mu)$ and $W(\mu)$. However, computation of $V(\mu)$ and $W(\mu)$ using the balanced truncation involves solving Lyapunov equations of the form (6.2) and (6.3), which is for large values of n computationally feasible only for a few different parameter values $\mu \in D$.

Suppose that we have computed the reduced-order models and their projection bases $V(\mu_i) \in \mathbb{R}^{n \times r}$ and $W(\mu_i) \in \mathbb{R}^{n \times r}$ for a few parameter samples μ_1, \dots, μ_M . Given this information, a question which naturally arises is, whether it is possible to approximate reduced-order models all over the parameter domain D , and if yes, how to do it efficiently. Some of the approaches that have already been discussed in the literature include:

Global bases. Large computational cost of computing different projection subspaces for each $\mu \in D$ can be avoided by instead constructing global bases V and W out of sampled local bases

$$W = \text{span}\{W(\mu_1), \dots, W(\mu_M)\}, \quad V = \text{span}\{V(\mu_1), \dots, V(\mu_M)\}.$$

Global bases W and V can be efficiently constructed, for example, using the (truncated) singular value decomposition (SVD). For each $\mu \in D$, this allows computation of reduced-order models by simply projecting the system matrices onto V and W . Examples where global bases have been used include moment matching techniques, see e.g. [BF14], as well as the interpolation-based techniques, see e.g. [BBBG11].

Interpolation of local bases. Depending on the problem, the global bases may become im-

practically large, resulting in reduced-order models which are not much smaller than the original. Instead, as explained in [AF08], good approximate local bases $V(\mu)$ and $W(\mu)$ for any given $\mu \in D$ can be computed by interpolating the sampled local bases $U(\mu_i)$ and $V(\mu_i)$, $i = 1, \dots, M$ on a tangent space to a Grassmanian manifold of all r -dimensional subspaces in \mathbb{R}^n . Implementing this approach first requires mapping of the sampled local bases to the tangent space at a reference point, which consists of computing $O(M)$ SVDs of $n \times r$ matrices. Now, computing approximate local bases $V(\mu)$ and $W(\mu)$ for any given $\mu \in D$ consists of interpolating at the tangent space, and mapping the result back to the space of all r -dimensional subspaces in \mathbb{R}^n , which requires a thin SVD computation. Finally, having computed local bases $V(\mu)$ and $W(\mu)$, reduced-order models are again obtained by simply projecting the system matrices onto $V(\mu)$ and $W(\mu)$.

Interpolation of reduced models. Computation of reduced-order models from approximate local bases still requires projecting system matrices onto them. As discussed in [AF11, PMEL10], this can be avoided by interpolating the exactly computed reduced-order models at μ_1, \dots, μ_M . Implementing this approach first requires *congruence transformation* of the sampled local bases so that the computed reduced-order models are expressed in the same generalized coordinate system, which can be achieved by solving $O(M)$ Procrustes problems. After this transformation, for any given $\mu \in D$, the reduced system matrices can either be interpolated directly [PMEL10], or on a manifold [AF11], similarly to the interpolation of local bases above.

For a more detailed discussion on these and other approaches, we refer to [BGW15].

It is important to note that in case of general parametric dependence in system matrices, using either global or locally interpolated bases is not computationally viable. As evaluating the reduced models first requires explicit computation of $A(\mu)$ and then, subsequently, multiplication with V and W , the computational cost of these approaches heavily depends on the dimension n of the original problem. However, it is possible to overcome this issue in certain special cases, such as when $A(\mu)$ (and other system matrices) admits affine linear decomposition w.r.t. μ :

$$A(\mu) = \theta_1(\mu)A_1 + \dots + \theta_Q(\mu)A_Q, \quad (6.4)$$

with $\theta_1, \dots, \theta_Q : D \rightarrow \mathbb{R}$ and $A_1, \dots, A_Q \in \mathbb{R}^{n \times n}$. As discussed in Section 2.3.4, if $Q \ll n^2$, pre-computing the matrices $W^T A_q V$, $q = 1, \dots, Q$ enables constructing of $A_r(\mu)$ in the following way

$$\tilde{A}(\mu) = \theta_1(\mu)W^T A_1 V + \dots + \theta_Q(\mu)W^T A_Q V, \quad \forall \mu \in D,$$

with a computational cost that is independent of n . Moreover, as discussed in Remark 2.15, even if the system matrices do not explicitly satisfy (6.4), methods such as MDEIM can sometimes be used to find accurate approximations of system matrices which admit affine linear decomposition.

6.2 Reduced basis method for Lyapunov equations

In this section, we will sketch how the reduced basis (RB) method, described in more details in Section 2.3, can be used to derive a global bases approach for solving Lyapunov equations. From now on, we consider Lyapunov equations of the form

$$-A(\mu)X(\mu)E - EX(\mu)A(\mu) = BB^T, \quad (6.5)$$

where we assume that

1. neither E nor B depend on the parameters;
2. E is symmetric, positive definite and $A(\mu)$ is symmetric, negative definite for all $\mu \in D$;
3. $A(\mu)$ admits affine linear decomposition w.r.t. μ (6.4).

The purpose of the first assumption is mainly to simplify the discussion. Our algorithms can be trivially extended to parameter-dependent E and B . On the other hand, as explained in Section 2.3, the last two assumptions are essential for the RB method. The second assumption is central to the error estimators we are using since it ensures that the system (6.5) is positive definite, and as discussed in Remark 2.18, dropping it would require the use of different estimators. The third assumption enables efficient construction of the compressed linear systems as well as the residual computation. As already discussed in Remark 2.15, the third assumption is commonly found in the literature; in many problems it either arises naturally or can be attained by techniques such as Empirical Interpolation method [BMNP04].

As explained in Section 2.2, the Lyapunov equation (6.5) can be equivalently written as the $n^2 \times n^2$ linear system

$$\mathcal{A}(\mu)x(\mu) = b, \quad (6.6)$$

with $\mathcal{A}(\mu) = -E \otimes A(\mu) - A(\mu) \otimes E$, $b = \text{vec}(BB^T)$ and $x(\mu) = \text{vec}(X(\mu))$. When applying the plain reduced basis (RB) method directly to the $n^2 \times n^2$ linear system (6.6), without exploiting the particular structure of $\mathcal{A}(\mu)$, the high dimensionality n^2 of the solution space leads to inefficiencies. For example, after M iterations of the offline phase, the complexity for orthonormalizing and storing V is $O(n^2 M^2)$ and $O(n^2 M)$, respectively. This puts a limitation on the size n that can be handled; n can be at most a few thousand. Another major disadvantage is that the approximate solution $\tilde{X}(\mu)$ obtained in the online phase is *not* guaranteed to be positive semidefinite, a property that is highly desirable in model reduction applications.

In the following, we show how the RB method can be modified and accelerated for the Lyapunov equation (6.5) by exploiting the structure in $\mathcal{A}(\mu)$ and by making use of low-rank properties of $X(\mu)$.

6.2.1 Low-rank structure in the offline phase

As already discussed in Section 2.2.1, when $m \ll n$, it is known that the singular values of the solution $X(\mu)$ to (6.5) decay very quickly. Hence, $X(\mu)$ can be well approximated by a low-rank matrix of the form $L(\mu)L(\mu)^T$, where the low-rank Cholesky factor $L(\mu)$ has $m_\mu \ll n$ columns.

The offline phase proceeds as described in Section 2.3.3, with the notable difference that for each sample μ_1, \dots, μ_M instead of the $n \times n$ (or $n^2 \times 1$ when vectorized) solutions $X(\mu_j)$ we collect the low-rank Cholesky factors $L(\mu_j) \in \mathbb{R}^{n \times m_j}$ in a subspace

$$\mathcal{U}_M = \text{range}(L_M) \subset \mathbb{R}^n,$$

where $L_M = [L(\mu_1), L(\mu_2), \dots, L(\mu_M)] \in \mathbb{R}^{n \times (m_1 + \dots + m_M)}$. In the absence of approximation error, we have $\mathcal{V}_M \subset \mathcal{U}_M \otimes \mathcal{U}_M$, where \mathcal{V}_M is the subspace containing vectorized snapshots, as in the plain RB method. For an orthonormal basis U of \mathcal{U}_M , this means that any vector $v \in \mathcal{V}_M$ can be represented as

$$v = (U \otimes U) \text{vec}(Y) = \text{vec}(UYU^T), \quad (6.7)$$

for some (small) matrix Y . The dimension of \mathcal{U}_M is given by

$$N = m_1 + \dots + m_M,$$

which can be – depending on m – significantly larger than M , the dimension of \mathcal{V}_M . Similarly to the discussion in Section 6.1 on the applicability of global bases as they increase in size, it is imperative to keep the dimensionality of \mathcal{U}_M under control. To attain a smaller value for N , while building the basis U , we repeatedly apply column compressions to matrix L_M using the truncated singular value decomposition (SVD) with relative truncation tolerance $\text{tol}_{\text{compr}}$.

The offline phase is stopped either after the maximum number of iterations M_{max} or when a satisfactory level of accuracy has been attained:

$$\Delta_M(\mu) < \varepsilon_{\text{LRRB}}, \quad \forall \mu \in D,$$

where $\Delta_M(\mu)$ are the error estimates to be defined in Section 6.2.3 and $\varepsilon_{\text{LRRB}}$ is the prescribed absolute tolerance on the values of error estimates.

6.2.2 Low-rank structure in the online phase

The online phase proceeds as described in Section 2.3.4, with the subspace \mathcal{V}_M replaced by $\mathcal{U}_M \otimes \mathcal{U}_M$. In view of (2.35) and (6.7), this means that we have to solve the compressed linear system

$$((U \otimes U)^T \mathcal{A}(\mu)(U \otimes U))y(\mu) = (U \otimes U)^T b,$$

which is equivalent to solving the compressed Lyapunov equation

$$-\tilde{A}(\mu)Y(\mu)\tilde{E} - \tilde{E}Y(\mu)\tilde{A}(\mu) = \tilde{B}\tilde{B}^T, \quad (6.8)$$

with $\tilde{A}(\mu) = U^T A(\mu)U$, $\tilde{E} = U^T E U$ and $\tilde{B} = U^T B$ for the unknown $Y(\mu) \in \mathbb{R}^{N \times N}$. As indicated in Section 2.2.2, (6.8) can be efficiently solved in $O(N^3)$ operations using the Bartels-Stewart algorithm [BS72]. The approximate solution of (6.5) is then obtained as

$$\tilde{X}(\mu) = UY(\mu)U^T. \quad (6.9)$$

The compressed matrices $\tilde{A}(\mu)$ and \tilde{E} inherit the negative/positive definiteness from $A(\mu)$ and E , respectively. Hence, both $Y(\mu)$ and $\tilde{X}(\mu)$ are guaranteed to stay positive semidefinite, in contrast to the approximate solutions obtained using the plain RB method.

To setup the Lyapunov equation (6.8) efficiently, we use the affine linear decomposition of $A(\mu)$ to obtain:

$$\tilde{A}(\mu) = U^T A(\mu)U = \theta_1(\mu)U^T A_1 U + \dots + \theta_Q(\mu)U^T A_Q U.$$

Again, the precomputation of $U^T A_q U$ for $q = 1, \dots, Q$, as well as $U^T E U$ and $U^T B$, allows us to reduce the complexity to $O(QN^2)$ for constructing the coefficient matrices and $O(N^3)$ for solving (6.8).

6.2.3 Low-rank structure in the error estimator

The computation of the error estimates proceeds as described in Section 2.3.5, with the approximate solution $\tilde{x}(\mu) = VY(\mu)$ replaced by $\tilde{x}(\mu) = (U \otimes U)\text{vec}(Y(\mu))$ which can be reshaped as $\tilde{X}(\mu)$ in (6.9). For a fixed parameter sample $\mu \in D$, the Frobenius norm of the residual

$$R(\mu) = BB^T + A(\mu)\tilde{X}(\mu)E + E\tilde{X}(\mu)A(\mu)$$

can be computed as

$$\begin{aligned} \|R(\mu)\|_F^2 &= (b - \mathcal{A}(\mu)\tilde{x}(\mu))^T (b - \mathcal{A}(\mu)\tilde{x}(\mu)) \\ &= b^T b - 2b^T \mathcal{A}(\mu)(U \otimes U)\text{vec}(Y(\mu)) \\ &\quad + \text{vec}(Y(\mu))^T (U \otimes U)^T \mathcal{A}(\mu)\mathcal{A}(\mu)(U \otimes U)\text{vec}(Y(\mu)). \end{aligned}$$

Again, by taking the affine linear decomposition of $A(\mu)$ into account we obtain

$$\begin{aligned}
 \|R(\mu)\|_F^2 &= \text{tr}((B^T B)(B^T B)) \\
 &+ 4 \sum_{q=1}^Q \theta_q(\mu) \text{tr}((B^T E U) Y(\mu) (U^T A_q B)) \\
 &+ 2 \sum_{q_1, q_2=1}^Q \theta_{q_1}(\mu) \theta_{q_2}(\mu) \text{tr}(Y(\mu) (U^T A_{q_2} A_{q_1} U) Y(\mu) (U^T E E U)) \\
 &+ 2 \sum_{q_1, q_2=1}^Q \theta_{q_1}(\mu) \theta_{q_2}(\mu) \text{tr}(Y(\mu) (U^T A_{q_1} E U) Y(\mu) (U^T A_{q_2} E U)),
 \end{aligned} \tag{6.10}$$

where $\text{tr}(\cdot)$ denotes the trace of the corresponding matrix.

Having precomputed $B^T E U$, $U^T A_q B$, $U^T A_q E U$, $U^T E E U$ and $U^T A_{q_2} A_{q_1} U$ for $q, q_1, q_2 = 1, \dots, Q$ then allows us to attain a complexity of $O(Q^2 N^3)$ for computing $\|R(\mu)\|_F$.

The difference to the true solution $X(\mu)$ of the Lyapunov equation (6.5) in the Frobenius norm can be estimated as

$$\|X(\mu) - \tilde{X}(\mu)\|_F \leq \frac{\|R(\mu)\|_F}{\lambda_{\min}(\mathcal{A}(\mu))} \leq \frac{\|R(\mu)\|_F}{\lambda_{\text{LB}}(\mu)} =: \Delta(\mu), \quad \mu \in D, \tag{6.11}$$

where $\lambda_{\text{LB}}(\mu)$ is a positive lower bound for $\lambda_{\min}(\mathcal{A}(\mu))$. Here, using the Frobenius norm is suitable because it allows efficient computation of such lower bounds $\lambda_{\text{LB}}(\mu)$. The eigenvalue properties of the Kronecker product and the minimax characterization of eigenvalues yield the following lower bound on $\lambda_{\min}(\mathcal{A}(\mu))$:

$$\begin{aligned}
 \lambda_{\min}(\mathcal{A}(\mu)) &= \min_{v \in \mathbb{R}^{n^2}} \frac{v^T (-A(\mu) \otimes E - E \otimes A(\mu)) v}{v^T v} \\
 &\geq \min_{v \in \mathbb{R}^{n^2}} \frac{v^T (-A(\mu) \otimes E) v}{v^T v} + \min_{v \in \mathbb{R}^{n^2}} \frac{v^T (-E \otimes A(\mu)) v}{v^T v} \\
 &= 2\lambda_{\min}(E)\lambda_{\min}(-A(\mu)) \geq 2\lambda_{\min}(E)\lambda_{\text{LB}}(-A(\mu)) =: \lambda_{\text{LB}}(\mu),
 \end{aligned} \tag{6.12}$$

where $\lambda_{\text{LB}}(-A(\mu))$ denotes a positive lower bound for the smallest eigenvalue of the symmetric positive definite $n \times n$ matrix $-A(\mu)$, which can be efficiently computed either using SCM (Algorithm 2) or the subspace-accelerated SCM (Algorithm 3).

6.3 Algorithm

In this section, we present a summary, in form of Algorithm 10, of our low-rank reduced basis approach for Lyapunov equations introduced in Section 6.2 and discuss its implementation and computational complexity.

Chapter 6. Low-rank approach to parameter dependent symmetric Lyapunov equations

Algorithm 10 Offline phase of the Low-rank Reduced Basis method for Lyapunov equations

Input: Training set $\Xi \subset D$, $E \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$ affine linear decomposition such that $A(\mu) = \theta_1(\mu)A_1 + \dots + \theta_Q(\mu)A_Q \in \mathbb{R}^{n \times n}$ is symmetric negative definite for every $\mu \in \Xi$. Error tolerance $\varepsilon_{\text{LRRB}}$.

Output: Reduced model for computation $\lambda_{LB}(\mu)$ produced by Algorithm 3 and a low-rank basis U for Cholesky factors such that $\Delta(\mu) < \varepsilon_{\text{LRRB}}$ for every $\mu \in \Xi$.

- 1: Compute $\lambda_{\min}(E)$.
 - 2: Use Algorithm 3 to compute $\lambda_{\min}(-A(\mu))$ for all $\mu \in \Xi$, as well as a reduced model for evaluating $\lambda_{\min}(-A(\mu))$ on D .
 - 3: Compute $\lambda_{LB}(\mu)$ according to (6.12) for all $\mu \in \Xi$.
 - 4: Set $M = 0$ and μ_{\max} to be a randomly chosen element of Ξ .
 - 5: **while** $\Delta_M(\mu_{\max}) > \varepsilon_{\text{LRRB}}$ **do**
 - 6: $\mu_{M+1} \leftarrow \mu_{\max}$
 - 7: Compute $L(\mu_{M+1})$ by solving (6.6) for $\mu = \mu_{M+1}$.
 - 8: Set $\mathcal{U}_{M+1} = \mathcal{U}_M + \text{range}(L(\mu_{M+1}))$ and update the truncated orthonormal basis U .
 - 9: Set $M \leftarrow M + 1$.
 - 10: Update the projected matrices $U^T A_q U$, $U^T E U$, $U^T B$, $B^T E U$, $U^T A_q B$, $U^T A_q E U$, $U^T E E U$ and $U^T A_{q_2} A_{q_1} U$ for $q, q_1, q_2 = 1, \dots, Q$.
 - 11: **for** $\mu \in \Xi$ **do**
 - 12: **if** $C_{\text{sat}} \Delta_M(\mu) < \Delta_{\max}$ **then**
 - 13: Exit the for loop.
 - 14: **end if**
 - 15: Recompute $Y(\mu)$ according to (6.8).
 - 16: Recompute $\|R(\mu)\|_F$ according to (6.10).
 - 17: Recompute $\Delta_M(\mu)$ according to (6.11).
 - 18: Update Δ_{\max} and μ_{\max} .
 - 19: **end for**
 - 20: **end while**
-

6.3.1 Computational details

In the following we discuss some details of an efficient implementation of Algorithm 10.

Computation of $\lambda_{\text{LB}}(\mu)$. Computation of a lower bound for $\lambda_{\min}(\mathcal{A}(\mu))$ requires estimating $\lambda_{\min}(-A(\mu))$ on D . As previously mentioned, we can use Algorithm 2 or Algorithm 3 to compute non-negative lower bounds on the whole parameter domain. In our implementation, we use Algorithm 3 with the same training set Ξ , and the number of sampled eigenpairs ℓ set to 1. In contrast to Section 3.5, we set the prescribed tolerance ε_{SCM} to 0.1, since computing more accurate estimates to $\lambda_{\min}(-A(\mu))$ will not significantly affect the values of the error estimates (6.11). The rest of the settings is as described in Section 3.5.

Updating U . Suppose that we are given the truncated SVD of $L_M = U\Sigma V^T$ from the previous iteration, where $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_N)$. Given the new low-rank Cholesky factor $L(\mu_{M+1}) \in \mathbb{R}^{n \times m_{M+1}}$, L_{M+1} can be written in the following form

$$L_{M+1} = [U\Sigma V^T, L(\mu_{M+1})] = \begin{bmatrix} U & L(\mu_{M+1}) \end{bmatrix} \begin{bmatrix} \Sigma & \\ & I_{m_{M+1}} \end{bmatrix} \begin{bmatrix} V^T & \\ & I_{m_{M+1}} \end{bmatrix}.$$

The SVD of L_{M+1} can now be efficiently computed by first computing the QR decomposition of $[U, L_{M+1}] = Q_U R_U$, followed by computing the SVD decomposition of the matrix $R_U \begin{bmatrix} \Sigma \\ I_{m_{M+1}} \end{bmatrix}$, and computing the updated matrices U and V in the end. The QR decomposition and the updated matrices U and V can be computed in $O(nN^2)$, while computing SVD requires $O(N^3)$ operations, resulting in the total computational cost of $O(nN^2 + N^3)$. Finally, the obtained singular value decomposition is compressed by truncating the singular values σ_i such that $\sigma_i/\sigma_1 < \text{tol}_{\text{compr}}$.

Updating the projected matrices. Before computing the projected matrices in line 10 of Algorithm 10, it is preferable to first form the products $A_1 U, \dots, A_Q U, EU$, which can be done in $O(n^2 N)$ operations. This allows computation of the projected matrices in $O(Q^2 N^2 n)$ operations, resulting in the total computational cost of $O(Qn^2 N + Q^2 N^2 n)$.

Computation of the next parameter sample μ_{M+1} . The next parameter sample μ_{M+1} is computed as the maximizer of the error estimate (6.11) on Ξ . In every iteration, this requires recomputing the approximate solution $Y(\mu)$ and the residual norm $\|R(\mu)\|_F$ on the whole training set Ξ , which can become computationally quite expensive. As described in Remark 2.17, the search for μ_{M+1} (lines 11–19 in Algorithm 10) can be optimized using the saturation assumption (2.38), which combined with the error estimates from the previous iteration often allows us to skip recomputations for a number of points in Ξ . Moreover, as in Remark 2.17, we denote with Δ_{max} and μ_{max} the current maximum error estimate and the point in Ξ where it was attained, respectively. In the implementation, we have worked with $C_{\text{sat}} = +\infty$ in the first five iterations of Algorithm 10, and with $C_{\text{sat}} = 1$ in the following iterations.

Computation of $L(\mu)$. For computing the low-rank Cholesky factor $L(\mu)$, it is preferable to use a method which computes the low-rank factor of the solution directly and is also able to exploit sparsity in $A(\mu)$ and E . One of such approaches is Algorithm 9 in Chapter 5 which seems to be very competitive for solving Lyapunov equations with low-rank right-hand side (see Example 5.13). As previously discussed in Section 2.2.2, the alternatives include the approaches based on the ADI method [Pen00] or the extended Arnoldi method [Sim07], which often compute an accurate approximation in only a few iterations, where each iteration consists of solving a linear system with shifted matrices $A(\mu) - \tau E$. In our implementation, we have used the implementation by Stykel of the Generalized low-rank ADI method, which was already discussed in Remark 2.14.

6.3.2 Computational complexity

In addition to the computational cost of Algorithm 3 which is discussed in Section 3.4.3, the computational cost of performing M iterations of Algorithm 10 is

$$O(Mk_{\text{LYAP}}m_{\text{CLS}} + MQNn^2 + M(N^3 + Q^2N^2n) + M\|\Xi\|(N^3 + Q^2N^3)),$$

where N is the number of columns in U , and $\|\Xi\|$ is the number of elements in Ξ .

The first term $O(Mk_{\text{LYAP}}m_{\text{CLS}})$ corresponds to the computational cost of solving M Lyapunov equations for the sampled parameters values, where k_{LYAP} is the number of iterations in the Lyapunov solver, and c_{LS} is the computational cost of solving one linear system with a shifted matrix $A(\mu) - \tau E$. For example, if $A(\mu)$ and E are sparse with bandwidth ρ , c_{LS} can be estimated at a cost of $O(n\rho^2)$, which corresponds to the computational cost of computing sparse LU factorization for $A(\mu) - \tau E$. The second term $O(MQNn^2)$ corresponds to the computational cost of computing products A_iU and EU in each iteration, while the third term $O(M^3 + MQ^2N^2n)$ corresponds to the computational cost of computing U and the projected matrices in line 10 of Algorithm 10. The fourth term $O(M\|\Xi\|N^3)$ corresponds to the computational cost of computing the approximate solutions $Y(\mu)$, while the fifth term $O(M\|\Xi\|Q^2N^3)$ corresponds to the computational cost of computing the error estimates (the residual norms $\|R(\mu)\|_F$) throughout the iterations.

It is important to note that the contributions to the obtained computational cost of n and $\|\Xi\|$ are independent, so both can be chosen large if necessary. Moreover in practice, as explained in Section 6.3.1, by assuming the saturation assumption, we can expect reduced contributions of the fourth and the fifth term to the overall complexity, since, for a fixed $\mu \in \Xi$, the approximate solution $Y(\mu)$ and $\|R(\mu)\|_F$ do not have to be recomputed in every iteration, but rather only a few times throughout the execution.

In the online phase, the computational cost for calculating the approximate solution of the Lyapunov equation and the accompanying error estimator is $O(QN^2 + N^3)$ and $O(Q^2N^3)$, respectively, and thus independent of n .

Defining the lower bound $\lambda_{\text{LB}}(\mu)$ as in (6.12) is very important, since using Algorithm 3 to estimate the smallest eigenvalues of $A(\mu)$ is significantly cheaper than to estimate the smallest eigenvalues of $\mathcal{A}(\mu)$, as the most computationally demanding part of Algorithm 3 is usually the computation of exact eigenpairs, which depends on the matrix size.

Remark 6.1. *The attained level of accuracy in Algorithm 10 clearly depends on the truncation tolerance $\text{tol}_{\text{compr}}$ used in the truncated SVD. Therefore, $\text{tol}_{\text{compr}}$ needs to be chosen in accordance with $\varepsilon_{\text{LRRB}}$. However, choosing a very small value for $\text{tol}_{\text{compr}}$ may result in large values of N and, thus, lead to high computational costs.*

6.4 Numerical examples

In this section, we present several numerical experiments to illustrate the properties of the low-rank RB method, given in Algorithm 10 and described in Section 6.2. Algorithms 3 and 10 have been implemented and tested in the same computing environment already described in Section 3.5.

We present convergence of the maximum error estimate (6.11) and evolution of the ranks of U , both with respect to the number of iterations and with respect to the execution time (in seconds). Additionally, we simulate the online phase of RB method, where we compute the error estimates and the exact error for the computed approximate solutions of the Lyapunov equation (6.5) at 100 randomly selected parameter values that do not belong to Ξ .

When implementing and testing Algorithm 10, we have made the following choices. We set the error tolerance to $\varepsilon_{\text{LRRB}} = 10^{-4}$, the maximum number of iterations to $M_{\text{max}} = 25$ and the surrogate set Ξ to be a random subset of D containing 1000 elements. The exact Cholesky factors $L(\mu_i)$ have been computed with the tolerance set to 10^{-10} . For running Algorithm 3, we have used the same training set Ξ .

Example 6.2. *We consider the parameter-dependent Lyapunov equation associated with an example from [KT11, Section 4]. The matrices $E, A(\mu) \in \mathbb{R}^{n \times n}$ arise as the stiffness and the mass matrix, respectively, from the finite element discretization of a stationary heat equation on a square domain Ω containing d disjoint discs, as shown in Figure 6.1. The heat conductivity coefficient in each of these discs is governed by a parameter $\mu^{(q)}$; thus $A(\mu)$ depends on d parameters and can be written as*

$$A(\mu) = A_1 + \sum_{q=1}^d \mu^{(q)} A_{q+1}, \quad \forall \mu \in D,$$

with A_1, \dots, A_{d+1} the discretizations of the Laplace operator on each of the subdomains. Each of the parameters $\mu^{(j)}$ is assumed to be in the interval $[0.1, 10]$, resulting in $D = [0.1, 10]^d$. The training set Ξ is chosen as a random subset of D consisting of 1000 parameter samples. We have set the truncation tolerance for the truncated SVD to $\text{tol}_{\text{compr}} = 10^{-8}$.

Figure 6.2 shows the performance of Algorithm 10 for the case $d = 1$. For this example, only

one iteration of Algorithm 3 is sufficient to provide reliable lower bounds for the smallest eigenvalue of $A(\mu)$. The ranks of the exact Cholesky factors $L(\mu)$ are approximately equal to 20 for all sampled μ_i , giving $m_i = 20$, while after $M = 4$ iterations, the dimension of the resulting subspace (after truncation) \mathcal{U}_M is $N = 53$. The error estimates and the exact error for the computed approximate solutions of the Lyapunov equation (6.5) at 100 randomly selected parameter values that do not belong to Ξ can be seen in Figure 6.2c.

We have also considered a more challenging version of the problem with $d = 4$, where $n = 1580$, and presented the results in Figure 6.3. For this example, Algorithm 3 converges in 10 iterations, the rank of Cholesky factors is approximately 25, which, after the maximum $M = 25$ iterations, results in U of dimension 198. Even though the desired tolerance of 10^{-4} in Algorithm 10 has not been reached, the true errors evaluated for 100 randomly selected parameter samples in D are all below the prescribed tolerance, as can be seen in Figure 6.3c. In order to attain the prescribed tolerance in the error estimates, it is necessary either to increase the allowed number of iterations, or to reduce the truncation tolerance $\text{tol}_{\text{compr}} = 10^{-8}$. However, based on the fact that the ranks of U are starting to stagnate after around 20 iterations, increasing the number of iterations would probably not help, which indicates the need for a smaller value of $\text{tol}_{\text{compr}}$ than 10^{-8} .

Motivated by the increasing computational cost of Algorithm 10 as $\text{rank}(U) = N$ gets larger, in the following, we additionally consider a slightly modified version of Algorithm 10. Instead of constructing a single basis U for \mathcal{U}_M with $\text{tol}_{\text{compr}} = 10^{-8}$, we construct two bases, U and \tilde{U} , with $\text{tol}_{\text{compr}} = 10^{-4}$ and $\text{tol}_{\text{compr}} = 10^{-8}$, respectively. The basis U is used as before, for the error estimation in the offline phase (Algorithm 10), while \tilde{U} is used later for computing approximate solutions in the online phase. This approach can be considered as heuristic, since there is no guarantee that $\tilde{U} \otimes \tilde{U}$ is a notably better solution space than $U \otimes U$. In practice, as we can see from the results presented in Figure 6.4, this modification leads to a decrease in ranks of U , resulting in a significant reduction in the execution time, while the accuracy in the online phase appears to be unaffected. Basis \tilde{U} has approximately equal rank, and provides roughly the same accuracy level as does U which was constructed with $\text{tol}_{\text{compr}} = 10^{-8}$ and shown in Figure 6.3c. Further investigation of this or a similar idea may be interesting for future research.

6.5 Conclusion

We have developed a low-rank version of the reduced basis method for solving parameter-dependent symmetric Lyapunov equations.

We have modified the reduced basis method in a way that the low-rank Cholesky factors are sampled instead of sampling the vectorized solutions of size $n^2 \times 1$. Consequently, the solution subspace \mathcal{U}_M is constructed as the range of the concatenated sampled low-rank Cholesky factors. This modification leads to computation of an approximate solution $\tilde{X}(\mu)$ inside the tensorized product $\mathcal{U}_M \otimes \mathcal{U}_M$, which not only improves accuracy, but also guarantees that $\tilde{X}(\mu)$ is positive semidefinite for all $\mu \in D$. Moreover, we have shown that, by computing a

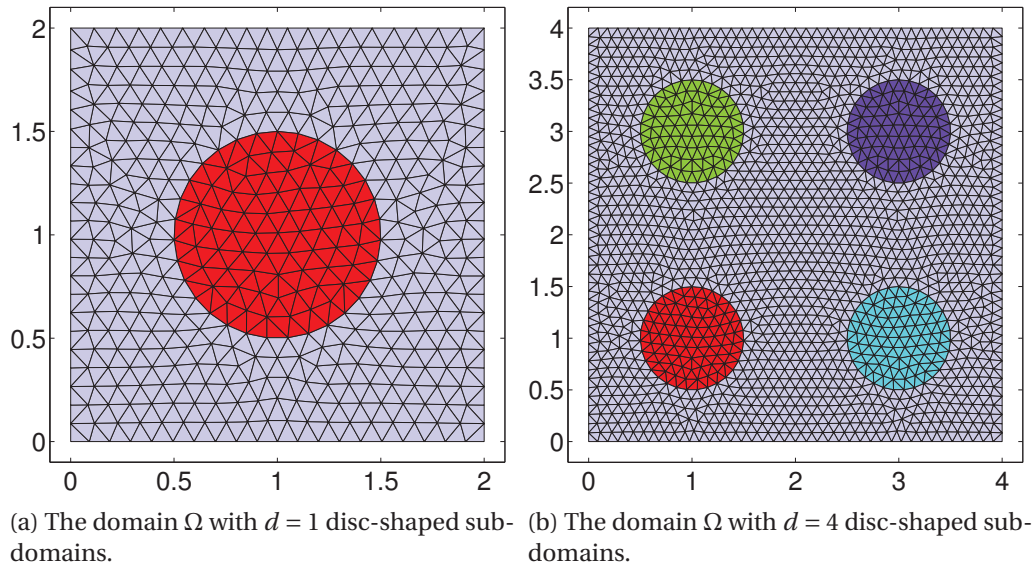
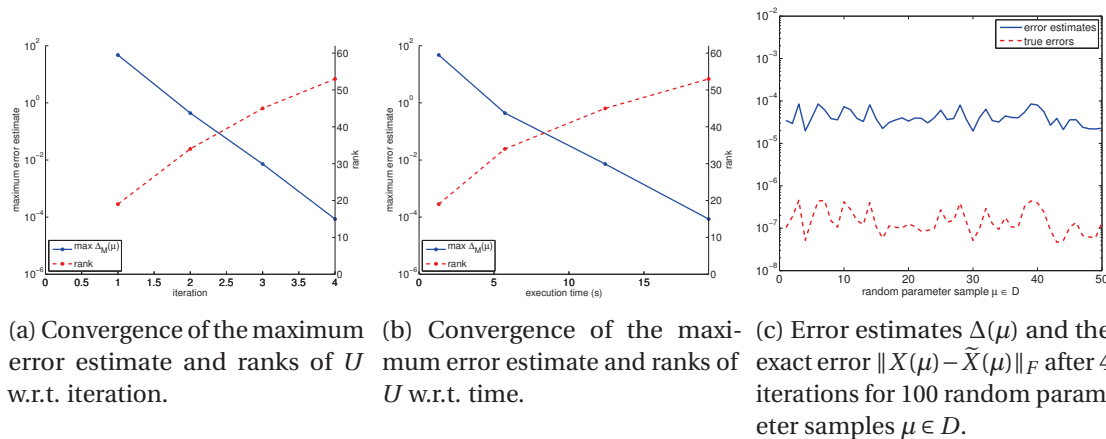


Figure 6.1: Finite element mesh used in Example 6.2.

Figure 6.2: The performance of Algorithm 10 for Example 6.2 with $d = 1$.

posteriori error estimates in the Frobenius norm, estimation of the smallest eigenvalue of a $n^2 \times n^2$ matrix $\mathcal{A}(\mu)$ can be avoided. Instead, it is sufficient to compute a positive lower bound for the smallest eigenvalues of $A(\mu) \in \mathbb{R}^{n \times n}$. Furthermore, by incorporating low-rank truncation, we have mitigated the growth in $\dim(\mathcal{U}_M)$.

The resulting solver, Algorithm 10, is demonstrated to perform well for a problem that has been discussed earlier in the literature. However, for more challenging problems which require even larger values of $\dim(\mathcal{U}_M)$, repeated computation of approximate solutions (2.35) might become a bottleneck. For such cases, we propose a heuristic approach where the sampling is driven by error estimates based on a coarser approximation to \mathcal{U}_M , whereas a finer one is used to compute approximate solutions in the online phase.

Chapter 6. Low-rank approach to parameter dependent symmetric Lyapunov equations

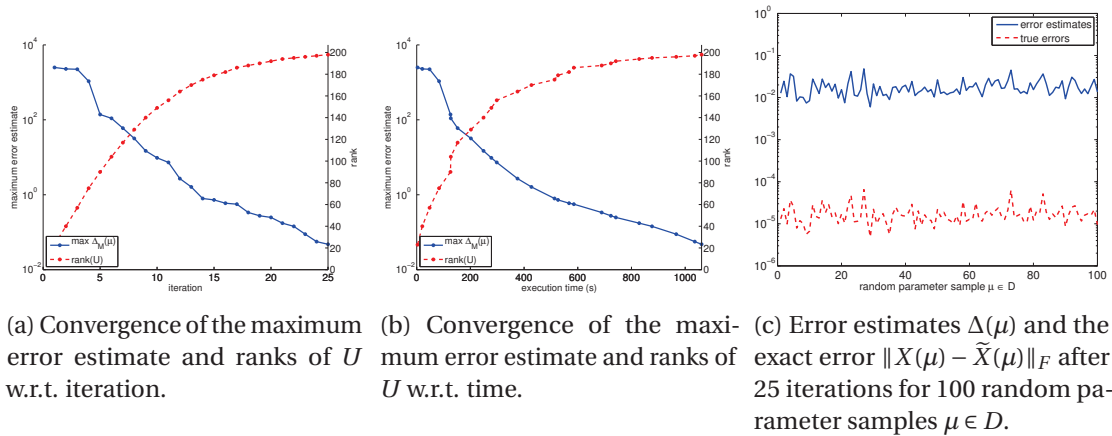


Figure 6.3: The performance of Algorithm 10 for Example 6.2 with $d = 4$.

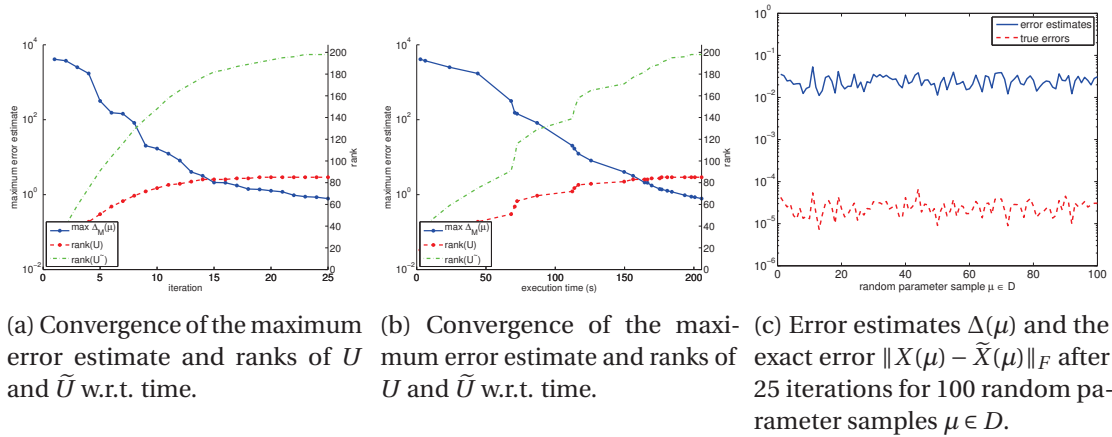


Figure 6.4: The performance of a modified version of Algorithm 10 with two orthonormal basis, U and \tilde{U} , for \mathcal{U}_M , as described in Example 6.2 with $d = 4$.

7 Conclusion

In this thesis, we have discussed the development of efficient algorithms for two numerical linear algebra problems arising in model reduction, estimation of the smallest eigenvalues of a parameter-dependent matrix, and solving linear matrix equations. The following contributions have been made.

In Chapter 3, we have presented new theoretical results about the existing state-of-the-art approach for estimating the smallest eigenvalues of a parameter-dependent Hermitian matrix $A(\mu)$, the Successive Constraint Method (SCM), which indicate its limitations and explain the slow convergence often observed in practice. We have demonstrated that these issues can be overcome by additionally incorporating subspace acceleration techniques, which implicitly exploit regularity in the smallest eigenvectors, resulting in a per iteration computational cost which is only slightly larger than in SCM. We have shown that our proposed subspace-accelerated approach (Algorithm 3) has better interpolation properties than SCM, with both subspace bounds interpolating the partial derivatives of $\lambda_{\min}(A(\mu))$. Furthermore, if $A(\mu)$ is analytic and $d = 1$, we present results which indicate that we can expect exponential convergence of Algorithm 3 on intervals where $\lambda_{\min}(A(\mu))$ stays simple. These improved theoretical properties of Algorithm 3 are reflected in the numerical experiments, where Algorithm 3 presents a significant improvement over SCM, both in terms of iterations and the total computational time, on a number of numerical examples from the literature. We have successfully applied our approach to the estimation of coercivity and inf-sup constants. We have observed that for problems with small gaps between the smallest eigenvalues and a large variation in the parameter space the convergence of the subspace lower bounds may still not be satisfactory and for such cases we propose a heuristic approach using residual-based lower bounds.

In Chapter 4, we have shown that the singular value problem $\sigma_{\min}(zI - A)$ can be recasted into a Hermitian eigenvalue problem linearly depending on two real parameters. As parameter-dependent Hermitian eigenvalue problems can be successfully addressed using Algorithm 3, this motivates the development of a novel projection-based approach (Algorithm 4) for computation of ε pseudospectra, which builds upon Algorithm 3 by additionally taking into account the specific problem structure and demands for high absolute accuracy. This requires several modifications in order to make Algorithm 4 computationally efficient and competitive to

other existing approaches, such as making the residual computation more numerically stable, accelerating the computation of the lower bounds, and introducing a "warm start" strategy. The new approach is particularly suited for computation of ε -pseudospectra in isolated parts of the spectrum, containing only few eigenvalues of A . Additionally, we have shown that the interpolation results from Chapter 3 naturally extend to the proposed singular value bounds, allowing us to provide *a priori* error estimates. We have compared the performance of Algorithm 4 to other existing approaches on a number of examples discussed in the literature. For larger values of n , the approximation of pseudospectra using Algorithm 4 is significantly faster than by using the straightforward grid-based evaluation, while providing satisfactory accuracy. In comparison to other projection-based approaches, Algorithm 4 provides higher relative accuracy w.r.t. to the subspace size, especially in the proposed upper bounds $\sigma_{\text{SUB}}(x, y; \mathcal{S}, \ell)$, as well as the rigorous error estimates. Moreover, to our knowledge, it is the first approach which provides certified upper bounds for ε -pseudospectra, enabling localization of eigenvalues.

In Chapter 5, we have developed a novel low-rank approach (Algorithm 9) based on greedy low-rank updates for addressing large-scale linear matrix equations whose solution can be well approximated with a low-rank matrix. Furthermore, the greedy low-rank updates in Algorithm 9 are combined with Galerkin projection and preconditioning strategies in order to ensure satisfactory convergence speed. The performance of our approach is further optimized by incorporating low-rank truncation and a restarting strategy. We have demonstrated that our solver, Algorithm 9, performs quite well for problems that have been discussed earlier in the literature, especially those for which the imposed limit on the subspace size is not reached. For more challenging problems that feature larger ranks, the need for constructing and solving the Galerkin systems may become a bottleneck.

In Chapter 6, we have developed a low-rank version of the reduced basis method for solving parameter-dependent symmetric Lyapunov equations (Algorithm 10). Instead of sampling the vectorized solutions of size $n^2 \times 1$ as in the standard reduced basis method, in Algorithm 10 the low-rank Cholesky factors are sampled, and consequently, the solution subspace \mathcal{U}_M is constructed as the range of the concatenated sampled low-rank Cholesky factors. The approximate solutions $\tilde{X}(\mu)$ are computed inside the tensorized product $\mathcal{U}_M \otimes \mathcal{U}_M$, which not only improves accuracy, but also guarantees that $\tilde{X}(\mu)$ is symmetric positive semidefinite for all $\mu \in D$. Moreover, we have shown that, by computing *a posteriori* error estimates in the Frobenius norm, the estimation of the smallest eigenvalue of the $n^2 \times n^2$ operator matrix can be avoided. Instead, it is sufficient to compute a positive lower bound for the smallest eigenvalues of $A(\mu) \in \mathbb{R}^{n \times n}$, which can be done efficiently using Algorithm 3. Furthermore, by incorporating low-rank truncation, we have mitigated the growth in $\dim(\mathcal{U}_M)$. The resulting solver, Algorithm 10, is demonstrated to perform well for a problem that has been discussed earlier in the literature.

Bibliography

- [AF08] D. Amsallem and C. Farhat, *Interpolation method for adapting reduced-order models and application to aeroelasticity*, AIAA journal **46** (2008), no. 7, 1803–1813.
- [AF11] ———, *An online method for interpolating linear parametric reduced-order models*, SIAM J. Sci. Comput. **33** (2011), no. 5, 2169–2198. MR 2837528
- [AMCK06] A. Ammar, B. Mokdad, F. Chinesta, and R. Keunings, *A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modeling of complex fluids*, Journal of Non-Newtonian Fluid Mechanics **139** (2006), no. 3, 153–176.
- [ApS15] MOSEK ApS, *The mosek optimization toolbox for matlab manual. version 7.1 (revision 28)*., 2015.
- [AS12] R. Andreev and C. Schwab, *Sparse tensor approximation of parametric eigenvalue problems*, Numerical analysis of multiscale problems, Lect. Notes Comput. Sci. Eng., vol. 83, Springer, Heidelberg, 2012, pp. 203–241. MR 3050915
- [ASZ02] A. C. Antoulas, D. C. Sorensen, and Y. Zhou, *On the decay rate of Hankel singular values and related issues*, Systems Control Lett. **46** (2002), no. 5, 323–342. MR 2011320 (2004g:93044)
- [Bau85] H. Baumgärtel, *Analytic perturbation theory for matrices and operators*, Operator Theory: Advances and Applications, vol. 15, Birkhäuser Verlag, Basel, 1985. MR 878974
- [BB12] P. Benner and T. Breiten, *Interpolation-based \mathcal{H}_2 -model reduction of bilinear control systems*, SIAM J. Matrix Anal. Appl. **33** (2012), no. 3, 859–885. MR 3023455
- [BB13] ———, *Low rank methods for a class of generalized Lyapunov equations and related issues*, Numer. Math. **124** (2013), no. 3, 441–470. MR 3066036
- [BBBG11] U. Baur, C. Beattie, P. Benner, and S. Gugercin, *Interpolatory projection methods for parameterized model reduction*, SIAM J. Sci. Comput. **33** (2011), no. 5, 2489–2518. MR 2861634

Bibliography

- [BD11] P. Benner and T. Damm, *Lyapunov equations, energy functionals, and model order reduction of bilinear and stochastic systems*, SIAM J. Control Optim. **49** (2011), no. 2, 686–711. MR 2801215 (2012f:93025)
- [BDD⁺00] Z. Bai, J. W. Demmel, J. J. Dongarra, A. Ruhe, and H. van der Vorst (eds.), *Templates for the solution of algebraic eigenvalue problems*, Software, Environments, and Tools, vol. 11, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2000, A practical guide. MR 1792141 (2001k:65003)
- [BES05] C. A. Beattie, M. Embree, and D. C. Sorensen, *Convergence of polynomial restart Krylov methods for eigenvalue computations*, SIAM Rev. **47** (2005), no. 3, 492–515.
- [BF14] P. Benner and L. Feng, *A robust algorithm for parametric model order reduction based on implicit moment matching*, Reduced order methods for modeling and computational reduction, MS&A. Model. Simul. Appl., vol. 9, Springer, Cham, 2014, pp. 159–185. MR 3241211
- [BG01] C. Bekas and E. Gallopoulos, *Cobra: parallel path following for computing the matrix pseudospectrum*, Parallel Comput. **27** (2001), no. 14, 1879–1896. MR 1862612
- [BG05] A. Böttcher and S. M. Grudsky, *Spectral properties of banded Toeplitz matrices*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2005. MR 2179973 (2006k:47054)
- [BGBMN91] A. Bunse-Gerstner, R. Byers, V. Mehrmann, and N. K. Nichols, *Numerical computation of an analytic singular value decomposition of a matrix valued function*, Numer. Math. **60** (1991), no. 1, 1–39. MR 1131497 (92k:65072)
- [BGW15] P. Benner, S. Gugercin, and K. Willcox, *A survey of projection-based model reduction methods for parametric dynamical systems*, SIAM Rev. **57** (2015), no. 4, 483–531. MR 3419868
- [BH96] T. Braconnier and N. J. Higham, *Computing the field of values and pseudospectra using the Lanczos method with continuation*, BIT **36** (1996), no. 3, 422–440, International Linear Algebra Year (Toulouse, 1995). MR 1410089 (97d:65015)
- [BLO03] J. V. Burke, A. S. Lewis, and M. L. Overton, *Robust stability and a criss-cross algorithm for pseudospectra*, IMA J. Numer. Anal. **23** (2003), no. 3, 359–375.
- [BLT09] P. Benner, R.-C. Li, and N. Truhar, *On the ADI method for Sylvester equations*, J. Comput. Appl. Math. **233** (2009), no. 4, 1035–1045. MR 2557293 (2010k:15031)
- [BMNP04] M. Barrault, Y. Maday, N. C. Nguyen, and A. T. Patera, *An ‘empirical interpolation’ method: application to efficient reduced-basis discretization of partial differential equations*, C. R. Math. Acad. Sci. Paris **339** (2004), no. 9, 667–672. MR 2103208

- [BR03] M. Bebendorf and S. Rjasanow, *Adaptive low-rank approximation of collocation matrices*, Computing **70** (2003), no. 1, 1–24.
- [Brü96] M. Brühl, *A curve tracing algorithm for computing the pseudospectrum*, BIT **36** (1996), no. 3, 441–454, International Linear Algebra Year (Toulouse, 1995). MR 1410090 (97d:65016)
- [BS72] R. H. Bartels and G. W. Stewart, *Algorithm 432: The solution of the matrix equation $AX + XB = C$* , Communications of the ACM **15** (1972), no. 9, 820–826.
- [BS06] Z. Bai and D. Skoogh, *A projection method for model reduction of bilinear dynamical systems*, Linear Algebra Appl. **415** (2006), no. 2-3, 406–425. MR 2227782 (2006m:93024)
- [BS13] Peter Benner and Jens Saak, *Numerical solution of large and sparse continuous time algebraic matrix Riccati and Lyapunov equations: a state of the art survey*, GAMM-Mitt. **36** (2013), no. 1, 32–52. MR 3095913
- [BT05] T. Betcke and L. N. Trefethen, *Reviving the method of particular solutions*, SIAM Rev. **47** (2005), no. 3, 469–491. MR MR2178637 (2006k:65344)
- [CAC10] F. Chinesta, A. Ammar, and E. Cueto, *Recent advances and new challenges in the use of the proper generalized decomposition for solving multidimensional models*, Arch. Comput. Methods Eng. **17** (2010), no. 4, 327–350. MR 2739942 (2011j:65239)
- [CCDS13] A. Chkifa, A. Cohen, R. DeVore, and C. Schwab, *Sparse adaptive Taylor approximation algorithms for parametric and stochastic elliptic PDEs*, ESAIM: M2AN Math. Mod. and Num. Anal. **47** (2013), no. 01, 253–280.
- [CEL11] E. Cancès, V. Ehrlacher, and T. Lelièvre, *Convergence of a greedy algorithm for high-dimensional convex nonlinear problems*, Mathematical Models and Methods in Applied Sciences **21** (2011), no. 12, 2433–2467.
- [CHMR09] Y. Chen, J. S. Hesthaven, Y. Maday, and J. Rodríguez, *Improved successive constraint method based a posteriori error estimate for reduced basis approximation of 2D Maxwell's problem*, M2AN Math. Model. Numer. Anal. **43** (2009), no. 6, 1099–1116. MR 2588434
- [Dam08] T. Damm, *Direct methods and ADI-preconditioned Krylov subspace methods for generalized Lyapunov equations*, Numer. Linear Algebra Appl. **15** (2008), no. 9, 853–871. MR 2464173 (2009j:65083)
- [Dav06] T. A. Davis, *Direct methods for sparse linear systems*, Fundamentals of Algorithms, vol. 2, SIAM, Philadelphia, PA, 2006. MR 2270673 (2007m:65002)
- [Dav07] E. B. Davies, *Linear operators and their spectra*, Cambridge Studies in Advanced Mathematics, vol. 106, Cambridge University Press, Cambridge, 2007. MR 2359869 (2008k:47001)

Bibliography

- [DK70] Chandler Davis and W. M. Kahan, *The rotation of eigenvectors by a perturbation. III*, SIAM J. Numer. Anal. **7** (1970), 1–46. MR 0264450
- [DP04] E. B. Davies and M. Plum, *Spectral pollution*, IMA J. Numer. Anal. **24** (2004), no. 3, 417–438. MR 2068830 (2005c:47027b)
- [DS14] S. V. Dolgov and D. V. Savostyanov, *Alternating minimal energy methods for linear systems in higher dimensions*, SIAM J. Sci. Comput. **36** (2014), no. 5, A2248–A2271. MR 3262607
- [DVM12] J. De Vlieger and K. Meerbergen, *A subspace method for unimodal symmetric eigenvalue optimization problems involving large scale matrices*, TW Reports (2012).
- [EB10] A. K. Eppler and M. Bollhöfer, *An alternative way of solving large Lyapunov equations*, Proceedings in Applied Mathematics and Mechanics **10** (2010), no. 1, 547–548.
- [EHS09] C. Engström, C. Hafner, and K. Schmidt, *Computation of lossy Bloch waves in two-dimensional photonic crystals*, J. Comput. Theor. Nanosci. **6** (2009), 1–9.
- [EPR10] J. L. Eftang, A. T. Patera, and E. M. Rønquist, *An “hp” certified reduced basis method for parametrized elliptic partial differential equations*, SIAM J. Sci. Comput. **32** (2010), no. 6, 3170–3200. MR 2746617 (2012b:65162)
- [ERS07] H. C. Elman, A. Ramage, and D. J. Silvester, *Algorithm 886: IFISS, a Matlab toolbox for modelling incompressible flow*, ACM Trans. Math. Software **33** (2007), no. 2, Art. 14, 18. MR 2326956
- [FG13] G. M. Flagg and S. Gugercin, *On the ADI method for the Sylvester equation and the optimal- \mathcal{H}_2 points*, Appl. Numer. Math. **64** (2013), 50–58. MR 2999597
- [FMPV15] I. Fumagalli, A. Manzoni, N. Parolini, and M. Verani, *Reduced basis approximation and a posteriori error estimates for parametrized elliptic eigenvalue problems*, Tech. report, MATHICSE, EPFL, 2015.
- [GBS08] A. Ghosh, S. Boyd, and A. Saberi, *Minimizing effective resistance of a graph*, SIAM Rev. **50** (2008), no. 1, 37–66. MR 2403057 (2009j:05071)
- [GHK03] L. Grasedyck, W. Hackbusch, and B. N. Khoromskij, *Solution of large scale algebraic matrix Riccati equations by use of hierarchical matrices*, Computing **70** (2003), no. 2, 121–165. MR 1982969 (2004f:65056)
- [GJK04] E. Gutkin, E. A. Jonckheere, and M. Karow, *Convexity of the joint numerical range: topological and differential geometric viewpoints*, Linear Algebra Appl. **376** (2004), 143–171. MR 2014890 (2004i:15021)
- [GK14] Luka Grubišić and Daniel Kressner, *On the eigenvalue decay of solutions to operator Lyapunov equations*, Systems Control Lett. **73** (2014), 42–47. MR 3270953

- [GLAM92] 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. Software **18** (1992), no. 2, 223–231. MR 1167892
- [GMM⁺15] F. Gong, O. Meyerson, J. Meza, M. Stoiciu, and A. Ward, *Explicit bounds for the pseudospectra of various classes of matrices and operators*, arXiv preprint arXiv:1505.05931 (2015).
- [GO11] N. Guglielmi and M. L. Overton, *Fast algorithms for the approximation of the pseudospectral abscissa and pseudospectral radius of a matrix*, SIAM J. Matrix Anal. Appl. **32** (2011), no. 4, 1166–1192. MR 2854608
- [GS98] S. K. Godunov and M. Sadkane, *Computation of pseudospectra via spectral projectors*, Linear Algebra Appl. **279** (1998), no. 1-3, 163–175. MR 1637865 (99i:65046)
- [Han08] A. C. Hansen, *On the approximation of spectra of linear operators on Hilbert spaces*, J. Funct. Anal. **254** (2008), no. 8, 2092–2126. MR 2402104 (2009c:47004)
- [Han15] A. Hannukainen, *Convergence analysis of GMRES for the Helmholtz equation via pseudospectrum*, arXiv preprint arXiv:1505.08072 (2015).
- [Hau19] F. Hausdorff, *Der Wertvorrat einer Bilinearform*, Math. Z. **3** (1919), no. 1, 314–316. MR 1544350
- [HDO11] B. Haasdonk, M. Dihlmann, and M. Ohlberger, *A training set and multiple bases generation approach for parameterized model reduction based on adaptive grids in parameter space*, Math. Comput. Model. Dyn. Syst. **17** (2011), no. 4, 423–442. MR 2823471
- [Hig08] N. J. Higham, *Functions of matrices*, SIAM, Philadelphia, PA, 2008.
- [HKC⁺10] D. B. P. Huynh, D. J. Knezevic, Y. Chen, J. S. Hesthaven, and A. T. Patera, *A natural-norm successive constraint method for inf-sup lower bounds*, Comput. Methods Appl. Mech. Engrg. **199** (2010), no. 29-32, 1963–1975. MR 2654002 (2011e:65085)
- [HKL15] H. Hakula, V. Kaarnioja, and M. Laaksonen, *Approximate methods for stochastic eigenvalue problems*, Appl. Math. Comput. **267** (2015), 664–681. MR 3399080
- [HNPR10] D. B. P. Huynh, N. C. Nguyen, A. T. Patera, and G. Rozza, *rbMIT software [Software]*, 2010, MIT, Cambridge, US.
- [Hoc01] M. E. Hochstenbach, *A Jacobi-Davidson type SVD method*, SIAM J. Sci. Comput. **23** (2001), no. 2, 606–628, Copper Mountain Conference (2000).
- [HR92] D. Y. Hu and L. Reichel, *Krylov-subspace methods for the Sylvester equation*, Linear Algebra Appl. **172** (1992), 283–313.

Bibliography

- [HR00] C. Helmberg and F. Rendl, *A spectral bundle method for semidefinite programming*, SIAM J. Optim. **10** (2000), no. 3, 673–696. MR 1741192 (2002b:90095)
- [HRS16] J. S. Hesthaven, G. Rozza, and B. Stamm, *Certified reduced basis methods for parametrized partial differential equations*, SpringerBriefs in Mathematics, Springer, Cham; BCAM Basque Center for Applied Mathematics, Bilbao, 2016, BCAM SpringerBriefs. MR 3408061
- [HRSP07] D. B. P. Huynh, G. Rozza, S. Sen, and A. T. Patera, *A successive constraint linear optimization method for lower bounds of parametric coercivity and inf-sup stability constants*, C. R. Math. Acad. Sci. Paris **345** (2007), no. 8, 473–478. MR 2367928 (2008j:93061)
- [HS95] M. Hochbruck and G. Starke, *Preconditioned Krylov subspace methods for Lyapunov matrix equations*, SIAM J. Matrix Anal. Appl. **16** (1995), no. 1, 156–171. MR 1311424 (95k:65045)
- [HSBTZ13] C. Hartmann, B. Schäfer-Bung, and A. Thöns-Zueva, *Balanced averaging of bilinear systems with applications to stochastic control*, SIAM J. Control Optim. **51** (2013), no. 3, 2356–2378.
- [HSZ14] J. S. Hesthaven, B. Stamm, and S. Zhang, *Efficient greedy algorithms for high-dimensional parameter spaces with applications to empirical interpolation and reduced basis methods*, ESAIM Math. Model. Numer. Anal. **48** (2014), no. 1, 259–283. MR 3177844
- [Joh78] C. R. Johnson, *Numerical determination of the field of values of a general complex matrix*, SIAM J. Numer. Anal. **15** (1978), no. 3, 595–602. MR 0474755
- [Joh89] ———, *A Gersgorin-type lower bound for the smallest singular value*, Linear Algebra Appl. **112** (1989), 1–7. MR 976325 (90a:15022)
- [Kat95] T. Kato, *Perturbation theory for linear operators*, Classics in Mathematics, Springer-Verlag, Berlin, 1995, Reprint of the 1980 edition. MR 1335452 (96a:47025)
- [KM06] Y. Kim and M. Mesbahi, *On maximizing the second smallest eigenvalue of a state-dependent graph Laplacian*, IEEE Trans. Autom. Control **51** (2006), no. 1, 116–120. MR 2192798 (2006g:05128)
- [KMMM15] F. Kangal, K. Meerbergen, E. Mengi, and W. Michiels, *A subspace method for large scale eigenvalue optimization*, arXiv preprint arXiv:1508.04214 (2015).
- [Kny01] A. V. Knyazev, *Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method*, SIAM J. Sci. Comput. **23** (2001), no. 2, 517–541. MR MR1861263 (2003g:65050)

- [KPT14] D. Kressner, M. Plešinger, and C. Tobler, *A preconditioned low-rank CG method for parameter-dependent Lyapunov matrix equations*, Numer. Linear Algebra Appl. **21** (2014), no. 5, 666–684. MR 3266228
- [Kre14] D. Kressner, *Lectures notes in computational linear algebra*, 2014.
- [KS15] D. Kressner and P. Sirković, *Truncated low-rank methods for solving general linear matrix equations*, Numer. Linear Algebra Appl. **22** (2015), no. 3, 564–583. MR 3338934
- [KSSS14] D. Kressner, P. Sirkovic, N. T. Son, and T. Stykel, *A low-rank Reduced Basis Method for Parameter-Dependent Lyapunov Equations*, 21st International Symposium on Mathematical Theory of Networks and Systems, 2014, pp. 42–45.
- [KT10] D. Kressner and C. Tobler, *Krylov subspace methods for linear systems with tensor product structure*, SIAM J. Matrix Anal. Appl. **31** (2010), no. 4, 1688–1714.
- [KT11] ———, *Low-rank tensor Krylov subspace methods for parametrized linear systems*, SIAM J. Matrix Anal. Appl. **32** (2011), no. 4, 1288–1316. MR 2854614
- [KV14] D. Kressner and B. Vandereycken, *Subspace methods for computing the pseudospectral abscissa and the stability radius*, SIAM J. Matrix Anal. Appl. **35** (2014), no. 1, 292–313. MR 3180856
- [Lan64] P. Lancaster, *On eigenvalues of matrices dependent on a parameter*, Numer. Math. **6** (1964), 377–387. MR 0171375
- [Lan78] H. J. Landau, *The notion of approximate eigenvalues applied to an integral equation of laser theory*, Quart. Appl. Math. **35** (1977/78), no. 1, 165–172. MR 0446101 (56 #4433)
- [LL05] C.-K. Li and R.-C. Li, *A note on eigenvalues of perturbed Hermitian matrices*, Linear Algebra Appl. **395** (2005), 183–190. MR 2112884 (2005i:15017)
- [LO96] A. S. Lewis and M. L. Overton, *Eigenvalue optimization*, Acta numerica, 1996, Acta Numer., vol. 5, Cambridge Univ. Press, Cambridge, 1996, pp. 149–190. MR 1624599 (99e:90072)
- [LSY98] R. B. Lehoucq, D. C. Sorensen, and C. Yang, *ARPACK users' guide*, SIAM, Philadelphia, PA, 1998, Solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods.
- [Lui97] S. H. Lui, *Computation of pseudospectra by continuation*, SIAM J. Sci. Comput. **18** (1997), no. 2, 565–573. MR 1433795 (98j:65028)
- [Mer12] A. Merz, *Computation of generalized Gramians for model reduction of bilinear control systems and time-delay systems*, Ph.D. thesis, TU Kaiserslautern, 2012.

Bibliography

- [MG07] J. Matousek and B. Gärtner, *Understanding and using linear programming*, Springer Science & Business Media, 2007.
- [MG14] H. Meidani and R. Ghanem, *Spectral power iterations for the random eigenvalue problem*, *AIAA Journal* **52** (2014), no. 5, 912–925.
- [MH03] J. C. Mason and D. C. Handscomb, *Chebyshev polynomials*, Chapman & Hall/CRC, Boca Raton, FL, 2003. MR 1937591 (2004h:33001)
- [MMa07] *Matrix Market*, 2007, Available at <http://math.nist.gov/MatrixMarket/>.
- [MMMVB15] K. Meerbergen, E. Mengi, W. Michiels, and R. Van Beeumen, *Computation of pseudospectral abscissa for large scale nonlinear eigenvalue problems*, Tech. report, 2015.
- [MMO⁺00] L. Machiels, Y. Maday, I. B. Oliveira, A. T. Patera, and D. V. Rovas, *Output bounds for reduced-basis approximations of symmetric positive definite eigenvalue problems*, *C. R. Math. Acad. Sci. Paris* **331** (2000), no. 2, 153–158. MR 1781533 (2001d:65148)
- [MN15] A. Manzoni and F. Negri, *Heuristic strategies for the approximation of stability factors in quadratically nonlinear parametrized PDEs*, *Adv. Comput. Math.* **41** (2015), no. 5, 1255–1288. MR 3428566
- [Moo81] B. C. Moore, *Principal component analysis in linear systems: controllability, observability, and model reduction*, *IEEE Trans. Automat. Control* **26** (1981), no. 1, 17–32.
- [MP02] D. Mezher and B. Philippe, *PAT—a reliable path-following algorithm*, *Numer. Algorithms* **29** (2002), no. 1-3, 131–152, Matrix iterative analysis and biorthogonality (Luminy, 2000). MR 1896950 (2003b:65037)
- [MYK14] E. Mengi, E. A. Yildirim, and M. Kiliç, *Numerical optimization of eigenvalues of Hermitian matrix functions*, *SIAM J. Matrix Anal. Appl.* **35** (2014), no. 2, 699–724. MR 3215062
- [NMA15] F. Negri, A. Manzoni, and D. Amsallem, *Efficient model reduction of parametrized systems by matrix discrete empirical interpolation*, *J. Comput. Phys.* **303** (2015), 431–454. MR 3422722
- [Nou08] A. Nouy, *Generalized spectral decomposition method for solving stochastic finite element equations: invariant subspace problem and dedicated algorithms*, *Comput. Methods Appl. Mech. Engrg.* **197** (2008), no. 51-52, 4718–4736. MR 2464512 (2009m:60151)
- [Nou10] ———, *Proper generalized decompositions and separated representations for the numerical solution of high dimensional stochastic problems*, *Archives of Computational Methods in Engineering* **17** (2010), 403–434.

- [NVP05] N. C. Nguyen, K. Veroy, and A. T. Patera, *Certified real-time solution of parametrized partial differential equations*, Handbook of Materials Modeling, Springer, 2005, pp. 1529–1564.
- [OR00] J. M. Ortega and W. C. Rheinboldt, *Iterative solution of nonlinear equations in several variables*, Classics in Applied Mathematics, vol. 30, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2000, Reprint of the 1970 original. MR 1744713 (2000j:65005)
- [Par98] B. N. Parlett, *The symmetric eigenvalue problem*, Classics in Applied Mathematics, vol. 20, SIAM, Philadelphia, PA, 1998, Corrected reprint of the 1980 original.
- [Pen00] T. Penzl, *Eigenvalue decay bounds for solutions of Lyapunov equations: the symmetric case*, Systems Control Lett. **40** (2000), no. 2, 139–144. MR 1829181 (2002b:93108)
- [PMEL10] H. Panzer, J. Mohring, R. Eid, and B. Lohmann, *Parametric model order reduction by matrix interpolation*, at Automatisierungstechnik **58** (2010), no. 8, 475–484.
- [PR07] A. T. Patera and G. Rozza, *Reduced basis approximation and a posteriori error estimation for parametrized partial differential equations*, MIT-Pappalardo Graduate Monographs in Mechanical Engineering, MIT, Cambridge, MA, US, 2007.
- [PRV⁺02] C. Prud’homme, D. V. Rovas, K. Veroy, L. Machiels, Y. Maday, A. T. Patera, and G. Turinici, *A mathematical and computational framework for reliable real-time solution of parametrized partial differential equations*, M2AN Math. Model. Numer. Anal. **36** (2002), no. 5, 747–771. MR 1955536
- [PS82] L. Pernebo and L. M. Silverman, *Model reduction via balanced state space representations*, IEEE Trans. Automat. Control **27** (1982), no. 2, 382–387. MR 680103
- [QMN16] A. Quarteroni, A. Manzoni, and F. Negri, *Reduced basis methods for partial differential equations*, Unitext, vol. 92, Springer, Cham, 2016, An introduction, La Matematica per il 3+2. MR 3379913
- [Rel69] Franz Rellich, *Perturbation theory of eigenvalue problems*, Assisted by J. Berkowitz. With a preface by Jacob T. Schwartz, Gordon and Breach Science Publishers, New York-London-Paris, 1969. MR 0240668
- [RHP08] G. Rozza, D.B.P. Huynh, and A. T. Patera, *Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations: application to transport and continuum mechanics*, Arch. Comput. Methods Eng. **15** (2008), no. 3, 229–275. MR 2430350 (2009j:65336)

Bibliography

- [RS78] M. Reed and B. Simon, *Methods of modern mathematical physics. IV. Analysis of operators*, Academic Press [Harcourt Brace Jovanovich Publishers], New York, 1978. MR 0493421 (58 #12429c)
- [RSH93] S. C. Reddy, P. J. Schmid, and D. S. Henningson, *Pseudospectra of the Orr-Sommerfeld operator*, SIAM J. Appl. Math. **53** (1993), no. 1, 15–47. MR 1202838 (94g:76021)
- [RSS01] M. Rojas, S. A. Santos, and D. C. Sorensen, *A new matrix-free algorithm for the large-scale trust-region subproblem*, SIAM J. Optim. **11** (2000/01), no. 3, 611–646. MR 1814035 (2002j:90055)
- [Saa92] Y. Saad, *Numerical methods for large eigenvalue problems: Theory and algorithms*, John Wiley, New York, 1992.
- [Sab06] J. Sabino, *Solution of large-scale Lyapunov equations via the block modified smith methods*, Ph.D. thesis, Department of Computational and Applied Mathematics, Rice University, Houston, TX, 2006.
- [SB08] V. Sima and P. Benner, *Experimental evaluation of new slicot solvers for linear matrix equations based on the matrix sign function*, Computer-Aided Control Systems, 2008. CACSD 2008. IEEE International Conference on, IEEE, 2008, pp. 601–606.
- [Sch07] P. J. Schmid, *Nonmodal stability theory*, Annual review of fluid mechanics. Vol. 39, Annu. Rev. Fluid Mech., vol. 39, Annual Reviews, Palo Alto, CA, 2007, pp. 129–162. MR 2309484 (2008h:76001)
- [SG98] V. Simoncini and E. Gallopoulos, *Transfer functions and resolvent norm approximation of large matrices*, Electron. Trans. Numer. Anal. **7** (1998), 190–201, Large scale eigenvalue problems (Argonne, IL, 1997). MR 1667647 (99i:65039)
- [Sim07] V. Simoncini, *A new iterative method for solving large-scale Lyapunov matrix equations*, SIAM J. Sci. Comput. **29** (2007), no. 3, 1268–1288.
- [Sim13] V. Simoncini, *Computational methods for linear matrix equations*, 2013, Preprint available from <http://www.dm.unibo.it/~simoncin/list.html>.
- [Sio68] Minoru Siotani, *Cancellation of Section 5 of: "Some applications of Loewner's ordering on symmetric matrices"*, Ann. Inst. Statist. Math. **20** (1968), 168. MR 0226786 (37 #2373)
- [Sir16] P. Sirković, *A reduced basis approach to large-scale pseudospectra computation*, Tech. report, 2016, Available from http://sma.epfl.ch/~anchpcommon/publications/RBPSC_PS.pdf.
- [SK16] P. Sirković and D. Kressner, *Subspace acceleration to parameter-dependent hermitian eigenproblems*, To appear in SIAM J Matrix Anal. Appl. (2016).

- [Sta91] Gerhard Starke, *Optimal alternating direction implicit parameters for nonsymmetric systems of linear equations*, SIAM J. Numer. Anal. **28** (1991), no. 5, 1431–1445.
- [Sty08] T. Stykel, *Low-rank iterative methods for projected generalized Lyapunov equations*, Electron. Trans. Numer. Anal. **30** (2008), 187–202. MR 2480076
- [SVH⁺06] S. Sen, K. Veroy, D. B. P. Huynh, S. Deparis, N. C. Nguyen, and A. T. Patera, “*Natural norm*” *a posteriori* error estimators for reduced basis approximations, J. Comput. Phys. **217** (2006), no. 1, 37–62. MR 2250524 (2007d:65104)
- [TE05] L. N. Trefethen and M. Embree, *Spectra and pseudospectra*, Princeton University Press, Princeton, NJ, 2005. MR 2155029 (2006d:15001)
- [Toe18] O. Toeplitz, *Das algebraische Analogon zu einem Satze von Fejér*, Math. Z. **2** (1918), no. 1-2, 187–197. MR 1544315
- [Tre99] L. N. Trefethen, *Computation of pseudospectra*, Acta numerica, 1999, Acta Numer., vol. 8, Cambridge Univ. Press, Cambridge, 1999, pp. 247–295. MR 1819647 (2002b:65062)
- [Tre08] C. Tretter, *Spectral theory of block operator matrices and applications*, Imperial College Press, London, 2008. MR 2463978 (2010e:47033)
- [TT96] K.-C. Toh and L. N. Trefethen, *Calculation of pseudospectra by the Arnoldi iteration*, SIAM J. Sci. Comput. **17** (1996), no. 1, 1–15, Special issue on iterative methods in numerical linear algebra (Breckenridge, CO, 1994). MR 1375262 (97i:65064)
- [Uhl14] F. Uhlig, *Faster and more accurate computation of the field of values boundary for n by n matrices*, Linear Multilinear Algebra **62** (2014), no. 5, 554–567. MR 3195951
- [VRP02] K. Veroy, D. V. Rovas, and A. T. Patera, *A posteriori error estimation for reduced-basis approximation of parametrized elliptic coercive partial differential equations: “convex inverse” bound conditioners*, ESAIM Control Optim. Calc. Var. **8** (2002), 1007–1028. MR 1932984 (2003m:35052)
- [Wac63] E. L. Wachspress, *Extended application of alternating direction implicit iteration model problem theory*, J. Soc. Indust. Appl. Math. **11** (1963), 994–1016. MR 0169373
- [Wac88] ———, *Iterative solution of the Lyapunov matrix equation*, Appl. Math. Lett. **1** (1988), no. 1, 87–90.
- [Wri02] T. G. Wright, *EigTool*, 2002, Available at <http://www.comlab.ox.ac.uk/pseudospectra/eigtool/>.

Bibliography

- [WT01] T. G. Wright and L. N. Trefethen, *Large-scale computation of pseudospectra using ARPACK and eigs*, SIAM J. Sci. Comput. **23** (2001), no. 2, 591–605, Copper Mountain Conference (2000). MR 1861267 (2002h:65061)

Curriculum Vitae

PETAR SIRKOVIĆ

Born: September 14th, 1988 in Zagreb, Croatia

Nationality: Croatian

EDUCATION

- Sep 2012 - ongoing PhD studies in Applied mathematics with advisor Prof. D. Kressner
Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland
Topic: *Low-rank approaches for parameteric eigenvalue problems
and matrix equations*
- Sep 2010 - Jul 2012 M.Sc. in Applied Mathematics at
University of Zagreb, Department of Mathematics, Zagreb, Croatia
Master thesis with Prof. A. Dujella, *LLL algorithm and
applications to cryptography*
- Oct 2007 - Jul 2010 B.Sc. in Mathematics at
University of Zagreb, Department of Mathematics, Zagreb, Croatia
- Sep 2003 - Jul 2007 V. gymnasium, Zagreb, Croatia

ADDITIONAL EDUCATION

- Aug–Oct 2011 JSC Guest Student Programme on Scientific Computing
Jülich Supercomputing Center, FZ Jülich
Scientific report: *Brain volume reconstruction-parallel
implementation of unimodal registration*

Curriculum Vitae

AWARDS

Apr 2012	Dean's award at the Faculty of Natural Sciences and Mathematics.
Jun 2011	Rector's award at the University of Zagreb.
Jun 2010	Best student award at the Department of Mathematics.
Aug 2009	2 nd prize at IMC 2009, Budapest.
Jul 2007	Bronze medal at IMO 2007, Vietnam.
2007-2012	Scholarship for gifted students, City of Zagreb, Croatia.
2011, 2013	Participation in the final online round of Google Code Jam.
October 2010	Winner of TopCoder Marathon Match 66.
May 2007	Winner of TCHS Team Competition 2007.
2003–2007	Winner of top prizes in high school state competitions in mathematics, physics, informatics and logics.

WORK EXPERIENCE

since 2012	Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland Research assistant
Aug–Oct 2011	Julich Supercomputing Centre, Julich, Germany Research intern
Jul 2009	Ekobit, Zagreb, Croatia Software developing internship
2007 – 2010	organization of mathematics and informatics camps for high school students in Croatia
2007 – 2011	V. gymnasium, Zagreb Mentor for students participating at state high school competitions

PUBLICATIONS

- 2016 P. Sirković: "Reduced basis approach to pseudospectra computation",
Technical report.
- 2015 P. Sirković and D. Kressner: "Subspace acceleration for large-scale
parameter-dependent Hermitian eigenproblems"
To appear in SIAM J. Matrix Anal. Appl.
- 2014 D. Kressner and P. Sirković: "Truncated low-rank methods for solving
general linear matrix equations"
Numerical Linear Algebra with Applications, May 2015
- 2011 P. Sirković and S. Stanko: "Revealing block-diagonal structure
of stochastic matrices and identification of metastable states of
Markov chains", submitted for rector prize (in Croatian)
- 2011 P. Sirković : "Parallel brain volume reconstruction –
registration of unimodal images",
Proceedings of JSC Guest Student Programme 2011

CONFERENCE PROCEEDINGS

- 2013 D. Kressner, **P. Sirković**, N. T. Son and T. Stykel:
"A low-rank reduced basis method for
parameter-dependent Lyapunov equations"
Extended abstract in Proceedings of MTNS 2014

Curriculum Vitae

CONFERENCE CONTRIBUTIONS

- Sep 21–22, 2015 Workshop on matrix equations and tensor techniques, Bologna, Italy
Talk: *A reduced basis approach for pseudospectra computation*
- Aug 24–28, 2015 New directions in numerical computation, Oxford, United Kingdom
Talk: *A Reduced Basis approach to large-scale pseudospectra computation*
- Aug 17–19, 2015 Pro*Doc Retreat, Disentis, Switzerland
Talk: *Solving parametrized eigenvalue problems with application to pseudospectra computation*
- Jul 13–15, 2015 Mathicse retreat 2015, Disentis, Switzerland
Talk: *Solving parametrized eigenvalue problems with application to pseudospectra computation*
- Jul 9–10, 2015 GAMM Workshop on Applied and Numerical Linear Algebra, Magdeburg, Germany
Talk: *A Reduced Basis approach to large-scale pseudospectra computation*
- Apr 17, 2015 Swiss Numerics Day, University of Geneva, Geneva, Switzerland
Poster: *A Reduced Basis approach to large-scale pseudospectra computation*
- Jul 7, 2014 The 21th International Symposium on Mathematical Theory of Networks and Systems, University of Groningen, Netherlands
Talk: *A low-rank reduced basis method for parameter-dependent Lyapunov equations*
- Jun 12–14, 2014 Mathicse retreat 2014, Disentis, Switzerland
Talk: *Parametrized eigenvalue estimation with applications*
- Oct 10–11, 2013 Workshop on Matrix Equations and Tensor Techniques
Poster: *Greedy low-rank approaches to general linear matrix equations*
- Sep 23–28, 2013 DAAD International School on Linear Optimal Control of Dynamic Systems, Osijek, Croatia
- Aug 14–16, 2013 Pro*Doc Retreat, Disentis, Switzerland
Talk: *Greedy low-rank approaches to general linear matrix equations*
- Jun 10–14, 2013 ApplMath13, Šibenik, Croatia
Talk: *Efficient eigenvalue estimation for parametrized problems in the reduced basis method*
- Apr 5, 2013 Swiss Numerics Colloquium, EPF Lausanne, Lausanne, Switzerland
Poster: *Greedy low-rank approaches to general linear matrix equations*

