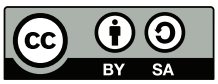
Projection Methods for Generalized Eigenvalue Problems

Revised Edition

Christoph Conrads https://christoph-conrads.name

July 2, 2016



This document is the revised edition of the Master's thesis *Projection Methods for Generalized Eigenvalue Problems* by Christoph Conrads; the cover page is different, the layout was slightly modified, the abstract in German as well as the declaration of originality were removed, and typos were fixed.

Abstract

This thesis treats the numerical solution of generalized eigenvalue problems (GEPs) $Kx = \lambda Mx$, where K, M are Hermitian positive semidefinite (HPSD). We discuss problem and solution properties, accuracy assessment of solutions, aspect of computations in finite precision, the connection to the finite element method (FEM), dense solvers, and projection methods for these GEPs. All results are directly applicable to real-world problems.

We present properties and origins of GEPs with HPSD matrices and briefly mention the FEM as a source of such problems.

With respect to accuracy assessment of solutions, we address quickly computable and structure-preserving backward error bounds and their corresponding condition numbers for GEPs with HPSD matrices. There is an abundance of literature on backward error measures possessing one of these features; the backward error in this thesis provides both.

In Chapter 3, we elaborate on dense solvers for GEPs with HPSD matrices. The standard solver reduces the GEP to a standard eigenvalue problem; it is fast but requires positive definite mass matrices and is only conditionally backward stable. The QZ algorithm for general GEPs is backward stable but it is also much slower and does not preserve any problem properties. We present two new backward stable and structure preserving solvers, one using deflation of infinite eigenvalues, the other one using the generalized singular value decomposition (GSVD). We analyze backward stability and computational complexity. In comparison to the QZ algorithm, both solvers are competitive with the standard solver in our tests. Finally, we propose a new solver combining the speed of deflation with the ability of GSVD-based solvers to handle singular matrix pencils.

Finally, we consider black-box solvers based on projection methods to compute the eigenpairs with the smallest eigenvalues of large, sparse GEPs with Hermitian positive definite matrices (HPD). After reviewing common methods for spectral approximation, we briefly mention ways to improve numerical stability. We discuss the automated multilevel substructuring method (AMLS) before analyzing the impact of off-diagonal blocks in block matrices on eigenvalues. We use the results of this thesis and insights in recent papers to propose a new divide-and-conquer eigensolver and to suggest a change that makes AMLS more robust. We test the divide-and-conquer eigensolver on sparse structural engineering matrices with 10,000 to 150,000 degrees of freedom.

2010 *Mathematics Subject Classification*. 65F15, 65F50, 65Y04, 65Y20.

Contents

1 Introduction

1.1 Problem Statement

This thesis treats the numerical solution of generalized eigenvalue problems $Kx = \lambda Mx$, where K, M are real symmetric positive semidefinite.

Definition 1.1. Let $K, M \in \mathbb{C}^{n,n}$. Finding $x \in \mathbb{C}^n \setminus \{0\}$ and $\lambda \in \mathbb{C}$ so that

$$Kx = \lambda Mx$$

is called a *generalized eigenvalue problem* (GEP). The pair (λ, x) is called *eigenpair*, λ is called (generalized) *eigenvalue*, and x is called (right) *eigenvector*.

In Chapter 2, we will discuss the numerical solution of GEPs in the context of scientific computing, we explain the connection to the Finite Element Method, and we will present (easily computable) error measures for the solution of GEPs. In Chapter 3, we will present numerically robust alternatives to the standard dense GEP solver and we will examine the practical performance of these solvers on a single core of a current CPU. Afterwards, we discuss the treatment of large, sparse GEPs.

1.2 Notation and Preliminaries

In this thesis, I_n denotes the $n \times n$ identity matrix and if the dimension is obvious, we omit the index. The variables $e_1, e_2, \ldots, e_n$ denote the ith column of I_n and their dimension is always apparent from the context. Let $A \in \mathbb{C}^{m,n}$, then $\operatorname{ran} A$ denotes the range of A and $\ker A$ denotes the kernel (null space) of A. We call an $m \times n$ matrix A isometric if it has orthonormal columns. Equivalently, $m \geq n$ and $A^*A = I_n$, where A^* is the complex conjugate transpose of A. When using the notation $A = [a_1, a_2, \ldots, a_n]$, the a_i are the columns of A. When we wish to use the value of the matrix A in row i and column j, then we can write A_{ij} or $A = [a_{ij}]$ so that a_{ij} is the (i, j) entry in A. The notation ΔA signifies a perturbation of A and we denote the complex conjugate transpose of ΔA by ΔA^*.

Let $A \in \mathbb{C}^{n,n}$, let $S \in \mathbb{C}^{n,s}$ be isometric. Sometimes we wish to solve GEPs and SEPs restricted to a given subspace. Let $S \in \mathbb{C}^{n,s}$ contain an orthonormal basis of this subspace (S is isometric), let $x_S \in \mathbb{C}^s$ be a solution in the subspace. Computing $x = Sx_S \in F^n$ is called *lifting* x_S.

Previously, we introduced the generalized eigenvalue problem and for completeness, we also introduce the standard eigenvalue problem.

Definition 1.2. Standard Eigenvalue Problem [HJ12, §1.1] Let $A \in \mathbb{C}^{n,n}$. Finding $x \in \mathbb{C}^n \setminus \{0\}$ and $\lambda \in \mathbb{C}$ so that

$$Ax = \lambda x$$

is called a (standard) *eigenvalue problem* (SEP). The pair (λ, x) is called *eigenpair*, λ is called *eigenvalue*, and x is called (right) *eigenvector*.

Not every $n \times n$ matrix possesses n linearly independent eigenvectors.

Theorem 1.1 ([HJ12, Theorem 1.3.7]). *Let $A \in \mathbb{C}^{n,n}$ have n linearly independent eigenvectors. Then there exist an invertible matrix $X \in \mathbb{C}^{n,n}$ and a diagonal matrix $\Lambda \in \mathbb{C}^{n,n}$ such that*

$$A = X\Lambda X^{-1}.$$

The columns $x_1, x_2, \ldots, x_n$ of X are the eigenvectors of A and the diagonal entries $\lambda_1, \lambda_2, \ldots, \lambda_n$ of Λ are the eigenvalues belonging to the eigenvectors.

Normal matrices are a well known subset of all diagonalizable matrices.

Definition 1.3 (Normal matrix [HJ12, Definition 2.5.1]). Let $A \in \mathbb{C}^{n,n}$. If

$$A^*A = AA^*,$$

then A is called *normal*.

Theorem 1.2 ([HJ12, Theorem 2.5.3]). *Let $A \in \mathbb{C}^{n,n}$ be normal. Then there exists a unitary matrix $X \in \mathbb{C}^{n,n}$ and a diagonal matrix $\Lambda \in \mathbb{C}^{n,n}$ such that*

$$A = X\Lambda X^*.$$

In this thesis, we often deal with Hermitian matrices.

Theorem 1.3 ([HJ12, Theorem 4.1.5]). *Let $A \in \mathbb{C}^{n,n}$ be Hermitian. Then there exists a unitary matrix $X \in \mathbb{C}^{n,n}$ and a diagonal matrix $\Lambda \in \mathbb{R}^{n,n}$ such that*

$$A = X\Lambda X^*.$$

If A is real, then all matrices can be taken to be real.

Observe that every Hermitian matrix is normal. An Hermitian matrix with positive eigenvalues is called *positive definite* (HPD) and an Hermitian matrix with non-negative eigenvalues is called *positive semidefinite* (HPSD). Accordingly, real symmetric matrices with these properties are called *symmetric positive definite* (SPD) and *symmetric positive semidefinite* (SPSD), respectively.

Another useful matrix decomposition is the singular value decomposition. For HPSD matrices, the SVD and the eigendecomposition are identical if the eigenvalues are sorted in ascending order.

Definition 1.4 (Singular Value Decomposition (SVD) [HJ12, §2.6]). Let $A \in \mathbb{C}^{m,n}$. There are unitary matrices $U \in \mathbb{C}^{m,m}$, $V \in \mathbb{C}^{n,n}$ and a diagonal matrix $\Sigma \in \mathbb{R}^{m,n}$ such that

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

This is called the *singular value decomposition* (SVD) of A. The columns $u_1, u_2, \ldots, u_m$ of U are called *left singular vectors*, the columns $v_1, v_2, \ldots, v_n$ of V are called *right singular vectors*, and the diagonal entries $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p \geq 0$ of Σ are called *singular values*, where $p = \min(m, n)$. If A is real, then all matrices can be taken to be real.

We can use the SVD to determine the rank of a matrix as well as bases for range and null space.

Theorem 1.4 ([MC, Corollary 2.4.6]). *Let $A \in \mathbb{C}^{m,n}$ have rank r. Then the following holds:*

- $\sigma_i > 0$, $i = 1, 2, \ldots, r$,
- $\sigma_i = 0$, $i = r+1, \ldots, p$, *where* $p = \min(m, n)$,
- $\operatorname{ran} A = \operatorname{span}\{u_1, u_2, \ldots, u_r\}$,
- $\ker A = \operatorname{span}\{v_{r+1}, \ldots, v_n\}$.

If $m \geq n$, then the first $\operatorname{rank} A$ columns of U form a basis for the range of A and we can use this property to define the so-called *thin SVD*.

Definition 1.5 (Thin SVD)**.** Let $m \geq n$, let $A \in \mathbb{C}^{m,n}$. Then there exists an isometric matrix $U \in \mathbb{C}^{m,n}$, a unitary matrix $V \in \mathbb{C}^{n,n}$, and a diagonal matrix $\Sigma \in \mathbb{R}^{n,n}$ such that

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

This is called the *thin SVD* of A. If A is real, then all matrices can be taken to be real.

We will denote vector and matrix norms with $\|\cdot\|$. In this thesis, we use induced matrix norms and the Frobenius norm.

Definition 1.6 (Induced matrix norm [HJ12, Definition 5.6.1])**.** Let $A \in \mathbb{C}^{m,n}$. Then we define the matrix p-norm as

$$\|A\|_p \coloneqq \max_{\|v\|_p = 1} \|Av\|_p.$$

Note that there are mainly three important norms in the definition above. Also, these matrix norms are called *subordinate matrix norms*, cf. [ASNA, §6.2] [MC, §2.3]. Let $A = [a_{ij}] \in \mathbb{C}^{m,n}$. Similar to the vector p-norms, the most frequently used induced matrix norms are

- the column sum norm $\|A\|_1 = \max_j \sum_{i=1}^m |a_{ij}|$
- the spectral norm $\|\cdot\|_2$, and
- the row sum norm $\|A\|_\infty = \max_i \sum_{j=1}^n |a_{ij}|$

The Frobenius norm is the Euclidean norm for vectors applied to a matrix:

$$\|A\|_F \coloneqq \sqrt{\sum_{i,j} |a_{ij}|^2}.$$

The following theorem highlights a desirable property of the spectral and the Frobenius norm.

Theorem 1.5 ([MC, §2.3.5])**.** *Let $A \in \mathbb{C}^{m,n}$, let $U \in \mathbb{C}^{m,m}$, $V \in \mathbb{C}^{n,n}$ be unitary matrices. Then it holds that*

$$\|U^*AV\|_p = \|A\|_p,\ p = 2, F,$$

i. e., the spectral and the Frobenius norm are unitarily invariant.

As a corollary, it holds that

$$\|A\|_2 = \sigma_1(A),$$
$$\|A\|_F = \sqrt{\Sigma_{i=1}^n \sigma_i^2}.$$

We also need further decompositions.

Definition 1.7 (Upper-trapezoidal matrix)**.** A matrix $= [a_{ij}] \in \mathbb{C}^{m,n}$ is called *upper-trapezoidal* if $a_{ij} = 0$ whenever $i > j$.

Definition 1.8 (QR factorization [HJ12, Theorem 2.1.14])**.** Let $A \in \mathbb{C}^{m,n}$. Then there exists a unitary matrix $U \in \mathbb{C}^{m,m}$ and an upper-trapezoidal matrix $R \in \mathbb{C}^{m,n}$ such that

$$A = QR.$$

This is called the *QR factorization* of A. If A is real, then all matrices can be taken to be real.

Definition 1.9 (Thin QR factorization)**.** Let $m \geq n$, let $A \in \mathbb{C}^{m,n}$. Then there exists an isometric matrix $U \in \mathbb{C}^{m,n}$ and an upper-triangular matrix $R \in \mathbb{C}^{n,n}$ such that

$$A = QR.$$

This is called the *thin QR factorization* of A. If A is real, all matrices can be taken to be real.

Similar to the QR factorization, we can define the QL, RQ, and LQ decompositions, where L is a lower triangular matrix. Additionally, we need the QR factorization with full column pivoting.

Definition 1.10 (QR factorization with full column pivoting [MC, §5.4.2])**.** Let $A \in \mathbb{C}^{m,n}$. Then there exist a unitary matrix $U \in \mathbb{C}^{m,m}$, an upper-trapezoidal matrix $R = [r_1, r_2, \ldots, r_n] \in \mathbb{C}^{m,n}$, and a permutation matrix $\Pi \in \mathbb{C}^{n,n}$ such that

$$A\Pi = QR,$$

$\|r_i\|_2 \geq \|r_j\|_2$, $i \leq j$, and $R_{ii} = 0$ whenever $i > \operatorname{rank} A$. This is called the *QR factorization with full column pivoting* of A. If A is real, then all matrices can be taken to be real.

Finally, we introduce the Cholesky decomposition with and without pivoting.

Definition 1.11 (Cholesky decomposition [HJ12, Corollary 7.2.9])**.** Let $A \in \mathbb{C}^{n,n}$ be HPD. Then there exists a unique lower triangular matrix $L \in \mathbb{C}^{n,n}$ such that

$$A = LL^*.$$

This is called the *Cholesky decomposition* of A. If A is real, then L can be taken to be real.

Definition 1.12 (Cholesky decomposition with complete pivoting [ASNA, §10.3])**.** Let $A \in \mathbb{C}^{n,n}$ be HPSD. Then there exist a lower triangular matrix $L \in \mathbb{C}^{n,n}$ and a permutation matrix $\Pi \in \mathbb{C}^{n,n}$ such that

$$A = \Pi L L^* \Pi^*.$$

Π is chosen so that the pivot element is the largest diagonal entry. This is called the *Cholesky decomposition with complete pivoting* of A. If A is real, then all matrices can be taken to be real.

In this thesis, we deal with GEPs with Hermitian matrices or *Hermitian GEPs* for short.

Definition 1.13 (Hermitian GEP)**.** A generalized eigenvalue problem $K - \lambda M$ is called *Hermitian* if K and M are Hermitian.

For generalized eigenvalue problems, hermiticity is not a property as powerful as for standard eigenvalue problem hence we will highlight some of these problems with a few examples found in [Par98, §15.2] and [Saa11, §9.2.1]. To that end, we need to introduce the concept of a GEP eigenvalue as a pair.

Definition 1.14. Let $K, M \in \mathbb{C}^{n,n}$. A pair $(\alpha, \beta) \neq (0, 0)$ of complex numbers is an eigenvalue of $K - \lambda M$ if

$$\det(\beta K - \alpha M) = 0.$$

We will emphasize the usefulness of pairs as eigenvalues with the following example where we cannot calculate $\lambda = {}^{\alpha}/_{\beta}$ for all eigenvalues.

Example 1.1. Let

$$K = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, M = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}.$$

This matrix pair has the eigenvalues $(1, 0)$ and $(0, 1)$.

For GEPs, hermiticity does not guarantee real eigenvalues.

Example 1.2 (Hermitian GEPs can have complex eigenvalues)**.** Let

$$K = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}, M = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

Then the matrix pencil $K - \lambda M$ has only complex eigenvalues (α, β) with $\beta = \pm \imath \alpha$, where $\imath$ is the imaginary unit.

Furthermore, there are circumstances where eigenvalues may be chosen arbitrarily.

Example 1.3 (Hermitian GEP with arbitrary eigenvalues I)**.** Let $K = 0, M = 0$. This Hermitian GEP has the eigenvalue (α, β), $(\alpha, \beta) \in \mathbb{C} \times \mathbb{C} \setminus \{0, 0\}$.

Unfortunately, non-trivial intersections of the null spaces are not a necessary criterion for the existence of GEPs with at least one arbitrary eigenvalue.

Example 1.4 (Hermitian GEP with arbitrary eigenvalues II)**.** Let

$$K = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, M = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

The intersection of the null spaces $\ker M = \operatorname{span} e_1$ and $\ker K = \operatorname{span} e_2$ is trivial yet $\det(\beta K - \alpha M) \equiv 0$ because the first row of K and second row of M are always linearly dependent. Moreover, the eigenvector x_0 corresponding to the arbitrary eigenvalue (α_0, β_0) is a function of the eigenvalue. Assuming α_0 and β_0 are nonzero, we have

$$x_0 = [{}^{\alpha_0}/_{\beta_0}, 1, 0]^T.$$

Evidently, the presence of arbitrary eigenvalues is a property of the matrix pencil and not of the individual matrices which makes their presence hard to detect. In fact, calculating the *distance to singularity* of a matrix pencil $K - \lambda M$ is an open research problem, cf. [MMW15] (Example 1.4 was constructed with the aid of Theorem 17). Pairs without arbitrary eigenvalues are called *regular*.

Definition 1.15 (Regular matrix pencil [Saa11, Definition 9.1])**.** A matrix pencil $K - \lambda M$ is called *regular* if

$$\det(\beta K - \alpha M) \not\equiv 0,$$

otherwise it is called *singular*.

For standard and regular generalized eigenvalue problems, eigenvalues are unique.

Definition 1.16 (Eigenvalue)**.** Let $r = \max_{\alpha,\beta} \operatorname{rank}(\beta K - \alpha M)$. If $\operatorname{rank}(\beta K - \alpha M) < r$, then $(\alpha, \beta) \neq (0, 0)$ is called *eigenvalue*.

Definition 1.17 (Regular eigenvector)**.** Let x be an eigenvector such that $\beta K x = \alpha M x$. If (α, β) is a unique eigenvalue, then x is called a *regular* eigenvector.

Now we introduce special cases of the cosine-sine decomposition (CSD) and the generalized singular value decomposition (GSVD). We can use these to prove properties and solve GEPs with HPSD matrices.

Definition 1.18 (2-by-1 CS Decomposition [MC, §2.5.4], [Bai92, §2])**.** Let $n, r \in \mathbb{N}$, let $n \geq r$, let $Q \in \mathbb{C}^{2n,r}$ be isometric and partition Q as

$$Q = \begin{matrix} \\ n \\ n \end{matrix} \begin{matrix} r \\ \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} \end{matrix}.$$

Then there are unitary matrices $U_1, U_2 \in \mathbb{C}^{n,n}$, $V \in \mathbb{C}^{r,r}$ and non-negative diagonal matrices $\Sigma_1, \Sigma_2 \in \mathbb{R}^{n,r}$ such that

$$Q = \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} = \begin{bmatrix} U_1 & 0 \\ 0 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 \\ \Sigma_2 \end{bmatrix} V^*,$$

where furthermore

$$\Sigma_1 = \begin{matrix} \\ r \\ n-r \end{matrix} \begin{matrix} r \\ \begin{bmatrix} C \\ 0 \end{bmatrix} \end{matrix}, \Sigma_2 = \begin{matrix} \\ r \\ n-r \end{matrix} \begin{matrix} r \\ \begin{bmatrix} S \\ 0 \end{bmatrix} \end{matrix},$$

with $C^2 + S^2 = I_r$. If Q is real, then all matrices may be taken to be real.

Denote the diagonal entries of C by c_i and let s_i denote the diagonal entries of S, $i = 1, 2, \ldots, r$. Since it holds that $c_i^2 + s_i^2 = 1$, we can regard both variables as the cosine and sine values of an angle $\theta_i \in [0, \pi/2]$ so that $c_i = \cos\theta_i$ and $s_i = \sin\theta_i$.

Definition 1.19 (Generalized Singular Value Decomposition [MC, §6.1.6], [Bai92, §2])**.** Let $n, r \in \mathbb{N}$, $n \geq r$, let $A, B \in \mathbb{C}^{n,r}$. Then there are unitary matrices $U_1, U_2 \in \mathbb{C}^{n,n}$, $Q \in \mathbb{C}^{r,r}$, non-negative diagonal matrices $\Sigma_1, \Sigma_2 \in \mathbb{R}^{n,r}$, and an upper-triangular matrix $R \in \mathbb{C}^{r,r}$ such that

$$\begin{bmatrix} A \\ B \end{bmatrix} = \begin{bmatrix} U_1 & 0 \\ 0 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 \\ \Sigma_2 \end{bmatrix} \begin{bmatrix} 0 & R \end{bmatrix} Q^*.$$

It holds that

$$\Sigma_1 = \begin{matrix} \\ r \\ n-r \end{matrix} \begin{matrix} r \\ \begin{bmatrix} C \\ 0 \end{bmatrix} \end{matrix}, \Sigma_2 = \begin{matrix} \\ r \\ n-r \end{matrix} \begin{matrix} r \\ \begin{bmatrix} S \\ 0 \end{bmatrix} \end{matrix},$$

where $C^2 + S^2 = I_r$. If A and B are real, then all matrices may be taken to be real.

The pairs (c_i, s_i) are called *generalized singular value pairs* and $\sigma_i := {}^{c_i}/_{s_i}$ are the *generalized singular values* of the matrix pencil (A, B). We define $\sigma_i := \infty$ whenever $c_i = 0, s_i = 1$. Note that since c_i and s_i are trigonometric functions of the same angle θ_i, we can also define $\sigma_i := \cot\theta_i$. Moreover, with

$$X := Q \begin{bmatrix} I_{n-r} & 0 \\ 0 & R^{-1} \end{bmatrix}$$

the columns of X are called the *right generalized singular vectors* of (A, B).

The GSVD reduces to the CSD if $\left[\begin{smallmatrix} A \\ B \end{smallmatrix}\right]$ is isometric. Thus there are two ways in practice to compute the GSVD of a pair of matrices: we can either compute it directly using the algorithms in LAPACK [BD92; BZ93] or we can reduce GSVD calculation to the problem of computing the CSD by using orthogonal factorizations (see Section 3.4).

The GSVD provides a wealth of information about the matrices A and B and their properties, among other things the null space of A, the null space of B, and their intersection. Most importantly for this thesis, we can compute the GSVD in practice and there is a simple relation between the GSVD and Hermitian definite GEPs that captures many interesting properties that are specific to GEPs with HPSD matrices.

Theorem 1.6 ([Bai92, §4.2, §4.3]). *Let $A, B \in \mathbb{C}^{n,n}$, let*

$$\begin{bmatrix} A \\ B \end{bmatrix} = \begin{bmatrix} U_1 & 0 \\ 0 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 \\ \Sigma_2 \end{bmatrix} \begin{bmatrix} 0 & R \end{bmatrix} Q^*$$

be the GSVD of (A, B), let $r = \operatorname{rank}[A^*, B^*]$, and let $X = [x_1, x_2, \ldots, x_n] \in \mathbb{C}^{n,n}$ be the matrix of right singular vectors. Then we solved implicitly the generalized eigenvalue problem $A^*Ax = \lambda B^*Bx$, where

- $\lambda_i = \sigma_i^2$, $i = 1, 2, \ldots, r$,
- *the right singular vectors X of (A, B) are the eigenvectors of (A^*A, B^*B),*
- $[x_1, x_2, \ldots, x_{n-r}]$ *is an orthonormal basis for* $\ker A^*A \cap \ker B^*B$.
- (λ_i, x_{n-r+i}) *are eigenpairs, $i = 1, 2, \ldots, r$.*

If A and B are real, then all matrices can be taken to be real.

Note that (∞, x) is an eigenpair of (A^*A, B^*B) iff $(0, x)$ is an eigenpair of (B^*B, A^*A).

Proof. Partition Q as

$$\begin{array}{c} \\ Q = \end{array} \begin{array}{c} \begin{array}{cc} r & n-r \end{array} \\ \begin{bmatrix} Q_0 & Q_r \end{bmatrix} \end{array} n.$$

It holds that

$$\begin{bmatrix} 0 & R \end{bmatrix} Q^* = \begin{bmatrix} 0 & R \end{bmatrix} \begin{bmatrix} Q_0^* \\ Q_r^* \end{bmatrix} = RQ_r^*$$

as well as

$$\Sigma_1^*\Sigma_1 = \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} C \\ 0 \end{bmatrix} = C^2,\ \Sigma_2^*\Sigma_2 = S^2.$$

The rest of the proof is simple substitution. Because Q is unitary, for all $q_0 \in \operatorname{span} Q_0$, $Q_r^*q_0 = 0$ so that $A^*Aq_0 = 0$ and $B^*Bq_0 = 0$ hence Q_0 is indeed a basis for $\ker A^*A \cap \ker B^*B$. Furthermore, we have

$$A^*A = Q_rR^*\Sigma_1^*U_1^*U_1\Sigma_1RQ_r^* = Q_rR^*C^2RQ_r^*$$

and similarly $B^*B = Q_rR^*S^2RQ_r^*$. With $RQ_r^*X = I_r$, it follows that $X^*A^*AX = C^2$ and $X^*B^*BX = S^2$. □

Earlier we stated that hermiticity is not a property as powerful for GEPs as for SEPs. However, in this thesis the matrices are also positive semidefinite and this allows us to derive a number of useful properties based on Theorem 1.6.

Theorem 1.7. *Let $K, M \in \mathbb{C}^{n,n}$ be HPSD. Then there exists an invertible matrix $X \in \mathbb{C}^{n,n}$ that diagonalizes K and M simultaneously. If K and M are real, then X is real as well.*

Proof. Because K and M are HPSD, there exist matrices $A, B \in \mathbb{C}^{n,n}$ such that $K = A^*A$ and $M = B^*B$ [HJ12, Theorem 7.2.7]. We can now apply Theorem 1.6. Let

$$\begin{bmatrix} A \\ B \end{bmatrix} = \begin{bmatrix} U_1 & 0 \\ 0 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 \\ \Sigma_2 \end{bmatrix} \begin{bmatrix} 0 & R \end{bmatrix} Q^*$$

be the GSVD of (A, B), let $X \in \mathbb{C}^{n,n}$ be the matrix of right singular vectors of (A, B). By definition, $AX = U_1\Sigma_1$ and $BX = U_2\Sigma_2$. Consequently,

$$\begin{aligned} X^*KX = X^*A^*AX = \Sigma_1^*\Sigma_1 = C^2, \\ X^*MX = X^*B^*BX = \Sigma_2^*\Sigma_2 = S^2, \end{aligned}$$

where C and S are diagonal.

If K and M are real, then A and B can be taken to be real [HJ12, Corollary 7.2.9]. Then X will be real as well. □

Above we have shown that a Hermitian matrix pencil may be singular. If we are dealing with pairs of HPSD matrices, this is true if and only if the intersection of the null spaces is non-trivial.

Theorem 1.8. *Let $K, M \in \mathbb{C}^{n,n}$ be HPSD. $K - \lambda M$ is regular iff $\ker K \cap \ker M$ is trivial.*

Proof. Let $Z = K + M$. From Weyl's Theorem [HJ12, Theorem 4.3.1] we deduce

$$\lambda_{\min}(Z) \geq \lambda_{\min}(K) + \lambda_{\min}(M) \geq 0.$$

The inequality on the right-hand side is strict iff $\ker K \cap \ker M$ is trivial, i. e., $\det(\beta K - \alpha M) \not\equiv 0$ iff $\ker K \cap \ker M$ is trivial. Applying the definition of regularity completes the proof. □

Theorem 1.8 has a useful implication: we can determine if a matrix pencil with HPSD matrices is regular by examining the matrix kernels; singular pencils like in Example 1.4 are not possible with pairs of HPSD matrices. Furthermore, detecting the null space intersection allows us to calculate the unique eigenvalues and this is exactly what the GSVD does.

Theorem 1.9. *Let $K, M \in \mathbb{C}^{n,n}$ be HPSD. Then the subspace of non-regular eigenvectors is unique and orthogonal to the subspace of regular eigenvectors.*

Proof. Every Hermitian matrix is diagonalizable, i. e., there exists a basis of eigenvectors for $\mathbb{C}^n$. Moreover, eigenvectors corresponding to different eigenvalues are orthogonal. Consequently, the kernel of a Hermitian matrix is orthogonal to its range.

Let $\mathcal{N} = \ker K \cap \ker M$, let $\mathcal{R} = \operatorname{ran} K \cup \operatorname{ran} M$. If $\mathcal{N}$ is the set containing only the origin, then the GEP is regular by Theorem 1.8 and there is nothing to prove. Otherwise let $u \in \mathcal{N}$, $u \neq 0$, and let $v \in \mathcal{R}$, $v \neq 0$. If $v \in \operatorname{ran} K$, then v is orthogonal to u because K is Hermitian and for these matrices $\ker K \perp \operatorname{ran} K$. Similarly, if $v \in \operatorname{ran} M$, then v must be orthogonal to u as well. Consequently, $\mathcal{N}$ must be orthogonal to $\mathcal{R}$.

Note $\mathcal{R}$ and $\mathcal{N}$ are both invariant subspaces of K and M. Therefore they are also eigenspaces of (K, M). From Theorem 1.8, we know that the matrix pencil (K, M) projected onto $\mathcal{R}$ must be regular. We also know (K, M) projected onto $\mathcal{N}$ is a pair of zero matrices. Thus, all regular eigenvectors of (K, M) can be found in $\mathcal{R}$ which is orthogonal to $\mathcal{N}$. □

In this thesis, we also need graph theory. Let $G = (V, E)$ be an undirected graph with nodes $V = \{1, 2, \ldots, n\}$, edges $E \subseteq \{\{i, j\} | i, j \in V\}$, and edge weights (or costs) $c : E \to R$. More specifically, we are dealing with simple graphs in this thesis; these graphs have no loops $\{i, i\}$ and there is at most one edge between every pair of vertices. For certain problems in graph theory, unweighted graphs are used and for these it holds that $c \equiv 1$. We can represent an undirected simple graph with a real symmetric matrix and vice versa, a real symmetric matrix *induces* an undirected simple graph.

Definition 1.20 (Adjacency matrix [Sed02, §17.3, §17.5])**.** Given an undirected simple graph $G = (V, E)$, let $n = |V|$. The *adjacency matrix* A of G is real symmetric $n \times n$ matrix with entries

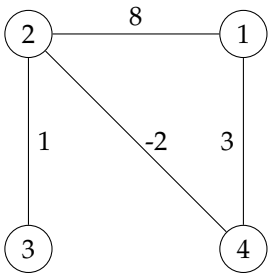
$$a_{ij} \coloneqq \begin{cases} c(\{i, j\}) & \text{if } \{i, j\} \in E, \\ 0 & \text{otherwise.} \end{cases}$$

Throughout this thesis, we ignore diagonal elements of Hermitian matrices when we discuss their induced graphs. For complex Hermitian matrices, we will explicitly provide a rule to compute the edge costs of the induced *weighted* graph. In graph theory, there is also the *adjacency list* representation of a graph [Sed02, §17.4]; from a numerical linear algebra perspective, this can be any sparse matrix representation of the adjacency matrix.

Example 1.5. Let

$$A = \begin{bmatrix} 0 & 8 & 0 & 3 \\ 8 & 0 & 1 & -2 \\ 0 & 1 & 0 & 0 \\ 3 & -2 & 0 & 0 \end{bmatrix}.$$

This induces the following graph:



Example 1.6. Let A be the $n \times n$ matrix of all ones except on the diagonal. Then the graph induced by A is the unweighted complete graph with n vertices.

2 Numerical Solution of Eigenvalue Problems

In this chapter, we will discuss how we can assess the accuracy of an approximate solution of a generalized eigenvalue problem with Hermitian positive semidefinite matrices (Section 2.1) and we will elaborate on the finite element method as a source of generalized eigenvalue problems in Section 2.2. We conclude the chapter with a brief remark on LAPACK.

2.1 Assessing Solution Accuracy

Given an approximate eigenpair $(\widetilde{\lambda}, \widetilde{x})$ to a GEP, we want to assess its accuracy. Obviously, we can examine the difference between $\widetilde{\lambda}$ and the exact eigenvalue λ it is supposed to approximate. Then $|\widetilde{\lambda} - \lambda|$ is called the *forward error* of $\widetilde{\lambda}$. Alternatively, we can try to modify the GEP so that $(\widetilde{\lambda}, \widetilde{x})$ is an exact eigenpair:

$$(K + \Delta K)\widetilde{x} = \widetilde{\lambda}(M + \Delta M)\widetilde{x}.$$

Given a norm for a pair of matrices, $\min_{\Delta K, \Delta M} \|(\Delta M, \Delta K)\|$ is the *backward error* [ASNA, §1.5]. For vector-valued quantities v, "measuring" a perturbation Δv is an obvious matter, e. g., we can take any vector p-norm. Selecting a suitable norm for matrix pairs is less obvious. In this thesis, we use the set of matrix polynomial norms proposed in [AAK11, §2].

Definition 2.1. Let $K, M \in \mathbb{C}^{n,n}$, let $\omega \in \mathbb{R}^2$, $\omega > 0$, let $P(t) = K - tM$. We define the matrix polynomial norm $\|P\|_{\omega,p,q}$ as follows:

$$\|P\|_{\omega,p,q} \coloneqq \|[{}^1\!/_{\omega_1}\|K\|_p, {}^1\!/_{\omega_2}\|M\|_p]\|_q.$$

Definition 2.2. Let $\Delta K, \Delta M \in \mathbb{C}^{n,n}$ be perturbations of square matrices K and M, respectively. Then we define the corresponding polynomial ΔP as

$$\Delta P(t) \coloneqq \Delta K - t\Delta M.$$

With the aid of these norms, we can define the backward error.

Definition 2.3 (Backward error of an eigenpair)**.** Let $(\widetilde{\lambda}, \widetilde{x})$ be an approximate eigenpair of the matrix pencil (K, M). Then the *backward error* of $(\widetilde{\lambda}, \widetilde{x})$ is defined as

$$\eta_{\omega,p,q}(\widetilde{\lambda}, \widetilde{x}) \coloneqq \min\{\|\Delta P\|_{\omega,p,q} : P(\widetilde{\lambda})\widetilde{x} + \Delta P(\widetilde{\lambda})\widetilde{x} = 0\}.$$

Example 2.1. Consider the backward error for an approximate eigenpair $(\widetilde{\lambda}, \widetilde{x})$ of the matrix pencil (K, M) defined in [HH98, §2.1]:

$$\min\{\varepsilon \geq 0 : (K + \Delta K)\widetilde{x} = \widetilde{\lambda}(M + \Delta M)\widetilde{x}, \|\Delta K\|_p \leq \varepsilon\|K\|_p, \|\Delta M\|_p \leq \varepsilon\|M\|_p\}.$$

We can find an equivalent definition utilizing the matrix polynomial norms with $\omega(p) \coloneqq [\|K\|_p, \|M\|_p]$ and $q = \infty$ so that

$$\|\Delta P\|_{\omega(p),p,\infty} = \min\left[\frac{\|\Delta K\|_p}{\|K\|_p}, \frac{\|\Delta M\|_p}{\|M\|_p}\right].$$

Let λ be a root of $P(t) = K - tM$ and consider the perturbed polynomial $P + \Delta P$. If $\|\Delta P(\lambda)\|$ is small, then $P + \Delta P$ will have a root $\widetilde{\lambda}$ near λ. Now given $h > 0$, let $f(h)$ be a function maximizing $|\widetilde{\lambda} - \lambda|$ subject to $\|\Delta P(t)\| \leq h$. Vice versa, if $(\widetilde{\lambda}, \widetilde{x})$ is an eigenpair approximation, then we can bound the forward error $|\widetilde{\lambda} - \lambda|$ by calculating $f(\eta_{\omega,p,q}(\widetilde{\lambda}, \widetilde{x}))$.

Definition 2.4 (Condition number of a simple eigenvalue [HH98, §2.2])**.** Let (λ, x) be an eigenpair of the matrix pencil (K, M), where λ is a simple, nonzero, finite eigenvalue. Given $\Delta\lambda$, Δx, let $\widetilde{\lambda} = \lambda + \Delta\lambda$, $\widetilde{x} = x + \Delta x$. Then the *condition number* of the eigenvalue λ is defined as

$$\kappa_{\omega,p,q}(\lambda, x) := \lim_{\varepsilon \to 0} \sup \left\{ \frac{1}{\varepsilon} \frac{|\Delta\lambda|}{|\lambda|} : P(\widetilde{\lambda})(\widetilde{x}) + \Delta P(\widetilde{\lambda})(\widetilde{x}) = 0 \right\},$$

subject to $\varepsilon \geq 0$, $\lim_{\varepsilon \to 0} \|\Delta x\| = 0$, and $\|\Delta P\|_{\omega,p,q} \leq \varepsilon \|P\|_{\omega,p,q}$.

Note that one must use the same norm for forward error, backward error, and condition number in order to get meaningful results. Also, the condition number is a property of the *problem* and independent of the method employed to calculate a solution. In practice, the first-order term of the Taylor expansion of f is used as condition number (we assume f is continuously differentiable) and consequently, the relationship between the forward and backward error is often expressed as

$$\text{forward error} \leq \text{condition number} \times \text{backward error}.$$

As a consequence, we can assess the accuracy of a an approximate solution without knowing exact solutions. If the condition number is small, then we call a problem *well conditioned*; if the condition number is large, then we call a problem *ill conditioned*. [ASNA, §1.6].

From a numerical linear algebra point of view, the best approach to assess the quality of a solution is the calculation of the backward error and to bound the forward error by computing the condition number for the following reasons:

- If we want to calculate the forward error, we need an exact solution which we may not have or which we cannot represent on a computer.
- There may be multiple solutions to the same problem (eigenvalue problems come to mind) forcing us to select one of them for the calculation of the forward error.
- After we calculated a value for the forward error, all we know about the backward error is that it is no larger than the forward error as if the problem was perfectly conditioned.
- Due to problem conditioning, there is no fool-proof criterion identifying accurate solutions using the forward error.

Furthermore, the following theorem allows us to derive an unambiguous criterion for an accurate solution that employs the backward error.

Theorem 2.1 ([ASNA, Theorem 2.2])**.** *Let $\alpha \in \mathbb{R}$ lie in the range of a finite precision arithmetic, let $\widetilde{\alpha}$ be the number closest to α in this finite precision arithmetic. Then*

$$|\widetilde{\alpha} - \alpha| \leq (1 + \delta)|\alpha|,\ \delta < \mathbf{u},$$

where $\mathbf{u}$ is the unit round-off.

We conclude, if the relative backward error $\eta^H_{p,q}(\widetilde{\lambda}, \widetilde{x})$ is at the level of the unit round-off and if the problem is well conditioned, that we can consider a solution to be accurate. Note the backward error here is based on norms and there may be circumstances where a *componentwise* error measure is more appropriate [ASNA, p. 4]. Next, we consider the preservation of problem structure when assessing accuracy.

Example 2.2. Let

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},$$

let $\widetilde{\lambda} = \imath$ be an approximate eigenvalue, where $\imath$ is the imaginary unit. Hermitian matrices have only real eigenvalues. Hence there is no Hermitian perturbation ΔA so that $\imath$ is an eigenvalue of $A + \Delta A$.

Perturbations that preserve (some of) the properties of a problem are called *structure preserving*. In this thesis, we will consider a perturbation structure preserving if it preserves hermiticity and we will indicate the corresponding backward error and its condition number with the superscript H, e. g., $\eta^H_{\omega,p,q}(\widetilde{\lambda}, \widetilde{x})$ and $\kappa^H_{\omega,p,q}(\widetilde{\lambda}, \widetilde{x})$. If ΔK and ΔM preserve hermiticity, then it holds that $\Delta P(t) = \Delta P^*(\bar{t})$, $t \in \mathbb{C}$ and for convenience, we will abbreviate this equality with $\Delta P = \Delta P^*$.

Definition 2.5 (Structured backward error of an eigenpair)**.** Let $(\widetilde{\lambda}, \widetilde{x})$ be an approximate eigenpair of the Hermitian matrix pencil (K, M). Then the *structured* backward error of $(\widetilde{\lambda}, \widetilde{x})$ is defined as

$$\eta^H_{\omega,p,q}(\widetilde{\lambda}, \widetilde{x}) \coloneqq \min\{\|\Delta P\|_{\omega,p,q} : P(\widetilde{\lambda})\widetilde{x} + \Delta P(\widetilde{\lambda})\widetilde{x} = 0,\ \Delta P = \Delta P^*\}.$$

Definition 2.6 (Structured condition number of a simple eigenvalue [HH98, §2.2])**.** Let (λ, x) be an eigenpair of the Hermitian matrix pencil (K, M), where λ is a simple, finite eigenvalue. Then the *structured* condition number of the eigenvalue λ is defined as

$$\kappa^H_{\omega,p,q}(\lambda, x) \coloneqq \lim_{\varepsilon \to 0} \sup \left\{ \frac{1}{\varepsilon} |\Delta\lambda| : P(\lambda + \Delta\lambda)(x + \Delta x) + \Delta P(\lambda + \Delta\lambda)(x + \Delta x) = 0 \right\},$$

subject to $\varepsilon \geq 0$, $\lim_{\varepsilon \to 0} \|\Delta x\| = 0$, $\|\Delta P\|_{\omega,p,q} \leq \varepsilon \|P\|_{\omega,p,q}$ and $\Delta P = \Delta P^*$.

Given a quantity s and its approximation $\widetilde{s}$, the term $|\widetilde{s} - s|$ is the *absolute error* while $|\widetilde{s} - s|/|s|$, $s \neq 0$, is the *relative error* [ASNA, §1.2]. The relative error is scale invariant (and dimensionless if s is a physical quantity) so throughout this thesis, we will use the relative backward error by using an appropriate weight vector ω:

$$\omega_{\text{rel}}(p) \coloneqq [\|K\|_p, \|M\|_p]. \tag{2.1}$$

For convenience, we introduce the following short hand:

$$\begin{aligned} \eta_{p,q}(\widetilde{\lambda}, \widetilde{x}) &\coloneqq \eta_{\omega_{\text{rel}}(p),p,q}(\widetilde{\lambda}, \widetilde{x}), & \kappa_{p,q}(\widetilde{\lambda}, \widetilde{x}) &\coloneqq \kappa_{\omega_{\text{rel}}(p),p,q}(\widetilde{\lambda}), \\ \eta^H_{p,q}(\widetilde{\lambda}, \widetilde{x}) &\coloneqq \eta^H_{\omega_{\text{rel}}(p),p,q}(\widetilde{\lambda}, \widetilde{x}), & \kappa^H_{p,q}(\widetilde{\lambda}, \widetilde{x}) &\coloneqq \kappa^H_{\omega_{\text{rel}}(p),p,q}(\widetilde{\lambda}). \end{aligned}$$

In this thesis, we will use the structured backward error $\eta^H_{F,2}(\widetilde{\lambda}, \widetilde{x})$ [AA11, §3] and its corresponding condition number because we can compute these quantities in a numerically stable way in time linear in the number of matrix entries (in time linear in n and the number of matrix

entries if the matrices are sparse). Given the structure preserving perturbations ΔK and ΔM minimizing $\eta_{F,2}^H(\widetilde{\lambda}, \widetilde{x})$, we are effectively calculating

$$\eta_{F,2}^H(\widetilde{\lambda}, \widetilde{x}) = \left\| \left[\frac{\|\Delta K\|_F}{\|K\|_F}, \frac{\|\Delta M\|_F}{\|M\|_F} \right] \right\|_2.$$

The following theorem describes how we can compute $\eta_{F,2}^H(\widetilde{\lambda}, \widetilde{x})$.

Theorem 2.2. *Let $(\widetilde{\lambda}, \widetilde{x})$ be an approximate eigenpair of the Hermitian matrix pencil (K, M), where $\widetilde{\lambda}$ is real finite and $\|\widetilde{x}\|_2 = 1$. Let $r = K\widetilde{x} - \widetilde{\lambda}M\widetilde{x}$. Then*

$$\eta_{F,2}^H(\widetilde{\lambda}, \widetilde{x}) = \sqrt{\frac{2\|r\|_2^2 - |r^*\widetilde{x}|^2}{\|K\|_F^2 + |\widetilde{\lambda}|^2\|M\|_F^2}}.$$

Proof. The theorem follows from [AA11, Theorem 3.10], where

$$\Lambda_m = [\|K\|_F, |\widetilde{\lambda}|\|M\|_F]$$

due to our use of the weight vector $\omega_{\text{rel}}(F)$. □

If (λ, x) is an eigenpair of the matrix pair (K, M), then $(1/\lambda, x)$ is an eigenpair of (M, K). We can exploit this fact to compute the backward error for eigenpairs with infinite eigenvalues.

Corollary 2.1. *Let $(\widetilde{\lambda}, \widetilde{x})$ be an approximate eigenpair of the Hermitian matrix pencil (K, M), where $\widetilde{\lambda}$ is real infinite and $\|\widetilde{x}\|_2 = 1$. Then*

$$\eta_{F,2}^H(\widetilde{\lambda}, \widetilde{x}) = \frac{1}{\|M\|_F}\sqrt{2\|M\widetilde{x}\|_2^2 - |\widetilde{x}^*M\widetilde{x}|^2}.$$

Note that we can explicitly compute the unique perturbations corresponding to the backward error $\eta_{F,2}^H(\widetilde{\lambda}, \widetilde{x})$.

Example 2.3. Let $c \geq 1$, let $K \coloneqq cI_n$, let $M \coloneqq I_n$. The eigenpairs of the matrix pencil are (c, e_i), $i = 1, 2, \ldots, n$. Consider the approximate eigenpair $(\widetilde{c}, e_1)$, $\widetilde{c} \geq 1$, $\widetilde{c} \neq c$, let $\tau \coloneqq \widetilde{c}/c$, and let $r = Ke_1 - \widetilde{c}Me_1 = (\widetilde{c} - c)e_1$ denote its residual. Then for the backward error $\eta_{2,2}^H(\widetilde{c}, e_1)$ we have [AA11, Theorem 3.10]:

$$\eta_{2,2}^H(\widetilde{c}, e_1) = \frac{\|r\|_2}{\sqrt{\|K\|_2^2 + |\widetilde{c}|^2\|M\|_2^2}} = \frac{|\tau - 1|}{\sqrt{\tau^2 + 1}}.$$

If $\widetilde{c}$ is very different from c, then τ is either very large or close to zero. In both cases, the backward error will be close to one, i. e., the matrix pencil (K, M) was perturbed strongly. Now consider the backward error $\eta_{F,2}^H(\widetilde{c}, e_1)$:

$$\eta_{F,2}^H(\widetilde{c}, e_1) = \sqrt{\frac{2\|r\|_2^2 - |e_1^*r|^2}{\|K\|_F^2 + |\widetilde{c}|^2\|M\|_F^2}} = \frac{1}{\sqrt{n}}\frac{|\tau - 1|}{\sqrt{\tau^2 + 1}} = \frac{1}{\sqrt{n}}\eta_{2,2}^H(\widetilde{c}, e_1).$$

In comparison to the backward error $\eta_{2,2}^H(\widetilde{c}, e_1)$, there is an additional factor $1/\sqrt{n}$. Thus, the larger the dimension of the matrices K and M, the smaller $\eta_{F,2}^H(\widetilde{c}, e_1)$. This dependency of the backward error $\eta_{F,2}^H(\widetilde{c}, e_1)$ on the matrix dimension is an undesirable property of the Frobenius norm.

Note we can explicitly compute the unique perturbations corresponding to the backward error $\eta_{F,2}^H$. For condition numbers of eigenvalues of Hermitian GEPs, it holds that structured condition numbers are not greater than the corresponding unstructured condition number. Hence we can use both condition numbers to find an upper bound for the forward error. Moreover, we deduce from the following theorem that both condition numbers are similar in our case ($p = F$ and $q = 2$).

Theorem 2.3 ([AAK11, Lemma 2.12])**.** *For a simple, finite, nonzero eigenvalue λ of an Hermitian matrix pencil (K, M), it holds that*

$$1/\sqrt{2}\,\kappa_{p,2}(\lambda, x) \leq \kappa_{p,2}^H(\lambda, x) \leq \kappa_{p,2}(\lambda, x),\ p = 2, F.$$

Proof. The proof can be found in [AAK11]. For the spectral case ($p = 2$), keep in mind that for Hermitian GEPs left and right eigenvectors are identical. Also, the authors assume normalized eigenvectors [AAK11, Eq. (2)]. □

There is no simple explicit expression for $\kappa_{F,2}^H(\lambda, x)$ [AAK11, §2.4][1] and in view of the bounds in Theorem 2.3, we chose to compute the unstructured condition number $\kappa(F, 2)\lambda, x$ instead knowing that it is a reasonable approximation to $\kappa_{F,2}^H(\lambda, x)$.

Theorem 2.4. *Let (λ, x) be an eigenpair of the Hermitian matrix pencil (K, M), where λ is simple and finite. Then we can compute the condition number $\kappa_{F,2}(\lambda, x)$ with*

$$\kappa_{F,2}(\lambda, x) = \frac{\|x\|_2^2}{|x^* M x|}\sqrt{\|K\|_F^2 + |\lambda|^2\|M\|_F^2}.$$

Proof. Insert $\omega_{\text{rel}}(F)$ into [AAK11, Equation (10)]. □

In order to compute condition numbers of infinite eigenvalues, recall that if (λ, x) is an eigenpair of the matrix pair (K, M), then $(1/\lambda, x)$ is an eigenpair of (M, K).

We can now approximate the forward error.

Theorem 2.5. *Let $(\widetilde{\lambda}, \widetilde{x})$ be an approximate eigenpair of an Hermitian matrix pencil (K, M), where $\widetilde{\lambda}$ is a simple, real finite eigenvalue and $\|\widetilde{x}\|_2 = 1$. Let $r = K\widetilde{x} - \widetilde{\lambda}M\widetilde{x}$. Then there exists an exact eigenvalue λ of (K, M) such that*

$$|\widetilde{\lambda} - \lambda| \leq \frac{1}{|\widetilde{x}^* M \widetilde{x}|}\sqrt{2\|r\|_2^2 - |r^*\widetilde{x}|^2}.$$

Proof. The error bound is the product of the backward error $\eta_{F,2}^H(\widetilde{\lambda}, \widetilde{x})$ and the corresponding condition number $\kappa_{F,2}(\widetilde{\lambda}, \widetilde{x})$. □

Numerical linear algebra problems are only one of a kind of the subproblems arising in scientific computing. Consider the work flow depicted in Figure 2.1. By the time we acquire an algebraic problem, the problem data may be polluted with multiple kinds of errors (see also [Bat96, Table 4.4]). Moreover, different users of scientific computing have varying accuracy requirements for their results. Therefore, in practice the most informative error measure of the quality of a solution may not be the backward error. We must also keep in mind that the solution quality is not the only relevant property in scientific computing, e. g., ease of implementation or wall-clock time of a solver may be important, too.

[1] See pp. 2218f instead of Section 5 for the explanation.

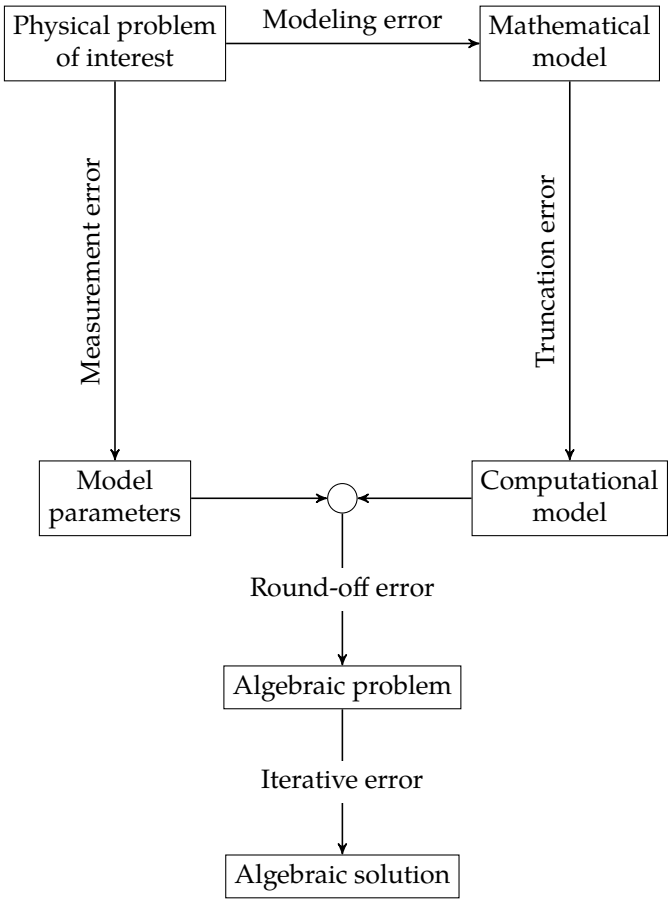


Figure 2.1: Sources of errors in scientific computing

The following expression is as common error measure for solutions of GEPs arising from structural mechanics problems [Bat96, 884f]:

$$\frac{\|K\widetilde{x} - \widetilde{\lambda}M\widetilde{x}\|_2}{\|K\widetilde{x}\|_2}. \tag{2.2}$$

This error measure is an upper bound for the structured backward error $\eta^H_{2,p}(\widetilde{\lambda}, \widetilde{x})$, $p = 2, \infty$, and we need following theorem to prove this.

Theorem 2.6. *Let $(\widetilde{\lambda}, \widetilde{x})$ be an approximate eigenpair of the Hermitian matrix pencil (K, M), where $\widetilde{\lambda}$ is real finite. Then*

$$\eta^H_{2,q}(\widetilde{\lambda}, \widetilde{x}) = \eta_{2,q}(\widetilde{\lambda}, \widetilde{x}),\ q = 2, \infty.$$

Proof. The proof for the case $q = \infty$ can be found in [HH98, Theorem 2.3], the proof for $q = 2$ is in [AA11, Theorem 3.10]. □

Theorem 2.7. *Let $(\widetilde{\lambda}, \widetilde{x})$ be an approximate eigenpair of the Hermitian matrix pencil (K, M), where $\widetilde{\lambda}$ is real finite. Then the error measure in equation* (2.2) *is an upper bound for the backward error $\eta^H_{2,q}(\widetilde{\lambda}, \widetilde{x})$, $q = 2, \infty$:*

$$\frac{\|K\widetilde{x} - \widetilde{\lambda}M\widetilde{x}\|_2}{\|K\widetilde{x}\|_2} \geq \eta^H_{2,q}(\widetilde{\lambda}, \widetilde{x}),\ q = 2, \infty.$$

Proof. Let $r = K\widetilde{x} - \widetilde{\lambda}M\widetilde{x}$. It holds that $\eta^H_{2,q}(\widetilde{\lambda}, \widetilde{x}) = \eta_{2,q}(\widetilde{\lambda}, \widetilde{x})$, $q = 2, \infty$. A closed expression for $\eta_{2,2}$ can be found in [AA11, Equation 1], where $\Lambda_m = [\|K\|_2, |\widetilde{\lambda}|\|M\|_2]$:

$$\eta_{2,2}(\widetilde{\lambda}, \widetilde{x}) = \frac{\|r\|_2}{\|\widetilde{x}\|_2}\sqrt{\|K\|_2^2 + |\widetilde{\lambda}|^2\|M\|_2^2}^{-1}.$$

The formula for $\eta_{2,\infty}$ is [HH98, Theorem 2.1]

$$\eta_{2,\infty}(\widetilde{\lambda}, \widetilde{x}) = \frac{\|r\|_2}{\|\widetilde{x}\|_2}(\|K\|_2 + |\widetilde{x}|\|M\|_2)^{-1}.$$

Substituting r into Equation (2.2) gives

$$\frac{\|K\widetilde{x} - \widetilde{\lambda}M\widetilde{x}\|_2}{\|K\widetilde{x}\|_2} = \frac{\|r\|_2}{\|K\widetilde{x}\|_2} \geq \frac{\|r\|_2}{\|K\|_2\|\widetilde{x}\|_2}.$$

The inequalities

$$\|K\|_2 \leq \|K\|_2 + |\widetilde{\lambda}|\|M\|_2,$$
$$\|K\|_2 \leq \sqrt{\|K\|_2^2 + |\widetilde{\lambda}|^2\|M\|_2^2}$$

complete the proof. □

Corollary 2.2. *Let $(\widetilde{\lambda}, \widetilde{x})$ be an approximate eigenpair of the Hermitian matrix pencil (K, M), where $\widetilde{\lambda}$ is real finite. Then the error measure in equation* (2.2) *is almost an upper bound for the backward error $\eta^H_{F,2}(\widetilde{\lambda}, \widetilde{x})$:*

$$\sqrt{2}\frac{\|K\widetilde{x} - \widetilde{\lambda}M\widetilde{x}\|_2}{\|K\widetilde{x}\|_2} \geq \eta_{F,2}(\widetilde{\lambda}, \widetilde{x}).$$

Proof. Apply Theorem 2.7 and use $\eta_{F,2}^H(\widetilde{\lambda},\widetilde{x}) \leq \sqrt{2}\eta_{2,2}(\widetilde{\lambda},\widetilde{x})$ [AA11, Theorem 3.10]. □

So reducing the error measure (2.2) is directly related to a reduction of the backward error which is a nice property reconciling this error measure with the desire in numerical linear algebra to minimize the backward error. At the same time, one needs to be cautious with Equation (2.2) if the stiffness matrix is badly conditioned because then $\|Kx\|_2 \ll \|K\|_2\|x\|_2$ so that

$$\frac{\|K\widetilde{x} - \widetilde{\lambda}M\widetilde{x}\|_2}{\|K\widetilde{x}\|_2} \gg \eta_{2,q}(\widetilde{\lambda},\widetilde{x}),\ q = 2,\infty.$$

In this case, the backward error may be at the round-off level while the error measure (2.2) is a large overestimate.

For structure preserving (hermiticity, definiteness) condition numbers of an eigenvalue with multiplicity larger than one, see [Nak12, §3]. Note that GEP solvers based on GSVD reduction (see Section 3.2.4) preserve semidefiniteness, too. The structured backward error $\eta_{p,q}^H(\widetilde{\lambda},\widetilde{x})$ preserves hermiticity and it also preserves semidefiniteness if it is infinite for every negative or complex eigenvalue.

2.2 Algebraic Eigenvalue Problems and the Finite Element Method

In this section, we focus on the origins of matrices arising from finite element discretizations in structural mechanics. This section has its own notation and all integrals are Lebesgue integrals, all derivatives are weak derivatives (see below).

Let Ω be a simply connected open set in d-dimensional space with a piecewise smooth boundary $\partial\Omega$, let $u : \Omega \to \mathbb{R}$, let

$$\operatorname{div} u = \sum_{i=1}^{d} \frac{\partial u}{\partial x_i}$$

denote the divergence, let ∇u denote the gradient of u, let $\cdot$ denote the scalar product. Let

$$\mathcal{L}u(x) = -\operatorname{div}(A(x)\nabla u) + b(x)\cdot\nabla u + c(x)u \tag{2.3}$$

be a second-order linear differential operator with coefficient functions $A : \Omega \to \mathbb{R}^{d,d}$, $b : \Omega \to \mathbb{R}^d$, and $c : \Omega \to \mathbb{R}$. $A, b, c \in L^\infty(\Omega)$, where $L^p(\Omega)$ is a Lebesgue space of integrable functions [GRS07, §3.2]

$$L^p(\Omega) \coloneqq \left\{ v : \Omega \to \mathbb{R} \;\middle|\; \int_\Omega |v|^p \,\mathrm{d}x < \infty \right\},\ p = 1, 2, \ldots,$$
$$L^\infty(\Omega) \coloneqq \{ v : \Omega \to \mathbb{R} \mid \operatorname{ess\,sup}_\Omega |v| < \infty \}$$

with norms

$$\|v\|_{L^p(\Omega)} \coloneqq \left(\int_\Omega |v|^p \,\mathrm{d}x \right)^{1/p},\ p = 1, 2, \ldots,$$
$$\|v\|_{L^\infty(\Omega)} \coloneqq \operatorname{ess\,sup}_\Omega |v|.$$

Moreover, we can introduce a scalar product on $L^2(\Omega)$:

$$(u, v) \coloneqq \int_\Omega uv \,\mathrm{d}x.$$

We want to find eigenvalues λ and eigenfunctions u of $\mathcal{L}$ subject to $u|_\Gamma = g$. Without loss of generality, we may assume $g \equiv 0$ [SF73, p. 70].

Definition 2.7 (Continuous eigenvalue problem [SF73, §6.1])**.** Let $\mathcal{L}$ be a second-order differential operator. We are looking for functions $u : \Omega \to \mathbb{R}$ and values $\lambda \in \mathbb{C}$ such that

$$\begin{aligned} \mathcal{L}u &= \lambda u \quad \text{in } \Omega, \\ u &= 0 \quad \text{on } \partial\Omega\,. \end{aligned}$$

The pair (λ, u) is called *eigenpair*, λ is called *eigenvalue*, and u is called *eigenfunction*.

Solving the continuous eigenvalue problem using Equation 2.3 requires every solution u to be differentiable twice and we can ease this requirement. Let $v : \Omega \to \mathbb{R}$ and consider the integral

$$-\int_\Omega v \operatorname{div}(A(x)\nabla u)\,\mathrm{d}x + \int_\Omega vb(x)\cdot\nabla u\,\mathrm{d}x + \int_\Omega vc(x)u\,\mathrm{d}x = \lambda\int_\Omega vu\,\mathrm{d}x.$$

We can apply Green's formula [NSV09, Equation (9)]

$$\int_\Omega v \operatorname{div} w\,\mathrm{d}x = -\int_\Omega \nabla v\cdot w\,\mathrm{d}x + \int_{\partial\Omega} vw\cdot n\,\mathrm{d}x,$$

where $w : \Omega \to \mathbb{R}^d$ and n is the unit normal vector on $\partial\Omega$. With the added requirement $v|_{\partial\Omega} = 0$, the integral transforms to

$$\int_\Omega \nabla v\cdot A(x)\nabla u\,\mathrm{d}x + \int_\Omega vb(x)\cdot\nabla u\,\mathrm{d}x + \int_\Omega vc(x)u\,\mathrm{d}x = \lambda\int_\Omega vu\,\mathrm{d}x. \tag{2.4}$$

Here, it suffices if u and v are differentiable once. We will now specify the function spaces to which u and v must belong to in order to solve the continuous eigenvalue problem and Equation (2.4). This will also permit us to clarify existence and uniqueness of solutions. First, we have to introduce multi-indices $\alpha := (\alpha_1, \alpha_2, \ldots, \alpha_n)$, $\alpha \geq 0$ [SF73, p. 137]. Let

$$|\alpha| := \sum_{i=1}^n \alpha_i,$$

let

$$D^\alpha u := \frac{\partial^{|\alpha|}u}{\partial x_1^{\alpha_1}\partial x_2^{\alpha_2}\ldots\partial x_n^{\alpha_n}}.$$

$C^p(\Omega)$ will denote the space of p-times continuously differentiable functions, $p = 1, 2, \ldots, \infty$, and $C_0^p(\Omega)$ is the subset of functions in $C^p(\Omega)$ with compact support [GRS07, p. XII].

Definition 2.8 (Weak derivative [GRS07, p. 131])**.** Let $u, w \in L^1(\Omega)$. If

$$\int_\Omega uD^\alpha v\,\mathrm{d}x = (-1)^{|\alpha|}\int_\Omega vw\,\mathrm{d}x$$

holds for all $v \in C_0^\infty(\Omega)$, then $D^\alpha u := w$ is called the *weak derivative* of u with respect to α.

By applying the definition of weak derivatives to all first-order derivatives, we can similarly define the weak gradient and the weak divergence of a function [GRS07, p. 132].

Definition 2.9 (Sobolev space $H^k(\Omega)$[SF73, p. 298] [NSV09, §2.1])**.** Let $k \in \mathbb{N}$. Then we define the Sobolev space $H^k(\Omega)$ as

$$H^k(\Omega) \coloneqq \{v : \Omega \to \mathbb{R} \mid D^\alpha v \in L_2(\Omega),\ |\alpha| \leq k\}.$$

$H^k(\Omega)$ is an inner product space with the scalar product

$$(u, v)_k \coloneqq \sum_{|\alpha| \leq k} (D^\alpha u, D^\alpha v).$$

The scalar product induces the norm $\|\cdot\|_k$:

$$\|v\|_k \coloneqq \sqrt{\sum_{|\alpha| \leq k} \|D^\alpha v\|^2_{L^2(\Omega)}}.$$

We will look for the solutions of Equation (2.4) in $H^1_0(\Omega)$.

Definition 2.10 ([SF73, pp. 11ff.] [GRS07, §3.2])**.** $H^1_0(\Omega)$ is the completion of $C^\infty_0 \cap H^1(\Omega)$ with respect to $\|\cdot\|_{H^1(\Omega)}$, i. e., for every Cauchy sequence (u_k), $u_k \in C^\infty_0$, there exists a $u \in H^1_0(\Omega)$ such that

$$\lim_{n\to\infty} \|u_k - u\|_{H^1(\Omega)} = 0.$$

Defining the bilinear form $a : H^1_0(\Omega) \times H^1_0(\Omega) \to \mathbb{R}$ by

$$a(u, v) \coloneqq \int_\Omega \nabla v \cdot A(x)\nabla u \,\mathrm{d}x + \int_\Omega v b(x) \cdot \nabla u \,\mathrm{d}x + \int_\Omega v c(x) u \,\mathrm{d}x$$

allows us to write the eigenvalue problem succinctly: find $u \in H^1_0(\Omega)$ and $\lambda \in \mathbb{R}$ such that

$$a(u, v) = \lambda(u, v)$$

for all $v \in H^1_0(\Omega)$.

If the eigenvalue problem arises from a structural mechanics problem [Bat96, §4] [Coo+01, §2.6], then u is the displacement, $c(x) \equiv 0$, and $A(x)$ is Hermitian as well as positive semidefinite. Moreover, if there are no rigid-body modes or mechanisms in the model underlying the PDE, then $A(x)$ is positive definite. For the problems in this thesis, we will ignore damping, hence $b(x) \equiv 0$. The resulting eigenvalue problems are used for analysis of system stability [Bat96, §3.2.3], modal analysis (mode superposition, [Bat96, §9.3]), or analysis of free vibrations. For the remainder of this section, we assume $A(x)$ is HPD. It follows, in structural mechanics $\mathcal{L}$ is a linear, self-adjoint, elliptic operator and $a(\cdot, \cdot)$ simplifies to

$$a(u, v) = \int_\Omega \nabla v \cdot A(x)\nabla u \,\mathrm{d}x.$$

Now we discretize the problem. Let $V_h \subset H^1_0(\Omega)$ have dimension n, let the *ansatz functions* $\phi^h_1, \phi^h_2, \ldots, \phi^h_n \in V_h$ be a basis of V_h, let h be a parameter describing the discretization. Then an eigenpair (λ^h, u^h) of the discretized problem must fulfill $a(u^h, v^h) = \lambda(u^h, v^h)$ for all $v^h \in V_h$. Let $u^h = \sum_{i=1}^n x^h_i \phi^h_i$, $x^h_i \in \mathbb{R}$. Using linearity of $a(\cdot, \cdot)$ and $(\cdot, \cdot)$, the equality

$$\sum_{i=1}^n a(\phi^h_i, \phi^h_j) x^h_i = \lambda^h \sum_{i=1}^n (\phi^h_i, \phi^h_j) x^h_i$$

must hold for all $\phi_j^h \in V_h$. This an algebraic eigenproblem. Let $K = [a(\phi_i^h, \phi_j^h)]_{i,j=1}^n \in \mathbb{R}^{n,n}$, let $M = [(\phi_i^h, \phi_j^h)]_{i,j=1}^n \in \mathbb{R}^{n,n}$. In order to find approximate eigenpairs (λ^h, u^h), we have to solve the algebraic GEP

$$Kx^h = \lambda^h M x^h.$$

So far we have described the Ritz-Galerkin method [GRS07, pp. 151f]. Let us partition Ω into smaller, geometrically simple subdomains and let each ansatz function be a low-degree polynomial that is nonzero only on a few subdomains while enforcing compatibility requirements on the subdomain boundaries. Then we acquire a *finite element method* (FEM, [SF73, §1.5] [GRS07, §4]).

K is called *stiffness matrix* and M is called *mass matrix*. Both matrices are sparse, real symmetric positive definite Gram matrices. We can make statements about the condition numbers of the matrices although these are influenced by the properties of $A(x)$ and by the choice of the ansatz space V_h.

Theorem 2.8 ([SF73, Theorem 5.1])**.** *For each variational problem and each choice of finite element there exists a constant $c > 0$ such that*

$$\kappa(K) \leq c h_{\min}^{-2}.$$

The constant depends inversely on the smallest eigenvalue of the given continuous problem, and it increases if the geometry of the elements becomes degenerate.

The finer the mesh, the worse the conditioning of the stiffness matrix. This problem can only be solved by a change in the discretization, e. g., by using polynomial elements with higher degree or by improving the mesh. Nevertheless, the condition number of the stiffness matrix will always reflect bad conditioning of the differential operator $\mathcal{L}$ (physical ill-conditioning). The condition number of consistent mass matrices is always bounded by a constant independent of h but this constant may be large for certain choices of V_h [SF73, §5]. For a GEP with HPD matrices, the condition number of each eigenvalue is bounded by $\|M^{-1}\|_2 = \|M\|_2^{-1} \kappa_2(M)$ [Nak12, §3] and for this reason, a GEP arising from a FE discretization in structural mechanics is well-conditioned.

The theory in this section is based on consistent and conforming finite element formulations. In practice, modelling errors may cause the stiffness matrix to be singular if rigid-body modes are present or K may be conditioned considerably worse than predicted by Theorem 2.8 [Kan+14, §1, §3]. The mass matrix can be diagonal, singular, or even indefinite [Bat96, §4.2.4] [Coo+01, §11.3].

For the remainder of this section assume $\lambda_1 \leq \lambda_2 \leq \dots$ and $\lambda_1^h \leq \lambda_2^h \leq \dots \leq \lambda_n^h$.

Theorem 2.9 ([SF73, Theorem 6.1])**.** *Let $V_h \subset H_0^1(\Omega)$ with dimension n. Then there exists a constant $c > 0$ such that for small h by*

$$\lambda_i \leq \lambda_i^h \leq \lambda_i + 2c\lambda_i^2 h^2,\ i = 1, 2, \dots, n.$$

The smaller the eigenvalue, the better its approximation and this is a vital insight for the sparse eigensolvers. Note $\lambda_i \leq \lambda_i^h$ holds only if consistent and conforming FE formulations are used. The inequality does also not hold when a discrete mechanics model with point masses is used (mass lumping).

Definition 2.11 (Energy norm)**.** Let $u \in H_0^1(\Omega)$. Then the *energy norm* is defined as

$$\|u\|_A \coloneqq \sqrt{a(u,u)}.$$

Here, $a(u,u)$ is a norm because $A(x)$ is HPD and $H_0^1(\Omega)$ cannot contain non-zero constant functions (their derivatives are zero but they do not have compact support). $a(u,u)$ being a norm can be proved formally with the aid of the Poincaré-Friedrichs inequality [NSV09, §2.5.1].

Theorem 2.10 ([SF73, Theorem 6.2]). *Let $V_h \subset H_0^1(\Omega)$ with dimension n. Let $c_1, c_2 > 0$. If all eigenfunctions u_i are pairwise orthogonal with respect to the scalar product $(\cdot,\cdot)$, then for $i = 1, 2, \ldots, n$*

$$\|u_i - u_i^h\|_0 \leq c_1 \lambda_i h^2,$$
$$\|u_i - u_i^h\|_A \leq c_2 \lambda_i h.$$

The estimates are the best possible.

We want to emphasize that a successful finite element *analysis* [Bat96, §1.2] requires an holistic view of all the steps involved. If a FE solution is deemed too inaccurate, a numerical analyst may, e. g., decide to increase the degree of the polynomial ansatz functions, use a finer mesh, or both. This results in denser, larger matrices and hampers the solution of the algebraic problem. Vice versa, mindlessly computing all algebraic eigensolutions with a small backward error is a waste of resources because the larger algebraic eigenvalues λ^h are increasingly worse approximations to their continuous counterparts (Theorem 2.9). What is more, a small backward error does not rectify a large a priori error in the discretization step.

2.3 LAPACK

Throughout this thesis, we utilize the matrix operations presented in Section 1.2. Most of these are implemented in LAPACK using state-of-the-art algorithms, including the correct computation of rank-revealing QR factorizations, cf. [DB08].

3 Generalized Eigenvalue Problem Solvers

In this chapter we discuss solvers for dense generalized eigenvalue problems (GEPs) with Hermitian positive semidefinite (HPSD) matrices. For general GEPs, the QZ algorithm is available [MC, §7.7] and unfortunately it seems to be much harder to exploit symmetry than in the standard case. It is only when the matrices are also semidefinite that alternative approaches to the QZ algorithm appear, the most popular being the reduction to a standard eigenvalue problem (SEP) by means of a Cholesky decomposition of the mass matrix. Unfortunately, this method is only conditionally backward stable so we will explore alternative solvers for GEPs with HPSD matrices in this chapter. We will compare computational complexity (with respect to the flop count) of the solvers, present pseudocode, and perform numerical experiments. $\mathbf{u}$ signifies the unit roundoff [ASNA, §2.1] [MC, §2.7.3].

3.1 The Computational Complexity of Iterative Solvers

In this chapter, we want to calculate the computational complexity of each solver. We can do this by summing the flop count of every operation but in order to acquire meaningful numbers, we feel compelled to show flop counts for iterative processes separately. Therefore, for all non-iterative operations we list the complexity given in the literature; for all iterative operations we signify the complexity with functions f whose parameters are the dimensions of the problem. Note that many solvers preprocess the matrix at hand, so even if the matrix is not square, there may be only a single parameter for the dimension of the problem.

Example 3.1. We want to calculate the full singular value decomposition (SVD) of an $m \times n$ matrix A, $m \geq n$. The standard approach for this problem is to bidiagonalize A so that

$$U_1 B V_1^* \coloneqq A$$

and iteratively compute the SVD of the bidiagonal matrix B:

$$U_2 \Sigma V_2^* \coloneqq B.$$

Assuming $U \coloneqq U_1 U_2$ and $V \coloneqq V_1 V_2$ are computed directly from U_1 and V_1, we require $4m^2n + 4mn^2$ flops for the bidiagonalization of A [MC, §5.4.8] and $f_{U,\Sigma,V}(n)$ flops for computing the SVD of B so overall the procedures uses $4m^2n + 4mn^2 + f_{U,\Sigma,V}(n)$ flops. Note the single parameter given to the function $f_{U,\Sigma,V}$; for matrices with more rows than columns, the preprocessing step reduces A to an upper bidiagonal matrix with n entries on the diagonal and $n-1$ entries on the superdiagonal.

Denoting the computational complexity of iterative processes with opaque functions allows us to sidestep the problem that we were unable to find flop counts for the LAPACK GSVD and the LAPACK CSD solver in the literature.

In Table 3.1, we list the flop counts for common linear algebra operations on $m \times n$ matrices. For unitary transformations, we assume the elementary reflectors are stored and we omit terms of order one or lower. Table 3.2 contains flop counts for common matrix decompositions. The

Operation	Flops	Source
Dot product	$2n$	[MC, §1.1.5]
Forward/backward substitution	n^2	[MC, §3.1.1, §3.1.2]
Cholesky factorization	$\frac{1}{3}\, n^3$	[MC, §4.2.5]
Cholesky factorization with pivoting	$\frac{1}{3}\, n^3 + 4nr - 2r^2$	[HHL07, §1.1]
Accumulating an $m \times n$ isometric matrix from k elementary reflectors	$4mnk - 2(m+n)k^2 + \frac{4}{3}\, k^3$	[MC, §5.1.6]
Householder QR factorization	$2mn^2 - \frac{2}{3}\, n^3$	[MC, §5.2.2]
Householder QR factorization with full column pivoting	$4mnr - 2(m+n)r^2 + \frac{4}{3}\, r^3$	[MC, §5.4.2]
Householder bidiagonalization	$4mn^2 - \frac{4}{3}\, n^3$	[MC, §5.4.8]
Householder tridiagonalization	$\frac{4}{3}\, n^3$	[MC, §8.3.1]

Table 3.1: Flop counts for common linear algebra operations of $m \times n$ matrices with $m \geq n \geq r, k$, where r is the rank of the matrix.

workspace size is depending on the exact solvers used for a given factorization and in this thesis, the author used the LAPACK Divide-and-Conquer solver for Hermitian SEPs (xSYEVD) which has the highest minimal workspace demand of all Hermitian eigensolvers in LAPACK. In practice, the workspace demand may be even higher if blocking is used.

Decomposition	Preprocessing	Computed Values	Flop Count	Iterative function
Singular value decomposition (SVD)	Bidiagonalization	Σ	$4mn^2 - 4/3n^3$	$f_\Sigma(n)$
		Σ, V	$4mn^2$	$f_{\Sigma,V}(n)$
		U, Σ	$4m^2n + 4/3n^3$	$f_{U,\Sigma}(n)$
		U, Σ, V	$4m^2n + 8/3n^3$	$f_{U,\Sigma,V}(n)$
Eigendecomposition (Hermitian SEP)	Tridiagonalization	Λ	$4/3\, n^3$	$f_\Lambda(n)$
		X, Λ	$8/3\, n^3$	$f_{X,\Lambda}(n)$
2-by-1 CS decomposition (CSD)	Bidiagonalization	C, S	$8mn^2 - 8/3n^3$	$f_{C,S}(n)$
		C, S, V	$8mn^2 - 4/3n^3$	$f_{C,S,V}(n)$

Table 3.2: Flop counts for common decompositions of $m \times n$ matrices, $m \geq n$

3.2 Solving Generalized Eigenvalue Problems

In this section, we present three approaches to solving generalized eigenvalue problems with HPD and HPSD matrices.

3.2.1 QZ Algorithm

The QZ algorithm computes the generalized Schur decomposition of a GEP [MC, §7.7]. The QZ algorithm is backward stable but we will not consider it further for the following reasons:

- The QZ algorithm computes flags of invariant subspaces instead of eigenvectors.
- The algorithm may calculate complex eigenvalues; exact eigenvalues are always real.
- When computing only eigenvalues, the flop count is on the order of $30n^3$; $46n^3$ if at least one of the unitary matrices is accumulated. For comparison, SEP reduction (see Section 3.2.2) requires $14n^3$ flops when computing eigenvalues and eigenvectors.

3.2.2 SEP Reduction

The standard approach for solving GEPs with HPD matrices reduces the problem to a Hermitian standard eigenvalue problem by computing the Cholesky decomposition $LL^* \coloneqq M$ of the mass matrix and solving the SEP $L^{-1}KL^{-*}x_L = \lambda x_L$. SEP reduction can be used whenever the mass matrix M is HPD and the stiffness matrix K is Hermitian and with such matrices, the GEP is always regular. The flop count (cf. Table 3.3) is ca. $5n^3 + f_{X,\Lambda}(n)$. The workspace size $2n^2 + 6n + 1$ reflects the demands of the LAPACK Divide-and-Conquer Hermitian eigensolver xSYEVD.

For the computed eigenvalues $\widetilde{\lambda}_i$, it holds that [MC, p. 464, §8.7.2]

$$|\widetilde{\lambda}_i - \lambda_i| \approx \mathbf{u}\|L^{-1}KL^{-*}\|_2,\ i = 1, 2, \ldots, n.$$

Due to the congruence transformation, we have

$$\|L^{-1}KL^{-*}\|_2 \leq \|K\|_2\|M^{-1}\|_2 = {}^{\|K\|_2}\!/_{\|M\|_2}\,\kappa_2(M).$$

Note $\|K\|_2\|M^{-1}\|_2$ is an upper bound for the largest eigenvalue and so we can read off the error bounds in two ways:

- Similar to a standard eigenvalue problem, there is an absolute error depending on the machine epsilon and the largest possible eigenvalue.
- With unit norm mass matrices, the absolute error $\mathbf{u}\|K\|_2$ is magnified by the mass matrix condition number.

It follows that if M is ill conditioned, then there is a large absolute error in the eigenvalues. Hence SEP reduction is simple, exploits hermiticity, allows the computation of subsets of the eigenpairs, and benefits from the vast improvements of Hermitian eigensolvers. However it is only conditionally backward stable.

Operation	Function	Flops
Factorize $LL^* \coloneqq M$	xPOTRF	$\frac{1}{3}\, n^3$
Compute $L^{-1}KL^{-*}$	xSYGST	$\frac{4}{3}\, n^3$
Solve SEP	xSYEVD	$\frac{8}{3}\, n^3 + f_{X,\Lambda(n)}$
Revert basis change	xTRSM	n^3
Solve GEP	xSYGVD	$5n^3 + f_{X,\Lambda(n)}$

Table 3.3: Flop count for a GEP solver using SEP reduction (workspace size: $2n^2 + 6n + 1$ units)

3.2.3 SEP Reduction with Deflation

If the mass matrix is singular, then plain SEP reduction cannot be used. In this case one has to deflate the infinite eigenvalues from the matrix pencil and Algorithm 1 is an implementation of the deflation procedure described in [MX15, Algorithm 2]. In finite precision arithmetic, the deflation step may ensure completion of the Cholesky decomposition and it it improves the conditioning of the deflated GEP (if the full rank part of the mass matrix is well-conditioned). If the mass matrix is singular, then the GEP may be singular as well and while Algorithm 1 cannot deal with non-regular matrix pencils, it can recognize them with the aid of the following theorem.

Theorem 3.1. *Let $A, B \in \mathbb{C}^{n,n}$ Hermitian, where B is singular and partitioned as*

$$B = \begin{bmatrix} B_{11} & 0 \\ 0 & 0 \end{bmatrix},$$

$B_{11} \in \mathbb{C}^{p,p}$, $p < n$. Partition A conformally to B so that

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{12}^* & A_{22} \end{bmatrix}.$$

Let $x = [0, x_2] \in \mathbb{C}^n$, $x_2 \in \mathbb{C}^{n-p}$, i. e., x is partitioned conformally to A and B and $x \in \ker B$. Then

$$\begin{bmatrix} A_{12} \\ A_{22} \end{bmatrix} x_2 = 0$$

if and only if

$$x \in \ker A \cap \ker B.$$

Proof. Consider the multiplication Ax. This gives

$$Ax = \begin{bmatrix} A_{11} & A_{12} \\ A_{12}^* & A_{22} \end{bmatrix} \begin{bmatrix} 0 \\ x_2 \end{bmatrix} = \begin{bmatrix} A_{12}x_2 \\ A_{22}x_2 \end{bmatrix}.$$

This expression is zero iff $x \in \ker A$. Since we have by definition $x \in \ker B$, it follows that $Ax = 0$ iff $x \in \ker A \cap \ker B$. □

Using this theorem, we can detect non-regular matrix pencils by checking for singular values of $[K_{21}^{(M)}, K_{22}^{(M)}]$ with value zero.

Input: $K, M \in \mathbb{C}^{n,n}$ HPSD
Output: A matrix pencil (A, B) without infinite eigenvalues, orthonormal bases for the subspaces corresponding to the finite and infinite eigenvalues

function DEFLATE(K, M)
  if $M = 0$ **then**
    return [], [], [], I_n
  end if

  Compute eigendecomposition: $X_M \Lambda_M X_M^* \leftarrow M$
  Sort eigenvalues in ascending order (permute columns of X_M accordingly)
  $r_M \leftarrow \operatorname{rank} M$
  $p \leftarrow n - r_M$

  if $r_M = n$ **then**
    return K, M
  end if

  $K^{(M)} \leftarrow X_M^* K X_M$
  Compute SVD: $U \Sigma W^* \leftarrow K^{(M)}(:, 1:p)$
  if $\sigma_p = 0$ **then**
    Error: (K, M) is not regular
  end if

  $U_{12} \leftarrow U(1:p, p+1:n)$
  $U_{22} \leftarrow U(p+1:n, p+1:n)$
  $A \leftarrow U_{22}^*(K_{22}^{(M)} U_{22} + K_{21}^{(M)} U_{12})$
  $B \leftarrow U_{22}^* \Lambda_M(p+1:n, p+1:n) U_{22}$

  return $A, B, X_M U(:, p+1:n), X_M(:, 1:p)$
end function

Algorithm 1: Pseudocode for the deflation of infinite eigenvalues of a Hermitian matrix pencil

Let κ_M denote the spectral condition number of the full-rank part of M. Thus it holds that for the backward error of the deflation procedure that [MX15, p. 15]

$$\|\Delta M\|_2 \in \mathcal{O}([\kappa_M^2 \mathbf{u} + 1]\kappa_2(U_{22})^2 \|M\|_2 \mathbf{u}),$$
$$\|\Delta K\|_2 \in \mathcal{O}([\kappa_M \|U_{22}^{-1}\|_2 \|K\|_2 + \|\widehat{K}\|_2 \|U_{22}^{-1}\|_2 + \rho \|K_{22}^{(M)}\|_2 \kappa_2(U_{22})^2]\mathbf{u}),$$

where

$$U^* \begin{bmatrix} K_{11}^{(M)} \\ K_{21}^{(M)} \end{bmatrix} = \begin{bmatrix} \widehat{K} \\ 0 \end{bmatrix}$$

and

$$\rho \coloneqq \|K_{12}^{(M)} (K_{22}^{(M)})^{-1}\|_2.$$

Observe that the backward error can be arbitrarily large if the full-rank part of M is ill conditioned. The backward error can also be large if the smallest angle $\theta_{\min}$ between the subspaces corresponding to finite and infinite eigenvalues, respectively, is close to zero. It holds that [MX15, §3] $\rho = \cot \theta_{\min}$ and $\sigma_{\min}(U_{22}) = \sin \theta_{\min}$ so the smaller $\theta_{\min}$, the larger ρ and $\|Q_{22}^{-1}\|_2$. We can fix the former problem by perturbing the mass matrix and setting all eigenvalues below a given cut-off to zero but then we will have eigenvalues with infinite forward error (eigenvalues that used to be finite and were infinite after the perturbation) and we may accidentally run into the second problem.

Example 3.2. Let $M = \operatorname{diag}(0, \mathbf{u}, 1)$, let

$$K = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & c \\ 0 & c & 2c^2 \end{bmatrix},$$

where $c \geq 1$ (note K is always positive definite). It holds that

$$\kappa_M = 1/\mathbf{u}, U_{22} = [e_2, e_3], \|U_{22}^{-1}\|_2 = 1, \theta_{\min} = \pi/2,$$

so $\|\Delta M\|_2 = \|M\|_2$ and $\|\Delta K\|_2 \approx \|K\|_2$ because of the ill conditioning of the full-rank mass matrix part (the subspaces corresponding to the finite and infinite eigenvalues, respectively, are orthogonal). Consider the modified mass matrix $M' = \operatorname{diag}(0, 0, 1)$. Here, the full-rank matrix part is perfectly conditioned for the negligible cost of a perturbation with norm $\mathbf{u}$ but now the smallest angle between the subspaces is almost zero whenever $c \gg 1$. With $c \gg 1$, it holds that

$$\|U_{21}\|_2 = \frac{c}{\sqrt{1 + c^2}} \approx 1$$

thus $\theta_{\min} = \arccos \|U_{21}\|_2 \approx 0$ and again, we cannot bound the backward error in a useful way.

We can detect the cases where the backward error bound is large. The case $\kappa_M \gg 1$ can be recognized immediately after the eigenvalue decomposition of the mass matrix and small angles between subspaces corresponding to finite and infinite eigenvalues, respectively, can be detected by computing the spectral norms of the off-diagonal block of the matrix U from the SVD because $\cos \theta_{\min} = \|U_{21}\|_2 = \|U_{12}\|_2$.

We can use the deflation procedure to acquire a GEP solver for singular and ill-conditioned mass matrices by first deflating the infinite eiegenvalues of the matrix pencil, solving the deflated GEP, and lifting the computed eigenvectors afterwards. The flop count can be found in Table 3.4. It sums up to

$$20/3 n^3 + 6n^2 r + 4nr^2 + 11r^3 + f_{X,\Lambda}(n) + f_{X,\Lambda}(r) + f_{U,\Sigma}(n - r).$$

Operation	Function	Flops
Copy M	xLACPY	n^2
Eigendecomposition $X\Lambda X^* = M$	xSYEVD	$\frac{8}{3}\, n^3 + f_{X,\Lambda}(n)$
Compute $K^{(M)} \coloneqq X^* K X$	–	$4n^3$
Copy $K^{(M)}$	xLACPY	n^2
SVD $K^{(M)}(:, 1:p)$	xGESVD	$4n^2 r + \frac{8}{3} r^3 + f_{U,\Sigma}(n-r)$
Compute $A \coloneqq U_{22}^*(K_{22}U_{22} + K_{21}U_{21})$	–	$2nr^2 + r^3 + r^2$
Compute $B \coloneqq U_{22}^* \Lambda_M(p+1:n, p+1:n) U_{22}$	–	$2r^3 + r^2$
Solve deflated GEP $A - \lambda B$	xSYGVD	$\frac{16}{3}\, r^3 + \frac{1}{2}\, r^2 + f_{X,\Lambda}(r)$
Revert basis changes: $X \coloneqq X_M U(:, p+1:n) X_{AB}$	–	$2n^2 r + 2nr^2$
Copy $X_M(:, 1:p)$	xLACPY	$n^2 - nr$

Table 3.4: Flop count for a GEP solver using SEP reduction with deflation (workspace size: $4n^2 + 6n + 1$ units)

Bounding the run time is difficult. Consider the case $r = 1$, then the flop count is on the order of

$$7n^3 + f_{X,\Lambda}(n) + f_{X,\Lambda}(1) + f_{U,\Sigma}(n-1)$$

whereas for $r = n$

$$28n^3 + f_{X,\Lambda}(n) + f_{X,\Lambda}(n-1) + f_{U,\Sigma}(1).$$

Unfortunately, these bounds are not informative because with $r = 1$ the GEP will have dimension 1 and in the second case ($r = n - 1$) we have to compute the SVD for a $n \times 1$ matrix. That is, we compute the Euclidean norm of a vector. The minimum workspace size is $4n^2 + 6n + 1$ units.

3.2.4 GSVD Reduction

Given a GSVD solver for matrix pairs (A, B), we showed in Theorem 1.6 that we are implicitly computing the eigendecomposition of the matrix pencil (A^*A, B^*B) so given a suitable decomposition $A^*A \coloneqq K$ and $B^*B \coloneqq M$, we can employ the GSVD to solve a GEP. The GSVD reduction is able to detect non-trivial intersections of the null spaces of a matrix pair and return an orthonormal basis for it. Moreover, the GSVD reduction preserves hermiticity as well as semidefiniteness of the matrices.

To employ the GSVD we need suitable matrix decompositions such as eigendecompositions or Cholesky factorizations with pivoting. For example for the eigendecomposition of the stiffness matrix, we have $X_K \Lambda_K X_K^* \coloneqq K$. In this case, $A \coloneqq \Lambda^{1/2} X_K^*$. In this thesis, we will use the Cholesky factorization with pivoting because it is cheaper to compute and because non-positive elements on the diagonal are an unambiguous indicator for matrices that are not positive definite (rank determination by means of singular or eigenvalues is less tangible). Given the Cholesky decomposition with pivoting of mass and stiffness matrix, i. e.,

$$R_K^* R_K = \Pi_K K \Pi_K^*, R_M^* R_M = \Pi_M M \Pi_M^*,$$

where Π_K, Π_M are permutation matrices, we have to compute the GSVD of $(R_K \Pi_K, R_M \Pi_M)$.

The GSVD solvers in Section 3.4 are all backward stable but this is not sufficient to guarantee numerically stable computation of the eigenvectors because the upper-triangular matrix R may be ill-conditioned.

Theorem 3.2. *Let $A, B \in \mathbb{C}^{n,n}$ such that $K = A^*A$, $M = B^*B$. Let*

$$\begin{bmatrix} A \\ B \end{bmatrix} = \begin{bmatrix} U_1 & 0 \\ 0 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 \\ \Sigma_2 \end{bmatrix} \begin{bmatrix} 0 & R \end{bmatrix} Q^*$$

be the GSVD of (A, B). Let $r = \operatorname{rank}[A^*, B^*]$ and partition Q as

$$Q = \begin{matrix} \\ n \end{matrix} \begin{matrix} \begin{matrix} r & n-r \end{matrix} \\ \begin{bmatrix} Q_0 & Q_r \end{bmatrix} \end{matrix}.$$

Then it holds that

$$R^*R = Q_r^*(A^*A + B^*B)Q_r.$$

Proof. Because U_1 and U_2 are unitary,

$$Q_r^* \begin{bmatrix} A^* & B^* \end{bmatrix} \begin{bmatrix} U_1 & 0 \\ 0 & U_2 \end{bmatrix} \begin{bmatrix} U_1 & 0 \\ 0 & U_2 \end{bmatrix}^* \begin{bmatrix} A \\ B \end{bmatrix} Q_r = Q_r^*(A^*A + B^*B)Q_r.$$

Moreover, by construction

$$\begin{bmatrix} U_1 & 0 \\ 0 & U_2 \end{bmatrix}^* \begin{bmatrix} A \\ B \end{bmatrix} Q_r = \begin{bmatrix} \Sigma_1 \\ \Sigma_2 \end{bmatrix} R.$$

Hence

$$Q_r^* \begin{bmatrix} A^* & B^* \end{bmatrix} \begin{bmatrix} A \\ B \end{bmatrix} Q_r = R^* \begin{bmatrix} \Sigma_1^* & \Sigma_2^* \end{bmatrix} \begin{bmatrix} \Sigma_1 \\ \Sigma_2 \end{bmatrix} R = R^* \begin{bmatrix} C^2 + S^2 \end{bmatrix} R = R^*R.$$

□

Theorem 3.3 (cf. [Tas15, §3])**.** *Let $A, B \in \mathbb{C}^{n,n}$ such that $K = A^*A$, $M = B^*B$, let R be the upper-triangular matrix from the GSVD of (A, B), and let $v \in \operatorname{ran} K \cup \operatorname{ran} M$. Then*

$$\sqrt{\frac{\max(\|A\|_2^2, \|B\|_2^2)}{\min_{\|v\|_2=1} \|[Av, Bv]\|_2}} \leq \kappa_2(R) \leq \sqrt{\frac{\|A\|_2^2 + \|B\|_2^2}{\min_{\|v\|_2=1} \|[Av, Bv]\|_2}}.$$

Proof. It holds that $\sigma_j^2(R) = \sigma_j(R^*R)$, where $\sigma_j(R)$ denotes the jth singular value of R and $\sigma_j(R^*R)$ is the jth singular value of R^*R. In conjunction with the previous theorem and $\sigma_1 = \|\cdot\|_2$ this gives $\|R\|_2^2 = \|R^*R\|_2 = \|A^*A + B^*B\|_2$ which we can bound with

$$\max(\|A\|_2, \|B\|_2) \leq \|R\|_2 \leq \sqrt{\|A\|_2^2 + \|B\|_2^2}.$$

For the smallest singular value it holds that

$$\sigma_r(R)^2 = \sigma_r(R^*R) = \min_{\|v\|_2=1} \|[Av, Bv]\|_2,$$

where r is the dimension of $\operatorname{ran} K \cup \operatorname{ran} M$. Combining these formulae gives the bounds. □

We conclude, the matrix R in the GSVD is ill conditioned if either

- the norms of A and B differ by several magnitudes, or
- there is a vector w "close" to $\ker A \cap \ker B$.

Operation	Function	Flops
Compute $\|K\|_F$	xLANSY	n^2
Compute $\|M\|_F$	xLANSY	n^2
Scale mass matrix with $s := \|K\|_F / \|M\|_F$	xLASCL	$\frac{1}{2}\, n^2$
Cholesky decomposition $A^*A = K$	xPSTRF	$\frac{1}{3}\, n^3 + 4nr_K - 2r_K^2$
Cholesky decomposition $B^*B = sM$	xPSTRF	$\frac{1}{3}\, n^3 + 4nr_M - 2r_M^2$
GSVD	–	$f_{\text{GSVD}}(n)$
Compute $X := QR^{-1}$	xTRSM	nr^2

Table 3.5: Flop count for a GEP solver using GSVD reduction where $r_K := \operatorname{rank} K$, $r_M := \operatorname{rank} M$, and $r := \operatorname{rank} [A^*, B^*]$.

In the latter case, we mean that both of

$$\frac{\|Av\|}{\|A\|}, \frac{\|Bv\|}{\|B\|}, \|v\| = 1, v \in \operatorname{ran} K \cup \operatorname{ran} M,$$

are small. We can prevent the first case by scaling the matrices, so that $\|A\| \approx \|B\|$ whereas we are unable to do anything about the second case because here, both A and B are ill conditioned.

Example 3.3. Let

$$A = \begin{bmatrix} \mathbf{u} & 0 \\ 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Observe that this is a symmetric standard eigenvalue problem. Using Theorem 3.3, we can estimate the condition number of R as $\kappa_2(R) \approx 1/\mathbf{u}$. Let $s := \|A\|_2 / \|B\|_2$ and let R' be the upper triangular matrix belonging to the scaled matrix pair (A, sB). Then $\kappa_2(R') \leq \sqrt{2}$. This is an evident improvement of the condition number.

Example 3.4. Let

$$A = \begin{bmatrix} 1 & 0 \\ 0 & \mathbf{u} \end{bmatrix}, B = \begin{bmatrix} \mathbf{u} & 0 \\ 0 & 1 \end{bmatrix}.$$

For the matrix pairs (A, A) and (B, B) it holds that $\kappa_2(R) \approx 1/\mathbf{u}$. Observe that ill-conditioned matrices are a necessary but not a sufficient condition for a ill-conditioned R, e. g., for the matrix pencil (A, B) we have $\kappa_2(R) \leq \sqrt{2}$ (note the identical scaling of A and B).

3.3 Solving Standard Eigenvalue Problems

Solving SEPs is a standard problem in numerical linear algebra and subject to ongoing research. The preprocessing reduces the matrix to upper Hessenberg form and for Hermitian matrices, this means a reduction to real symmetric tridiagonal form. For general matrices, the Francis QR algorithm reduces a given (upper Hessenberg) matrix to Schur form [Fra61] [MC, §7.5] and selected eigenvectors can be computed with inverse iteration [MC, §7.6]. For Hermitian matrices there are several solvers available in LAPACK:

- Francis QR for Hermitian matrices [Dem97, §5.3.1] [MC, §8.3]

- Divide-and-Conquer (DC) [Dem97, §5.3.3]
- Bisection in conjunction with inverse iteration (BI) [Dem97, §5.3.4]
- Multiple Relatively Robust Representations (MRRR) [Dhi97]

See [Dem+07] for a performance and accuracy comparison between these solvers. BI and MRRR support computations of a subset of the eigenpairs. If only eigenvalues are desired, square-root-free variants of the QR algorithm [Par98, §8.15] exist which should be faster than the methods above (computing the square root of a floating point number is expensive, even with hardware support).

3.4 Computing the Generalized Singular Value Decomposition

The GSVD can be computed either directly or indirectly by reduction to the CS decomposition. In this section, we elaborate on both approaches. Note that both methods are backward stable. Denoting quantities computed in finite precision with a tilde, it holds that

$$\begin{aligned}
\|\widetilde{U}_1^*\widetilde{U}_1 - I\| &\leq \mathbf{u}, \\
\|\widetilde{U}_2^*\widetilde{U}_2 - I\| &\leq \mathbf{u}, \\
\|\widetilde{Q}^*\widetilde{Q} - I\| &\leq \mathbf{u}, \\
\|\widetilde{U}_1^* A\widetilde{Q} - \widetilde{\Sigma}_1\widetilde{R}\| &\leq \mathbf{u}\|A\|, \\
\|\widetilde{U}_2^* B\widetilde{Q} - \widetilde{\Sigma}_2\widetilde{R}\| &\leq \mathbf{u}\|B\|,
\end{aligned}$$

that is, to within round-off error, the computed matrices $\widehat{U}_1$, $\widehat{U}_2$, and $\widehat{Q}$ are unitary and the rows of $\widehat{U}_1^* A\widehat{Q}$ and $\widehat{U}_2^* B\widehat{Q}$ are parallel [BD92, §5.1].

3.4.1 Direct Computation

There are algorithms for the direct computation of the GSVD. Of these, the algorithm in [BD92] is implemented in LAPACK. The problem can be preprocessed with the algorithm in [BZ93] (also implemented in LAPACK) recognizing the null spaces of the matrices, the intersection of the null spaces, and reducing the remainder of the matrices to upper triangular form. The amount of work of the preprocessing step depends on the dimension of the matrices and their rank so even for square matrices, the flop count would be a polynomial in three variables. Thus we omit the computational complexity analysis of the GSVD preprocessing.

3.4.2 Computation via QR Factorizations and CSD

The computation of the GSVD by means of QR factorizations and the CSD is straightforward and shown in Algorithm 2 [MC, §8.7.5] [Bai92, §5.3]. The first step is an orthogonal factorization of the matrix $G = \left[\begin{smallmatrix} A \\ B \end{smallmatrix}\right]$ that reveals the rank r. Here, we used the QR decomposition with full column pivoting because of the simplicity and the speed. In order to determine the rank, we only have to examine the diagonal of the upper triangular factor. The second step is the calculation of the CSD. The third step serves to compute the right-hand side unitary matrix Q from the matrices V and R_1. The last step is to revert the column permutations in the first step.

In this implementation we determine the rank of R_1 by comparing the the modulus of the diagonal elements $(R_1)_{ii}$ with $n\mathbf{u}\|G\|_F$. Because of the column pivoting, it holds that $|(R_1)_{ii}| \geq |(R_1)_{jj}|$ for all $i < j$ so R_1 has numerical rank r iff

$$|(R_1)_{rr}| > n\mathbf{u}\|G\|_F, |(R_1)_{r+1,r+1}| \leq n\mathbf{u}\|G\|_F.$$

Input: $A, B \in \mathbb{C}^{n,r}$
Output: GSVD of (A, B), $U_1, U_2 \in \mathbb{C}^{n,n}$, $Q \in \mathbb{C}^{r,r}$, $\Sigma_1, \Sigma_2 \in \mathbb{C}^{n,r}$, $R \in \mathbb{C}^{r,r}$
function GSVD-VIA-QR+CSD(A, B)
$G \leftarrow \left[\begin{smallmatrix} A \\ B \end{smallmatrix}\right]$
QR decomposition with column pivoting: $Q_1 R_1 \leftarrow G\Pi$
Determine the rank r of R_1

2-by-1 CSD of $Q_1(:, 1:r)$, get U_1, U_2, V, Σ_1, Σ_2
RQ decomposition: $\begin{bmatrix} 0 & R \end{bmatrix} Q_2 \leftarrow V^* R_1(1:r, :)$
Revert column permutation: $Q \leftarrow \Pi Q_2^*$

return U_1, U_2, Q, Σ_1, Σ_2, R
end function

Algorithm 2: Computation of the GSVD by using QR factorizations and the CS decomposition

In Table 3.6, we list the flop count for our implementation when all matrices are square and of the same dimension. In this case, the overall flop count is

$$^{10}/_{3}n^3 + 4n^2 r + 12nr^2 + f_{C,S,V}(r).$$

We can bound the run time in the non-trivial cases by considering $r = 1$ and $r = n$ giving approximately

$$3n^3 + f_{C,S,V}(1)$$

for the order of the minimum flop count

$$19n^3 + f_{C,S,V}(n)$$

for the maximum.

3.4.3 Computation via QR Factorizations and SVD

From a mathematical point of view, we can calculate the 2-by-1 CSD of a matrix

$$Q = \begin{matrix} n \\ n \end{matrix} \overset{r}{\begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix}}$$

by calculating the SVD of one of the blocks Q_1, Q_2 or by computing the SVD of both blocks and by reordering the singular values and the singular vectors as necessary. Problems arise if singular values are clustered [Sut09, §1.1] and moreover, this approach does not exploit the fact that the SVD is dealing with blocks of an isometric matrix. The flop count of the GSVD solver employing QR factorizations and the 2-by-1 CSD from Table 3.6 holds except for the flop

Operation	Function	Flops
Copy K	xLACPY	n^2
Copy M	xLACPY	n^2
Determine $\|[A, B]\|_F$	xLANGE	$2n^2$
QR with full column pivoting	xGEQP3	$10/3\, n^3$
Copy R_1	xLACPY	nr
Accumulate Q_1	xORGQR	$4nr^2 - 2/3\, r^3$
2-by-1 CSD	xORCSD2BY1	$8nr^2 - 4/3\, r^3 + f_{C,S,V}(r)$
Copy V^*	xLACPY	r^2
Compute V^*R_1	xGEMM	$2nr^2$
Compute $RQ_2 \coloneqq V^*R_1$	xGERQF	$2nr^2 + 2/3\, r^3$
Accumulate Q_2	xORGRQ	$4n^2r - 4nr^2 + 4/3\, r^3$
Reorder columns of Q_2	xLAPMT	n^2

Table 3.6: Flop count for a GSVD solver using QR factorizations and the CS decomposition if all matrices are square (required workspace: $2n^2 + 17n - 4$ units)

count of the CSD computation. When using only one SVD, then the CSD can be computed in $4nr^2 + f_{\Sigma,V}(r)$ flops (cf. Table 3.2). Thus with one SVD, the worst-case flop count ($r = n$) reduces to ca. $17n^3 + f_{\Sigma,V}(n)$ flops. When computing two SVDs, there is a second bidiagonalization and another SVD (no matrix accumulation) costing $f_\Sigma(r)$. Furthermore, the workspace size demand reduces to $5n$ with $r = n$.

3.5 Numerical Experiments

In this section, we will compare the wall clock times of the solvers above on a single CPU. The wall clock time accuracy is one hundredth of a second. The tests are conducted with Netlib BLAS and LAPACK.

We use the backward error $\eta^H_{F,2}(\cdot,\cdot)$ defined in Section 2.1 to assess the accuracy of solutions and we expect every solver to compute eigenpairs (λ, x) with

$$\eta^H_{F,2}(\lambda, x) < n\varepsilon,$$

where $\varepsilon = 2\mathbf{u}$ is the machine epsilon. We require the GSVD solvers to recognize matrix pencils with non-trivial intersections of their null spaces. For non-regular eigenvectors x, we expect the solvers to return eigenpairs $(-1, x)$ and we measure their accuracy by computing

$$\eta^H_{F,2}(-1, x) \coloneqq \sqrt{\left(\frac{\|Kx\|_2}{\|K\|_F}\right)^2 + \left(\frac{\|Mx\|_2}{\|M\|_F}\right)^2}.$$

The test problems are all BCS structural engineering matrices [DGL89] with real symmetric positive definite (SPD) or real symmetric positive semidefinite (SPSD) matrices and no more than 3600 degrees of freedom as well as the NLEVP test problem "shaft" (stiffness and mass matrix only) [Bet+13]. Overall there are 21 test problems and a list of them can be found in Table 3.7. All pairs of test matrices were multiplied (in double precision) by an orthogonal matrix from both sides to avoid artificially small backward errors due to diagonal matrices. All test problems are solved in double and in single precision.

Problem	DOF	Problem	DOF
bcsstk01	48	bcsstk09	1083
bcsstk03	112	bcsstk27	1224
bcsstk04	132	bcsstk11	1473
bcsstk22	138	bcsstk12	1473
bcsstk05	153	bcsstk14	1806
NLEVP shaft	400	bcsstk26	1922
bcsstk06	420	bcsstk13	2003
bcsstk07	420	bcsstk23	3134
bcsstk20	485	bcsstk24	3562
bcsstk19	817	bcsstk21	3600
bcsstk08	1074		

Table 3.7: The name of the stiffness matrix for all dense test problems ordered by the number of degrees of freedom.

We use performance profiles [DM02] to visualize the results; for every solver s and for different values of τ, a performance profile shows the fraction $\rho_s(\tau)$ of all problems where the solver s is no more than τ times slower than the fastest solver (the fastest solver may be different for every problem). Inaccurate results or failure to compute the eigendecomposition are penalized by assigning large, artificial wall clock times.

In Figure 3.1, we see the performance profiles of the four dense GEP solvers when computing in single precision. Table 3.8 compares the relative wall clock times of the solvers considering only successful test cases. Figure 3.2 and Table 3.9 contain the results for the double precision calculations.

In Figure 3.1, one can see the standard solver and deflation solve only two thirds of all test problems successfully whereas the GSVD-based solvers always compute all eigenpairs accurately. If the standard and deflation solver compute accurate solutions, then they are also considerably faster than the GSVD-based solvers. Considering only the successfully solved problems, we can gather from Table 3.8 that the direct GSVD solver is on average twenty to thirty times slower than the fastest solver while QR+CSD solver is only four times slower.

In double precision the results are similar except

- the deflation solver completes successfully for all test problems,
- the standard solver solves an additional problem accurately, and
- the direct GSVD solver is slowed down significantly.

In fact, for one problem the direct GSVD solver is 95 times slower than the fastest solver.

As expected, the standard solver fails whenever the mass matrix is rank-deficient. The deflation solver fails whenever it detects non-trivial intersections of mass and stiffness matrix kernels and this happened often in single precision because of the ill-conditioned stiffness matrices in conjunction with rank-deficient mass matrices.

If a robust solver is needed, a GSVD-based solver using QR factorizations and CS decompositions is the method of choice. In double precision, the deflation solver is the method of choice because of the superior performance.

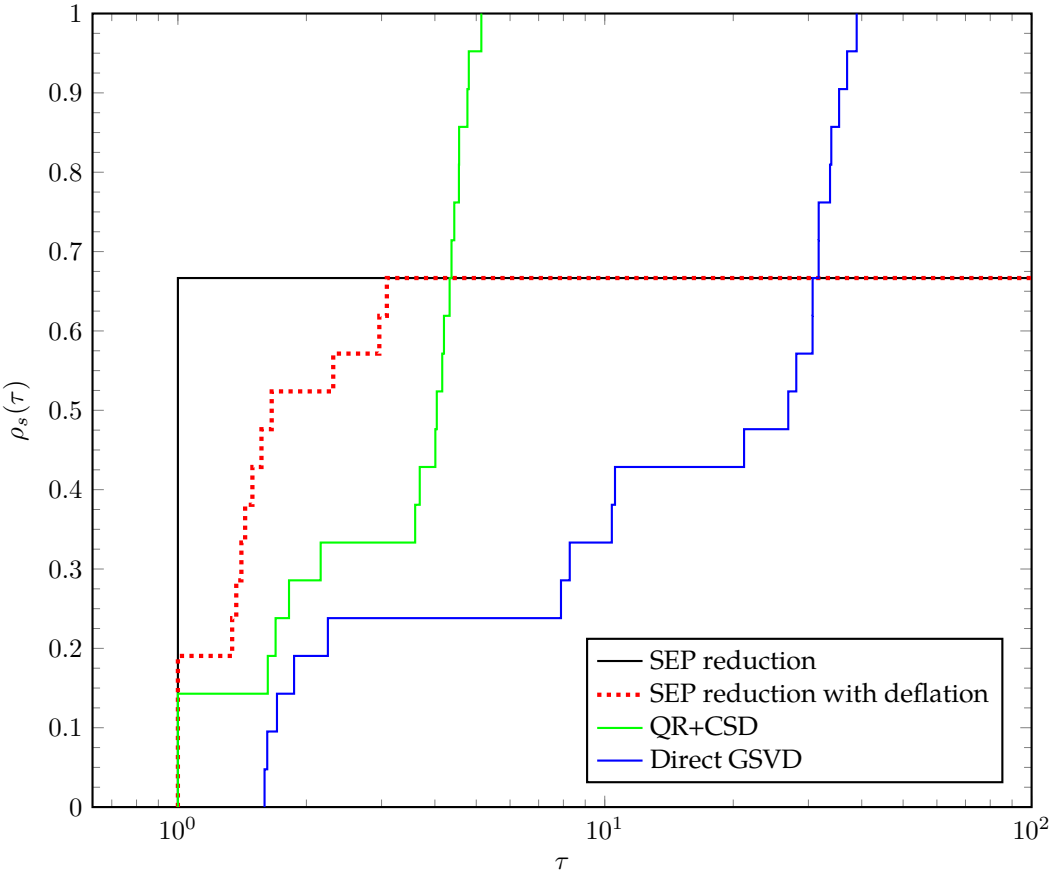


Figure 3.1: Performance profiles dense GEP solvers (32 Bit)

Solver s	$\min r_{p,s}$	$\max r_{p,s}$	mean $r_{p,s}$	median $r_{p,s}$
SEP reduction	1.00	1.00	1.00	1.00
SEP reduction with deflation	1.00	3.09	1.62	1.42
QR+CSD	1.00	5.14	3.38	4.04
Direct GSVD	1.60	38.93	20.29	26.91

Table 3.8: Relative wall clock times for all solvers (32 Bit). Only successfully solved test cases are considered, i. e., the solver did not terminate prematurely and for each eigenpair the backward error was less than $n\varepsilon$. The variables $r_{p,s}$ denote the *performance ratio* [DM02, §2].

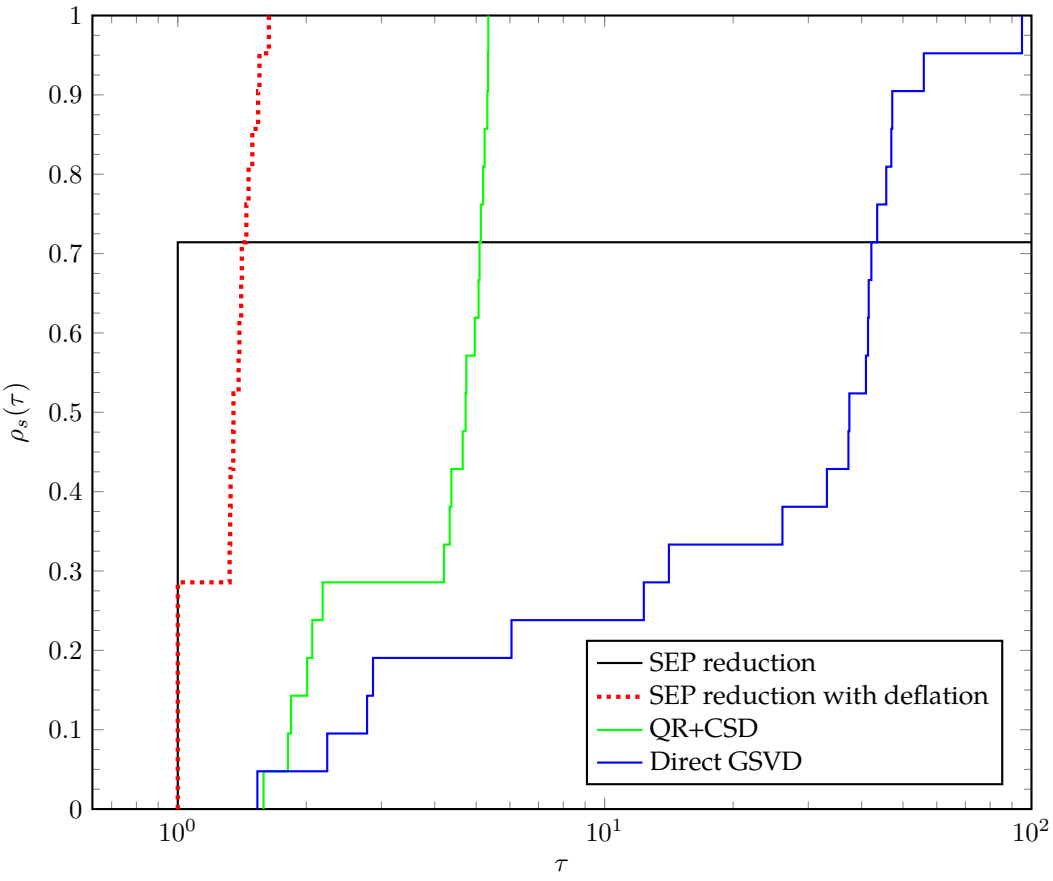


Figure 3.2: Performance profiles dense GEP solvers (64 Bit)

Solver s	$\min r_{p,s}$	$\max r_{p,s}$	mean $r_{p,s}$	median $r_{p,s}$
SEP reduction	1.00	1.00	1.00	1.00
SEP reduction with deflation	1.00	1.63	1.31	1.35
QR+CSD	1.59	5.34	4.06	4.72
Direct GSVD	1.54	95.01	32.21	37.44

Table 3.9: Relative wall clock times for all solvers (64 Bit). Only successfully solved test cases are considered, i. e., the solver did not terminate prematurely and for each eigenpair the backward error was less than $n\varepsilon$. The variables $r_{p,s}$ denote the *performance ratio* [DM02, §2].

Solver	Workspace	BC	WC
Direct GSVD	$1n^2 + 6n + 1$	–	–
SEP reduction	$2n^2 + 6n + 1$	$5n^3$	
QR+CSD	$3n^2 + 17n - 4$	$4n^3$	$21n^3$
SEP reduction with deflation	$4n^2 + 6n + 1$	$7n^3$	$28n^3$

Table 3.10: The GEP solvers discussed in this chapter, sorted by workspace size, showing the order of the best-case (BC) and the worst-case (WC) flop count.

3.6 Conclusion

Using the information from this chapter, we compiled workspace size, best-case, and worst-case flop counts for the GEP solvers in Table 3.10. Comparing the numbers in Table 3.10 with the results from Section 3.5, we have to concede that flop counts do not allow us to predict real-world performance (on modern CPUs). Consider for example highest-order term for the worst-case flop count. Judging by the coefficients, the speed of GSVD reduction (QR+CSD solver) and SEP reduction with deflation should be comparable whereas the pure SEP reduction solver should be significantly faster than both of them. This is not the case in practice.

Observe that in Figure 3.1 and Figure 3.2, greater robustness–the ability to deal with ill-conditioned matrices or even singular pencils–is synonymous with lesser speed. This should be kept in mind in the context of scientific computing and in finite element (FE) analyses, i. e., ill-conditioned matrices require more robust, more complex, slower algebraic solvers.

As we saw in Section 2.2, with consistent and conforming FE formulations the finite element mass matrices are always conditioned well and the algebraic eigenvalues approximate the continuous eigenvalues from above. In the face of the speed and simplicity of SEP reduction and in consideration of the fact that the matrix of eigenvectors simultaneously diagonalizes matrices, we question the use of ill-conditioned lumped mass matrices for mode superposition.

Another finding from the numerical experiments is that computing the GSVD directly is slower than the GSVD calculation via QR factorizations and CSD for the test problems in this thesis when LAPACK is used.

The Ingenuity of the CSD Approach for Solving GEPs

Dense solvers preprocess the matrices before starting the iterative phase and for different solvers, there are different standard reductions, e. g., Hermitian eigenvalue problems are reduced to real tridiagonal form. Surely, the reduction of an Hermitian GEP to a pair of Hermitian tridiagonal matrices is a good condensation of the original problem. In this section, we will show that a solver for a GEP with HPSD matrices using QR factorizations and the CSD

- computes basis for the $\ker K \cap \ker M$ and its orthogonal complement,
- implicitly reduces the matrices of the regular GEP part of (K, M) to tridiagonal form

before the iterative phase. This is undoubtedly a very efficient condensed representation. Note the simultaneous tridiagonalization of a pair of Hermitian indefinite matrices is also possible [Gar+03] and if the mass matrix is non-singular, then a tridiagonal-diagonal reduction is possible [Tis04].

Consider a GEP solver using QR factorizations and the CSD, let A, B such that $A^*A = K$ and $B^*B = M$. For the thin QR factorization with full column pivoting, it holds

$$Q_1 R_1 \coloneqq \begin{bmatrix} A \\ B \end{bmatrix} \Pi,$$

where $r = \operatorname{rank}[A^*, B^*]$, $Q_1 \in \mathbb{C}^{2n,r}$, $R_1 \in \mathbb{C}^{r,n}$, R_1 has full row rank, and $\Pi \in \mathbb{C}^{n,n}$ is a permutation matrix. Consider we compute an RQ decomposition of R_1 so that

$$\begin{bmatrix} 0 & R_g \end{bmatrix} Q_g^* \coloneqq R_1$$

and partition Q_g as

$$\begin{array}{c} \\ Q_g = \end{array} \begin{array}{c} \begin{array}{cc} n-r & r \end{array} \\ \begin{bmatrix} Q_0 & Q_s \end{bmatrix} \end{array} n.$$

The columns of Q_0 are a basis for $\ker K \cap \ker M$ and the columns of Q_s form a basis for the orthogonal complement so that $(Q_s^* K Q_s, Q_s^* M Q_s)$ is a regular matrix pencil (Theorem 1.8).

The 2-by-1 CSD solver bidiagonalizes Q_{1K} and Q_{1M} before starting the iterative phase. Let $U_{1B}, U_{2B} \in \mathbb{C}^{n,n}$, $V_B \in \mathbb{C}^r$ be unitary matrices, let $B_1, B_2 \in \mathbb{C}^{n,r}$ be upper bidiagonal. Then

$$Q_1 = \begin{bmatrix} U_{1B} & 0 \\ 0 & U_{2B} \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} V_B^*.$$

The accomplishment of this step is an implicit simultaneous tridiagonalization of the matrix pencil $(Q_s^* K Q_s, Q_s^* M Q_s)$. Let $S \coloneqq Q_s R_g^{-1} V_B$. Then

$$(S^* K S, S^* M S) = (B_1^* B_1, B_2^* B_2)$$

is a regular GEP of a pair of HPSD tridiagonal matrices. The transformation from a pair of tridiagonal to bidiagonal matrices is trivial if the matrices are positive definite (Cholesky decomposition can be used) but this is not the case for the problems in this thesis. Hence the bidiagonal reduction is a non-trivial step.

Robust SEP Reduction with Deflation

In Section 3.5, we saw the deflation procedure had to terminate often in single precision because the GEPs were singular. When solving GEPs with QR and CS decompositions, we are computing an orthonormal basis for the regular part of the GEP. Combining these two facts, we propose the following procedure:

- determine an orthonormal basis Q_s for the regular part of a GEP,
- project the GEP onto $\operatorname{span} Q_s$,
- use SEP reduction with deflation,
- lift the eigenvectors.

This solver is backward stable, retains hermiticity, and allows computing a subset of the eigenvalues.

4 Projection Methods for Large, Sparse Generalized Eigenvalue Problems

Given two subspaces $\mathcal{S}$ and $\mathcal{C}$ of $\mathbb{C}^n$, a *projection method* [Saa11, §4.3] for an eigenvalue problem tries to approximate an eigenpair $(\widetilde{\lambda}, \widetilde{x})$ so that $\widetilde{x} \in \mathcal{S}$ and $K\widetilde{x} - \widetilde{\lambda}M\widetilde{x} \perp \mathcal{C}$ for some given inner product. For Hermitian eigenvalue problems, *orthogonal* projection methods with $\mathcal{S} = \mathcal{C}$ are the most sensible choice. Examples for projections methods are Krylov subspace methods [Saa11, §6], Jacobi-Davidson methods [Saa11, §8.4] [FSV98], and LOBPCG [Kny01].

In this chapter, we will discuss approaches for finding the eigenpairs with the smallest eigenvalues of generalized eigenvalue problems (GEPs) $Kx = \lambda Mx$, where $K, M \in \mathbb{C}^{n,n}$ are large, sparse, Hermitian positive semidefinite (HPSD) matrices and (K, M) is regular. Every eigenvalue λ of such a problem is real and non-negative. Given $\lambda_c > 0$, we are looking for all eigenpairs (λ, x) where $\lambda \leq \lambda_c$.

4.1 Spectral Approximation for Large, Sparse Matrices

For large, sparse matrices we cannot use direct solvers because they are in practice guaranteed to compute full matrices at some point. Thus, we need to approach the (generalized) eigenvalue problem differently for these matrices. First of all, we will discuss basic concepts for solving large, sparse standard eigenvalue problem. Afterwards, we will show how these techniques apply to generalized eigenvalue problems.

Given a basis for a subspace, we can solve eigenvalue problems restricted to this subspace and this method is called the *Rayleigh-Ritz procedure* (see Algorithm 3, [Saa11, §4.3.1] [Par98, §11.3]). Intuitively, if the subspace is an eigenspace, then the Rayleigh-Ritz procedure should compute exact eigenpairs and the following theorem confirms this belief.

Input: $A \in \mathbb{C}^{n,n}$ diagonalizable, $S \in \mathbb{C}^{n,s}$ with full column rank
Output: Approximate eigenpairs $(\widetilde{\lambda}_i, \widetilde{x}_i)$ of A, $\widetilde{x}_i \in \operatorname{ran} S$, $i = 1, 2, \ldots, s$
function RAYLEIGH-RITZ(A, S)
Compute a thin QR decomposition: $QR \leftarrow S$
$A_Q \leftarrow Q^* A Q$
Compute eigendecomposition: $X_Q \Lambda_Q X_Q^* \leftarrow A_Q$
Lift eigenvectors: $\widetilde{X} \leftarrow Q X_Q$
return Λ_Q, $\widetilde{X}$
end function

Algorithm 3: Rayleigh-Ritz procedure

Theorem 4.1 ([Saa11, §4.3.1]). *Let $A \in \mathbb{C}^{n,n}$ be diagonalizable, let $S \in \mathbb{C}^{n,m}$ be isometric and such that $\operatorname{ran} S$ is an invariant subspace of A. Let (λ, x_S) be an eigenpair of S^*AS. Then (λ, Sx_S) is an exact eigenpair of A.*

We conclude that finding a subspace containing the eigenvectors of the desired eigenvalues is equivalent to computing a subset of the eigenpairs directly. To that end, the following method is helpful in finding a desired eigenvector.

Example 4.1 (Power method[MC, §7.3.1] [Saa11, §4.1.1] [Par98, §4.2])**.** Let $A \in \mathbb{C}^{n,n}$, let (λ_i, x_i) be the eigenpairs of A, where $\|x_i\| = 1$, $i = 1, 2, \ldots, n$. Let $v = \sum_{i=1}^{n} c_i x_i$, $c_1 \neq 0$. Assume that $|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_n|$. Thus, A has only simple eigenvalues and is diagonalizable. Finally, let $w_k = A^k v$, let $v_k = {}^{w_k}/_{\|w_k\|}$. Observe that

$$w_k = \sum_{i=1}^{n} \lambda_i^k c_i x_i.$$

Consequently, $\lim_{k\to\infty} v_k = x_1$.

The power method computes an eigenvector of the eigenvalue largest in modulus and it is applicable to non-diagonalizable matrices as well. Note that in finite precision arithmetic, the power method may converge to x_1 even if $c_1 = 0$ ($v \neq 0$); see [ASNA, §1.15]. We can quantify how quickly the method improves a given vector.

Definition 4.1 (Convergence factor)**.** Let $m \leq n$, let $A \in \mathbb{C}^{n,n}$ have eigenvalues $|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_m|$, where $|\lambda_1| > 0$. The ratio

$$\rho := \frac{|\lambda_2|}{|\lambda_1|}$$

is called the *convergence factor*.

The smaller the convergence factor, the faster the power method converges to an eigenvector of the eigenvalue largest in modulus. If this eigenvalue is a non-simple eigenvalue, then the power method computes only one of the eigenvectors and if there are two distinct eigenvalues with maximum modulus, then the power method calculates a linear combination of the eigenvectors corresponding to these two eigenvalues. We can avoid these problems by iterating with multiple vectors simultaneously and this will be discussed below. Furthermore, we can improve the convergence factor by transforming the spectrum of A.

Let $A \in \mathbb{C}^{n,n}$ have eigenvalues $|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_m|$. Let $\sigma \in \mathbb{C}$. Then the convergence factor ρ_1 of the power method applied to $A - \sigma I$ (*shifted power method*, [Saa11, §4.1.2]) is

$$\rho_1 = \max_{i\neq 1} \frac{|\lambda_i - \sigma|}{|\lambda_1 - \sigma|}.$$

The convergence factor ρ_2 of the power method applied to A^{-1} (inverse iteration, [Saa11, §4.1.3]) is

$$\rho_2 = \frac{|\lambda_m|}{|\lambda_{m-1}|},$$

i. e., we are computing an eigenvector for the eigenvalue smallest in modulus. Combining the two transformations above gives the *shift-and-invert* method $(A - \sigma I)^{-1}$ with convergence factor ρ_3:

$$\rho_3 = \max_{i\neq m} \frac{|\lambda_m - \sigma|}{|\lambda_i - \sigma|}.$$

Examples for more elaborate transformations are matrix polynomials (cf. [Saa11, §4.4, §7.1]) and Cayley transformations $(A - \sigma I)^{-1}(A - \tau I)$, $\tau \in \mathbb{C}$ [Bai+00, §11.2.1] (see [Kre11, §17.2, §17.3] for the effects on the spectrum).

Observe that if $\sigma \approx \lambda_m$, then one iteration of shift-and-invert computes a good approximation to an eigenvector of λ_m so if we know exact eigenvalues, then we can compute eigenvectors and vice versa. The Rayleigh quotient iteration tries to use this fact by selecting a different shift in every iteration.

Definition 4.2 (Rayleigh quotient [Saa11, §1.9.1])**.** Let $A \in \mathbb{C}^{n,n}$ be normal. Then the *Rayleigh quotient* $r : \mathbb{C}^n \setminus \{0\} \to \mathbb{C}$ is defined as

$$r(v) \coloneqq \frac{v^* A v}{v^* v}.$$

Note that if x is an eigenvector of A, then $r(x)$ calculates the corresponding eigenvalue. Furthermore, given A and a vector $v \neq 0$ the Rayleigh quotient minimizes $\|Av - \sigma v\|_2$. Given a vector $v_0 \neq 0$, the *Rayleigh quotient iteration* (RQI, [MC, §8.2.3] [Saa11, §4.1.3]) in Algorithm 4 computes an eigenpair (λ, x) unless a v_k is a linear combination of eigenvectors corresponding to different eigenvalues with the same modulus. If RQI converges, then it does so cubically for normal matrices [Par74, §13].

Input: $A \in \mathbb{C}^{n,n}$ normal with eigenvalues $|\lambda_1| > |\lambda_2| > \cdots > |\lambda_m|$, $v_0 \in \mathbb{C}^n \setminus \{0\}$
Output: An eigenpair of A

```
function RAYLEIGH-QUOTIENT-ITERATION(A, v_0)
    for k = 0, 1, 2, ... do
        σ_k ← r(v_k)
        if (σ_k, v_k) is sufficiently accurate then
            return (σ_k, v_k)
        end if
        w_{k+1} ← (A − σ_k I)^{-1} v_k
        v_{k+1} ← w_{k+1} / ‖w_{k+1}‖
    end for
end function
```

Algorithm 4: Rayleigh quotient iteration (RQI)

So far we have discussed iterative methods using a single vector during the iteration. These approaches have problems with distinct eigenvalues with identical modulus and they cannot compute the eigenspace of semisimple eigenvalues. *Subspace iteration* (SI, [Saa11, §5]) applies the power method to multiple linear independent vectors simultaneously and this rectifies the shortcomings of methods working with a single vector listed above but it also introduces new challenges. We have to avoid repeatedly computing eigenvectors of the dominant eigenvalues, i. e., we have to use *deflation* [Saa11, §4.2.3] [Par98, §5]. In finite precision arithmetic, deflation can be implemented in two ways. *Hard locking* [Saa11, §5.3.1] leaves converged eigenvectors unchanged and orthogonalizes the vectors spanning the search space and the converged eigenvectors, e. g., by means of Householder reflections [MC, §5.1.2] or two iterations of Gram-Schmidt (CGS2, [Gir+05, §3]). If the convergence criteria are too loose, hard locking may prevent the convergence of other eigenvectors [Sta05, §5]. *Soft locking* avoids this problem by marking eigenvectors fulfilling the convergence criterion and subsequently these marked eigenvectors are not subjected to inverse or power iterations but they are used for the Rayleigh-Ritz procedure.

A simple SI variant without locking can be found in Algorithm 5. It employs the Rayleigh-Ritz procedure in every iteration which improves its convergence properties considerably as the following theorem shows.

Theorem 4.2. *Let $A \in \mathbb{C}^{n,n}$ be normal with eigenpairs (λ_i, x_i), where $|\lambda_1| \leq |\lambda_2| \leq \cdots \leq |\lambda_n|$, $i = 1, 2, \ldots, n$. Let $S_k \in \mathbb{C}^{n,s}$ have full column rank, let P_k be an orthogonal projector onto $\operatorname{ran} S_k$, $k = 0, 1, 2, \ldots$. The columns of the matrices S_k span search spaces improved by inverse iterations. If $x_i \notin \ker S_k$ for all i and k, then there are non-negative constants $c_{k,i}$ such that*

$$\frac{\|(I - P_k)x_i\|_2}{\|x_i\|_2} \leq c_{k,i}\rho_i^k,$$

where ρ_i is the convergence factor of the ith approximate eigenvector:

$$\rho_i = \frac{|\lambda_i|}{|\lambda_{s+1}|}.$$

Proof. The theorem follows from [Saa11, Theorem 5.2] and the following discussion up to Equation (5.11) taking into account that A is normal. □

Input: $A \in \mathbb{C}^{n,n}$ normal, $S \in \mathbb{C}^{n,s}$ with full column rank
Output: The eigenpairs corresponding to the s eigenvalues (including multiplicities) smallest in modulus
function SUBSPACE-ITERATION(A, S_0)
  for $k = 0, 1, 2, \ldots$ **do**
    $\widetilde{\Lambda}_k, \widetilde{X}_k \leftarrow$ RAYLEIGH-RITZ(A, S_k)
    if eigenpairs are sufficiently accurate **then**
      return $\widetilde{\Lambda}_k$, $\widetilde{X}_k$
    end if
    $S_{k+1} \leftarrow A^{-1}\widetilde{X}_k$
  end for
end function

Algorithm 5: Subspace iteration with Rayleigh-Ritz procedure

The shift-and-invert procedure computes an eigenvector of σ if the shift is an eigenvalue. Thus, with SI and for any search space dimension, the search space may "collapse" into an eigenspace if the shift is an eigenvalue. This can be prevented with the idea underlying the next theorem [JKL99, §3].

Theorem 4.3. *Let $A \in \mathbb{C}^{n,n}$ be normal with simple eigenvalues, let (λ_i, x_i) be the eigenpairs of A, $i = 1, 2, \ldots, n$. Let $\sigma = \lambda_j$ and $\|x_j\|_2 = 1$ for some fixed $j \in \{1, 2, \ldots, n\}$. Let*

$$\mathcal{A} = \begin{bmatrix} A - \sigma I & x_j \\ x_j^* & 0 \end{bmatrix}.$$

Then $\mathcal{A}$ is normal as well as non-singular and it holds that

$$\mathcal{A}\begin{bmatrix} x_i \\ 0 \end{bmatrix} = (\lambda_i - \sigma)\begin{bmatrix} x_i \\ 0 \end{bmatrix}, \; i \neq j,$$
$$\mathcal{A}\begin{bmatrix} x_j \\ 1 \end{bmatrix} = \begin{bmatrix} x_j \\ 1 \end{bmatrix}.$$

The eigenvalues of $\mathcal{A}$ are 1, -1, and $\lambda_i - \sigma$, $i \neq j$. If A is Hermitian, then $\mathcal{A}$ is Hermitian, too.

Proof. For $x_i \neq x_j$, it holds that

$$\mathcal{A}\begin{bmatrix} x_i \\ 0 \end{bmatrix} = \begin{bmatrix} A - \sigma I & x_j \\ x_j^* & 0 \end{bmatrix}\begin{bmatrix} x_i \\ 0 \end{bmatrix} = \begin{bmatrix} (\lambda_i - \sigma)x_i \\ x_j^* x_i \end{bmatrix}.$$

Since A is normal, $x_j^* x_i = 0$ so we proved the first equality. The second equality can be shown to be correct by substitution.

Next, we show that $\mathcal{A}$ is normal and non-singular. The two equations in the theorem readily show n of the $n+1$ eigenpairs and with careful thought, we can construct another vector being orthogonal to all of these eigenvectors for which it holds that

$$\mathcal{A}\begin{bmatrix} -x_j \\ 1 \end{bmatrix} = -\begin{bmatrix} -x_j \\ 1 \end{bmatrix}.$$

We found $n+1$ orthogonal eigenvectors. Thus, $\mathcal{A}$ is normal [MC, Corollary 7.1.4]. As we can see, the $n+1$ eigenvalues of $\mathcal{A}$ are 1, -1, and $\lambda_i - \sigma$, $i \neq j$. A had only simple eigenvalues so $\lambda_i - \sigma \neq 0$, $i \neq j$, and consequently, $\mathcal{A}$ has full rank. □

The theorem can be generalized to semisimple eigenvalues by using an orthonormal basis of the eigenspace belonging to $\sigma = \lambda_j$, cf. [JL99]. Moreover, we can increase the numerical robustness when solving systems of equations with $\mathcal{A}$ by examining the magnitude of $|\lambda_i - \sigma|$, $i \neq j$.

For generalized eigenvalue problems with HPD matrices, we can use the theory above by setting $A \coloneqq M^{-1}K$. Note that we can avoid the need for mass matrix inverses, e. g., consider shift-and-invert:

$$(M^{-1}K - \sigma I)^{-1} = (M^{-1}K - \sigma M^{-1}M)^{-1} = (K - \sigma M)^{-1}M.$$

As long as the matrix pencil is regular and as long as the shift is not a generalized eigenvalue, we can solve systems of linear equations $(K - \sigma M)x = b$. Furthermore, if the mass matrix if M is HPD, then the Rayleigh quotient for GEPs is

$$r(v) \coloneqq \frac{v^* K v}{v^* M v}.$$

Eigenvectors for regular GEPs with HPSD matrices are still orthogonal but with respect to a different inner product. For non-standard inner products, CGS2 provides numerically stable orthogonalization [Roz+12, §5]. Pseudocode for the Rayleigh-Ritz procedure for GEPs can be found in Algorithm 6 and the generalization of Theorem 4.3 is given below.

Theorem 4.4 ([JL99, §3])**.** *Let $K, M \in \mathbb{C}^{n,n}$, where K is Hermitian, M is Hermitian positive definite, and such that (K, M) has only simple eigenvalues. Let (λ_i, x_i) be the eigenpairs of (K, M), $i = 1, 2, \ldots, n$. Let $\sigma = \lambda_j$ and $x_j^* M x_j = 1$ for some fixed $j \in \{1, 2, \ldots, n\}$. Let*

$$\mathcal{A} = \begin{bmatrix} K - \sigma M & M x_j \\ x_j^* M & 0 \end{bmatrix}.$$

Then $\mathcal{A}$ is Hermitian and it holds that

$$\mathcal{A}\begin{bmatrix} x_i \\ 0 \end{bmatrix} = (\lambda_i - \sigma)\begin{bmatrix} x_i \\ 0 \end{bmatrix}, \; i \neq j,$$
$$\mathcal{A}\begin{bmatrix} x_j \\ 1 \end{bmatrix} = \begin{bmatrix} x_j \\ 1 \end{bmatrix}.$$

The eigenvalues of $\mathcal{A}$ are 1, -1, and $\lambda_i - \sigma$, $i \neq j$.

Note that the *subspace iteration method* (SIM, [Bat96, §11.6]) is subspace iteration with $A = K^{-1}M$ and the Rayleigh-Ritz procedure for GEPs (see Algorithm 6).

Input: $K, M \in \mathbb{C}^{n,n}$ Hermitian, $S \in \mathbb{C}^{n,s}$ with full column rank
Output: Approximate eigenpairs $(\widetilde{\lambda}_i, \widetilde{x}_i)$ of (K, M), $\widetilde{x}_i \in \operatorname{ran} S$, $i = 1, 2, \ldots, s$
function RAYLEIGH-RITZ(K, M, S)
Compute a thin QR decomposition: $QR \leftarrow S$
$K_Q \leftarrow Q^*KQ$
$M_Q \leftarrow Q^*MQ$

Solve $K_Q X_Q = M_Q X_Q \Lambda_Q$
Lift eigenvectors: $\widetilde{X} \leftarrow QX_Q$

return Λ_Q, $\widetilde{X}$
end function

Algorithm 6: Rayleigh-Ritz procedure for regular GEPs

4.2 Improving Numerical Stability

In this section, we shed light on how identical scaling of the matrices in a matrix pair and diagonal scalings can improve the numerical stability by reducing condition numbers of eigenvalues.

Theorem 4.5 ([Ste01, Theorem 4.12])**.** *Let (λ, x) be an eigenpair of the Hermitian matrix pencil (K, M), where λ is a simple eigenvalue. Then the condition number of λ is*

$$\frac{\|x\|_2^2}{\sqrt{|x^*Kx|^2 + |x^*Mx|^2}}.$$

Similar to the condition number of the matrix R in Theorem 3.3, the condition number of the eigenvalue can be large if the norms of the matrices K and M differ in magnitude. We conclude, similar norms of the matrices in a pencil never worsen the condition numbers of the eigenvalues.

Next, we try to decide on a strategy for balancing the matrix pencil by means of a diagonal scaling (DKD, DMD). We can determine D either by analyzing both matrices simultaneously or by analyzing only one of the matrices. A motivation for the latter strategy is given by the next theorem which shows that as soon as one of the matrices in the pencil is well-conditioned, all eigenvectors will be close to orthogonal.

Theorem 4.6 ([Nak12, §3])**.** *Let λ be an eigenvalue of the Hermitian matrix pencil (K, M) with multiplicity m, where M is Hermitian positive definite. Let X be the matrix of m eigenvectors corresponding to the eigenvalue λ, let $X^*MX = I$. Let $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_m > 0$ be the singular values of X^*X. Then the condition numbers of λ under Hermitian definite perturbations ΔK, ΔM are*

$$(1 + |\lambda|)\sigma_i,\ i = 1, 2, \ldots, m,$$

where $\|\Delta K\|_2, \|\Delta M\|_2 \leq 1$. It holds $\sigma_1/\sigma_m \leq \kappa_2(M)$.

The Jacobi scaling D of an Hermitian matrix $A = [a_{ij}]$ is chosen so that the ith diagonal element of DAD is one whenever $a_{ii} \neq 0$. Experiments with real SPD matrices show that Jacobi scaling

never enlarges the condition number and the reduction of the condition number is comparable to an iterative algorithm [BM12, §2]. Considering that mass matrices in finite element analysis are real SPSD and often diagonal, we can acquire modified GEPs with orthogonal eigenvectors.

In [War81], a method for a diagonal scaling for (non-Hermitian) matrix pencils is proposed so that the modulus of all entries are of the same magnitude (this is the balancing algorithm for GEPs in the LAPACK function xGGBAL). Unfortunately, this method may worsen the accuracy of the computed eigenvalues [LvD06, §6]. Another approach is to use diagonal matrices D_ℓ, D_r in order to equilibrate the Euclidean norms of every row and every column to one and in the experiments in [LvD06], this method never reduces the accuracy of the computed eigenvalues and greatly improves it for diagonalizable matrix pencils. For Hermitian matrix pencils, $D \coloneqq D_\ell = D_r$ and D can be computed directly. Note that this scaling may have adverse effects if it is used in conjunction with GSVD-based GEP solvers that scale one of the matrices in the pencil such that $\|K\| \approx \|M\|$.

In finite precision arithmetic, the entries of D should be rounded to the nearest power of 2 (to the nearest power of the base of the floating point arithmetic) in order to avoid round-off errors. Moreover, the balancing methods should ignore very small diagonal entries (Jacobi scaling) or rows small in norm (for the balancing in [LvD06]), respectively, e. g., with Jacobi scaling, the diagonal entries d_i of D could be chosen as

$$d_i \coloneqq \begin{cases} \textsc{nearest-power-of-2}\left(\sqrt{\dfrac{\max_j a_{jj}}{a_{ii}}}\right) & \text{if } a_{ii} > n\varepsilon \max_j a_{jj} \\ 1, & \text{otherwise,} \end{cases}$$

where ε is the machine epsilon.

4.3 Automated Multilevel Substructuring

Given $\lambda_c > 0$ and a GEP with HPD matrices, the automated multilevel substructuring method (AMLS, [Kap01; Gao+08; BL04]) computes approximations to all eigenpairs (λ, x), where $\lambda \leq \lambda_c$. It is well suited for problems with matrices arising from finite element analysis in structural mechanics and low accuracy demands. In this case, AMLS delivers results considerably faster than shift-and-invert Lanczos (SIL) [Kap01, §7] [Gao+08, §4]. AMLS is based on component mode synthesis (CMS, [CB68]).

Initially, AMLS orders mass and stiffness matrix to give them a certain block structure and this block structure is retained by all transformations applied by AMLS to the matrix pencil. Let $S \in \mathbb{C}^{n,m}$ have full column rank. Throughout this section, A^S_{ij} denotes the i, j block of a matrix A after a congruence transformation involving S. Note that there is a change of basis whenever a congruence transformation is executed because there is a matrix multiplied from the right-hand side to A: $Ax = AS S^\dagger x$, where $S^\dagger$ is a generalized inverse of S. In the style of the naming convention for matrices, it holds that $x = Sx^S$.

4.3.1 Nested Dissection

Nested dissection (ND, [Geo73; LRT79] [MC, §11.7]) is a fill-in reducing matrix ordering. In Figure 4.1, a matrix A with one level of substructuring is shown. The blocks $A_{1,1}$ and $A_{2,2}$ are called *substructure blocks*; the block $A_{3,3}$ is called *coupling block*. For minimal fill-in and better processing speed, $A_{1,1}$ and $A_{2,2}$ should be comparable in dimension while $A_{3,3}$ is small. $A_{3,3}$ can be empty and in this case, A is block diagonal. ND can be applied recursively to the substructure

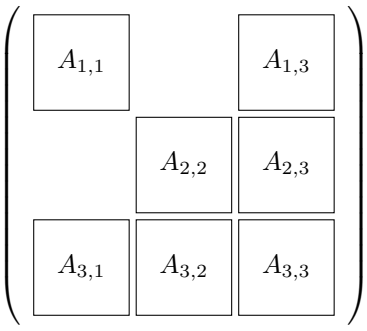


Figure 4.1: A real symmetric matrix A with one level of nested dissection ordering. The blocks $A_{1,1}$, and $A_{2,2}$ are substructure blocks; the block $A_{3,3}$ is the coupling block.

blocks and an example can be found in Figure 4.2. This figure shows a ND ordering with two levels with substructure blocks $A_{1,1}$, $A_{2,2}$, $A_{4,4}$, $A_{5,5}$, and $A_{6,6}$; the blocks $A_{3,3}$, $A_{7,7}$, and $A_{8,8}$ are coupling blocks. The first level of substructuring consists of the blocks $A_{1:3,1:3}$, $A_{4:7,4:7}$, and $A_{8,8}$.

From the point of view of graph theory, a nested dissection ordering is a minimal vertex cut. Let $G = (V, E)$ be the unweighted graph induced by the matrix at hand, let $V_1, V_2, S \subseteq V$ be disjoint, $V = V_1 \cup V_2 \cup S$, and let $1/2 \leq \alpha < 1$. Then the minimal vertex cut problem means finding V_1, V_2, and S such that $|S|$ is minimal, $|V_1|, |V_2| \leq \alpha n$, and no vertex in V_1 is adjacent to a vertex in V_2 [LRT79, pp. 347 sq.].

4.3.2 Algorithm

Pseudocode for the AMLS method can be found in Algorithm 7. Initially, AMLS computes a nested dissection ordering of the graph induced by mass and stiffness matrix. Let $\Pi \in \mathbb{C}^{n,n}$ be the permutation matrix corresponding to the ND ordering of the induced graph. With the naming convention above, we have

$$K^\Pi = \Pi^* K \Pi, M^\Pi = \Pi^* M \Pi$$

giving the transformed matrix pencil $K^\Pi x^\Pi = \lambda M^\Pi x^\Pi$. Throughout this section, let ℓ denote the number of blocks on the diagonal and let n_i be the dimension of the blocks K^Π_{ii}, M^Π_{ii}.

The second step in AMLS is the computation of a block LDL^T decomposition of the stiffness matrix such that $LDL^* := K^\Pi$. The factorization is used to block diagonalize K giving the GEP $K^L x^L = \lambda M^L x^L$, where $K^L = L^{-1} K^\Pi L^{-*} = D$ and $M^L = L^{-1} M^\Pi L^{-*}$. Note that L is a lower unit triangular matrix possessing the same block structure as K^Π and M^Π (unit triangular meaning with ones one the diagonal). Moreover, K is positive definite so the LDL^T decomposition exists without pivoting.

During the third step, AMLS solves all block diagonal GEPs

$$K^L_{jj} x^j_k = \lambda^j_k M^L_{jj} x^j_k,\ j = 1, 2, \ldots, \ell,\ k = 1, 2, \ldots, n_j.$$

For each block diagonal GEP, we introduce matrices

$$\Lambda^L_j = \operatorname{diag}(\lambda^j_1, \lambda^j_2, \ldots, \lambda^j_{n_j}) \in \mathbb{C}^{n_j, n_j},$$
$$X^L_j = [x^j_1, x^j_2, \ldots, x^j_{n_j}] \in \mathbb{C}^{n_j, n_j}$$

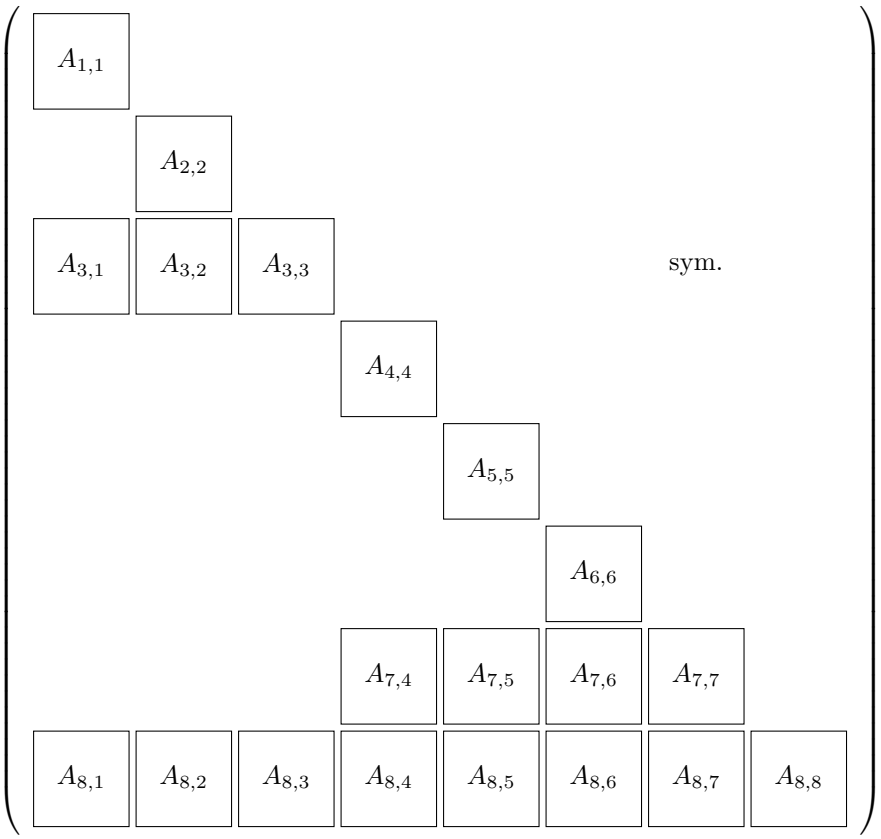


Figure 4.2: A real symmetric matrix A with two levels of nested dissection ordering.

such that $K_{jj}^L X_j^L = M_{jj}^L X_j^L \Lambda_j^L$. Finally, let

$$\widetilde{\Lambda}^L = \mathrm{diag}(\Lambda_1^L, \Lambda_2^L, \ldots, \Lambda_\ell^L),$$
$$\widetilde{X}^L = \mathrm{diag}(X_1^L, X_2^L, \ldots, X_\ell^L).$$

Note that we just lifted the approximate eigenvectors x_k^j to $\mathbb{C}^n$ so $\widetilde{\Lambda}^L$ and $\widetilde{X}^L$ contain approximations to the eigenpairs of the matrix pencil (K^L, M^L).

As the fourth step, AMLS performs *modal truncation*. Let $(\widetilde{\lambda}_i^L, \widetilde{x}_i^L)$, $i = 1, 2, \ldots, n$, be the approximate eigenpairs of (K^L, M^L) calculated in the previous step, i. e., $\widetilde{x}_i^L$ is the ith column of $\widetilde{X}^L$. Given λ_c and a tolerance $c_s \geq 1$, AMLS retains all eigenvectors $\widetilde{x}_i^L$ where the corresponding eigenvalue is less or equal to $c_s\lambda_c$. Specifically, let $\mathcal{I} = \{i = 1, 2, \ldots, n \mid \widetilde{\lambda}_i^L \leq c_s\lambda_c\}$. Then AMLS constructs a matrix S with vectors $\widetilde{x}_i^L$, $i \in \mathcal{I}$, as columns.

Next, the method executes the Rayleigh-Ritz procedure on the subspace $\mathrm{ran}\, S$ giving eigenpairs (λ^S, x^S) of (K^S, M^S). Finally, the method reverses the basis changes in order to acquire approximate eigenpairs $(\widetilde{\lambda}, \widetilde{x})$ of (K, M). It holds $\widetilde{\lambda} = \lambda^S$, $\widetilde{x} = \Pi L^{-*} S x^S$.

Input: $K, M \in \mathbb{C}^{n,n}$ HPD, $\lambda_c > 0$, $c_s \geq 1$
Output: Approximate eigenpairs $(\widetilde{\lambda}, \widetilde{x})$, where $\widetilde{\lambda} \leq \lambda_c$
function AMLS(K, M, λ_c, c_s)
Compute nested dissection ordering Π with ℓ diagonal blocks
$K^\Pi \leftarrow \Pi^* K \Pi$
$M^\Pi \leftarrow \Pi^* M \Pi$

Compute a block LDL^T decomposition: $LDL^* \leftarrow K$
$K^L \leftarrow L^{-1} K^\Pi L^{-*}$
$M^L \leftarrow L^{-1} M^\Pi L^{-*}$

for all $j = 1, 2, \ldots, \ell$ **do**
Solve $K_{jj}^L X_j^L = M_{jj}^L X_j^L \Lambda_j^L$
end for

$\widetilde{\Lambda}^L \leftarrow \mathrm{diag}(\Lambda_1^L, \Lambda_2^L, \ldots, \Lambda_\ell^L)$
$\widetilde{X}^L \leftarrow \mathrm{diag}(X_1^L, X_2^L, \ldots, X_\ell^L)$

Execute modal truncation, get matrix S
Execute Rayleigh-Ritz procedure: $\widetilde{\Lambda}, \widetilde{X}^L \leftarrow$ RAYLEIGH-RITZ(K, M, S)

return $\widetilde{\Lambda}$, $\Pi L^{-*} \widetilde{X}^L$
end function

Algorithm 7: Pseudocode for the automated multilevel substructuring method (AMLS)

4.3.3 Remarks

Recall that a nested dissection ordering of a real symmetric matrix A corresponds to the a minimal vertex separator in the unweighted graph induced by A. AMLS operates on pairs of matrices so a vertex separator for one of the matrices may not be a vertex separator for

the other matrix and vice versa. For matrix pairs originating from conforming finite element formulations with first-order polynomial ansatz functions, the unweighted graphs induced by mass and stiffness matrix are identical and we may use either matrix to calculate a minimal vertex separator. For general matrix pairs, a proper ordering can be ensured by calculating a nested dissection ordering of the matrix $|K| + |M|$.

The LDL^T decomposition of a singular stiffness matrix is still possible if all singular diagonal blocks are permuted to the lower right. This operation increases fill-in and enlarges the size of the final diagonal block.

In the description above, the modal truncation precedes the projection on the subspace $\operatorname{ran} S$ but these two operations are interchangeable. Let $s = |\mathcal{I}|$, let $P \in \mathbb{C}^{n,s}$ have the vectors e_i, $i \in \mathcal{I}$, as columns. Then $S = \widetilde{X}^L P$ and furthermore $S^* A S = P^* (\widetilde{X}^L)^* A \widetilde{X}^L P$. Clearly, $(\widetilde{X}^L)^* A \widetilde{X}^L$ can be calculated as soon as AMLS solved the GEPs on the block diagonal in step three. Now if every vector $\widetilde{x}_i^L$ is M-normal, i. e., $(\widetilde{x}_i^L)^* M \widetilde{x}_i^L = 1$, then the mass matrix $(\widetilde{X}^L)^* M^L \widetilde{X}^L$ has identity matrices on its block diagonal and $(\widetilde{X}^L)^* K^L \widetilde{X}^L = \widetilde{\Lambda}^L$.

AMLS is not a geometric domain decomposition method [Smi97]. If the matrices originate from a finite element discretization, then the nested dissection ordering does correspond to a partition in the domain underlying the continuous problem but it is only after the LDL^T decomposition that substructures are examined. Let $\phi_i^h \in V_h \subset H_0^1(\Omega)$, $i = 1, 2, \ldots, n$, be the finite element ansatz functions, cf. Section 2.2. Then the entries of the mass and stiffness matrix are the inner products of the ansatz functions:

$$K = [a(\phi_i^h, \phi_j^h)]_{i,j=1}^n,$$
$$M = [(\phi_i^h, \phi_j^h)]_{i,j=1}^n.$$

With the proper ansatz functions, the matrices K^L, M^L can be generated directly by the finite element method. Accordingly, recall the congruence transformations applied to the original matrices:

$$K^L = L^{-1} \Pi^* K \Pi L^{-*},$$
$$M^L = L^{-1} \Pi^* M \Pi L^{-*}.$$

Then K^L and M^L are generated by the finite element method if the ansatz functions

$$\varphi_i^h \coloneqq \sum_{j=1}^n (L^{-1})_{ij} \phi_{\pi(j)}^h,\ i = 1, 2, \ldots, n,$$

are used. L^{-1} is block lower triangular but nevertheless, some of the modified ansatz functions φ_i^h may span the whole domain. Consequently, it would be more apt to consider AMLS as an algebraic multigrid method [McC94, §4].

Modal truncation is a common way to perform modal reduction and thus sometimes called *standard modal reduction* in contrast to the more elaborate *optimal modal reduction* [GBP04].

4.3.4 Exact Eigenpairs

The AMLS method does not provide mechanisms to directly control the approximation properties of the computed eigenpairs. Strictly speaking, AMLS is not an eigensolver. Nevertheless, AMLS can be used to quickly generate starting subspaces for one of the iterative methods in Section 4.1, e. g., the subspace iteration method [Bat96, §11.6] which has proven to be a well-grounded choice [YVC13] [CMM16, §6].

4.4 Eigenvalues and GEPs with Block Matrices

In this section we deal with matrix pencils (K, M), where K and M possess the same block structure. Given an eigenpair of a GEP on the block diagonal of (K, M), we will analyze how well this eigenpair approximates an exact eigenvalue of (K, M). To that end, we will calculate perturbation bounds for these approximate eigenvalues, once without utilizing eigenvectors and a second time with eigenvectors.

Throughout this section, mass and stiffness matrix are 2×2 block matrices with identical partitions, $(\widetilde{\lambda}, \widetilde{x})$ is an approximate eigenpair of (K, M), and $r = K\widetilde{x} - \widetilde{\lambda}M\widetilde{x}$ is the residual. Both $\widetilde{x}$ and r are partitioned conformally to K and M. Hence

$$K = \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix}, M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}, \widetilde{x} = \begin{bmatrix} \widetilde{x}_1 \\ \widetilde{x}_2 \end{bmatrix}, r = \begin{bmatrix} r_1 \\ r_2 \end{bmatrix}.$$

Since we want to examine how well the eigenvalues of a GEP on the block diagonal approximate an exact eigenvalue λ of (K, M), we will assume that $(\widetilde{\lambda}, \widetilde{x}_1)$ is an exact eigenpair of (K_{11}, M_{11}) so $\widetilde{x}_2 = 0$ and

$$r = \begin{bmatrix} 0 \\ K_{21}\widetilde{x}_1 - \widetilde{\lambda}M_{21}\widetilde{x}_1 \end{bmatrix}.$$

4.4.1 Eigenvalue Perturbation Bounds without Eigenvectors

The following theorem is shown for the sake of completeness and allows us to bound the perturbation of an eigenvalue from a GEP on the block diagonal.

Theorem 4.7 ([Li+11, Corollary 2.10]). *Let $K, M \in \mathbb{C}^{n,n}$ be 2×2 block matrices, where K is Hermitian and M is Hermitian positive definite. Let $\widetilde{\lambda}_1 \leq \widetilde{\lambda}_2 \leq \cdots \leq \widetilde{\lambda}_m$ be the eigenvalues of (K_{11}, M_{11}). Let*

$$\begin{aligned} A &= M_{11}^{-1/2} K_{11} M_{11}^{-1/2}, \\ E &= M_{11}^{-1/2} K_{12} M_{22}^{-1/2}, \\ F &= M_{11}^{-1/2} M_{12} M_{22}^{-1/2}. \end{aligned}$$

Then there are m eigenvalues $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_m$ of (K, M) such that

$$|\widetilde{\lambda}_i - \lambda_i| \leq \frac{\|E - AF\|_2}{\sqrt{1 - \|F\|_2^2}}, \; i = 1, 2, \ldots, m.$$

The next theorem requires all eigenvalues of all GEPs on the block diagonal and delivers sharper and more intuitive bounds.

Theorem 4.8 ([Li+11, Corollary 2.9]). *Let $K, M \in \mathbb{C}^{n,n}$ be 2×2 block matrices, where K is Hermitian and M is Hermitian positive definite. Let $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ be the eigenvalues of (K, M), let $\widetilde{\lambda}_1 \leq \widetilde{\lambda}_2 \leq \cdots \leq \widetilde{\lambda}_n$ be the union of eigenvalues of (K_{11}, M_{11}) and (K_{22}, M_{22}). Let*

$$\begin{aligned} E &= M_{11}^{-1/2} K_{12} M_{22}^{-1/2}, \\ F &= M_{11}^{-1/2} M_{12} M_{22}^{-1/2}. \end{aligned}$$

Then

$$|\widetilde{\lambda}_i - \lambda_i| \leq \|E - \widetilde{\lambda}_i F\|_2, \; i = 1, 2, \ldots, n.$$

Note that if the matrix of eigenvectors X_1 of (K_{11}, M_{11}) and the matrix of eigenvectors X_2 of (K_{22}, M_{22}) are available and if $X_1^* M_{11} X_1 = I$, $X_2^* M_{22} X_2 = I$, then E and F in Theorem 4.8 can be replaced with

$$E = X_1^* K_{12} X_2,$$
$$F = X_1^* M_{12} X_2.$$

Consider the scenario where only one of the matrices is perturbed on the off-diagonal. If $K_{12} = 0$, $M_{12} \neq 0$, then

$$|\widetilde{\lambda}_i - \lambda_i| \leq |\widetilde{\lambda}_i| \|F\|_2 \Leftrightarrow \frac{|\widetilde{\lambda}_i - \lambda_i|}{|\widetilde{\lambda}_i|} \leq \|F\|_2,$$

i. e., a *relative* error is induced. If $K_{12} \neq 0$, $M_{12} = 0$, then an *absolute* error is induced because

$$|\widetilde{\lambda}_i - \lambda_i| \leq \|E\|_2.$$

In this thesis, we seek the smallest eigenvalues of a matrix pencil with HPSD matrices. Consequently, absolute errors are less desirable than relative errors (if $\|E\|$ and $\|F\|$ are similar).

4.4.2 Eigenvalue Perturbation Bounds with Eigenvectors

Perturbation bounds on an eigenvalue using eigenvalues *and* eigenvectors can be calculated by means of the forward error in Theorem 2.5. The forward error bounds in Theorem 2.5 are based on the Frobenius norm of the mass and the stiffness matrices but the forward error calculation is slightly simpler if we use the spectral norm instead and if r is orthogonal to $\widetilde{x}$, then the bounds are also slightly sharper by a factor $\sqrt{2}$. Moreover, using the spectral norm eases the comparison with the bounds doing without eigenvectors in the previous section.

Theorem 4.9. *Let $(\widetilde{\lambda}, \widetilde{x})$ be an approximate eigenpair of the Hermitian matrix pencil (K, M), where $\widetilde{\lambda}$ is real finite and $\|\widetilde{x}\|_2 = 1$. Let $r = K\widetilde{x} - \widetilde{\lambda} M \widetilde{x}$. Then*

$$\eta_{2,2}(\widetilde{\lambda}, \widetilde{x}) = \frac{\|r\|_2}{\sqrt{\|K\|_2^2 + |\widetilde{\lambda}|^2 \|M\|_2^2}}.$$

Proof. Use [AA11, Theorem 3.10] and [AA11, Eq. (1)] with $\Lambda_m = [\|K\|_2, |\widetilde{\lambda}| \|M\|_2]$. □

Corollary 4.1. *Let $(\widetilde{\lambda}, \widetilde{x})$ be an approximate eigenpair of the Hermitian matrix pencil (K, M), where $\widetilde{\lambda}$ is real infinite and $\|\widetilde{x}\|_2 = 1$. Then*

$$\eta_{2,2}(\widetilde{\lambda}, \widetilde{x}) = \frac{1}{\|M\|_2} \|M\widetilde{x}\|_2.$$

Theorem 4.10. *Let (λ, x) be an eigenpair of the Hermitian matrix pencil (K, M), where λ is simple and finite. Then we can compute the condition number $\kappa_{2,2}(\lambda, x)$ with*

$$\kappa_{2,2}(\lambda, x) = \frac{\|x\|_2^2}{|x^* M x|} \sqrt{\|K\|_2^2 + |\lambda|^2 \|M\|_2^2}.$$

Proof. Use [AAK11, Eq. (10)] in conjunction with the weight vector $\omega_{\text{rel}}(2)$. □

Theorem 4.11. *Let $(\widetilde{\lambda}, \widetilde{x})$ be an approximate eigenpair of an Hermitian matrix pencil (K, M), where $\widetilde{\lambda}$ is a simple, real finite eigenvalue and $\|\widetilde{x}\|_2 = 1$. Let $r = K\widetilde{x} - \widetilde{\lambda}M\widetilde{x}$. Then there is an exact eigenvalue λ of (K, M) such that*

$$|\widetilde{\lambda} - \lambda| \leq \frac{1}{|\widetilde{x}^* M \widetilde{x}|} \|r\|_2.$$

Proof. The error bound is the product of the backward error $\eta_{2,2}(\widetilde{\lambda}, \widetilde{x})$ and the corresponding condition number $\kappa_{2,2}(\widetilde{\lambda}, \widetilde{x})$. □

Judging by the norm of the residual in the forward error bound, the observation that off-diagonal blocks in the stiffness matrix cause an absolute error while the off-diagonal blocks in the mass matrix induce a relative error can be made here, too.

As an immediate improvement over the eigenvector-free perturbation bounds , the forward error can be calculated for all Hermitian mass matrices instead of only Hermitian positive definite mass matrices. Moreover, the forward error can cope with the scenario where (λ, x) is an eigenpair of (K, M) and off-diagonal perturbations are applied to (K, M), i. e., we seek perturbation bounds for λ with respect to the matrix pencil $(\widetilde{K}, \widetilde{M})$, where

$$\widetilde{K} = \begin{bmatrix} K_{11} & K_{12} + E^* \\ K_{21} + E & K_{22} \end{bmatrix}, \widetilde{M} = \begin{bmatrix} M_{11} & M_{12} + F^* \\ M_{21} + F & M_{22} \end{bmatrix}.$$

A disadvantage of the forward error bounds is the fact that they apply only to simple eigenvalues whereas the eigenvector-free bounds are not impaired by a multiple eigenvalue.

4.4.3 Application to AMLS

AMLS computes approximate eigenpairs of a matrix pencil by solving all GEPs on the block diagonal and the results in this section can be applied to compute perturbation bounds for an eigenvalue of a GEP on the block diagonal. Recall that the stiffness matrix is block diagonal when the approximate eigenpairs are computed. Hence $K_{12} = 0$, $K_{21} = 0$, the residual simplifies to

$$r = \begin{bmatrix} 0 \\ -\widetilde{\lambda} M_{21} \widetilde{x}_1 \end{bmatrix},$$

the M-norm of $\widetilde{x}$ is then

$$|\widetilde{x}^* M \widetilde{x}| = |\widetilde{x}_1^* M_{11} \widetilde{x}_1|$$

and substituting these equations into the forward error of Theorem 4.11 yields

$$|\widetilde{\lambda} - \lambda| \leq \frac{1}{|\widetilde{x}_1^* M_{11} \widetilde{x}_1|} \|\widetilde{\lambda} M_{21} \widetilde{x}_1\|_2.$$

AMLS users seek the smallest eigenvalues of a matrix pencil and the stiffness matrix is block diagonal because of a previously computed block LDL^T decomposition. Consequently, there is only a relative error associated with each approximate eigenvalue. We conclude that the block LDL^T decomposition is an important step in the AMLS method to ensure good approximations to the eigenvalues of the matrix pencil (K, M).

During the modal truncation step, AMLS selects all approximate eigenpairs $(\widetilde{\lambda}, \widetilde{x})$ where $\widetilde{\lambda} \leq c_s \lambda_c$. Given c_s, we want to determine the maximum value of $\|M_{21} \widetilde{x}_1\|_2$ such that the perturbation of the exact eigenvalue λ_c is no larger than $c_s \lambda_c$, that is, we miss none of the desired

eigenpairs. This means that if $\widetilde{\lambda} = c_s\lambda_c$ and if $\lambda = \lambda_c$, then the right-hand side of this expression should be no larger than the difference between $c_s\lambda_c$ and λ_c:

$$|\widetilde{\lambda} - \lambda| \leq \frac{1}{|\widetilde{x}_1^* M_{11} \widetilde{x}_1|} \|\widetilde{\lambda} M_{21} \widetilde{x}_1\|_2 \overset{!}{\leq} (c_s - 1)\lambda_c.$$

After proper shifting of the terms and substituting $\widetilde{\lambda} = c_s\lambda_c$, it follows that

$$\|M_{21}\widetilde{x}_1\|_2 \leq |\widetilde{x}_1^* M_{11} \widetilde{x}_1| \frac{c_s - 1}{c_s}.$$

In the AMLS method, this inequality can be used to check if c_s is sufficiently large.

Consider the case $c_s \to \infty$. Then

$$\|M_{21}\widetilde{x}_1\|_2 \leq |\widetilde{x}_1^* M_{11} \widetilde{x}_1|.$$

Furthermore, assume the perturbation bound holds for eigenpairs of (K_{22}, M_{22}), as well, i. e.,

$$\|M_{21}^* \widetilde{x}_2\|_2 \leq |\widetilde{x}_2^* M_{22} \widetilde{x}_2|.$$

With this choice of c_s, the modal truncation step in AMLS will retain all approximate eigenpairs from the GEPs on the block diagonal so that the computed search space for the eigenpairs is all of $\mathbb{C}^n$. Consequently, the AMLS method will return exact eigenpairs. Now observe that there are HPSD matrices violating these two conditions.

Example 4.2 (A HPSD matrix where $\|M_{21}\|_2 \geq \|M_{11}\|_2$)**.** Let $c \geq 0$, let $\delta \in \{0, 1\}$. Then

$$M = \begin{bmatrix} 1 & c \\ c & c^2 + \delta \end{bmatrix}$$

is positive semidefinite ($\delta = 0$) or positive definite ($\delta = 1$), respectively.

In this section, we considered perturbation bounds for a *single* approximate eigenpair and although AMLS is guaranteed to calculate exact eigenpairs for huge values of c_s, the error bounds do not capture this behavior. With this thought, we want to highlight that perturbation bounds for single eigenpairs are intrinsically limited in their predictive abilities, especially if a large number of approximate eigenpairs is used to construct a subspace.

4.4.4 Minimizing Eigenvalue Perturbation

In this section, we discuss ways to permute a matrix pair in order to acquire a pair of identically partitioned 2×2 block matrices where the diagonal blocks are similar in dimension and where the eigenvalue perturbation due to the off-diagonal blocks is minimized.

For now, consider the case of a single HPSD matrix A instead of a matrix pencil. From a linear algebra point of view, a 2×2 block matrix without off-diagonal entries is a block *diagonal* matrix with two clearly distinguishable invariant subspaces. It follows that we have to find approximate subspaces if we want to transform A into a matrix with block structure and the better the approximation, the smaller the off-diagonal entries. In practice, we cannot exploit this observation because do not know invariant subspaces.

Let us consider the problem from the point of view of graph theory: a 2×2 block matrix without off-diagonal entries corresponds to a graph with two partitions such that there is no edge connecting the two partitions. Usually a matrix does not possess such an ordering so instead we can minimize the number of edges connecting the two partitions or the sum of weights of these edges. This is an instance of the minimum bisection problem (or the k-way graph partitioning problem for $k = 2$).

Definition 4.3 (Minimum bisection problem [Bul+15, §2])**.** Given an undirected simple graph $G = (V, E)$ with non-negative edge weights and $|V|$ even, the goal of the *minimum bisection problem* is to find disjoint vertex subsets $V_1, V_2 \subset V$ such that $V = V_1 \cup V_2$ and $|V_1| = |V_2|$ minimizing the cost of edges connecting the subsets V_1 and V_2:

$$z^* \coloneqq \min_{\substack{V_1, V_2 \subset V \\ V = V_1 \dot\cup V_2 \\ |V_1| = |V_2|}} \sum_{\substack{\{i,j\} \in E \\ i \in V_1, j \in V_2}} c(\{i,j\}).$$

This problem is mathematically difficult and hard to approximate [Bul+15, §2.3]. Nevertheless, a variety of heuristics quickly deliver good results in practice. Interestingly, "the most successful heuristic for partitioning large graphs is the *multilevel graph partitioning* approach" [Bul+15, §6] (emphasis mine). In this thesis, we deal with matrices arising from finite element matrices yet we do not assume availability of the mesh that was used to generate these matrices. If the mesh is available but not the matrices, then *mesh partitioning* [Bul+15, §3.1] can be applied. If additionally geometric information is accessible, then *geometric partitioning* [Bul+15, §4.5] can be employed.

We want to relate the objective value minimized by bisection to a norm of the off-diagonal block A_{21}. The graph algorithm minimizes a sum of non-negative edge weights so a corresponding norm must be an absolute vector norm applied to a matrix [HJ12, §5.7], e. g., a vector p-norm. Naturally, we can use the Frobenius norm here (the Euclidean vector norm) which integrates nicely into the backward error of Section 2.1. To that end, let

$$c(\{i,j\}) = |a_{ij}|^2.$$

It follows $\|A_{21}\|_F^2 = z^*$. We would like to note that if every edge has the same non-zero weight, then we are minimizing the number of edges connecting the partitions V_1 and V_2; A_{21} will be as sparse as possible.

Next, consider matrix pencils (K, M). Let $K = [k_{ij}]$, $M = [m_{ij}]$. We will try to determine weights that minimize eigenvalue perturbation of eigenpairs $(\widetilde{\lambda}, \widetilde{x})$ of (K_{11}, M_{11}). It holds that

$$|\widetilde{\lambda} - \lambda| \leq \frac{1}{|\widetilde{x}_1^* M_{11} \widetilde{x}_1|} \|K_{21}\widetilde{x}_1 - \widetilde{\lambda} M_{21}\widetilde{x}_1\|_2.$$

Strictly speaking, we have to minimize the right-hand size for all vectors $\widetilde{x}_1$ in the desired subspace over all possible matrix permutations–this is already an intractable problem for small n. Hence we will gradually simplify the expression and as a start, let us assume the M-norm of $\widetilde{x}$ is constant:

$$|\widetilde{\lambda} - \lambda| \leq c\|K_{21}\widetilde{x}_1 - \widetilde{\lambda} M_{21}\widetilde{x}_1\|_2,\ c \in \mathbb{R}^+.$$

Furthermore, we assume $K_{21}\widetilde{x}_1 \perp M_{21}\widetilde{x}_1$ so

$$\|K_{21}\widetilde{x}_1 - \widetilde{\lambda} M_{21}\widetilde{x}_1\|_2^2 = \|K_{21}\widetilde{x}_1\|_2^2 + \|\widetilde{\lambda} M_{21}\widetilde{x}_1\|_2^2.$$

Moreover, we consider the Frobenius norm of the off-diagonal blocks:

$$\|K_{21}\widetilde{x}_1 - \widetilde{\lambda} M_{21}\widetilde{x}_1\|_2^2 \leq \|K_{21}\|_F^2 + \|\widetilde{\lambda} M_{21}\|_F^2.$$

We still have to pick a value for $\widetilde{\lambda}$. In our opinion, any choice $0 \leq \widetilde{\lambda} \leq \lambda_c$ is sensible, e. g., $\widetilde{\lambda} = 0$ consistently minimizes the absolute error (unless K is diagonal). Thus, we introduce a user-provided parameter λ_w ("w" for "weight") yielding the following expression to be optimized:

$$\|K_{21}\|_F^2 + \lambda_w^2 \|M_{21}\|_F^2.$$

We can minimize this expression by solving the minimum bisection problem on the graph $G = (V, E)$ with nodes $V = \{1, 2, \ldots, n\}$, edges $E = \{\{i, j\} : |k_{ij}| + |m_{ij}| \neq 0\}$, and edge weights

$$c(\{i, j\}) = |k_{ij}|^2 + \lambda_w^2 |m_{ij}|^2.$$

We want to mention that matrix pairs arising conforming FE formulations possess the same pattern of non-zero entries whereas with mass lumping, the mass matrix is diagonal.

4.4.5 Backward Error Bounds

If $K_{21} = 0$, $M_{21} = 0$, then every exact eigenpair of (K_{ii}, M_{ii}) will be an eigenpair of (K, M) after lifting the eigenvectors and in this case, it makes sense to compute the eigenpairs of (K_{ii}, M_{ii}), $i = 1, 2$, to full possible accuracy. On the other hand, if the residual of the approximate eigenpair will be large, e. g., if $\|K_{11}^{-1/2} K_{12} K_{22}^{-1/2}\|_2 \approx 1$ and $\|K\| \gg \|M\|$, then we can save some effort by not computing $(\widetilde{\lambda}, \widetilde{x}_1)$ to full accuracy. In this section, we analyze the effect of off-diagonal blocks on the backward error of $(\widetilde{\lambda}, \widetilde{x})$.

Recall that

$$\eta_{F,2}^H(\widetilde{\lambda}, \widetilde{x}) = \sqrt{\frac{2\|r\|_2^2 - |r^* \widetilde{x}|^2}{\|K\|_F^2 + |\widetilde{\lambda}|^2 \|M\|_F^2}}$$

and

$$r = \begin{bmatrix} 0 \\ K_{21}\widetilde{x}_1 - \widetilde{\lambda} M_{21} \widetilde{x}_1 \end{bmatrix}.$$

Since $r^* \widetilde{x} = 0$ and since $\|K\|_F$, $\|M\|_F$ are known, we have to determine $K_{21}\widetilde{x}_1$ and $M_{21}\widetilde{x}_1$. For every matrix A, it holds that $\|A\|_2 \leq \|A\|_F$. Consequently,

$$\begin{aligned} 0 \leq &\|K_{21}\widetilde{x}_1\|_2 \leq \|K_{21}\|_F, \\ 0 \leq &\|M_{21}\widetilde{x}_1\|_2 \leq \|M_{21}\|_F. \end{aligned}$$

This is all we can prove without additional assumptions. Thus, we assume $K_{21}\widetilde{x}_1 \perp M_{21}\widetilde{x}_1$ and that the off-diagonal blocks have a uniform singular value distribution, i. e., $\sigma_i = {}^1\!/_p (p - i + 1)\sigma_1$, $i = 1, 2, \ldots, p$, where p is the minimum of the number of rows and the number of columns. For a matrix A with these singular values, it follows that

$$\|A\|_F^2 = \sum_{i=1}^{p} \sigma_i^2 = \sum_{i=1}^{p} \sigma_1^2 \frac{1}{p^2} i^2 = \sigma_1^2 \frac{1}{p^2} \frac{p(p+1)(2p+1)}{6} \approx \sigma_1^2 \frac{p}{3}.$$

Thus,

$$\sigma_1 \approx \sqrt{\frac{3}{p}} \|A\|_F.$$

The expected value of $\|Av\|_2$ with a uniform singular value distribution, $\|v\|_2 = 1$, is then

$$\mathbb{E}[\|Av\|_2] \approx {}^1\!/_2 \, \sigma_1.$$

Therefore,

$$\mathbb{E}\left[\|r\|_2^2\right] = \|K_{21}\widetilde{x}_1\|_2^2 + \|\widetilde{\lambda} M_{21}\widetilde{x}_1\|_2^2 \approx \frac{1}{4}\left(\sigma_1(K_{21})^2 + \sigma_1(\widetilde{\lambda} M_{21})^2\right) = \frac{1}{4}\frac{3}{p}\left(\|K_{21}\|_F^2 + \|\widetilde{\lambda} M_{21}\|_F^2\right).$$

In summary, under the assumptions that the off-diagonal blocks have a uniform singular value distribution and that $K_{21}\widetilde{x}_1$ is always orthogonal to $M_{21}\widetilde{x}_1$, the following expression gives the expected backward error for an approximate eigenpair:

$$\mathbb{E}\left[\eta_{F,2}^H(\widetilde{\lambda},\widetilde{x})\right] = \sqrt{\frac{2\mathbb{E}[\|r\|_2^2]}{\|K\|_F^2+|\widetilde{\lambda}|^2\|M\|_F^2}} = \sqrt{\frac{1}{p}\frac{3}{2}\frac{\|K_{21}\|_F^2+|\widetilde{\lambda}|^2\|M_{21}\|_F^2}{\|K\|_F^2+|\widetilde{\lambda}|^2\|M\|_F^2}}.$$

So far, we assumed $(\widetilde{\lambda},\widetilde{x}_1)$ is an exact eigenpair of (K_{11},M_{11}) and now we analyze the case when this is not true, i. e., $\widetilde{x}_2 = 0$ still holds but $K_{11}\widetilde{x}_1 - \widetilde{\lambda}M_{11}\widetilde{x}_1 \neq 0$.

Theorem 4.12. *Let $(\widetilde{\lambda},\widetilde{x})$ be an approximate eigenpair of (K,M), where $\|\widetilde{x}\|_2 = 1$ and $\widetilde{x}_2 = 0$. Let $\eta_{\max}$ be a positive constant such that*

$$\sqrt{\frac{2\mathbb{E}[\|r\|_2^2]}{\|K\|_F^2+|\widetilde{\lambda}|^2\|M\|_F^2}} \leq \eta_{\max}.$$

If

$$\eta_{F,2}^H(\widetilde{\lambda},\widetilde{x}_1) \leq \eta_{\max}$$

with respect to the matrix pencil (K_{11},M_{11}), then

$$\mathbb{E}\left[\eta_{F,2}^H(\widetilde{\lambda},\widetilde{x})\right] \leq \sqrt{2}\eta_{\max}.$$

Proof. It holds that

$$\eta_{F,2}^H(\widetilde{\lambda},\widetilde{x}) = \sqrt{\frac{2\|r\|_2^2-|r^*\widetilde{x}|^2}{\|K\|_F^2+|\widetilde{\lambda}|^2\|M\|_F^2}} = \sqrt{\frac{2\|r_1\|_2^2+2\|r_2\|_2^2-|r_1^*\widetilde{x}_1|^2}{\|K\|_F^2+|\widetilde{\lambda}|^2\|M\|_F^2}}.$$

Observe that

$$\sqrt{\frac{2\|r_1\|_2^2-|r_1^*\widetilde{x}_1|^2}{\|K\|_F^2+|\widetilde{\lambda}|^2\|M\|_F^2}} \leq \sqrt{\frac{2\|r_1\|_2^2-|r_1^*\widetilde{x}_1|^2}{\|K_{11}\|_F^2+|\widetilde{\lambda}|^2\|M_{11}\|_F^2}} = \eta_{F,2}^H(\widetilde{\lambda},\widetilde{x}_1) \leq \eta_{\max}.$$

By assumption,

$$\sqrt{\frac{2\mathbb{E}[\|r_2\|_2^2]}{\|K\|_F^2+|\widetilde{\lambda}|^2\|M\|_F^2}} \leq \eta_{\max}.$$

Substituting the upper bounds into the backward error completes the proof. □

If $(\widetilde{\lambda},\widetilde{x}_1)$ is not an exact eigenpair of (K_{11},M_{11}), then this does not cause a large backward error of $(\widetilde{\lambda},\widetilde{x})$ as long as $\eta_{F,2}^H(\widetilde{\lambda},\widetilde{x}) \approx \eta_{F,2}^H(\widetilde{\lambda},\widetilde{x})$. The applicability of this insight hinges on the difficulty of determining the constant $\eta_{\max}$. In consideration of the fact that we seek all eigenpairs (λ,x) with eigenvalues $0 \leq \lambda \leq \lambda_c$, we can maximize

$$\frac{\|K_{21}\|_F^2+|\widetilde{\lambda}|^2\|M_{21}\|_F^2}{\|K\|_F^2+|\widetilde{\lambda}|^2\|M\|_F^2}$$

subject to $0 \leq \widetilde{\lambda} \leq \lambda_c$. Treating this expression as as a rational function quickly delivers the solutions:

$$\eta_{\max} = \begin{cases} \sqrt{\frac{1}{p}\frac{3}{2}}\frac{\|K_{21}\|_F}{\|K\|_F} & \text{if } \|M_{21}\| = 0 \text{ or } \frac{\|K_{21}\|_F}{\|M_{21}\|_F} \geq \frac{\|K\|_F}{\|M\|_F}, \\ \sqrt{\frac{1}{p}\frac{3}{2}}\frac{\|K_{21}\|_F^2 + \lambda_c^2\|M_{21}\|_F^2}{\|K\|_F^2 + \lambda_c^2\|M\|_F^2} & \text{otherwise.} \end{cases}$$

4.5 A Multilevel Eigensolver

AMLS is rightfully called the "automated *multilevel substructuring*" method. It is also rightfully not called the "automated *multilevel eigensolver*" method because it does not utilize the substructuring in any way other than acquiring many small GEPs on the block diagonal of the transformed mass and stiffness matrices. Additionally, there are no feedback mechanisms and no control systems to improve the generated search space or to keep its dimension in check. In this section, we will investigate a multilevel eigensolver method without these weaknesses.

4.5.1 Developing AMLS Further

When designing the eigensolver, we make the following assumptions:

- The user seeks eigenpairs (in contrast to eigenvalues),
- mass and stiffness matrix are given explicitly (in contrast to matrix-free methods),
- mass and stiffness matrix are HPSD, and
- the matrix pencil is regular.

The latter assumption is needed if shift-and-invert is employed. At least ten years have passed since the inception of AMLS and in the meantime, the following capabilities and insights were gained:

- Numerically stable solution of GEPs with HPSD matrices (Chapter 3),
- quickly computable structured backward error bounds for eigenpairs (Section 2.1),
- quickly computable forward error bounds for eigenvalues (Section 2.1),
- improving numerical stability (Section 4.2),
- the condition numbers of a multiple eigenvalue of an Hermitian GEP [Nak12],
- convergence issues caused by (hard) locking [Sta05].

Let us now gather some ideas for the new solver. Since we want to design a multilevel solver, we can apply the divide-and-conquer paradigm [Cor+09, §2.3.1]. Furthermore, a multilevel eigensolver will generate many intermediate, approximate results so it might be appropriate to use single precision even if the problem is given in double precision (cf. [Lan+06]). The numerically stable GEP solvers allow us to treat the matrix pencils (K, M) and (M, K) as equals, e. g., we can compute the largest and the smallest eigenvalues of a matrix pencil with the same

code by swapping matrices. The backward error allows us to perform controlled modifications of the problem at hand and the forward error can be handy when selecting shifts.

Let us now step back from AMLS. We might consider using other matrix decompositions than the block LDL^T decomposition or no factorization at all. Similarly, we should consider other graph partitionings than vertex separators (nested dissection). Moreover, without a decomposition step we are free to use any graph partitioning algorithm instead of only fill-in reducing orderings. We could examine off-diagonal blocks. In order to improve approximate eigenvalues, we could start to use shift-and-invert or even polynomial acceleration [Saa11, §5.3.3]). From a practical point of view, we have a vested interest in removing the LDL^T decomposition because this decomposition creates much fill-in (see also [HL07, p. 7]). Additionally, the mass matrix is often diagonal and without a decomposition step, we can use a diagonal scaling to turn the mass matrix into the identity matrix, acquiring orthogonal eigenvectors in the process.

4.5.2 Description

We propose the following eigensolver for GEPs with HPSD matrices finding all eigenpairs (λ, x) where $\lambda \leq \lambda_c$, $\lambda_c > 0$. As dense eigensolvers, we use one of the solvers from Chapter 3. To subdivide the initial problem, we use the method outlined in Section 4.4.4 and we apply divide-and-conquer on the two partitions and start a recursion until the diagonal blocks are sufficiently small to be treated as dense problems. After we computed solutions to the GEPs on the block diagonal, we have approximate eigenpairs which can be improved with the methods of Section 4.1.

Pseudocode for the proposed method can be found in Algorithm 8. If the matrices are small enough, we solve the GEP directly with a dense solver and return. Otherwise, we partition the matrix as described in Section 4.4.4. Obviously, it makes no sense to find highly accurate solutions of the GEPs $(K_{ii}^{\Pi}, M_{ii}^{\Pi})$, $i = 1, 2$, if $\|K_{21}^{\Pi}\|$ or $\|M_{21}\|$ are large in comparison $\|K_{ii}\|$ and $\|M_{ii}\|$, $i = 1, 2$, respectively. Hence we calculate the maximum expected backward error $\eta_{\max,0}$ based on the analysis in Section 4.4.5 and set

$$\eta'_{\max} := \max(\eta_{\max}, \eta_{\max,0}),$$

where $\eta'_{\max}$ is the maximum backward error for the subproblems. In the subsequent combine phase of Algorithm 8, we need to ensure that we find all eigenpairs (λ, x), where $\lambda \leq \lambda_c$, we need prescribe accuracy requirements, and we need to define how we improve approximate eigenpairs. We can guarantee to find all desired eigenpairs by calculating and analyzing the forward error, selecting every approximate eigenpair $(\widetilde{\lambda}, \widetilde{x})$, where

$$\widetilde{\lambda} - |\widetilde{\lambda} - \lambda| \leq \lambda_c.$$

The accuracy requirements consist of upper bounds for backward as well as forward error:

$$\eta_{F,2}^{H}(\widetilde{\lambda}, \widetilde{x}) \leq \eta_{\max},\ |\widetilde{\lambda} - \lambda| \leq \Delta\lambda_{\max}.$$

Note that the multilevel eigensolver is able to to approximate a certain number of eigenpairs as well as finding the eigenpairs with the largest eigenvalues of a pencil if the pencil (M, K) is considered instead.

4.5.3 More Robust AMLS with Intermediate GEP Solves

The success of the AMLS method depends massively on the choice of the parameter c_s: if c_s is too small, the eigenvalue approximations are unusable but if c_s is too large, the computed

Input: $K, M \in \mathbb{C}^{n,n}$ HPSD, $\lambda_c > 0$, weighting $0 \leq \lambda_w \leq \lambda_c$, $0 < \eta_{\max} \leq 1$
Output: Approximate eigenpairs $(\widetilde{\lambda}, \widetilde{x})$, where $\widetilde{\lambda} \leq \lambda_c$ and $\eta_{F,2}^H(\widetilde{\lambda}, \widetilde{x}) \leq \eta_{\max}$

```
function MULTILEVEL-GEP-SOLVE(K, M, λ_c, λ_w, η_max)
  # Terminate recursion?
  if n is small enough then
      Solve KX = MXΛ directly
      return (Λ, X)
  end if

  # Divide: subdivide problem
  Compute undirected graph G, see Section 4.4.4
  Partition matrices, get permutation matrix Π
  K^Π ← Π* K Π
  M^Π ← Π* M Π

  # Conquer: recursion
  Compute maximum backward error η'_max for subproblems
  Λ̃_1, X̃_1 ← MULTILEVEL-GEP-SOLVE(K^Π_11, M^Π_11, λ_c, λ_w, η'_max)
  Λ̃_2, X̃_2 ← MULTILEVEL-GEP-SOLVE(K^Π_22, M^Π_22, λ_c, λ_w, η'_max)

  # Combine: construct accurate eigenpairs of (K, M)
  S ← diag(X̃_1, X̃_2)
  Improve the search space ran S
  Execute Rayleigh-Ritz procedure: Λ̃, X̃ ← RAYLEIGH-RITZ(K, M, S)

  return Λ̃, ΠX̃
end function
```

Algorithm 8: Pseudocode for a multilevel eigensolver for generalized eigenvalue problems

search space may be extremely large. Based on the observations in this section, we suggest a small change in AMLS to reduce the dependence on c_s.

AMLS is a method based on the divide-and-conquer paradigm. In order to highlight this point, we rewrote the AMLS pseudocode from Algorithm 7 as a divide-and-conquer algorithm in Algorithm 9 (the assembly of the matrices K^L, M^L, and L is explained below). Comparing the pseudocode of the multilevel eigensolver with AMLS reveals there are only two major differences between these two methods. The first difference is that the AMLS method decomposes the stiffness matrix whereas the multilevel eigensolver does not. Consequently, the AMLS method must use a fill-in reducing ordering whereas the multilevel eigensolver is free to use any ordering. Moreover, AMLS has to perform additional congruence transformation not present in the multilevel eigensolver. The second major difference can be found in the combine phase of the solvers. Ignoring the assembly of the matrices K^L, M^L, and L related to the stiffness matrix decomposition, AMLS does apparently not utilize the Rayleigh-Ritz procedure or spectral approximation methods.

We think this is a real drawback because applying the Rayleigh-Ritz procedure during the recursion improves the eigenpair approximations and most importantly, it helps the solver to recognize excessively large search spaces. Furthermore, it makes AMLS more robust in the presence of strongly clustered eigenvalues. We conjecture, AMLS can be turned into a robust, true black-box solver with this modification. We want to mention, the Rayleigh-Ritz procedure does not have to be executed in every recursive AMLS call–just often enough to prevent the worst-case scenarios discussed above.

In this paragraph we discuss the recursive computation of the block matrices in Algorithm 9. The LDL^T decomposition of the stiffness matrix K^Π yields the lower unit triangular matrix L and the diagonal matrix $D = K^L$ (unit triangular meaning with ones on the diagonal). Given L_{11}, L_{22}, K^L_{11}, and K^L_{22}, the matrix K^L can be completed:

$$\begin{aligned} K^L_{33} &= K^\Pi_{33} - K^\Pi_{31}(K^\Pi_{11})^{-1}(K^\Pi_{31})^* - K^\Pi_{32}(K^\Pi_{22})^{-1}(K^\Pi_{32})^*, \\ K^L &= \operatorname{diag}(D_{11}, D_{22}, D_{33}). \end{aligned}$$

Keep in mind that $K^\Pi_{ii} = L_{ii}K^L_{ii}L^*_{ii}$, $i = 1, 2$. In practice we should store the Cholesky decomposition of $D_{ii} = K^L_{ii}$ instead of the diagonal block itself since their inverses are needed repeatedly. Now we can calculate the missing entries of L:

$$\begin{aligned} L_{31} &= K^\Pi_{31}L^{-*}_{11}D^{-1}_{11}, \\ L_{32} &= K^\Pi_{32}L^{-*}_{22}D^{-1}_{11}, \\ L_{33} &= I. \end{aligned}$$

These calculations allow us to assemble L:

$$L = \begin{bmatrix} L_{11} & & \\ & L_{22} & \\ L_{31} & L_{32} & L_{33} \end{bmatrix}.$$

At last, we can compute entries of M^L:

$$\begin{aligned} M^L_{31} &= M_{31}L^{-*}_{11} - L_{31}M^L_{11}, \\ M^L_{32} &= M_{32}L^{-*}_{22} - L_{32}M^L_{22}, \\ M^L_{33} &= M_{33} - M^L_{31}L^*_{31} - M^L_{32}L^*_{32} - L_{31}L^{-1}_{11}M^*_{31} - L_{32}L^{-1}_{22}M^*_{32}. \end{aligned}$$

Thus,

$$M^L = \begin{bmatrix} M_{11}^L & & (M_{31}^L)^* \\ & M_{22}^L & (M_{32}^L)^* \\ M_{31}^L & M_{32}^L & M_{33}^L \end{bmatrix}.$$

4.6 The Multilevel Eigensolver in Practice

We implemented the multilevel eigensolver from Section 4.5 in Python 2 using Intel MKL, NumPy, SciPy, and the graph partitioning software METIS [KK98]. We use the backward error $\eta_{F,2}^H(\cdot,\cdot)$ from Section 2.1 and its corresponding condition number $\kappa_{F,2}(\cdot,\cdot)$. Dense GEPs are solved by the deflation solver from Chapter 3, we balance the matrix pencil using the method in [LvD06] (see Section 4.2), and we improve approximate eigenpairs using the Cayley transformation $(K-\sigma M)^{-1}(K-\tau M)$ from Section 4.1 with $\sigma = 0$ and τ in the largest approximate eigenvalue in the search space.

4.6.1 Adaptive Backward Error Control is Unnecessary

One of the findings used for the multilevel eigensolver was the theory from Section 4.4.5 to control the backward error in the subproblems, i. e., given the Frobenius norm of the off-diagonal blocks of mass and stiffness matrix, we were able to bound the expected backward error of an approximate eigenpair under certain conditions. To avoid unnecessary computations, the idea was to avoid reducing the backward error of the approximate eigenpairs $(\widetilde{\lambda}_i, \widetilde{x}_i)$ of the subproblems below the expected backward error, where $\widetilde{\lambda}_i$ was less or equal to the largest desired eigenvalue:

$$\widetilde{\lambda}_i \leq \lambda_c \Rightarrow \eta_{F,2}^H(\widetilde{\lambda}_i, \widetilde{x}_i) \overset{!}{\leq} \eta_{\max}.$$

In the subproblems the smallest eigenvalues are approximated from above for a GEP with HPSD matrices and initially, the solver did not respect this property. In fact, the solver removed an approxmate eigenpair $(\widetilde{\lambda}_i, \widetilde{x}_i)$ from the search space when $\widetilde{\lambda}_i > c_s\lambda_c$ where $c_s \geq 1$ is a given constant. This caused the eigensolver to miss desired eigenpairs and in order to fix this issue, we made the implementation control the backward *and* forward error, i. e., every approximate eigenpair had to satisfy the following conditions if the approximate eigenvalue was below the cutoff:

$$\widetilde{\lambda}_i \leq \lambda_c \Rightarrow \eta_{F,2}^H(\widetilde{\lambda}_i, \widetilde{x}_i) \overset{!}{\leq} \eta_{\max},\ \Delta\widetilde{\lambda}_i/\widetilde{\lambda}_i \overset{!}{\leq} 1,$$

where $\Delta\lambda_i$ is the forward error of $\widetilde{\lambda}_i \neq 0$.

In our implementation and for our test problems, reducing the relative forward error below one sped up the solver and reduced the number of test problems with missing desired eigenpairs. As a side effect of the forward error control, the backward error of the approximate eigenpairs was almost always below the single precision epsilon $\varepsilon \approx 1.19 \cdot 10^{-7}$. The single precision epsilon in turn is in every test problem much smaller than the maximum allowed backward error. Clearly, adaptive backward error control is superfluous.

4.6.2 Bisection is Unnecessary

In Section 4.4.4, we derived a method to minimize the Frobenius norm of the off-diagonal blocks in a Hermitian 2×2 block matrix based on graph bisection. Disabling bisection had no significant effect on the multilevel GEP solver in our test problems.

Input: $K, M \in \mathbb{C}^{n,n}$ HPD, $\lambda_c > 0$, $c_s \geq 1$
Output: permutation matrix Π, matrix L, K^L, M^L, approximate eigenpairs $(\widetilde{\lambda}, \widetilde{x})$

function AMLS(K, M, λ_c, c_s)

# Terminate recursion?
if n is small enough **then**
Solve $KX = MX\Lambda$ directly
Perform modal truncation, get Λ', X'
return I, I, K, M, Λ', X'
end if

Divide: subdivide problem
Compute nested dissection ordering P with two substructure blocks
$K^P \leftarrow P^* K P$
$M^P \leftarrow P^* M P$

Conquer: recursion
$\Pi_{11}, L_{11}, K_{11}^L, M_{11}^L, \widetilde{\Lambda}_1, \widetilde{X}_1 \leftarrow \text{AMLS}(K_{11}^P, M_{11}^P, \lambda_c, c_s)$
$\Pi_{22}, L_{22}, K_{22}^L, M_{22}^L, \widetilde{\Lambda}_2, \widetilde{X}_2 \leftarrow \text{AMLS}(K_{22}^P, M_{22}^P, \lambda_c, c_s)$

Combine: combine permutations
$\Pi \leftarrow \text{diag}(\Pi_{11}, \Pi_{22}, I)$
$K^\Pi \leftarrow \Pi^* K^P \Pi$
$M^\Pi \leftarrow \Pi^* M^P \Pi$

Combine: complete matrices (see text for details)
Complete K^L
Complete L
Complete M^L

Combine: compute eigendecomposition
Solve $K_{33}^L X_3^L = M_{33}^L X_3^L \Lambda_3^L$
Perform modal truncation, get $\widetilde{\Lambda}_3$, $\widetilde{X}_3$

$\widetilde{X} \leftarrow \text{diag}(\widetilde{X}_1, \widetilde{X}_2, \widetilde{X}_3)$
$\widetilde{\Lambda} \leftarrow \text{diag}(\widetilde{\Lambda}_1, \widetilde{\Lambda}_2, \widetilde{\Lambda}_3)$

return $P\Pi$, L, K^L, M^L, $\widetilde{\Lambda}$, $\widetilde{X}$
end function

Algorithm 9: Pseudocode for the automated multilevel substructuring method (AMLS) as divide-and-conquer algorithm. The computation of the matrices K^L, M^L, and L are explained in the text.

4.6.3 Solving System of Linear Equations with the Schur Complement

The eigensolver improves the approximate eigenpairs by means of subspace iterations. In every iteration, systems of linear equations (SLEs) $Kx = b$, $x, b \in \mathbb{C}^{n,n}$, need to be solved and because there may be a large number of different right-hand side vectors b, the author preferred direct solvers in the implementation. Initially, the solver used sparse LU decompositions provided by SuperLU [Li05] but for the larger test matrices, e. g., bmwcra_st1 ($n = 148,700$), the matrix factors had a memory footprint of several gigabytes. We were willing to trade time for predictable memory demands and decided to replace the sparse LU decompositions with direct substructuring [SBG96, §4.1] using sequences of Schur complements. Nowadays, iterative substructuring methods are more common.

Definition 4.4 (Schur complement [HJ12, §0.25])**.** Let $A \in \mathbb{C}^{n,n}$ be Hermitian and partitioned as as a 2×2 block matrix, where none of the blocks are empty:

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}.$$

Let A_{11} be non-singular. The matrix

$$S \coloneqq A_{22} - A_{12}^* A_{11}^{-1} A_{12}$$

is called the *Schur complement of A_{11} in A*.

The Schur complement occurs naturally during block Gaussian elimination:

$$\begin{bmatrix} I & 0 \\ -A_{21}A_{11}^{-1} & I \end{bmatrix} \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} I & -A_{11}^{-1}A_{21}^* \\ 0 & I \end{bmatrix} = \begin{bmatrix} A_{11} & 0 \\ 0 & S \end{bmatrix}.$$

Consider the SLE $Ax = b$ and partition x, b conformally to A such that

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}.$$

There exists a closed expression for x using the inverse of A_{11} and its Schur complement S. Let L be the lower triangular matrix block diagonalizing A:

$$L = \begin{bmatrix} I & 0 \\ -A_{21}A_{11}^{-1} & I \end{bmatrix}.$$

Then x is

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} A_{11}^{-1}b_1 - A_{11}^{-1}A_{21}^* S^{-1}(b_2 - A_{21}A_{11}^{-1}b_1) \\ S^{-1}(b_2 - A_{21}A_{11}^{-1}b_1) \end{bmatrix}.$$

Observe that in order to solve the SLE $Ax = b$, we need to be able to solve SLEs $A_{11}x' = b'$ and $Sx'' = b''$ but there is no need to store L or explicitly factorize A_{11}. So far we did not explain how we solve SLEs $A_{11}x' = b'$. If A_{11} is small enough, the we can solve this SLE directly. Otherwise we can use the Schur complement again, e. g., we partition A_{11} into a 2×2 block matrix and compute the Schur complement of the upper left block in A_{11}. This approach yields a recursive method that can be found in and Algorithm 10.

In the previous section, we noted that the matrix ordering computed by graph bisection did not have a significant positive effect on the multilevel eigensolver. Now let us interpret this result liberally to mean that the multilevel GEP solver is not affected by matrix orderings *in*

Input: $A \in \mathbb{C}^{n,n}$ full rank, Hermitian with 2×2 block structure, conformally partitioned $b \in \mathbb{C}^{n,n}$
Output: The Schur complement of A_{11}

```
function SOLVE-SLE(A, b)
    if A is small enough then
        Solve Ax = b directly
        return x
    end if

    # Compute the Schur complement
    Solve A_11 P = A_12: P ← SOLVE-SLE(A_11, A_12)
    S ← A_22 − A_21 P

    # Solve Ax = b
    Solve A_11 u = b_1: u ← SOLVE-SLE(A, b_1)
    v ← b_2 − A_21 u:
    Solve S x_2 = v directly
    Solve A_11 w = A_12 x_2: w ← SOLVE-SLE(A_11, v)
    x_1 ← u − w

    return x
end function
```

Algorithm 10: Solving SLEs by recursively computing Schur complements

practice and consider that we recursively compute nested dissection orderings for K, K_{11}, K_{22}, and so on such that

$$\begin{bmatrix} K_{11} & 0 & K_{13} \\ 0 & K_{22} & K_{23} \\ K_{31} & K_{32} & K_{33} \end{bmatrix}.$$

If K_{33} is small in dimension compared to K, then we can permanently store *all* Schur complements needed in Algorithm 10 without too much overhead. Moreover, if there are multiple right-hand sides b, then the cost of computing the Schur complements amortizes over the run time of the solver.

4.6.4 Numerical Experiments

All test matrices are part of the University of Florida Sparse Matrix Collection [DH11]. Note that only the test problems with the stiffness matrices gyro_k and bcsstk36 come with a corresponding mass matrix (gyro_m, bcsstm36); all other test problems are unfortunately standard eigenvalue problems.

We tested the implementation on a computer with 8 GB RAM and an AMD Athlon II X2 270 CPU. The results can be seen in Table 4.1. The first column contains the name of the stiffness matrix of every test problem and the second column shows the number of degrees of freedom. The third column contains the cutoff values $\lambda_c > 0$ while the fourth column shows the number of computed eigenpairs (λ, x) with $\lambda \leq \lambda_c$. WC time is the *wall-clock time* taken by the solver and the last column contains the CPU time.[1] Wall-clock time and CPU time are given in minutes.

[1] CPU time is the amount of time the CPU was busy executing instructions of a program. For computers with more than one CPU core, this value may be larger than the wall-clock time.

Problem	n	λ_c	n_c	WC time	CPU time
bcsstk17	10,974	$1.0{\cdot}10^{2}$	518	0.3	0.5
bcsstk18	11,948	$2.0{\cdot}10^{1}$	951	1.5	2.7
bcsstk25	15,439	$1.0{\cdot}10^{2}$	929	3.9	7.2
gyro_k	17,361	$1.0{\cdot}10^{15}$	431	10.0	17.7
bcsstk36	23,052	$1.0{\cdot}10^{6}$	1,073	11.2	20.1
vanbody	47,072	$2.0{\cdot}10^{1}$	1,265	22.8	40.9
ct20stif	52,329	$2.0{\cdot}10^{3}$	628	27.3	49.4
bmw7st_1	141,347	$1.0{\cdot}10^{-2}$	79	191.0	370.5
bmwcra_1	148,770	$1.0{\cdot}10^{3}$	39	121.0	235.0

Table 4.1: The test results of a multilevel GEP solver implementation with direct substructuring on a computer with dual-core CPU and 8 GB RAM. All times are given in minutes.

We consider an approximate eigenpair converged if its backward error is less than the single precision epsilon $\varepsilon \approx 1.19 \cdot 10^{-7}$ and if its relative forward error is less than one.

Observe that there is a jump in the time taken by the solver when computing eigenpairs of the matrix pair (gyro_k, gyro_m). The multilevel GEP solver selects the eigenvectors spanning the search space based on their corresponding eigenvalue and for this pencil, the eigenvalues are densely clustered. For example, the dimension of the final search space of this pencil is 2356 and there are 431 desired eigenpairs while for the matrix pair (bcsstk36, bcsstm36) the final search space has dimension 2664 containing more than 1,000 desired eigenpairs. The solver also took an unusually long time to completion when finding eigenpairs of the stiffness matrix bcw7st_1. Here, the solver struggled and failed to reduce the relative forward error below 1 with ten subspace iterations. Usually, the solver requires no more than three subspace iterations in any subproblem to meet the convergence criterion. We did not count this as a solver failure because all eigenpairs had a backward error less than the double precision epsilon.

We also tested the implementation on a cluster node with two AMD Opteron 2218 CPUs and 16 GB virtual memory limit. The results can be found in Table 4.2. We used the additional memory to increase the number of desired eigenpairs to about 1,000. When computing eigenpairs of bmw7st_1, the solver tried to reduce the relative forward error below 1 and failed again to do so with ten subspace iterations taking 6.6 hours wall-clock time (the maximum backward error of the desired eigenpairs was below the double precision epsilon). With the usual three subspace iterations, the eigensolver would have taken only four hours overall for finding the desired eigenpairs. The eigensolver has to cope with a comparatively large search space of dimension 1759 when finding eigenpairs of bmwcra_1. On the upside, the wall-clock time required by the solver does not seem to be quadratic (or worse) in the number of desired eigenpairs.

Problem	n	λ_c	n_c	WC time	CPU time
bcsstk17	10,974	$1.0{\cdot}10^{2}$	518	0.5	1.2
bcsstk18	11,948	$2.0{\cdot}10^{1}$	951	2.2	7.4
bcsstk25	15,439	$1.0{\cdot}10^{2}$	929	5.7	19.6
gyro_k	17,361	$1.0{\cdot}10^{16}$	2,360	103.5	350.5
bcsstk36	23,052	$1.0{\cdot}10^{6}$	1,073	17.4	54.8
vanbody	47,072	$2.0{\cdot}10^{1}$	1,265	34.1	110.5
ct20stif	52,329	$5.0{\cdot}10^{3}$	947	61.0	195.9
bmw7st_1	141,347	$1.0{\cdot}10^{0}$	1,053	499.8	1,661.7
bmwcra_1	148,770	$1.0{\cdot}10^{4}$	124	623.3	2,038.9

Table 4.2: The test results of a multilevel GEP solver implementation with direct substructuring on a computer with two dual-core CPUs and 16 GB RAM. All times are given in minutes.

5 Conclusion

In thesis we discussed the numerical solution of generalized eigenvalue problem (GEPs) $Kx = \lambda Mx$, where K, M are Hermitian positive semidefinite (HPSD). All results were directly applicable to real-world problems.

In Chapter 2 we presented a Hermiticity-preserving backward error for simple eigenvalues that can be computed in linear time and its corresponding condition number. We elaborated on the finite element method (FEM) as a source of GEPs with HPSD matrices.

In Chapter 3 we discussed solvers for dense GEPs with HPSD matrices. The standard solver is fast and preserves Hermiticity but it requires positive definite mass matrices and is only conditionally stable. Deflating the infinite eigenvalues allows the standard solver to be used for problems with singular mass matrices, as well, but it does not guarantee a small backward error either and the deflation procedure cannot handle singular matrix pencils. Solvers based on the generalized singular value decomposition (GSVD) are backward stable, they preserve Hermiticity and semidefiniteness, and they automatically determine the regular part of a GEP. We implemented these solvers and found out that the standard solver always computes accurate solutions in our test with real-world matrices if the infinite eigenvalues are deflated from the matrix pencil. The deflation overhead is small in our tests. Moreover, in all test problems, the fastest GSVD solver was no more than five times slower than the fastest solver. As a byproduct of our benchmarks, we determined computing the GSVD by means of QR factorizations and the CS decomposition is much faster than directly calculating the GSVD in Netlib LAPACK. We also showed how the deflation procedure must be modified in order to be able to handle singular matrix pencils.

In Chapter 4 we discussed ways to compute all eigenpairs (λ, x), $\lambda \leq \lambda_c$, with large, sparse, Hermitian positive definite (HPD) matrices, where $\lambda_c > 0$ is a user-provided value. We briefly presented standard methods for spectral of approximation of large, sparse matrices and methods for improving numerical stability. Afterwards we reviewed the automated multilevel substructuring method (AMLS). AMLS is often able to quickly deliver good approximations to the eigenpairs (λ, x), where $\lambda \leq \lambda_c$. Next we analyzed the perturbation of exact eigenvalues of blocks on the diagonal caused by off-diagonal blocks in 2×2 block matrices and we acquired slightly stronger perturbation bounds if eigenvectors are available. When AMLS computes the approximate eigenvalues, the stiffness matrix is block diagonal and with the analysis of perturbation bounds of 2×2 block matrices, we were able to conclude that this property is helpful when approximating small eigenvalues. Furthermore, we described a method to minimize eigenvalue perturbation by off-diagonal blocks and we calculated its impact on the backward error.

Finally, we used the results to propose a new multilevel eigensolver based on the divide-and-conquer paradigm, the perturbation results for 2×2 block matrices, and the dense GEP solvers from Chapter 3. We implemented the solver and tested it with large, sparse matrices arising from finite element discretizations in structural engineering. We found out that we can ignore the results on eigenvalue perturbation in 2×2 block matrices. Nevertheless, on a computer with a dual-core CPU and 8 GB RAM, the solver calculated eigenpairs of problems with up to 150,000 degrees of freedom in about three hours. On a cluster node with two dual-core CPUs and 16 GB virtual memory limit, we computed up to 1,000 eigenpairs on the same set of problems in less than eleven hours.

Bibliography

[AA11] B. Adhikari and R. Alam. "On backward errors of structured polynomial eigenproblems solved by structure preserving linearizations". In: *Linear Algebra and its Applications* 434.9 (2011), pp. 1989–2017. ISSN: 0024-3795. DOI: 10.1016/j.laa.2010.12.014.

[AAK11] B. Adhikari, R. Alam, and D. Kressner. "Structured eigenvalue condition numbers and linearizations for matrix polynomials". In: *Linear Algebra and its Applications* 435.9 (2011), pp. 2193–2221. ISSN: 0024-3795. DOI: 10.1016/j.laa.2011.04.020.

[ASNA] N. J. Higham. *Accuracy and Stability of Numerical Algorithms*. 2nd ed. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2002. ISBN: 978-0-89871-521-7. DOI: 10.1137/1.9780898718027.

[Bai+00] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst, eds. *Templates for the Solution of Algebraic Eigenvalue Problems. A Practical Guide*. Software, Environments and Tools. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2000. ISBN: 978-0-89871-471-5. DOI: 10.1137/1.9780898719581. URL: http://web.cs.ucdavis.edu/~bai/ET/contents.html.

[Bai92] Z. Bai. *The CSD, GSVD, Their Applications and Computations*. IMA Preprint Series 958. Minneapolis, MN, USA: University of Minnesota, 1992. HDL: 11299/1875.

[Bat96] K.-J. Bathe. *Finite Element Procedures*. Upper Saddle River, NJ, USA: Prentice-Hall, 1996. ISBN: 978-0-13-301458-7.

[BD92] Z. Bai and J. W. Demmel. *Computing the Generalized Singular Value Decomposition*. 1992. LAPACK Working Note 46.

[Bet+13] T. Betcke, N. J. Higham, V. Mehrmann, C. Schröder, and F. Tisseur. "NLEVP: A Collection of Nonlinear Eigenvalue Problems". In: *ACM Transactions on Mathematical Software* 39.2 (2013), 7:1–7:28. ISSN: 0098-3500. DOI: 10.1145/2427023.2427024.

[BL04] J. K. Bennighof and R. B. Lehoucq. "An Automated Multilevel Substructuring Method for Eigenspace Computation in Linear Elastodynamics". In: *SIAM Journal on Scientific Computing* 25.6 (2004), pp. 2084–2106. ISSN: 1064-8275. DOI: 10.1137/S1064827502400650.

[BM12] A. M. Bradley and W. Murray. *Matrix-Free Approximate Equilibration*. Stanford, CA, USA, 2012. arXiv: 1110.2805v2 [math.NA].

[Bul+15] A. Buluç, H. Meyerhenke, I. Safro, P. Sanders, and C. Schulz. *Recent Advances in Graph Partitioning*. 2015. arXiv: 1311.3144v3 [cs.DS].

[BZ93] Z. Bai and H. Zha. "A New Preprocessing Algorithm for the Computation of the Generalized Singular Value Decomposition". In: *SIAM Journal on Scientific Computing* 14.4 (1993), pp. 1007–1012. ISSN: 1064-8275. DOI: 10.1137/0914060.

[CB68] R. R. Craig Jr. and M. C. C. Bampton. "Coupling of Substructures for Dynamic Analyses". In: *AIAA Journal* 6.7 (1968), pp. 1313–1319. ISSN: 0001-1452. DOI: 10.2514/3.4741.

[CMM16] C. Conrads, V. Mehrmann, and A. Międlar. "Adaptive numerical solution of eigenvalue problems arising from finite element models. AMLS vs. AFEM". In: *A Panorama of Mathematics. Pure and Applied*. Ed. by C. M. da Fonseca, D. V. Huynh, S. Kirkland, and V. K. Tuan. Contemporary Mathematics 658. Providence, RI, USA: American Mathematical Society, 2016, pp. 197–226. ISBN: 978-1-4704-1668-3. DOI: 10.1090/conm/658/13127.

[Coo+01] R. D. Cook, D. S. Malkus, M. E. Plesha, and R. J. Witt. *Concepts and Application of Finite Element Analysis*. 4th ed. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2001. ISBN: 978-0-471-35609-5.

[Cor+09] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. 3rd ed. Boston, MA, USA: MIT Press, 2009. ISBN: 978-0-262-03384-8.

[DB08] Z. Drmač and Z. Bujanović. "On the Failure of Rank-Revealing QR Factorization Software. A Case Study". In: *ACM Transactions on Mathematical Software* 35.2 (2008), 12:1–12:28. ISSN: 0098-3500. DOI: 10.1145/1377612.1377616.

[Dem+07] J. W. Demmel, O. A. Marques, B. N. Parlett, and C. Vömel. *Performance and Accuracy of LAPACK's Symmetric Tridiagonal Eigensolvers*. 2007. LAPACK Working Note 183.

[Dem97] J. W. Demmel. *Applied Numerical Linear Algebra*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 1997. ISBN: 978-0-898713-89-3.

[DGL89] I. Duff, R. Grimes, and J. Lewis. "Sparse Matrix Test Problems". In: *ACM Transactions on Mathematics Software* 15.1 (1989), pp. 1–14. ISSN: 0098-3500. DOI: 10.1145/62038.62043.

[DH11] T. A. Davis and Y. Hu. "The University of Florida Sparse Matrix Collection". In: *ACM Transactions on Mathematical Software* 38.1 (2011), 1:1–1:25. ISSN: 0098-3500. DOI: 10.1145/2049662.2049663.

[Dhi97] I. S. Dhillon. "A New $\mathcal{O}(n^2)$ Algorithm for the Symmetric Tridiagonal Eigenvalue/Eigenvector Problem". PhD thesis. Berkeley, CA, USA: University of California, Berkeley, 1997. URL: http://www.cs.utexas.edu/users/inderjit/public_papers/thesis.pdf.

[DM02] E. D. Dolan and J. J. Moré. "Benchmarking optimization software with performance profiles". In: *Mathematical Programming* 91.2 (2002), pp. 201–213. ISSN: 0025-5610. DOI: 10.1007/s101070100263.

[Fra61] J. G. F. Francis. "The QR Transformation. A Unitary Analogue to the LR Transformation – Part 1". In: *The Computer Journal* 4.3 (1961), pp. 265–271. ISSN: 0010-4620. DOI: 10.1093/comjnl/4.3.265.

[FSV98] D. R. Fokkema, G. L. G. Sleijpen, and H. A. Van der Vorst. "Jacobi–Davidson Style QR and QZ Algorithms for the Reduction of Matrix Pencils". In: *SIAM Journal on Scientific Computing* 20.1 (1998), pp. 94–125. ISSN: 1064-8275. DOI: 10.1137/S1064827596300073.

[Gao+08] W. Gao, X. S. Li, C. Yang, and Z. Bai. "An Implementation and Evaluation of the AMLS Method for Sparse Eigenvalue Problems". In: *ACM Transactions on Mathematical Software* 34.4 (2008), 20:1–20:28. ISSN: 0098-3500. DOI: 10.1145/1377596.1377600.

[Gar+03] S. D. Garvey, F. Tisseur, M. I. Friswell, J. E. T. Penny, and U. Prells. "Simultaneous tridiagonalization of two symmetric matrices". In: *International Journal for Numerical Methods in Engineering* 57.12 (2003), pp. 1643–1660. ISSN: 1097-0207. DOI: 10.1002/nme.733.

[GBP04] D. Givoli, P. E. Barbone, and I. Patlashenko. "Which are the important modes of a subsystem?" In: *International Journal for Numerical Methods in Engineering* 59.12 (2004), pp. 1657–1678. ISSN: 1097-0207. DOI: 10.1002/nme.935.

[Geo73] A. George. "Nested Dissection of a Regular Finite Element Mesh". In: *SIAM Journal on Numerical Analysis* 10.2 (1973), pp. 345–363. ISSN: 0036-1429. DOI: 10.1137/0710032.

[Gir+05] L. Giraud, J. Langou, M. Rozložník, and J. van den Eshof. "Rounding error analysis of the classical Gram-Schmidt orthogonalization process". In: *Numerische Mathematik* 101.1 (2005), pp. 87–100. ISSN: 0029-599X. DOI: 10.1007/s00211-005-0615-4.

[GRS07] C. Grossmann, H.-G. Roos, and M. Stynes. *Numerical Treatment of Partial Differential Equations*. Universitext. Translated and revised from the 3rd (2005) German edition by Martin Stynes. Berlin, Germany: Springer-Verlag, 2007. ISBN: 978-3-540-71582-5. DOI: 10.1007/978-3-540-71584-9.

[HH98] D. J. Higham and N. J. Higham. "Structured Backward Error and Condition of Generalized Eigenvalue Problems". In: *SIAM Journal on Matrix Analysis and Applications* 20.2 (1998), pp. 493–512. ISSN: 0895-4798. DOI: 10.1137/S0895479896313188. MIMS EPrint 343.

[HHL07] S. Hammarling, N. J. Higham, and C. Lucas. *LAPACK-Style Codes for Pivoted Cholesky and QR Updating*. Manchester, UK, 2007. MIMS EPrint 689.

[HJ12] R. A. Horn and C. R. Johnson. *Matrix Analysis*. 2nd ed. New York, NY, USA: Cambridge University Press, 2012. ISBN: 978-0-521-54823-6.

[HL07] U. L. Hetmaniuk and R. B. Lehoucq. "Multilevel Methods for Eigenspace Computations in Structural Dynamics". In: *Domain Decomposition Methods in Science and Engineering XVI*. Ed. by O. B. Widlund and D. E. Keyes. Lecture Notes in Computational Science and Engineering 55. Berlin, Germany: Springer-Verlag, 2007, pp. 103–113. ISBN: 978-3-540-34468-1. DOI: 10.1007/978-3-540-34469-8_9.

[JKL99] H.-J. Jung, M.-C. Kim, and I.-W. Lee. "An improved subspace iteration method with shifting". In: *Computer & Structures* 70.6 (1999), pp. 625–633. ISSN: 0045-7949. DOI: 10.1016/S0045-7949(98)00201-6.

[JL99] H.-J. Jung and I.-W. Lee. "An improved subspace iteration method with shift for structures with multiple natural frequencies". In: *Journal of Sound and Vibration* 227.2 (1999), pp. 271–291. ISSN: 0022-460X. DOI: 10.1006/jsvi.1999.2344.

[Kan+14] R. Kannan, S. Hendry, N. J. Higham, and F. Tisseur. "Detecting the causes of ill-conditioning in structural finite element models". In: *Computers & Structures* 133 (2014), pp. 79–89. ISSN: 0045-7949. DOI: 10.1016/j.compstruc.2013.11.014. MIMS EPrint 1997.

[Kap01] M. F. Kaplan. "Implementation of Automated Multilevel Substructuring for Frequency Response Analysis of Structures". PhD thesis. Austin, TX, USA: University of Texas at Austin, 2001. HDL: 2152/10611.

[KK98] G. Karypis and V. Kumar. "A Fast and Highly Quality Multilevel Scheme for Partitioning Irregular Graphs". In: *SIAM Journal on Scientific Computing* 20.1 (1998), pp. 359–392. ISSN: 1064-8275. DOI: 10.1137/S1064827595287997.

[Kny01] A. V. Knyazev. "Toward the Optimal Preconditioned Eigensolver. Locally Optimal Block Preconditioned Conjugate Gradient Method". In: *SIAM Journal on Scientific Computing* 23.2 (2001), pp. 517–541. ISSN: 1064-8275. DOI: 10.1137/S1064827500366124.

[Kre11] E. Kreyszig. *Advanced Engineering Mathematics*. 10th ed. Hoboken, NJ, USA: Wiley, 2011. ISBN: 978-0-470-45836-5.

[Lan+06] J. Langou, J. Langou, P. Luszczek, J. Kurzak, A. Buttari, and J. Dongarra. "Exploiting the Performance of 32 bit Floating Point Arithmetic in Obtaining 64 bit Accuracy. Revisiting Iterative Refinement for Linear Systems". ICL Friday Lunch Talk by Julie Langou. May 19, 2006. URL: http://icl.cs.utk.edu/projectsfiles/iter-ref/files/iter-ref.pdf.

[Li+11] R.-C. Li, Y. Nakatsukasa, N. Truhar, and S. Xu. "Perturbation of Partitioned Hermitian Definite Generalized Eigenvalue Problems". In: *SIAM Journal on Matrix Analysis and Applications* 32.2 (2011). See also erratum [Li+13], pp. 642–663. ISSN: 0895-4798. DOI: 10.1137/100808356.

[Li+13] R.-C. Li, Y. Nakatsukasa, N. Truhar, and S. Xu. "Erratum: Perturbation of Partitioned Hermitian Definite Generalized Eigenvalue Problems". In: *SIAM Journal on Matrix Analysis and Applications* 34.1 (2013). Erratum for [Li+11], pp. 280–281. ISSN: 0895-4798. DOI: 10.1137/120874795.

[Li05] X. S. Li. "An Overview of SuperLU. Algorithms, Implementation, and User Interface". In: *ACM Transactions on Mathematical Software* 31.3 (2005), pp. 302–325. ISSN: 0098-3500. DOI: 10.1145/1089014.1089017.

[LRT79] R. J. Lipton, D. J. Rose, and R. E. Tarjan. "Generalized Nested Dissection". In: *SIAM Journal on Numerical Analysis* 16.2 (1979), pp. 346–358. ISSN: 0036-1429. DOI: 10.1137/0716027.

[LvD06] D. Lemonnier and P. van Dooren. "Balancing Regular Matrix Pencils". In: *SIAM Journal on Matrix Analysis and Applications* 28.1 (2006), pp. 253–263. ISSN: 0895-4798. DOI: 10.1137/S0895479804440931.

[MC] G. H. Golub and C. F. Van Loan. *Matrix Computations*. 4th ed. Baltimore, MD, USA: Johns Hopkins University Press, 2012. ISBN: 978-1-4214-0794-4.

[McC94] S. F. McCormick, ed. *Multigrid Methods*. Frontiers in Applied Mathematics. Second printing. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 1994. ISBN: 978-1-61197-188-0. DOI: 10.1137/1.9781611971057.

[MMW15] C. Mehl, V. Mehrmann, and M. Wojtylak. "On the Distance to Singularity via Low Rank Perturbations". In: *Operators and Matrices* 9.4 (2015), pp. 733–772. ISSN: 1846-3886. DOI: 10.7153/oam-09-44.

[MX15] V. Mehrmann and H. Xu. "Structure preserving deflation of infinite eigenvalues in structured pencils". In: *Electronic Transactions on Numerical Analysis* 44 (2015), pp. 1–24. ISSN: 1068-9613. URL: http://etna.mcs.kent.edu/volumes/2011-2020/vol44/.

[Nak12] Y. Nakatsukasa. "On the condition numbers of a multiple eigenvalue of a generalized eigenvalue problem". In: *Numerische Mathematik* 121.3 (2012), pp. 531–544. ISSN: 0029-599X. DOI: 10.1007/s00211-011-0440-x.

[NSV09] R. H. Nochetto, K. G. Siebert, and A. Veeser. "Theory of adaptive finite element methods. An introduction". In: *Multiscale, Nonlinear and Adaptive Approximation*. Ed. by R. DeVore and A. Kunoth. Berlin, Germany: Springer-Verlag, 2009, pp. 409–542. ISBN: 978-3-642-03412-1. DOI: 10.1007/978-3-642-03413-8_12.

[Par74] N. B. Parlett. "The Rayleigh Quotient Iteration and Some Generalizations for Nonnormal Matrices". In: *Mathematics of Computation* 28.127 (1974), pp. 679–693. ISSN: 0025-5718. DOI: 10.2307/2005689.

[Par98] B. N. Parlett. *The Symmetric Eigenvalue Problem*. Classics in Applied Mathematics 20. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 1998. ISBN: 978-0-89871-402-9. DOI: 10.1137/1.9781611971163.

[Roz+12] M. Rozložník, M. Tůma, A. Smoktunowicz, and J. Kopal. "Numerical stability of orthogonalization methods with a non-standard inner product". In: *BIT Numerical Mathematics* 52.4 (2012), pp. 1035–1058. ISSN: 0006-3835. DOI: 10.1007/s10543-012-0398-9.

[Saa11] Y. Saad. *Numerical Methods for Large Eigenvalue Problems. Revised Edition*. Classics in Applied Mathematics 66. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2011. ISBN: 978-1-61197-072-2. DOI: 10.1137/1.9781611970739.

[SBG96] B. F. Smith, P. E. Bjørstad, and W. D. Gropp. *Domain Decomposition. Parallel Multilevel Methods for Elliptic Partial Differential Equations*. New York, NY, USA: Cambridge University Press, 1996. ISBN: 978-0-521-49589-9.

[Sed02] R. Sedgewick. *Algorithms in C++. Part 5: Graph Algorithms*. 3rd ed. 8th printing, November 2006. Boston, MA, USA: Addison-Wesley, 2002. ISBN: 978-0-201-35118-6.

[SF73] G. Strang and G. J. Fix. *An Analysis of the Finite Element Method*. Prentice-Hall Series in Automatic Computation. Englewood Cliffs, NJ, USA: Prentice-Hall, 1973. ISBN: 0-13-032946-0.

[Smi97] B. F. Smith. "Domain Decomposition Methods for Partial Differential Equations". In: *Parallel Numerical Algorithms*. Ed. by D. E. Keyes, A. Sameh, and V. Venkatakrishnan. ICASE/LaRC Interdisciplinary Series in Science and Engineering 4. Berlin, Germany: Springer-Verlag, 1997, pp. 225–243. ISBN: 978-94-010-6277-0. DOI: 10.1007/978-94-011-5412-3_8.

[Sta05] A. Stathopoulos. *Locking issues for finding a large number of eigenvectors of hermitian matrices*. Tech. rep. WM-CS-2005-09. Revised June 2006. Williamsburg, VA, USA: College of William & Mary, 2005.

[Ste01] G. W. Stewart. *Matrix Algorithms*. Vol. 2: *Eigensystems*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2001. ISBN: 978-0-898714-14-2.

[Sut09] B. D. Sutton. "Computing the complete CS decomposition". In: *Numerical Algorithms* 50.1 (2009), pp. 33–65. ISSN: 1017-1398. DOI: 10.1007/s11075-008-9215-6.

[Tas15] L. Taslaman. "An Algorithm for Quadratic Eigenproblems with Low Rank Damping". In: *SIAM Journal on Matrix Analysis and Applications* 36.1 (2015), pp. 251–272. ISSN: 0895-4798. DOI: 10.1137/140969099.

[Tis04] F. Tisseur. "Tridiagonal-Diagonal Reduction of Symmetric Indefinite Pairs". In: *SIAM Journal on Matrix Analysis And Applications* 26.1 (2004), pp. 215–232. ISSN: 0895-4798. DOI: 10.1137/S0895479802414783. MIMS EPrint 467.

[War81] R. C. Ward. "Balancing the Generalized Eigenvalue Problem". In: *SIAM Journal on Scientific and Statistical Computing* 2.2 (1981), pp. 141–152. ISSN: 0196-5204. DOI: 10.1137/0902012.

[YVC13] J. Yin, H. Voss, and P. Chen. "Improving eigenpairs of automated multilevel substructuring with subspace iterations". In: *Computers & Structures* 119 (2013), pp. 115–124. ISSN: 0045-7949. DOI: 10.1016/j.compstruc.2013.01.004.